HITACHI
Inspire the Next

Hitachi Web Server

手引書

3020-3-U17-20

マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，巻末の「ソフトウェアマニュアルのサービス ご案内」をご参照ください。

■対象製品

P-2441-E184 Hitachi Web Server 04-10 （適用 OS：Windows Server 2003 Enterprise Edition, Windows Server 2003 Standard Edition, Windows Server 2003 R2 Enterprise Edition, Windows Server 2003 R2 Standard Edition, Windows Server 2003 Enterprise x64 Edition, Windows Server 2003 Standard x64 Edition, Windows Server 2003 R2 Enterprise x64 Edition, Windows Server 2003 R2 Standard x64 Edition, Windows Server 2008 Enterprise Edition, Windows Server 2008 Standard Edition, Windows Server 2008 Enterprise x64 Edition, Windows Server 2008 Standard x64 Edition, Windows Server 2008 R2 Enterprise Edition, Windows Server 2008 R2 Standard Edition）

P-1M41-E181 Hitachi Web Server 04-10 （適用 OS：AIX 5L V5.3, AIX V6.1）※

P-1J41-E181 Hitachi Web Server 04-10 （適用 OS：HP-UX 11i V2（IPF), HP-UX 11i V3（IPF））※

P-9S41-E181 Hitachi Web Server 04-10 （適用 OS：Red Hat Enterprise Linux AS 4（x86), Red Hat Enterprise Linux 5 Advanced Platform（x86), Red Hat Enterprise Linux ES 4（x86), Red Hat Enterprise Linux 5（x86), Red Hat Enterprise Linux AS 4（AMD64 & Intel EM64T), Red Hat Enterprise Linux 5 Advanced Platform（AMD/Intel 64), Red Hat Enterprise Linux ES 4（AMD64 & Intel EM64T), Red Hat Enterprise Linux 5（AMD/Intel 64））※

P-9V41-E181 Hitachi Web Server 04-10 （適用 OS：Red Hat Enterprise Linux AS 4（IPF), Red Hat Enterprise Linux 5 Advanced Platform（Intel Itanium））※

P-9D41-E181 Hitachi Web Server 04-10 （適用 OS：Solaris 9（SPARC), Solaris 10（SPARC））※

P-2443-7D84 uCosminexus Application Server Standard 08-50 （適用 OS：Windows Server 2003 Enterprise Edition, Windows Server 2003 Standard Edition, Windows Server 2003 R2 Enterprise Edition, Windows Server 2003 R2 Standard Edition, Windows Server 2003 Enterprise x64 Edition, Windows Server 2003 Standard x64 Edition, Windows Server 2003 R2 Enterprise x64 Edition, Windows Server 2003 R2 Standard x64 Edition, Windows Server 2008 Enterprise Edition, Windows Server 2008 Standard Edition, Windows Server 2008 Enterprise x64 Edition, Windows Server 2008 Standard x64 Edition, Windows Server 2008 R2 Enterprise Edition, Windows Server 2008 R2 Standard Edition, Windows Vista Business, Windows Vista Enterprise, Windows Vista Ultimate, Windows XP Professional）

P-2443-7K84 uCosminexus Application Server Enterprise 08-50 （適用 OS：Windows Server 2003 Enterprise Edition, Windows Server 2003 Standard Edition, Windows Server 2003 R2 Enterprise Edition, Windows Server 2003 R2 Standard Edition, Windows Server 2003 Enterprise x64 Edition, Windows Server 2003 Standard x64 Edition, Windows Server 2003 R2 Enterprise x64 Edition, Windows Server 2003 R2 Standard x64 Edition, Windows Server 2008 Enterprise Edition, Windows Server 2008 Standard Edition, Windows Server 2008 Enterprise x64 Edition, Windows Server 2008 Standard x64 Edition, Windows Server 2008 R2 Enterprise Edition, Windows Server 2008 R2 Standard Edition, Windows Vista Business, Windows Vista Enterprise, Windows Vista Ultimate, Windows XP Professional）

P-2443-7S84 uCosminexus Service Platform 08-50 （適用 OS：Windows Server 2003 Enterprise Edition, Windows Server 2003 Standard Edition, Windows Server 2003 R2 Enterprise Edition, Windows Server 2003 R2 Standard Edition, Windows Server 2003 Enterprise x64 Edition, Windows Server 2003 Standard x64 Edition, Windows Server 2003 R2 Enterprise x64 Edition, Windows Server 2003 R2 Standard x64 Edition, Windows Server 2008 Enterprise Edition, Windows Server 2008 Standard Edition, Windows Server 2008 Enterprise x64 Edition, Windows Server 2008 Standard x64 Edition, Windows Server 2008 R2 Enterprise Edition, Windows Server 2008 R2 Standard Edition, Windows Vista Business, Windows Vista Enterprise, Windows Vista Ultimate, Windows XP

Professional）
P-2443-7E84 uCosminexus Developer Standard 08-50 （適用 OS：Windows Server 2003 Enterprise Edition, Windows Server 2003 Standard Edition, Windows Server 2003 R2 Enterprise Edition, Windows Server 2003 R2 Standard Edition, Windows 7 Proffesional, Windows 7 Enterprise, Windows 7 Ultimate, Windows Vista Business, Windows Vista Enterprise, Windows Vista Ultimate, Windows XP Professional）
P-2443-7F84 uCosminexus Developer Professional 08-50 （適用 OS：Windows Server 2003 Enterprise Edition, Windows Server 2003 Standard Edition, Windows Server 2003 R2 Enterprise Edition, Windows Server 2003 R2 Standard Edition, Windows 7 Proffesional, Windows 7 Enterprise, Windows 7 Ultimate, Windows Vista Business, Windows Vista Enterprise, Windows Vista Ultimate, Windows XP Professional）
P-2443-7T84 uCosminexus Service Architect 08-50 （適用 OS：Windows Server 2003 Enterprise Edition, Windows Server 2003 Standard Edition, Windows Server 2003 R2 Enterprise Edition, Windows Server 2003 R2 Standard Edition, Windows 7 Proffesional, Windows 7 Enterprise, Windows 7 Ultimate, Windows Vista Business, Windows Vista Enterprise, Windows Vista Ultimate, Windows XP Professional）
P-1M43-7D81 uCosminexus Application Server Standard 08-50 （適用 OS：AIX 5L V5.3, AIX V6.1）※
P-1M43-7K81 uCosminexus Application Server Enterprise 08-50 （適用 OS：AIX 5L V5.3, AIX V6.1）※
P-1M43-7S81 uCosminexus Service Platform 08-50 （適用 OS：AIX 5L V5.3, AIX V6.1）※
P-1J43-7D81 uCosminexus Application Server Standard 08-50 （適用 OS：HP-UX 11i V2（IPF), HP-UX 11i V3（IPF））※
P-1J43-7K81 uCosminexus Application Server Enterprise 08-50 （適用 OS：HP-UX 11i V2（IPF), HP-UX 11i V3（IPF））※
P-1J43-7S81 uCosminexus Service Platform 08-50 （適用 OS：HP-UX 11i V2（IPF), HP-UX 11i V3（IPF））※
P-9S43-7D81 uCosminexus Application Server Standard 08-50 （適用 OS：Red Hat Enterprise Linux AS 4（x86), Red Hat Enterprise Linux 5 Advanced Platform（x86), Red Hat Enterprise Linux ES 4（x86), Red Hat Enterprise Linux 5（x86), Red Hat Enterprise Linux AS 4（AMD64 & Intel EM64T), Red Hat Enterprise Linux 5 Advanced Platform（AMD/Intel 64), Red Hat Enterprise Linux ES 4（AMD64 & Intel EM64T), Red Hat Enterprise Linux 5（AMD/Intel 64））※
P-9S43-7K81 uCosminexus Application Server Enterprise 08-50 （適用 OS：Red Hat Enterprise Linux AS 4（x86), Red Hat Enterprise Linux 5 Advanced Platform（x86), Red Hat Enterprise Linux ES 4（x86), Red Hat Enterprise Linux 5（x86), Red Hat Enterprise Linux AS 4（AMD64 & Intel EM64T), Red Hat Enterprise Linux 5 Advanced Platform（AMD/Intel 64), Red Hat Enterprise Linux ES 4（AMD64 & Intel EM64T), Red Hat Enterprise Linux 5（AMD/Intel 64））※
P-9S43-7S81 uCosminexus Service Platform 08-50 （適用 OS：Red Hat Enterprise Linux AS 4（x86), Red Hat Enterprise Linux 5 Advanced Platform（x86), Red Hat Enterprise Linux ES 4（x86), Red Hat Enterprise Linux 5（x86), Red Hat Enterprise Linux AS 4（AMD64 & Intel EM64T), Red Hat Enterprise Linux 5 Advanced Platform（AMD/Intel 64), Red Hat Enterprise Linux ES 4（AMD64 & Intel EM64T), Red Hat Enterprise Linux 5（AMD/Intel 64））※
P-9V43-7D81 uCosminexus Application Server Standard 08-50 （適用 OS：Red Hat Enterprise Linux AS 4（IPF), Red Hat Enterprise Linux 5 Advanced Platform（Intel Itanium））※

P-9V43-7K81 uCosminexus Application Server Enterprise 08-50 （適用 OS：Red Hat Enterprise Linux AS 4（IPF），Red Hat Enterprise Linux 5 Advanced Platform（Intel Itanium））※
P-9V43-7S81 uCosminexus Service Platform 08-50 （適用 OS：Red Hat Enterprise Linux AS 4（IPF），Red Hat Enterprise Linux 5 Advanced Platform（Intel Itanium））※
P-9D43-7D81 uCosminexus Application Server Standard 08-50 （適用 OS：Solaris 9（SPARC），Solaris 10（SPARC））※
P-9D43-7K81 uCosminexus Application Server Enterprise 08-50 （適用 OS：Solaris 9（SPARC），Solaris 10（SPARC））※
※印の製品については，サポート時期をご確認ください。
これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

■輸出時の注意

本製品を輸出される場合には，外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上，必要な手続きをお取りください。
なお，ご不明な場合は，弊社担当営業にお問い合わせください。

■商標類

■発行

2008 年 9 月 （第 1 版）3020-3-U17
2010 年 2 月 （第 2 版）3020-3-U17-20

■著作権

変更内容

変更内容（3020-3-U17-20）Hitachi Web Server　04-10

追加・変更内容	変更個所
最低限必要なディレクティブの一覧を記載した。	2.3.1(2)，3.3(2)，6.1.3(6)
Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel64)，Red Hat Enterprise Linux 5 (AMD/Intel64) において，SSL 機能の使用方法を追加した。	2.4.2，5.1，5.1.1(1)
一般ユーザアカウントによる運用方法を追加した。	2.4.3，7.2.1(6)
Windows 7 と Windows Server 2008 R2 上で Hitachi Web Server を動作させる場合の注意事項を追加した。	3.1(3)，3.4.1(4)，3.4.2(6)
ログファイルのサイズに関する注意事項を追加した。	4.2.1
ユティリティがシグナルを受信した場合の処理を追加した。	4.2.3(4)，4.2.4(4)
crldownload ユティリティの対象のオペレーティングシステムを記載した。	5.3.1
次のディレクティブを追加した。 HWSNotModifiedResponseHeaders	6.1.1，6.2.4(24)
ディレクティブにパス情報を指定する場合の注意事項を追加した。	6.1.3(3)
PidFile ディレクティブの注意事項を追加した。	6.2.2(11)
サービス起動時のエラーメッセージを追加した。	7.2.1(3)
Hitachi Web Server がシグナルを受信した場合に出力するエラーメッセージに，対象のシグナルを追加した。	7.2.1(6)
コネクションタイムアウトが発生した場合のエラーメッセージを追加した。	7.2.1(7)
Widnows Server 2008 のサーバクラスタの設定方法を追加した。	付録 F

はじめに

このマニュアルは，次に示すプログラムプロダクトのインストール手順と環境設定について説明したものです。

- P-1J41-E181　Hitachi Web Server
- P-1J43-7D81　uCosminexus Application Server Standard
- P-1J43-7K81　uCosminexus Application Server Enterprise
- P-1J43-7S81　uCosminexus Service Platform
- P-1M41-E181　Hitachi Web Server
- P-1M43-7D81　uCosminexus Application Server Standard
- P-1M43-7K81　uCosminexus Application Server Enterprise
- P-1M43-7S81　uCosminexus Service Platform
- P-2441-E184　Hitachi Web Server
- P-2443-7D84　uCosminexus Application Server Standard
- P-2443-7E84　uCosminexus Developer Standard
- P-2443-7F84　uCosminexus Developer Professional
- P-2443-7K84　uCosminexus Application Server Enterprise
- P-2443-7S84　uCosminexus Service Platform
- P-2443-7T84　uCosminexus Service Architect
- P-9D41-E181　Hitachi Web Server
- P-9D43-7D81　uCosminexus Application Server Standard
- P-9D43-7K81　uCosminexus Application Server Enterprise
- P-9S41-E181　Hitachi Web Server
- P-9S43-7D81　uCosminexus Application Server Standard
- P-9S43-7K81　uCosminexus Application Server Enterprise
- P-9S43-7S81　uCosminexus Service Platform
- P-9V41-E181　Hitachi Web Server
- P-9V43-7D81　uCosminexus Application Server Standard
- P-9V43-7K81　uCosminexus Application Server Enterprise
- P-9V43-7S81　uCosminexus Service Platform

■対象読者

WWW 環境で Hitachi Web Server（Web サーバ）を構築，管理するシステム管理者を対象としています。前提となるハードウェアとオペレーティングシステム，ネットワーク，HTTP およびマシンの操作についての知識を持つ方を対象としています。

また，ディレクトリサービスを使用する場合には，次の前提知識を持つ方を対象としています。

- LDAP（Lightweight Directory Access Protocol）

また，Hitachi Directory Server Version 2 と連携する場合には，次の前提知識を持つ方を対象としています。

- Hitachi Directory Server Version 2

■マニュアルの構成

このマニュアルは，次に示す章と付録から構成されています。

第 1 章　Hitachi Web Server とは

Hitachi Web Server の位置づけ，製品の概要について説明しています。

第 2 章　運用の準備と起動，停止（UNIX 版）

Hitachi Web Server（UNIX 版）のプログラムのインストール，環境設定など運用前に理解しておいていただきたいことについて説明しています。

第 3 章　運用の準備と起動，停止（Windows 版）

Hitachi Web Server（Windows 版）のプログラムのインストール，環境設定など運用前に理解しておいていただきたいことについて説明しています。

第 4 章　システムの運用方法

起動・停止方法，Web サーバ環境を運用に合わせて実行するコマンドの使用方法について説明しています。

第 5 章　SSL による認証，暗号化

SSL による認証，暗号化について説明しています。

第 6 章　ディレクティブ

Web サーバ環境を運用に合わせて設定するディレクティブの文法について説明しています。

第 7 章　メッセージ

Hitachi Web Server が出力するメッセージについて説明しています。

付録 A　ステータスコード

Web ブラウザに表示するステータスコードについて説明しています。

付録 B　CGI プログラムに渡す環境変数

Hitachi Web Server が CGI プログラムに渡す環境変数について説明しています。

付録 C　高信頼化システム監視機能 HA モニタによるシステム監視 (クラスタリングシステムの運用)

高信頼化システム監視機能 HA モニタについて説明しています。

付録 D　MC/ServiceGuard によるシステム監視 (クラスタリングシステムの運用)

MC/ServiceGuard を利用した Web サーバの運用例について説明しています。

付録 E　HACMP for AIX によるシステム監視 (クラスタ・マルチプロセッシングの運用)

HACMP for AIX を利用した Web サーバの運用例について説明しています。

付録 F　Microsoft サーバクラスタによるシステム監視

Microsoft サーバクラスタを利用した Web サーバの運用例について説明しています。

付録 G　バージョン 03-00 以降への移行方法

バージョン 03-00 以降へ移行する方法について説明しています。

付録 H　用語解説

マニュアルの中で使用している用語について説明しています。

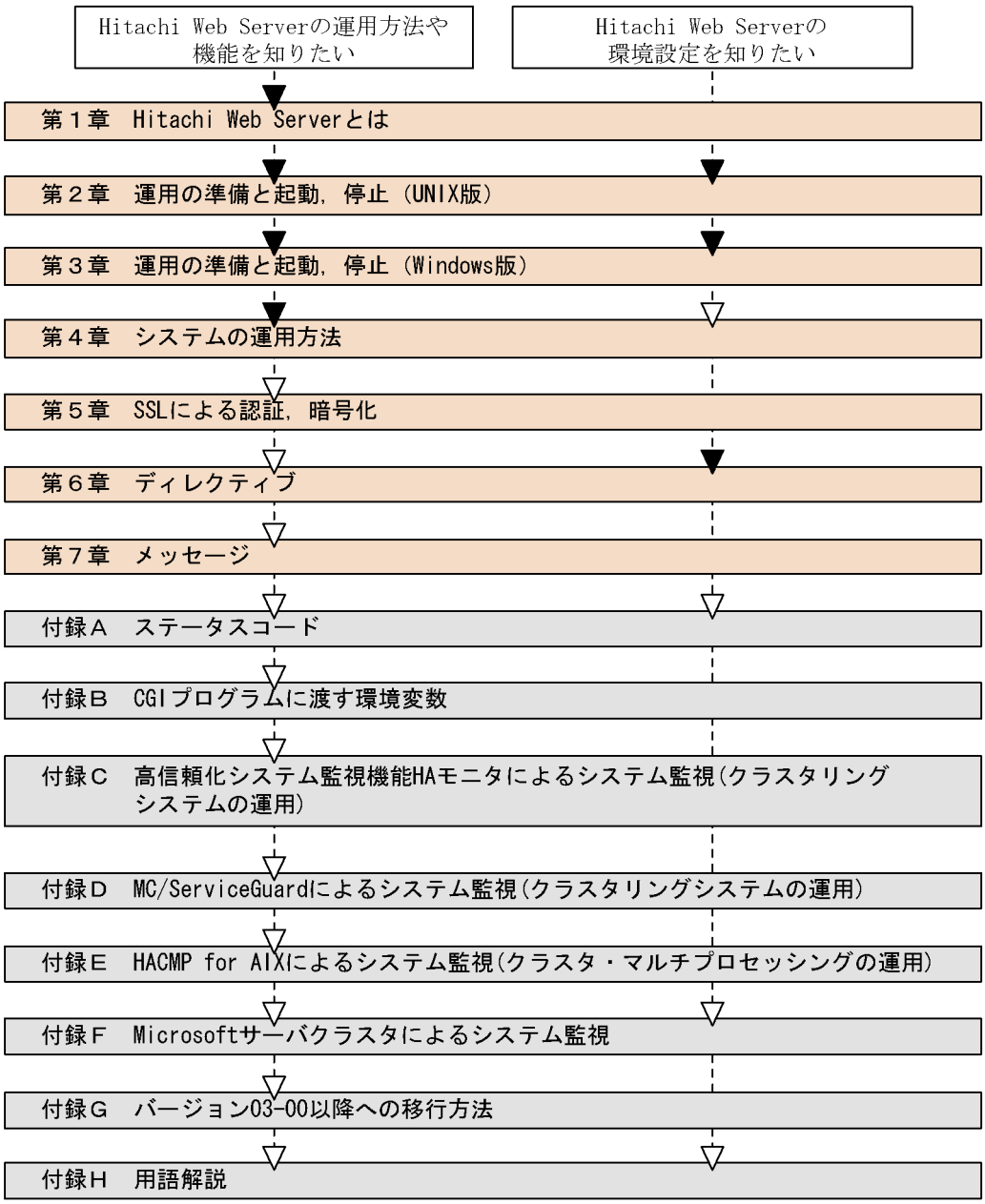
■関連マニュアル

- 高信頼化システム監視機能　HA モニタ　AIX(R) 編（3000-9-130）※
- 高信頼化システム監視機能　HA モニタ　HP-UX 編（3000-9-131）※
- 高信頼化システム監視機能　HA モニタ　Linux(R) 編（3000-9-132）※
- 高信頼化システム監視機能　HA モニタ　メッセージ（3000-9-134）※
- 日立ディレクトリサービス　導入編（3020-3-825）

注※

本文中で使用している HA モニタのマニュアル名は，AIX(R) 編，HP-UX(R) 編，Linux(R) 編およびメッセージを省略して表記しています。使用しているプラットフォームに応じて AIX 用，HP-UX 用または Linux 用のマニュアルを参照してください。また，必要に応じてメッセージのマニュアルを参照してください。

■読書手順



■このマニュアルでの表記

このマニュアルで使用している表記と，対応する製品名を次に示します。

表記			製品名
Internet Explorer			Internet Explorer(R)
IPF			Itanium (R) Processor Family
uCosminexus Application Server			uCosminexus Application Server Enterprise
			uCosminexus Application Server Standard
UNIX	AIX		AIX 5L V5.3 AIX V6.1
	HP-UX	HP-UX（IPF）	HP-UX 11i V2（IPF） HP-UX 11i V3（IPF）
	Linux	Linux（32 ビット）	Red Hat Enterprise Linux AS 4（x86） Red Hat Enterprise Linux 5 Advanced Platform（x86） Red Hat Enterprise Linux ES 4（x86） Red Hat Enterprise Linux 5（x86） Red Hat Enterprise Linux AS 4（AMD64 & Intel EM64T） Red Hat Enterprise Linux 5 Advanced Platform（AMD/Intel 64） Red Hat Enterprise Linux ES 4（AMD64 & Intel EM64T） Red Hat Enterprise Linux 5（AMD/Intel 64）
		Linux（IPF）	Red Hat Enterprise Linux AS 4（IPF） Red Hat Enterprise Linux 5 Advanced Platform（Intel Itanium）
	Solaris		Solaris 9 Solaris 10
Windows	Windows Server 2003	Windows Server 2003 Enterprise Edition	Microsoft(R) Windows Server(R) 2003, Enterprise Edition 日本語版
		Windows Server 2003 Standard Edition	Microsoft(R) Windows Server(R) 2003, Standard Edition 日本語版
		Windows Server 2003 R2 Enterprise Edition	Microsoft(R) Windows Server(R) 2003 R2，Enterprise Edition 日本語版
		Windows Server 2003 R2 Standard Edition	Microsoft(R) Windows Server(R) 2003 R2，Standard Edition 日本語版
		Windows Server 2003 Enterprise x64 Edition	Microsoft(R) Windows Server(R) 2003, Enterprise x64 Edition 日本語版
		Windows Server 2003 Standard x64 Edition	Microsoft(R) Windows Server(R) 2003, Standard x64 Edition 日本語版
		Windows Server 2003 R2 Enterprise x64 Edition	Microsoft(R) Windows Server(R) 2003 R2，Enterprise x64 Edition 日本語版

表記			製品名
		Windows Server 2003 R2 Standard x64 Edition	Microsoft(R) Windows Server(R) 2003 R2，Standard x64 Edition 日本語版
	Windows Server 2008	Windows Server 2008 Enterprise Edition	Microsoft(R) Windows Server(R) 2008 Enterprise 32-bit 日本語版
		Windows Server 2008 Standard Edition	Microsoft(R) Windows Server(R) 2008 Standard 32-bit 日本語版
		Windows Server 2008 Enterprise x64 Edition	Microsoft(R) Windows Server(R) 2008 Enterprise 日本語版
		Windows Server 2008 Standard x64 Edition	Microsoft(R) Windows Server(R) 2008 Standard 日本語版
	Windows Server 2008 R2	Windows Server 2008 R2 Enterprise Edition	Microsoft(R) Windows Server(R) 2008 R2 Enterprise 日本語版
		Windows Server 2008 R2 Standard Edition	Microsoft(R) Windows Server(R) 2008 R2 Standard 日本語版
	Windows 7	Windows 7 Professional	Microsoft(R) Windows(R) 7 Professional 日本語版
		Windows 7 Enterprise	Microsoft(R) Windows(R) 7 Enterprise 日本語版
		Windows 7 Ultimate	Microsoft(R) Windows(R) 7 Ultimate 日本語版
	Windows Vista	Windows Vista Business	Microsoft(R) Windows Vista(R) Business 日本語版
		Windows Vista Enterprise	Microsoft(R) Windows Vista(R) Enterprise 日本語版
		Windows Vista Ultimate	Microsoft(R) Windows Vista(R) Ultimate 日本語版
	Windows XP Professional		Microsoft(R) Windows(R) XP Professional Operating System

■このマニュアルで使用している記号

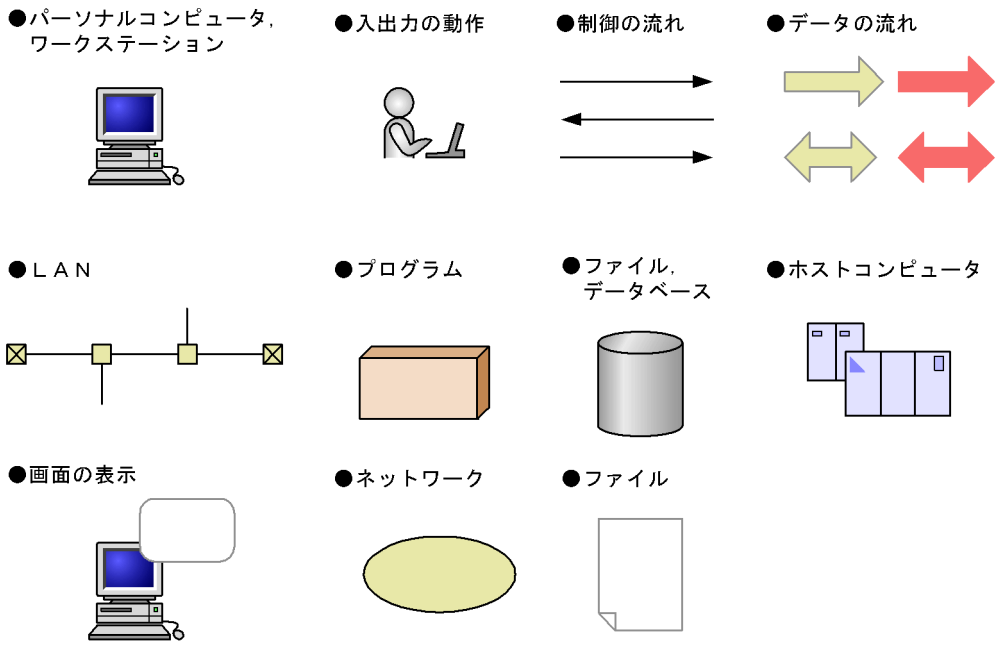
文法説明をするときに使用する記号について説明します。

記号	意味
〔 〕	〔 〕内の項目を省略できます。 **(例)** A〔,B〕〔,C〕のとき，次の 4 通りの指定ができます。 A A,B A,B,C A,C
{ }	{ } 内のどれか一つを選んで指定します。 **(例)** A {,B | ,C} のとき，次の 2 通りの指定ができます。 A,B A,C

記号	意味
|	選択肢の区切りを表しています。
__（アンダーライン）	項目の指定を省略したときに，システムが仮定する値を表しています。
…	この記号の直前に位置する項目を繰り返し指定できます。
～	この記号の直前に位置する項目をこの記号以降の文法規則に従って，記述することを示します。
《 》	項目の指定を省略したときに，システムが仮定する値を表しています。
(( ))	指定できる値の範囲を表しています。
U	UNIX 版だけに有効なディレクティブを表しています。
W	Windows 版だけに有効なディレクティブを表しています。

■図中で使用する記号

このマニュアルの図中で使用する記号を，次のように定義します。



■ Windows の場合のフォルダとパスの表記

このマニュアルでは，Windows，HP-UX，AIX，Linux，および Solaris で共通の内容の場合，Windows の「フォルダ」を「ディレクトリ」と表記しています。Windows の場合，「ディレクトリ」を「フォルダ」に置き換えてお読みください。

■常用漢字以外の漢字の使用について

このマニュアルでは，常用漢字を使用することを基本としていますが，次に示す用語については，常用漢字以外の漢字を使用しています。

- 鍵（かぎ）
- 個所（かしょ）
- 同梱（どうこん）
- 汎用（はんよう）
- 必須（ひっす）
- 漏洩（ろうえい）

■ KB（キロバイト）などの単位表記について

1KB（キロバイト），1MB（メガバイト），1GB（ギガバイト），1TB（テラバイト）はそれぞれ1,024 バイト，$1{,}024^2$ バイト，$1{,}024^3$ バイト，$1{,}024^4$ バイトです。

■ Hitachi Web Server のホームページ

インターネットのホームページを開設しています。
http://www.hitachi.co.jp/Prod/comp/soft1/webserver/index.html

目次

1 Hitachi Web Server とは

この章では，Hitachi Web Server の概要について説明します。

1.1 Hitachi Web Server の概要

近年の Web 関連技術はインターネット，イントラネットおよびエクストラネットへと拡大，発展しています。それに伴い，Web サーバは，トランザクション処理を含む業務システムや，基幹系システムなどの，ミッションクリティカルな環境で利用されるようになってきました。そのような環境で利用できる基幹業務システム向け Web サーバである Hitachi Web Server は，きめ細かな保守サービス，テクニカルサービスによって，信頼性の高いシステムをサポートしています。

1.2 Hitachi Web Server の特長

Hitachi Web Server は，全世界で高いシェアを持つ Apache HTTP Server をベースに開発しています。Hitachi Web Server でのサポート範囲は，このマニュアルの記述範囲です。

Hitachi Web Server で使用できる主な機能には次のものがあります。

- ユーザ認証とアクセス保護
- バーチャルホスト
- リバースプロキシ
- 流量制限機能
- 有効期限設定機能
- ヘッダカスタマイズ機能
- CGI プログラムの実行
- 静的コンテンツキャッシュ機能
- ディレクトリインデクス表示
- イメージマップ

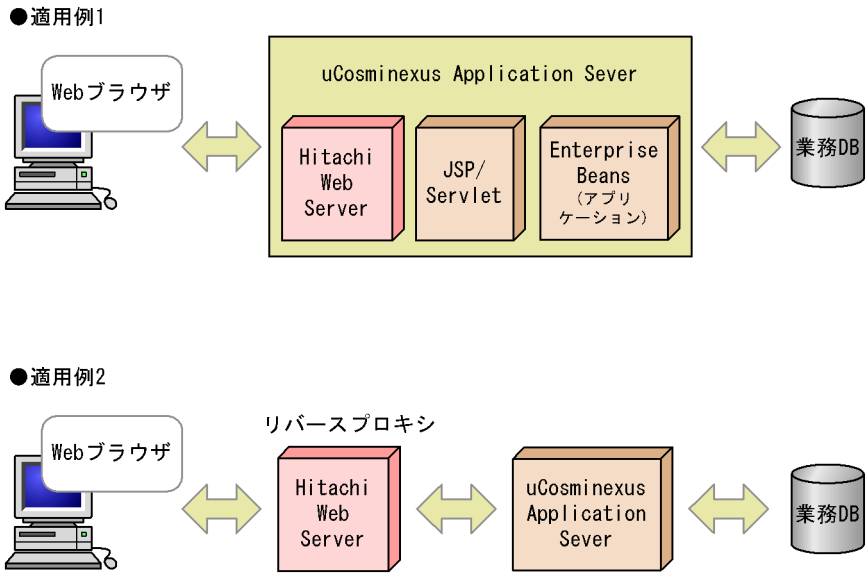
また，Hitachi Web Server は，RSA Security 社の製品である BSAFE(R) SSL-C を導入し，SSL（Secure Sockets Layer）を実装しています。これによって，データの改ざん，なりすまし（クライアントから見たサーバのなりすましおよびサーバから見たクライアントのなりすまし）および盗聴を防止し，情報の安全性を確保できます。

uCosminexus Application Server での適用例

Hitachi Web Server は，uCosminexus Application Server を構成する製品の一つです。

uCosminexus Application Server での適用例を次に示します。

図 1-1 uCosminexus Application Server での適用例



2 運用の準備と起動，停止（UNIX 版）

この章では，Hitachi Web Server を運用する前に，知っておいていただきたいことおよび起動と停止について説明します。

2.1 Hitachi Web Server を運用するためのシステム構成

Hitachi Web Server を運用するために必要なシステム構成について説明します。

(1) ハードウェア構成

(a) サーバ

Hitachi Web Server の適用機種，使用するメモリ所要量，およびディスク占有量については，リリースノートを参照してください。

(b) クライアント

Web ブラウザが動作できる端末

(c) ネットワーク関連

- イーサネットなどのネットワーク（必須）
- ドメインネームシステムサーバ（任意）
- ロードバランサ（任意）
- SSL アクセラレータ（任意）
- ファイアウォール（任意）

(2) ソフトウェア構成

(a) サーバ

ディレクトリサービス（LDAP サーバ）を利用してユーザ認証およびアクセス保護をする場合に，必要なソフトウェア。

● ユーザを一元管理するためのディレクトリサービスが利用できるソフトウェア
P-1B44-A121 Hitachi Directory Server Version 2　02-01 以降

● 日立ディレクトリサービスにアクセスするためのソフトウェア
- AIX 版
  OS に同梱されている ldap.client.rte ファイルセット
- Linux（32 ビット），Solaris 版
  同等の機能が製品に同梱されています。

日立ディレクトリサービスの導入方法については，マニュアル「日立ディレクトリサービス　導入編」を参照してください。

なお，同じ LDAP 対応のディレクトリサービスとして「Sun Java(TM) System Directory Server」も使用できます。

2.2 インストールとアンインストール

Hitachi Web Server のインストール・アンインストールは，製品として提供されている CD-ROM に格納されている日立 PP インストーラ（Hitachi PP Installer）を使用します。

日立 PP インストーラは，スーパユーザ以外では実行できません。このため，インストール対象のマシンにスーパユーザ（root）でログインしてください。

Hitachi Web Server をインストールする前に，Hitachi Web Server のすべてのプログラムおよび関連ファイルへのすべての操作を終了してください。また，日立 PP インストーラを使用すると，/opt および /etc のディレクトリ属性が変更される場合があります。

日立 PP インストーラを使ったインストールが失敗した場合は，Hitachi Web Server のすべてのプログラムおよび関連ファイルへのすべての操作が終了していることを確認してから，再度インストールしてください。

2.2.1 インストール

（1）日立 PP インストーラの起動

(a) CD-ROM からの起動

ここでは CD-ROM からの起動方法を説明します。

● **CD-ROM ファイルシステムのマウント**

最初に，CD-ROM ファイルシステムをマウントしておく必要があります。マウントするためには，次のコマンドを入力します。

下線部のデバイススペシャルファイル名および CD-ROM ファイルシステムのマウントディレクトリ名は，使用環境に合わせて変更してください。

```
mount /dev/dsk/c0t2d0 /cdrom
```

● **CD-ROM セットアッププログラムの起動**

CD-ROM セットアッププログラムを起動して，日立 PP インストーラをハードディスク上にインストールします。

次に CD-ROM セットアッププログラムのコマンドを示します。このコマンドを実行すると，同時に日立 PP インストーラも起動します。

下線部には使用する CD-ROM ディレクトリ名を指定してください。なお，CD-ROM 内のディレクトリやファイル名は，マシンによっては，小文字に見えます。ls コマンドで確認の上，小文字であれば，小文字で入力してください。

```
/cdrom/OS名称/SETUP /cdrom
```

OS 名称：HP-UX の場合は HPUX，Solaris の場合は SOLARIS，Linux の場合は LINUX，AIX の場合は AIX と指定してください。

（2） PP のインストール

（a） インストーラメインメニュー

/etc/hitachi_setup コマンドを実行すると，次に示すインストーラメインメニューが表示されます。

```
Hitachi PP Installer  04-06

    L) List Installed Software.
    I) Install Software.
    D) Delete  Software.
    Q) Quit.

    Select Procedure ===>

    +---------------------------------------------------------------------+
      CAUTION!
      YOU SHALL INSTALL AND USE THE SOFTWARE PRODUCT LISTED IN THE
      "List Installed Software." UNDER THE TERMS AND CONDITION OF
      THE SOFTWARE LICENSE AGREEMENT ATTACHED TO SUCH SOFTWARE PRODUCT.
    +---------------------------------------------------------------------+


  All Rights Reserved. Copyright (C) 1994, 2006, Hitachi, Ltd.
```

（b） PP インストール画面

インストーラメインメニューで［i］または［I］を選択すると，次に示す PP インストール画面が表示されます。

```
         PP-No.          VR      PP-NAME
<@>001 P-XX41-XXXX     0310    Hitachi Web Server
F)Forward B) Backward J) Down K) Up Space) Select/Unselect I) Install Q) Quit
```

注　P-XX41-XXXX にはインストールする製品の形名が表示されます。

インストールしたい PP にカーソルを移動させ，スペースキーで選択します。選択した PP の左側には，"＜＠＞" が表示されます。

続いて，［i］または［I］を入力すると，最下行に次のメッセージが表示されます。

```
Install PP? (y: install,  n: cancel) ==>
```

ここで，［y］または［Y］を選択すると，インストールが開始されます。［n］または［N］を選択すると，インストールが中止され，PP インストール画面に戻ります。

［q］または［Q］を入力すると，インストーラメインメニューに戻ります。

2.2.2 アンインストール

アンインストール方法について説明します。

アンインストールは Hitachi Web Server を必ず停止してから，実行してください。

（1） 日立 PP インストーラの起動

/etc/hitachi_setup コマンドを実行します。日立 PP インストーラが起動されます。

```
/etc/hitachi_setup
```

（2） アンインストールの選択

（a） インストーラメインメニュー

/etc/hitachi_setup コマンドを実行すると，次に示すインストーラメインメニューが表示されます。

```
Hitachi PP Installer  04-06


  L) List Installed Software.
  I) Install Software.
  D) Delete  Software.
  Q) Quit.


  Select Procedure ===>


  +---------------------------------------------------------------------+
    CAUTION!
    YOU SHALL INSTALL AND USE THE SOFTWARE PRODUCT LISTED IN THE
    "List Installed Software." UNDER THE TERMS AND CONDITION OF
    THE SOFTWARE LICENSE AGREEMENT ATTACHED TO SUCH SOFTWARE PRODUCT.
  +---------------------------------------------------------------------+


All Rights Reserved. Copyright (C) 1994, 2006, Hitachi, Ltd.
```

（b） PP 削除画面

インストーラメインメニューで［d］または［D］を選択すると，次に示す PP 削除画面が表示されます。アンインストールできるプログラムの一覧が表示されます。

```
        PP-No.           VR      PP-NAME
<@>001 P-XX41-XXXX     0310    Hitachi Web Server
        :
        :
        :
F) Forward B) Backward J) Down K) Up Space) Select/Unselect D) Delete Q) Quit
```

注　P-XX41-XXXX にはアンインストールする製品の形名が表示されます。

ここでアンインストールしたい PP にカーソルを移動させ（この場合は Hitachi Web Server），スペースキーを押します。選択した PP の左側には，"＜@＞" が表示されます。このとき，ほかの PP を間違って選択しないように注意してください。

（3） アンインストールの開始

アンインストールしたい PP 名に "＜@＞" が表示されているのを確認して［d］または［D］を入力してください。最下行に次のメッセージが表示されます。

```
Delete PP? (y: delete,  n: cancel) ==>
```

ここで，［y］または［Y］を選択すると，アンインストールが開始されます。［n］また

は［N］を選択すると，アンインストールが中止され，PP 削除画面に戻ります。

2.2.3 PP 一覧の表示

インストーラメインメニューで［l］または［L］を選択すると，次に示す PP 一覧表示画面が表示されます。これは，このマシンにインストールされている PP の一覧です。

［p］または［P］を選択すると，インストール済みの PP 一覧が "/tmp/hitachi_PPLIST" に出力されます。

［q］または［Q］を選択するとインストーラメインメニューに戻ります。

```
    PP-No.          VR      Install date        PP-NAME
 001 P-XX41-XXXX     0310    2007/04/01 12:00   Hitachi Web Server
       :
       :
       :
F) Forward B) Backward Q) Quit P) Print to /tmp/hitachi_PPLIST ==>
```

注 P-XX41-XXXX にはインストール済みの製品の形名が表示されます。

2.3 運用環境を定義する

Hitachi Web Server の動作を定義するファイルについて説明します。

2.3.1 環境の定義方法

(1) ディレクトリ構成

Hitachi Web Server をインストールしたときの，ディレクトリ構成を次に示します。この構成は変更しないでください。

図 2-1 ディレクトリ構成

```
/
 └opt
   └hitachi
     └httpsd              ルートディレクトリ
       ├admin
       │ └bin             複数サーバ環境生成ユティリティ格納ディレクトリ
       │                  (httpsd.conf.org, hwsserveredit, hwsconfigedit)
       ├bin               実行ファイル格納ディレクトリ(htpasswd, cosmippenv, cosmippenvc)
       ├build             モジュール作成用ファイル格納ディレクトリ
       ├cgi-bin           CGIプログラム格納ディレクトリ
       ├conf              設定ファイル格納ディレクトリ(httpsd.conf, mime.types)
       │ └ssl             SSL用ディレクトリ
       │   ├cacert        CA証明書格納ディレクトリ
       │   ├cacerts       CA証明書のハッシュリンク用ディレクトリ
       │   ├crl
       │   │ ├DER         CRL格納ディレクトリ(DER形式)
       │   │ └PEM         CRL格納ディレクトリ(PEM形式)
       │   └server        サーバ秘密鍵，サーバ証明書用ディレクトリ
       ├htdocs            デフォルトドキュメントルートディレクトリ(index.html)
       ├icons             アイコン画像格納ディレクトリ
       ├include           ヘッダファイル格納ディレクトリ
       ├libexec           共有ライブラリ格納ディレクトリ
       ├logs              ログ，プロセスIDファイル格納ディレクトリ
       ├maintenance       保守情報収集機能用ディレクトリ
       ├sbin              管理者用ユティリティ(httpsdctl, logresolve, rotatelogs,
       │                  rotatelogs2, crldownload, sslpasswd, hwstraceinfoなど)
       ├servers           複数サーバ環境生成ディレクトリ
       └sslc              BSAFE SSL-Cディレクトリ
         └bin             SSL関連ユティリティ(sslccert, sslckey)
           └demoCA        sslccertユティリティ設定ファイル格納ディレクトリ(sslc.cnf)
```

(2) コンフィグファイル

Hitachi Web Server の動作環境を定義するファイルをコンフィグファイルといいます。なお，コンフィグファイルのコメント行以外に，マルチバイト文字および Unicode の補助文字は指定できません。

Hitachi Web Server の動作を定義するファイルには，httpsd.conf ファイル，mime.types ファイルおよびアクセスコントロールファイル（.htaccess）の三つがありま

す。SSL ユティリティの動作を定義するファイルは，sslc.cnf ファイルです。各ファイルの用途を次に示します。

表 2-1　コンフィグファイルの用途

ファイル名	用途	標準提供
httpsd.conf	Hitachi Web Server の動作環境を各種ディレクティブで定義します。システム管理者が管理します。	○
mime.types	コンテンツのファイル拡張子とコンテントタイプ（MIME タイプ）の関連づけを定義します。システム管理者が管理します。	○
.htaccess	アクセス制御を定義するアクセスコントロールファイル。必要に応じてエンドユーザがアクセス制御するディレクトリ下に作成します（デフォルトファイル名は .htaccess）。	×
sslc.cnf	SSL のユティリティについての情報を定義します。システム管理者が管理します。sslc ユティリティだけで使用できます。 認証局（CA）に提出する CSR を作成する場合，ユティリティで使用する値を指定しておくと，効率良く CSR を作成できます。	○
Include ディレクティブで指定したファイル	Hitachi Web Server の動作環境を各種ディレクティブで定義します。システム管理者が管理します。	×

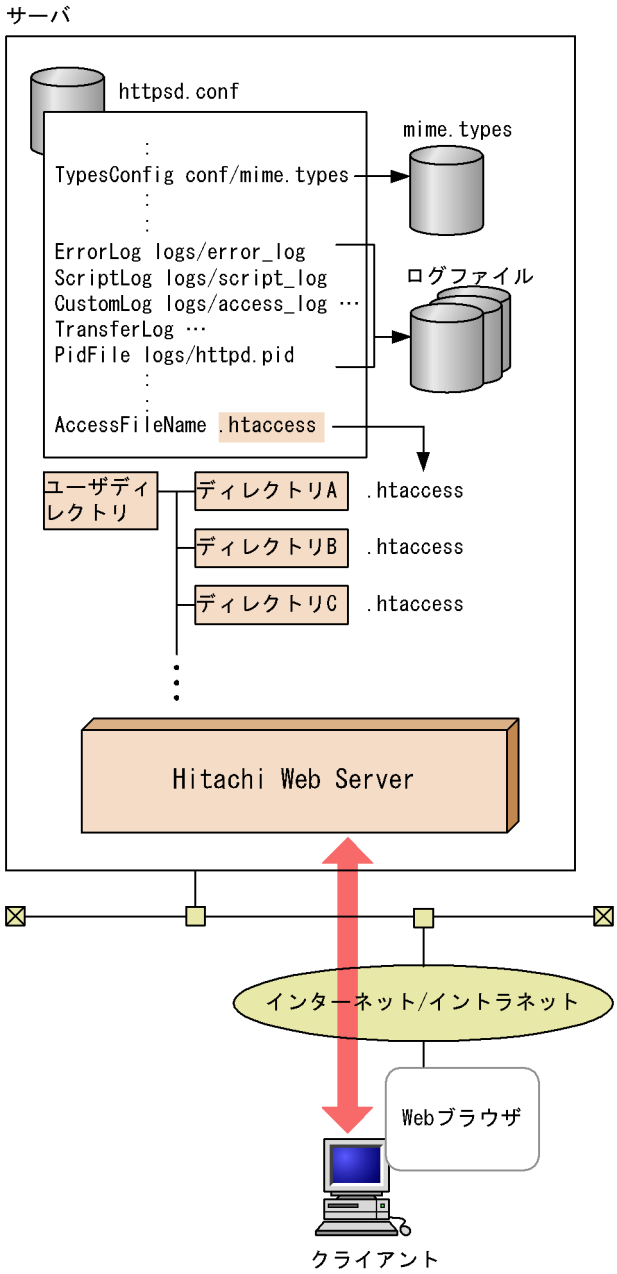
（凡例）

○：標準提供する。

×：標準提供しない。

コンフィグファイルの関連を次に示します。

図 2-2　コンフィグファイルの関連



2.3.2　システムパラメタの定義方法

Hitachi Web Server に必要なシステムパラメタの定義方法について説明します。設定方法の詳細については，ご使用の OS のマニュアルを参照してください。

（1） 共有メモリセグメントの最大サイズ

Hitachi Web Server では，サーバプロセスのステータス情報と内部のトレース情報を共有メモリに採取します。OS には，共有メモリセグメント一つ当たりの最大サイズについて制限するシステムパラメタがあります。

サーバプロセスのステータス情報で使用する共用メモリの大きさ（バイト）は 400KB，トレース情報は「MaxClients ディレクティブの値× 7KB」です。共有メモリセグメントの最大サイズのシステムパラメタには，それら以上の値を設定してください。

システムパラメタの例を次に示します。なお，OS のシステムパラメタの種別およびその内容は，使用している OS またはカーネルのバージョンごとに異なります。システムパラメタの内容および設定方法の詳細については，ご使用の OS のマニュアルを参照してください。

Linux（32 ビット），Linux（IPF）の場合（括弧内は設定ファイル例）

共有メモリセグメントの最大サイズ：kernel.shmmax（/etc/sysctl.conf）

Solaris の場合

共有メモリセグメントの最大サイズ（Solaris 8, 9）：shmsys:shminfo_shmmax
プロジェクト当たりの共有メモリセグメントの最大サイズ（Solaris 10）：project.max-shm-memory

AIX，HP-UX（IPF）の場合

共有メモリセグメントの最大サイズ：shmmax

（2） 最大プロセス数

システム上のプロセス数およびユーザ当たりのプロセス数は，OS のシステムパラメタによって制限されています。次に示す起動プロセス数から，運用環境における起動プロセス数の合計値を考慮し，システムパラメタを設定してください。

制御プロセス

起動プロセス数：1 個
変更対象システムパラメタ：Web サーバを起動したユーザにおける最大プロセス数

サーバプロセス

起動プロセス数：MaxClients ディレクティブ指定値
変更対象システムパラメタ：User ディレクティブに指定したユーザにおける最大プロセス数

CGI プロセス

CGI プログラムの実行が許可されている場合，CGI プロセスはリクエスト処理時に各サーバプロセスから起動されます。
起動プロセス数：MaxClients ディレクティブ指定値
変更対象システムパラメタ：User ディレクティブに指定したユーザにおける最大プ

ロセス数

gcache サーバ

SSL セション管理機能を有効にしている場合起動します。
起動プロセス数：1 個
変更対象システムパラメタ：User ディレクティブに指定したユーザにおける最大プロセス数

Hitachi Web Server のプロセス構造については，「4.1.1　Hitachi Web Server のプロセス構造（UNIX 版）」を参照してください。

システムパラメタの例を次に示します。なお，OS のシステムパラメタの種別およびその内容は，使用している OS またはカーネルのバージョンごとに異なります。システムパラメタの内容および設定方法の詳細については，ご使用の OS のマニュアルを参照してください。

Linux（32 ビット），Linux（IPF）の場合（括弧内は設定ファイル例）

システム全体の最大プロセス数：kernel.threads-max（/etc/sysctl.conf）
ユーザ当たりの最大プロセス数：nproc（/etc/security/limits.conf）

Solaris の場合

システム全体の最大プロセス数：max_nprocs
ユーザ当たりの最大プロセス数：maxuprc

AIX の場合

ユーザ当たりの最大プロセス数：maxuproc

HP-UX（IPF）の場合

システム全体の最大プロセス数：nproc
ユーザ当たりの最大プロセス数：maxuprc

（3） 最大ファイル数 (ファイルディスクリプタ数)

システム上でオープンできるファイル数およびユーザ当たりオープンできるファイル数は，OS のシステムパラメタによって制限されています。次に示す，Hitachi Web Server で使用するファイルディスクリプタ数を考慮して，これらのシステムパラメタを設定してください。

Linux（32 ビット），Linux（IPF）の場合

ファイルディスクリプタ数 ＝
（50 ＋ A × B ＋ C ＋ 11 × C × D ＋ 8 × E ＋ (F ＋ I) × G) × 1.2

Solaris，AIX，HP-UX（IPF）の場合

ファイルディスクリプタ数 ＝
（50 ＋ A × B ＋ C ＋ 3 × C × D ＋ 5 × E ＋ (F ＋ I) × G) × 1.2

A：Listen ディレクティブ指定数
B：ホストに割り当てられた IP アドレスの数
C：CustomLog，ErrorLog，HWSRequestLog，TransferLog ディレクティブ指定の総数
D：パイプを指定する場合は 1，指定しない場合は 0
E：同時実行 CGI 数（MaxClients 指定値）
F：SSL を使用する場合は 3，使用しない場合は 2
G：同時実行リクエスト数（MaxClients 指定値）
I：リバースプロキシを使用する場合は 1，使用しない場合は 0
なお，CGI プログラム内および Hitachi Web Server に同梱されていない外部モジュール内で使用するファイルディスクリプタの数は含みません。

システムパラメタの例を次に示します。なお，OS のシステムパラメタの種別およびその内容は，使用している OS またはカーネルのバージョンごとに異なります。システムパラメタの内容および設定方法の詳細については，ご使用の OS のマニュアルを参照してください。

Linux（32 ビット），Linux（IPF）の場合（括弧内は設定ファイル例）

- システム全体の最大ファイル数：fs.file-max（/etc/sysctl.conf）

Solaris の場合

Solaris では，一つのプロセスがオープンできるファイル記述子（ファイルディスクリプタ）数を設定します。

- 一つのプロセスがオープンできるファイル記述子数の「弱い」限度：rlim_fd_cur
  rlim_fd_cur には，50 以上を設定してください。
- 一つのプロセスがオープンできるファイル記述子の「強い」限度：rlim_fd_max
  rlim_fd_max には，rlim_fd_cur 指定値以上を設定してください。

AIX の場合

AIX では，一つのプロセスがオープンできるファイルディスクリプタ数を設定します。

- ユーザ・プロセスが一度にオープンさせることができるファイル・ディスクリプタの数のソフト制限：nofiles
  nofiles には，50 以上を設定してください。
- ユーザ・プロセスが一度にオープンさせることができるファイル・ディスクリプタの数のハード制限：nofiles_hard
  nofiles_hard には，nofiles 指定値以上を設定してください。

AIX，HP-UX（IPF）の場合

- システム全体の最大ファイル数：nfiles
- プロセスごとのファイル記述子の論理的最大数の初期値：maxfiles
  maxfiles には，50 以上を設定してください。
- プロセスごとのファイル記述子の物理的最大数：maxfiles_lim

maxfiles_lim には，maxfiles 指定値以上を設定してください。

2.4 起動と停止

Hitachi Web Server の起動および停止方法について説明します。Hitachi Web Server は httpsdctl ユティリティでテスト，起動，停止および再起動します。デフォルト以外の httpsd.conf ファイル名を指定する場合またはサーバのルートディレクトリや httpsd.conf ファイルを起動時に指定する場合は，httpsd で起動します。

2.4.1 Hitachi Web Server を起動，停止する（httpsdctl ユティリティ）

Hitachi Web Server の起動および停止をする httpsdctl ユティリティについて説明します。

（1）形式

```
/opt/hitachi/httpsd/sbin/httpsdctl {start | stop | restart | graceful |
gracefulstop | configtest | help}
```

（2）オプション

● start

Hitachi Web Server を起動します。暗号化した秘密鍵を使って，SSL を利用する場合は，起動時に，秘密鍵のパスワードの入力要求があります。

● stop

Hitachi Web Server を停止します。

● restart

Hitachi Web Server を再起動します。実行中のサーバプロセスは，直ちに停止します。すべてのサーバプロセス終了後に再起動します。再起動時には，MaxClients ディレクティブ指定値の変更は反映されないで，前回の値が引き継がれます。Listen ディレクティブ指定値および SSL 通信で使用する秘密鍵の設定（SSLCertificateKeyFile ディレクティブ）を変更した場合は，いったん Hitachi Web Server を停止してから，起動し直してください。

● graceful

Hitachi Web Server を再起動します。実行中のサーバプロセスは，実行終了後に停止します。サーバプロセスは，随時，新しいコンフィグファイルに基づいて起動します。再起動時には，MaxClients ディレクティブ指定値の変更は反映されないで，前回の値が引き継がれます。Listen ディレクティブ指定値および SSL 通信で使用する秘密鍵の設定（SSLCertificateKeyFile ディレクティブ）を変更した場合は，いったん Hitachi Web Server を停止してから，起動し直してください。

● gracefulstop

Hitachi Web Server を停止します。実行中のサーバプロセスは，実行終了後に停止します。実行が終了しない場合は，HWSGracefulStopTimeout ディレクティブに指定した待ち時間が経過すると終了します。

● configtest

コンフィグファイルの文法チェックをします。文法エラーがあると，画面にエラーメッセージを表示します。このオプションを指定した場合は，Hitachi Web Server は起動しません。

● help

httpsdctl のヘルプを表示させます。

（3） 起動確認方法

Hitachi Web Server の起動を確認するには，制御プロセスを確認してください。詳細は，「4.1.3　稼働管理について」の「(3)　制御プロセスの監視」を参照してください。

（4） 使用例

Hitachi Web Server を起動します。暗号化した秘密鍵を使用している場合はパスワードを入力します。

```
/opt/hitachi/httpsd/sbin/httpsdctl start
Enter PEM pass phrase:
```

（5） 注意事項

- httpsdctl stop および gracefulstop による Web サーバ停止操作実行時に，Hitachi Web Server のコンフィグファイルの定義が不正な場合，httpsdctl の実行はエラーとなり Web サーバは停止しません。
- httpsdctl restart および graceful による Web サーバ再起動実行時に，Hitachi Web Server のコンフィグファイルの定義が不正な場合，httpsdctl の実行はエラーとなり Web サーバは停止しないで再起動しません。
- httpsdctl ユティリティによる Hitachi Web Server の起動，再起動および停止操作を実行した場合，起動完了および停止完了を示すメッセージは出力されません。

2.4.2　Hitachi Web Server を起動する（httpsd）

Hitachi Web Server は httpsd でも起動できます。普通は，この方法では起動しません。デフォルト以外の httpsd.conf ファイル名称を指定する場合またはサーバのルートディレクトリや httpsd.conf ファイルを指定して起動する場合に，この方法を使用します。

（1） 形式

```
/opt/hitachi/httpsd/sbin/httpsd〔〔-d ディレクトリ〕〔-f ファイル名〕〔-R ディレクトリ〕｜-v｜-t〕〔-D HWS_OPTION_HWS2〕
```

（2） オプション

● -d **ディレクトリ**

ServerRoot ディレクティブがコンフィグファイルに指定されていない場合の，デフォルト値を指定できます。

● -f **ファイル名**

httpsd.conf ファイルを指定できます。絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスで指定します。

● -R **ディレクトリ**

DSO 実行ライブラリが格納されているディレクトリを絶対パスで指定します。

● -v

バージョン情報を表示させます。このオプションを指定した場合は，Hitachi Web Server は起動しません。

● -t

コンフィグファイルの文法チェックをします。文法エラーがあると，画面にエラーメッセージを表示します。このオプションを指定した場合は，Hitachi Web Server は起動しません。

● -D HWS_OPTION_HWS2

Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel64), Red Hat Enterprise Linux 5 (AMD/Intel64) において SSL 機能を使用にする場合に指定します。他のプラットフォームや，SSL 機能を使用しない場合は，指定しないでください。

（3） 再起動方法

kill コマンドで Hitachi Web Server を再起動できます。

```
kill {-HUP｜-USR1} `cat PidFileディレクティブ指定値`
```

● -HUP

httpsdctl ユティリティの restart に相当する再起動をします。

● -USR1

httpsdctl ユティリティの graceful に相当する再起動をします。

● PidFile ディレクティブ指定値

PidFile ディレクティブで指定した値（ファイル名）を指定します。

（4） 終了方法

httpsd で Hitachi Web Server を起動した場合，次に示すコマンドを実行してプロセスを終了し，Hitachi Web Server を停止してください。

```
kill {-TERM|-USR2} `cat PidFileディレクティブ指定値`
```

● -TERM

httpsdctl ユティリティの stop に相当する停止をします。

● -USR2

httpsdctl ユティリティの gracefulstop に相当する停止をします。

（5） 起動確認方法

Hitachi Web Server の起動を確認するには，制御プロセスを確認してください。詳細は，「4.1.3　稼働管理について」の「(3)　制御プロセスの監視」を参照してください。

2.4.3　一般ユーザアカウントによる運用

Hitachi Web Server は，通常の運用方法として，スーパーユーザによる運用を想定しています。

インストールした状態では，スーパーユーザによる運用ができるように各種設定が施されています。

このことから，スーパーユーザ以外のユーザ（以下，一般ユーザと呼びます）で運用する場合，Hitachi Web Server の設定ファイルや関連するディレクトリ・ファイルの各種設定内容の変更が必要になります。また，Hitachi Web Server の一部の機能については，一般ユーザによる運用は制限事項になるものがあります。

ここでは，スーパーユーザと一般ユーザの違い，一般ユーザによる Hitachi Web Server を運用するための環境構築方法，制限事項について説明します。

（1） 各プロセスの権限

スーパーユーザまたは一般ユーザで運用した場合，Hitachi Web Server の各プロセスの権限を次に示します。

表 2-2 各プロセスの権限

項番	プロセス	スーパーユーザによる運用	一般ユーザによる運用
1	制御プロセス	スーパーユーザ	一般ユーザ
2	rotatelogs，rotatelogs2 プロセス		
3	サーバプロセス	User，Group ディレクティブで指定したユーザ，グループ	
4	CGI プロセス		
5	gcache サーバ		

（2）UNIX におけるスーパーユーザと一般ユーザの違い

UNIX において，スーパーユーザは一般ユーザと異なり，システムの管理者権限を持つユーザになります。UNIX におけるスーパーユーザと一般ユーザの権限の差異（一例）を次に示します。

表 2-3 UNIX におけるスーパーユーザと一般ユーザの権限の差異（一例）

項番	項目	スーパーユーザ	一般ユーザ
1	別のユーザが起動したプロセスの停止	可	不可
2	well-known ポート（1023 番以下のポート）を開く	可	不可
3	明示的に読み取り / 書き込み権限が与えられていないファイルへのアクセス	可	不可

一般ユーザで Hitachi Web Server を運用する場合，Hitachi Web Server の制御プロセスの権限が一般ユーザ権限で動作するため，このときの挙動はスーパーユーザで Hitachi Web Server を運用した場合と異なる場合があります。したがって，一般ユーザで Hitachi Web Server を運用する場合は，スーパーユーザとの権限の差異を意識しながら環境を構築する必要があります。

（3）リソースの所有者・グループの変更

Hitachi Web Server のコンテンツ，設定ファイル類，および Hitachi Web Server が動作する際にアクセスする各種ファイル・ディレクトリについて，UNIX 上での所有者・グループを変更します。

最低限，インストールディレクトリ（/opt/hitachi/httpsd ディレクトリ）以下のリソースに対しては変更が必要です。

将来，リソースの所有者・グループを元に戻したい場合は，変更作業の前に現在のリソースに対して，所有者とグループを保存しておきます。

保存作業は，スーパーユーザで実行します。保存例を以下に示します。

（例）

/opt/hitachi/httpsd ディレクトリ以下のリソースに対して，所有者とグループの一覧を作成する。

```
ls laR /opt/hitachi/httpsd
```

変更作業は，スーパーユーザで実行します。変更例を以下に示します。

（例）

/opt/hitachi/httpsd ディレクトリ以下のリソースに対して，所有者 (hwsuser) とグループ (hwsgroup) を変更する。

```
chown R hwsuser:hwsgroup /opt/hitachi/httpsd
```

（4） httpsd の起動

Hitachi Web Server を運用する一般ユーザを使用して，httpsd を起動してください。

httpsd の停止または再起動をする場合は，起動時と同じ一般ユーザで操作してください。

（5） 制限事項

以下に示すユティリティは，一般ユーザによる運用に対応していません。スーパーユーザで運用してください。

- crldownload ユティリティ
- htpasswd ユティリティ
- hwscollect ユティリティ
- hwsserveredit ユティリティ
- logresolve ユティリティ
- sslccert ユティリティ
- sslckey ユティリティ
- sslpasswd ユティリティ

一般ユーザによる運用では以下に示すディレクティブは指定できません。指定があっても無視します。

- Group ディレクティブ
- User ディレクティブ

一般ユーザによる運用では，well-known ポート（1023 番以下のポート）を開くことが

できません。

以下のディレクティブにポート番号を指定する際は注意してください。

- Listen ディレクティブ
- Port ディレクティブ
- SSLCacheServerPort ディレクティブ

3 運用の準備と起動，停止（Windows 版）

この章では，Hitachi Web Server を運用する前に，知っておいていただきたいことおよび起動と停止について説明します。

3.1 Hitachi Web Server を運用するためのシステム構成

Hitachi Web Server を運用するために必要なシステム構成について説明します。

（1） ハードウェア構成

(a) サーバ

Hitachi Web Server の適用機種や，使用するメモリ所要量およびディスク占有量については，リリースノートを参照してください。

(b) クライアント

Web ブラウザが動作できる端末

(c) ネットワーク関連

- イーサネットなどのネットワーク（必須）
- ドメインネームシステムサーバ（任意）
- ロードバランサ（任意）
- SSL アクセラレータ（任意）
- ファイアウォール（任意）

（2） ソフトウェア構成

(a) サーバ

ディレクトリサービス（LDAP サーバ）を利用してユーザ認証およびアクセス保護をする場合に，必要なソフトウェア。

● ユーザを一元管理するためのディレクトリサービスが利用できるソフトウェア
P-1B44-A121 Hitachi Directory Server Version 2　02-01 以降

日立ディレクトリサービスの導入方法については，マニュアル「日立ディレクトリサービス　導入編」を参照してください。

なお，同じ LDAP 対応のディレクトリサービスとして「Sun Java(TM) System Directory Server」も使用できます。

（3） Windows 7，Windows Vista，Windows Server 2008 R2，および Windows Server 2008 使用時の注意事項

(a) コマンド実行時の注意事項

Windows 7，Windows Vista，Windows Server 2008 R2，および Windows Server 2008 使用時の注意事項 Windows Vista および Windows Server 2008 上で Hitachi Web Server を動作させる場合，このマニュアルに記載されているコマンドはすべて管理者権

限で実行する必要があります。Hitachi Web Server のコマンドは，「管理者：コマンドプロンプト」で実行してください。「管理者：コマンドプロンプト」は，Windows 7，Windows Vista，Windows Server 2008 R2，および Windows Server 2008 で提供されている機能を使用して起動してください。

(b) 設定ファイル更新時の注意事項

Windows 7，Windows Vista，Windows Server 2008 R2，および Windows Server 2008 上で Hitachi Web Server の設定ファイルを更新する場合は，更新するプログラムを必ず管理者権限で実行してください。

3.2 インストールとアンインストール

Hitachi Web Server のインストールは，製品として提供されている CD-ROM に格納されている日立総合インストーラを使用する方法と，JP1/NETM/DM および Groupmax Remote Installation を利用してリモートでする方法があります。JP1/NETM/DM については，JP1/NETM/DM のシステム運用のマニュアルを参照してください。

インストールおよびアンインストールは，管理者権限を持つユーザ以外では実行できません。インストールおよびアンインストールするマシンに管理者権限を持つユーザでログインしてください。

3.2.1 インストール

日立総合インストーラによるインストール手順を以下に示します。

(1) インストールする前に

インストールを始める前に，次の内容を確認してください。

- 必要なディスクの空き容量が確保されているかを，確認してください。
- 動作しているアプリケーションをすべて終了させてください。

(2) インストール

日立総合インストーラ (HCD_INST.EXE) を起動し，インストーラの指示に従ってインストールを進めてください。インストール時には次の項目を設定します。

● ユーザ情報
ユーザ名，会社名を入力します。システムから取得した値をデフォルト値として表示します。

● インストール先ディレクトリ
デフォルトでは，OS インストールドライブ ¥Program Files¥Hitachi¥httpsd です。uCosminexus Application Server としてインストールしたときのデフォルトは，OS インストールドライブ ¥Program Files¥Hitachi¥Cosminexus¥httpsd です。なお，インストール先ディレクトリ名として，多バイト文字は使用できません。

● スタートメニューのプログラム名
デフォルトは "Hitachi Web Server" です。

3.2.2 アンインストール

アンインストール方法について説明します。

(1) プログラムの終了

Hitachi Web Server を必ず停止してから，アンインストールを実行してください。

（2） サービスのアンインストール

httpsd コマンドで登録したサービスがある場合，それらを削除してからアンインストールを実行してください。

（3） Hitachi Web Server の削除

［コントロールパネル］-［アプリケーションの追加と削除］から「Hitachi Web Server」を削除してください。uCosminexus Application Server としてインストールした場合のアンインストール方法については，uCosminexus Application Server のマニュアルを参照してください。

（4） ユーザファイルの削除

インストール後に作成されるファイル，コンフィグファイルは削除されません。また，その他ファイルやディレクトリが削除されないで残る場合があります。これらを削除する場合，エクスプローラを使用してください。

3.3 運用環境の定義ファイル

Hitachi Web Server の動作を定義するファイルについて説明します。

(1) ディレクトリ構成

Hitachi Web Server をインストールしたときの，ディレクトリ構成を次に示します。この構成は変更しないでください。

図 3-1 ディレクトリ構成

```
インストール先ディレクトリ
└httpsd             ルートディレクトリ(httpsd.exe)
  ├admin
  │ └bin            複数サーバ環境生成ユティリティ格納ディレクトリ(httpsd.conf.org,
  │                 hwsserveredit.exe, hwsconfigedit.exe)
  ├bin              実行ファイル格納ディレクトリ(htpasswd.exe)
  ├cgi-bin          CGIプログラム格納ディレクトリ
  ├conf             設定ファイル格納ディレクトリ(httpsd.conf, mime.types)
  │ └ssl            SSL用ディレクトリ
  │   ├cacert       CA証明書格納ディレクトリ
  │   ├crl
  │   │ ├DER        CRL格納ディレクトリ(DER形式)
  │   │ └PEM        CRL格納ディレクトリ(PEM形式)
  │   └server       サーバ秘密鍵，サーバ証明書用ディレクトリ
  ├htdocs           デフォルトドキュメントルートディレクトリ(index.html)
  ├icons            アイコン画像格納ディレクトリ
  ├include          ヘッダファイル格納ディレクトリ
  ├libexec          共有ライブラリ格納ディレクトリ
  ├libldap          LDAP用ライブラリ格納場所
  ├logs             ログ，プロセスIDファイル格納ディレクトリ
  ├modules          モジュール格納ディレクトリ(mod_hws_ldap.so, mod_proxy.so,
  │                 mod_expires.so, mod_headers.so, mod_hws_cache.so,
  │                 mod_hws_qos.so, mod_proxy_http.so)
  ├sbin             管理者用ユティリティ(logresolve.exe, rotatelogs.exe,
  │                 rotatelogs2.exe, crldownload.exe, sslpasswd.exe, hwstraceinfo.exe)
  ├servers          複数サーバ環境生成ディレクトリ
  └sslc             BSAFE SSL-Cディレクトリ
    └bin            SSL関連ユティリティ(sslccert.exe, sslckey.exe)
      └demoCA       sslccertユティリティ設定ファイル格納ディレクトリ(sslc.cnf)
```

(2) コンフィグファイル

Hitachi Web Server の動作環境を定義するファイルをコンフィグファイルといいます。なお，コンフィグファイルのコメント行以外に，マルチバイト文字および Unicode の補助文字は指定できません。

Hitachi Web Server の動作を定義するファイルには，httpsd.conf ファイル，mime.types ファイルおよびアクセスコントロールファイル（.htaccess）の三つがあります。SSLC ユティリティの動作を定義するファイルは，sslc.cnf ファイルです。各ファイルの用途を次に示します。

表 3-1 コンフィグファイルの用途

ファイル名	用途	標準提供
httpsd.conf	Hitachi Web Server の動作環境を各種ディレクティブで定義します。システム管理者が管理します。	○
mime.types	コンテンツのファイル拡張子とコンテントタイプ（MIME タイプ）の関連づけを定義します。システム管理者が管理します。	○
.htaccess	アクセス制御を定義するアクセスコントロールファイル。必要に応じてエンドユーザがアクセス制御するディレクトリ下に作成します（デフォルトファイル名は .htaccess）。	×
sslc.cnf	SSL のユティリティについての情報を定義します。システム管理者が管理します。sslc ユティリティだけで使用できます。 認証局（CA）に提出する CSR を作成する場合，ユティリティで使用する値を指定しておくと，効率良く CSR を作成できます。	○
Include ディレクティブで指定したファイル	Hitachi Web Server の動作環境を各種ディレクティブで定義します。システム管理者が管理します。	×

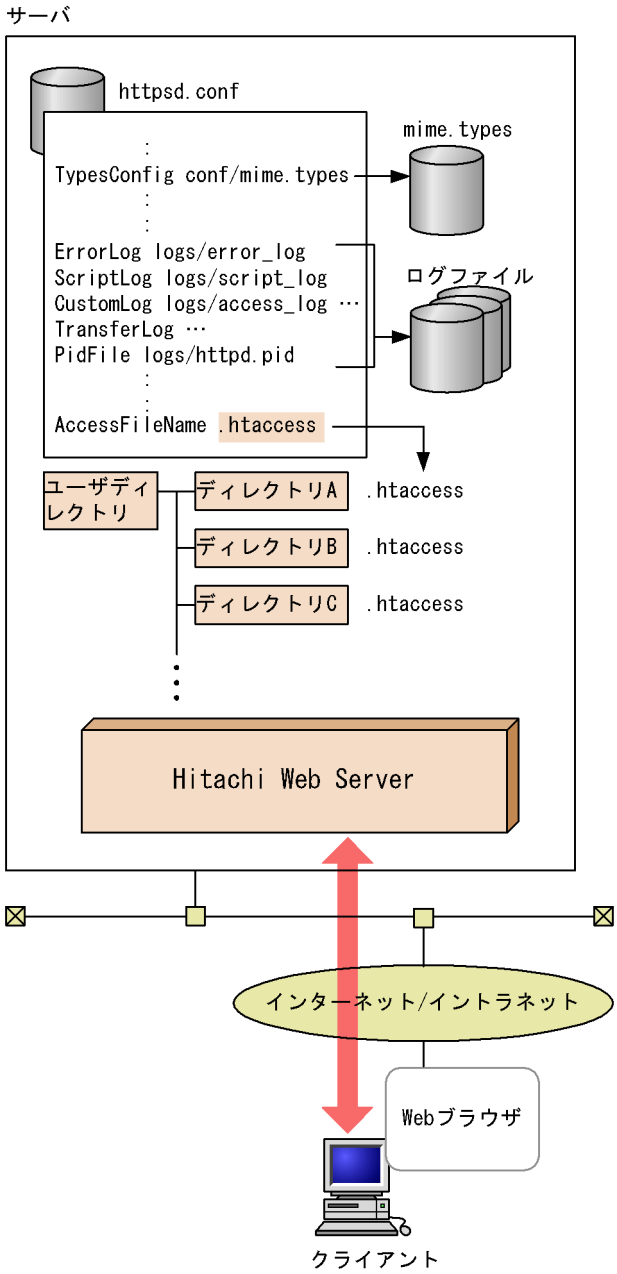
（凡例）

○：標準提供する。

×：標準提供しない。

コンフィグファイルの関連を次に示します。

図 3-2 コンフィグファイルの関連



3.4　起動と停止

Hitachi Web Server の起動および停止方法について説明します。

3.4.1　Hitachi Web Server の起動，停止

Hitachi Web Server をインストールすると，"Hitachi Web Server" という名称のサービスとしてシステムに登録されます。このとき，手動起動するサービスとして登録されるため，システム起動時には自動起動されません。

Hitachi Web Server を起動，停止および再起動するには，次の方法があります。

- スタートメニューからサービスとしての起動，停止および再起動
- コントロールパネルからサービスとしての起動，停止
- コマンドプロンプトからの起動，停止および再起動

Hitachi Web Server をサービスとして実行する場合のユーザアカウントは，インストール時点では "LocalSystem" です。Hitachi Web Server は，CGI プログラム，API 接続モジュールを含め，このユーザアカウントで実行されます。それ以外のユーザアカウントで実行したい場合は，「3.4.2　一般ユーザアカウントによる運用」を参照してください。

（1）スタートメニューからサービスとしての起動，停止および再起動

［スタート］-［プログラム］-［Hitachi Web Server］メニューのショートカットから，起動する場合は「サーバ起動」，停止する場合は「サーバ停止」，再起動する場合は「サーバ再起動」を選択して実行できます。

なお，uCosminexus Application Server としてインストールした場合は，［スタート］-［プログラム］-［Cosminexus］-［Hitachi Web Server］メニューのショートカットになります。

（2）コントロールパネルからサービスとしての起動，停止

コントロールパネルからサービス画面を表示し，次に「Hitachi Web Server」を選択して，起動する場合は「開始 (S)」ボタン，停止する場合は「停止 (T)」ボタンを押します。サービス画面からの再起動はできません。

（3）コマンドプロンプトからの起動，停止および再起動

コマンドプロンプトから httpsd コマンドを入力します。httpsd コマンドについて次に説明します。

(a) 形式

```
"インストール先ディレクトリ¥httpsd.exe" 〔〔-d ディレクトリ〕〔-f ファイル名〕〔〔-n "サービス名"〕 〔-k {start | stop | restart | gracefulstop | install | uninstall}〕〕 | -v | -t 〕
```

(b) オプション

● -d ディレクトリ

ServerRoot ディレクティブがコンフィグファイルに指定されていない場合の，デフォルト値を設定できます。

● -f ファイル名

httpsd.conf ファイルを指定できます。絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスで指定します。

● -n " サービス名 "

Hitachi Web Server のサービス名を指定します。サービス名は，"（引用符）で囲んで指定してください。サービス名に指定できる文字数の上限値は，128 文字です。サービス名は，ASCII コードで指定してください。また，次に示す文字は指定できません。

'¥'，'/'，'"'，制御コード，マルチバイト文字

サービス名のデフォルト値は「Hitachi Web Server」です。

本オプションを指定する場合は，-k オプションも合わせて指定する必要があります。

● -k start

Hitachi Web Server を起動します。-n " サービス名 " が指定されている場合は，該当するサービスを起動します。

● -k stop

Hitachi Web Server を停止します。-n " サービス名 " が指定されている場合は，該当するサービスを停止します。

● -k restart

Hitachi Web Server を再起動します。

● -k gracefulstop

Hitachi Web Server を停止します。実行中のサーバスレッドは，実行終了後に停止します。実行が終了しない場合は，HWSGracefulStopTimeout ディレクティブに指定した待ち時間が経過すると終了します。

● -k install

Hitachi Web Server をサービスとして登録します。-n " サービス名 " が指定されている場合は，該当するサービスを登録します。サービス登録時，スタートアップの種類は「手動」になります。サービス起動する Hitachi Web Server の ServerRoot ディレクティブのデフォルト値は，このコマンド実行時の httpsd.exe のパスまたは -d オプ

ションで指定した値になります。

- -k uninstall
  Hitachi Web Server をサービスから削除します。-n " サービス名 " が指定されている場合は，該当するサービスを削除します。削除しようとしたサービスが起動中の場合は，サービスを停止してからサービスを削除します。
- -v
  バージョン情報を表示します。このオプションを指定した場合は，Hitachi Web Server は起動しません。
- -t
  コンフィグファイルの文法をチェックします。文法エラーがあると，画面にエラーメッセージを表示します。このオプションを指定した場合は，Hitachi Web Server は起動しません。

（4） ターミナルサービスを利用したリモートマシンからの Hitachi Web Server の操作

Hitachi Web Server では，Windows 7，Widnows Vista，Windows XP Professional，Windows Server 2008 R2，Windows Server 2008，Windows Server 2003 R2，および Windows Server 2003 のターミナルサービス機能を利用して，サーバマシン上にある Hitachi Web Server の起動・停止，ユティリティの実行などの操作をリモートマシンから実行できます。ただし，サーバマシンが Windows Server 2003 の場合，クライアントプログラムはサーバマシンのコンソールセション（セション 0）に接続している必要があります。

ターミナルサービスの操作については，OS のマニュアルを参照してください。

（5） 注意事項

スタートメニューまたはコントロールパネルからの停止，コマンドプロンプトからの -k stop オプションによる停止時に，サーバスレッドが実行中の場合には，実行終了を最大 30 秒間待った後に停止します。

3.4.2 一般ユーザアカウントによる運用

Hitachi Web Server をサービスとして実行する場合のユーザアカウントは，インストール時点では "LocalSystem" です。Hitachi Web Server は，CGI プログラム，API 接続モジュールを含め，このユーザアカウントで実行されます。

ここでは，さまざまな権限が与えられたグループには所属しないで，Web サーバの動作に必要な権限だけが設定された一般ユーザアカウントで運用する方法について説明します。

(1) 一般ユーザアカウントの作成

Hitachi Web Server サービスを起動する一般ユーザアカウントを作成する方法について説明します。

一般ユーザアカウントの作成方法

1. **コントロールパネルから［管理ツール］－［コンピュータの管理］を開きます。**
2. **［コンピュータの管理］－［システムツール］－［ローカルユーザーとグループ］－［ユーザー］を開きます。**
3. **操作メニューから［新しいユーザー］を選択し，必要事項を入力します。**
   パスワードは必ず入力してください。

新規作成した一般ユーザアカウントは，デフォルトではグループの設定が付加されています。次の手順に従って，グループの設定を削除してください。

グループの設定の削除方法

1. **コントロールパネルから［管理ツール］－［コンピュータの管理］を開きます。**
2. **［コンピュータの管理］－［システムツール］－［ローカルユーザーとグループ］－［ユーザー］を開きます。**
3. **新規作成したユーザの［プロパティ］を開き，［所属するグループ］タブを表示します。**
4. **登録されているグループを削除します。**

(2) ユーザ権利の割り当て

新規作成した一般ユーザアカウントに，ユーザ権利を割り当てる方法について説明します。

ユーザ権利の割り当て方法

1. **コントロールパネルから［管理ツール］－［ローカルセキュリティポリシー］を開きます。**
2. **［セキュリティの設定］－［ローカルポリシー］－［ユーザー権利の割り当て］を開きます。**
3. **［サービスとしてログオン］をダブルクリックして開きます。**
4. **「ユーザーまたはグループの追加」ボタンで該当するユーザアカウントを追加します。**

［サービスとしてログオン］の権限を明示的に設定しない場合でも，サービスのログオンアカウントを変更した一般ユーザアカウントには，権限が自動的に付加されます。サービスのログオンアカウントの変更については，「(3) サービスのログオンアカウントの変更」を参照してください。

(3) サービスのログオンアカウントの変更

Hitachi Web Server サービスのログオンアカウントを一般ユーザアカウントに変更する方法について説明します。

サービスのログオンアカウントの変更方法

1. **コントロールパネルから［管理ツール］－［サービス］を開きます。**
2. Hitachi Web Server **の［プロパティ］－［ログオン］タブを開きます。**
3. **「アカウント」ラジオボタンを選択し，一般ユーザアカウントを設定します。このとき，「**(1) **一般ユーザアカウントの作成」で設定したパスワードを正しく入力してください。また，パスワードを無期限にするかどうかを指定してください。**

(4) ディレクトリおよびファイルのアクセス権限の設定

Hitachi Web Server がアクセスするディレクトリおよびファイルのアクセス権限に，作成した一般ユーザアカウントのフルコントロール権限を追加してください。

(5) サービスの起動

サービス起動権限を持つユーザアカウントで Hitachi Web Server サービスを起動してください。一般ユーザアカウントには，サービス起動権限はありません。

(6) 注意事項

hwstraceinfo ユティリティを使用する場合は，「(3) サービスのログオンアカウントの変更」で指定した一般ユーザアカウントで実行してください。Administrators 権限を持つユーザアカウントでは実行できません。なお，Windows 7，Windows Vista，Windows Server 2008 R2 および Windows Server 2008 の場合は，hwstraceinfo.exe.manifest ファイルを一時的に別名で保存してから，「(3) サービスのログオンアカウントの変更」で指定した一般ユーザアカウントで実行してください。hwstraceinfo ユティリティを実行したあと，hwstraceinfo.exe.manifest ファイル名を元に戻してください。

4 システムの運用方法

この章では，Web サーバ環境を運用に合わせて設定するディレクティブおよびユティリティの使用方法について説明します。

4.1　Hitachi Web Server の処理とディレクティブとの関係

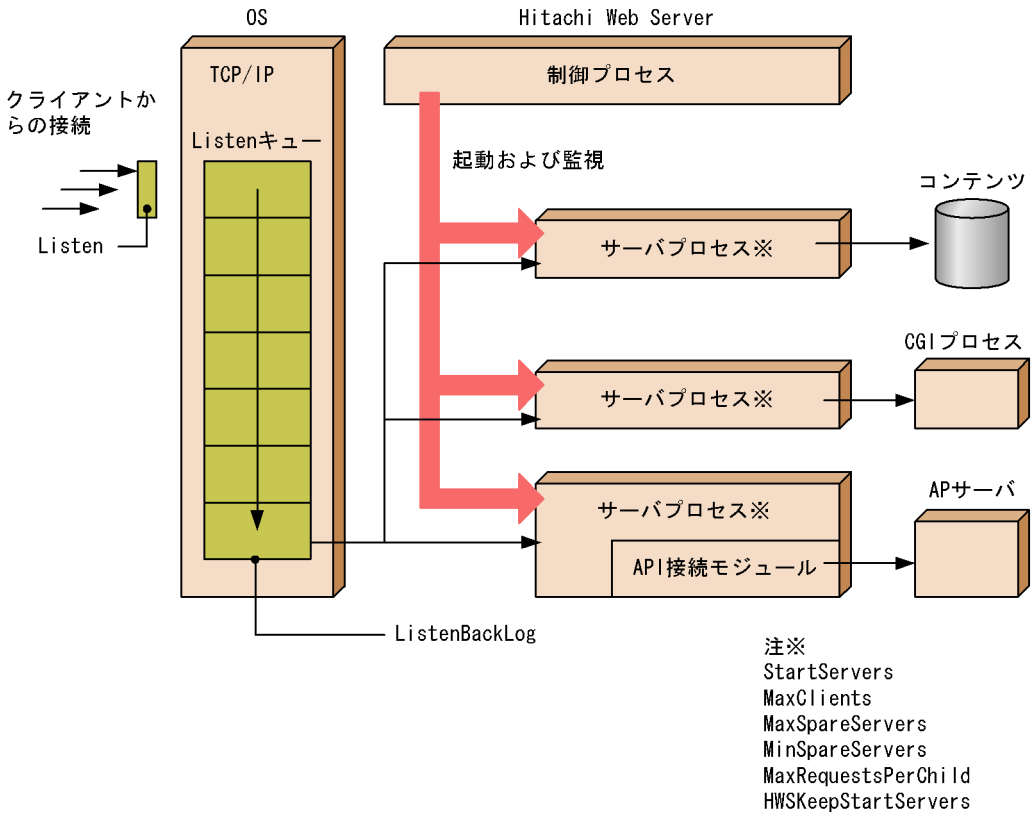
Hitachi Web Server の処理とディレクティブとの関係について，説明します。

4.1.1　Hitachi Web Server のプロセス構造（UNIX 版）

（1）プロセス構造

Hitachi Web Server のプロセス構造を次に示します。

図 4-1　Hitachi Web Server のプロセス構造（UNIX 版）



Hitachi Web Server を起動すると，制御プロセスが起動します。制御プロセスは，リクエストを処理するサーバプロセスを起動し，その稼働を監視します。制御プロセスは，最初に StartServers ディレクティブで指定した個数のサーバプロセスを生成します。その後のサーバプロセス数は，MinSpareServers，MaxSpareServers ディレクティブ指定値に基づいて増減していきます。サーバプロセス数の最大値は，MaxClients ディレク

ティブで指定します。サーバプロセス数の増減は，制御プロセスが管理します。この処理をメンテナンスと呼びます。

クライアントからの TCP 接続は，Listen ディレクティブで指定した IP アドレスとポートから OS が受信し，OS 内の Listen キューに保留します。Listen キューのサイズは，ListenBacklog ディレクティブで指定できます。Listen キューに格納できなかった TCP 接続は確立されません。Listen キューに格納された TCP 接続は，サーバプロセスの一つが取り出して処理を行います。

一つのサーバプロセスは，一つの TCP 接続を取得して処理します。また，一つのサーバプロセスは，MaxRequestsPerChild ディレクティブに指定した個数の HTTP リクエストを処理すると終了します。このときは，制御プロセスが新たなサーバプロセスを生成して処理を続行します。

制御プロセスは，Hitachi Web Server を起動したユーザ，グループ権限で動作します。サーバプロセスは，User，Group ディレクティブで指定したユーザ，グループ権限で動作します。制御プロセスおよびサーバプロセスともに，プロセス名（実行プログラム名）は httpsd です。制御プロセスのプロセス ID は，PidFile ディレクティブに指定したファイルに出力します。

(2) プロセス数の遷移

メンテナンスは，サーバの負荷集中を避けるために，1 秒ごとに 2^{n-1} (n は連続メンテナンス実行回数 ,6 以上は n=6) 個ずつサーバプロセスを生成します。サーバプロセスは，MinSpareServers ディレクティブで指定した数の待ちプロセスができるかまたは全プロセス数が MaxClients ディレクティブで指定した数になるまで生成されます。1 回のメンテナンスで 8 個以上のサーバプロセスを生成した場合は，エラーログ（info レベル）にその旨出力します。

リクエストの処理が終了すると，サーバプロセスは待ち状態になります。待ち状態のプロセスが増加すると，メンテナンスのタイミングで，MaxSpareServers ディレクティブで指定した数だけ残して，サーバプロセスを終了させます。

(a) 留意点

- StartServers ディレクティブには，Web サーバの起動・再起動直後から大量のリクエストを処理しなければならないような場合は，大きな値を指定してください。
- Web サーバを起動した後のプロセス数は，MaxSpareServers および MinSpareServers ディレクティブによって制御されるため，StartServers ディレクティブ指定値は意味がありません（HWSKeepStartServers ディレクティブで On を指定した場合を除く）。MinSpareServers および MaxSpareServers ディレクティブは，急にリクエストが多発しても対応できるように待ち状態のプロセスを準備するために指定します。プロセスのメンテナンスでエラーログ（info レベル）が頻繁に出力されるような場合には，待ちプロセス数を増やすよう調整してください。
- より多くのサーバプロセスを常時待ち状態にしておくと，より多くのクライアントか

らの同時接続要求を受け付けられます。しかし，それだけサーバリソースを消費するために注意が必要です。

- CGI プログラムの負荷が高く CPU を使い尽くしているような場合は，MaxClients ディレクティブの値を小さくしリクエストを受け付けないようにする必要があります。MaxClients ディレクティブで指定した数のプロセスがすべて処理中の場合は，ListenBacklog ディレクティブの指定によって，キューに保留されます。
- 一つのサーバプロセスは，MaxRequestsPerChild ディレクティブで指定された回数のリクエスト処理を実行した後，終了します。ただし，MaxRequestsPerChild ディレクティブに 0 を指定した場合，リクエスト処理の回数でサーバプロセスが終了することはありません。MaxRequestsPerChild ディレクティブの指定は，エンドユーザが作成したアプリケーションプログラムなどでメモリリークを起こすおそれがある場合に有効です。
- サーバプロセスに異常終了シグナルが送信された場合（API 接続モジュールで障害になった場合も含む），プロセスは，エラーログ（notice レベル）にそのことを出力します。notice レベルのエラーログは，LogLevel ディレクティブの指定に関係なく出力されます。
- StartServers ディレクティブで指定した数のサーバプロセスを，MaxSpareServers，MinSpareServers ディレクティブの指定に関係なく，常に起動しておきたい場合，HWSKeepStartServers ディレクティブで On を指定してください。サーバプロセス数が StartServers ディレクティブで指定した数を下回った場合に，新しいプロセスを生成して回復します。
- Hitachi Web Server が停止すると PidFile ディレクティブに指定したファイルは削除されます。しかし，Hitachi Web Server を停止しないでマシンをシャットダウンした場合など，Hitachi Web Server が外部から強制的に終了させられたときには，PidFile ディレクティブに指定したファイルは削除されないで残るため，注意してください。

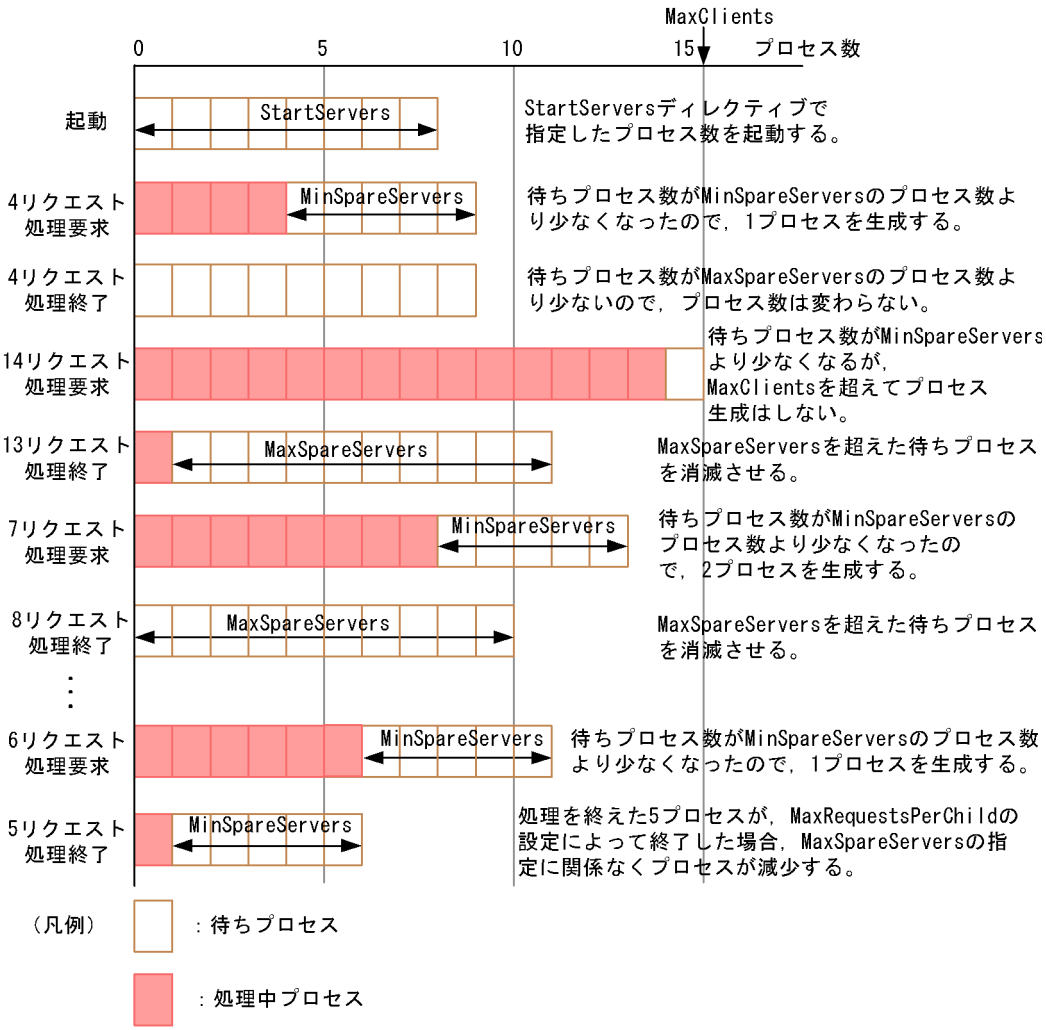
(b) プロセス数の遷移例

HWSKeepStartServers の指定が Off の場合の，プロセス数の遷移例を図 4-2 に示します。

● ディレクティブの指定値（HWSKeepStartServers の指定が Off の場合）

```
StartServers 8
MaxSpareServers 10
MinSpareServers 5
MaxClients 15
HWSKeepStartServers Off
MaxRequestsPerChild 10000
KeepAlive Off
```

図 4-2　プロセス数の遷移例（HWSKeepStartServers の指定が Off の場合）

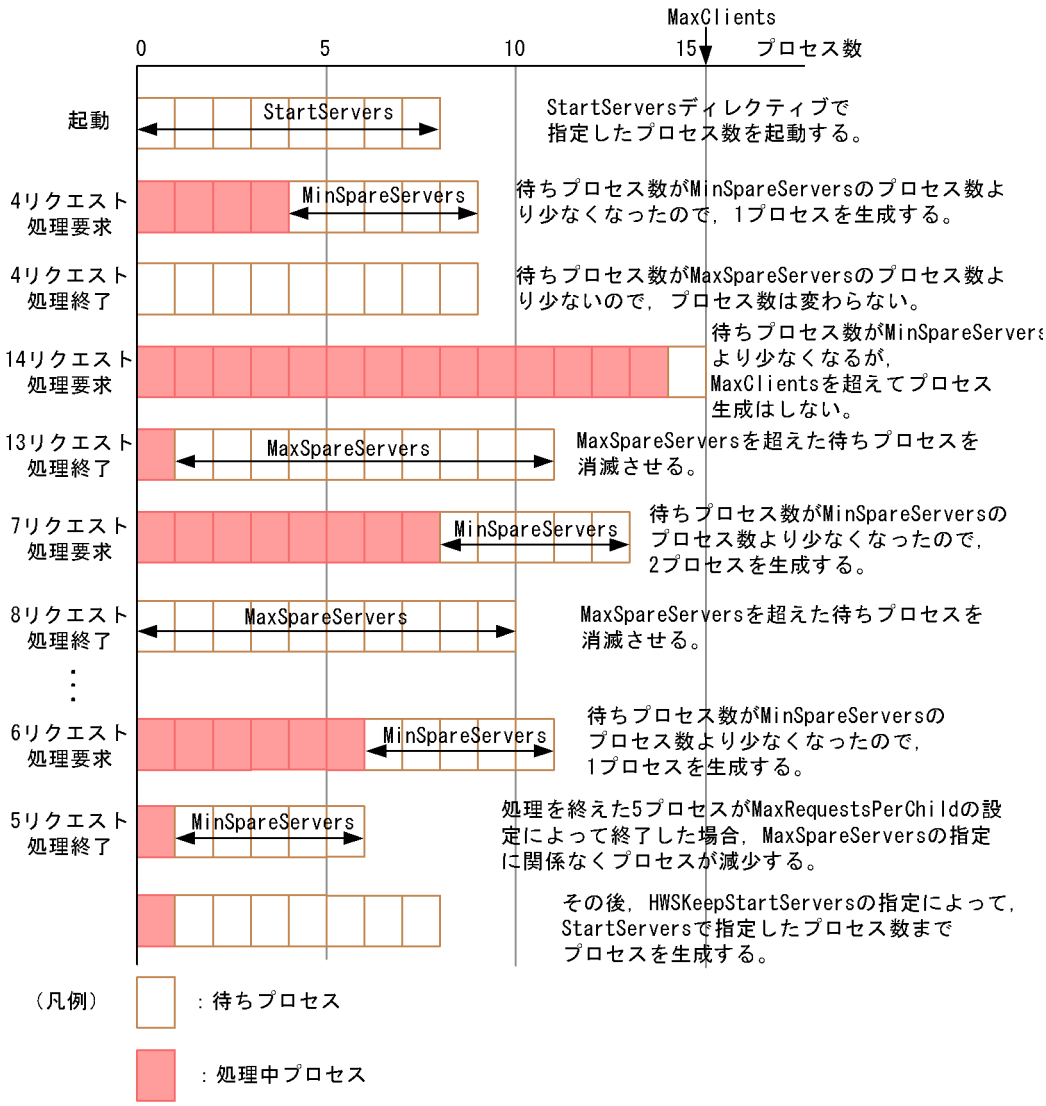


HWSKeepStartServers の指定が On の場合の，プロセス数の遷移例を図 4-3 に示します。

● ディレクティブの指定値（HWSKeepStartServers の指定が On の場合）

```
StartServers 8
MaxSpareServers 10
MinSpareServers 5
MaxClients 15
HWSKeepStartServers On
MaxRequestsPerChild 10000
KeepAlive Off
```

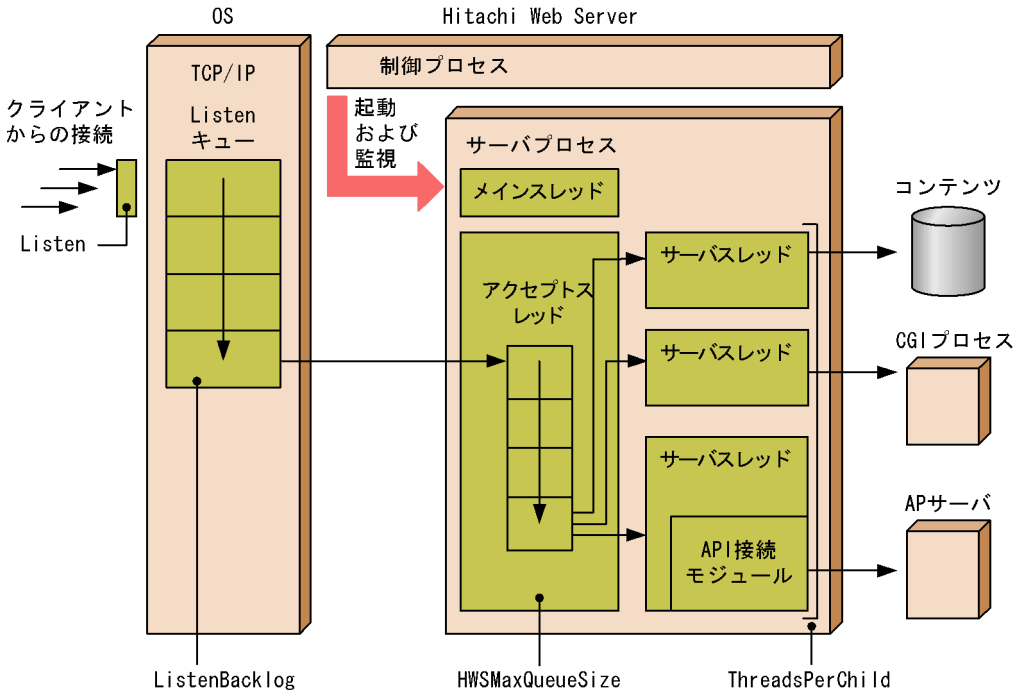
図 4-3　プロセス数の遷移例（HWSKeepStartServers の指定が On の場合）



4.1.2　Hitachi Web Server のプロセス構造（Windows 版）

Hitachi Web Server のプロセス構造を次に示します。

図 4-4　Hitachi Web Server のプロセス構造（Windows 版）



Hitachi Web Server を起動すると，制御プロセスが起動します。制御プロセスは，サーバプロセスを起動しその稼働を監視します。サーバプロセスが起動すると同時にメインスレッドが起動します。メインスレッドは，リクエストを受信するアクセプトスレッドを一つと，リクエストを処理するサーバスレッドを ThreadsPerChild ディレクティブで指定した個数だけ起動します。

クライアントからの TCP 接続は，Listen ディレクティブで指定した IP アドレスとポートから OS が受信し，OS 内の Listen キューに保留します。Listen キューのサイズは ListenBacklog ディレクティブで指定できます。Listen キューに格納できなかった TCP 接続は確立されません。アクセプトスレッドは Listen キューから TCP 接続を取り出し，HWSMaxQueueSize ディレクティブで指定した大きさのリクエストキューに登録します。それをサーバスレッドの一つが取り出して HTTP リクエストを受信し処理します。HWSMaxQueueSize ディレクティブ指定値を超えたためリクエストキューに格納できなかった TCP 接続は，アクセプトスレッドによって閉じられます。

サーバスレッド数が増減することはありません。

4.1.3 稼働管理について

サーバプロセス（Windows 版の場合はサーバスレッド）の稼働状況を管理する上で必要となる，持続型接続の動作原理とタイマ監視機能について，説明します。

(1) 持続型接続（KeepAlive）

持続型接続（KeepAlive）は，クライアントからのリクエストに対してレスポンスを返した後も TCP コネクションを切断しないで，同じクライアントからの次のリクエストを待つ機能です。

この機能は，KeepAlive ディレクティブに On を指定し，かつクライアント側が対応している場合に使用できます。TCP コネクションを切断しないため，クライアントが複数のリクエストを連続して送信する場合にはレスポンス時間を短縮できます。

次のリクエストを待つ間はサーバプロセスがクライアントに占有されますが，この待ち時間は KeepAliveTimeout ディレクティブで設定できます。また，1 クライアントが持続型接続を利用して何回までリクエストを処理できるかを MaxKeepAliveRequests ディレクティブで設定します。

(2) タイマ監視

次の場合，Timeout ディレクティブに設定された値によって，タイマ監視ができます。

- クライアントからのリクエスト受信（コネクション確立後，HTTP プロトコルの受信）時
- クライアントへのレスポンス送信時
- CGI プログラムへのリクエスト送信時
- CGI プログラムへのリクエスト送信後からレスポンス受信まで
- CGI プログラムからのレスポンス受信時
- リバースプロキシを使用している場合，Web サーバへのリクエスト送信後からレスポンス受信まで
- リバースプロキシを使用している場合，Web サーバからのレスポンス受信時

Timeout ディレクティブに設定された値を過ぎてもレスポンスがない（タイムアウトになった）場合は，その旨エラーログ（info レベル）に出力し，コネクションを切断します。

(3) 制御プロセスの監視

PidFile ディレクティブで指定したファイルに出力される ID のプロセスを監視すると，Hitachi Web Server の制御プロセスを監視できます。監視するプロセス名（実行プログラム名）は Windows 版の場合 httpsd.exe，UNIX 版の場合 httpsd です。

制御プロセスを監視する際には，PidFile ディレクティブで指定したファイルに格納された ID のプロセスが Hitachi Web Server のプロセスであることを必ず確認してください。

Hitachi Web Server のプロセスであることを確認するには，プロセスの実行プログラム名が，Windows 版の場合 httpsd.exe，UNIX 版の場合 httpsd であることを確認してください。

4.2 ログを採取する

4.2.1 ログの種類

ログの種類を次に示します。

表 4-1 ログの種類

ログの種類	指定するディレクティブ	機能
アクセスログ	TransferLog	• デフォルトフォーマットのログを採取します。 • rotatelogs ユティリティまたは rotatelogs2 ユティリティを使用して，定期的または定量的に分割できます。 • LogFormat ディレクティブでフォーマットを変更できます。
	CustomLog	• カスタマイズしたフォーマットでログを採取します。 • rotatelogs ユティリティまたは rotatelogs2 ユティリティを使用して，定期的または定量的に分割できます。 • LogFormat ディレクティブで定義したフォーマットを CustomLog ディレクティブに指定できます。
エラーログ	ErrorLog	• エラー発生時のメッセージのログを採取します。 • rotatelogs ユティリティまたは rotatelogs2 ユティリティを使用して，定期的または定量的に分割できます。 • LogLevel ディレクティブで，採取するログのレベルを指定できます。 • HWSRequestLog ディレクティブを指定していない場合に，モジュールトレースを採取できます。 モジュールトレースの詳細については「4.2.6 モジュールトレースの採取」を参照してください。
	ScriptLog	• CGI スクリプトのエラーログを採取します。
リクエストログ	HWSRequestLog	• リクエストログを採取します。リクエストログとして，次のトレースを採取できます。 **モジュールトレース** モジュールの各関数の実行時および CGI プログラムの実行時に採取されるトレースです。モジュールトレースの詳細については「4.2.6 モジュールトレースの採取」を参照してください。 **リクエストトレース** リクエスト処理開始時や完了時などで採取されるトレースです。リクエストトレースの詳細については「4.2.7 リクエストトレースの採取」を参照してください。 I/O **フィルタトレース** モジュールが実装している入出力フィルタ関数の実行時に採取されるトレースです。I/O フィルタトレースの詳細については「4.2.8 I/O フィルタトレースの採取」を参照してください。 • rotatelogs ユティリティまたは rotatelogs2 ユティリティを使用して，定期的または定量的に分割できます。
プロセス ID ログ	PidFile	• 制御プロセス ID のログを採取します。

ログの種類	指定するディレクティブ	機能
イベントログ	なし	• サービスから起動する際に発生するエラーを記録します（Windows 版）。
内部トレース	HWSTraceLogFile	• 共有メモリのトレース情報を出力します。
共有メモリ ID ログ	HWSTraceIdFile	• 共有メモリ ID を格納します。
コアファイル※	CoreDumpDirectory	• Hitachi Web Server 障害発生時のコアダンプの出力先を指定します（UNIX 版）。
	SSLCacheServerRunDir	• gcache サーバ（SSL で利用）障害発生時のコアダンプの出力先を指定します（UNIX 版）。

注※
OS でコアファイルを出力する設定を行った場合に出力されます。
設定方法については各 OS のマニュアルを参照してください。

アクセスログ，エラーログ，リクエストログのサイズが 2GB を超えた場合，Hitachi Web Server が異常終了したり，再起動できない場合があります。定期的にログファイルを退避するか，「4.2.3　ログを分割する」や「4.2.4　ログファイルをラップアラウンドさせて使用する」を参照してログファイルのサイズが 2GB を超えないように設定してください。

4.2.2　ログの採取方法

アクセスログ，エラーログ，プロセス ID のログおよびリクエストログの採取方法について説明します。

(1) アクセスログ

(a) デフォルトフォーマットのアクセスログ

TransferLog ディレクティブを指定して，ログを採取します。

デフォルトフォーマットのアクセスログの形式を次に示します。

```
クライアントホスト名△クライアントの識別情報△クライアントユーザ名△アクセス時刻△"リクエストライン"△ステータスコード△送信バイト数
```

（凡例）
△：空白

（出力例）

```
172.17.40.30 - - [25/Dec/2000:16:23:59 +0900] "GET / HTTP/1.0" 200 3546
```

(b) カスタムフォーマットのアクセスログ

CustomLog ディレクティブを指定して，ログを採取します。フォーマットの指定方法には，二つあります。

- 直接 CustomLog ディレクティブにフォーマットを指定する

(例)

```
CustomLog logs/access.log "%h %l %u %t ¥"%r¥" %>s %b"
```

- LogFormat ディレクティブでフォーマットに対するラベル名を定義して，そのラベル名を CustomLog ディレクティブに指定する

(例)

```
LogFormat "%h %l %u %t ¥"%r¥" %>s %b" common
CustomLog logs/access.log common
```

(2) エラーログ

(a) エラーメッセージログ

ErrorLog ディレクティブを指定して，ログを採取します。LogLevel ディレクティブで採取するエラーのレベルを指定します。

(b) CGI スクリプトのエラーログ

ScriptLog ディレクティブを指定して，CGI スクリプトのエラーログを採取します。

(3) プロセス ID のログ

PidFile ディレクティブを指定して，制御プロセス ID のログを採取します。

(4) リクエストログ

HWSRequestLog ディレクティブと HWSRequestLogType ディレクティブを指定して，リクエストログを採取します。リクエストログとは，モジュールトレース，リクエストトレースおよび I/O フィルタトレースの総称です。

モジュールトレースの詳細については「4.2.6　モジュールトレースの採取」，リクエストトレースの詳細については「4.2.7　リクエストトレースの採取」，I/O フィルタトレースの詳細については「4.2.8　I/O フィルタトレースの採取」を参照してください。

(5) 各トレースの出力先

(a) モジュールトレースの出力先

モジュールトレースの出力先は，エラーログまたはリクエストログのどちらか一方になります。どちらに出力されるかは，ディレクティブの指定によって決まります。モジュールトレースの出力先と出力条件を次に示します。

表 4-2 モジュールトレースの出力先と出力条件

出力先	出力条件
リクエストログ	HWSRequestLog ディレクティブの指定があり，かつ，HWSRequestLogType ディレクティブに module-info または module-debug を指定した場合
エラーログ	HWSRequestLog ディレクティブの指定がなく，かつ，LogLevel ディレクティブに info または debug を指定した場合

モジュールトレースの詳細については「4.2.6 モジュールトレースの採取」を参照してください。

(b) リクエストトレースおよび I/O フィルタトレースの出力先

リクエストトレースおよび I/O フィルタトレースの出力先はリクエストログになります。

HWSRequestLog ディレクティブの指定があり，かつ，HWSRequestLogType ディレクティブが出力条件を満たしている場合にリクエストログに出力されます。
HWSRequestLogType ディレクティブの出力条件については，「4.2.7 リクエストトレースの採取」および「4.2.8 I/O フィルタトレースの採取」を参照してください。

4.2.3 ログを分割する（rotatelogs ユティリティ）

アクセスログやエラーログを一定時間単位（例えば，24 時間ごと）に分割して，複数のファイルに出力できます。rotatelogs ユティリティは次のディレクティブに指定できます。

- CustomLog ディレクティブ
- ErrorLog ディレクティブ
- HWSRequestLog ディレクティブ
- TransferLog ディレクティブ

ユティリティの指定方法を次に示します。

(1) 形式

```
rotatelogs 分割ログファイルのプリフィックス ログ分割時間間隔 〔-fnum ファイル数〕〔-diff GMTに対する時差〕
```

(2) オペランド

● 分割ログファイルのプリフィックス

分割ログファイルのプリフィックスを絶対パスで指定します。
「プリフィックス .nnnnnnnnnn」というファイルに，ログを採取します。

nnnnnnnnnn：ログ採取開始時刻を表します。ログ採取時刻とは次の式で示す値です。

（(1970 年 1 月 1 日の 0 時 0 分 0 秒 (GMT：Greenwich Mean Time) を起点とした，ログを出力する時間の通算秒数÷ログ分割時間間隔）の小数点以下を切り捨てた値）×ログ分割時間間隔

● **ログ分割時間間隔**　～ ((1 － 31536000))
一つのログファイルを採取する時間間隔を秒単位に指定します。指定した時間が経過するごとに，新規ファイルにログを採取します。

● -fnum **ファイル数**　～ ((1 － 256))
分割したログファイルのファイル数を指定します。分割したファイル数がここで指定した数を超えた場合，最も古いファイルから削除されます。このオペランドを省略した場合，ファイルは削除されません。

● -diff GMT **に対する時差**　～ ((-1439 － 1439))
ログファイルを分割する基準となる時間を，GMT に対する時差として分単位で指定します。指定しないまたは 0 を指定すると，1970 年 1 月 1 日 0 時 0 分 0 秒 (GMT) が基準時間となります。GMT に対するローカルタイムの差が n 時間である場合に，ローカルタイムの m 時 0 分 0 秒を基準にする場合には，(n-m) × 60 を指定します。JST の 0 時 0 分 0 秒を基準にする場合には，(+9-0) × 60 で 540 を指定します。

(3) 使用方法

ディレクティブに，"| プログラム名 " の形式で指定して使用します。ログファイルを定期的に別ファイルに分割して採取します。

(例) Windows 版　24 時間ごとに，アクセスログを分割して C:¥Program Files¥Hitachi¥httpsd¥logs¥access.nnnnnnnnnn ファイルに採取します。分割時間を日本時間に設定し，日本時間の毎 0 時にログファイルを分割する場合の指定を次に示します。

```
TransferLog "|¥"¥"C:/Program Files/Hitachi/httpsd/sbin/rotatelogs.exe¥" ¥"C:/
Program Files/Hitachi/httpsd/logs/access¥" 86400 -diff 540¥""
```

ログファイル名：C:¥Program Files¥Hitachi¥httpsd¥logs¥access.nnnnnnnnnn

ログ分割時間間隔：86400 秒（＝ 24 時間）

(例) UNIX 版　24 時間ごとに，アクセスログを分割して /opt/hitachi/httpsd/logs/access.nnnnnnnnnn ファイルに採取します。分割時間を日本時間に設定し，日本時間の毎 0 時にログファイルを分割する場合の指定を次に示します。

```
TransferLog "|/opt/hitachi/httpsd/sbin/rotatelogs /opt/hitachi/httpsd/logs/
access 86400 -diff 540"
```

ログファイル名：/opt/hitachi/httpsd/logs/access.nnnnnnnnnn

ログ分割時間間隔：86400 秒（＝ 24 時間）

(4) 注意事項

- -fnum オペランドの指定によるログファイルの制御は Web サーバの再起動時に，ディレクトリ名またはログファイルのプリフィックスを変更すると，以前に採取したログファイルは削除されません。この場合は運用に応じて削除してください。
- Web サーバを起動または再起動してから，指定したログ分割間隔時間が経過した場合，分割したログファイルのプリフィックスに一致するファイルの数が -fnum オペランドの指定値を超えると，作成時間の古いログファイルから削除されます。
- 分割ログファイルのプリフィックスは絶対パスで指定してください。
- サービスとして起動した場合には，制御プロセスのログは採取されません（Windows 版）。
- ログファイルは，そのファイルを開いているプロセスがある間は削除できません。このため，-fnum で指定した値より多いファイルが残ることがあります。例えば，制御プロセスがログを出力したファイルは，制御プロセスが終了するまで削除されません（Windows 版）。
- rotatelogs ユティリティは SIGTERM，SIGUSR1 および SIGHUP シグナルを受信してもプロセス終了処理を実施しませんが，制御プロセスとサーバプロセスが終了すればプロセス終了します（UNIX 版）。

4.2.4 ログファイルをラップアラウンドさせて使用する（rotatelogs2 ユティリティ）

アクセスログやエラーログをログファイルサイズで分割して，複数のファイルにラップアラウンドして出力できます。rotatelogs2 ユティリティは次のディレクティブに指定できます。

- CustomLog ディレクティブ
- ErrorLog ディレクティブ
- HWSRequestLog ディレクティブ
- TransferLog ディレクティブ

ユティリティの指定方法を次に示します。

(1) 形式

```
rotatelogs2 ログファイルプリフィックス名　ログファイルサイズ　ログファイル個数
```

(2) オペランド

● ログファイルプリフィックス名

出力するログファイルのプリフィックス名を絶対パスで指定します。
出力するログファイルは「プリフィックス .nnn」のファイル名となります。

.nnn は .001 からログファイル個数で指定した値までです。
「ログファイル個数」を nnn 面とすると，nnn 面のうち，Hitachi Web Server 起動時の更新時刻が最新のものが，カレントのログファイルとなります。ログファイルは，ファイル名称のプリフィックスに拡張子 .001 ～ .nnn を付けて区別します。カレントのログファイルの拡張子が .mmm であった場合，カレントのログファイルがいっぱいになると，続きは，.mmm+1 のログファイルをクリアして出力されます。.mmm が .nnn と一致した場合，続きは .001 に出力されます。
Windows 版の場合，「プリフィックス .index」のインデクス番号格納用ファイルが作成されます。このファイルは .nnn 管理用ファイルであり，rotatelogs2 ユティリティの起動時に作成され，停止時に削除されます。ただし，起動エラーの一部などで削除されないことがありますが，以降の Web サーバの動作に影響はありません。

● **ログファイルサイズ** ～ ((1 － 2097151))
ログファイルの最大サイズ（単位：KB）を指定します。
ログを出力するタイミングで，最大サイズを超えていると，次のログファイルをクリアして続きが出力されます。

● **ログファイル個数** ～ ((1 － 256))
出力するログファイルの最大数を指定します。
最大サイズを超えて次のログファイルに移る場合，それまで処理していたログファイルの拡張子が最大個数と同じとき，再度 .001 のファイルから使用します。

(3) 使用方法

ディレクティブに，"| プログラム名 " の形式で指定して使用します。

(例) 4,096KB ごとにエラーログを最大 5 個採取する場合

```
ErrorLog "|¥"¥"C:/Program Files/Hitachi/httpsd/sbin/rotatelogs2.exe¥" ¥"C:/
Program Files/Hitachi/httpsd/logs/errorlog¥" 4096 5¥""
```

errorlog.001 ～ errorlog.005 の順番にログが出力されます。errorlog.005 が 4,096KB を超えると errorlog.001 をクリアして続きが出力されます。Hitachi Web Server 起動時に，すでにこれらのログファイルがある場合には，更新時刻の最も新しいログファイルが出力対象のログファイルとなります。このログファイルのサイズがすでに 4,096KB を超えている場合には，次のログファイルをクリアして続きが出力されます。4,096KB を超えない場合は，このファイルの続きに出力されます。

(4) 注意事項

- ログファイルプリフィックス名は絶対パスで指定してください。
- Hitachi Web Server 起動時の出力ログファイルは，更新日時が最新のものを対象とするため，誤ってファイルを更新した場合は正しいファイルへの出力ができなくなります。
- ログファイルサイズには，同一秒内に複数のファイルが指定サイズを超えるような小さいサイズを指定しないでください。このようなサイズを指定した場合には，最も新

しいログファイル以外が出力対象となり正しくローテーションされなくなることがあります。

- コンフィグファイル内に，同一のログファイルプリフィックス名を複数個所で指定しないでください。複数個所で指定した場合には，最も新しいログファイル以外が出力対象となり正しくローテーションされなくなることがあります。
- Web サーバはサービスとして起動してください。サービスとして起動しない場合，Web サーバの停止または再起動の際に，不当にログファイルがクリアされることがあります（Windows 版）。
- インデクス番号格納用ファイルは，rotatelogs2 ユティリティが動作中の間は絶対に編集や削除をしないでください。編集するとログが正しく出力されないことがあります（Windows 版）。
- Web サーバの起動時にインデクス番号格納用ファイル「プリフィックス .index」と同名のファイルが存在した場合，ファイルは上書きされます（Windows 版）。
- rotatelogs2 ユティリティは SIGTERM，SIGUSR1 および SIGHUP シグナルを受信してもプロセス終了処理を実施しませんが，制御プロセスとサーバプロセスが終了すればプロセス終了します（UNIX 版）。

4.2.5 ログファイルの IP アドレスをホスト名に変換する（logresolve ユティリティ）

logresolve ユティリティは，レコードの先頭が IP アドレスであるアクセスログファイル内の IP アドレスをホスト名に変換し，新規ログファイルに出力します。変換規則は，ホスト名のルックアップの逆引きによります。

（1）形式

```
logresolve 〔-s ファイル名〕 〔-c〕 < アクセスログファイル名 > 新ログファイル名
```

（2）オペランド

● -s ファイル名

変換したときの情報を出力するファイルを指定します。このファイルには次のような情報が出力されます。

- IP アドレスと変換後のホスト名
- 変換できなかった IP アドレス
- 変換した IP アドレスの数

● -c

変換後のホスト名が変換前の IP アドレスと一致するかどうかチェックする場合に指定します。

● アクセスログファイル名

入力ログファイル名を指定します。入力したファイルの IP アドレスからホスト名のルックアップの逆引きをします。レコードの先頭は，必ず IP アドレスでなければなりません。ホスト名の検索に失敗した場合，新ログファイルには IP アドレスが出力されます。

● 新ログファイル名

IP アドレスをホスト名に変換したアクセスログを出力するファイル名を指定します。

(3) 使用方法

logs¥access.log に格納しているアクセスログ内の IP アドレスをホスト名に変換します。

アクセスログファイル：logs¥access.log

新ログファイル：logs¥new_access.log

```
logresolve < logs¥access.log > logs¥new_access.log
```

4.2.6 モジュールトレースの採取

Web サーバは複数のモジュール※から構成され，これらのモジュールは特定のタイミングで実行される複数の関数から構成されています。**モジュールトレース**とは，モジュールの各関数の実行時および CGI プログラムの実行時に採取されるトレースのことです。モジュールトレースは，HWSRequestLog ディレクティブの指定の有無によって，採取先などの採取方法が変わります。

注※ モジュールには，Web サーバに LoadModule ディレクティブで動的に組み込んで使用する外部モジュールと，httpsd 実行ファイルに含まれる内部モジュールとがあります。

(1) トレース対象

モジュールトレースのトレース対象を次に示します。

表 4-3 モジュールトレースのトレース対象

トレース対象	トレースの契機
モジュール	モジュールは複数の関数から構成されています。これらの関数は，初期化処理とリクエスト対応処理に分類されます。トレースはリクエスト対応処理についての関数に対して採取します。
CGI プログラム	CGI プログラム実行時にトレースを採取します。

(2) 採取方法

HWSRequestLog ディレクティブを指定していない場合は，ErrorLog ディレクティブに指定したファイルに対して，LogLevel ディレクティブの指定に従って採取します。

HWSRequestLog ディレクティブを指定している場合は，HWSRequestLog ディレクティブに指定したファイルに対して，HWSRequestLogType ディレクティブの指定に従って採取します。

注意事項

ErrorLog ディレクティブに指定したファイルに対してログを採取する場合は，バーチャルホスト単位にファイルを分けることができます。HWSRequestLog ディレクティブに指定したファイルに対してログを採取する場合は，バーチャルホスト単位にファイルを分けることができません。

(3) 採取レベル

LogLevel ディレクティブまたは HWSRequestLogType ディレクティブの指定によって，採取するモジュールトレースのレベルを変更できます。各レベルで採取するトレースの内容を次に示します。

(a) info レベル

障害発生の原因となるおそれのある外部モジュールおよび CGI プログラムについて採取します。

LogLevel ディレクティブに info を指定または HWSRequestLogType ディレクティブに module-info を指定した場合に採取します。

(b) debug レベル

info レベルのほかに，リクエストごとに動作する内部モジュールについてのトレースも採取します。

LogLevel ディレクティブに debug を指定または HWSRequestLogType ディレクティブに module-debug を指定した場合に採取します。

(4) トレースフォーマット

モジュールトレースの出力項目は次のとおりです。

なお，以降の記述で「サーバプロセス ID」は Windows 版の場合「サーバスレッド ID」です。

(a) モジュール

● info レベルで出力される場合

コール時

```
[時刻]△[info]△hws△:△module△-->△(モジュールファイル名称[関数オフセット])(サーバプロセスID)
```

リターン時

```
[時刻]△[info]△hws△:△module△<--△(モジュールファイル名称[関数オフセット])(サーバプロセスID)(結果コード)
```

（凡例）

△：空白

関数オフセットと関数の対応を次に示します。

表 4-4　関数オフセットと関数の対応

関数オフセット	関数名	意味
[0]	create_request	リクエスト開始処理を実行中
[1]	post_read_request	リクエスト読み込み後処理を実行中
[2]	quick_handler	リクエストされた URL 変換前処理を実行中
[3]	translate_name	リクエストされた URL から，実際のファイル名への変換処理実行中
[4]	map_to_storage	ディスクアクセスを伴わないリクエスト処理を実行中
[5]	header_parser	リクエストヘッダ解析処理実行中
[6]	access_checker	認証済みユーザからリクエストされた URL に対してホスト名および IP アドレスによるアクセス権限チェック実行中
[7]	check_user_id	ユーザ ID のチェック処理実行中
[8]	auth_checker	認証済みユーザからリクエストされた URL に対してアクセス権限（Require）チェック実行中
[9]	type_checker	MIME タイプのチェック処理実行中
[10]	fixups	リクエスト実行前処理を実行中
[11]	insert_filter	フィルタ挿入処理を実行中
[12]	handler	ハンドラを実行中
[13]	insert_error_filter	エラー応答前処理を実行中
[14]	log_transaction	ログ出力処理を実行中
[15]	error_log	エラーログ出力後処理を実行中
[16]	get_suexec_identity	ユーザ情報取得処理を実行中

（出力例）

```
[Fri Jul 15 17:29:43 2005] [info] hws : module --> (mod_example.c[1])(1864)
[Fri Jul 15 17:29:43 2005] [info] hws : module <-- (mod_example.c[1])(1864)(-1)
```

● debug レベルで出力される場合

コール時

[時刻]△[debug]△ファイル名称(行番号):△hws△:△module△-->△(モジュールファイル名称[関数オフセット])(サーバプロセスID)

リターン時

[時刻]△[debug]△ファイル名称(行番号):△hws△:△module△<--△(モジュールファイル名称[関数オフセット])(サーバプロセスID)(結果コード)

(凡例)

△：空白

(出力例)

```
[Fri Jul 15 17:29:43 2005] [debug] request.c(69): hws : module -->
(mod_alias.c[3])(1864)
[Fri Jul 15 17:29:43 2005] [debug] request.c(69): hws : module <--
(mod_alias.c[3])(1864)(-1)
```

(b) CGI プログラム

● info レベルで出力

コール時

[時刻]△[info]△hws△:△cgi△-->△(exec=cgiファイル名称)(argv0=実行プログラム名称)(args=引数 ※)(サーバプロセスID)(CGIプロセスID)

注※ args による引数は，GET /cgi-bin/isindex?aaa+bbb+ccc HTTP/1.0 のように，= ではなく，+ で連結されたクエリーが指定された場合にだけ表示します。

リターン時

[時刻]△[info]△hws△:△cgi△<--△(exec=cgiファイル名称)(argv0=実行プログラム名称)(サーバプロセスID)(CGIプロセスID)

(凡例)

△：空白

（出力例）

```
[Fri Jul 15 19:48:08 2005] [info] hws : cgi --> (exec=C:/Program Files/Hitachi/
httpsd/cgi-bin/isindex)(argv0=isindex)(args=aaa+bbb+ccc)(1784)(1144)
[Fri Jul 15 19:48:08 2005] [info] hws : cgi <-- (exec=C:/Program Files/Hitachi/
httpsd/cgi-bin/isindex)(argv0=isindex)(1784)(1144)
```

(5) 使用方法

(a) 使用例

リクエストログに，info レベルのモジュールトレースおよびリクエストトレースを出力する例を示します。

```
HWSRequestLogType module-info request
HWSRequestLog logs/hwsrequest.log
```

(b) 異常時のトレース例

● 外部モジュールで異常が発生した場合

```
[Fri Jul 15 10:29:29 2005] [info] hws : module --> (mod_example.c[1])(1800)
[Fri Jul 15 10:29:30 2005] [notice] Parent: child process exited with status
3221225477 -- Restarting.
```

mod_example.c[1]，つまり，post_read_request に該当する関数内で異常が発生しました。
mod_example.c[1] に該当する関数を調査してください。

● CGI プログラムからレスポンスがない場合

```
[Fri Jul 15 19:48:39 2005] [info] hws : cgi --> (exec=C:/Program Files/Hitachi/
httpsd/cgi-bin/test-sleep)(argv0=test-sleep)(1800)(2276)
[Fri Jul 15 19:48:49 2005] [info] [client 192.168.1.1] Premature end of script
headers: test-sleep
[Fri Jul 15 19:48:49 2005] [info] hws : cgi <-- (exec=C:/Program Files/Hitachi/
httpsd/cgi-bin/test-sleep)(argv0=test-sleep)(1800)(2276)
```

CGI プログラム　test-sleep の処理でタイムアウトが発生しているので，この関数を調査してください。

4.2.7 リクエストトレースの採取

リクエストトレースとは，次のときに採取されるトレースのことです。

- リクエスト処理開始時

- リクエスト処理完了時
- KeepAlive 接続の場合，次のリクエストラインの受信完了時
- リクエスト処理開始からリクエストライン受信完了前のコネクション切断時

HWSRequestLog ディレクティブが指定されていた場合，かつ HWSRequestLogType ディレクティブで request が指定された場合に有効となります。障害発生時に Web サーバにリクエストが届いているかどうかを確認する場合などに有用です。

(1) トレースフォーマット

リクエストトレースの出力項目は次のとおりです。

なお，以降の記述で「サーバプロセス ID」は Windows 版の場合「サーバスレッド ID」です。

● リクエスト処理開始時

```
[時刻]△client△:△hws△-->△(クライアントIPアドレス:ポート番号,サーバIPアドレス:ポート番号[A])(サーバプロセスID)
```

● リクエスト処理完了時

```
[時刻]△client△:△hws△<--△(クライアントIPアドレス:ポート番号,サーバIPアドレス:ポート番号[R])(サーバプロセスID)
```

● KeepAlive 接続での次のリクエストライン受信完了時

```
[時刻]△client△:△hws△-->△(クライアントIPアドレス:ポート番号,サーバIPアドレス:ポート番号[K])(サーバプロセスID)
```

● リクエスト処理開始からリクエストライン受信完了前のコネクション切断時

```
[時刻]△client△:△hws△<--△(クライアントIPアドレス:ポート番号,サーバIPアドレス:ポート番号[X])(サーバプロセスID)
```

(凡例)

△：空白

(出力例)

```
[Tue Nov 21 15:18:40 2006] client : hws -->
(192.168.2.1:5245,192.168.1.1:80[A])(1716)
[Tue Nov 21 15:18:41 2006] client : hws <--
(192.168.2.1:5245,192.168.1.1:80[R])(1716)
```

4.2.8 I/O フィルタトレースの採取

I/O **フィルタトレース**とは，モジュールが実装している入出力フィルタ関数の実行時に採取されるトレースのことです。

HWSRequestLog ディレクティブが指定されていた場合，かつ HWSRequestLogType ディレクティブで filter が指定された場合に有効となります。モジュール内のフィルタで発生した障害を切り分ける場合などに有用です。ただし，出力量が多いため，デバッグ目的以外では使用しないでください。

(1) トレースフォーマット

I/O フィルタトレースの出力項目は次のとおりです。

なお，以降の記述で「サーバプロセス ID」は Windows 版の場合「サーバスレッド ID」です。

● 入力フィルタコール時

```
[時刻]△hws△:△in-filter※△-->△(フィルタ名称[フィルタタイプ番号])(サーバプロセスID)
```

● 入力フィルタリターン時

```
[時刻]△hws△:△in-filter※△<--△(フィルタ名称[フィルタタイプ番号])(サーバプロセスID)(戻り値)
```

(凡例)

△：空白

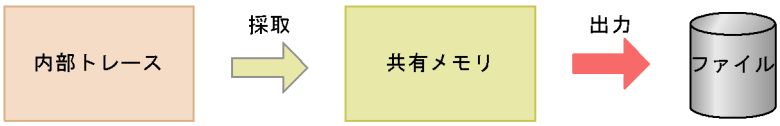
注※　出力フィルタの場合は，「out-filter」になります。

(出力例)

```
[Tue Nov 21 15:18:40 2006] hws : in-filter --> (core_in[60])(1716)
[Tue Nov 21 15:18:40 2006] hws : in-filter <-- (core_in[60])(1716)(0)
```

4.2.9 内部トレースの採取（hwstraceinfo ユティリティ）

アプリケーションプログラムの実行時やリクエスト受け取り時など，システムで発生した事象は内部トレースとして採取されています。内部トレースは，共有メモリにいったん出力され，その後ディレクティブの指定やユティリティによって，ファイルに出力されます。



(1) トレース情報の採取

Web サーバの各種事象発生を契機に内部トレースが共有メモリに採取されます。共有メモリのメモリ識別子は，HWSTraceIdFile ディレクティブに指定したファイルに格納されます。

(2) ファイルへの出力方法

共有メモリに採取された内部トレースは，サーバプロセスの異常終了時または hwstraceinfo ユティリティの実行によって，ファイルに出力されます。サーバプロセスが異常終了した場合は，HWSTraceLogFile ディレクティブで指定したファイルに出力されます。

hwstraceinfo ユティリティでは，共有メモリのメモリ識別子，出力先のファイル名を指定します。hwstraceinfo ユティリティは，UNIX 版の場合，User ディレクティブで指定したユーザまたはスーパユーザだけが実行できます。また，Windows 版の場合，管理者権限を持つユーザだけが実行できます。

内部トレースの出力ファイルサイズは次のとおりです。

UNIX 版の場合

ps -efl コマンドの出力サイズ＋ vmstat コマンドの出力サイズ＋ ipcs -a コマンドの出力サイズ＋ 7KB × MaxClient 値

Windows 版の場合

7KB × ThreadPerChild 値

(3) hwstraceinfo ユティリティ

hwstraceinfo ユティリティの指定方法を説明します。

(a) 形式

```
hwstraceinfo -i 共有メモリ識別子 {-l ファイル名|-r}
```

(b) オペランド

● -i 共有メモリ識別子

HWSTraceIdFile ディレクティブで指定したファイルに出力されている共有メモリ識別子を指定します。

● -l ファイル名

-i で指定した共有メモリ識別子に該当するトレースを出力するファイルを指定します。

● -r

-i で指定した共有メモリ識別子に割り当てられている共有メモリを解放します。UNIX 版では，Web サーバが終了してもトレース用の共有メモリは残ります。残った共有メモリを解放するためにこのオペランドを使用します。Windows 版では，Web サーバ終了時にトレース用の共有メモリは解放されますので，このオペランドは提供していません。

(c) 使用例

共有メモリ識別子 1800_1133780652_0 に該当するトレースを traceinfo.log ファイルに出力する例を示します。

```
hwstraceinfo -i 1800_1133780652_0 -l traceinfo.log
```

(4) 共有メモリの解放および再起動時の注意（UNIX 版の場合）

Hitachi Web Server が終了しても，トレース情報を残すために共有メモリは解放しません。また，サーバを再起動する場合は，共有メモリが再利用されます。

サーバを停止した後に起動した場合は，HWSTraceIdFile ディレクティブに指定したファイルの値を基に，いったん共有メモリを解放して，再度確保します。ただし，次のような場合は，以前使用していた共有メモリが解放できなくなりますので，注意してください。

- 同一ユーザで再起動していない（User ディレクティブまたは Group ディレクティブの値が変更されている）
- HWSTraceIdFile ディレクティブの値を変更している
- HWSTraceIdFile ディレクティブで指定していたファイルが消去されている

共有メモリを解放する場合は，-r を指定した hwstraceinfo ユティリティを実行してください。

4.2.10 保守情報収集機能（hwscollect ユティリティ）

Web サーバが異常終了および無応答となった場合などに，保守員が障害調査を実施するためのコアダンプ，エラーログ，アクセスログなどの資料が必要となります。hwscollect ユティリティによって，これら障害調査のための資料を一括して収集できます。hwscollect ユティリティは UNIX 版だけで有効です。

hwscollect ユティリティは，root 権限で実行する必要があります。

(1) 形式

```
hwscollect 収集情報出力先ディレクトリ 〔-f 定義ファイル名〕
```

(2) オペランド

● 収集情報出力先ディレクトリ

収集した情報を tar のアーカイブファイルとして出力する場合の，出力先のディレクトリを指定します。

アーカイブファイルの名称は，HWSyyyymmddhhmmss.tar となります。ここで yyyymmdd は hwscollect を起動した日付，hhmmss は hwscollect を起動した 24 時間制の時刻で，それぞれローカルタイムです。

● -f 定義ファイル名

hwscollect.conf ファイルを指定します。絶対パスまたはカレントディレクトリからの相対パスで指定します。

(3) 使用方法

HWS を標準的な構成でインストールした場合の使用方法を示します。

```
/opt/hitachi/httpsd/maintenance/hwscollect /tmp
```

(4) コンフィグファイルの設定

hwscollect.conf で hwscollect の動作を定義します。hwscollect.conf は，キーワードと値をスペースで区切って記述します。キーワードは，大文字小文字を区別しません。行の最初に # を付けるとコメント行になります。ファイル名はすべて絶対パスで指定してください。コンフィグファイルのキーワードと指定について次に示します。

表 4-5 コンフィグファイルのキーワードと指定

キーワード	指定する値	指定	複数指定	ワイルドカード
ServerRoot	httpsd.conf の ServerRoot ディレクティブの値を指定します。	必須	×	×
conf	httpsd.conf のファイル名を指定します。	必須	×	×
trcinfo	hwstraceinfo コマンドの存在するディレクトリを指定します。	必須	×	×
trcid	httpsd.conf の HWSTraceIdFile ディレクティブで指定されるファイル名を指定します。	必須	×	×

キーワード	指定する値	指定	複数指定	ワイルドカード
PidFile	httpsd.conf の PidFile ディレクティブで指定されるファイル名を指定します。	必須	×	×
CORE	httpsd.conf の CoreDumpDirectory ディレクティブの値と，SSLCacheServerRunDir ディレクティブの値を指定します。	任意	○	○
LOG	httpsd.conf の ErrorLog ディレクティブ，HWSRequestLog ディレクティブおよび TransferLog ディレクティブや CustomLog ディレクティブなどのログを指定するファイル名を指定します。	任意	○	○
FILES	そのほか，障害解析に役立つファイルがあれば指定します。	任意	○	○

（凡例）

○：指定できる。

×：指定できない。

(5) コンフィグファイルの指定例

コンフィグファイルの指定例を示します。

```
ServerRoot /opt/hitachi/httpsd
conf /opt/hitachi/httpsd/conf/httpsd.conf
trcinfo /opt/hitachi/httpsd/sbin/
trcid /opt/hitachi/httpsd/logs/hws.trcid
PidFile /opt/hitachi/httpsd/logs/httpd.pid
CORE /opt/hitachi/httpsd/logs/core*
LOG /opt/hitachi/httpsd/logs/error*
LOG /opt/hitachi/httpsd/logs/access*
LOG /opt/hitachi/httpsd/logs/hws.trclog*
LOG /opt/hitachi/httpsd/logs/hwsrequest*
```

(6) ディスク使用量

- 一時的に使用するファイル
  200KB+7KB × MaxClients 値
- 収集した情報を出力する tar ファイル
  core ファイルの容量＋ log ファイルの容量＋一時的に使用するファイルの容量

(7) 注意事項

- 収集情報出力先ディレクトリには，core ファイルを含む保守情報のアーカイブファイルが作成されるため，空き領域を確保してください。

- 収集情報出力先ディレクトリに出力ファイルおよび一時ファイルを作成します。このため，収集情報出力先ディレクトリは書き込み可能としてください。
- CORE，LOG および FILES にディレクトリを指定すると，指定されたディレクトリ下のファイルのすべてを採取します。このため，ルートディレクトリなど上位ディレクトリを指定すると，大量かつ不要な情報を採取してしまうため，注意が必要です。

4.3 サーバマシンのバーチャル化（バーチャルホスト）

バーチャルホストは 1 台のサーバマシンを複数台のマシンに見せます。その方法は次に示す二つがあります。

- サーバ名に基づくバーチャルホスト（Name-Based Virtual Hosts）
- IP アドレスに基づくバーチャルホスト（IP-Based Virtual Hosts）

(1) サーバ名に基づくバーチャルホスト

サーバ名に基づくバーチャルホストは，一つの IP アドレスに対して複数のホスト名を DNS サーバなどで定義しておき，クライアントからそのホスト名でアクセスすることで，複数ホストのように見せます。ネットワークインタフェースを複数設定する必要はありません。サーバ名に基づくバーチャルホストでは異なる複数の SSL 対応ホストは構築できません。異なる複数の SSL 対応ホストを構築する場合は，IP アドレスに基づくバーチャルホストで構築してください。

（例）1 台のサーバマシン（IP アドレス：172.17.40.10）上の一つの Web サーバでポートを一つオープンし，Web ブラウザからのリクエストに応じてホストを切り替える運用をする。

Web ブラウザからの要求が，http://www1.xxx.soft.hitachi.co.jp/ の場合
C:/Program Files/Hitachi/httpsd/htdocs1/index.html（DirectoryIndex の指定が index.html の場合）を参照します。

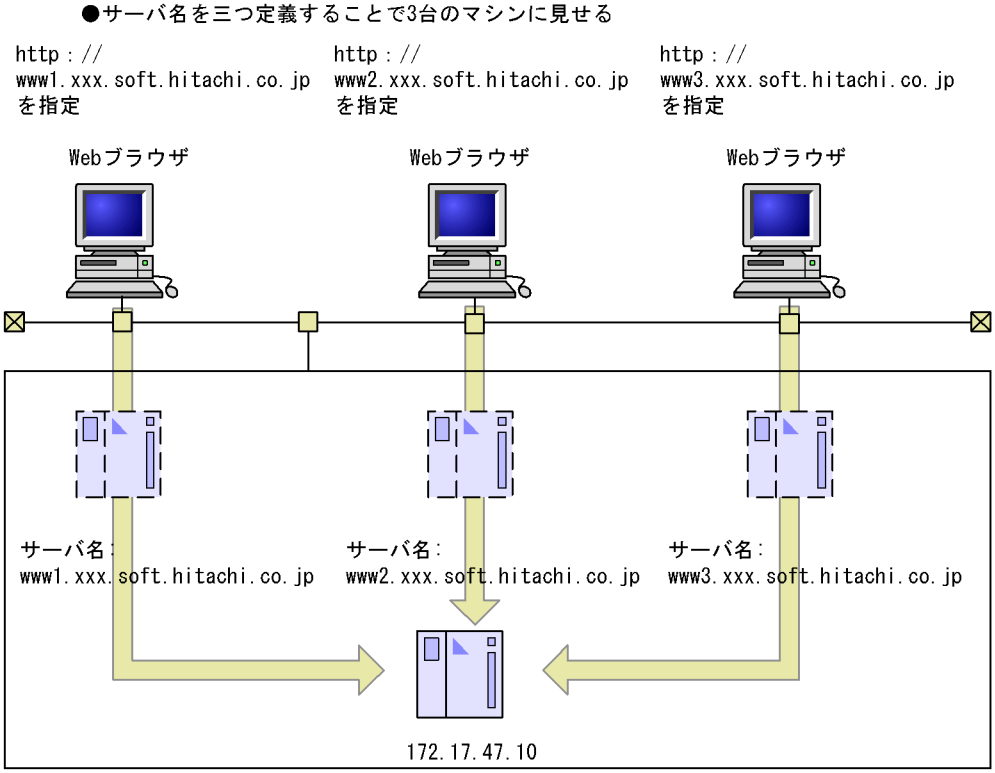
Web ブラウザからの要求が，http://www3.xxx.soft.hitachi.co.jp/ の場合
C:/Program Files/Hitachi/httpsd/htdocs3/index.html（DirectoryIndex の指定が index.html の場合）を参照します。

ただし，この方法は Web ブラウザからのリクエスト中の Host ヘッダで，"Host: www1.xxx.soft.hitachi.co.jp" のようにホスト名（必要に応じてポート番号）を指定してきた場合だけ利用できます。古い Web ブラウザや，簡易タイプの Web ブラウザでは利用できないことがあるので注意が必要です。その場合，最も上位に記述された <VirtualHost> ブロックの指定（この例では www1.xxx.soft.hitachi.co.jp）が有効になります。

```
Port 80                                                    …1.
NameVirtualHost 172.17.40.10                               …2.
<VirtualHost 172.17.40.10>                                 …3.
DocumentRoot "C:/Program Files/Hitachi/httpsd/htdocs1"     …4.
ServerName www1.xxx.soft.hitachi.co.jp                     …5.
</VirtualHost>
<VirtualHost 172.17.40.10>                                 …6.
DocumentRoot "C:/Program Files/Hitachi/httpsd/htdocs2"     …7.
ServerName www2.xxx.soft.hitachi.co.jp                     …8.
</VirtualHost>
<VirtualHost 172.17.40.10>                                 …9.
DocumentRoot "C:/Program Files/Hitachi/httpsd/htdocs3"     …10.
ServerName www3.xxx.soft.hitachi.co.jp                     …11.
</VirtualHost>
```

1. ポート番号は一つ
2. サーバ名に基づくバーチャルホストの IP アドレス
3. バーチャルホスト 1 の定義
4. ルートディレクトリの定義
5. サーバ名 1 の定義
6. バーチャルホスト 2 の定義
7. ルートディレクトリの定義
8. サーバ名 2 の定義
9. バーチャルホスト 3 の定義
10. ルートディレクトリの定義
11. サーバ名 3 の定義

注 www1.xxx.soft.hitachi.co.jp，www2.xxx.soft.hitachi.co.jp，www3.xxx.soft.hitachi.co.jp は，DNS サーバなどに 172.17.40.10 ホストのホスト名として登録されていなければなりません。



(2) IP アドレスに基づくバーチャルホスト

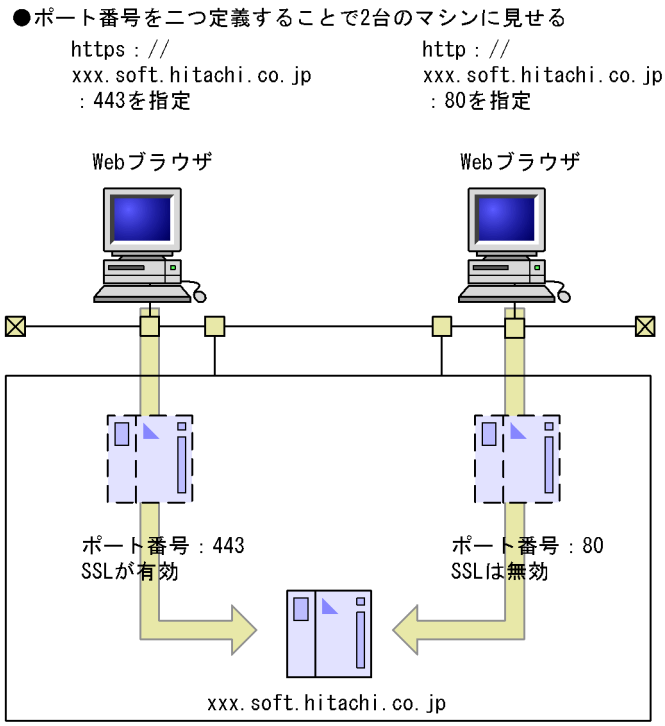
IP アドレスに基づくバーチャルホストは次の三つの方法でクライアントには複数ホストのように見せます。

- 複数のポートを使用
- 1 台のサーバマシンに複数のネットワークインタフェースを指定
- IP アドレスのエイリアスを指定

(例 1) 1 台のサーバマシン上の一つの Web サーバでポートを二つオープンし，SSL 対応 Web サーバと非対応 Web サーバの二つのホストとして運用する。

```
Listen 443                                                              …1
Listen 80                                                               …2
SSLDisable                                                              …3
<VirtualHost xxx.soft.hitachi.co.jp:443>                                …4
    DocumentRoot "C:/Program Files/Hitachi/httpsd/ssldocs"
   SSLEnable                                                            …5
    SSLCertificateFile "C:/Program Files/Hitachi/httpsd/conf/ssl/server/
httpsd.pem"
    SSLCertificateKeyFile "C:/Program Files/Hitachi/httpsd/conf/ssl/server/
httpsdkey.pem"
</VirtualHost>
<VirtualHost xxx.soft.hitachi.co.jp:80>                                 …6
    DocumentRoot "C:/Program Files/Hitachi/httpsd/htdocs"
   SSLDisable                                                           …7
</VirtualHost>
```

1. ポート番号の定義
2. ポート番号の定義
3. メインのサーバは SSL を無効に設定
4. ポート番号 443 のバーチャルホストの定義
5. SSL 有効
6. ポート番号 80 のバーチャルホストの定義
7. SSL 無効



（例 2） 1 台のサーバマシン上に二つの NIC（Network Interface Card）（IP アドレス

:172.17.40.10，172.17.40.20）を備え，一つの Web サーバで Web ブラウザからのリクエストに応じてホストを切り替えて運用する。

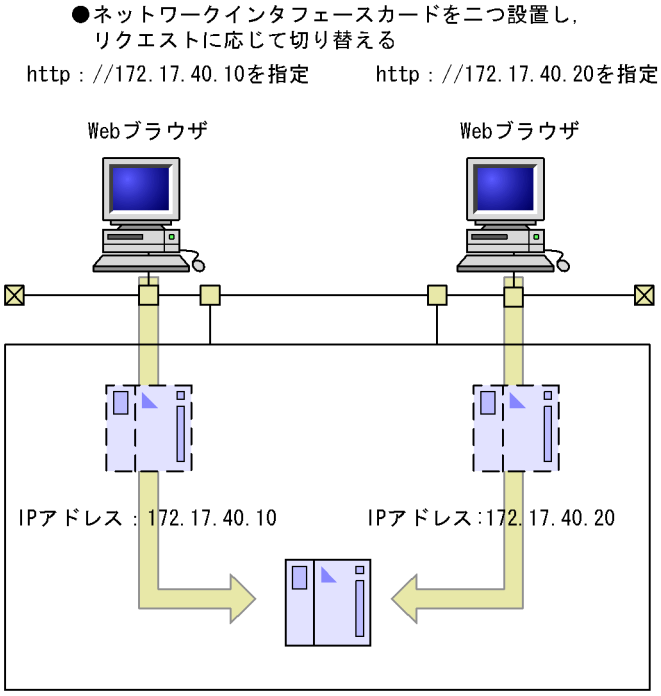
Web ブラウザからのリクエストが，http://172.17.40.10/ の場合

C:/Program Files/Hitachi/httpsd/htdocs1/index.html（DirectoryIndex の指定が index.html の場合）を参照します。

Web ブラウザからのリクエストが，http://172.17.40.20/ の場合

C:/Program Files/Hitachi/httpsd/htdocs2/index.html（DirectoryIndex の指定が index.html の場合）を参照します。

```
Port 80
<VirtualHost 172.17.40.10>
DocumentRoot "C:/Program Files/Hitachi/httpsd/htdocs1"
ServerName www10.xxx.soft.hitachi.co.jp
</VirtualHost>
<VirtualHost 172.17.40.20>
DocumentRoot "C:/Program Files/Hitachi/httpsd/htdocs2"
ServerName www20.xxx.soft.hitachi.co.jp
</VirtualHost>
```



4.4 Web サーバでの CGI プログラムの実行

CGI プログラムとは，Web サーバ上で動作するプログラムです。この CGI プログラムを使用すれば，静的な HTML へのアクセスだけでは実現できないインタラクティブな Web アクセスができます。

(1) CGI プログラムの定義

CGI プログラムを実行するには，ScriptAlias ディレクティブで CGI プログラムがあるディレクトリを指定する方式，AddHandler ディレクティブを使用しファイル拡張子に cgi-script ハンドラを指定する方式および SetHandler ディレクティブで cgi-script ハンドラを指定する方式があります。

httpsd.conf で設定する場合は，CGI プログラムの管理のしやすさの点で，ScriptAlias ディレクティブによる設定を推奨します。

(a) ScriptAlias ディレクティブの指定例

CGI プログラムのパス名を C:/Program Files/Hitachi/httpsd/cgi-bin/CGI プログラムファイル名とし，これに対してクライアントから /cgi-bin/CGI プログラムファイル名でアクセスする場合

```
ScriptAlias /cgi-bin/ "C:/Program Files/Hitachi/httpsd/cgi-bin/"
```

(b) AddHandler ディレクティブの指定例

● **ファイル拡張子** .cgi **に** cgi-script **ハンドラを指定する場合**

```
AddHandler cgi-script .cgi
```

なお，Options ディレクティブで ExecCGI オプションの設定が必要です。

(c) SetHandler ディレクティブの指定例

● script **で始まるファイル名に対するリクエストに対して，**cgi-script **ハンドラを指定する場合**

```
<FilesMatch ^script>
  SetHandler cgi-script
  Options ExecCGI
</FilesMatch>
```

(2) CGI プログラムの呼び出し

CGI プログラムは Web ブラウザから次の形式の URL を指定して呼び出します。

```
http://ホスト名〔:ポート番号〕/パス名〔?問い合わせ文字列〕
```

● ホスト名〔: ポート番号〕

Web サーバが起動しているホスト名または IP アドレスと，ポート番号を指定します。ポート番号を省略すると，ポート番号 80 にリクエストを送信します。

● パス名

パス名は CGI プログラムのパスを指定します。

● 問い合わせ文字列

CGI プログラムに渡すパラメタです。そのキーワードと値の組を指定します。Web ブラウザのフォームにデータを記述した場合，リクエストラインに自動的に設定されます。

(3) CGI プログラムに渡す情報

Web サーバから CGI プログラムに環境変数を渡します。詳細は「付録 B　CGI プログラムに渡す環境変数」を参照してください。

(4) CGI プログラムの例

CGI プログラムのサンプルプログラムと，その実行例を説明します。

サンプル CGI プログラム

Windows 版で使用可能なサンプルプログラムのソース例を次に示します。これは Perl 言語で書かれたプログラムで，ファイル名を test-cgi.pl とします。

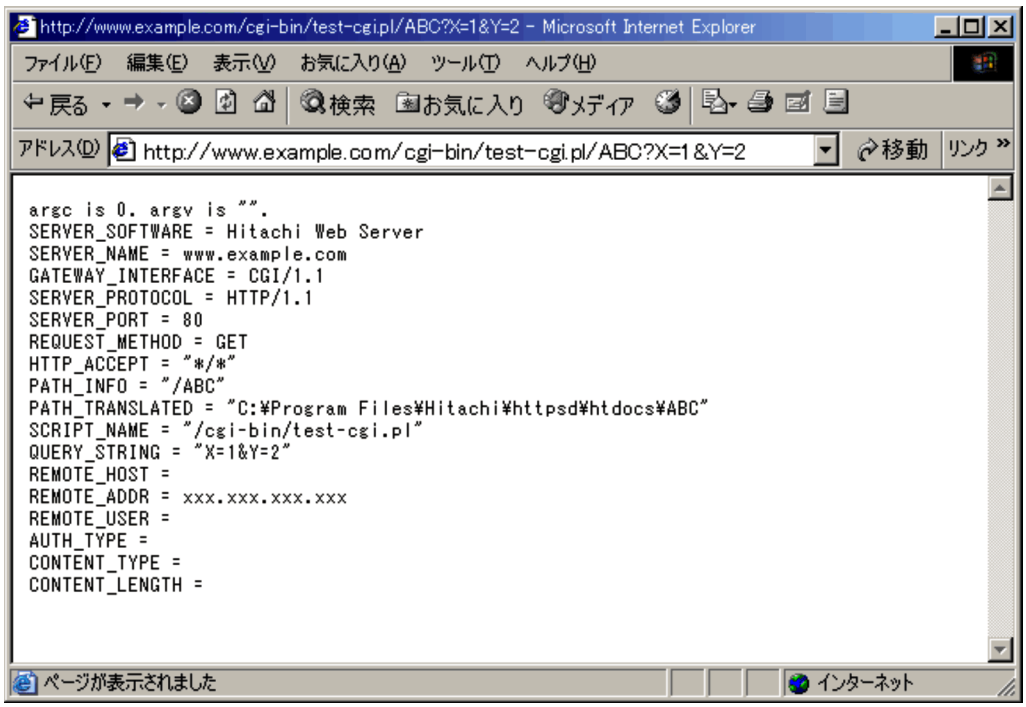
```
#! c:¥bin¥perl.exe

$argc=$#ARGV+1;
print "Content-Type: text/plain¥n";
print "¥n";
print "argc is $argc. argv is ¥"@ARGV¥".¥n";
print "SERVER_SOFTWARE = $ENV{'SERVER_SOFTWARE'}¥n";
print "SERVER_NAME = $ENV{'SERVER_NAME'}¥n";
print "GATEWAY_INTERFACE = $ENV{'GATEWAY_INTERFACE'}¥n";
print "SERVER_PROTOCOL = $ENV{'SERVER_PROTOCOL'}¥n";
print "SERVER_PORT = $ENV{'SERVER_PORT'}¥n";
print "REQUEST_METHOD = $ENV{'REQUEST_METHOD'}¥n";
print "HTTP_ACCEPT = ¥"$ENV{'HTTP_ACCEPT'}¥"¥n";
print "PATH_INFO = ¥"$ENV{'PATH_INFO'}¥"¥n";
print "PATH_TRANSLATED = ¥"$ENV{'PATH_TRANSLATED'}¥"¥n";
print "SCRIPT_NAME = ¥"$ENV{'SCRIPT_NAME'}¥"¥n";
print "QUERY_STRING = ¥"$ENV{'QUERY_STRING'}¥"¥n";
print "REMOTE_HOST = $ENV{'REMOTE_HOST'}¥n";
print "REMOTE_ADDR = $ENV{'REMOTE_ADDR'}¥n";
print "REMOTE_USER = $ENV{'REMOTE_USER'}¥n";
print "AUTH_TYPE = $ENV{'AUTH_TYPE'}¥n";
print "CONTENT_TYPE = $ENV{'CONTENT_TYPE'}¥n";
print "CONTENT_LENGTH = $ENV{'CONTENT_LENGTH'}¥n";
```

CGI プログラムの実行

Web ブラウザに次に示すように指定して，サンプル CGI プログラムを呼び出します。

```
http://www.example.com/cgi-bin/test-cgi.pl/ABC?X=1&Y=2
```

サンプルプログラムの実行結果



(5) CGI プログラムに渡す追加情報

CGI/1.1 の環境変数以外に Web サーバから CGI プログラムに情報を渡す場合の指定方法について説明します。

コンフィグファイルに CGI プログラムに渡す環境変数や，その値を指定できます。CGI プログラムに渡さない環境変数の指定もできます。

```
PassEnv 環境変数                    CGIプログラムに渡す環境変数の指定
SetEnv 環境変数 値                  CGIプログラムに渡す環境変数とその値の指定
UnsetEnv 環境変数                   CGIプログラムに渡さない環境変数の指定
```

(6) 環境変数の定義

クライアントのリクエストを基に，環境変数を定義できます。リクエストしているクライアントのホスト名や IP アドレスなどを基に環境変数を定義したり，環境変数の設定を解除したりできます。

```
SetEnvIfNoCase Request_URI "¥.(gif)|(jpg)$" request_is_image
```

この場合，ファイル拡張子が .gif または .jpg のとき（このディレクティブの場合，大文字，小文字の区別はしません），request_is_image という環境変数を CGI プログラムに渡します。

（7） Windows で CGI プログラムを利用するときの注意事項

(a) CGI プログラム作成時の注意

CGI プログラムとサーバスレッド間のデータ送受信には，CGI プログラムの標準入力，標準出力，標準エラー出力を使用しています。データ送受信中には Timeout ディレクティブは有効になります。CGI プログラム作成時には，データの送受信完了後は，標準入出力などを閉じるかまたは終了してください。

(b) CGI プログラムの強制終了

CGI プログラムは Web サーバが停止しても，CGI プログラム自身が処理を終えるまで終了しません。CGI プログラムを強制終了するには「タスクマネージャ」から終了させます。

（8） UNIX 版で CGI プログラムを利用するときの注意事項

CGI プログラムには，User，Group ディレクティブ指定値での実行権限が必要です。

（9） パス情報指定時の注意事項

リクエスト URL に，CGI プログラムに渡すパス情報が指定された場合，そのパス情報を環境変数 PATH_INFO に，パス情報をファイルシステム上のパスに変換した値を環境変数 PATH_TRANSLATED に設定します。パス情報をファイルシステム上のパスに変換する際には，DocumentRoot ディレクティブに指定されたパスを基点とします。パス情報に対し，Alias ディレクティブなどで別名を指定している場合は，その指定に従って変換します。

Web サーバの設定によって，環境変数 PATH_TRANSLATED に設定されたパスへのアクセスを許可していない場合には，エラーログにアクセス拒否のメッセージを出力します。このメッセージを出力した場合でも，Web サーバは CGI プログラムを実行し，リクエスト処理を続行します。この際，環境変数 PATH_TRANSLATED も CGI プログラムに渡されます。

（例）ドキュメントルートが "C:/Program Files/Hitachi/httpsd"，Web ブラウザからの要求が "http://www.example.com/cgi-bin/test-cgi.pl/ABC"（CGI プログラム "test-cgi.pl" を実行するリクエストに，パス情報として "/ABC" を付加）の場合の，エラーログ出力例を次に示します。

```
[Fri Feb 20 12:00:00 2004] [error] [client 192.168.1.1] client denied by server
configuration: C:/Program Files/Hitachi/httpsd/ABC
```

4.5 ユーザ認証とアクセス制御

Web サーバに対するアクセス制御方法には次に示す方法があります。

- ユーザ名およびパスワードによるアクセス制御
- クライアントのホスト名または IP アドレスによるアクセス制御
- ディレクトリに対するアクセス制御
- ディレクトリサービスを利用したアクセス制御

4.5.1 ユーザ名およびパスワードによるアクセス制御

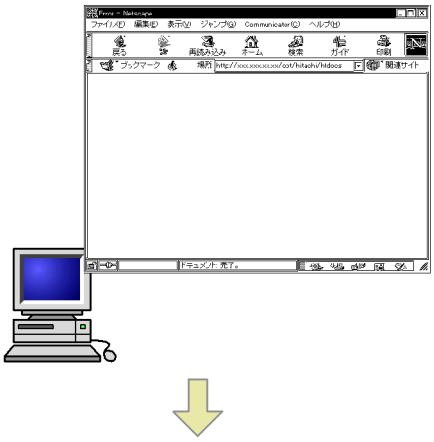
ユーザ名とそのパスワードは htpasswd ユティリティを使用して，パスワードファイルに登録します。登録されているユーザ名に対して，ホスト内のディレクトリやファイルなどのアクセス権を定義できます。htpasswd ユティリティの使用方法については，「(1)　ユーザ名とパスワードのパスワードファイルへの登録およびパスワードの変更」を参照してください。

（例）C:¥Program Files¥Hitachi¥httpsd¥htdocs¥ ディレクトリ下を特定のユーザだけに公開する

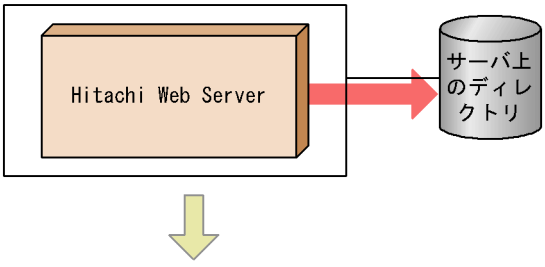
htpasswd ユティリティを使用してあらかじめユーザ名とパスワードをパスワードファイル（C:¥Program Files¥Hitachi¥httpsd¥htdocs¥.htpasswd）に登録しておいてください。httpsd.conf ファイルに次に示すディレクティブを設定します。ユーザが C:¥Program Files¥Hitachi¥httpsd¥htdocs¥ にアクセスすると Web サーバはステータスコード 401　Authorization Required を応答し，Web ブラウザでユーザ名およびパスワードの入力を要求します。

```
<Directory "C:/Program Files/Hitachi/httpsd/htdocs">
    AuthType Basic
    AuthName "realm 1"
    AuthUserFile "C:/Program Files/Hitachi/httpsd/htdocs/.htpasswd"
    Require valid-user
</Directory>
```

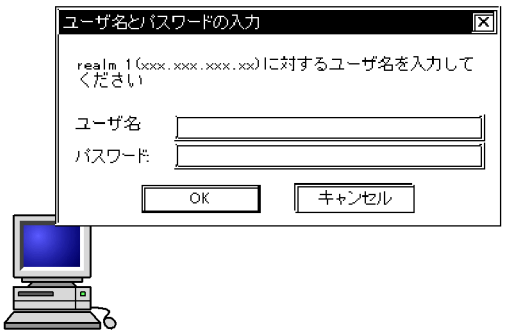
1. Webブラウザからリクエストの送信



2. Hitachi Web Serverがリクエストを受け付け，リクエストがあったディレクトリのアクセスを制御する。



3. Webブラウザにユーザ IDとパスワードの入力を促すメッセージを出力する。



(1) ユーザ名とパスワードのパスワードファイルへの登録およびパスワードの変更

htpasswd ユティリティを使用して，パスワードファイルにユーザ名，パスワードの登録および変更ができます。

htpasswd ユティリティの使用方法について次に説明します。

(a) 形式

```
htpasswd 〔-b〕〔-c | -D〕 パスワードファイル名 ユーザ名〔パスワード〕
```

(b) オペランド

● -b

パスワードをコマンドラインに指定する場合に指定します。

● -c

新規にパスワードファイルを作成する場合に指定します。すでに作成しているパスワードファイルにユーザを追加する場合や，パスワードを変更する場合には，指定する必要はありません。

● -D

ユーザの登録を削除する場合に指定します。指定したパスワードファイルに，指定したユーザが登録されている場合に，パスワードファイルから該当するユーザを削除します。

● パスワードファイル名

パスワードを登録，変更または削除するパスワードファイルを指定します。

● ユーザ名

パスワードを登録，変更または削除するユーザ名を指定します。

● パスワード

登録または変更するパスワードを指定します。-b オプションを指定したときだけ指定できます。

(c) 使用方法

パスワードファイル名と，登録するユーザ名またはパスワードを変更するユーザ名を指定して htpasswd を起動すると，そのユーザのパスワードの入力が要求されます。入力確認を含め，2 回パスワードを入力すると，パスワードファイルにそのユーザのユーザ名と，パスワードが登録されます。

```
C:¥>"Program Files¥Hitachi¥httpsd¥bin¥htpasswd.exe" .passwd userxx     …1.
New password:                                                          …2.
Re-type new password:                                                  …3.
Updating password for userxxx                                          …4.
C:¥>
```

1. userxx のパスワードの変更
2. 新パスワード入力
3. 新パスワード再入力
4. 新パスワードの登録終了

登録を削除する場合は，-D オプション，パスワードファイル名および削除するユーザ名を指定して htpasswd を起動します。

```
C:¥>"Program Files¥Hitachi¥httpsd¥bin¥htpasswd.exe" -D .passwd userxx  …1.
Deleting passwd for userxx                                             …2.
C:¥>
```

1. userxx の登録削除
2. userxx の登録削除終了

(d) 注意事項

- Windows 版のパスワードの最大長は 128 文字です。ユーザ名の最大長は 128 文字です。UNIX 版のパスワードの最大長は，パスワード読み取り関数であるシステムコール getpass() の最大長と 128 文字のどちらか短い方です。getpass() についての詳細は，ご使用の OS のマニュアルを参照してください。
- htpasswd ユティリティ実行時は，パスワードファイルの作成先と同じディレクトリに，作業ファイルが一時的に作成されます。作業ファイル名は，「パスワードファイル名 . プロセス ID」です。この作業ファイルは，htpasswd ユティリティの終了時に自動的に削除されます。ただし，実行中にキャンセルした場合など，作業ファイルが削除されないことがあります。作業ファイルが残っている場合は，手動で削除してください。

4.5.2 クライアントのホスト名または IP アドレスによるアクセス制御

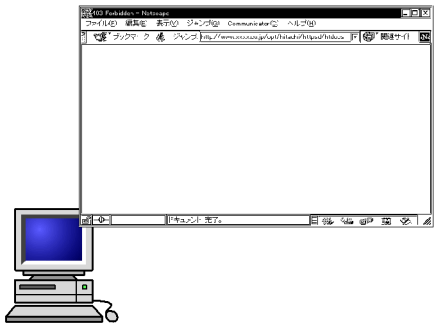
クライアントのホスト名や IP アドレスによってアクセス制御するには，Allow from ディレクティブや Deny from ディレクティブを使用します。Allow from ディレクティブでアクセスを許可するホスト，Deny from ディレクティブでアクセスを禁止するホストを指定します。

(例) C:¥Program Files¥Hitachi¥httpsd¥htdocs¥ ディレクトリ下を，プロキシ経由のリクエストによる参照を禁止する

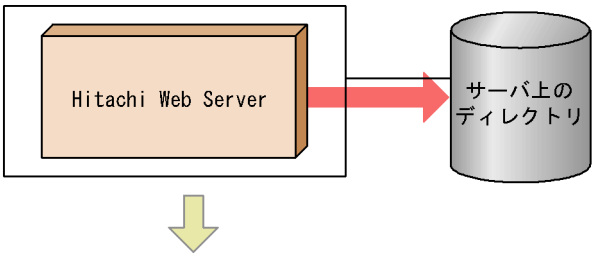
httpsd.conf ファイルに次に示すディレクティブを設定します。ユーザが C:¥Program Files¥Hitachi¥httpsd¥htdocs¥ にアクセスする場合，proxy.xxx.soft.hitachi.co.jp をプロキシとして利用している Web ブラウザはステータスコード 403 Forbidden でアクセスを拒否されます。

```
<Directory "C:/Program Files/Hitachi/httpsd/htdocs">     ディレクトリの定義
    Order deny,allow                                     アクセス許可と禁止の優先順位定義
    Deny from proxy.xxx.soft.hitachi.co.jp   アクセスの禁止
</Directory>
```

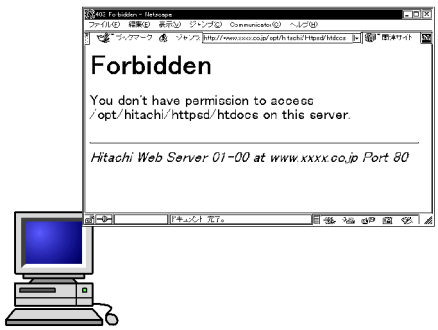
1. Webブラウザからリクエストの送信



2. Hitachi Web Serverがリクエストを受け付け，リクエストがあったディレクトリのアクセスを制御する。



3. proxy.xxx.soft.hitachi.co.jpをプロキシとして利用しているブラウザはステータスコード403 Forbiddenでアクセスを拒否される。



4.5.3 ディレクトリに対するアクセス制御

アクセスコントロールファイル（.htaccess）を特定のディレクトリ下に作成すれば，そのディレクトリに対するアクセス権を設定できます。そのファイルにアクセスを許可または拒否するクライアント名（IP アドレス）やユーザ名を指定します。

(1) アクセスコントロールファイル

アクセスコントロールファイルを特定のディレクトリ下に作成すれば，そのディレクトリに対するアクセス権を設定できます。アクセスコントロールファイルの名称は，AccessFileName ディレクティブで指定します。デフォルトは .htaccess です。

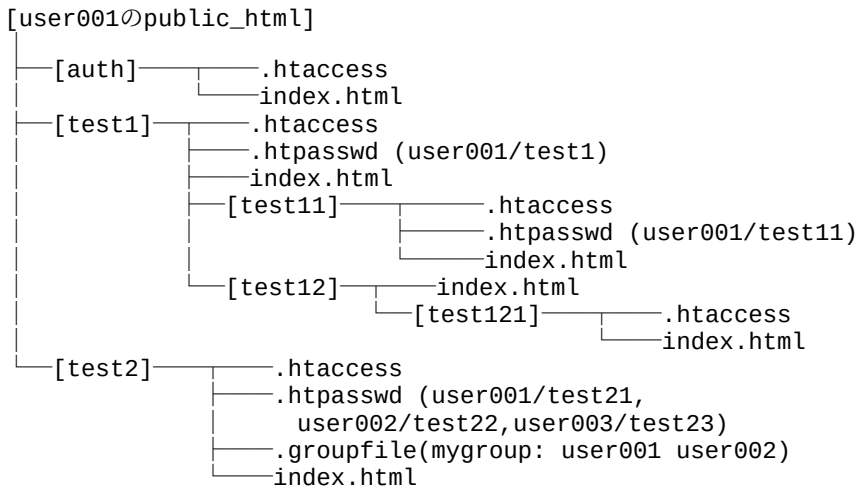
アクセスコントロールファイルによるアクセス制御は，Web サーバを再起動することなく，有効になります。ただし，正しく機能させるためには，httpsd.conf の AllowOverride ディレクティブを適切な上書き許可レベルに設定する必要があります。

アクセスコントロールファイルにパスワードファイルを指定すると，ユーザがそのディレクトリにアクセスする場合にユーザ名およびパスワードの入力を要求します。

注 アクセスコントロールファイル（.htaccess）とパスワードファイル（.htpasswd）は 1 対 1 である必要はありません。異なるアクセスコントロールファイルの AuthUserFile ディレクティブに同じパスワードファイルを指定できます。

(2) アクセス権の設定例

次のようなディレクトリ構成で，各ディレクトリに対してアクセスコントロールファイルにアクセス権を設定する



● auth ディレクトリ下のアクセス権の定義（auth/.htaccess ファイル）

IP アドレスが 172.18.102.11 および 172.16.202.4 のサーバからのアクセスを拒否しま

す。

```
Order deny,allow                        …1.
Deny from 172.18.102.11 172.16.202.4  …2.
```

1. アクセス拒否の定義を先に評価
2. アクセス拒否の定義

● **test1 ディレクトリ下のアクセス権の定義（**test1/.htaccess **ファイル）**

ユーザ名 =user001，パスワード =test1 を入力した場合だけ，test1/index.html および test1/test12/index.html へのアクセスを許可します。

```
AuthUserFile C:/user001/public_html/test1/.htpasswd      …1.
AuthName "test1 Directory"                               …2.
AuthType Basic
<Limit GET POST>                                         …3.
   Require user user001                                  …4.
</Limit>
```

1. パスワードファイルの定義
   パスワードファイルに登録しているユーザ名とパスワード
   ユーザ名：user001，パスワード：test1
2. realm 名の定義
3. メソッドに対する定義
4. ユーザ名：user001 のアクセスを許可

● **test1/test11 ディレクトリ下のアクセス権の定義（**test1/test11/.htaccess **ファイル）**

ユーザ名 =user001，パスワード =test11 を入力した場合だけ，test1/test11/index.html へのアクセスを許可します。

```
AuthUserFile C:/user001/public_html/test1/test11/.htpasswd      …1.
AuthName "test11 Directory"                                     …2.
AuthType Basic
<Limit GET POST>                                                …3.
   Require user user001                                         …4.
</Limit>
```

1. パスワードファイルの定義
   パスワードファイルに登録しているユーザ名とパスワード
   ユーザ名：user001，パスワード：test11
2. realm 名の定義
3. メソッドに対する定義
4. ユーザ名：user001 のアクセスを許可

● **test1/test12/test121 ディレクトリ下のアクセス権の定義（**test1/test12/test121/

.htaccess ファイル）

ユーザ名 =user001，パスワード =test1 を入力し，Web ブラウザが MSIE の場合だけ，test1/test12/test121/index.html へのアクセスを許可します。

```
Order deny,allow      …1.
Allow from env=MSIE   …2.
Deny from all         …3.
```

1. アクセス拒否の定義を先に評価
2. Web ブラウザが MSIE の場合，アクセスを許可
3. すべてのホストからのアクセスを拒否

ただし，httpsd.conf に次のディレクティブを定義しているものとします。

```
SetEnvIf User-Agent ".*MSIE.*" MSIE
```

● test2 ディレクトリ下のアクセス権の定義（test2/.htaccess ファイル）

mygroup グループのユーザ名，パスワードを入力した場合だけ，test2/index.html へのアクセスを許可します。

```
AuthUserFile C:/user001/public_html/test2/.htpasswd        …1.
AuthGroupFile C:/user001/public_html/test2/.groupfile      …2.
AuthName "test2 Directory"                                 …3.
AuthType Basic
<Limit GET POST>                                           …4.
   Require group mygroup                                   …5.
</Limit>
```

1. パスワードファイルの定義
   パスワードファイルに登録しているユーザ名とパスワード
   ユーザ名：user001，パスワード：test21
   ユーザ名：user002，パスワード：test22
   ユーザ名：user003，パスワード：test23
2. グループファイルの定義
   グループファイルに登録しているグループ名
   グループ名：mygroup
   mygroup に登録しているユーザ名：user001，user002，user003
3. realm 名の定義
4. メソッドに対する定義
5. グループ名：mygroup のアクセスを許可

4.5.4 ディレクトリサービスを利用したユーザ認証とアクセス制御

ディレクトリサービス（以降，LDAP **サーバ**と呼びます）と連携して，パスワードファイルを作成しないでユーザ認証ができます。また，LDAP サーバ内の属性でアクセス制御ができます。

この機能は，AIX，Linux（32 ビット），Solaris および Windows 版で使用できます。

HP-UX（IPF）および Linux（IPF）版では使用できません。

（1） LDAP サーバでユーザ認証するための準備

LDAP サーバを利用した認証例を説明します。

この機能を利用するためには，httpsd.conf に次のディレクティブを設定して LDAP 関連機能を処理するモジュールを組み込んでください。次の 1，2 の順に，二つのモジュールを順番に組み込んでください。

組み込みを指定した行以降に，LDAP を利用したユーザ認証のディレクティブが設定できます。

(a) コンフィグファイル httpsd.conf 上での組み込み方法

1. ライブラリの組み込み
   - Linux（32 ビット）版
     LoadFile libexec/libldapssl41.so
     このライブラリは Web サーバをインストールしたときに標準で格納されています。
   - Solaris 版
     LoadFile libexec/libldapssl41.so
     このライブラリは Web サーバをインストールしたときに標準で格納されています。
   - AIX 版
     ファイルセット ldap.client.rte をインストールした後，次の設定をしてください。
     LoadFile /usr/lib/libldap.a　　（AIX 5L V5.2 以前の場合）
     LoadFile /usr/lib/libibmldap.a　（AIX 5L V5.3 の場合）
   - Windows 版
     LoadFile libldap/nsldap32v50.dll
     このライブラリは Web サーバをインストールしたときに標準で格納されています。
2. LDAP 認証モジュールの組み込み
   - UNIX 版
     LoadModule hws_ldap libexec/mod_hws_ldap.so
   - Windows 版
     LoadModule hws_ldap modules/mod_hws_ldap.so

(2) LDAP サーバでの認証方法

ユーザ認証する場合，<Directory>,.htaccess にはパスワードファイルを使用した場合と同様に，AuthType ディレクティブと AuthName ディレクティブを指定します。また，Require valid-user と LDAPRequire ディレクティブを指定することで，LDAP サーバと連携したユーザ認証ができます。

C:/Program Files/Hitachi/httpsd/cgi-bin/ の CGI を使用する場合に，ユーザ ID とパスワードを入力させて認証する例を次に示します。

(例)

```
LDAPServerName ldap.server.hitachi.com
LDAPServerPort 389
<Directory "C:/Program Files/Hitachi/httpsd/cgi-bin">
AuthName LDAP-TASK
AuthType Basic
Require valid-user
LDAPRequire
</Directory>
```

LDAP サーバ内の属性でアクセス制御もできます。

例えば，社員登録番号が 100 番から 200 番のユーザだけアクセスを許可することもできます。詳細については LDAPRequire ディレクティブを参照してください。

(3) LDAP サーバでのアクセス制御

認証されたユーザが，該当コンテンツを利用できるかどうかを定義できます。

```
LDAPRequire 〔%DN属性%〕 〔LDAP検索フィルタ〕
```

LDAP 検索フィルタに，LDAP サーバに登録されている情報を基に，アクセス権限について定義します。

例えば，認証されたユーザの中で taro と hanako だけをアクセスさせたい場合には，次のように定義します。定義した情報は，あらかじめ LDAP サーバ内に登録してください。

(例)

```
LDAPRequire %cn% (|(cn=taro)(cn=hanako))
```

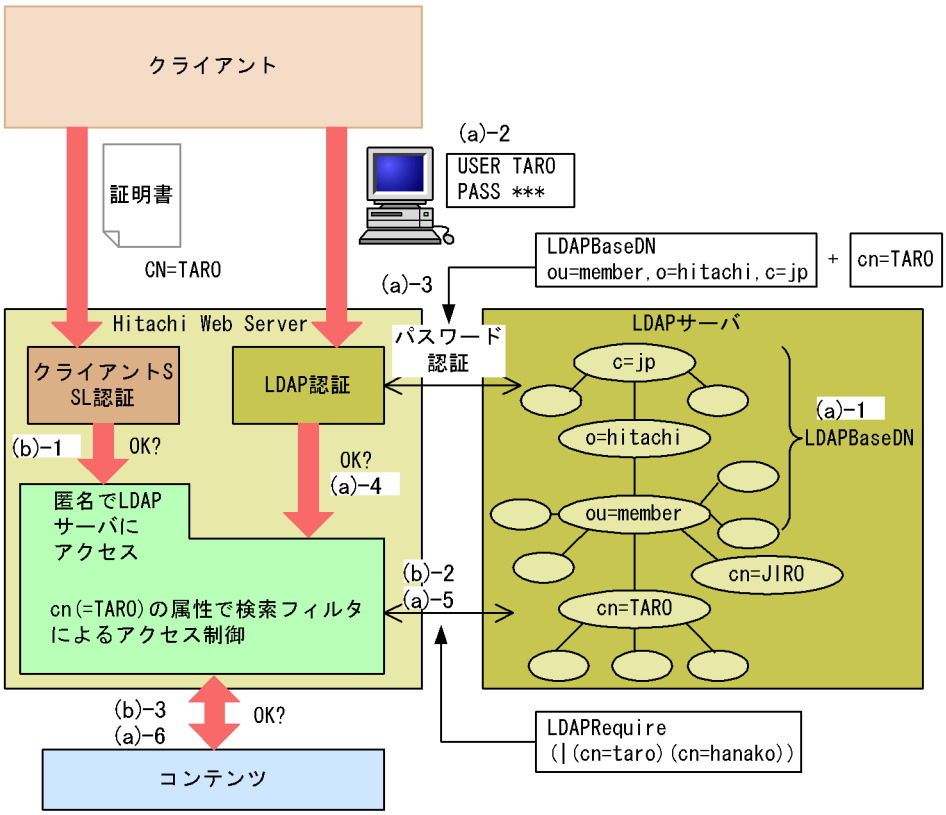
(4) ユーザ認証とアクセス制御の関係

LDAP サーバにユーザ名が cn として登録されている場合を例に説明します。SSL クライ

アント認証と LDAPRequire ディレクティブを組み合わせた場合，証明書による認証後，LDAP サーバにクライアントが登録されているかどうかを確認します。このとき，クライアント証明書内のサブジェクトの Common Name（CN）をユーザ名として扱い，パスワードを使用しない匿名アクセスとして LDAP サーバにアクセスし確認します。LDAP サーバにアクセスできなかった場合は，ステータスコード 500 Internal Server Error を応答します。

ユーザ認証とアクセス制御の関係を次の図に基づいて説明します。

図 4-5　ユーザ認証とアクセス制御の関係



(a) LDAP サーバによる認証

1. LDAP サーバでの認証をするためには，各ユーザが登録されている DN（認証するユーザが登録されているエントリ：ou=member，o=hitachi，c=jp など）を，あらかじめ LDAPBaseDN ディレクティブに定義しておきます。
2. LDAPRequire ディレクティブが定義されたコンテンツをアクセスする場合，クライアントをこの DN 内の情報を使って認証します。Web ブラウザ上にユーザ名とパスワードの入力を要求する画面が表示されます。
3. ユーザ名，パスワードを入力すると，cn= ユーザ名と LDAPBaseDN ディレクティブに定義した DN を組み合わせて，認証するユーザの DN を作成して，パスワード認証

します。この場合，ユーザの DN は cn=TARO,ou=member,o=hitachi,c=jp になります。
4. LDAP サーバの DN に登録されたパスワードと，クライアントが入力したパスワードが一致しなければ，このユーザに対してステータスコード 401 Authorization Required を応答し，アクセスを拒否します。
5. パスワードが一致しても LDAPRequire ディレクティブに LDAP 検索フィルタの指定がある場合は，検索フィルタの記述とユーザの DN が一致するかどうかを判断します。
6. 一致していればアクセスを許可します。cn=JIRO はパスワードが一致していても，検索フィルタの記述には一致しないため，LDAPNoEntryStatus ディレクティブに従ったステータスコード (デフォルトでは 401 Authorization Required) を応答し，アクセスを拒否します。

(b) SSL クライアント証明書がある場合

1. クライアント（Web ブラウザ）からアクセスするときに，SSL クライアント証明書を受け付けた場合には，LDAP サーバでは認証しません。SSL で認証します。
2. LDAPRequire ディレクティブを定義した場合には，LDAP サーバをアクセスして，アクセス制御します。
   クライアント証明書のサブジェクトの Common Name（CN）を，クライアントの名前として LDAP サーバを検索します。CN が LDAP サーバにない場合，ステータスコード 401 Authorization Required を応答します。
3. クライアント証明書の CN が LDAP サーバにある場合，検索フィルタを使って，このフィルタに一致するかどうか確認します。検索フィルタが (|(cn=TARO)(cn=HANAKO)) の場合，証明書の CN が TARO であれば検索フィルタに一致するのでアクセスできます。また，CN が JIRO の場合は，検索フィルタに一致しないため，LDAPNoEntryStatus ディレクティブに従ったステータスコード (デフォルトでは 401 Authorization Required) を応答し，アクセスを拒否します。
   LDAP サーバには必ずユーザを区別するための cn があり，これと証明書の CN が一致していると仮定して動作するため，この規則を基に SSL クライアント証明書を作成してください。

(5) 複数の LDAP サーバでユーザ認証

LDAP サーバは並列に複数指定できます。そのため，異なるユーザが登録されている LDAP サーバを併用してユーザ認証ができます。また，ディレクトリ単位にも指定できるため，コンテンツごとに LDAP サーバを変更できます。

(a) LDAP サーバの複数指定

LDAPServerName，LDAPServerPort および LDAPBaseDN ディレクティブに複数の LDAP サーバに対応したサーバ名，ポート番号および DN を指定できます。最初に指定した LDAP サーバの優先度が最も高く，指定した順に優先度は低くなります。

(b) ディレクトリ単位に LDAP サーバを指定

次に示す LDAP 関連のディレクティブはディレクトリ単位に指定できます。指定したディレクティブは httpsd.conf，<VirtualHost>，<Directory> の順に上位ディレクトリから下位ディレクトリへ継承します。

ディレクトリ単位に指定できる LDAP 関連ディレクティブ

- LDAPServerName
- LDAPServerPort
- LDAPTimeout
- LDAPBaseDN

4.6 ファイル名一覧の表示

ディレクトリ内のファイル名一覧を Web ブラウザに表示する機能をディレクトリインデクスといいます。ディレクトリインデクス機能を有効にするには次に示すディレクティブを定義します。

```
Options +Indexes
```

このとき，すべてのファイルを表示させることはセキュリティ上危険です。IndexIgnore ディレクティブでインデクス表示させないファイルを指定する必要があります。

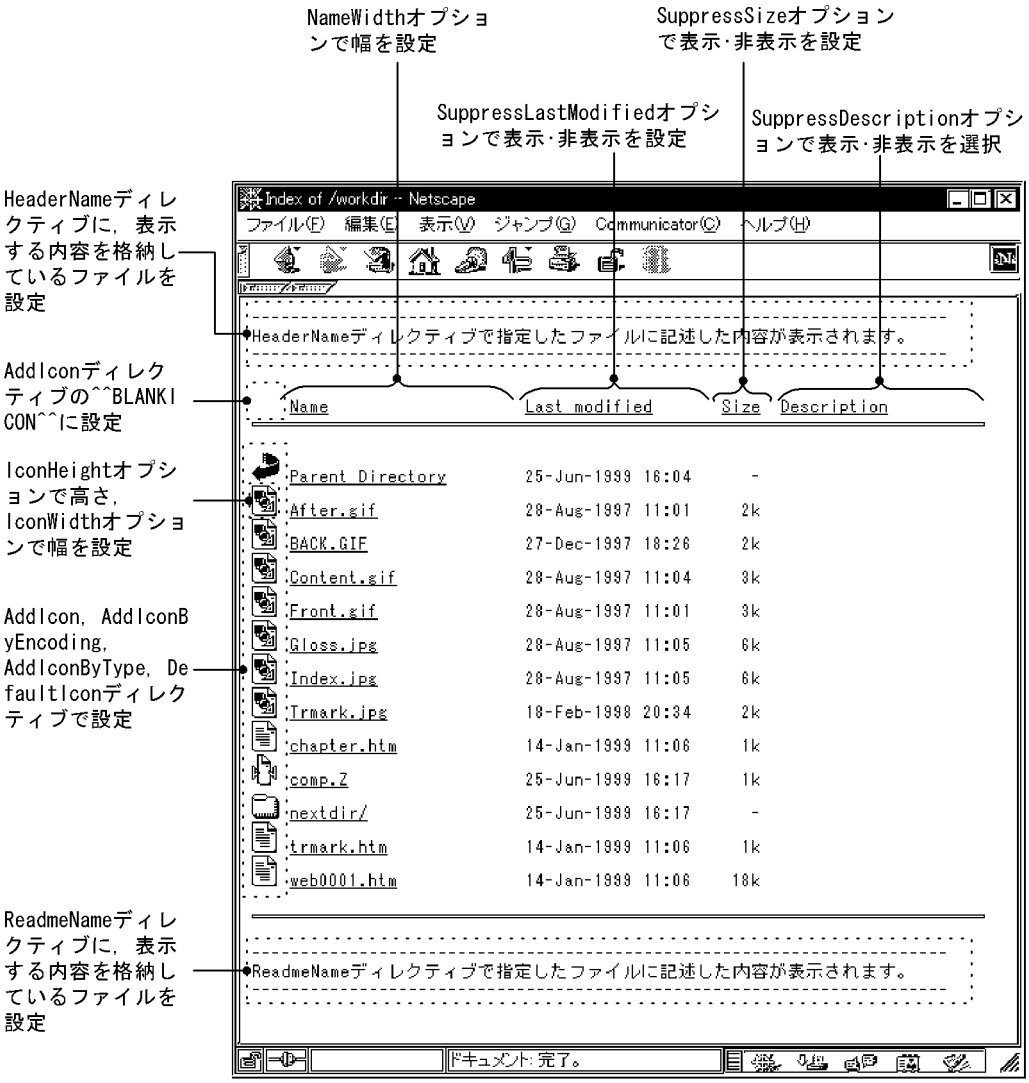
ただし，Options +Indexes を指定していても，DirectoryIndex ディレクティブに指定しているファイル（デフォルトは index.html ファイル）がそのディレクトリ下にある場合は，その指定されているファイルが表示されます。

さらに，ディレクトリインデクスを整形表示する場合は，次のディレクティブを指定します。

```
IndexOptions +FancyIndexing
```

整形表示機能の詳細設定は IndexOptions ディレクティブ，AddIcon ディレクティブで指定します。ディレクトリインデクス機能で表示される画面と，各ディレクティブで設定する内容を次に示します。

図 4-6 整形表示機能についての定義内容



なお，マルチバイト文字列を含むファイル名の表示はできません。

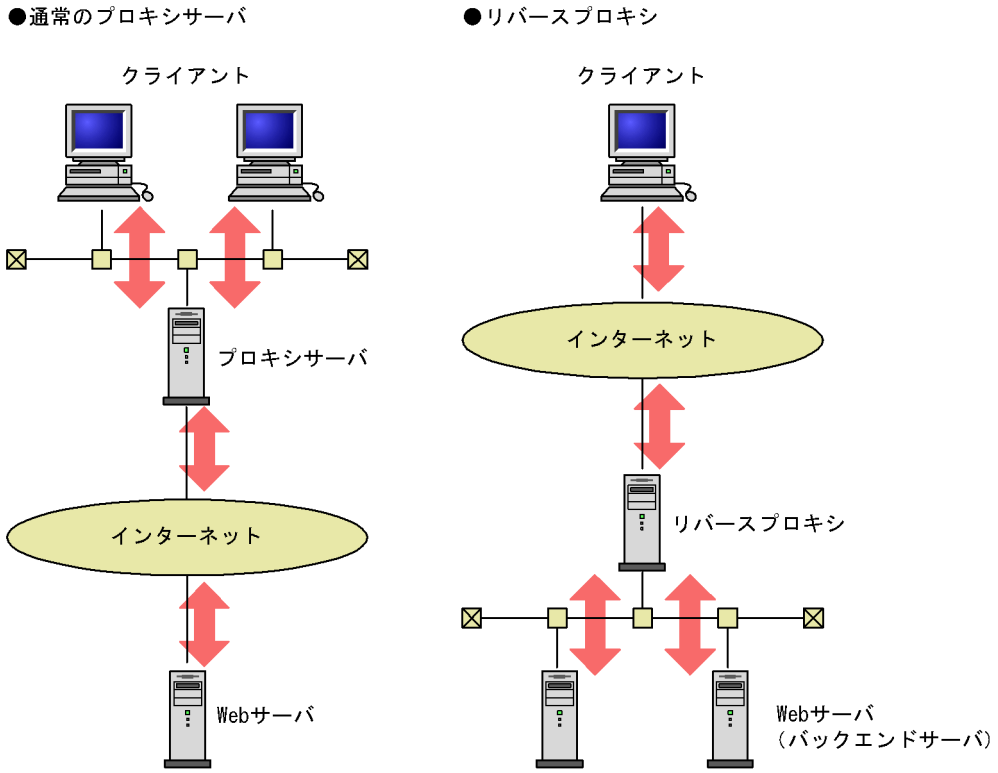
また，HeaderName ディレクティブおよび ReadmeName ディレクティブで指定したファイルで使用している文字セットが，デフォルトの文字セット (UTF-8) と異なる場合は，ディレクトリインデクス表示において文字化けが発生します。この場合，IndexOptions ディレクティブの Charset オプションで，HeaderName ディレクティブや ReadmeName ディレクティブで指定したファイルで使用している文字セットを指定してください。

4.7　リバースプロキシの設定

直接インターネットに接続できないクライアントからのリクエストをクライアントに代わって Web サーバに送信する代行サーバを**プロキシサーバ**といいます。通常，プロキシサーバは，クライアントとインターネットとの接点に設置されます。これに対し，インターネットと Web サーバとの接点にプロキシサーバを設置した場合を**リバースプロキシ**といいます。リバースプロキシでは，クライアントからのリクエストを Web サーバに代わってプロキシサーバが処理します。

通常のプロキシサーバとリバースプロキシの相違を次に示します。

図 4-7　通常のプロキシサーバとリバースプロキシの相違



リバースプロキシを使用してできることを次に示します。

- コンテンツへの直接アクセスを防止できます。
  Web サーバに重要な情報（クレジットカード番号のデータベースなど）を保持している場合，リバースプロキシと Web サーバを別のマシンに設定し，悪意のあるアクセスから Web サーバを守り，情報の漏えいを防げます。
- プロキシサーバに負荷の高い SSL 処理を集約できます。
  リバースプロキシを使用し，SSL の処理を別のマシンですれば，Web サーバに掛かる

負荷を分散できます。

● クライアントに影響を与えないで，Web サーバを分割できます。
Web サーバを分割した場合でも，リバースプロキシが代行するので，クライアントは分割前と同じインタフェースでアクセスできます。

(1) プロキシモジュールの組み込み

リバースプロキシを使用するためにはプロキシモジュールの組み込みが必要です。プロキシモジュールを組み込むにはコンフィグファイル（httpsd.conf）に次に示すディレクティブを指定します。プロキシモジュールを組み込むには，コンフィグファイル（httpsd.conf）に次に示すディレクティブを指定します。UNIX 版の場合は，必ず次に示す順序で LoadModule ディレクティブを指定してください。

- UNIX 版

```
LoadModule proxy_module libexec/mod_proxy.so
LoadModule proxy_http_module libexec/mod_proxy_http.so
```

- Windows 版

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

(2) ディレクティブの設定方法

リバースプロキシを設定する各ディレクティブの設定例を次に示します。

ここでは各アドレスを次のように仮定しています。

リバースプロキシ：www.example.com

バックエンドサーバ：backend.example.com

(a) リクエスト URL の再割り当ておよびリクエストヘッダの再割り当て

次のように ProxyPass ディレクティブを設定すると，クライアントからの "http://www.example.com/news/oct-2001" というリクエストは "http://backend.example.com/oct-2001" というリクエストに変更されます。

```
ProxyPass /news/ http://backend.example.com/
```

Host: ヘッダは "Host:www.example.com" から "Host:backend.example.com" に再割り当てします。そして，リバースプロキシはバックエンドサーバからのレスポンスをクライ

アントに応答します。

(b) 応答ヘッダの再割り当て

Redirect ディレクティブの指定，イメージマップの利用または末尾を /（スラッシュ）で閉じないディレクトリ指定のリクエストなど，バックエンドサーバでリダイレクトが指示された場合には，バックエンドサーバからのレスポンスの Location ヘッダにバックエンドサーバのアドレスが記載されます。これをそのままクライアントに応答すると，クライアントはリダイレクトをリバースプロキシではなく，直接バックエンドサーバにリクエストします。そこで，ProxyPassReverse ディレクティブに次のように指定し，リダイレクトリクエストもリバースプロキシを通るリクエストになるようにします。

```
ProxyPassReverse /news/ http://backend.example.com/
```

これで，Location ヘッダはリバースプロキシのアドレスに変更されます。

(c) Set-Cookie ヘッダの再割り当て

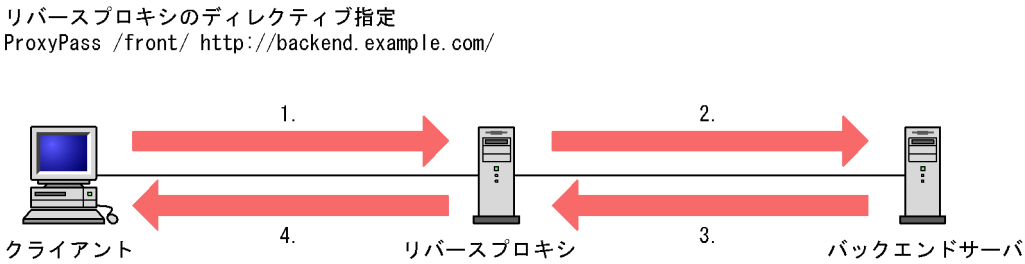
バックエンドサーバがクライアントに返す Set-Cookie ヘッダには，ドメイン名およびパス名が指定される場合があります。これは，Set-Cookie ヘッダのドメイン名およびパス名に一致したリクエストの場合だけ，クライアントにクッキーを送信させるためです。

Set-Cookie ヘッダの再割り当てをしない場合と再割り当てをする場合について説明します。

● Set-Cookie ヘッダの再割り当てをしない例

バックエンドサーバが応答したドメイン名およびパス名を含む Set-Cookie ヘッダをリバースプロキシがそのままクライアントに応答する例を次の図に示します。なお，図中の数字は，説明文の項番と対応しています。

図 4-8　Set-Cookie ヘッダの再割り当てをしない例



1. : http://www.example.com/front/cgi-bin/test-cgi.pl
2. : http://backend.example.com/cgi-bin/test-cgi.pl
3. : Set-Cookie:～; domain=backend.example.com; path=/cgi-bin/
4. : Set-Cookie:～; domain=backend.example.com; path=/cgi-bin/

1. クライアントからリバースプロキシに対して，http://www.example.com/front/cgi-bin/test-cgi.pl がリクエストされます。
2. リバースプロキシは，URL を変換してバックエンドサーバへ転送します。
3. リバースプロキシは，バックエンドサーバからドメイン名 domain=backend.example.com，パス名 path=/cgi-bin/ の Set-Cookie ヘッダを受信します。
4. リバースプロキシは，バックエンドサーバから受信した Set-Cookie ヘッダをそのままクライアントに返します。

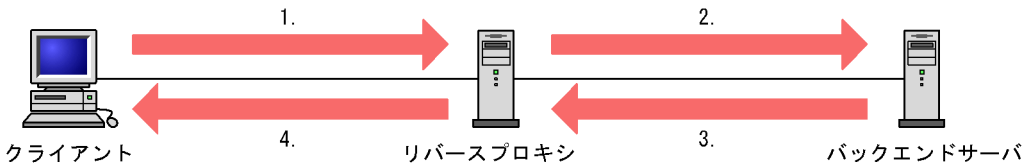
この場合，クライアントはリバースプロキシを経由する /front/cgi-bin/ 以下へのリクエストについて，Set-Cookie ヘッダで受信したクッキーを送信しません。これは，クライアントが受信した Set-Cookie ヘッダのドメイン名 backend.example.com が，リバースプロキシのドメイン名 www.example.com と異なるためです。また，パス名についても同様に適合しません。

● Set-Cookie ヘッダの再割り当てをする例

バックエンドサーバが Set-Cookie ヘッダで応答したクッキーをクライアントから受け取るためには，HWSProxyPassReverseCookie ディレクティブの指定が必要です。HWSProxyPassReverseCookie ディレクティブを指定して Set-Cookie ヘッダの再割り当てをする例を次の図に示します。なお，図中の数字は，説明文の項番と対応しています。

図 4-9　Set-Cookie ヘッダの再割り当てをする例

```
リバースプロキシのディレクティブ指定
ProxyPass /front/ http://backend.example.com/
HWSProxyPassReverseCookie /front/
```




```
1. : http://www.example.com/front/cgi-bin/test-cgi.pl
2. : http://backend.example.com/cgi-bin/test-cgi.pl
3. : Set-Cookie:～; domain=backend.example.com; path=/cgi-bin/
4. : Set-Cookie:～; path=/front/cgi-bin/
```

1. クライアントからリバースプロキシに対して，http://www.example.com/front/cgi-bin/test-cgi.pl がリクエストされます。
2. リバースプロキシは，URL を変換してバックエンドサーバへ転送します。
3. リバースプロキシは，バックエンドサーバからドメイン名 domain=backend.example.com，パス名 path=/cgi-bin/ の Set-Cookie ヘッダを受信します。
4. リバースプロキシは，再割り当てした Set-Cookie ヘッダをクライアントに返します。

この場合，クライアントはリクエスト URL のパス部分 /front/cgi-bin/test-cgi.pl に対して，前方一致するパス名 /front/cgi-bin/ の Set-Cookie ヘッダを受信します。また，クライアントが受信する Set-Cookie ヘッダにはドメイン名が含まれていません。これは，クライアントがリクエストした URL のドメイン名 www.example.com が Set-Cookie ヘッダに指定されている場合と同じ意味となります。したがって，リバースプロキシを経由したバックエンドサーバへのリクエストに，Set-Cookie ヘッダで設定したクッキーを送信させることができます。

(3) システム構築例

リバースプロキシとバックエンドサーバに Hitachi Web Server を使用してシステムを構築する場合の設定例を次に示します。

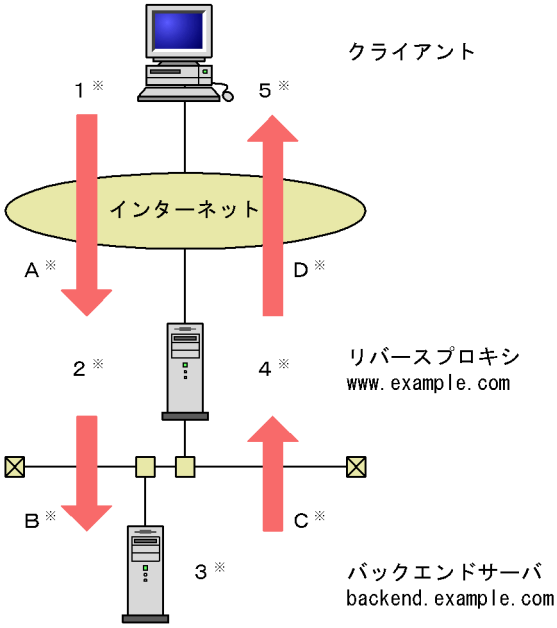
システムの構築時には，リダイレクト処理に注意して設定する必要があります。バックエンドサーバ上のディレクトリに対し，URL の最後に / (スラッシュ) を付けないでアクセスした場合，バックエンドサーバは Location ヘッダを付加したリダイレクト要求を返信します。このとき，Location ヘッダの値をバックエンドサーバのアドレスからリバースプロキシのアドレスに変換し，クライアントの再要求先をリバースプロキシ経由に変更する必要があります。

ここでは，システムのネットワーク構成を図 4-10 のように仮定しています。また各アドレスを次のように仮定しています。

リバースプロキシ：www.example.com

バックエンドサーバ：backend.example.com

図 4-10 ネットワーク構成



注※ 以降で説明する(a), (b)の各リダイレクト処理の流れと対応しています。

(a) 推奨する構成

ProxyPass ディレクティブに指定するホスト名，パス名と，ProxyPassReverse ディレクティブに指定するホスト名，パス名は同一の値としてください。また，バックエンドサーバ側のすべてのバーチャルホストで ServerName ディレクティブを指定し，その値はリバースプロキシ側の ProxyPassReverse ディレクティブに設定したホスト名と同一にしてください。

図 4-10 に示すネットワーク構成で，リバースプロキシおよびバックエンドサーバの設定を表 4-6 のようにした場合のリダイレクト処理の流れは，表 4-7 のようになります。

表 4-6 推奨する構成の設定例

設定場所	設定内容
リバースプロキシ	ServerName www.example.com ProxyPass /before/ http://backend.example.com/after/ ProxyPassReverse /before/ http://backend.example.com/after/
バックエンドサーバ	ServerName backend.example.com

表 4-7 推奨する構成でのリダイレクト処理の流れ

図中の位置	説明
1	"http://www.example.com/before/dir" にアクセスします。
2	ProxyPass ディレクティブの値に従い，"http://backend.example.com/after/dir" にアクセスします。また，Host ヘッダの値を backend.example.com に書き換えて転送します。
3	URL の末尾に /（スラッシュ）が付いていないため，URL の末尾に /（スラッシュ）を付けた URL を作成し，それを Location ヘッダに設定してリダイレクト要求を返します。
4	ProxyPassReverse ディレクティブの値に従い，Location ヘッダを "http://www.example.com/before/dir/" に書き換えて転送します。
5	Location ヘッダに従い，"http://www.example.com/before/dir/" に改めてアクセスします。
A	Host ヘッダの値は "www.example.com" です。
B	Host ヘッダの値は "backend.example.com" です。
C	Location ヘッダの値は "http://backend.example.com/after/dir/" です。
D	Location ヘッダの値は "http://www.example.com/before/dir/" です。

注

バックエンドサーバからの応答がステータスコード（302 Found や 404 Not Found など）になった場合，リバースプロキシはその HTML ドキュメントをそのままクライアントに転送します。「404 Not Found」などの HTML ドキュメントに記載されるバックエンドサーバ名や，「302 Found」などに記載されるリダイレクト先のリンクアドレスはリバースプロキシの情報に変更されません。バックエンドサーバ側で ErrorDocument ディレクティブを使用またはリバースプロキシ側で ProxyErrorOverride ディレクティブを使用して，バックエンドサーバの情報をクライアントに見せないようにしてください。

(b) リバースプロキシ側で ProxyPreserveHost ディレクティブに On を設定する構成

通常，リバースプロキシはクライアントから受信した Host ヘッダの値を ProxyPass ディレクティブの値に従って変更し，バックエンドサーバに転送します。クライアントが送信した Host ヘッダの値をバックエンドサーバ側でも Host ヘッダの値として取得したい場合は，リバースプロキシ側で ProxyPreserveHost ディレクティブの値を On に設定します。このとき，次の点に注意してください。

- バックエンドサーバ側の ServerName ディレクティブには，リバースプロキシの ServerName と同じ値を指定してください。
- ProxyPassReverse ディレクティブに設定するホスト名は，リバースプロキシおよびバックエンドサーバの ServerName と同じ値にしてください。

図 4-10 に示すネットワーク構成で，リバースプロキシおよびバックエンドサーバの設定を表 4-8 のようにした場合のリダイレクト処理の流れは表 4-9 のようになります。

表 4-8 リバースプロキシ側で ProxyPreserveHost に On を設定する構成の設定例

設定場所	設定内容
リバースプロキシ	ServerName www.example.com ProxyPass /before/ http://backend.example.com/after/ ProxyPassReverse /before/ http://www.example.com/after/ ProxyPreserveHost On
バックエンドサーバ	ServerName www.example.com

表 4-9 リバースプロキシ側で ProxyPreserveHost に On を設定する構成でのリダイレクト処理の流れ

図中の位置	説明
1	"http://www.example.com/before/dir" にアクセスします。
2	ProxyPass ディレクティブの値に従い，"http://backend.example.com/after/dir" にアクセスします。また，ProxyPreserveHost ディレクティブの値が On に設定されているため，Host ヘッダの値は www.example.com のままです。
3	URL の末尾に /（スラッシュ）が付いていないため，URL の末尾に /（スラッシュ）を付けた URL を作成し，それを Location ヘッダに設定してリダイレクト要求を返します。
4	ProxyPassReverse ディレクティブの値に従い，Location ヘッダを "http://www.example.com/before/dir/" に書き換えて転送します。
5	Location ヘッダに従い，"http://www.example.com/before/dir/" に改めてアクセスします。
A	Host ヘッダの値は "www.example.com" です。
B	Host ヘッダの値は "www.example.com" です。
C	Location ヘッダの値は "http://www.example.com/after/dir/" です。
D	Location ヘッダの値は "http://www.example.com/before/dir/" です。

(4) 注意事項

(a) 基本的な注意事項

- リバースプロキシはリクエスト URL のパターンによって機能を設定します。このため，特定のリクエストはリバースプロキシとしてほかのバックエンドサーバに転送，それ以外のリクエストはリバースプロキシ自身が Web サーバとして応答するという設定もできます。しかし，このような設定は，リクエストがリバースプロキシか Web サーバかどちらで処理したかがわかりにくくなります。したがって，リバースプロキシを使用する場合は次のような設定にして，すべてのリクエストをリバースプロキシからバックエンドサーバへ転送することを推奨します。

```
ProxyPass / http://転送先バックエンドサーバアドレス/
```

リバースプロキシと Web サーバを共用する場合は，バーチャルホストで機能を分けた運用ができます。

- リバースプロキシでは，クライアントから受信した Host ヘッダの値を X-Forwarded-Host ヘッダに格納し，Host ヘッダの値を ProxyPass ディレクティブの指定値に変換してバックエンドサーバへ転送します。このため，バックエンドサーバ側のアプリケーションでクライアントが送信した Host ヘッダの値を参照する場合は，リバースプロキシが送信した X-Forwarded-Host ヘッダの値を参照してください。ただし，ProxyPreserveHost ディレクティブの値に On を設定している場合は，リバースプロキシが送信した Host ヘッダの値をそのまま参照してください。
- リバースプロキシを経由してバックエンドサーバにアクセスする場合，バックエンドサーバが提供する HTML コンテンツでのリンク先は，バックエンドサーバ上の URL ではなく，リバースプロキシにアクセスされる URL を指定する必要があります。このほか，画像やスタイルシートなどのコンテンツの参照先 URL を記述する場合も同じように注意が必要です。

  （例）

  次の状態にあるときに，index.html から index2.html へリンクを張るとします。
  - リバースプロキシ側で ProxyPass ディレクティブの値が /before/ http:// バックエンドサーバのアドレス /after/ と指定されている。
  - バックエンドサーバ側で index.html と index2.html が同じディレクトリ内（/after/ 以下）に存在する。

  この場合の index.html の記述方法とアクセス可否の関係を次に示します。

表 4-10　リンクの記述方法とリンク可否の関係

リンクの記述	リンクをクリックしたときのアクセス可否
<A HREF="index2.html"> リンク </A>	○
<A HREF="/before/index2.html"> リンク </A>	○
<A HREF="http:// リバースプロキシのアドレス /before/index2.html"> リンク </A>	○
<A HREF="/after/index2.html"> リンク </A>	×

- リバースプロキシは，HTTP バージョン 0.9 をサポートしていません。

(b) ProxyPass ディレクティブに関する注意事項

- ProxyPass ディレクティブで指定するパス名とリクエスト URL は完全に等しいか，パス名がリクエスト URL の先頭から含まれていれば，適合と判断します。
  ただし，パス名の終端が /（スラッシュ）でない場合，リクエスト URL と完全に等しいかまたは先頭からディレクトリとして含まれていれば適合と判断します。
  適合すると，ProxyPass ディレクティブに指定したパス名にリクエスト URL の先頭からパス名と等しい部分を除いた残りの部分を追加してリクエストを転送します。
  ProxyPass ディレクティブに指定するパス名は終端を /（スラッシュ）で閉じたもの

を指定してください。次に ProxyPass ディレクティブの指定とリクエストの関係を次に示します。

表 4-11 ProxyPass ディレクティブの指定とリクエストの関係

ProxyPass ディレクティブの指定例	リクエスト	適合可否	リクエスト転送先
ProxyPass /abc/ http://backend.example.com/	http:// リバースプロキシのアドレス /abc/	○	http://backend.example.com/
	http:// リバースプロキシのアドレス /abc	×	―
	http:// リバースプロキシのアドレス /abc/def	○	http://backend.example.com/def
ProxyPass /abc http://backend.example.com/	http:// リバースプロキシのアドレス /abc	○	http://backend.example.com/
	http:// リバースプロキシのアドレス /abc/	○	http://backend.example.com//
	http:// リバースプロキシのアドレス /abc/def	○	http://backend.example.com//def

（凡例）

○：適合する。
×：適合しない。
―：該当しない。

- ProxyPass ディレクティブを複数指定し，リクエスト URL が複数のパス名に一致した場合，先に指定した ProxyPass ディレクティブが有効になります。

（例）

/abc/def/ へのリクエストを処理するバックエンドサーバ：backend1.example.com
/abc/def/ 以外の /abc/ へのリクエストを処理するバックエンドサーバ：backend2.example.com
ほかのすべてのリクエストを処理するバックエンドサーバ：backend3.example.com
このように設定するには次の順序で指定してください。

```
ProxyPass /abc/def/ http://backend1.example.com/
ProxyPass /abc/ http://backend2.example.com/
ProxyPass / http://backend3.example.com/
```

- ProxyPass ディレクティブの指定によるリクエスト URL は Web サーバの機能である自プロセス内の該当ファイルの検索より先に転送します。したがって，リクエスト URL に適合するファイルがある場合でも，ProxyPass ディレクティブのパス名に適合すれば，バックエンドサーバへのリクエストに変換して転送します。

- /（スラッシュ）で閉じていないディレクトリを指定したリクエスト URL の場合，リバースプロキシではリダイレクトを応答しません。

  （例）ProxyPass /ab/ http://backend.example.com/ **の場合**

  リクエストが http:// リバースプロキシのアドレス /ab のとき，適合していないと判断して，リバースプロキシ内に /ab がなければ，「404 Not Found」を返します。

(c) ProxyPassReverse ディレクティブに関する注意事項

- ProxyPassReverse ディレクティブに指定した URL とバックエンドサーバから受信した Location ヘッダの値は，完全に等しいかまたは URL がリクエスト URL の先頭から含まれていれば，適合と判断します。適合すると，アドレスをリバースプロキシとして，ProxyPassReverse ディレクティブの指定に従って，クライアントに送信します。

  （例）バックエンドサーバからの応答の Location **ヘッダが，**Location: http:// **バックエンドサーバのアドレス** /docs/memo/ **の場合**

  ProxyPassReverse ディレクティブの指定が，
  ProxyPassReverse /path/ http:// バックエンドサーバのアドレス /docs/
  と指定されていれば，クライアントに返す Location ヘッダは
  Location: http:// リバースプロキシのアドレス /path/memo/
  となります。

  ProxyPassReverse ディレクティブを複数指定した場合，先に指定した方が有効になります。
- リバースプロキシが ProxyPassReverse ディレクティブの設定値に従って Location ヘッダの値を変換してクライアントに転送する際，Location ヘッダの値のスキームは現在のコネクションで使用しているものを設定します。例えば，http でアクセスしている場合，http を設定します。このため，http でアクセスしている場合に Location ヘッダで https にリダイレクトさせるときは，バックエンドサーバ側でリバースプロキシのホスト名を Location ヘッダの値に設定しておくなどして ProxyPassReverse ディレクティブの値と一致しないようにしてください。

(d) HWSProxyPassReverseCookie ディレクティブに関する注意事項

- HWSProxyPassReverseCookie ディレクティブは，バックエンドサーバが応答した Set-Cookie ヘッダを変換する場合に指定します。HWSProxyPassReverseCookie ディレクティブに，ProxyPass ディレクティブのパス名と同じ値を指定することで，ProxyPass ディレクティブ単位に設定できます。
- リバースプロキシのディレクティブ指定が次のような場合の Set-Cookie ヘッダの変換規則について説明します。

```
ProxyPass /front/ http://backend.example.com/
HWSProxyPassReverseCookie /front/
```

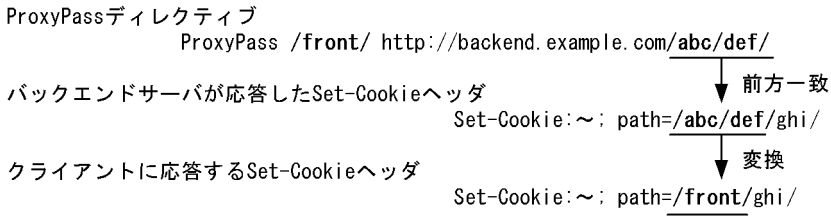
表 4-12 Set-Cookie ヘッダの変換規則

項番	クライアントに応答する Set-Cookie ヘッダ	バックエンドサーバが応答する Set-Cookie ヘッダ	変換規則の説明
1	Set-Cookie: ～ ; path=/front/	Set-Cookie: ～ ; path=/	バックエンドサーバが応答する Set-Cookie ヘッダにドメイン名が指定されていない場合は，Set-Cookie ヘッダのパス名 /（スラッシュ）を /front/ に置き換えます。
2	Set-Cookie: ～ ; path=/front/	Set-Cookie: ～ ; domain=backend.example.com ; path=/	バックエンドサーバが応答する Set-Cookie ヘッダのドメイン名が，ProxyPass ディレクティブで指定した転送先 URL のドメイン名と完全に一致している場合は，Set-Cookie ヘッダのパス名 /（スラッシュ）を /front/ に置き換えます。また，Set-Cookie ヘッダのドメイン名を削除してクライアントに返します。
3	Set-Cookie: ～ ; domain=.example.com; path=/	Set-Cookie: ～ ; domain=.example.com; path=/	バックエンドサーバが応答する Set-Cookie ヘッダのドメイン名が，.（ピリオド）から始まるドメイン名である場合は，バックエンドサーバが応答した Set-Cookie ヘッダをそのままクライアントに返します。
4	Set-Cookie: ～ ; domain=other.example.com; path=/	Set-Cookie: ～ ; domain=other.example.com; path=/	バックエンドサーバが応答する Set-Cookie ヘッダのドメイン名が ProxyPass ディレクティブで指定した転送先 URL のドメイン名と異なる場合は，バックエンドサーバが応答した Set-Cookie ヘッダをそのままクライアントに返します。
5	Set-Cookie: ～	Set-Cookie: ～	バックエンドサーバが応答する Set-Cookie ヘッダにドメイン名およびパス名が指定されていない場合は，バックエンドサーバが応答した Set-Cookie ヘッダをそのままクライアントに返します。

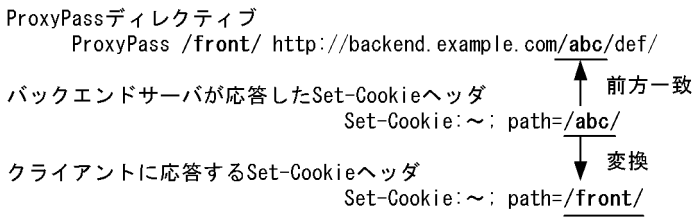
- リバースプロキシのディレクティブ指定が次の場合に，バックエンドサーバが応答した Set-Cookie ヘッダのパス名を変換する規則について説明します。

```
ProxyPass /front/ http://backend.example.com/abc/def/
HWSProxyPassReverseCookie /front/
```

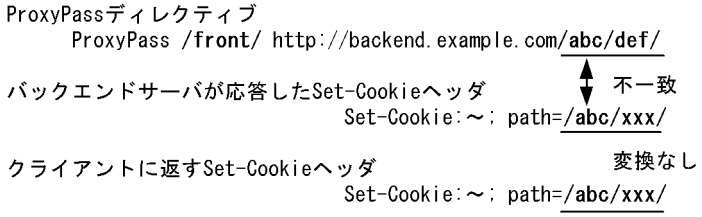
- **バックエンドサーバが返す Set-Cookie ヘッダのパス名が /abc/def/ghi/ の場合**
  ProxyPass ディレクティブの転送先 URL のパス名部分が，Set-Cookie ヘッダのパス名に前方から一致する場合は，一致したパス名部分を ProxyPass ディレクティブのパス名で置き換えます。



- **バックエンドサーバが返す Set-Cookie ヘッダのパス名が /abc/ の場合**
  Set-Cookie ヘッダのパス名が，ProxyPass ディレクティブの転送先 URL のパス名部分に前方から一致する場合は，Set-Cookie ヘッダのパス名として ProxyPass ディレクティブのパス名で置き換えます。



- **バックエンドサーバが返す Set-Cookie ヘッダのパス名が /abc/xxx/ の場合**
  ProxyPass ディレクティブの転送先 URL のパス名部分と Set-Cookie ヘッダのパス名が一致しない場合は，リバースプロキシでの Set-Cookie ヘッダ変換は実行しません。バックエンドサーバが応答した Set-Cookie ヘッダをそのままクライアントに返します。



(e) 性能に関する注意事項

ProxyPass ディレクティブにドメイン名またはホスト名を指定している場合，DNS への問い合わせが発生します。バックエンドサーバの IP アドレスがわかっている場合は，hosts ファイルにあらかじめ IP アドレスを記載しておくことによって，名前解決の時間を短縮できます。

4.8 稼働状況の表示（ステータス情報表示）

稼働中のプロセス数，待機中のプロセス数および各プロセスのステータス（R，W，L など）を Web ブラウザに表示します（Windows 版の場合はサーバスレッド数）。この情報を基に，StartServers，MinSpareServers，MaxSpareServers，MaxClients ディレクティブなどをチューニングできます（Windows 版の場合は ThreadsPerChild ディレクティブ）。各ディレクティブの詳細は，「4.1 Hitachi Web Server の処理とディレクティブとの関係」を参照してください。

ExtendedStatus ディレクティブで On を指定すると，より詳細な情報が表示されます。

（1） server-status ハンドラの指定

ステータス情報の表示機能を利用するには，次に示すように server-status ハンドラを指定します。

```
<Location /server-status>
      SetHandler server-status
</Location>
```

ただし，Web サーバのステータス情報はアクセス制御して，エンドユーザには非公開にするのが一般的です。

（2） URL の指定

ステータス情報を表示するには，Web ブラウザから次に示す形式で URL を指定します。なお，server-status は，タイミングによって一時的に正しく表示されない場合があります。

```
http://ホスト名〔:ポート番号〕/server-status〔? {refresh=更新間隔 | auto | notable} 〕
```

refresh= 更新間隔，auto，notable はそれぞれ & でつないで指定できます。ただし，auto はプレーンテキスト形式であるため，notable と同時に指定するのは意味がありません。

● refresh= **更新間隔**　((1-3600))

Web ブラウザ上のステータス情報を更新する間隔を秒単位で指定します。ただし，Web ブラウザが HTTP レスポンスヘッダの Refresh ヘッダに対応した機能をサポートしている必要があります。指定可能範囲外の値が指定された場合は 60 秒が設定されます。

● auto

プレーンテキスト形式で表示します。プレーンテキスト形式のため，ほかのプログラムで容易に処理できます。

● notable

<TABLE> タグを用いない HTML でステータス情報を表示します。

＜指定例＞

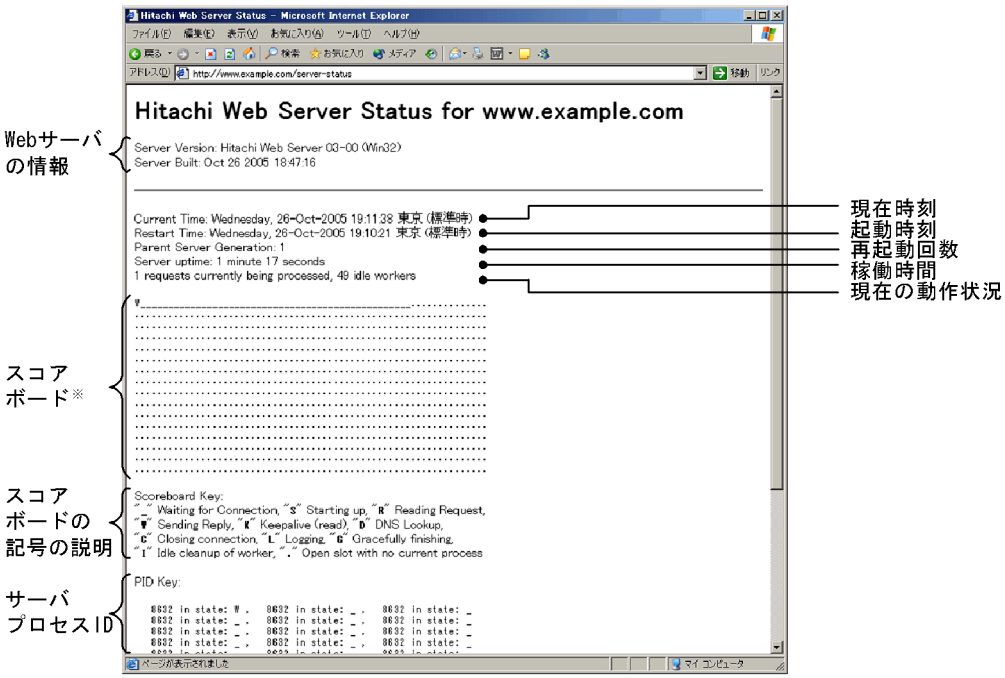
```
http://www.example.com/server-status?refresh=60&notable
```

＜表示例＞

```
http://www.example.com/server-status
```

このように指定した場合の，ステータス情報の表示例を次に示します。表示形式は，UNIX 版と Windows 版で若干異なります。

図 4-11 ステータス情報の表示例



注※ スコアボードには，稼働中のサーバプロセスの状態を記号で示す。
記号の意味を次に示す。

_ ... リクエスト待ち状態
S ... 起動処理中
R ... クライアントからのリクエストを受信中
W ... リクエストの処理実行，およびクライアントへレスポンス送信中
K ... 持続型接続状態でリクエスト受信待ち
D ... ルックアップ中(HostnameLookupsディレクティブ参照)
C ... 接続を終了中
L ... ログ出力処理中
G ... gracefulリスタートにおける処理終了待ち
I ... スレッド停止中
. ... 起動していない状態

(3) 取得できる情報

ステータス情報の表示機能で取得できる情報を次に示します。ExtendedStatus ディレクティブで On を指定すると，詳細な情報を取得できます。

表 4-13 ステータス情報の表示機能で取得できる情報（auto 指定がない場合）

項番	内容	説明	ExtendedStatus の値と取得可否	
			Off	On
1	Server Version	サーバのバージョン	○	○
2	Server Built	サーバのビルド時間	○	○
3	Current Time	現在時刻	○	○
4	Restart Time	起動時刻	○	○
5	Parent Server Generation	サーバプロセスの再起動回数（初期値 0）	○	○
6	Server uptime	サーバプロセスの稼働時間	○	○
7	Total accesses	合計アクセス回数	×	○
8	Total Traffic	合計通信量	×	○
9	CPU Usage: u vvv s www cu xxx cs yyy - zzz% CPU load	ユーザ時間，システム時間，子プロセスのユーザ時間，子プロセスのシステム時間，CPU 使用率（UNIX 版）	×	○
10	xxx requests/sec - yyy B/second - zzz B/request	1 秒当たりのリクエスト数， 1 秒当たりの通信量， 1 リクエスト当たりの通信量	×	○
11	xxx requests currently being processed, yyy idle workers	リクエスト処理中のサーバプロセス（スレッド）数， リクエスト待ち状態のサーバプロセス（スレッド）数	○	○
12	スコアボード	個々のスレッドの動作状況	○	○
13	Scoreboard Key	スコアボードの凡例	○	○
14	PID Key	個々のスレッドのサーバプロセス ID と動作状況	○	×
15	Srv	サーバプロセスの識別子と再起動回数	×	○
16	PID	プロセス ID	×	○
17	Acc	アクセス数（コネクション単位 / スレッド単位 / スロット単位）	×	○
18	M	動作状況	×	○
19	CPU	CPU 時間（秒）（UNIX 版）	×	○
20	SS	最後の処理開始からの経過秒	×	○
21	Req	最後の処理に要したミリ秒	×	○
22	Conn	コネクションに対する通信量	×	○
23	Child	プロセスの通信量	×	○
24	Slot	スロットの通信量	×	○
25	Client	最後の処理のクライアント	×	○

項番	内容	説明	ExtendedStatus の値と取得可否	
			Off	On
26	VHost	バーチャルホスト名	×	○
27	Request	最後の処理のリクエストライン	×	○

（凡例）
○：取得できる。
×：取得できない。

表 4-14 ステータス情報の表示機能で取得できる情報（auto 指定がある場合）

項番	内容	説明	ExtendedStatus の値と取得可否	
			Off	On
1	Total accesses	合計アクセス回数	×	○
2	Total kBytes	合計通信量	×	○
3	CPULoad	CPU 使用率（UNIX 版）	×	○
4	Uptime	サーバプロセスの稼働時間（秒）	×	○
5	ReqPerSec	1 秒当たりのリクエスト数	×	○
6	BytesPerSec	1 秒当たりの通信量	×	○
7	BytesPerReq	1 リクエスト当たりの通信量	×	○
8	BusyWorkers	リクエスト処理中のサーバプロセス（スレッド）数	○	○
9	IdleWorkers	リクエスト待ち状態のサーバプロセス（スレッド）数	○	○
10	スコアボード	個々のスレッドの動作状況	○	○

（凡例）
○：取得できる。
×：取得できない。

(4) 注意事項

サーバステータス表示機能で表示される「Current Time」および「Restart Time」のタイムゾーンの情報に，マルチバイト文字が設定されることがあります。このとき，Hitachi WebServer では，これらの文字列をすべてエスケープ（「¥x」から始まる接頭辞と 16 進コードで構成される文字列に置換）します。

4.9 流量制限機能

Web サーバへのアクセスの増加や，業務アプリケーションなどの影響で Web サーバの負荷が高くなった場合に，Web サイトにアクセスするユーザ数を制限するなどして，Web サービスの処理効率を維持する機能を流量制限機能といいます。

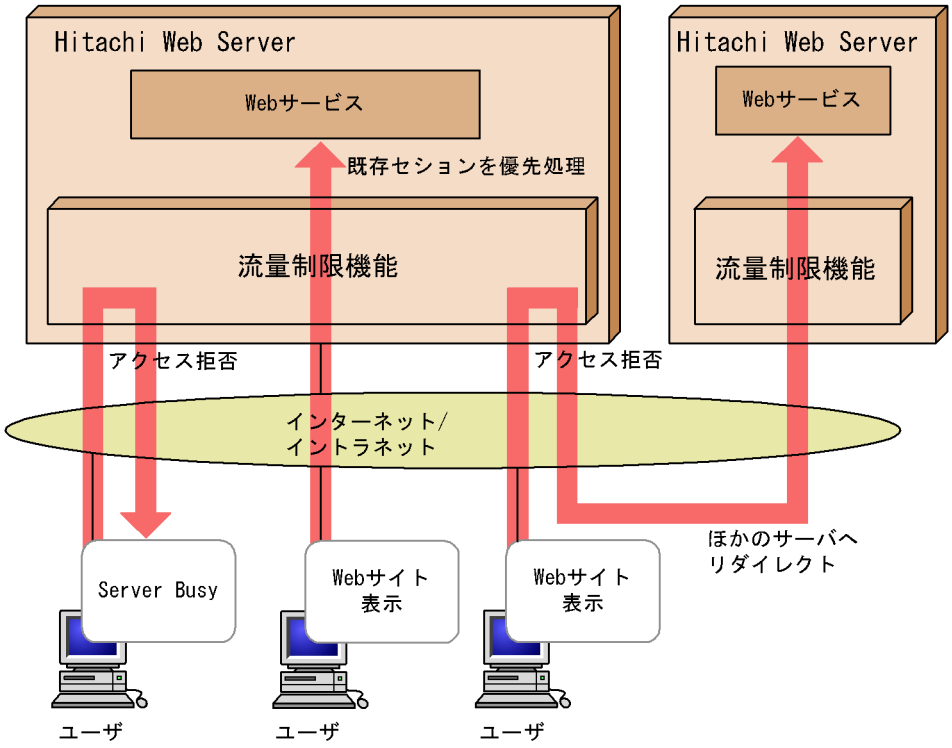
Hitachi Web Server に mod_hws_qos モジュールを組み込めば，流量制限機能を使用できます。流量制限機能を使用すると，次に示すことができます。なお，以降の記述で「サーバプロセス数」は，Windows 版の場合「サーバスレッド数」のことです。

- リクエスト処理を実施するサーバプロセス数を制限すれば，Web サイトに同時にアクセスするユーザ数を制限できます。高負荷時には，制限値を超えたリクエストに対してすぐに拒否レスポンスを返したり，ほかの Web サーバへリダイレクトさせたりすれば，レスポンスタイムを維持できます。
- クッキーを使用したセション管理によって，高負荷時，新しいセションを拒否し，すでにアクセスしているユーザのレスポンスタイムを維持できます。

十分な Web サービスを提供するには，同時にアクセスするユーザ数を満たすだけのサーバプロセス数が必要です。Web サーバが 1 台で不十分なら，複数台用意し，負荷分散機でアクセスを分散させるなどして Web サービスを保証できるように運用設計してください。

このように Web サービスの資源を用意しても，一時的に過負荷状態が発生する場合に備えて，流量制限機能を使用してください。mod_hws_qos による流量制限機能の概要を次に示します。

図 4-12 mod_hws_qos による流量制限機能の概要



(1) mod_hws_qos モジュールの組み込み

流量制限機能を使用するためには mod_hws_qos モジュールの組み込みが必要です。mod_hws_qos モジュールを組み込むには，コンフィグファイル（httpsd.conf）に次に示すディレクティブを指定します。

- UNIX 版

```
LoadModule hws_qos libexec/mod_hws_qos.so
```

- Windows 版

```
LoadModule hws_qos modules/mod_hws_qos.so
```

(2) ディレクティブの設定方法

流量制限機能を使用するための，各ディレクティブの設定例を次に示します。

(a) サーバプロセス数の制限によるリクエスト拒否

次のように指定すると，リクエスト処理中のサーバスレッド数が 13 の場合には，新たなリクエスト要求はステータスコード 503 で拒否されます。

- UNIX 版

```
MaxClients 15
QOSRejectionServers 2
QOSCookieServers 0
```

- Windows 版

```
ThreadsPerChild 15
QOSRejectionServers 2
QOSCookieServers 0
```

(b) クッキーを使用したセション管理

クッキーを使用したセション管理には，Hitachi Web Server で作成したクッキーを使用する HWS 作成モードと，Hitachi Web Server 以外の外部モジュールなどで作成されたクッキーを使用するユーザ作成モードがあります。QOSCookieName ディレクティブを用いて，どちらの方式を使用するかを選択します。QOSCookieName ディレクティブの詳細については，「6.2　ディレクティブの詳細」を参照してください。

HWS 作成モード

リクエスト処理を実施した場合には，Hitachi Web Server で作成したクッキーが，レスポンスヘッダの Set-Cookie ヘッダに付けられます。Hitachi Web Server で作成したクッキーを持ったリクエスト要求は，持っていないリクエスト要求よりも優先して処理されます。

ユーザ作成モード

Hitachi Web Server 以外で作成されたクッキーがレスポンスヘッダの Set-Cookie ヘッダに付けられる場合に，そのクッキーを使用した流量制限が実行されます。そのクッキーを持ったリクエスト要求は，持っていないリクエスト要求よりも優先して処理されます。

次のように指定すると，リクエスト処理中のサーバスレッド数が 10 の場合，クッキーを持っていない新しいセションのリクエスト要求は拒否されますが，クッキーを持った継続セションは処理されます。リクエスト処理中のサーバスレッド数が 13 の場合は，クッキーを持っているかどうかに関係なく拒否されます。この例では HWS 作成モードで動作します。

- UNIX 版

```
MaxClients 15
QOSRejectionServers 2
QOSCookieServers 5
```

- Windows 版

```
ThreadsPerChild 15
```

```
QOSRejectionServers 2
QOSCookieServers 5
```

(c) リダイレクト

流量制限機能によってリクエスト処理を拒否する場合には，ステータスコード 503 でレスポンスメッセージを返送しますが，次のように指定すると，ほかの Web サーバへリダイレクトさせることができます。/index.html へのリクエスト要求が流量制限機能によって拒否された場合には，www1.hitachi.co.jp の Web サーバの index.html をレスポンスヘッダに設定し，ステータスコード 302 で返送されます。

```
QOSRedirect /index.html http://www1.hitachi.co.jp/index.html
```

(d) レスポンスメッセージのカスタマイズ

次のように指定すると，拒否されたリクエストに対するレスポンスは，ステータスコード 503 で，レスポンスメッセージは，htdocs/busy.html の内容が返送されます。

```
QOSResponse file "text/html; charset=ISO-8859-1" htdocs/busy.html
```

(3) レスポンスメッセージ

(a) サーバからクライアントに送信されるクッキーについて

HWS 作成モード

Hitachi Web Server で作成したクッキーは，Set-Cookie ヘッダによってクライアントへ返送されます。Set-Cookie ヘッダは一つのレスポンスに複数指定できるため，ここで作成するクッキーは，ほかのクッキーに影響しません。Hitachi Web Server で返送する Set-Cookie ヘッダを次に示します。

```
Set-Cookie: NAME=VALUE; expires=DATE; path=/; domain=DOMAIN_NAME; secure
```

NAME=VALUE

NAME には QOSCookieName ディレクティブで指定した名称が設定されます。VALUE にはリクエスト制御用の値が設定されます。

expires=DATE

クッキーが無効となる時刻。" リクエストの受信時刻＋ QOSCookieExpires ディレクティブ設定値 " によって求めた値が，RFC822 形式で設定されます。

path=/

クッキーが有効となる URL。このモジュールでは，クッキーが有効となるドメイン内の，すべての URL で有効となるように設定されます。

domain=DOMAIN_NAME

クッキーが有効となるドメイン。QOSCookieDomain ディレクティブで指定されます。

secure

SSL による通信時だけ，クッキーをクライアントからサーバに送信するかどうかの指定。QOSCookieSecure ディレクティブで指定されます。

ユーザ作成モード

Hitachi Web Server ではクッキーを作成しません。Hitachi Web Server 以外で作成されたクッキーが，Set-Cookie ヘッダによってクライアントへ返送されます。

(b) 流量制限機能によって拒否された場合のヘッダについて

流量制限機能によって拒否された場合，レスポンスメッセージをキャッシュできなくするためのヘッダである Expires を，レスポンスヘッダに含めます。これは，サーバ側でリクエスト処理が可能であっても，プロキシまたはブラウザでキャッシュされると，キャッシュされたメッセージがブラウザに表示されて，サーバにリクエストしない場合があるためです。また，拒否メッセージ送信後はサーバ側からコネクションを切断します。

そのほかのレスポンスヘッダは，次のように設定されます。Content-Type には，AddDefaultCharset で指定した文字セットが付加されます。

(i) ステータスコード 503 の標準メッセージ

Content-Type：text/html

(ii)QOSResponse によるカスタマイズされたメッセージ

Content-Type：QOSResponse ディレクティブ指定値

(iii)QOSRedirect によるステータスコード 302 のメッセージ

Content-Type：text/html

Location：QOSRedirect ディレクティブ指定値

(4) 注意事項

- クッキーの受け付けを拒否しているクライアントは，クッキーをサーバに送信しないため，「(2)(b)　クッキーを使用したセション管理」の機能が無効となります。
- KeepAlive によって接続した場合には，接続後最初のリクエストを処理するときだけ判定処理が行われ，同一接続上で 2 回目以降のリクエストでは判定されません。最初に接続したリクエストとは異なる流量制限設定へのリクエストである場合も，2 回目以降のリクエストでは判定されません。
- 流量制限機能によってアクセス拒否された場合に送信されるメッセージは，ErrorDocument ディレクティブを指定しても変更されません。また，Redirect ディレクティブや RedirectMatch ディレクティブの処理は，mod_hws_qos モジュールの制御によって処理を継続すると判定された後に実行されます。
- 流量制限機能を使用しても，QOSRejectionServers ディレクティブの設定数を超えた同時リクエストを受信した場合は，リクエストの拒否が正しくできないことがあります。QOSResponse ディレクティブで指定された HTML ファイルに画像データなどのリンクが含まれている場合には，画像データを取得するため，さらにアクセスします。このアクセスも流量制限処理の対象となり，画像データが取得できない場合がありま

す。HTML ファイル作成時には，リンクの設定に注意してください。

- エラードキュメントの文字セットは，Windows 版では HWSErrorDocumentMETACharset ディレクティブの設定を有効とします。
- クッキーを使用したセション管理を URL ごとまたは VirtualHost ごとに設定する場合は，QOSCookieName ディレクティブで別の名称を指定する必要があります。
- QOSCookieServers ディレクティブおよび QOSRejectionServers ディレクティブは，サーバプロセス数（ThreadsPerChild ディレクティブまたは MaxClients ディレクティブで設定）より後ろに指定してください。サーバプロセス数よりも前に指定すると，サーバが起動できない場合があります。

4.10 ヘッダカスタマイズ機能

HTTP 通信では，Web ブラウザと Web サーバの間でさまざまな HTTP ヘッダが用いられます。Web ブラウザおよび Web サーバは，受信した HTTP ヘッダから，その後の動作を決定する場合があります。Web ブラウザが HTTP リクエストの送信時に付加する HTTP ヘッダをリクエストヘッダ，Web サーバが応答時に付加する HTTP ヘッダをレスポンスヘッダといいます。Web サーバが受信したリクエストヘッダや送信するレスポンスヘッダを追加，変更または削除して，Web サーバまたは Web ブラウザに特定の動作をさせるための機能をヘッダカスタマイズ機能といいます。

Hitachi Web Server に mod_headers モジュールを組み込むことで，ヘッダカスタマイズ機能を使用できます。

(1) mod_headers モジュールの組み込み

ヘッダカスタマイズ機能を使用するためには mod_headers モジュールの組み込みが必要です。mod_headers モジュールを組み込むには，コンフィグファイル（httpsd.conf）に次に示すディレクティブを指定します。

- UNIX 版

```
LoadModule headers_module libexec/mod_headers.so
```

- Windows 版

```
LoadModule headers_module modules/mod_headers.so
```

(2) ディレクティブの設定方法

ヘッダカスタマイズ機能は，Header ディレクティブおよび RequestHeader ディレクティブで指定します。ヘッダカスタマイズ機能を使用するための，ディレクティブの設定例を次に示します。

(a) レスポンスヘッダを設定する場合

Header ディレクティブの set 指示子によって，レスポンスヘッダを設定できます。ほかのモジュールですでに同じ名前のレスポンスヘッダが設定されている場合は，ヘッダ値を上書きします。

レスポンスヘッダに Expires: Sat, 1 Jan 2000 00:00:00 GMT を設定する例を次に示します。ただし，有効期限を動的に設定する場合は，有効期限設定機能を使用してください。

```
Header set Expires "Sat, 1 Jan 2000 00:00:00 GMT"
```

(b) レスポンスヘッダを追加する場合

Header ディレクティブの add 指示子によって，レスポンスヘッダを追加できます。ほかのモジュールですでに同じ名前のレスポンスヘッダが設定されていても，別のヘッダとして設定されます。同じ名前のレスポンスヘッダを複数行設定する場合に使用します。

レスポンスヘッダに Set-Cookie: HOSTNAME=HOST1; path=/; domain=www.example.com; secure を追加する例を次に示します。

```
Header add Set-Cookie "HOSTNAME=HOST1; path=/; domain=www.example.com; secure"
```

(3) 注意事項

- レスポンスヘッダのうち，Date，Server，Content-Type，Content-Length，Last-Modified ヘッダなどは，カスタマイズできない場合があります。また，LoadModule ディレクティブでほかのモジュールを組み込んだ場合，これら以外のヘッダでもカスタマイズできない場合があります。

4.11 有効期限設定機能

Web サーバ上のコンテンツに有効期間を設定すると，その期間中，キャッシュ機能をサポートしているクライアントやプロキシサーバは，Web サーバにアクセスしないで，自身のキャッシュにアクセスするようになるため，効率的です。

Hitachi Web Server に mod_expires モジュールを組み込むことで，有効期限設定機能を使用できます。有効期限設定機能を使用すると，次に示すことができます。

- 有効期限設定機能を使用すると，レスポンスに Expires ヘッダおよび Cache-Control ヘッダが追加されます。
- Expires ヘッダでは有効期限がグリニッジ標準時（GMT）で設定され，Cache-Control ヘッダでは max-age 指示子に，有効期限までの時間が秒単位で設定されます。

設定された Expires ヘッダおよび Cache-Control ヘッダの扱いは，クライアントやプロキシサーバに依存します。

(1) mod_expires モジュールの組み込み

有効期限設定機能を使用するためには，mod_expires モジュールの組み込みが必要です。mod_expires モジュールを組み込むには，コンフィグファイル（httpsd.conf）に次に示すディレクティブを指定します。

- UNIX 版

```
LoadModule expires_module libexec/mod_expires.so
```

- Windows 版

```
LoadModule expires_module modules/mod_expires.so
```

(2) ディレクティブの設定方法

有効期限設定機能を使用するための，ディレクティブの設定例を次に示します。

(a) デフォルトの有効期限の設定

Web サーバ上のすべてのコンテンツを対象に，ExpiresDefault ディレクティブでデフォルトの有効期限を設定します。有効期限は，ファイルの更新時刻またはクライアントがアクセスした時刻を基準にして設定します。

次のように指定すると，クライアントがアクセスした時刻から 60 秒後を有効期限として，Expires ヘッダおよび Cache-Control ヘッダがレスポンスに追加されます。

```
ExpiresActive On
ExpiresDefault A60
```

ExpiresDefault の "A" 指定は，クライアントがアクセスした時刻を基準時刻としていることを示します。

(b) MIME タイプ別の有効期限の設定

ExpiresByType ディレクティブで MIME タイプ別に有効期限を設定します。ExpiresDefault ディレクティブで設定されたデフォルトの有効期限は，この設定によって MIME タイプ別に上書きされます。有効期限は，ファイルの更新時刻またはクライアントがアクセスした時刻を基準にして設定します。

次のように指定すると，MIME タイプが text/html の場合にだけ，ファイルの更新時刻から 1 時間後を有効期限として，Expires ヘッダおよび Cache-Control ヘッダがレスポンスに追加されます。

```
ExpiresActive On
ExpiresByType text/html M3600
```

ExpiresByType の "M" 指定は，ファイルの更新時刻を基準時刻としていることを示します。

(3) 注意事項

- ファイルの更新時刻を基準時刻として設定する場合，ディスク上のファイルにアクセスしないリクエスト（ステータス情報を表示するリクエストなど）では，更新時刻が存在しないため，Expires ヘッダおよび Cache-Control ヘッダは追加されません。
- Hitachi Web Server で標準提供されていないモジュールを，LoadModule ディレクティブで組み込んだ場合，Expires ヘッダおよび Cache-Control ヘッダが操作されるおそれがあります。
- ヘッダカスタマイズ機能を同時に使用する場合，ヘッダカスタマイズ機能では Expires ヘッダおよび Cache-Control ヘッダを操作しないでください。

4.12 静的コンテンツキャッシュ機能

ディスク上に格納されている静的コンテンツファイルをメモリ上にキャッシュし，キャッシュからブラウザに返送することで，静的コンテンツのレスポンスタイムを短縮できます。

この機能は，Windows 版だけで使用できます。

(1) モジュールの組み込み

静的コンテンツキャッシュ機能を使用するには，mod_hws_cache モジュールを組み込みます。mod_hws_cache モジュールを組み込むには，コンフィグファイル（httpsd.conf）に次を指定します。

```
LoadModule hws_cache_module modules/mod_hws_cache.so
```

(2) ディレクティブの設定方法

キャッシュ機能を使用するためのハンドラ名 hws_cache を，AddHandler ディレクティブまたは SetHandler ディレクティブに指定します。

(a) 特定の拡張子のファイルをキャッシュする設定

特定の拡張子のファイルをキャッシュするには，AddHandler ディレクティブを設定します。設定方法を次に示します。

```
<Directory "C:/Program Files/Hitachi/httpsd/htdocs">
  AddHandler hws_cache .html
</Directory>
```

(b) 特定の URL 配下のファイルをキャッシュする設定

特定の URL 配下のファイルをキャッシュするには SetHandler ディレクティブを設定します。設定方法を次に示します。

```
<Directory "C:/Program Files/Hitachi/httpsd/icons">
<Files "?*">
  SetHandler hws_cache
</Files>
</Directory>
```

(3) ファイルのキャッシュ，キャッシュの更新およびキャッシュの削除の契機

(a) ファイルがキャッシュされる契機

ファイルがキャッシュされる契機を次に示します。ただし，ファイルサイズがHWSContentCacheMaxFileSize ディレクティブの設定値より大きい場合は，キャッシュされません。

- キャッシュ対象のファイルがリクエストされた場合。

(b) キャッシュが更新される契機

キャッシュが更新される契機を次に示します。ただし，ファイルサイズがHWSContentCacheMaxFileSize ディレクティブの設定値より大きい場合は，キャッシュされません。

- キャッシュされているファイルの更新日時が変更された場合。

(c) キャッシュが削除される契機

キャッシュが削除される契機を次に示します。

- キャッシュされるファイルの合計サイズが HWSContentCacheSize ディレクティブで指定された値を超えた場合（このとき，最も長い間リクエストされなかったキャッシュファイルから削除されます)。
- Web サーバを再起動した場合。

(4) メモリ使用量

静的コンテンツキャッシュ機能で使用されるメモリ量の最大値は，次の計算式で算出できます。

計算式

```
(HWSContentCacheMaxFileSize×ThreadsPerChild)＋
HWSContentCacheSize
```

(5) 注意事項

リバースプロキシ，CGI およびそのほかの動的コンテンツ処理を行う URL 配下のファイルをキャッシュ対象としないでください。キャッシュ対象を誤って指定すると，動的コンテンツが正常に処理されないおそれがあります。

使用例を次に示します。

```
ScriptAlias /cgi-bin/ "C:/Program Files/Hitachi/httpsd/cgi-bin/"
                                                    ＊1
LoadModule hws_cache_module modules/mod_hws_cache.so
<Directory "C:/Program Files/Hitachi/httpsd/icons/">
                                        ＊2
<Files "?*">
  SetHandler hws_cache
</Files>
</Directory>
```

注 下線部＊1と＊2で設定されるパスを重複させないでください。
＊2を "C:/Program Files/Hitachi/httpsd/cgi-bin/" と設定した場合には，CGI プログラムが静的ファイルとして応答され，CGI プログラムの内容が漏洩するなど，予期できない結果となるおそれがあります。

4.13 複数の Web サーバ環境の生成（hwsserveredit ユティリティ）

バーチャルホストでなく，1 台のサーバマシンで複数の Web サーバを運用する場合，各 Web サーバで httpsd.conf の準備などの環境設定が必要です。この環境設定を補助するのが hwsserveredit ユティリティです。

（1） 形式

```
hwsserveredit {-add|-delete|-check} サーバ名
```

（2） オペランド

● -add

サーバ環境を新規作成する場合に指定します。ユティリティは要求を受け付けると，servers ディレクトリの下に，指定されたサーバ名と同名のディレクトリを作成し，httpsd.conf と logs ディレクトリを作成します。また，Windows 版の場合は，サーバ名を用いて Hitachi Web Server をサービスとして登録します。

● -delete

サーバ環境を削除する場合に指定します。ユティリティは要求を受け付けると，servers ディレクトリ下のサーバ名と同名のディレクトリを削除します。また，Windows 版の場合は，サーバ名のサービスを削除します。

● -check

サーバ環境が構築済みかどうかを確認する場合に指定します。ユティリティは要求を受け付けると，-add オペランド要求時に作成したリソースがある場合は，サーバ起動環境構築済みと判断します。

● **サーバ名**　～ ((1 － (220 － Hitachi Web Server インストールディレクトリのパス長)，かつ 128 バイト以下))

サーバ単位にユニークな文字列を指定します。ただし，"Hitachi Web Server" は指定できません。また，文字列中の空白を取り除くと，"HitachiWebServer" となる文字列（"Hitachi WebServer" など）は指定できません。

（3） 使用方法

(a) リソースの作成

各サーバのサーバ名を決定してから，hwsserveredit ユティリティを実行します。
"HWS1" というサーバ名の場合は，次を指定します。

```
hwsserveredit -add HWS1
```

hwsserveredit ユティリティは，インストール時に作成されている servers ディレクトリの下に，ディレクトリとファイルを作成します。また，Windows 版の場合，ディレクトリとファイルの作成と同時に，サーバ名を使用してサービスを登録します。

次にディレクトリとファイルの構成を示します。

```
+httpsd
    └servers
        └HWS1
           ├conf
           │  └httpsd.conf
           └logs
```

(b) httpsd.conf の編集

作成されたリソースのうち，httpsd.conf のディレクティブ値を変更してください。

複数環境を生成する場合には，次のディレクティブの設定値を，ほかの環境と競合しないように変更してください。

- ServerName ディレクティブ
- Port ディレクティブまたは Listen ディレクティブ

UNIX 版の場合は，さらに次のディレクティブの設定値を，環境に合わせて変更してください。

- User ディレクティブ
- Group ディレクティブ

そのほかのディレクティブは，hwsserveredit ユティリティによって，変更しなくても起動できるように設定されています。運用に応じて必要があれば変更してください。

なお，各ディレクティブの詳細については，「6.2　ディレクティブの詳細」を参照してください。

(c) サーバの起動

次の方法でサーバを起動してください。

- UNIX 版の場合
  `/opt/hitachi/httpsd/sbin/httpsd -f servers/HWS1/conf/httpsd.conf`
- Windows 版の場合
  `"インストール先ディレクトリ¥httpsd.exe" -n HWS1 -k start`

  または，コントロールパネルから HWS1 サービスを起動してください。

(d) 複数サーバ環境の設定

複数サーバ環境を設定する場合は，(a) から (c) の操作を繰り返してください。

(4) 注意事項

- ユティリティは，管理者権限を持つユーザで実行してください。また，ユティリティの場所は移動しないでください。
- サーバ名には，ASCII コードで指定してください。また，次に示す文字は指定できません。
  '¥'，'/'，':'，','，';'，'*'，'?'，'"'，'<'，'>'，'|'，'$'，'%'，'^'，'`'，'!'，'('，')'，'='，'+'，'{'，'}'，'@'，'['，']'，'~'，制御コード
- サーバ名には，ピリオドだけで構成される名称は指定できません。また，名称の前後に連続した空白を指定した場合には，それらを取り除きます。
- サービスへの登録時，スタートアップの種類は手動で登録します。
- 登録したサービスの表示名を変更しないでください。

4.14 イメージマップ

画像（画像のファイル）に複数のリンクを定義できます。その指定個所をクリックすると，その画像の座標位置やイメージマップファイル名が Web サーバに Web ブラウザから送信されます。Web サーバは，そのイメージマップファイルと座標位置から対応した URL を検索して，Web ブラウザに応答します。これをイメージマップといいます。

イメージマップを使用するには，imap-file ハンドラにマップファイル拡張子を対応付ける定義が必要です。

```
AddHandler imap-file .map
```

（1）イメージマップファイルの文法

イメージマップデータの指定形式には次の 3 とおりの指定があります。

```
形状名称 指定値 座標
形状名称 指定値 " 説明文 " 座標
形状名称 指定値 座標 " 説明文 "
```

" **説明文** " はマップファイルメニュー表示時の説明文，**座標**は画像の座標を示しています。

形状名称を表 4-15 に，**指定値**を表 4-16 に示します。

表 4-15　形状名称と座標の指定形式

形状名称	意味	座標の指定	座標の説明
base	マップファイル内の相対 URL のベースを指定。	なし	－
default	poly，circle，rect に該当しないで，point 指定もない場合のリンクを指定する。		
poly	多角形を指定する。点を 3 ～ 100 個指定する。	x1,y1 x2,y2 … xn,yn	多角形の各座標位置（3 ～ 100 点の座標）
circle	円を指定する。中心点と，円周上の 1 点を指定する。	x1,y1 x2,y2	中心座標と円周上の 1 点の座標
rect	四角形を指定する。対角の 2 点を指定する。	x1,y1 x2,y2	対角の 2 点の座標
point	点を指定する。カーソルに最も近い point が有効になる。	x1,y1	点

（凡例）

－：該当しない。

注

座標の指定で (0,0) を含んでいる場合でも，イメージマップ画像の座標 (0,0) をマウスポインタでポイントすると，マップファイルメニューが表示されます。

表 4-16 指定値

指定値	意味
URL	リンク先を指定する。相対ディレクトリの場合，base，ImapBase ディレクティブが有効になる。
map	マップファイルメニューを表示する。
menu	
referer	ステータスコード 302 Found を応答する。
nocontent	ステータスコード 204 No Content を応答する。base 以外に有効。
error	ステータスコード 500 Server Error を応答する。base 以外に有効。

(2) イメージマップの定義例

イメージマップを利用するための操作を次に示します。

1. httpsd.conf ファイルに，次に示すディレクティブを設定します。.map 拡張子名が URL で指定されたときにイメージマップを実行します。

   ```
   AddHandler imap-file .map
   ```

   （ファイル拡張子 .map に imap-file ハンドラを定義）
2. 上記で定義されたファイル拡張子のファイルにリンク先を定義します。
3. HTML 文書内に次の HTML 構文を記述します。

   ```
   <A HREF="/ディレクトリ名/マップファイル名"><IMG SRC="画像データ名"
   ISMAP></A>
   ```

イメージマップファイルの定義例や，実際の表示例などを次に示します。

図 4-13　イメージマップファイルの定義例

マップファイルの定義内容

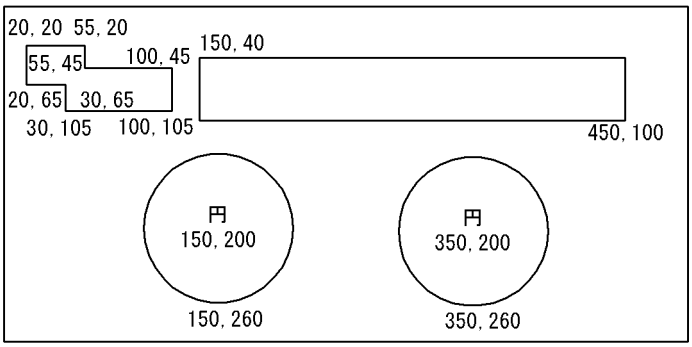
```
default /
poly map "Map menu" 20,20 55,20 55,45 100,45 100,105 30,105 30,65 20,65
rect http://www.hitachi.co.jp/ 150,40 450,100
circle http://www.hitachi.co.jp/Prod/comp/ "地点１" 150,200 150,260
circle http://www.hitachi.co.jp/Prod/comp/soft1/ 350,200 350,260 "地点２"
```

イメージマップファイル

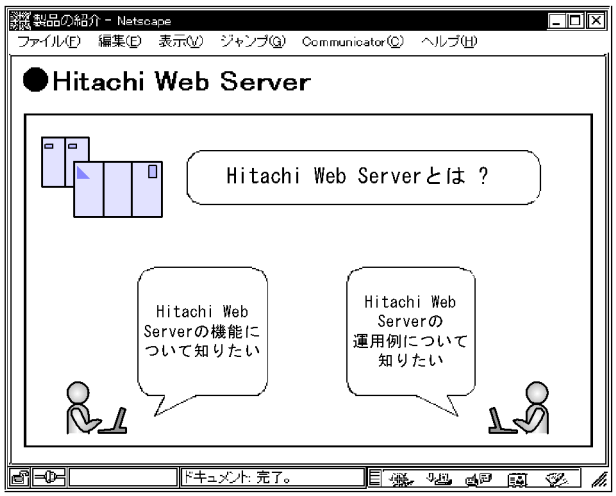
htmlファイルの定義内容

```
<A HREF="/maps/imagemap1.map">
    <IMG ISMAP SRC="/images/imagemap1.gif">
</A>
```

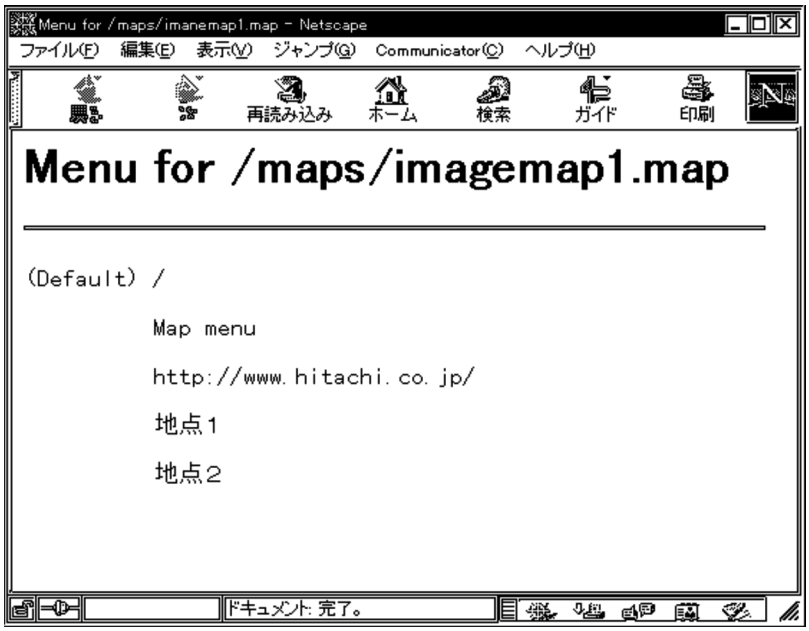
実際のマップの定義



対応



この例で，poly で指定された部分をクリックすると，次に示すようなマップファイルメニューが表示されます。



(3) 注意事項

マップファイルメニューで使用している文字セットが，デフォルトの文字セット (ISO-8859-1) と異なる場合は，マップファイルメニューの表示において文字化けが発生します。この場合，HWSImapMenuCharset ディレクティブで，マップファイルメニューで使用している文字セットを指定してください。

4.15 IPv6 による通信

従来の IPv4 による通信だけでなく，IPv6 による通信ができます。IPv4 または IPv4 と IPv6 のデュアルスタック環境で動作させることができます。

この機能は，Linux（32 ビット）および Windows 版で使用できます。

AIX，Solaris，HP-UX（IPF）および Linux（IPF）版では使用できません。

4.15.1 サポート範囲

（1）IPv6 に対応しているディレクティブ

Listen ディレクティブや VirtualHost ディレクティブなどのディレクティブに，IPv6 アドレスを指定することによって，IPv6 による通信や IPv6 アドレスに対応したバーチャルホストの指定などができます。

IPv6 に対応しているディレクティブを次に示します。

<VirtualHost>，AddIcon，AddIconByEncoding，AddIconByType，Allow from，CustomLog，DefaultIcon，Deny from，ErrorDocument，ExtendedStatus，HostnameLookups，HWSSetEnvIfIPv6，ImapBase，ImapDefault，Listen，LogFormat，NameVirtualHost，ProxyPass，ProxyPassReverse，QOSCookieDomain，QOSRedirect，Redirect，RedirectMatch，ServerAlias，ServerName，ServerSignature，SetEnvIf，SetEnvIfNoCase，TransferLog，UseCanonicalName

各ディレクティブの詳細については，「6.2 ディレクティブの詳細」を参照してください。

（2）ディレクティブに IPv6 アドレスを指定するときの注意

ディレクティブに IPv6 アドレスを記述する場合は，「[IPv6 アドレス]」のように IPv6 アドレスを [ ] で囲んで指定してください。また，ディレクティブに IPv6 アドレスとポート番号を併記する場合は，「[IPv6 アドレス]: ポート番号」のように IPv6 アドレスを [ ] で囲み，「:」の後ろにポート番号を指定します。

ただし，次のディレクティブに IPv6 アドレスを記述する場合は，IPv6 アドレスを [ ] で囲まないで指定してください。

- Allow from ディレクティブ
- Deny from ディレクティブ
- HWSSetEnvIfIPv6 ディレクティブ

なお，IPv6 アドレスを指定する場合は，リンクローカルアドレスを指定できません。サ

イトローカルアドレスまたはグローバルアドレスを指定してください。

(3) 制限事項

次の機能については，IPv6 には対応していません。

SSL 通信の一部

SSLCacheServerPort ディレクティブにポート番号を指定している場合，Web サーバ本体と gcache サーバとの間は，IPv4 による通信を行います。IPv6 による通信はできません。

なお，IPv6 ソケットを使用した SSL 通信はできます。

ディレクトリサービス（LDAP サーバ）

LDAPServerName ディレクティブには，IPv6 アドレスまたは IPv6 アドレスに対応したホスト名は指定できません。

アドレス制限

BindAddress ディレクティブには，IPv6 アドレスは指定できません。

クライアントの確認

IdentityCheck ディレクティブに On を指定しても，IPv6 ソケットを使用している場合は機能しません。

環境変数の設定

SetEnvIf ディレクティブと SetEnvIfNoCase ディレクティブの正規表現には，IPv6 アドレスを指定できません。IPv6 アドレスを指定する場合は，HWSSetEnvIfIPv6 ディレクティブを使用してください。

crldownload ユティリティ

「-h ホスト名」には，IPv6 アドレスまたは IPv6 アドレスに対応したホスト名は指定できません。

4.15.2 IPv6 による通信の準備（httpsd.conf ファイルの編集）

IPv6 アドレスの指定

httpsd.conf ファイルに Port ディレクティブまたはポート番号だけの Listen ディレクティブを指定した場合は，IPv4 アドレスを使用したリクエストだけを受け付けます。IPv6 アドレスを使用する場合は，IPv6 アドレスを指定した Listen ディレクティブの設定が必要です。

例えば，次のように設定すると，IPv4 アドレスまたは IPv6 アドレスを使用したリクエストを受け付けるようになります。

```
Listen 80
Listen [::]:80
```

5 SSL による認証，暗号化

この章では，SSL による認証，暗号化について説明します。

5.1 SSL で認証，暗号化する

Hitachi Web Server は SSL（Secure Sockets Layer）プロトコルを使用すれば，送受信する情報を保全できます。Hitachi Web Server は SSL バージョン 2，バージョン 3 および TLS（Transport Layer Security）バージョン 1 に対応しており，SSL サーバ認証，SSL クライアント認証ができます。SSL の機能を次に示します。

- 通信相手を確認し，特定するために認証します。
- サーバとクライアントの間で転送するデータを暗号化します。
- 転送中に改ざんされたデータを検出します。

これらの機能は SSL 関連ユティリティで作成された秘密鍵と，認証局（CA）が発行した証明書を Web サーバにインストールすれば，利用できます。

5.1.1 SSL 通信のための準備

SSL による認証や，データの暗号化を使用するには，Web サーバに秘密鍵と認証局（CA）が発行した証明書をインストールする必要があります。

手順を次に示します。

1. 秘密鍵の作成
   sslckey genrsa ユティリティを使用して，Web サーバの秘密鍵を作成します。
2. CSR（証明書発行要求）の作成
   sslccert req ユティリティを使用して，CSR を作成します。
3. CA へ CSR を送付
   2. で作成した CSR を CA に送付します。
4. 証明書の入手
   PEM 形式の証明書を CA から入手します。
5. httpsd.conf ファイルの編集（ディレクティブの定義）
   SSL を有効にするために，SSLEnable ディレクティブを指定します。CA から入手した PEM 形式の証明書は SSLCertificateFile ディレクティブ，Web サーバの秘密鍵は SSLCertificateKeyFile ディレクティブに指定します。

   （例）SSL を有効にして，PEM 形式の証明書および Web サーバの秘密鍵を定義
   - UNIX 版の場合

```
SSLEnable
SSLCertificateFile /opt/hitachi/httpsd/conf/ssl/server/httpsd.pem
SSLCertificateKeyFile /opt/hitachi/httpsd/conf/ssl/server/httpsdkey.pem
```

   - Windows 版の場合

```
SSLEnable
SSLCertificateFile "C:/Program Files/Hitachi/httpsd/conf/ssl/server/
httpsd.pem"
SSLCertificateKeyFile "C:/Program Files/Hitachi/httpsd/conf/ssl/server/
httpsdkey.pem"
```

SSL を使用して通信する場合，Web ブラウザからは，https:// でリクエストします。ポート番号を省略した場合，SSL の標準では 443 ポートを使用します。したがって，Port または Listen ディレクティブで 443 ポートを指定するのが一般的です。

6. Web サーバの再起動
   httpsd.conf ファイルの定義を有効にするには，Web サーバを再起動する必要があります。ただし，SSLCertificateKeyFile ディレクティブの設定を変更した場合は，いったん，Web サーバを停止後，起動し直してください。

SSL を無効にする場合には，5. の指定を無効化し，SSLDisable ディレクティブを指定して再起動します。

（1） Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel64)，Red HatEnterprise Linux 5 (AMD/Intel64) で SSL を使用する場合

(a) Hitachi Web Server の起動方法

以下に示すどちらかの方法で Hitachi Web Server を起動してください。

- オプション「-D HWS_OPTION_HWS2」を指定する方法
  Hitachi Web Server の起動時に，オプション「-D HWS_OPTION_HWS2」の指定を追加します。
  /opt/hitachi/httpsd/sbin/httpsd 〔HWS オプション〕〔-D HWS_OPTION_HWS2〕
  HWS オプション：マニュアル記載されている httpsd に指定可能なオプション
  -D HWS_OPTION_HWS2：Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel64) または Red Hat Enterprise Linux 5 (AMD/Intel64) で SSL を使用するための専用オプション
- 環境変数「HWS_OPTION_HWS2」を設定する方法
  Hitachi Web Server を起動するシェルの環境変数として「HWS_OPTION_HWS2=1」が指定されている場合，SSL を使用することができます。環境変数「HWS_OPTION_HWS2」が指定されていない場合，または「HWS_OPTION_HWS2」に 1 以外の値が指定されている場合は，SSL は使用できません。

(b) 起動確認方法

5.1.1(1) に示す方法で Hitachi Web Server を起動した後，エラーログに以下のメッセージが出力されていることを確認してください。

［時刻情報］［notice］Hitachi Web Server configured --resuming normal operations

［時刻情報］［notice］Server built: サーバのビルド時刻

［時刻情報］［notice］Parent 制御プロセス ID:Using config file " コンフィグファイル名 "

［時刻情報］［notice］Special option:HWS2

(c) ユティリティ

以下の標準ユティリティの代わりに，変更後ユティリティを実行してください。マニュアル内の標準ユティリティに関する記載は，変更後ユティリティに関する記載として読み替えてください。

標準ユティリティ	変更後ユティリティ
sslpasswd	sslpasswd2
sslckey	sslckey2
sslccert	sslccert2

標準ユティリティと変更後ユティリティを組み合わせて使用することはできません。例えば sslckey ユティリティで作成した鍵ファイルを，sslccert2 ユティリティの引数に使用することはできません。

(d) 注意事項

- 対象となるプラットフォームは Red Hat Enterprise Linux 5 Advanced Platform(AMD/Intel64)，または Red Hat Enterprise Linux 5 (AMD/Intel64) のみです。Red HatEnterprise Linux 5 Advanced Platform (x86) または Red Hat Enterprise Linux 5(x86) では SSL は使用できません。
- Red Hat Enterprise Linux 4 Advanced Platform (AMD/Intel64)，または Red HatEnterprise Linux 4 (AMD/Intel64) で SSL を使用する場合は，「5.1.1(1)Red HatEnterprise Linux 5 Advanced Platform (AMD/Intel64)，Red Hat EnterpriseLinux 5 (AMD/Intel64) で SSL を使用する場合」に記載された内容を実行しないでください。通常の起動方法で SSL を使用することができます。
- httpsdctl ユティリティを用いて再起動・停止処理を実施する場合は，httpsdctl を実行するシェルの環境変数に「HWS_OPTION_HWS2=1」を設定してください。

5.1.2 SSL 通信の手順

SSL 通信の手順を次に示します。2 ～ 6 の手順を SSL での**ハンドシェイク**と呼びます。

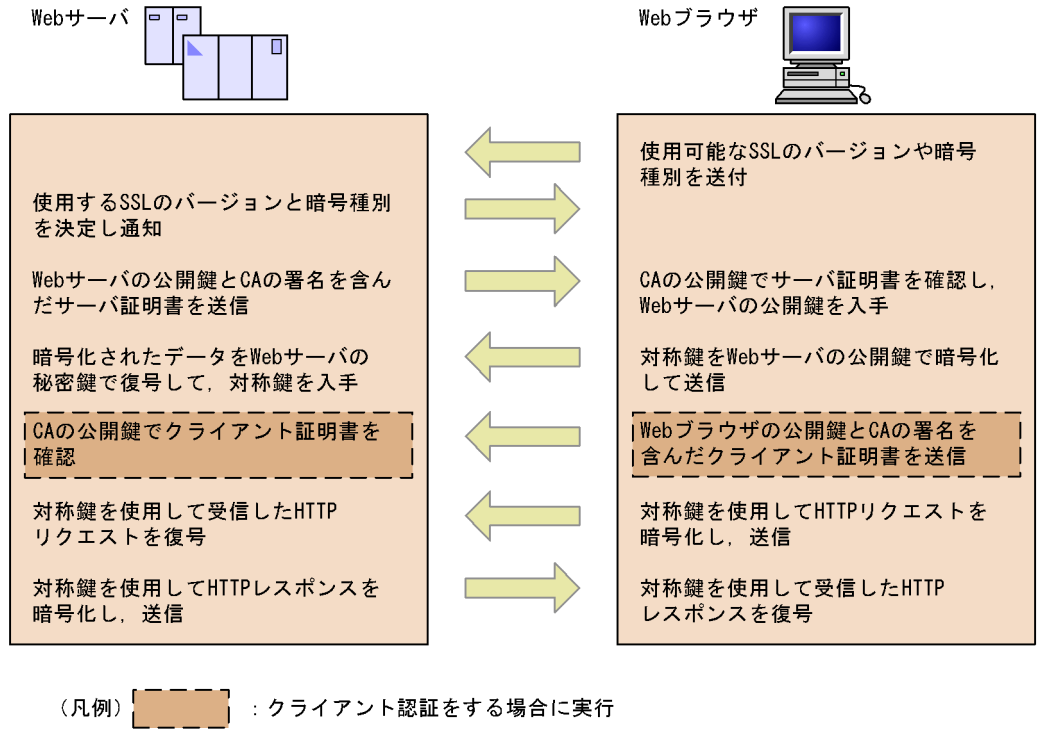
1. Web ブラウザから，https:// へのリクエストを実行します。
2. Web ブラウザは，使用できる SSL のバージョンや暗号種別を示すデータを Web サーバに送付します。
3. Web サーバは，使用する SSL のバージョンと暗号種別を決定し，Web ブラウザに通

知します。また，公開鍵と CA の署名が入った証明書を Web ブラウザに送付します。

4. Web ブラウザは Web ブラウザが持っている CA の公開鍵を使用して，送付された証明書が改ざんされていないことを確認して，Web サーバの公開鍵を入手します。
5. Web ブラウザは，通信で Web サーバと共有する対称鍵を作成し，Web サーバの公開鍵で暗号化して送信します。また，Web ブラウザが持っている証明書をクライアント認証のために Web サーバに提示する場合，証明書を送信します。
   Web サーバの公開鍵で暗号化したデータは，対になる秘密鍵がないと復号できません。つまり，データ送信先の Web サーバだけがデータの内容を解読できます。
6. Web サーバは受信した対称鍵を Web サーバの秘密鍵で復号し，入手します。Web ブラウザからの証明書を受信した場合は，証明書を確認します。
7. Web ブラウザと，Web サーバの間で共有された対称鍵を使用して，HTTP リクエストまたはレスポンスを暗号化し，送受信します。

SSL 通信でのリクエスト処理を次に示します。

図 5-1　SSL 通信でのリクエスト処理



5.1.3　SSL での暗号強度について

SSL では，ハンドシェイクの際に，クライアント側と Web サーバ側の両方に有効であり，最も強い強度の暗号が選択されます。Web サーバ側の暗号種別は，

SSLRequiredCiphers ディレクティブで指定します。この指定で，常にすべての暗号種別を有効にしておけば，クライアントが持つ最も強い強度の暗号で通信できることになります。

Hitachi Web Server は，128bit RC4，256bit AES などの強い暗号強度をサポートしています。強い暗号強度対応の SSL サーバ証明書も使用できます。

5.1.4 SSL セション管理

現在多くの Web ブラウザは，SSL セションを用いてハンドシェイクを簡略化する機能を実装しています。ブラウザ側のこの機能と Web サーバ側の SSL セション管理機能を用いることで，SSL による通信性能の向上を図れます。

UNIX 版と Windows 版では，SSL セションを管理する方法が異なります。UNIX 版では，SSL セションを管理するサーバ（**gcache サーバ**といいます）を使用します。gcache サーバでは，指定したポートまたはファイルを通じて，SSL セション ID，有効期限およびそのセションについての情報を受信して，管理します。gcache サーバによって，Hitachi Web Server のリクエスト処理プロセス間で SSL セション ID などのデータを共有できます。Windows 版では，Web サーバの構造上，gcache サーバを使用しないで SSL セションを管理します。

(1) UNIX 版の場合

(a) gcache サーバの起動・停止

SSL が有効な状態にし，かつ gcache サーバを起動するために必要なディレクティブを指定して Hitachi Web Server を起動すると，gcache サーバが起動されます。SSL が有効な状態とは，SSLEnable ディレクティブを指定しているかまたは SSLDisable ディレクティブを指定していないホスト（バーチャルホストを含めて）がある状態のことです。

gcache サーバを起動するには，次のディレクティブの指定が必要です。

- SSLCacheServerPath
- SSLCacheServerPort

また，Web サーバを停止すると，同時に gcache サーバも停止します。Web サーバを再起動すると gcache サーバはいったん停止し，その後，起動します。

(b) セション管理領域

SSL のセションを確立すると，そのセション情報は gcache サーバと Web サーバプロセス内にキャッシュされます。gcache サーバのキャッシュ領域のサイズは SSLSessionCacheSize ディレクティブ，Web サーバプロセス内キャッシュは SSLSessionCacheSizePerChild ディレクティブで指定できます。

SSLSessionCacheSize ディレクティブに 0 を指定した場合，SSL セション管理は機能し

ません。

SSL セションの有効時間は SSLSessionCacheTimeout ディレクティブの指定値かまたはキャッシュサイズが SSLSessionCacheSize ディレクティブで指定したキャッシュサイズに達するまでの時間のどちらか短い方になります。

キャッシュサイズが SSLSessionCacheSize ディレクティブで指定したキャッシュサイズに達した場合，新しいセション情報を保持するメモリ領域が確保できるまで，古いセション情報から順に削除されます。

キャッシュされたセション情報は次回以降のセション確立の際に再利用することで，SSL のハンドシェイクを簡略化します。

(c) 注意事項

1. セション ID などのデータは，Web サーバまたは gcache サーバが停止するとクリアされます。
2. gcache サーバが異常停止した場合には，SSL セションは維持できません。ただし，Web サービスは停止しません。
3. セキュリティの関係上，Web サーバ間で gcache サーバの共用はできません。そのため，Web サーバ間の SSL セション共有はできません。
4. SSLSessionCacheSize ディレクティブに 0 を指定した場合，gcache サーバは起動しません。
5. セションキャッシュ領域はバーチャルホストごとに分かれません。

(2) Windows 版の場合

(a) セション管理領域

SSL のセションを確立すると，そのセション情報は Web サーバプロセス内にキャッシュされます。このキャッシュ領域のサイズは SSLSessionCacheSize ディレクティブで指定できます。SSLSessionCacheSize ディレクティブに 0 を指定した場合，SSL セション管理は機能しません。

SSL セションの有効時間は SSLSessionCacheTimeout ディレクティブの指定値かまたはキャッシュサイズが SSLSessionCacheSize ディレクティブで指定したキャッシュサイズに達するまでの時間のどちらか短い方になります。

キャッシュされたセション情報は次回以降のセション確立の際に再利用することで，SSL のハンドシェイクを簡略化します。

(b) 注意事項

- セション ID などのデータは，Web サーバを停止するとクリアされます。
- セションキャッシュ領域はバーチャルホストごとに分かれません。

5.1.5 SSL クライアント認証の準備

SSL クライアント認証をする場合は，「5.1.1 SSL 通信のための準備」の 1 ～ 5 に加えて，次の操作をして，再起動してください。

1. クライアント証明書を Web ブラウザにインストール
   証明書の発行に使用する CA の指示に従い，Web ブラウザにクライアント証明書をインストールします。
2. CA の証明書の入手
   クライアント証明書を発行した CA の証明書（PEM 形式）を入手します。
3. httpsd.conf ファイルの編集（ディレクティブの定義）
   SSL クライアント認証を有効にするために，SSLVerifyClient に 2 を指定し，SSLVerifyDepth ディレクティブに 2 以上を指定します。CA から入手した PEM 形式の証明書は，SSLCACertificateFile または SSLCACertificatePath ディレクティブに指定します。

   注
   SSLCACertificatePath ディレクティブは Windows 版では指定できません。

5.1.6 証明書の有効性の検証

SSL クライアント認証のときに，クライアント証明書の検証だけでなく，その時点のクライアント証明書の有効性を CRL（Certificate Revocation List）を使用して検証できます。CRL は，検証したいクライアント証明書を発行した CA から入手します。

(1) CRL ファイルの形式

CRL は DER 形式または PEM 形式のファイルを使用します。DER 形式の CRL はバイナリ形式のファイルで，PEM 形式の CRL はそれを Base64 エンコード処理し，データの前後に，"-----BEGIN X509 CRL-----"，"-----END X509 CRL-----" というタグを付けたものです。

（例）PEM 形式の CRL

```
C:¥Program Files¥Hitachi¥httpsd¥conf¥ssl¥crl¥PEM>type crl.pem
-----BEGIN X509 CRL-----
MIIBGDCBwwIBATANBgkqhkiG9w0BAQQFADB0MQswCQYDVQQGEwJKUDERMA8GA1UE
CBMIS2FuYWdhd2ExFTATBgNVBAcTDFlva29oYW1hLXNoaTERMA8GA1UEChMITE9D
QUwtQ0ExDDAKBgNVBAsTA2NhMTEaMBgGA1UEAxMRY2ExLmhpdGFjaGkuY28uanAX
DTAxMDgyOTA0NDIzMFoXDTAxMDgzMDA1NTIzMFowGzAZAghx2Sa8AAAAARcNMDEw
ODI4MDQ1MTI5WjANBgkqhkiG9w0BAQQFAANBAJorY7DUJ91uthNlAA+PT6zw6rVo
uZLFeYZPNVXgF217YOCtJtKDT+16bR5kgk0p/1xIbgReshjMNTmXPqARNjE=
-----END X509 CRL-----
```

（2） Hitachi Web Server への CRL 適用方法

CRL を使用してクライアント証明書の有効性を検証する場合は，「5.1.5　SSL クライアント認証の準備」に加えて，次の操作をして，Web サーバを再起動してください。

1. CRL の入手
   各 CA の CRL 配布点から CRL ファイルを入手し，適切なディレクトリに格納します。CRL を LDAP サーバで管理する場合は，crldownload ユティリティを利用できます。
2. httpsd.conf の編集（ディレクティブの定義）
   CRL を有効にするために，CRL ファイルを格納したディレクトリを SSLCRLDERPath または SSLCRLPEMPath ディレクティブに設定します。
3. Web サーバを起動または再起動します。
4. 既存の CRL を更新する場合には，ディレクトリに格納されている古い CRL を削除して新規 CRL を追加するかまたは古い CRL を新規 CRL で上書きした後，Web サーバを再起動します。
5. CRL を新規に追加した場合，CRL を削除した場合にも，Web サーバを再起動してください。

（3） CRL を使用したクライアント証明書検証

CRL を使用したクライアント証明書検証では次の項目を確認します。

- CRL 自体が有効であるかどうか。
- 次回発行日より前かどうか。
- クライアント証明書のシリアル番号が記載されていないかどうか。

(a) CRL クライアント証明書検証でクライアント証明書が有効と判定される条件

CRL クライアント証明書検証でクライアント証明書が有効と判定される条件には次に示すものがあります。

- 証明書を発行した CA が発行した CRL を読み込んでいない場合。
- 現在時刻が CRL の次回発行日より前であり，かつ該当する接続のクライアント証明書のシリアル番号が CRL に記載されていない場合。
- 現在時刻が CRL 発行日より後で，次回発行日が指定されてなく，かつ CRL に該当する接続のクライアント証明書のシリアル番号が記載されていない場合。
- 現在時刻が CRL の次回発行日以後で，CRL に該当する接続のクライアント証明書のシリアル番号が記載されてなく，かつ SSLCRLAuthoritative ディレクティブが Off に設定されている場合。

(b) CRL クライアント証明書検証でクライアント証明書が無効と判定される条件

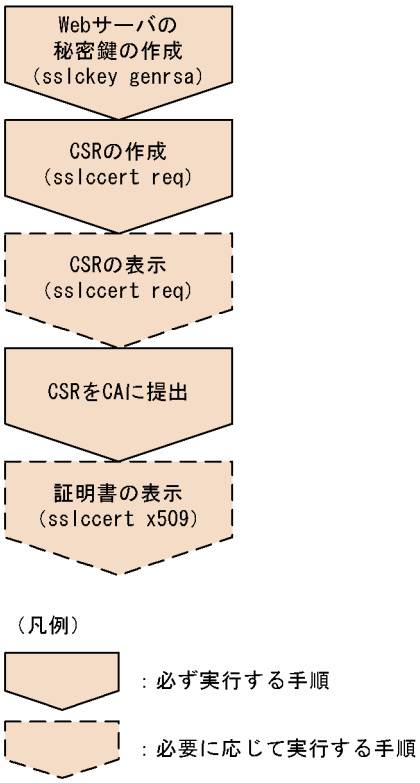
CRL クライアント証明書検証でクライアント証明書が無効と判定される条件には次に示すものがあります。

- CRL が有効でない場合。
- 該当する接続のクライアント証明書のシリアル番号が CRL に記載されている場合。
- 現在時刻が CRL 次回発行日以後であり，該当する接続のクライアント証明書のシリアル番号が CRL に記載されてなく，かつ SSLCRLAuthoritative ディレクティブが On に設定されている場合。

5.2 証明書取得手順

証明書を取得するまでの手順を次に示します。

図 5-2 証明書取得手順



上の図の手順で CA の署名済みの証明書ファイルを取得したあと，証明書ファイルの "-----BEGIN CERTIFICATE-----" から，"-----END CERTIFICATE----" の部分を別ファイルに保存します（標準提供の httpsd.conf では httpsd.pem）。このファイルを SSLCertificateFile ディレクティブに定義することで，SSL が利用できるようになります。

- sslckey
  SSL を利用する場合の証明書取得に必要な Web サーバの秘密鍵を作成するユティリティ
- sslccert
  SSL を利用する場合の証明書取得に必要な証明書発行要求 (CSR) を作成するユティリティ

5.2.1 Web サーバの秘密鍵の作成

sslckey genrsa ユティリティを使用して，Web サーバの秘密鍵を作成します。作成した Web サーバの秘密鍵のファイルは，SSLCertificateKeyFile ディレクティブに指定します。

(1) 形式

```
sslckey genrsa -rand ファイル名〔:ファイル名…〕 〔-des｜-des3〕-out 鍵ファイル 〔512 ｜1024｜2048〕
```

(2) オペランド

- -rand **ファイル名〔: ファイル名…〕**
  乱数生成に利用する任意のファイルを指定します。乱数生成用のファイルは，十分大きい適当なファイル（例：C:¥WINNT¥NOTEPAD.EXE）を指定してください。
  Windows 版では，ファイル名は一つだけ指定できます。複数指定はできません。
- **〔**-des ｜ -des3**〕**
  秘密鍵を暗号化する場合，暗号種別を指定します。このオペランドを指定すると，秘密鍵作成時にパスワードの入力要求があります。パスワードは 128 文字以内です。
  また，証明書発行要求（CSR）の作成時（「5.2.2　証明書発行要求（CSR）の作成」参照）および Web サーバ起動時にパスワードの入力要求があります。なお，Web サーバ起動時のパスワードの入力は省略できます（「5.4　パスワード付きサーバ秘密鍵の使用」参照）。-des を指定した場合，暗号種別として DES(Data Encryption Standard) が選択されます。-des3 を指定した場合，トリプル DES が選択されます。
  Web サーバと Web ブラウザ間の通信での暗号種別とは関係ありません。
- -out　**鍵ファイル**
  Web サーバの秘密鍵を出力するファイルを指定します。
- **〔**512 ｜ 1024 ｜ 2048**〕**
  作成する Web サーバの秘密鍵のビット長を指定します。

(3) 使用例

Web サーバの秘密鍵 httpsdkey.pem を作成します。

```
sslckey genrsa -rand file1:file2:file3:file4:file5 -out demoCA/httpsdkey.pem
1024
```

file1，file2，file3，file4，file5：任意のファイル

5.2.2 証明書発行要求（CSR）の作成

sslccert req ユティリティを使用して，証明書発行要求（CSR）を作成します。ここで作成した CSR ファイルを CA に提出して，署名済みの証明書を発行してもらいます。CSR は，PKCS#10 に準拠した形式で作成されます。

（1） 形式

```
sslccert req -config コンフィグファイル -new 〔-MD5｜-SHA1〕 -key 鍵ファイル -out CSR ファイル
```

（2） オペランド

- -config **コンフィグファイル**
  sslc.cnf ファイルを指定します。sslccert req ユティリティ実行時に Country Name（国名コード），Locality Name（市町村名）などの情報の入力要求があります。sslc.cnf ファイルの定義内容を使用する場合は入力する必要はなく，そのまま Enter キーを押してください。sslc.cnf ファイルに定義されていない場合または sslc.cnf ファイルの定義内容とは異なる指定をする場合は，入力要求時に指定してください。また，Country Name（国名コード），Locality Name（市町村名）などの情報を指定しない場合は，"."（ピリオド）を 1 文字だけ入力し，Enter キーを押してください。
- 〔-MD5 ｜ -SHA1〕
  CSR 作成時に使用する署名アルゴリズムを指定します。
  -MD5 ：md5WithRSAEncryption を使用します。
  -SHA1：sha1WithRSAEncryption を使用します。
- -key **鍵ファイル**
  Web サーバの秘密鍵のファイルを指定します。
- -out CSR **ファイル**
  作成した CSR を出力するファイルを指定します。

（3） 使用例

Web サーバの秘密鍵 httpsdkey.pem を作成します。

```
sslccert req -config demoCA/sslc.cnf -new -SHA1 -key demoCA/httpsdkey.pem -out demoCA/httpsd.csr
```

`demoCA/httpsdkey.pem`：鍵ファイル

`demoCA/httpsd.csr`：CSR ファイル

（4） 注意事項

Web サーバの名称とテスト用 CA の名称を同じものにすると，ブラウザのセキュリティ

チェックによって正しい SSL 接続ができません。CSR 作成時とテスト用 CA の証明書作成時の Organization Name，Organization UnitName，Common Name などは，異なるフレーズを設定してください。

5.2.3 証明書発行要求（CSR）の内容表示

証明書発行要求（CSR）の内容を表示します。

(1) 形式

```
sslccert req -in CSRファイル -text
```

(2) オペランド

- -in CSR **ファイル**
  表示する CSR ファイルを指定します。

(3) 使用例

```
sslccert req -in demoCA/httpsd.csr -text
```

`demoCA/httpsd.csr`：表示する CSR ファイル

5.2.4 証明書の内容表示

証明書ファイルの内容を表示します。

"-----BEGIN CERTIFICATE-----" から，"-----END CERTIFICATE----" の証明書ファイルの内容を表示します。

(1) 形式

```
sslccert x509 -in 証明書ファイル -text
```

(2) オペランド

- -in **証明書ファイル**
  表示する証明書ファイルを指定します。

(3) 使用例

```
sslccert x509 -in demoCA/httpsd.pem -text
```

demoCA/httpsd.pem：表示する証明書ファイル

5.2.5 証明書の形式変換

証明書の形式を変換します。必要に応じて使用します。

（1） 形式

```
sslccert x509 -inform 入力形式 -outform 出力形式 -in 入力ファイル -out 出力ファイル
```

（2） オペランド

- -inform **入力形式**
  入力形式：{DER ｜ PEM}
- -outform **出力形式**
  出力形式：{DER ｜ PEM}
- -in **入力ファイル**
  変換前の証明書ファイルを指定します。
- -out **出力ファイル**
  変換後の証明書ファイルを指定します。

5.2.6 ハッシュリンクの作成（UNIX 版）

証明書の妥当性チェックのために，証明書を発行した CA の証明書を SSLCACertificateFile ディレクティブまたは SSLCACertificatePath ディレクティブで指定します。SSLCACertificatePath ディレクティブには，証明書発行元の CA の証明書をポイントするハッシュ値を使用したシンボリックリンク（ハッシュリンク）を格納したディレクトリを指定します。ハッシュ値は sslccert x509 ユティリティで作成します。

SSLCACertificatePath ディレクティブを指定すると，Web サーバでの証明書の検索はハッシュ値を用いて効率良く実行できます。したがって，CA の証明書が多い場合は，SSLCACertificateFile ディレクティブよりも SSLCACertificatePath ディレクティブを推奨します。なお，ハッシュ値は一つの証明書に一つである必要があるので，ハッシュリンク作成時には，複数の証明書が混在したファイルは指定できません。

SSLCACertificatePath ディレクティブで指定するハッシュリンクディレクトリ内のシンボリックリンク生成時には，ハッシュ値に .0 を付ける必要があります。また，SSLCACertificatePath ディレクティブで指定するディレクトリは，User，Group ディレクティブで指定したユーザでアクセスできるように，ディレクトリに読み込み権限，実行権限を設定してください。

（1） 形式

```
sslccert x509 -noout -hash < CAの証明書ファイル
```

（2） オペランド

- **CA の証明書ファイル**

  ハッシュリンク値を作成する CA の証明書ファイルを指定します。

（3） 使用例

ハッシュリンクのディレクトリおよび CA の証明書が次に示すディレクトリ，ファイルの場合の例を示します。

/opt/hitachi/httpsd/conf/ssl/cacerts：ハッシュリンクディレクトリ

/opt/hitachi/httpsd/sslc/bin/demoCA/cacert.pem：CA の証明書

```
cd /opt/hitachi/httpsd/conf/ssl/cacerts
ln -s /opt/hitachi/httpsd/sslc/bin/demoCA/cacert.pem `sslccert x509 -noout
-hash < /opt/hitachi/httpsd/sslc/bin/demoCA/cacert.pem`.0
```

これによって，/opt/hitachi/httpsd/sslc/bin/demoCA/cacert.pem についてのハッシュリンク xxxxxxxx.0 が作成されます。

5.2.7 sslckey ユティリティおよび sslccert ユティリティの使用例

sslckey ユティリティおよび sslccert ユティリティの使用例を示します（ただし，使用例は Windows 版の例です）。なお，この例で使用している Common Name，Email Address などは，すべて架空のものです。

（1） 秘密鍵の作成（sslckey ユティリティ）

秘密鍵の生成の使用例を次に示します。

使用例

```
C:¥Program Files¥Hitachi¥httpsd¥sslc¥bin>sslckey genrsa -rand file -des3 -out
demoCA/httpsdkey.pem 1024
Loading 'entropy' into random state - 67144 semi-random bytes loaded
Generating 2 prime RSA private key, 1024 bit long modulus
...+++++
...+++++

e is 65537 (0x10001)
Enter PEM pass phrase:                <--- 4文字以上のパスワードを入力する
Verifying password - Enter PEM pass phrase:          <--- 再入力する
```

注　-rand オペランドの指定値 file には，適当なファイルを指定してください。

秘密鍵の内容

秘密鍵の内容を，次に示します。

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,838F6F481ABB2A00

Hvzt8P1deXb6+kEAU2TW8zS5eeXfQwh7ZrhNwATsVvDJp+MIg3gTP6aBHqoF4mve
5mC3PROTakKYe12Sard63kZujRrGo+Lp70E5ZYKuagKh7TrySWfIICFezsVwXewP
XrDMx2gtLzK9mz2/4ZzQ/bykaByhKXeCVqvRhkRGmGy40DU5ja+h3jTLw5C0YUDm
AVf/OBwKWNGPB3Aua7e801csEECENRbWmRs2MCzVt4c3+iRgovRbDC1A1+pGtjL2
WFa4z8JHumsCCqGSUYMHDFIkpi3yJYDEsRN4obj5qnEnq3mG9CngZg5SPBYQFGTR
udXCOT+iOREi4iGH/Wft1IJUi9OPm94dJ+UmMOXJAZfN8wN3ATbhqaVyzftV9Tvt
MZhxiaGASTaJii6KQDXgjDLGQntUtx5jkILDRYA7f/EOoXGFuqTf2s9JNmThg6IU
CK3Ud5XYM0fhi/5y5PoeiyFFuuQWZz5bLYX8IZ0YE3KKhzfZuCsCrCdIlfGBm6s+
Degs/5IB+xUOm2zFoiH6n4wP39QI23TQTsE4hQkgkFLfAg2FUNYN0cGRWU4hJllY
hYcrxXrqkwEsiB3VDCgvSsiknZyNhdZKQzKQXJGKFdekZzrUVIv+QPrjwjG9ELTR
FPBoa4deumyyIeb90A4SKNS8wbFkgI9lKWXU7/87pg6D55Geya+WguzqbKAqizse
CU02ou1HmtNofufc8Gk9xRl4MyGehb/RicVwM3IdU1tp6OLImxNzcUsM2SrqwFZz
L/u9EK9ByzmuQlUzVRe+4UG8wNrEnD5t/405Ukoug7JzgA7s2b4Flw==
-----END RSA PRIVATE KEY-----
```

（2） 証明書発行要求（CSR）の作成（sslccert ユティリティ）

sslccert req ユティリティを使用して，証明書発行要求（CSR）を作成します。ここで作成した CSR ファイルを CA に提出して，署名済みの証明書を発行してもらいます。なお，Web サーバの秘密鍵作成時にパスワードを設定した場合は，CSR 作成時に秘密鍵のパスワードの入力要求があります。

設定する項目および内容は，CSR を提出する CA の指示に従ってください。

使用例

```
C:¥Program Files¥Hitachi¥httpsd¥sslc¥bin>sslccert req -config demoCA/sslc.cnf
-new -SHA1 -key demoCA/httpsdkey.pem -out demoCA/httpsd.csr
Using configuration from demoCA/sslc.cnf
Enter PEM pass phrase:                  <--- 秘密鍵のパスワードを入力する
You will be prompted to enter information to incorporate
into the certificate request.
This information is called a Distinguished Name or a DN.
There are many fields however some can remain blank.
Some fields have default values.
Enter '.', to leave the field blank.
-----
Country Name (2 letter code) [JP]:
State or Province Name (full name) [Kanagawa]:
Locality Name (eg, city) [Yokohama-shi]:
Organization Name (eg, company) []:HITACHI
Organizational Unit Name (eg, section) []:WebSite
Common Name (eg, YOUR name) []:www.hws.hitachi.co.jp
Email Address [www-admin@server.example.com]:www-admin@hws.hitachi.co.jp

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:           <--- 入力しない場合はリターンキーを押してください
An optional company name []:       <--- 入力しない場合はリターンキーを押してください
```

CSR の形式

CSR の形式を次に示します。

```
-----BEGIN CERTIFICATE REQUEST-----
MIIB6DCCAVECAQAwgacxCzAJBgNVBAYTAkpQMREwDwYDVQQIEwhLYW5hZ2F3YTEV
MBMGA1UEBxMMWW9rb2hhbWEtc2hpMRAwDgYDVQQKEwdISVRBQ0hJMRAwDgYDVQQL
EwdXZWJTaXRlMR4wHAYDVQQDExV3d3cuaHdzLmhpdGFjaGkuY28uanAxKjAoBgkq
hkiG9w0BCQEWG3d3dy1hZG1pbkBod3MuaGl0YWNoaS5jby5qcDCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEAxq4ChoNI3JXQKKmimWeWXgg+7wwjvPLk3awnpg9U
Xt5lLSqL6d71w2chFIaj4OYDNkbQKtto3qTX/wo37XmK+u9dIfKFwFwNDA7AVKMX
Zrl1nIugT5Vb1htwZpBuCDAHi7HiaeCQYJvE3e3roKib5SGmbyZ6erPt+py0c4py
HgsCAwEAAaAAMA0GCSqGSIb3DQEBBQUAA4GBAFwl4q/yBM7jzSIEMOXDnJPxC5gw
XJBDna+rFXxaT6aelUEubKyCC2MXb9sdMC4cPfnIwyibLn/n2beDCZoaHOPsHZ+e
3ROaNkVdF3xmdgGzeG3yJBUQRFghlBefJLdiQcbavL5jjOCWWYy9KytOS2mO9PaT
U2f2SuQzc8ZED0JN
-----END CERTIFICATE REQUEST-----
```

5.2.8 プロンプトモードでの sslckey ユティリティおよび sslccert ユティリティの実行

sslckey ユティリティおよび sslccert ユティリティは，プロンプトモードでも利用できます。プロンプトモードにするためには，次のように入力します。

(1) 形式

- UNIX 版の場合

```
# /opt/hitachi/httpsd/sslc/bin/sslckey
sslckey>
```

```
# /opt/hitachi/httpsd/sslc/bin/sslccert
sslccert>
```

- Windows 版の場合

```
C:¥>"C:¥Program Files¥Hitachi¥httpsd¥sslc¥bin¥sslckey"
sslckey>
```

```
C:¥>"C:¥Program Files¥Hitachi¥httpsd¥sslc¥bin¥sslccert"
sslccert>
```

sslckey ユティリティのプロンプトモードでは，genrsa コマンドを 5.2.1 に記述した形式で利用できます。また，sslccert ユティリティのプロンプトモードでは，req，x509 の各コマンドを，5.2.2 ～ 5.2.6 に記述した形式で利用できます。

（2） 使用例

```
sslccert>req -in demoCA/httpsd.csr -text
```

プロンプトモードは次に示すコマンドで終了します。

```
sslccert>exit
```

5.3 CRL の運用

5.3.1 CRL のダウンロード

LDAP サーバのエントリにアクセスして，指定した属性から無効になった証明書のリスト（CRL）をダウンロードします。CRL を取得するエントリ，属性および CRL の形式はあらかじめ LDAP 管理者に確認してください。

次に CRL をダウンロードする crldownload ユティリティについて説明します。

crldownload ユティリティは，AIX，Solaris，Linux（32 ビット）および Windows 版で使用できます。HP-UX（IPF）および Linux（IPF）版では使用できません。

（1） 形式

```
crldownload -b 検索ベースDN -L LDAPライブラリ名 -o ファイル名〔-a 属性〕〔-D バインドDN〕〔-h ホスト名〕〔-H〕〔-p ポート番号〕〔-w パスワード〕
```

（2） オプション

● -b **検索ベース** DN
CRL の格納されているエントリの DN を指定します。

● -L LDAP **ライブラリ名**
使用する LDAP ライブラリファイル名を指定します。

● -o **ファイル名**
CRL を出力するファイル名を指定します。

● -a **属性**　～《certificateRevocationList;binary》
CRL が格納されている属性を指定します。

● -D **バインド** DN
バインドする DN を指定します。省略した場合は，匿名バインドが実行されます。

● -h **ホスト名**　～《localhost》
アクセスする LDAP サーバのホスト名または IP アドレスを指定します。

● -H
ヘルプを表示させる場合に指定します。このオプションは，ほかのオプションとは，併用できません。

● -p **ポート番号**　～((1 － 65535))《389》
アクセスする LDAP サーバのポート番号を指定します。

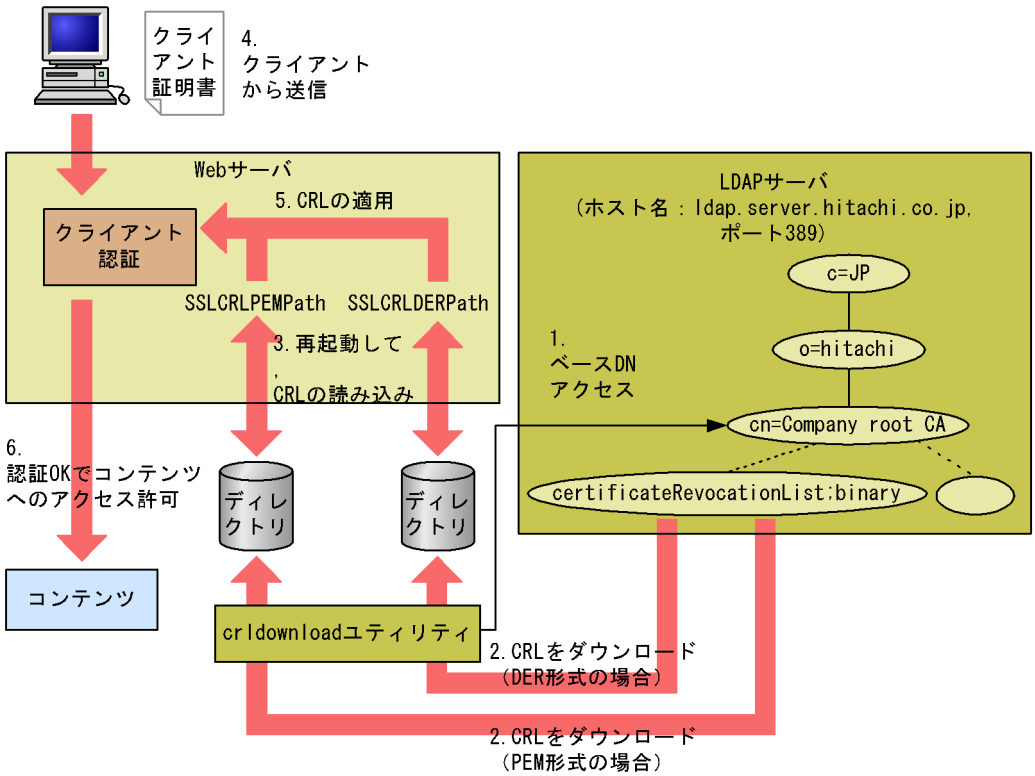
● -w パスワード　～《NULL》

バインド DN にバインドする場合に使用するパスワードを指定します。省略した場合は，パスワードを使用しません。

（3） 使用方法

crldownload ユティリティの使用方法を次に示します。

図 5-3　crldownload ユティリティの使用方法



1. crldownload ユティリティで，CRL を格納しているエントリにアクセスします。
2. エントリ内の属性から CRL を取得します。格納されている CRL が DER 形式の場合，SSLCRLDERPath ディレクティブで指定したディレクトリにダウンロードします。CRL が PEM 形式の場合，SSLCRLPEMPath ディレクティブで指定したディレクトリにダウンロードします。
3. Web サーバを再起動します。
4. クライアントは，SSL でアクセスしたときに証明書を送信します。
5. クライアント証明書を認証するとき，取得した CRL を適用します。
6. 認証に成功した場合には，コンテンツにアクセスできます。

注

SSLCRLPEMPath，SSLCRLDERPath ディレクティブで指定されたディレクトリ

内に，不適切な形式のファイルがあった場合，Web サーバは起動しません。そのため，crldownload ツールを使用したときは，これらのディレクトリに CRL を格納する前に，正しい形式の CRL であるかどうかを確認してください。

(4) 使用例

(a) DER 形式の CRL のダウンロード

次に示すスクリプトを実行して CRL をダウンロードします。このスクリプトのファイル名を UNIX 版は /opt/hitachi/httpsd/sbin/hws_getCRL.sh，Windows 版は C:¥Program Files¥Hitachi¥httpsd¥sbin¥hws_getCRL.bat とします。

● スクリプトの実行内容

LDAP サーバ内に DER 形式で格納されている CRL をダウンロードし，Web サーバを再起動します。そのときに出力されたメッセージはログファイルに格納します。

CRL の格納先ファイル名

UNIX 版

/opt/hitachi/httpsd/conf/ssl/crl/DER/rootCA.crl

Windows 版

C:¥Program Files¥Hitachi¥httpsd¥conf¥ssl¥crl¥DER¥rootCA.crl

ログファイル名

UNIX 版

/opt/hitachi/httpsd/logs/crl_log

Windows 版

C:¥Program Files¥Hitachi¥httpsd¥logs¥crl_log

スクリプトの内容 (UNIX **版の場合**)

```
#!/bin/sh
################################################################################
##  Hitachi Web Server
##  All Rights Reserved. Copyright (C) 2001-2008, Hitachi, Ltd.
################################################################################
#parameters
LIB="/opt/hitachi/httpsd/libexec/libldapssl41.so"
HOST="ldap.server.hitachi.co.jp"
PORT="389"
BASE="cn=Company root CA, o=Hitachi, c=JP"
ATTR="certificateRevocationList;binary"
FILE="/opt/hitachi/httpsd/conf/ssl/crl/DER/rootCA.crl"

LOG="/opt/hitachi/httpsd/logs/crl_log"
TMP="/opt/hitachi/httpsd/conf/ssl/crl/tmp-rootCA.crl"

#download
TOOL="/opt/hitachi/httpsd/sbin/crldownload"
HTTPSD="/opt/hitachi/httpsd/sbin/httpsdctl graceful"
LOGTIME=""
LANG=C

if `$TOOL -L $LIB -h $HOST -p $PORT -b "$BASE" -a "$ATTR" -o $TMP >> $LOG
2>&1`then
  if `mv -f $TMP $FILE >> $LOG 2>&1`
  then
    $HTTPSD >> $LOG 2>&1
    exit 0
  else
    LOGTIME=`date`
    echo "[$LOGTIME] Moving $TMP to $FILE failed" >> $LOG
    rm -f $TMP >> /dev/null 2>&1
  fi
else
  LOGTIME=`date`
fi
echo "[$LOGTIME] Stop restarting Hitachi Web Server." >> $LOG
exit 1
```

スクリプトの内容 (Windows **版の場合**)

```
@echo off
REM
################################################################################
REM ## All Rights Reserved. Copyright (C) 2002-2008, Hitachi, Ltd.
REM
################################################################################
REM #parameters
SETLOCAL
SET LIB="C:¥Program Files¥Hitachi¥httpsd¥libldap¥nsldap32v50.dll"
SET HOST="ldap.server.hitachi.co.jp"
SET PORT="389"
SET BASE="cn=Company root CA, o=Hitachi, c=JP"
SET ATTR="certificateRevocationList;binary"
SET FILE="C:¥Program Files¥Hitachi¥httpsd¥conf¥ssl¥crl¥DER¥rootCA.crl"

SET FORM="DER"
SET LOG="C:¥Program Files¥Hitachi¥httpsd¥logs¥crl_log"
SET TMPCRL="C:¥Program Files¥Hitachi¥httpsd¥conf¥ssl¥crl¥tmp-rootCA.crl"

REM #download
SET TOOL="C:¥Program Files¥Hitachi¥httpsd¥sbin¥crldownload.exe"
SET HTTPSD="C:¥Program Files¥Hitachi¥httpsd¥httpsd.exe"

%TOOL% -L %LIB% -h %HOST% -p %PORT% -b %BASE% -a %ATTR% -o %TMPCRL% >> %LOG%
2>&1 || GOTO ERR

COPY %TMPCRL% %FILE% >> %LOG% 2>&1 || GOTO CPERR
DEL %TMPCRL% >> %LOG% 2>&1
%HTTPSD% -n "Hitachi Web Server" -k restart >> %LOG% 2>&1
GOTO TOOLEND

:CPERR
 ECHO Moving %TMPCRL% to %FILE% failed >> %LOG%
 DEL %TMPCRL% >> %LOG% 2>&1
 DEL %TMPSSL% >> %LOG% 2>&1
 GOTO ERR

:ERR
 ECHO Stop restarting Hitachi Web Server. >> %LOG%

:TOOLEND
 endlocal
 echo on
```

● **スクリプトの実行方法**（UNIX **版の場合**）

```
/opt/hitachi/httpsd/sbin/hws_getCRL.sh
```

● **スクリプトの実行方法**（Windows **版の場合**）

```
C:¥> "C:¥Program Files¥Hitachi¥httpsd¥sbin¥hws_getCRL.bat"
```

(b) 定期的なダウンロードと Web サーバの再起動

● UNIX 版

スーパユーザまたはスーパユーザから許可されたユーザは crontab コマンド※を使用した CRL の定期的なダウンロードや，Web サーバの再起動ができます。
crontab コマンドに crldownload コマンドまたは crldownload コマンドを記述したスクリプトを実行する時間を指定して，定期的に CRL をダウンロードし，Web サーバを再起動します。
注※ OS コマンドの一つ。cron へのジョブの登録や制御をするときに使用します。詳細な指定方法は各 OS マニュアルを参照してください。

crontab **コマンドの指定方法**

```
# crontab -
分  時  日  月  曜日  コマンド
```

各 crontab ファイルエントリは，六つのフィールドから成る行で構成されます。各フィールドは，スペースまたはタブで区切られ，それぞれ次に示す値を含みます。
分：コマンドを実行する分（0 から 59）
時：コマンド実行の時間（0 から 23）
日：コマンド実行の日（1 から 31）
月：コマンド実行の月（1 から 12）
曜日：コマンド実行の曜日（日曜日から土曜日までを示す 0 から 6）
コマンド：実行するシェルコマンド
*（アスタリスク）は有効な値すべてを意味します。

crontab **コマンドの指定例**

毎日，午前 8 時に CRL をダウンロードして，Web サーバを再起動する（(a) のスクリプトを実行する）には，次のように指定します。

```
# crontab -
0 8 * * * /opt/hitachi/httpsd/sbin/hws_getCRL.sh
 (<Ctrl>+<d>キーで入力を終了します)
#
```

● Windows 版

at コマンド※を使用して CRL の定期的なダウンロードや，Web サーバの再起動ができます。
at コマンドに crldownload コマンドまたは crldownload コマンドを記述したスクリプトを実行する時間を指定して，定期的に CRL をダウンロードし，Web サーバを再起動します。
注※ OS コマンドの一つ。詳細な指定方法は各 OS マニュアルを参照してください。

at コマンドの指定方法

```
C:¥>at ［¥¥コンピュータ名］時刻［/every:日付［,...］|/next:日付［,...］］"コマンド"
```

at コマンドの指定例

毎日，午前 8 時に CRL をダウンロードして，Web サーバを再起動する（(a) のスクリプトを実行する）には，次のように指定します。

```
C:¥>at 8:00 /every:M,T,W,Th,F,S,Su "C:¥Program
Files¥Hitachi¥httpsd¥sbin¥hws_getCRL.bat"
```

（5） Solaris 版での注意事項

標準提供している LDAP ライブラリファイル以外を使用する場合は，環境変数 LD_LIBRARY_PATH に，使用する LDAP ライブラリファイルの格納ディレクトリを設定してください。

5.4 パスワード付きサーバ秘密鍵の使用

パスワードによって保護されているサーバ秘密鍵を使用する場合，パスワードをあらかじめファイルに格納しておき，ディレクティブを設定することで，サーバ起動時のパスワード入力を省略できます。その手順を以下に示します。なお，Windows 版 Hitachi Web Server でパスワードによって保護されているサーバ秘密鍵を使用する場合には，この手順は必須です。

1. sslckey ユティリティによって，パスワード付きのサーバ秘密鍵を作成します。
2. sslpasswd ユティリティによって，パスワードファイルを作成します。
3. 作成したパスワードファイルを指定した SSLCertificateKeyPassword ディレクティブを，サーバ秘密鍵ファイルを指定した SSLCertificateKeyFile ディレクティブとともに httpsd.conf に設定します。
4. サーバを起動または再起動します。

パスワードファイルの内容の漏洩には注意する必要があります。サーバ秘密鍵の格納ディレクトリに加え，パスワードファイルの格納ディレクトリでも，他ユーザからのアクセスを禁止するようにディレクトリパーミッションやファイルパーミッションの設定をしてください。

SSLCertificateKeyPassword ディレクティブで指定するパスワードファイルを作成する sslpasswd ユティリティについて次に示します。

5.4.1 sslpasswd ユティリティ

(1) 形式

```
sslpasswd サーバ秘密鍵ファイル名 パスワードファイル名
```

(2) オペランド

● サーバ秘密鍵ファイル名

パスワードによって保護されたサーバ秘密鍵を指定します。

● パスワードファイル名

パスワードファイルを出力するファイル名を指定します。

(3) 使用例

```
sslpasswd httpsdkey.pem .keypasswd
```

(4) 注意事項

- パスワードファイル名として，既存のファイル名は指定できません。

- Windows 版の sslpasswd ユティリティで作成したパスワードファイルは UNIX 版では使用できません。
- UNIX 版の sslpasswd ユティリティで作成したパスワードファイルは Windows 版では使用できません。

6 ディレクティブ

この章では，httpsd.conf ファイルおよびアクセスコントロールファイルに定義するディレクティブについて説明します。

6.1 ディレクティブ一覧

6.1.1 ディレクティブ一覧

コンフィグファイルに指定できるディレクティブの一覧を次に示します。

表 6-1 ディレクティブ一覧

設定内容	ディレクティブ	複数指定
httpsd.conf ファイル内のブロックの定義	<Directory>	○
	<DirectoryMatch>	○
	<Files>	○
	<FilesMatch>	○
	<IfModule>	○
	<Limit>	○
	<Location>	○
	<LocationMatch>	○
	<VirtualHost>	○
サーバの基本的な定義	ServerName	×
	Port	×
	User 	×
	Group 	×
	ServerAdmin	×
	ServerRoot	×
	ServerSignature	×
	Listen	○
	BindAddress	×
	LoadModule	○
	LoadFile	○
	Include	○
	ExtendedStatus	×
	ServerTokens	×
	CoreDumpDirectory 	×
	FileETag	×
コンテンツを管理するための定義	UserDir	○
	DocumentRoot	×

設定内容	ディレクティブ	複数指定
	ErrorDocument	○
Web ブラウザからのリクエストについての定義（Alias）	Alias	○
	AliasMatch	○
	Redirect	○
	RedirectMatch	○
Web ブラウザへのレスポンスについての定義	HWSNotModifiedResponseHeaders	○
MIME タイプについての定義	DefaultType	×
	TypesConfig	×
	AddCharset	○
	AddDefaultCharset	×
	AddType	○
	ForceType	×
	HWSErrorDocumentMETACharset W	×
コンテントネゴシエーションについての定義	LanguagePriority	○
	AddEncoding	○
	AddLanguage	○
	DefaultLanguage	×
	CacheNegotiatedDocs	×
	MultiviewsMatch	×
ハンドラについての定義	AddHandler	○
	SetHandler	×
Web サーバの性能についての定義	StartServers U	×
	MinSpareServers U	×
	MaxSpareServers U	×
	MaxClients U	×
	MaxRequestsPerChild U	×
	Timeout	×
	ListenBacklog	×
	ThreadsPerChild W	×
	HWSMaxQueueSize W	×

設定内容	ディレクティブ	複数指定
	HWSKeepStartServers U	×
KeepAlive の定義	KeepAlive	×
	MaxKeepAliveRequests	×
	KeepAliveTimeout	×
リクエストを制限する定義	LimitRequestBody	×
	LimitRequestFields	×
	LimitRequestFieldsize	×
	LimitRequestLine	×
CGI，環境変数の定義	ScriptAlias	○
	ScriptAliasMatch	○
	UseCanonicalName	×
	BrowserMatch	○
	BrowserMatchNoCase	○
	PassEnv	○
	SetEnv	○
	UnsetEnv	○
	SetEnvIf	○
	SetEnvIfNoCase	○
	Action	○
	Script	○
	ScriptInterpreterSource W	×
	HWSSetEnvIfIPv6	○
ディレクトリインデクスの表示内容の定義	DirectoryIndex	×
	FancyIndexing	×
	AddIconByEncoding	○
	AddIconByType	○
	AddIcon	○
	DefaultIcon	×
	ReadmeName	×
	HeaderName	×
	IndexIgnore	○
	IndexOrderDefault	×
	AddAltByEncoding	○
	AddAltByType	○

設定内容	ディレクティブ	複数指定
	AddAlt	○
	AddDescription	○
	IndexOptions	○
Web サーバへのアクセスを制御する定義	AccessFileName	×
	AllowOverride	×
	AuthName	×
	AuthType	×
	AuthGroupFile	×
	AuthUserFile	×
	AuthAuthoritative	×
	Require	○
	Options	×
	Order	×
	Allow from	○
	Deny from	○
	Satisfy	×
	TraceEnable	×
	IdentityCheck 	×
SSL による暗号化および認証の定義	SSLRequireSSL	×
	SSLEnable	×
	SSLDisable	×
	SSLCertificateFile	×
	SSLCertificateKeyFile	×
	SSLCACertificatePath 	×
	SSLCACertificateFile	×
	SSLVerifyClient	×
	SSLVerifyDepth	×
	SSLRequiredCiphers	×
	SSLRequireCipher	○
	SSLBanCipher	○
	SSLDenySSL	×
	SSLFakeBasicAuth	×
	SSLCacheServerPort 	×
	SSLSessionCacheTimeout	×

設定内容	ディレクティブ	複数指定
	SSLCacheServerPath U	×
	SSLCacheServerRunDir U	×
	SSLSessionCacheSize	×
	SSLSessionCacheSizePerChild U	×
	SSLCRLAuthoritative	×
	SSLCRLDERPath	×
	SSLCRLPEMPath	×
	SSLExportCertChainDepth	×
	SSLExportClientCertificates	×
	SSLCertificateKeyPassword	×
	SSLProtocol	×
Web サーバを運用形態に合わせて複数ホストに見せる定義	NameVirtualHost	○
	ServerAlias	○
	ServerPath	×
イメージマップファイルについての定義	ImapDefault	×
	ImapBase	×
	ImapMenu	×
	HWSImapMenuCharset	×
採取するログの定義	HostnameLookups	×
	ErrorLog	×
	LogLevel	×
	LogFormat	○
	CustomLog	○
	TransferLog	○
	PidFile	×
	ScriptLog	×
	ScriptLogBuffer	×
	ScriptLogLength	×
	HWSLogSSLVerbose	×
	HWSLogTimeVerbose	×
	HWSRequestLog	×
	HWSRequestLogType	×

設定内容	ディレクティブ	複数指定
	HWSSuppressModuleTrace	○
ディレクトリサーバについての定義	LDAPBaseDN	×
	LDAPRequire	○
	LDAPServerName	×
	LDAPServerPort	×
	LDAPSetEnv	○
	LDAPTimeout	×
	LDAPUnsetEnv	○
	LDAPNoEntryStatus	×
採取するトレースの定義	HWSTraceIdFile	×
	HWSTraceLogFile	×
	HWSStackTrace 	×
リバースプロキシについての定義	ProxyPass	○
	ProxyPassReverse	○
	ProxyVia	×
	ProxyErrorOverride	×
	ProxyPreserveHost	×
	HWSProxyPassReverseCookie	○
流量制限機能についての定義	QOSCookieDomain	×
	QOSCookieExpires	×
	QOSCookieName	○
	QOSCookieSecure	×
	QOSCookieServers	×
	QOSRedirect	○
	QOSRejectionServers	×
	QOSResponse	×
ヘッダカスタマイズ機能についての定義	Header	○
	RequestHeader	○
有効期限設定機能についての定義	ExpiresActive	×
	ExpiresByType	○
	ExpiresDefault	×
計画停止についての定義	HWSGracefulStopLog	×
	HWSGracefulStopTimeout	×

設定内容	ディレクティブ	複数指定
静的コンテンツキャッシュ機能についての定義	HWSContentCacheMaxFileSize W	×
	HWSContentCacheSize W	×

（凡例）

○：指定できる。
×：指定できない。

注

特に記述がないかぎり，ファイル名はディレクトリ名を含めた形式（パス情報付き）で記述できます。

6.1.2 正規表現

ディレクティブの指定に使用できる正規表現を次に示します。

表 6-2 正規表現

記号	機能	使用例	使用例の意味
.	任意の 1 文字。	a...c	a の後に任意の 3 文字と c が続く。abcdc は適合する。
*	直前の 1 文字の 0 個以上の繰り返し。	ab*cd*	ac，abbbbc，abbbbcd は適合する。
+	直前の 1 文字の 1 個以上の繰り返し。	ab*c+	abbbc は適合する。abbb は適合しない。
?	直前の 1 文字があるかないか。	abbbc?	abbbc，abbb は適合する。
|	選択肢の区切り。	a|bc|d	a，bc または d。
¥	直後の特殊文字（. ^$*+?|¥[]()\{\}）の 1 文字。ただし，¥ を表す場合は ¥¥¥。	¥.	. と適合する。
		¥¥¥	1 文字の ¥ と適合する。
^	行の先頭に適合する。	^ab	abcde は適合する。
$	行の末尾に適合する。	abc$	aaabc は適合する。
{m}	直前の正規表現の m 個の繰り返し。	a{5}	aaaaa が適合する。
{m,}	直前の正規表現の m 個以上の繰り返し。	a{3,}	aaa，aaaa は適合する。aa は適合しない。
{m,n}	直前の正規表現の m 個以上 n 個以下の繰り返し。	a{3,5}	aaa，aaaa，aaaaa が適合する。aa，aaaaaa は適合しない。
[文字列]	文字列にある任意の 1 文字※。	[abc]* または [a-c]*	aaa，bbb，ccc，cba，aab は適合する。

記号	機能	使用例	使用例の意味
[^ 文字列]	文字列にない任意の 1 文字。	[^0-9]	数字以外の 1 字が適合する。
(文字列)	文字列をグループ化する。	(ab)+	ababab が適合する。ababb は適合しない。
		aa(xx|yy)bb	aaxxbb，aayybb が適合する。

注※

次の 3 文字は，[文字列] 内で特殊な意味を持ちます。

^：[の次に指定して，文字列に含まれないものを示すために用います。

]：文字列の最後を示すために用います。

-：範囲を指定するために用います。

また，これら特殊文字の前の ¥ は省略されます。

[文字列] 内で特殊な意味を持つ文字を通常の文字として指定するには，次のようにします。なお，^] - ¥ 以外の特殊文字は，通常の文字として扱われます。

^ ：文字列の先頭以外で指定します。（例）[ab^yz]

]：文字列の先頭に指定します。（例）[]abxy]

-：最後に指定します。（例）[abxy-]

¥：¥¥¥ と指定します。（例）[¥¥¥abxy]

6.1.3 ディレクティブについての注意事項

(1) 記述できる場所について

ディレクティブによっては，記述できる場所が制限されているものがあります。「6.2 ディレクティブの詳細」では，各ディレクティブの記述できる場所を次のような形式で記述します。

記述できる場所	説明
httpsd.conf	VirtualHost ブロック，Directory ブロック以外の httpsd.conf ファイル
<VirtualHost>	httpsd.conf ファイルの VirtualHost ブロック
<Directory>	httpsd.conf ファイルの Directory ブロック，Location ブロック，Files ブロック
.htaccess	AccessFileName ディレクティブで指定したアクセスコントロールファイル
<Location>	httpsd.conf ファイルの Location ブロック

また，ディレクティブは次に示す順に参照されます。

1. VirtualHost ブロック，Directory ブロック以外の httpsd.conf ファイル
2. httpsd.conf ファイルの VirtualHost ブロック
3. httpsd.conf ファイルの Directory ブロック
4. アクセスコントロールファイル
5. httpsd.conf ファイルの Files ブロック

6. httpsd.conf ファイルの Location ブロック

Directory ブロックの AllowOverride ディレクティブの定義（上書き許可レベル）によって，アクセスコントロールファイルで定義しているディレクティブを有効または無効にできます。

(2) 上書き許可

AllowOverride ディレクティブで上書きを許可する場合の許可レベルを定義します。各ディレクティブの上書き許可レベルは，各ディレクティブで説明します。許可レベルは複数あります。詳細は，AllowOverride ディレクティブを参照してください。なお，各ディレクティブの説明で .htaccess が指定でき，かつ上書き許可レベルの記述がない場合には，許可レベルは All になります。

(3) ディレクティブに指定するパス情報

ディレクトリ名，ファイル名またはパス名を指定するディレクティブの場合，ディレクティブの種類によって，指定できるパス情報が異なります。

パスの種類には，次のものがあります。各ディレクティブのパス情報は，各ディレクティブで説明します。

- 絶対パスしか指定できない（Windows 版の場合，ドライブ名を付けた指定も絶対パスに含まれる）。
- ServerRoot ディレクティブの指定値からの相対パスで指定できる（ただし，ServerRoot ディレクティブの指定が先に必要）。

また，パス情報にネットワーク上のディレクトリやファイルを指定することはできません。ネットワークを使用したファイルシステム上のディレクトリやファイルを指定することもできません。

(4) コメント行

コンフィグファイル中，行の最初に # を付けると，コメント行になります。ただし，ディレクティブを指定したあとに # から始まる文字列を記述しても，# 以降をコメントとして扱いません。コメント行を指定する場合の記述例を次に示します。

正しい例

```
#Deny from all
```

行はコメント行として扱われます。

誤った例

```
Deny from all    #comment
```

#comment はディレクティブ指定値として扱われます。コメントとしては扱われません。

(5) IPv6 アドレスを指定するときの注意

ディレクティブに IPv6 アドレスを記述する場合は，「[IPv6 アドレス]」のように IPv6 アドレスを [ ] で囲んで指定してください。また，ディレクティブに IPv6 アドレスとポート番号を併記する場合は，「[IPv6 アドレス]: ポート番号」のように IPv6 アドレスを [ ] で囲み，「:」の後ろにポート番号を指定します。

ただし，次のディレクティブに IPv6 アドレスを記述する場合は，IPv6 アドレスを [ ] で囲まないで指定してください。

- Allow from ディレクティブ
- Deny from ディレクティブ
- HWSSetEnvIfIPv6 ディレクティブ

Windows 版で IPv6 アドレスを指定する場合は，リンクローカルアドレスまたはサイトローカルアドレスを指定できません。グローバルアドレスを指定してください。

Linux（32 ビット）版で IPv6 アドレスを指定する場合は，リンクローカルアドレスを指定できません。サイトローカルアドレスまたはグローバルアドレスを指定してください。

(6) 最低限設定が必要なディレクティブ

Hitachi Web Server を起動するために，最低限設定が必要なディレクティブは以下のとおりです。

- 最低限設定が必要なディレクティブ
  User（UNIX 版）
  Group（UNIX 版）
  ServerName
  SSLDisable（SSL を利用しない場合）
- SSL を利用する場合にさらに最低限必要なディレクティブ
  SSLCertificateFile
  SSLCertificateKeyFile

6.2 ディレクティブの詳細

6.2.1 < で始まるディレクティブ

ブロック定義のディレクティブを参照順に示します。

1. <Directory> ディレクティブ，<DirectoryMatch> ディレクティブ，アクセスコントロールファイル
2. <Files> ディレクティブ，<FilesMatch> ディレクティブ
3. <Location> ディレクティブ

（1） <Directory　ディレクトリ名 > ・・・ </Directory>

(a) 内容

特定のディレクトリに対してディレクティブを定義する場合に指定します。**ディレクトリ名**にディレクトリ名を指定し，そのディレクトリとサブディレクトリだけに有効なディレクティブを定義するブロックを指定できます。

ディレクトリ名は，絶対パスで指定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
<Directory />                                              …1.
     Options None                                          …2.
     AllowOverride None                                    …3.
</Directory>                                               …4.

<Directory "C:/Program Files/Hitachi/httpsd/htdocs">       …5.
     Options Indexes                                       …6.
     AllowOverride None                                    …7.
     Order allow,deny                                      …8.
     Allow from all                                        …9.
</Directory>                                               …10.
```

1. ルートディレクトリの定義
2. 機能はすべて無効
3. すべての上書き禁止
4. 定義終わり
5. C:/Program Files/Hitachi/httpsd/htdocs ディレクトリの定義
6. ディレクトリインデクス表示可
7. すべての上書き禁止
8. Allow ディレクティブの指定を Deny ディレクティブの指定より先に評価
9. すべてのホストからのアクセスを許可
10. 定義終わり

（2） <DirectoryMatch　正規表現 >・・・</DirectoryMatch>

(a) 内容

正規表現で記述した条件を満たすディレクトリに対してディレクティブを定義する場合に指定します。ディレクトリ名を正規表現で指定し，そのディレクトリとサブディレクトリだけに有効なディレクティブを定義するブロックを指定できます。

正規表現のディレクトリ名は，絶対パスで指定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

（3） <Files　ファイル名 >・・・</Files>

(a) 内容

特定のファイルに対してディレクティブを定義する場合に指定します。**ファイル名**にファイル名を指定し，そのファイルだけに有効なディレクティブを定義するブロックを指定できます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

（4） <FilesMatch　正規表現 >・・・</FilesMatch>

(a) 内容

正規表現で記述した条件を満たすファイルに対してディレクティブを定義する場合に指定します。ファイル名を正規表現で指定し，そのファイルだけに有効なディレクティブを定義するブロックを指定できます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

（5） <IfModule　〔!〕モジュール名 >・・・</IfModule>

(a) 内容

指定したモジュールが組み込まれているとき，ブロック内で指定したディレクティブが有効になります。モジュール名の前に！を付けた場合は，指定したモジュールが組み込まれていないとき，ブロック内で指定したディレクティブが有効になります。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(6) <Limit メソッド名 〔メソッド名 …〕>・・・</Limit>

(a) 内容

特定の HTTP プロトコルメソッドに対して，ディレクティブを定義する場合に指定します。メソッド名に指定したメソッドだけに有効なアクセス制御のディレクティブを定義するブロックを指定できます。メソッド名は複数指定できます。

メソッド名：GET，POST，PUT，DELETE，CONNECT，OPTIONS

（HEAD は GET に含まれています）

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 指定例

```
<Directory />
  <Limit PUT DELETE>                    …1.
    Order deny,allow                    …2.
    Deny from all                       …3.
    Allow from .your_domain.com         …4.
  </Limit>                              …5.
</Directory>
```

1. PUT および DELETE メソッドに対する定義
2. Deny ディレクティブの指定を，Allow ディレクティブの指定よりも先に評価
3. すべてのホストからの PUT および DELETE メソッドによるアクセスは不可
4. .your_domain.com からの PUT および DELETE メソッドによるアクセスを許可
5. 定義終わり

(7) <Location URL>・・・</Location>

(a) 内容

特定の URL で示す場所へのリクエストについて，ディレクティブを定義する場合に指定します。ただし，URL に，? 以降（問い合わせ文字列）は指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
  <Location /server-status>             …1.
       SetHandler server-status         …2.
       Order deny,allow                 …3.
       Deny from all                    …4.
       Allow from .your_domain.com      …5.
  </Location>                           …6.
```

1. URL /server-status の定義

2. このディレクトリのリクエストは server-status ハンドラに関連づける
3. Deny ディレクティブの指定を Allow ディレクティブの指定よりも先に評価
4. すべてのホストからのアクセスは不可
5. .your_domain.com からのアクセスを許可
6. 定義終わり

（8） <LocationMatch　正規表現 >・・・</LocationMatch>

（a） 内容

正規表現で記述した条件を満たす URL へのリクエストについてディレクティブを定義する場合に指定します。ただし，URL に，? 以降（問い合わせ文字列）は指定できません。

（b） 記述できる場所

httpsd.conf，<VirtualHost>

（9） <VirtualHost　{ホスト名｜ IP アドレス〔: ポート番号〕}　〔{ホスト名｜ IP アドレス〔: ポート番号〕}　…〕 >・・・</VirtualHost>

（a） 内容

ホスト名または IP アドレス〔: ポート番号〕で示すホストへのリクエストについてディレクティブを定義する場合に指定します。

なお，IPv6 アドレスに対応したホスト名も指定できます。IP アドレスに IPv6 アドレスを指定する場合は，IPv6 アドレスを [　] で囲んでください。

（b） 記述できる場所

httpsd.conf

（c） 指定例

```
<VirtualHost 172.17.40.30:80>
    :
</VirtualHost>
<VirtualHost [fec0::123:4567:89ab:cdef]:80>
    :
</VirtualHost>
```

6.2.2　A で始まるディレクティブ

（1） AccessFileName　ファイル名　〔ファイル名　…〕

～《.htaccess》

（a） 内容

アクセス制御のディレクティブを定義しているファイル（アクセスコントロールファイル）のファイル名を定義します。AllowOverride ディレクティブで許可されていれば，

コンテンツリクエスト時に毎回このファイルを参照しアクセス制限がチェックされます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
AccessFileName .htaccess
```

アクセスコントロールファイルのファイル名は，.htaccess

(2) Action {MIME タイプ｜ハンドラ} CGI スクリプト名

(a) 内容

MIME タイプまたはハンドラで指定したコンテンツが Web ブラウザからリクエストされたとき，実行させるスクリプトを CGI スクリプト名で指定します。CGI スクリプト名は，URL で指定します。このディレクティブを複数指定する場合，同じ MIME タイプに異なる CGI スクリプトは指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
Action image/gif /cgi-bin/images.cgi
```

(3) AddAlt "文字列" 拡張子 〔拡張子 …〕

(a) 内容

ディレクトリインデクス表示時に，拡張子に指定したファイルと関連づけて文字列を表示する場合に指定します。一つの文字列に対して複数の拡張子が指定できます。テキストモードの Web ブラウザのようにアイコン表示ができない環境で，ファイルの属性を表示する場合などに利用できます。

拡張子に指定できるものを次に示します。

- ファイル拡張子
- ワイルドカード表記のファイル拡張子またはファイル名
- ファイル名

このディレクティブを複数指定する場合，同じ拡張子に異なる文字列は指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
AddAlt "HTML" htm html
```

拡張子が htm または html のファイルの場合，文字列 "HTML" を表示します。

(4) AddAltByEncoding "文字列" MIME エンコーディング 〔MIME エンコーディング …〕

(a) 内容

ディレクトリインデクス表示時に，アイコンが表示できない環境で，MIME エンコーディング（x-compress など）と関連づけて文字列を表示する場合に指定します。一つの文字列に対して複数の MIME エンコーディングが指定できます。このディレクティブを複数指定する場合，同じ MIME タイプに異なる文字列は指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
AddAltByEncoding "gzip" x-gzip
```

(5) AddAltByType "文字列" MIME タイプ 〔MIME タイプ …〕

(a) 内容

ディレクトリインデクス表示時に，アイコンが表示できない環境で，MIME タイプ（text/html など）と関連づけて文字列を表示する場合に指定します。一つの文字列に対して複数の MIME タイプが指定できます。このディレクティブを複数指定する場合，同じ MIME タイプに異なる文字列は指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
AddAltByType "plain text" text/plain
```

(6) AddCharset 文字セット 拡張子 〔拡張子 …〕

(a) 内容

ファイル拡張子に対する文字セットを指定します。文字セットは Content-Type ヘッダに charset= の値として設定されます。クライアントに対して文字セットを明示する場合に使用します。このディレクティブを複数指定する場合，同じ拡張子に異なる文字列は指定できません。指定する拡張子は，AddType ディレクティブまたは TypesConfig ディレクティブで指定したファイルで，MIME タイプの関連づけが必要です。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
AddCharset EUC-JP .euc
AddCharset ISO-2022-JP .jis
AddCharset SHIFT_JIS .sjis
```

(7) AddDefaultCharset 〔On ｜ Off ｜文字セット〕

(a) 内容

ファイル拡張子に対する文字セットのデフォルト値を指定します。AddCharset ディレクティブの設定に対するデフォルト値となります。Content-Type が text/plain，text/html の場合のデフォルトとして設定されます。

On：デフォルト文字セットとして ISO-8859-1 を設定します。

Off：文字セットを設定しません。

文字セット：指定した文字セットをデフォルト文字セットとします。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
AddDefaultCharset ISO-2022-JP
```

(8) AddDescription "文字列" ファイル名 〔ファイル名 …〕

(a) 内容

ディレクトリインデクス整形表示時に，ファイル名で指定したファイル拡張子，ワイル

ドカード表記ファイル名またはパス情報なしの完全なファイル名に対して，説明文として文字列を表示する場合に指定します。なお，ファイル名にスラッシュで終わる文字列を指定した場合，* が付けられワイルドカード指定と同様に見なされます。

ファイル名に指定できるものを次に示します。

- ファイル拡張子
- ワイルドカード表記のファイル名
- ファイル名

このディレクティブを複数指定する場合，同じファイル名に異なる文字列は指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
AddDescription "The planet Mars" /web/pics/mars.gif
```

(9) AddEncoding 圧縮形式 拡張子

(a) 内容

Web サーバ内の圧縮データを Web ブラウザに表示させるときに必要な拡張子と圧縮形式の関連づけを指定します。Web ブラウザに圧縮ファイルの展開の情報として Content-Encoding ヘッダを Web サーバから送信する場合に設定します。このヘッダを利用した運用は，Web ブラウザ側の実装に依存します。このディレクティブを複数指定する場合，同じ拡張子に異なる圧縮形式は指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
AddEncoding x-compress Z     拡張子がZのファイルの圧縮形式はx-compress
AddEncoding x-gzip gz        拡張子がgzのファイルの圧縮形式はx-gzip
```

(10) AddHandler ハンドラ名 拡張子 〔拡張子 …〕

(a) 内容

ハンドラで処理するファイル拡張子を対応付ける場合に定義します。

指定できるハンドラ名を次に示します。このディレクティブを複数指定する場合，同じ拡張子に異なるハンドラ名は指定できません。

cgi-script：CGI スクリプトの実行

imap-file：イメージマップ処理

server-status：ステータスの表示

hws_cache：静的コンテンツのキャッシュ

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
AddHandler cgi-script .cgi                拡張子.cgiはcgi-scriptハンドラ
AddHandler imap-file map                   拡張子mapはimap-fileハンドラ
```

(11) AddIcon {(文字列，URL)｜URL} 拡張子 〔拡張子 …〕

(a) 内容

拡張子などにディレクトリインデクスのアイコンを対応付けて表示する場合に指定します。文字列には画像表示ができない Web ブラウザの場合に表示する文字を指定します。URL にアイコンの画像ファイルの URL を指定します。自ホスト内の画像ファイルを指定する場合，URL の「http：//IP アドレス」は省略できます。なお，URL の「http：//IP アドレス」を省略しないで IPv6 アドレスを指定する場合は，IPv6 アドレスを［ ］で囲んでください。

拡張子として指定できるものを次に示します。

- ファイル拡張子
- ワイルドカード表記のファイル拡張子またはファイル名
- ファイル名

拡張子として ^^DIRECTORY^^ を記述すると，ディレクトリに対するアイコンを設定できます。また，^^BLANKICON^^ と指定すると，ディレクトリインデクスを表示した場合の，表示内容のヘッダのインデントを合わせるためのアイコンを設定できます。

このディレクティブを複数指定する場合，同じ拡張子に異なる文字列や URL は指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
AddIcon /icons/tar.gif .tar
```

拡張子が .tar の場合のアイコン定義

```
AddIcon /icons/layout.gif .html .shtml .htm .pdf
```

拡張子が .html，.shtml，.htm，.pdf の場合のアイコン定義

```
AddIcon /icons/text.gif .txt
```

拡張子が .txt の場合のアイコン定義

```
AddIcon /icons/back.gif ..
```

親ディレクトリのアイコン定義

```
AddIcon /icons/hand.right.gif README
```

README ファイルのアイコン定義

```
AddIcon /icons/folder.gif ^^DIRECTORY^^
```

ディレクトリの場合のアイコン定義

```
AddIcon /icons/blank.gif ^^BLANKICON^^
```

ディレクトリインデクスのヘッダのインデントアイコン定義

```
AddIcon http://[fec0::123:4567:89ab:cdef]/icons/text.gif .txt
```

IPv6 アドレスを指定する場合のアイコン定義

(12) AddIconByEncoding {(文字列，URL)｜URL} MIME エンコーディング 〔MIME エンコーディング …〕

(a) 内容

ディレクトリインデクスの整形表示時のアイコンを MIME エンコーディングと対応付けて表示する場合に指定します。文字列には画像表示ができない Web ブラウザの場合に表示する文字を指定します。URL にアイコンの画像ファイルの URL を指定します。自ホスト内の画像ファイルを指定する場合，URL の「http：//IP アドレス」は省略できます。なお，URL の「http：//IP アドレス」を省略しないで IPv6 アドレスを指定する場合は，IPv6 アドレスを [ ] で囲んでください。

このディレクティブを複数指定する場合，同じ MIME タイプに異なる文字列や URL は指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
AddIconByEncoding (CMP, /icons/compressed.gif) x-compress x-gzip
```

MIME エンコーディング x-compress および x-gzip の場合のアイコン定義

(13) AddIconByType {(文字列，URL)｜URL} MIME タイプ〔MIME タイプ …〕

(a) 内容

ディレクトリインデクスの整形表示時のアイコンを MIME タイプと対応付けて表示する場合に指定します。画像表示ができない Web ブラウザの場合に表示する文字は文字列で指定できます。また，URL で表示するアイコンの画像ファイル名の場所を指定できます。なお，URL の「http：//IP アドレス」を省略しないで IPv6 アドレスを指定する場合は，IPv6 アドレスを［ ］で囲んでください。

このディレクティブを複数指定する場合，同じ MIME タイプに異なるファイル名は指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
AddIconByType (TXT, /icons/text.gif) text/*
```

MIME タイプ　text/* の場合のアイコン定義

```
AddIconByType (IMG, /icons/image2.gif) image/*
```

MIME タイプ　image/* の場合のアイコン定義

```
AddIconByType (SND, /icons/sound2.gif) audio/*
```

MIME タイプ　audio/* の場合のアイコン定義

```
AddIconByType (VID, /icons/movie.gif) video/*
```

MIME タイプ　video/* の場合のアイコン定義

(14) AddLanguage　言語コード　拡張子

(a) 内容

ドキュメントで使用する言語を指定します。言語コードは Content-Language レスポンスヘッダに設定されます。このディレクティブを指定すると，Web ブラウザの言語設定で言語コードの優先順位（Accept-Language ヘッダ）がリクエストに設定されている場合，Web サーバから送信するコンテンツを選択するコンテントネゴシエーションができます。言語コードは Web ブラウザが送信するヘッダ情報に依存します。基本的には，ISO639 に定義されている言語コードに従って指定します。なお，コンテントネゴシエーションを有効にするためには，Options ディレクティブで MultiViews オプションを設定しなければなりません。このディレクティブを複数指定する場合，同じ拡張子に異なる言語コードは指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
AddLanguage ja .ja                日本語
AddLanguage en .en                英語
AddLanguage fr .fr                フランス語
AddLanguage de .de                ドイツ語
AddLanguage da .da                デンマーク語
AddLanguage el .el                ギリシャ語
AddLanguage it .it                イタリア語
```

(15) AddType　MIME タイプ　拡張子 〔拡張子 …〕

(a) 内容

TypesConfig ディレクティブで指定したファイルに未定義のコンテンツの拡張子と MIME タイプを関連づけたい場合に指定します。このディレクティブを複数指定する場合，同じ拡張子に異なる MIME タイプは指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
AddType text/html .shtml
```

MIME タイプ text/html と拡張子 .shtml を関連づけます。

(16) Alias　URL　ディレクトリ名

(a) 内容

Web ブラウザからリクエストされた特定の URL を別名に置き換える場合に指定します。ただし，URL には，? 以降（問い合わせ文字列）を指定できません。URL で指定されたディレクトリを，ディレクトリ名で指定したディレクトリに置き換えて，Web ブラウザに表示します。

ディレクトリ名は，絶対パスで指定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
Alias /icons/ "C:/Program Files/Hitachi/httpsd/icons/"
```

/icons/ を C:/Program Files/Hitachi/httpsd/icons/ に置き換えます。

(17) AliasMatch　正規表現　新パス

(a) 内容

Web ブラウザからリクエストされた URL を別名に置き換える場合に指定します。ただし，URL には，? 以降（問い合わせ文字列）を指定できません。

正規表現で記述した条件を満たす URL が Web ブラウザからリクエストされた場合，指定した新パスのコンテンツを Web ブラウザに表示します。正規表現で括弧（）を使用してグループ化している場合，その i 番目のグループの表現にマッチした文字列を，新パスで $i を使用して参照できます。i には 1 から 9 までの数字を指定します。

新パスは，絶対パスで指定してください。また，新パスの文字として，'$' または '&' を含める場合は，その文字の前に '¥' を付加してください。なお，$i を指定する際には，'$' の前に '¥' を付加する必要はありません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
AliasMatch ^/html/(.*) "C:/htdocs/html/$1"
```

"/html/" で始まるリクエストのとき，/html/ 部分を C:/htdocs/html/ に置き換えます。例えば，/html/index.html へのアクセスの場合，C:/htdocs/html/index.html に置き換えます。

(18) Allow from {ホスト | all | env= 環境変数} [{ホスト | env= 環境変数} …]

(a) 内容

Web サーバへアクセスできるクライアントを制限する場合に指定します。ホストにはアクセスを許可するホストのドメイン名，IP アドレス，サブネット，ネットマスクを指定できます。すべてのホストからのアクセスを許可する場合は all を指定します。

また，ホストには，IPv6 アドレスに関するドメイン名，アドレスおよびプレフィックス長も指定できます。IPv6 アドレスを指定する場合は，IPv6 アドレスを [ ] で囲まないでください。プレフィックス長は，「IPv6 アドレス / プレフィックス長」の形式で指定します。プレフィックス長は 10 進数で指定してください。

env= 環境変数を指定すると，サーバへのアクセスを環境変数の値で制御できます。BrowserMatch，BrowserMatchNoCase，SetEnvIf，SetEnvIfNoCase ディレクティブと併せて使用すれば，HTTP リクエストヘッダフィールドに基づいてアクセスを制限できます。

Allow ディレクティブ（アクセス許可）と Deny ディレクティブ（アクセス制限）は，Order ディレクティブで評価の順序を設定できます。

ホスト	意味
ドメイン名	ドメイン名で指定したホストからのアクセスを許可する。
IP アドレス	IP アドレスで指定したホストからのアクセスを許可する。
サブネット	サブネット（IP アドレスの最初の 1 から 3 バイト）で指定したホストからのアクセスを許可する。
ネットマスク	ネットマスク表記（例 :10.1.0.0/255.255.0.0）で指定したホストからのアクセスを許可する。 10.1.0.0/16 形式で表記した場合，10.1.0.0/255.255.0.0 と同じ意味である。

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

Limit レベル

(d) 指定例

(例 1)

```
SetEnvIf User-Agent Mozilla.* access_ok
<Directory /docroot>
    Order deny,allow
    Deny from all
    Allow from env=access_ok
```

```
</Directory>
```

この場合，User-Agent の文字列が Mozilla を含むブラウザからのリクエストだけがアクセスを許可されて，ほかのアクセスは拒否されます。

(例 2)

ホストに IPv6 アドレスを指定する場合は，次のように指定します。

```
allow from fec0::123:4567:89ab:cdef
```

また，プレフィックス長を指定するとき，次の指定はどれも同じ意味となります。

```
allow from fec0:0:0:1230::/64
allow from FEC0:0:0:1230::/64
allow from fec0::1230:0:0:0:0/64
allow from fec0:0000:0000:1230:0000:0000:0000:0000/64
```

(19) AllowOverride　指示子　〔指示子　…〕

～《All》

(a) 内容

AccessFileName ディレクティブで指定したファイルでアクセス情報定義の上書きを許可するかどうかを設定します。各指示子によって制御できるディレクティブは，各ディレクティブの上書き許可の記述を参照してください。

指示子	内容
AuthConfig	AuthGroupFile，AuthName，AuthType，AuthUserFile，Require ディレクティブなど サーバへのアクセス制御関連のディレクティブの上書きを許可
FileInfo	AddType，AddEncoding，AddLanguage ディレクティブなど コンテンツ管理，MIME タイプ，暗号化などファイル情報関連のディレクティブの上書きを許可
Indexes	FancyIndexing，AddIcon，AddDescription ディレクティブなど ディレクトリインデクス関連のディレクティブの上書きを許可
Limit	Allow from，Deny from，Order ディレクティブ ホスト名または IP アドレスを用いたアクセス制御の上書きを許可
Options	Options ディレクトリの使用を許可
All	すべての上書きを許可する
None	すべての上書きを禁止する

(b) 記述できる場所

<Directory>

(20) AuthAuthoritative {On | Off}

(a) 内容

ユーザ認証をする場合の制御方法を指定します。

On：AuthUserFile，AuthGroupFile，Require ディレクティブの設定によるユーザ認証をします。ユーザ未登録またはパスワード不整合の場合は 401 エラーステータスを Web ブラウザに表示します。

Off：AuthUserFile，AuthGroupFile，Require ディレクティブの設定によるユーザ認証をします。そのとき，パスワード不整合の場合は 401 エラーステータスを Web ブラウザに表示します。さらに，ユーザ未登録の場合には他製品のモジュール（機能）でユーザ認証をします。

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

AuthConfig レベル

(21) AuthGroupFile ファイル名

(a) 内容

グループでユーザ認証をする場合，認証するグループのリストを格納しているファイル名を指定します。ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

グループファイルはテキストエディタを使用して次に示すようなフォーマットで作成してください。

グループ名：ユーザ名〔 ユーザ名 …〕

任意のグループ名に，ユーザ認証のためのパスワードファイルに登録しているユーザ名を定義します。1 行につき 1 グループで指定します。グループファイルには複数グループを定義できます。同じグループ名の行を複数指定した場合には，同じグループ名に登録されているすべてのユーザ名を含んだ一つのグループが定義されます。

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

AuthConfig レベル

(22) AuthName　realm 名

(a) 内容

ユーザ認証する場合の realm 名（Web ブラウザのユーザ認証画面に表示される）を指定します。このディレクティブを指定する場合は AuthType，Require，AuthUserFile（または AuthGroupFile）ディレクティブを必ず指定しなければなりません。ただし，ディレクトリサービスを利用したユーザ認証を行う場合は，AuthUserFile（または AuthGroupFile）ディレクティブの指定は必要ありません。

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

AuthConfig レベル

(23) AuthType　認証タイプ名

(a) 内容

ユーザ認証する場合の認証制御のタイプを指定します。認証タイプ名として "Basic" が指定できます。このディレクティブを指定する場合は AuthName，Require，AuthUserFile（または AuthGroupFile）ディレクティブを必ず指定しなければなりません。ただし，ディレクトリサービスを利用したユーザ認証を行う場合は，AuthUserFile（または AuthGroupFile）ディレクティブの指定は必要ありません。

Basic：Base64 コード変換をします。

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

AuthConfig レベル

(24) AuthUserFile　ファイル名

(a) 内容

ユーザ名でユーザ認証をする場合，認証するユーザ名とパスワードのリストを格納しているファイル名を指定します。

ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

AuthConfig レベル

6.2.3 B，C，D で始まるディレクティブ

(1) BindAddress {IP アドレス | *}

～《*》

(a) 内容

Web サーバをインストールしたサーバ機に割り当てられた IP アドレスのうち，どの IP アドレスから Web サーバに接続できるようにするかを指定します。IP アドレスに IPv6 アドレスは指定できません。どの IPv4 アドレスからも接続できるようにする場合には，* を指定します。Listen ディレクティブを指定した場合は，BindAddress ディレクティブの指定は無視されます。

(b) 記述できる場所

httpsd.conf

(2) BrowserMatch "ブラウザ名" 環境変数〔= 値〕〔環境変数〔= 値〕 …〕

(a) 内容

Web ブラウザごとに環境変数を設定する場合に指定します。設定する値のデフォルト値は 1 です。環境変数の前に ! が付いたときは，その環境変数の設定を解除します。ブラウザ名は正規表現で指定でき，大文字，小文字を区別します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 指定例

```
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4¥.0b2;" nokeepalive downgrade-1.0
force-response-1.0
BrowserMatch "RealPlayer 4¥.0" force-response-1.0
BrowserMatch "Java/1¥.0" force-response-1.0
BrowserMatch "JDK/1¥.0" force-response-1.0
BrowserMatch "Microsoft Data Access Internet Publishing Provider"
redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[012]" redirect-carefully
BrowserMatch "^gnome-vfs" redirect-carefully
```

指定例で示した環境変数の意味を次に示します。

環境変数	内容
nokeepalive	KeepAlive 接続を無効にします。Via ヘッダがリクエストに付加されている場合は，KeepAlive 接続を無効にできません。
downgrade-1.0	HTTP/1.1 以上のリクエストを，HTTP/1.0 のリクエストとして扱います。
force-response-1.0	HTTP/1.0 のリクエストに対して，常に HTTP/1.0 のレスポンスを応答します。
redirect-carefully	ディレクトリへのアクセスで URL の最後に '/' を付加していなく，かつそれが GET メソッド以外を使用していたとき，クライアントにリダイレクトを要求しません。

（3）BrowserMatchNoCase "ブラウザ名" 環境変数〔= 値〕〔環境変数〔= 値〕 …〕

(a) 内容

Web ブラウザごとに環境変数を設定する場合に指定します。設定する値のデフォルト値は 1 です。環境変数の前に ! が付いたときは，その環境変数の設定を解除します。ブラウザ名は正規表現で指定でき，大文字，小文字を区別しません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

（4）CacheNegotiatedDocs 〔{On ｜ Off}〕

(a) 内容

コンテントネゴシエーションをするリクエストで，クライアント側のキャッシュを有効にするかどうかを指定します。ディレクティブの引数を省略した場合は，On を指定した場合と同様の動作をします。ディレクティブを設定しない場合は，Off を指定した場合と同様の動作をします。このディレクティブの指定は，HTTP/1.1 のリクエストに対しては無効です。

On：キャッシュされるようになります。

Off：Expires ヘッダが付けられてキャッシュされなくなります。

(b) 記述できる場所

httpsd.conf

（5）CoreDumpDirectory ディレクトリ名 U

～《ServerRoot ディレクティブ指定値》

(a) 内容

コアをダンプするディレクトリを指定します。絶対パスまたは ServerRoot ディレクティ

ブの指定値からの相対パスが指定できます。なお，指定したディレクトリには，User，Group ディレクティブで指定したユーザ，グループからの書き込み権限を付与する必要があります。Linux 版では，ディレクティブをコンフィグファイルに指定した場合だけ有効となります。

(b) 注意事項

Solaris および HP-UX の場合，ユーザ ID が変更されたプロセスについては，コアはダンプされません。Hitachi Web Server をスーパーユーザで起動すると，User ディレクティブで指定したユーザに変更されます。そのため，コアをダンプする事象が発生した場合でもダンプされません。

(c) 記述できる場所

httpsd.conf

(6) CustomLog {ファイル名｜パイプ} {" フォーマット " ｜ラベル名} 〔env=〔!〕環境変数〕

(a) 内容

任意のフォーマットのログをファイルに出力させる場合に指定します。フォーマットは LogFormat ディレクティブで指定するフォーマットと同様です。

このディレクティブを複数指定する場合，同じファイル名は複数指定できません。

ファイル名：ログの出力先ファイル名を指定します。ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

パイプ：標準入力からログ情報を受け取るプログラムを "| プログラム名 " の形式で指定します。Web サーバはログ情報に含める改行コードを CRLF にして渡します。

（Windows 版での注意事項）

パイプで指定されたプログラムは，制御プロセスと Web サーバプロセス用にログ情報を受け取るそれぞれ別のプロセスとして生成されます。これをパイププロセスと呼びます。サービスとして Web サーバを起動する場合には次の点に注意してください。

- 制御プロセスのログ情報取得不可
  サービスとして Web サーバを起動した場合には，制御プロセスからのログ情報を受け取るための標準入力は NUL デバイスに関連づけられているため，制御プロセス用のパイププロセスは，制御プロセスからのログ情報を受け取ることはできません。制御プロセスからのログ情報とは，Web サーバ起動，停止時のエラーログ情報であり，これらの情報は採取できないことになります。Web サーバ起動後のエラーログ，アクセスログの情報は Web サーバプロセスからのログ情報となりますので，Web サーバプロセス用のパイププロセスで受け取れます。
- プログラム作成時の留意点
  制御プロセス用のパイププロセスは，NUL デバイスからのデータ読み込み処理で，

read() のバッファが小さいと入力データ待ち状態が解除されないことがあります。
read() のバッファを十分大きい値を取るなどしてパイププロセスが入力データ待ち状態にならないようにしてください。

- プログラムに引数を指定する場合の注意
  プログラム，引数に空白を含む場合には，¥" で囲んでください。
  プログラム，引数を ¥" で囲む場合には，全体も ¥" で囲んでください。
  （例）

```
CustomLog "|¥"¥"C:/Program Files/Hitachi/httpsd/sbin/
rotatelogs.exe¥" ¥"C:/Program Files/Hitachi/httpsd/logs/access¥"
86400 -diff 540¥"" common
```

" フォーマット "：ログフォーマットを指定します。指定できるフォーマット名を表 6-3，表 6-4 に示します。

ラベル名：LogFormat ディレクティブで定義したラベル名を指定します。

env= 環境変数：指定した環境変数が設定されている場合に，ログを採取します。

env=! 環境変数：指定した環境変数が設定されていない場合に，ログを採取します。

表 6-3　フォーマット一覧

フォーマット	意味
%A ※1	Web サーバの IP アドレス。
%a ※1	クライアントの IP アドレス。
%B	送信バイト数（HTTP ヘッダおよび chunked エンコーディングによって追加されたデータを除く）。
%b	送信バイト数（HTTP ヘッダおよび chunked エンコーディングによって追加されたデータを除く）。ただし，0 の場合は -（ハイフン）。
%{cookie_name}C	Cookie ヘッダ値に含まれるクッキー名 cookie_name の値。Cookie ヘッダ値に複数の cookie_name が見つかった場合，すべての値を出力する。
%D	リクエスト処理時間をマイクロ秒で表示。
%{env_name}e	env_name に指定した環境変数の値。
%f	クライアントが要求したディレクトリまたはファイル名。
%H	リクエストプロトコル (HTTP/1.0 など)。
%h ※2	クライアントのホスト名。
%I	リクエストとヘッダを含む，全受信バイト数。
%{header_name}i	header_name に指定した HTTP 通信によるリクエスト中の http プロトコルヘッダの値。
%l	クライアントの識別情報（IdentityCheck ディレクティブが On，かつクライアント上で identd が動作している場合）。

フォーマット	意味
%m	リクエストメソッド (GET，POST など)。
%{note_name}n	note_name に指定した Web サーバ内モジュールの注記の値。
%O	ヘッダを含む，全送信バイト数。
%{header_name}o	header_name に指定した HTTP 通信で送信中の http プロトコルヘッダの値。
%P	HTTP 通信のリクエストを処理するプロセス ID。
%{hws_thread_id}P	HTTP 通信のリクエストを処理するスレッド ID。Windows 版で有効。
%p	ポート番号。
%q	問い合わせ文字列。
%r	HTTP 通信のリクエストの先頭行。
%s	ステータス（内部リダイレクトされた場合はオリジナルを示す)。
%T	リクエスト処理に掛かった時間（秒)。HWSLogTimeVerbose ディレクティブで On を指定すると，ミリ秒単位まで表示。
%t	リクエスト処理を開始した時刻。HWSLogTimeVerbose ディレクティブで On を指定すると，ミリ秒単位まで表示。
%{format}t	リクエスト処理を開始した時刻。strftime() で定義されているフォーマットを format に指定する。
%U	URL。
%u	クライアントのユーザ名（ユーザ認証をした場合)。
%V※2	UseCanonicalName ディレクティブの指定に従い，ServerName ディレクティブ指定値，サーバ名または IP アドレス。
%v	サーバ名。
%X	レスポンス完了時の接続ステータス。 ＋：レスポンス送信後も接続を維持する。 －：レスポンス送信後に接続を切断する。 X：レスポンス完了前に接続を切断する。
%>s	最終ステータス。

注

フォーマットで示す{ }は選択を意味するものではありません。{ }内の太字はログを採取する変数名を，細字は文字列そのままを記述します。

注※ 1

フォーマットに %A または %a を指定した場合，IPv6 アドレスも出力できます。

注※ 2

フォーマットに %h または %V を指定した場合，IPv6 アドレスに対応したホスト名または IPv6 アドレスも出力できます。

表 6-4 SSL 関連のログフォーマット一覧

フォーマット	意味
%{version}c	SSL のバージョン
%{cipher}c	現在の通信で使用している暗号種別
%{clientcert}c	SSL クライアント証明書の subject の Distinguished Name

フォーマットの % の後ろにステータスコードを記述できます。

（例）エラーステータスコード 400 および 501 の場合，http プロトコルヘッダの User-Agent 値のログを採取する。

```
%400,501 {User-Agent}i
```

（例）エラーステータスコード 200，304 および 302 の 3 種類以外の場合，http プロトコルヘッダの Referer 値のログを採取する。

```
%!200,304,302 {Referer}i
```

また，env= は，指定した環境変数の設定によって，ログの採取を分ける場合に指定します。

（例）gif へのアクセスは gif.log に，gif 以外へのアクセスは nongif.log にログを採取する。

```
SetEnvIf Request-URI ¥.gif$ gif-image
CustomLog gif.log common env=gif-image
CustomLog nongif.log common env=!gif-image
```

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
CustomLog logs/access.log common
CustomLog logs/ssl.log "%t %{version}c %{cipher}c %{clientcert}c"
```

(7) DefaultIcon　URL

(a) 内容

ディレクトリインデクスで表示するアイコンを指定します。AddIcon，AddIconByType および AddIconByEncoding ディレクティブのどれにも該当しない場合に表示するアイコンの URL を指定します。なお，URL の「http：//IP アドレス」を省略しないで IPv6 アドレスを指定する場合は，IPv6 アドレスを [　] で囲んでください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
DefaultIcon /icons/unknown.gif
```

(8) DefaultLanguage 言語コード

(a) 内容

ドキュメントで使用するデフォルトの言語を指定します。指定した言語コードは Content-Language レスポンスヘッダに設定されます。AddLanguage ディレクティブの設定に対するデフォルト値となります。デフォルト値が設定されていない場合，Content-Language レスポンスヘッダは送信しません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(9) DefaultType MIME タイプ

～《text/plain》

(a) 内容

TypesConfig ディレクティブで指定したファイルで定義した MIME タイプのどれにも該当しないコンテンツに対して使用する MIME タイプ名を指定します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
DefaultType text/plain
```

(10) Deny from {ホスト | all | env= 環境変数} 〔{ホスト | env= 環境変数} …〕

(a) 内容

Web サーバへアクセスできるクライアントを制限する場合に指定します。ホストにはアクセスを禁止するホストのドメイン名，IP アドレス，サブネット，ネットマスクを指定できます。すべてのホストからアクセスを禁止する場合は，all を指定します。

また，ホストには，IPv6 アドレスに関するドメイン名，アドレスおよびプレフィックス長も指定できます。IPv6 アドレスを指定する場合は，IPv6 アドレスを [ ] で囲まないでください。プレフィックス長は，「IPv6 アドレス / プレフィックス長」の形式で指定します。プレフィックス長は 10 進数で指定してください。

env= 環境変数を指定すると，サーバへのアクセスを環境変数の値で制御できます。BrowserMatch，BrowserMatchNoCase，SetEnvIf，SetEnvIfNoCase ディレクティブと併せて使用すれば，HTTP リクエストヘッダフィールドに基づいてアクセスを制限できます。

Allow ディレクティブ（アクセス許可）と Deny ディレクティブ（アクセス制限）は，Order ディレクティブで評価の順序を設定できます。

ホスト	意味
ドメイン名	ドメイン名で示すホストからのアクセスを禁止する。
IP アドレス	IP アドレスで示すホストからのアクセスを禁止する。
サブネット	サブネット（IP アドレスの最初の 1 から 3 バイト）で指定したホストからのアクセスを禁止する。
ネットマスク	ネットマスク表記（例：10.1.0.0/255.255.0.0）で指定したホストからのアクセスを禁止する。 10.1.0.0/16 形式で表記した場合 10.1.0.0/255.255.0.0 と同じ意味である。

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

Limit レベル

(11) DirectoryIndex ファイル名 〔ファイル名 …〕

～《index.html》

(a) 内容

Web ブラウザからのリクエストが特定のコンテンツを指定していない場合に，デフォルトとしてクライアントに送信するコンテンツのファイル名を指定します。ファイル名を複数指定した場合は，先に指定したファイル名を優先して送信します。

ここで指定したファイルがリクエストされたディレクトリにない場合，Options ディレクティブの指定によって Web ブラウザの表示が変わります。

- Indexes が有効の場合
  Web ブラウザに Web サーバで作成したディレクトリのインデクスを表示します。
- Indexes が無効の場合

ステータスコード 403 Forbidden を応答します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
DirectoryIndex index.html
```

ファイル名の指定がないリクエストの場合，ディレクトリに index.html があれば表示させます。

(12) DocumentRoot　ディレクトリ名

～《/opt/hitachi/httpsd/htdocs》(UNIX 版)

～《ServerRoot ディレクティブのデフォルト値 /htdocs》(Windows 版)

(a) 内容

コンテンツを格納するドキュメントルートディレクトリを絶対パスで指定します。ディレクトリ名の終端には /（スラッシュ）を記述しないでください。

ディレクトリ名は，絶対パスで指定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
DocumentRoot "C:/Program Files/Hitachi/httpsd/htdocs"
```

6.2.4 E，F，G，H，I で始まるディレクティブ

(1) ErrorDocument　エラーステータス番号　{テキスト｜ローカル URL ｜フル URL}

(a) 内容

エラーが発生したときに，Web ブラウザへ表示するメッセージをカスタマイズする場合に指定します。

テキスト：文字列を " で囲み指定します。

ローカル URL：先頭に / を記述して，自サイト内のコンテンツを指定します。

フル URL：http:// または https:// で始まる URL を記述し，他サイトのコンテンツを指

定します。

このディレクティブに指定できるエラーステータス番号と，テキスト，ローカル URL，フル URL の指定可否について，次に示します。

エラーステータス番号（意味）	テキスト	ローカル URL	フル URL
400 (Bad Request)	○	×	×
401 (Authorization Required)	○	○	×
403 (Forbidden)	○	○	○
404 (Not Found)	○	○	○
405 (Method Not Allowed)	○	○	○
406 (Not Acceptable)	○	○	○
408 (Request Time-out)	×	×	×
410 (Gone)	○	○	○
411 (Length Required)	○	×	×
412 (Precondition Failed)	○	○	○
413 (Request Entity Too Large)	○	○	○
414 (Request-URI Too Large)	○	×	×
416 (Requested Range Not Satisfiable)	○	○	○
417 (Expectation Failed)	○	×	×
500 (Internal Server Error)	○	○	○
501 (Method Not Implemented)	○	○	○
502 (Bad Gateway)	○	○	×
503 (Service Temporarily Unavailable)	○※	○※	○※
506 (Variant Also Negotiates)	○	○	○

（凡例）

○：指定できる。

×：指定できない。

注※

流量制限機能が返すメッセージをカスタマイズする場合は，QOSResponse ディレクティブまたは QOSRedirect ディレクティブを使用してください。

このディレクティブ指定時には，次の点に留意してください。

- このディレクティブを複数指定する場合，同じエラー番号に異なる指定はできません。
- CGI プログラム内で設定されたエラーステータスに対しては，メッセージをカスタマイズできません。
- ローカル URL，フル URL の指定先でエラーとなる場合は，カスタマイズできません。
- ローカル URL の指定先でコンテントネゴシエーションが発生する場合は，エラーと

なりカスタマイズできないことがあります。
- LoadModule ディレクティブによって動的に接続したモジュール内で設定されたエラーステータスに対しても，そのモジュールの実装方法によってメッセージをカスタマイズできない場合があります。
- フル URL の指定時には，ステータスコード 302 Found および Location ヘッダに新パスを設定した応答を返します。通常，ステータスコード 302 を受けた Web ブラウザは，Location ヘッダに指定されたアドレスに対して自動的にリダイレクトします。
- フル URL の指定時には，IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。IPv6 アドレスを指定する場合は，IPv6 アドレスを [ ] で囲んでください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
ErrorDocument 500 "Server Error."
ErrorDocument 404 /missing.html
ErrorDocument 403 http://some.other_server.com/
subscription_info.html
ErrorDocument 404 http://[fec0::123:4567:89ab:cdef]/missing.html
```

(2) ErrorLog {ファイル名|パイプ}

～《logs/error_log》(UNIX 版)

～《logs/error.log》(Windows 版)

(a) 内容

エラーログを出力するファイル名を指定します。出力するログの内容は，LogLevel ディレクティブで選択できます。

ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

ファイル名：エラーログを格納するファイル名を指定します。ServerRoot ディレクティブ指定値からの相対パスで指定できます。

パイプ：標準入力からエラーログ情報を受け取るプログラムを "| プログラム名 " の形式で指定します。Windows 版での注意事項は，CustomLog ディレクティブを参照してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
ErrorLog logs/error.log
```

(3) ExtendedStatus {On | Off}

(a) 内容

server-status ハンドラによるステータス表示形式で，それぞれのリクエストの拡張ステータス情報を表示するかどうかを指定します。

On：拡張ステータス情報を表示します。この場合，クライアントの IP アドレスが IPv6 アドレスでも表示します。ただし，最大表示数は 31 バイトです。

Off：拡張ステータス情報を表示しません。

(b) 記述できる場所

httpsd.conf

(4) ExpiresActive {On | Off}

(a) 内容

レスポンスに Expires ヘッダおよび Cache-Control ヘッダを追加するかどうかを指定します。

On：Expires ヘッダおよび Cache-Control ヘッダを追加します。

Off：Expires ヘッダおよび Cache-Control ヘッダを追加しません。

(b) 注意事項

- 有効期限設定機能を使用するためには mod_expires モジュールの組み込みが必要です。有効期限設定機能の詳細は，「4.11 有効期限設定機能」を参照してください。
- ExpiresDefault ディレクティブまたは ExpiresByType ディレクティブを指定していない場合は，ExpiresActive ディレクティブに On が指定されていても，レスポンスに Expires ヘッダおよび Cache-Control ヘッダは追加されません。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(d) 上書き許可

Indexes レベル

(5) ExpiresByType MIME タイプ {A | M} 時間

～ ((0 － 2147483647))（単位：秒）

(a) 内容

レスポンスに Expires ヘッダおよび Cache-Control ヘッダを追加する場合に，指定する

MIME タイプのドキュメントに対する有効期限を指定します。このディレクティブは ExpiresActive ディレクティブで On を指定している場合に有効になります。ExpiresDefault ディレクティブで設定されたデフォルトの有効期限は，この設定によって MIME タイプ別に上書きされます。

基準時刻を A または M で指定し，基準時刻から有効期限までの時間を秒単位で指定します。A または M と，時間との間に空白は入りません。

A：クライアントがアクセスした時刻を基準時刻とします。

M：ファイルを最後に修正した時刻を基準時刻とします。

(b) 注意事項

- 有効期限設定機能を使用するためには mod_expires モジュールの組み込みが必要です。有効期限設定機能の詳細は，「4.11　有効期限設定機能」を参照してください。
- グリニッジ標準時（GMT）の 2038 年 1 月 19 日 3 時 14 分 7 秒を超えないように，有効期限を設定してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(d) 上書き許可

Indexes レベル

(e) 指定例

```
ExpiresByType text/html A604800
```

(6) ExpiresDefault {A | M} 時間

～ ((0 － 2147483647))（単位：秒）

(a) 内容

レスポンスに Expires ヘッダおよび Cache-Control ヘッダを追加する場合に，デフォルトの有効期限を指定します。このディレクティブは ExpiresActive ディレクティブで On を指定している場合に有効になります。この設定は ExpiresByType ディレクティブによって MIME タイプごとに上書きされます。

基準時刻を A または M で指定し，基準時刻から有効期限までの時間を秒単位で指定します。A または M と，時間との間に空白は入りません。

A：クライアントがアクセスした時刻を基準時刻とします。

M：ファイルを最後に修正した時刻を基準時刻とします。

(b) 注意事項

- 有効期限設定機能を使用するためには mod_expires モジュールの組み込みが必要で

す。有効期限設定機能の詳細は，「4.11　有効期限設定機能」を参照してください。
- グリニッジ標準時（GMT）の 2038 年 1 月 19 日 3 時 14 分 7 秒を超えないように，有効期限を設定してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(d) 上書き許可

Indexes レベル

(e) 指定例

```
ExpiresDefault A604800
```

(7) FancyIndexing　{On ｜ Off}

(a) 内容

ディレクトリインデクスを表示する場合に，整形表示（ファンシーインデクス）をするかどうかを指定します。

On：整形表示をします。

Off：整形表示をしません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
FancyIndexing On
```

整形表示機能を使用します。

(8) FileETag　〔{+ ｜ -}〕オプション　〔{{+ ｜ -}〕オプション …〕

～《All》

(a) 内容

ETag レスポンスヘッダフィールドを作成するために使用されるファイル属性値を指定します。このディレクティブが指定されていない場合，ETag レスポンスヘッダフィールドにはファイルに割り振られた一意な ID，最終更新時刻およびバイト数が設定されます。

オプションに＋－を指定しない場合は，オプションで指定した属性値が使用されます。

オプションに＋－を指定する場合は，FileETag ディレクティブによって設定された属性値を変更できます。

＋：設定されている属性値にオプションで指定した属性値が追加されます。

－：設定されている属性値からオプションで指定した属性値が削除されます。

指定できるオプションの一覧を次に示します。

オプション	意味
Inode	ファイルに割り振られた一意な ID が含まれます。
Mtime	ファイルの最終更新時刻が含まれます。
Size	ファイルのバイト数が含まれます。
All	Inode，Mtime，Size のオプションがすべて有効になります。
None	Etag ヘッダが付きません。

(b) 注意事項

- FileETag ディレクティブの Inode オプションを有効にした場合，負荷分散をしている Web サーバ環境などで，同一のコンテンツを要求するごとに，異なる ID が Etag ヘッダに含まれることがあります。このため，同一コンテンツでありながらその Etag ヘッダの内容が異なり，ブラウザやプロキシでのキャッシングにとって不都合となることがあります。この場合，FileETag ディレクティブによって，Inode オプションを無効にするように指定することで回避できます。
- ＋－を使用しないでこのディレクティブを複数指定すると，最後に指定したディレクティブだけが有効になります。
- －を付加した属性値だけを指定した場合は，All オプションを指定した場合と同じ動作になります。
- All オプションと None オプションには，＋－を指定できません。
- オプションに '-Inode -Mtime -Size' と指定した場合は，このディレクティブを指定していない場合と同じ状態になります。ETag レスポンスヘッダフィールドにはファイルの inode 番号，最終更新時刻およびバイト数が設定されます。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(d) 上書き許可

FileInfo レベル

(e) 指定例

(例 1)

```
FileETag Inode Mtime Size
FileETag -Inode
```

この指定では，ファイルの最終更新時刻およびバイト数が属性値として使用されます。

(例 2)

```
FileETag Inode Mtime
FileETag Size
```

この指定では，ファイルのバイト数が属性値として使用されます。

(例 3)

```
FileETag All
FileETag -Inode -Mtime -Size
```

この指定では，ファイルの一意な ID，最終更新時刻およびバイト数が属性値として使用されます。

(9) ForceType　MIME タイプ

(a) 内容

<Directory> ブロックまたはアクセスコントロールファイルに定義し，特定のディレクトリ下のすべてのコンテンツに対して使用する MIME タイプを指定します。none を指定すると，それまでの ForceType ディレクティブの指定が無効になります。

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(10) Group　グループ名 

～《#-1》

(a) 内容

サーバプロセスが動作するときのグループ名を指定します。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
Group nogroup                          グループ名nogroupを定義
```

(11) Header {{set | append | add} ヘッダ　ヘッダ値〔env=〔!〕環境変数〕 | unset　ヘッダ}

(a) 内容

200 番台のステータスコード応答時のレスポンスヘッダをカスタマイズする場合に指定します。リバースプロキシとして使用する場合，バックエンドの Web サーバが返すステータスコードの値にかかわらず，レスポンスヘッダをカスタマイズします。

set：ヘッダを設定します。ヘッダがある場合は，指定したヘッダ値に書き換えます。

append：存在するヘッダにヘッダ値を追加します。存在するヘッダ値との間は，コンマで区切られます。ヘッダがない場合は，ヘッダを設定します。

add：ヘッダがあっても，別の行にヘッダを設定します。同じヘッダを複数行設定する場合に使用します。

unset：指定したヘッダがある場合，そのヘッダをすべて削除します。

env= 環境変数：指定した環境変数が設定されている場合に，Header ディレクティブで指定した内容を実行します。

env=! 環境変数：指定した環境変数が設定されていない場合に，Header ディレクティブで指定した内容を実行します。

ヘッダ値に空白がある場合は，"（引用符）で囲む必要があります。ヘッダ値は文字だけから成る文字列，フォーマット指示子を含む文字列または両方から成る文字列を指定できます。フォーマット指示子を次に示します。

フォーマット	意味
%t	リクエストを受け取った時刻を，1970 年 1 月 1 日 0 時 0 分 0 秒 (GMT：Greenwich Mean Time) から経過した時間で表示する。単位はマイクロ秒。先頭には "t=" が付けられる。
%D	リクエスト処理に掛かった時間を表示する。単位はマイクロ秒。先頭には "D=" が付けられる。
%{env_name}e	環境変数 env_name の値。

(b) 注意事項

ヘッダカスタマイズ機能を使用するためには mod_headers モジュールの組み込みが必要です。ヘッダカスタマイズ機能については，「4.10　ヘッダカスタマイズ機能」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(d) 上書き許可

FileInfo レベル

(e) 指定例

```
Header set Cache-Control no-cache
```

(12) HeaderName　ファイル名

(a) 内容

ディレクトリインデクス表示時のヘッダに付けるコメントを記述したファイルのファイル名（パス情報なし）を指定します。HTML またはプレーンテキストで記述できます。ただし，AddType ディレクティブまたは TypesConfig ディレクティブで指定したファイルで，MIME タイプが正しく定義されている必要があります。プレーンテキストでコメントを作成した場合，ディレクトリインデクス表示の HTML には <PRE> タグが追加されます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
HeaderName HEADER.html
```

各ディレクトリ下の HEADER.html の内容をヘッダに付けます。

(13) HostnameLookups　{On ｜ Off ｜ double}

(a) 内容

CGI の REMOTE_HOST 環境変数の IP アドレスおよびログファイルに出力するクライアントの IP アドレスをホスト名に変換するために，ホスト名のルックアップの逆引きをするかどうかを指定します。なお，逆引きを使用する場合，レスポンスが遅くなります。

On：IP アドレスをホスト名に変換します。

Off：IP アドレスをホスト名に変換しません。

double：IP アドレスをホスト名に変換します。その後，再変換し，IP アドレスが正しいかどうかを確認します。

このディレクティブは，IPv6 アドレスにも対応しています。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>

(c) 指定例

```
HostnameLookups Off
```

IP アドレスをホスト名に変換しません。

(14) HWSContentCacheMaxFileSize　サイズ　W

～((1 － 2097093))《256》(単位：KB)

(a) 内容

キャッシュ可能なファイルのサイズの上限値を KB 単位で指定します。
HWSContentCacheMaxFileSize ディレクティブに HWSContentCacheSize ディレクティブの値よりも大きな値を設定した場合には，HWSContentCacheSize ディレクティブの値が設定されます。

(b) 注意事項

静的コンテンツキャッシュ機能を使用するためには mod_hws_cache モジュールの組み込みが必要です。静的コンテンツキャッシュ機能については，「4.12　静的コンテンツキャッシュ機能」を参照してください。

(c) 記述できる場所

httpsd.conf

(d) 指定例

```
HWSContentCacheMaxFileSize 32
```

(15) HWSContentCacheSize　サイズ　W

～((1 － 2097093))《8192》(単位：KB)

(a) 内容

サーバプロセス内にキャッシュするデータのメモリサイズの上限値を KB 単位で指定します。

(b) 注意事項

静的コンテンツキャッシュ機能を使用するためには mod_hws_cache モジュールの組み込みが必要です。静的コンテンツキャッシュ機能については，「4.12　静的コンテンツキャッシュ機能」を参照してください。

(c) 記述できる場所

httpsd.conf

(d) 指定例

```
HWSContentCacheSize 1024
```

(16) HWSErrorDocumentMETACharset　{On ｜ Off ｜文字セット}　W

(a) 内容

エラーが発生したときに Web ブラウザへ表示するメッセージ（以降，エラードキュメン

トと呼びます）についての文字セットを設定します。文字セットは，エラードキュメント中に META タグで charset= の値として設定されます。ErrorDocument ディレクティブで，カスタマイズされたエラードキュメントは，このディレクティブの META タグによる文字セットの設定対象とはなりません。

On：文字セット ISO-8859-1 を設定します。

Off：文字セットを設定しません。

文字セット：指定した文字セットを設定します。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
HWSErrorDocumentMETACharset ISO-2022-JP
```

(17) HWSGracefulStopLog {On | Off}

(a) 内容

計画停止時に，強制停止待ち時間を経過した後に強制停止させたリクエスト情報を，エラーログファイルに出力するかどうかを指定します。

On：強制停止させたリクエスト情報をエラーログファイルに出力します。

Off：強制停止させたリクエスト情報をエラーログファイルに出力しません。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
HWSGracefulStopLog On
```

(18) HWSGracefulStopTimeout 強制停止時間

～ ((0 － 3600))《300》（単位：秒）

(a) 内容

計画停止時に，実行中のリクエストを直ちに終了するまでの強制停止待ち時間を秒単位で指定します。なお，0 を指定すると，強制停止待ち時間の上限は設定されません。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
HWSGracefulStopTimeout 600
```

(19) HWSImapMenuCharset　文字セット

～《ISO-8859-1》

(a) 内容

次の場合のメニュー表示に対する文字セットを指定します。

- イメージマップファイルの指定値に map を指定した場合
- イメージマップ画像の座標 (0,0) をマウスでポイントした場合
- 座標指定のない形でイメージマップファイルがリクエストされた場合

文字セットは，レスポンスの Content-Type ヘッダで charset= の値として設定されます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
HWSImapMenuCharset SHIFT_JIS
```

(20) HWSKeepStartServers　{On ｜ Off}

(a) 内容

サーバプロセスの稼働数を StartServers ディレクティブに指定した数だけ維持するかどうかを指定します。

On：StartServers ディレクティブに指定した数だけ，稼働しているサーバプロセスが維持されます。サーバプロセス数が StartServers ディレクティブ指定値より小さくなった場合，新しいプロセスが生成されます。この機能は，プロセス数に関する各ディレクティブの指定値が，次の関係にある場合に有効です。

MinSpareServers < StartServers ≦ MaxClients

かつ

MinSpareServers < MaxSpareServers ≦ MaxClients

StartServers ディレクティブ設定値が，MinSpareServers ディレクティブ設定値より小さい場合は，MinSpareServers ディレクティブの値でサーバプロセス数が維持されます。

Off：StartServers ディレクティブに指定した数の稼働しているサーバプロセスは維持されません。

プロセス数に関連するほかのディレクティブについては，「4.1　Hitachi Web Server の処理とディレクティブとの関係」を参照してください。

(b) 記述できる場所

httpsd.conf

(21) HWSLogSSLVerbose {On ｜ Off}

(a) 内容

クライアントとサーバ間の SSL ハンドシェイク処理中に，ログに出力されるエラーのうち info レベルおよび error レベルのエラーについて，詳細情報を表示するかどうかを指定します。SSL を有効にする場合には，このディレクティブを On に設定することを推奨します。

On：詳細情報を表示します。

Off：詳細情報を表示しません。

(b) 記述できる場所

httpsd.conf

(22) HWSLogTimeVerbose {On ｜ Off}

(a) 内容

エラーログ※とリクエストログの時刻，アクセスログのアクセス時刻，リクエスト処理に掛かった時間 (%T)，およびリクエスト処理を開始した時刻 (%t) をミリ秒まで表示するかどうかを指定します。

注※ ErrorLog ディレクティブで指定するエラーログが対象になります。ScriptLog ディレクティブで指定する CGI スクリプトのエラーログは対象になりません。

On：時刻および時間をミリ秒まで表示します。

Off：時刻および時間を秒まで表示します。

(b) 記述できる場所

httpsd.conf

(23) HWSMaxQueueSize リクエストキューサイズ W

～ ((0 － 2147483647))《8192》

(a) 内容

クライアントからのリクエストについての最大の待ちリクエスト数を指定します。0 を指定した場合は，無制限となります。このディレクティブで指定したリクエストキューサイズを超えたクライアントからのリクエストは，サーバ側で切断されます。

(b) 記述できる場所

httpsd.conf

(24) HWSNotModifiedResponseHeaders ヘッダ名〔ヘッダ名 …〕

(a) 内容

ステータスコード 304 Not Modified をクライアントへ送信する際に付加するレスポンスヘッダを指定します。

なお，次のヘッダについては，このディレクティブに指定がなくてもレスポンスに付加します。ただし，必ず付加するのではなく，外部モジュールまたはサーバ内部などで設定された場合にだけ付加します。

- Date
- Server
- Connection
- Keep-Alive
- ETag
- Content-Location
- Expires
- Cache-Control
- Vary
- Warning
- WWW-Authenticate
- Proxy-Authenticate

(b) 記述できる場所

httpsd.conf

(c) 指定例

HWSNotModifiedResponseHeaders Set-Cookie Set-Cookie2

(25) HWSProxyPassReverseCookie パス名

(a) 内容

リバースプロキシを使用する場合，リバースプロキシはバックエンドサーバから受信した Set-Cookie ヘッダを変換します。これは，Web ブラウザが Set-Cookie ヘッダを受信したあとに，リバースプロキシを経由するバックエンドサーバへのリクエストに対して，クッキーを送信させるために必要になります。

パス名：ProxyPass ディレクティブと同じパス名を指定します。

(b) 注意事項

リバースプロキシを使用するためには mod_proxy モジュールおよび mod_proxy_http モジュールの組み込みが必要です。リバースプロキシの詳細は，「4.7　リバースプロキシの設定」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>

(26) HWSRequestLog　{ファイル名｜パイプ}

(a) 内容

リクエストログを出力するファイル名を指定します。リクエストログとは，モジュールトレース，リクエストトレースおよび I/O フィルタトレースの総称です。出力するリクエストログの種別は，HWSRequestLogType ディレクティブで選択できます。

ファイル名：リクエストログを出力するファイル名を指定します。ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスを指定できます。

パイプ：標準入力からリクエストログ情報を受け取るプログラムを「｜プログラム名」の形式で指定します。Windows 版での注意事項は，CustomLog ディレクティブを参照してください。

(b) 注意事項

- このディレクティブを省略した場合のモジュールトレース出力先は，ErrorLog ディレクティブで指定したファイルになります。モジュールトレースの採取レベルは，LogLevel ディレクティブで指定してください。モジュールトレースの詳細は，「4.2.6　モジュールトレースの採取」を参照してください。
- リクエストトレースと I/O フィルタトレースの出力先を，ErrorLog ディレクティブで指定したファイルにすることはできません。

(c) 記述できる場所

httpsd.conf

(27) HWSRequestLogType　トレース種別 〔トレース種別 …〕

～《module-info request》

(a) 内容

HWSRequestLog ディレクティブで設定するリクエストログに出力するトレース種別を指定します。トレース種別を次に示します。

トレース種別	内容
module-debug	内部モジュールに対するモジュールトレースおよび module-info 相当のトレースを出力します。出力量が多いため，デバッグ目的以外では指定しないでください。
module-info	外部モジュールと CGI プログラム実行時のモジュールトレースを出力します。
request	リクエスト処理開始時およびリクエスト処理完了時にトレースを出力します。また，KeepAlive 接続の場合は，次のリクエストライン受信完了時にもトレースを出力します。これらのトレースをリクエストトレースと呼びます。
filter	モジュールが実装している入出力フィルタ関数の実行契機を示す I/O フィルタトレースを出力します。出力量が多いため，デバッグ目的以外では指定しないでください。
none	リクエストログを採取しません。

(b) 注意事項

指定したトレース種別に none が含まれている場合，リクエストログを一切採取しません。

(c) 記述できる場所

httpsd.conf

(28) HWSSetEnvIfIPv6 リクエスト値 IPv6 アドレス 環境変数〔= 値〕〔環境変数〔= 値〕…〕

(a) 内容

クライアントまたはサーバの IPv6 アドレスを基に環境変数を定義します。リクエスト値が IPv6 アドレスで表した条件を満たす場合，指定した環境変数を設定します。設定する値のデフォルト値は 1 です。環境変数の前に「!」が付いたときは，その環境変数の設定を解除します。

リクエスト値として，次に示す値を指定できます。

リクエスト値	意味
Remote_Addr	クライアントの IPv6 アドレス
Server_Addr	リクエストを受信したサーバの IPv6 アドレス

IPv6 アドレスは，[ ] で囲まないで指定してください。なお，IPv6 アドレスの後に，10 進数でプレフィックス長も指定できます。プレフィックス長は，「IPv6 アドレス / プレフィックス長」の形式で指定します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
HWSSetEnvIfIPv6 Remote_Addr fec0:0:0:1230::/64 IPV6_CLIENT
```

クライアントの IPv6 アドレスが fec0:0:0:1230 から始まる場合，環境変数 IPV6_CLIENT を設定します。

(29) HWSStackTrace {On | Off} U

(a) 内容

サーバプロセスが異常終了した場合に，スタックトレースの内容をエラーログファイルに出力するかどうかを指定します。HP-UX 版だけで有効です。

On：スタックトレースの内容をエラーログファイルに出力します。

Off：スタックトレースの内容をエラーログファイルに出力しません。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
HWSStackTrace On
```

(30) HWSSuppressModuleTrace モジュールファイル名 〔all | hook | handler〕

(a) 内容

モジュールトレースの出力を抑止するモジュールファイル名および抑止する関数種別を指定します。

all：指定したモジュールが出力するモジュールトレースをすべて抑止します。

hook：指定したモジュールが出力するモジュールトレースのうち，handler 関数以外のモジュールトレースを抑止します。関数の種別については，「4.2.6　モジュールトレースの採取」の表 4-4 を参照してください。

handler：指定したモジュールが出力するモジュールトレースのうち，handler 関数のモジュールトレースを抑止します。関数の種別については，「4.2.6　モジュールトレースの採取」の表 4-4 を参照してください。

モジュールファイル名には，エラーログまたはリクエストログに出力されるモジュールファイル名称を指定します。次の例のモジュールトレースを抑止する場合は，モジュールファイル名に "mod_example.c" を指定します。

(例)

```
[Mon Dec 18 14:57:14 2006] [info] hws : module -->
(mod_example.c[12])(1896)
[Mon Dec 18 14:57:14 2006] [info] hws : module <--
(mod_example.c[12])(1896)(-1)
```

Hitachi Web Server が標準提供している外部モジュールとモジュールファイル名の対応を次に示します。

表 6-5 Hitachi Web Server が標準提供している外部モジュールとモジュールファイル名の対応

モジュール名	モジュールファイル名
mod_expires.so	mod_expires.c
mod_headers.so	mod_headers.c
mod_hws_cache.so	mod_hws_cache.c
mod_hws_ldap.so	mod_hws_ldap.c
mod_hws_qos.so	mod_hws_qos.c
mod_proxy.so	mod_proxy.c
mod_proxy_http.so	モジュールトレースは出力されません。

Hitachi Web Server が標準提供している外部モジュール以外を使用する場合は，そのモジュールのトレースが出力される可能性があります。また，LogLevel ディレクティブに debug を設定または HWSRequestLogType ディレクティブに module-debug を設定している場合は，内部モジュールに対するトレースも出力されます。

なお，このディレクティブは，複数指定できます。同じモジュールファイル名を指定した場合は，後に指定したものが有効となります。

(b) 注意事項

CGI プログラム実行時のモジュールトレースは抑止できません。

(c) 記述できる場所

httpsd.conf

(d) 指定例

(例 1)

```
HWSSuppressModuleTrace mod_proxy.c all
```

この指定では，プロキシモジュール内のすべての関数に対するモジュールトレースを抑止します。

(例 2)

```
HWSSuppressModuleTrace mod_proxy.c hook
```

この指定では，プロキシモジュール内の handler 以外の関数に対するモジュールト

レースを抑止します。

(31) HWSTraceIdFile　ファイル名

～《logs/hws.trcid》

(a) 内容

トレース採取のための共有メモリ ID を格納するファイル名を指定します。ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

このファイルは複数の Web サーバでは共有できません。同一 ServerRoot ディレクティブ指定で複数の Web サーバを起動する場合は，このディレクティブで異なるファイル名を指定する必要があります。

(b) 記述できる場所

httpsd.conf

(32) HWSTraceLogFile　ファイル名

～《logs/hws.trclog》

(a) 内容

サーバプロセスが異常終了した場合に共有メモリに採取されたトレースを出力するファイル名を指定します。ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

トレースは複数のファイルにラップアラウンドして出力します。

UNIX 版では，最大 5 ファイル出力します。出力するファイルは「指定したファイル名 .nn」のファイル名となります。nn は 01 から 05 までです。Hitachi Web Server の起動時には，「指定したファイル名 .01」がカレントの出力ファイル名となります。カレントの出力ファイル名が「指定したファイル名 .nn」であった場合にトレースをファイルに出力すると，次のカレントのファイル名は「指定したファイル名 .nn+1」になります。なお，「指定したファイル名 .nn」が .05 の場合には，次のカレントのファイル名は「指定したファイル名 .01」になります。

Windows 版では，最大 2 ファイル出力します。出力するファイルは「指定したファイル名 .01」または「指定したファイル名 .02」のファイル名になります。Hitachi Web Server の起動時には，「指定したファイル名 .01」がカレントの出力ファイル名となります。カレントの出力ファイル名が「指定したファイル名 .01」のときにトレースをファイルに出力すると，次のカレントのファイル名は「指定したファイル名 .02」になります。なお，カレントの出力ファイル名が「指定したファイル名 .02」のときにトレースをファイルに出力すると，次のカレントのファイル名は「指定したファイル名 .01」になります。

(b) 記述できる場所

httpsd.conf

(33) IdentityCheck {On | Off} U

(a) 内容

クライアントホストの identd デーモンを使用してクライアントの確認をするかどうかを指定します。ident については，RFC1413 を参照してください。

ただし，クライアントホストが IPv6 アドレスの場合は，On を指定しても identd デーモンを使用してクライアントの確認をしません。また，ログフォーマットに %l を指定している場合，CGI 環境変数 REMOTE_IDENT には「unknown」を出力します。

On：identd デーモンを使用してクライアントの確認をします。

Off：identd デーモンを使用してクライアントの確認をしません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>

(34) ImapBase {map | referer | URL}

(a) 内容

イメージマップファイルの base 行のデフォルトを指定します。

map：マップファイルの場所

referer：ドキュメントの場所（イメージマップを表示した HTML ファイルの場所）

URL：指定した URL

URL には，IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(35) ImapDefault {error | nocontent | map | referer | URL}

(a) 内容

イメージマップファイルの default 行のデフォルトを指定します。

error：標準のエラーメッセージを表示します（ステータスコード 500 Server Error を応答します）。

nocontent：リクエストを無視します（ステータスコード 204 No Content を応答します）。

map：マップファイル中の URL をメニュー表示します。

referer：ステータスコード 302 Found を応答します。

URL：指定した URL のコンテンツを表示します。

URL には，IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(36) ImapMenu {none | formatted | semiformatted | unformatted}

(a) 内容

イメージマップファイルの指定値に map を与えた場合またはイメージマップ画像の (0,0) 座標をマウスでポイントした場合のメニュー表示を指定します。座標指定のない形でイメージマップファイルがリクエストされた場合の動作もこの設定に従います。

none：メニューは生成しません。このときの動作は，マップファイル中の default 行の指定に従います。

formatted：ヘッダおよびリンク一覧を表示します。マップファイル中のコメントは無視されます。

semiformatted：リンク一覧を表示します。マップファイル中のコメントも表示します。

unformatted：マップファイル中に HTML を記述することで，メニューの形式を自由に設定できます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(37) Include ファイル名

(a) 内容

ファイル名で指定したファイルをコンフィグファイルとして利用できるようにします。

ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが

指定できます。このディレクティブを複数指定する場合，マージされた内容が使用されます。ファイル内に同じディレクティブがある場合，後に指定した方で上書きされます。

(b) 記述できる場所

httpsd.conf

(38) IndexIgnore ファイル名 〔ファイル名 …〕

(a) 内容

ディレクトリインデクス表示時に，Web ブラウザに表示させないファイル名を指定します。正規表現でも指定できます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

(39) IndexOptions 〔{+ | -}〕オプション 〔〔{+ | -}〕オプション …〕

(a) 内容

ディレクトリインデクスの整形表示機能のオプション設定をします。オプションの前に + を指定するかまたは +- を省略するとそのオプションが有効になります。デフォルトではすべてのオプションが無効です。指定できるオプションの一覧を次に示します。

表 6-6 オプション一覧

オプション	意味
Charset= 文字セット ～《ISO-8859-1》 U ～《UTF-8》 W	インデクス表示するページの文字セットを指定します。HeaderName ディレクティブまたは ReadmeName ディレクティブで指定したファイルで使用している文字セットが，デフォルトの文字セット（UNIX 版：ISO-8859-1，Windows 版：UTF-8）と異なる場合は，このオプションで HeaderName ディレクティブまたは ReadmeName ディレクティブで指定したファイルと同じ文字セットを指定してください。このオプションでは，= 文字セットを省略できません。また，-Charset 指定時も +Charset 指定時と同様の動作をします。

オプション	意味
DescriptionWidth〔= {文字数 |*}〕《23，30，42 または 49》	ファイル説明文エリアの幅を文字数 (1 文字＝ 1 バイト) で指定します。* を指定した場合は AddDescription ディレクティブで指定したファイル説明文の最大長に合わせて表示します。このオプションを省略した場合のファイル説明文エリアの幅は，23 バイト (ただし，SuppressSize 指定時＋ 7，SuppressLastModified 指定時＋ 19) です。 -DescriptionWidth 指定時は = {文字数 |*} を省略できます。この場合の表示幅は 23 バイトです。
FancyIndexing	ディレクトリインデクスの整形表示機能を有効にします。
FoldersFirst	ファイルよりディレクトリを先にインデクス表示する場合に指定します。ただし，FancyIndexing が有効な場合だけです。
IconsAreLinks	ディレクトリインデクス整形表示時のアイコンをファイルに対するアンカーにします。
IconHeight〔= ピクセル数〕((>0))《22》	ディレクトリインデクス整形表示時のアイコンの高さをピクセル数で指定します。IconWidth オプションと一緒に指定します。インデクスを表示する HTML の IMG タグの HEIGHT 属性になります。
IconWidth〔= ピクセル数〕((>0))《20》	ディレクトリインデクス整形表示時のアイコンの幅をピクセルで指定します。IconHeight オプションと一緒に指定します。インデクスを表示する HTML の IMG タグの WIDTH 属性になります。
IgnoreCase	ディレクトリインデクス整形表示時に，ファイル名およびディレクトリ名の大文字と小文字の区別をしないで並べ替えます。
NameWidth〔= {文字数 | *}〕《23》	ファイル名およびディレクトリ名エリアの幅を文字数（1 文字＝ 1 バイト）で指定します。* を指定した場合はファイル名およびディレクトリ名の最大長に合わせて表示します。 = {文字数 | *} を省略する場合は必ず -NameWidth と指定してください。
ScanHTMLTitles	AddDescription ディレクティブの指定がない場合に，HTML ファイル中の <TITLE> タグを検索し，説明文として表示します。
SuppressColumnSorting	ファイル名，ディレクトリ名，最終更新日時，ファイルサイズおよびファイルの説明文の各カラムでインデクスを並べ替える機能を抑止します。
SuppressDescription	ファイルの説明文を表示しません。
SuppressHTMLPreamble	HeaderName ディレクティブが指定されている場合，HeaderName ディレクティブで指定されたファイルの内容と，自動生成される HTML ヘッダ部（<HTML> や <TITLE> など）が共に出力されます。このオプションは，HeaderName ディレクティブで指定されたファイルが HTML で記述されている場合，自動生成される HTML ヘッダ部の出力を抑制します。
SuppressLastModified	最終更新日時を表示させません。
SuppressSize	ファイルサイズを表示させません。

オプション	意味
TrackModified	ディレクトリ表示のためのレスポンスの HTTP レスポンスヘッダに，Last-Modified 値と Etag 値を設定します。このオプションを指定すると，クライアントは HEAD リクエストでディレクトリはファイル構成の変更を確認できるため，クライアントのキャッシュ機能を有効に活用できます。このオプションはオペレーティングシステムとファイルシステムが stat() をサポートしている場合だけ有効です。

(b) 注意事項

- このディレクティブを複数指定する場合，同じファイル名に異なる文字列は指定できません。
- IconHeight，IconWidth，NameWidth で = 値を指定する場合，- の指定はできません。
- 設定されたオプションは，httpsd.conf，<VirtualHost>，<Directory>，.htaccess の順で，また，上位ディレクトリから下位ディレクトリへ継承します。継承したオプションを最終的にマージして，インデクス整形表示形式を決定します。
- httpsd.conf で +- を付けてオプションを指定しても無効になります。ただし，httpsd.conf，<VirtualHost>，<Directory>，.htaccess の順で，また，下位ディレクトリに継承されます。継承されたオプション指定はマージ処理で有効になります。参照順位が下位の指定場所でオプションの指定がある場合または次に示すディレクティブのどれかの指定がある場合，マージ処理が実行されます。
  - AddAlt
  - AddAltByEncoding
  - AddAltByType
  - AddDescription
  - AddIcon
  - AddIconByEncoding
  - AddIconByType
  - DefaultIcon
  - HeaderName
  - ReadmeName

  (例)

  httpsd.conf ファイルに IndexOptions +FancyIndexing +IconsAreLinks を指定した場合，下位の指定場所でインデクス関係のディレクティブ指定がなければ FancyIndexing，IconsAreLinks は無効になります。

  httpsd.conf ファイルに IndexOptions +FancyIndexing +IconsAreLinks，かつ下位ディレクトリのアクセスコントロールファイルに，AddDescription " テキストファイル " *.txt を指定した場合，FancyIndexing，IconsAreLinks は有効になります。
- +- 指定のない Charset，IconHeight，IconWidth，NameWidth ディレクティブを指定すると，その指定場所内でそのオプションが指定されている位置より前に指定されている +- 付のオプション（Charset，IconHeight，IconWidth，NameWidth を除い

て）は無効になります。

（例）

IndexOptions FancyIndexing -IconsAreLinks IconHeight IconWidth

この場合，FancyIndexing，IconHeight，IconWidth ディレクティブが有効になります。IconsAreLinks の - 指定は継承されません。

- 指定場所間で同じディレクトリのインデクスを対象にオプション指定した場合のマージ処理は，参照順位がより後方の指定場所で +- のないオプションを指定すると，先に指定したオプションは無効になります。ただし，IconHeight，IconWidth，NameWidth は無効になりません。

  （例 1）

  - httpsd.conf ファイルの指定
    IndexOptions +FancyIndexing +IconsAreLinks
  - アクセスコントロールファイルの指定
    IndexOptions FancyIndexing SuppressLastModified

  これらを指定した場合，IconsAreLinks は無効になります。FancyIndexing，SuppressLastModified は有効になります。

  （例 2）

  - httpsd.conf ファイルの指定
    IndexOptions SuppressColumnSorting +FancyIndexing +IconsAreLinks
  - アクセスコントロールファイルの指定
    IndexOptions FancyIndexing SuppressLastModified

  これらを指定した場合，SuppressColumnSorting，IconsAreLinks は無効になります。また，FancyIndexing，SuppressLastModified は有効になります。

- 指定場所間で，同じディレクトリのインデクスを対象にオプション指定した場合のマージ処理は，同じオプションに対して + と - の両方を指定すると - 指定が有効になります。

  （例）

  - httpsd.conf ファイルの指定
    IndexOptions +FancyIndexing -IconsAreLinks
  - アクセスコントロールファイルの指定
    IndexOptions +IconsAreLinks

  これらを指定した場合，IconsAreLinks は無効になります。

- 同じ指定場所で +- を指定しないオプションを指定すると，Charset，IconHeight，IconWidth，NameWidth ディレクティブ以外の +- で指定したオプションは無効になります。

  （例 1）

  - httpsd.conf ファイルの指定
    IndexOptions +IconsAreLinks FancyIndexing +SuppressLastModified

  この場合，IconsAreLinks は無効になります。

(例 2)

- <VirtualHost> ブロック，<Directory> ブロックまたはアクセスコントロールファイルの指定
  IndexOptions +IconsAreLinks FancyIndexing +SuppressLastModified

この場合，IconsAreLinks，SuppressLastModified は無効になります。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(d) 上書き許可

Indexes レベル

(40) IndexOrderDefault {Ascending | Descending} {Name | Date | Size | Description}

(a) 内容

ディレクトリインデクス表示での，ファイルの並び順のデフォルトを指定します。

Ascending：昇順

Descending：降順

Name：ファイル名で並べます。

Date：ファイル更新日付で並べます。

Size：ファイルサイズで並べます。

Description：AddDescription ディレクティブで指定した説明文で並べます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

6.2.5 K，L で始まるディレクティブ

(1) KeepAlive {On | Off}

(a) 内容

KeepAlive 接続を有効にするかどうかを指定します。実際に KeepAlive が実行されるのはクライアント側も KeepAlive に対応している場合だけです。KeepAlive はサーバプロセスとクライアントとのコネクションが持続されるので，連続したリクエストのレスポ

ンスが良くなります。反面，サーバプロセスが特定のクライアント専用になるので，Web サーバ全体としてサービス能力が低下することもあります。KeepAliveTimeout，MaxKeepAliveRequests ディレクティブを使用して調整する必要があります。

On：持続型接続（KeepAlive）を有効にします。

Off：持続型接続（KeepAlive）を無効にします。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
KeepAlive On
```

(2) KeepAliveTimeout 時間

～ ((0 － 65535))《15》（単位：秒）

(a) 内容

KeepAlive 接続時の要求待ち時間を秒単位で指定します。この時間以上経過しても，クライアントから次のリクエストが来ない場合，コネクションを切断します。KeepAlive はサーバプロセスが特定のクライアントに占有されます。ある Web ページから次の Web ページへ移る場合に必要とする標準的な時間以上は，タイムアウトにしてコネクションを切断し，サーバプロセスをほかのリクエストの処理に当てるようにします。時間に 0 を指定した場合は，KeepAlive 接続が無効になります。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
KeepAliveTimeout 15
```

KeepAlive 接続時の要求待ち時間は 15 秒

(3) LanguagePriority 言語コード 〔言語コード …〕

(a) 内容

使用言語を優先順位の高い順に指定します。コンテントネゴシエーションで，Web ブラウザからのリクエストに言語コードの優先順位（Accept-Language ヘッダ）が含まれていない場合に，ここで指定した優先順位が使用されます。ここで指定する言語コードなどについては，AddLanguage ディレクティブを参照してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
LanguagePriority ja en fr de
```

優先順位は日本語，英語，フランス語，ドイツ語の順

(4) LDAPBaseDN DN 値 〔DN 値…〕

(a) 内容

LDAP サーバで認証する場合，検索を開始する最上位の DN を指定します。この DN では，Web サーバからのアクセスを許可する必要があります。

この DN の下に，検索するすべてのユーザエントリとグループエントリが必要です。また，Web サーバにこれらすべてのエントリ，属性へのアクセス権限が必要です。
LDAPServerName ディレクティブで複数の LDAP サーバが指定されている場合，
LDAP サーバごとに DN を指定してください。また，DN を複数指定する場合，DN ごとに "（引用符）で囲みます。DN の指定が一つの場合は引用符で囲む必要はありません。DN 中に引用符が含まれる場合は引用符の前に ¥ を付けます。

(b) 注意事項

LDAP サーバを利用したユーザ認証を使用するためには mod_hws_ldap モジュールの組み込みが必要です。LDAP サーバを利用したユーザ認証については，「4.5.4 ディレクトリサービスを利用したユーザ認証とアクセス制御」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>

(5) LDAPNoEntryStatus {Authorization | Forbidden}

(a) 内容

LDAP サーバを利用したユーザ認証に成功しても，LDAPRequire ディレクティブのアクセス制御によってアクセスを拒否する場合，Web サーバが Web ブラウザに返すステータスコードを指定します。

Authorization：ステータスコード 401 を返します。

Forbidden：ステータスコード 403 を返します。

(b) 注意事項

LDAP サーバを利用したユーザ認証を使用するためには mod_hws_ldap モジュールの組み込みが必要です。LDAP サーバを利用したユーザ認証については，「4.5.4 ディレクトリサービスを利用したユーザ認証とアクセス制御」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>

(6) LDAPRequire 〔%DN 属性 %〕〔LDAP 検索フィルタ〕

(a) 内容

AuthName ディレクティブ，AuthType ディレクティブおよび Require valid-user ディレクティブと一緒に指定してアクセス制御するユーザの範囲を指定します。

先頭に%で囲んだ文字列がある場合，この文字列をクライアントが入力したユーザ名を識別する DN 属性として利用します。%がない場合には DN 属性として cn（エントリによって定義される人を識別する必須属性）を仮定します。

LDAPBaseDN ディレクティブの指定値に，クライアントが入力したユーザ名を DN 属性として設定された値と組み合わせることで，ユーザが登録されている DN を求めます。この DN とクライアントが入力したパスワードを使って LDAP サーバで認証します。

指定したユーザが LDAP サーバで認証され，かつ LDAP 検索フィルタに当てはまる場合にコンテンツをアクセスできます。フィルタが指定されていない場合には，検索フィルタとして (objectClass=*) が設定されます。

SSL クライアント認証と併用した場合，LDAP サーバへのアクセスは，クライアント証明書内の Subject フィールドの，CN の値をユーザ名として，パスワードなしの匿名アクセスになります。LDAP サーバを検索した結果，ユーザ名が LDAP サーバに登録されており，かつ LDAP 検索フィルタに当てはまる場合にコンテンツをアクセスできます。フィルタが指定されていない場合には，検索フィルタとして (objectClass=*) が設定されます。

このディレクティブを指定した場合，そのディレクトリ内に指定されている Require ディレクティブ指定値は無効になりますが，Require ディレクティブの指定は必須です。

検索フィルタは次の形式で定義できます。

（属性　演算子　値）

演算子として次の演算子が使用できます。

表 6-7　検索フィルタで使用できる演算子

検索種類	シンボル	説明
Equality	=	指定値に設定された属性エントリを含むエントリを返します。 **例：**cn=hitachi taro
Substring	=< 文字列 >*< 文字列 >	指定の部分文字列を含む属性を持ったエントリを返します。 **例：**cn=hita*，cn=*hanako，cn=*hi*，cn=h*hanako

検索種類	シンボル	説明
Greater than or equal to	>=	指定値以上の属性を含むエントリを返します。 **例：**employeenumber>=100
Less than or equal to	<=	指定値以下の属性を含むエントリを返します。 **例：**employeenumber<=100
Presence	=*	指定の属性を含むエントリを返します。 **例：**cn=*，telephonenumber=*，manager=*

さらに，これらの検索フィルタを複数組み合わせたフィルタを作成できます。

(演算子 (検索フィルタ)(検索フィルタ)…)

この場合，次の演算子が使用できます。

認証するユーザが一つの属性に対して複数の属性エントリを持っている場合，一つの属性エントリが演算に一致する場合にアクセスを許可します。

表 6-8　複数の検索フィルタ間で使用できる演算子

演算子	シンボル	説明
And	&	すべてのフィルタが真のエントリを返します。 **例：**(&(filter)(filter)(filter)…)
Or	|	最低一つの指定フィルタが真のエントリを返します。 **例：**(|(filter)(filter)(filter)…)
Not	!	指定したフィルタが真でないエントリを返します。 **例：**(!(filter))

注　Not 演算子の場合，フィルタを複数指定できません。

認証に失敗した場合，ステータスコード 401 Authorization Required を応答します。フィルタ条件に一致しない場合，LDAPNoEntryStatus ディレクティブに従ったステータスコード (デフォルトでは 401 Authorization Required) を応答します。また，フィルタ形式の文法に誤りがある場合，ステータスコード 500 Internal Server Error を応答します。

同一ユーザを複数指定する場合，その中の一つでアクセス権限を与えれば，アクセスできます。

(b) 注意事項

LDAP サーバを利用したユーザ認証を使用するためには mod_hws_ldap モジュールの組み込みが必要です。LDAP サーバを利用したユーザ認証については，「4.5.4　ディレクトリサービスを利用したユーザ認証とアクセス制御」を参照してください。

(c) 記述できる場所

<Directory>，.htaccess

(d) 上書き許可

AuthConfig レベル

(e) 指定例

- ユーザ名が hitachi taro と hitachi hanako にアクセス権限を与える場合
  ```
  LDAPRequire (|(cn=hitachi taro)(cn=hitachi hanako))
  ```
- ユーザ ID が 99001 から 99029 までの人と，99051 から 99059 までの人にアクセス権限を与える場合
  ```
  LDAPRequire
  (|(&(uid>=99001)(uid<=99029))(&(uid>=99051)(uid<=99059)))
  ```

(7) LDAPServerName {ホスト名 | IP アドレス} [{ホスト名 | IP アドレス} …]

～《127.0.0.1》

(a) 内容

LDAP サーバのホスト名または IP アドレスを指定します。複数の LDAP サーバを指定する場合，それぞれの LDAP サーバに対応するポート番号（LDAPServerPort ディレクティブ）と検索を開始する最上位の DN（LDAPBaseDN ディレクティブ）を指定してください。指定した LDAP サーバ，ポート番号および DN の数が一致しないと，LDAP サーバでの認証をしません。エラーログを出力し，Web ブラウザにステータスコード 500 を返します。

ただし，このディレクティブで指定するすべての LDAP サーバで同じポート番号を使用する場合，ポート番号の指定は簡略化できます。

また，複数の LDAP サーバを指定する場合，優先順位の高い順に指定します。それぞれのリクエストに対しては必ず最初に指定された LDAP サーバからユーザ認証をします。

デフォルトはローカルホスト（127.0.0.1）です。

なお，ホスト名に IPv6 アドレスに対応したホスト名は指定できません。また，IP アドレスに IPv6 アドレスは指定できません。

(b) 注意事項

LDAP サーバを利用したユーザ認証を使用するためには mod_hws_ldap モジュールの組み込みが必要です。LDAP サーバを利用したユーザ認証については，「4.5.4　ディレクトリサービスを利用したユーザ認証とアクセス制御」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>

(d) 指定例

```
LDAPServerName server01 server02 server03
LDAPServerPort 389
LDAPBaseDN "ou=employee, o=hitachi, c=jp" "ou=employee,
o=example.com" "o=hitachi, c=jp"
```

このように指定すると，LDAP サーバ，ポート番号および DN の組み合わせは次のようになります。

利用する LDAP サーバの順	ホスト名	ポート番号	検索を開始する DN
1	server01	389	ou=employee，o=hitachi，c=jp
2	server02	389	ou=employee，o=example.com
3	server03	389	o=hitachi，c=jp

(8) LDAPServerPort　ポート番号 〔ポート番号…〕

～ ((1 － 65535))《389》

(a) 内容

LDAP サーバのポート番号を指定します。LDAPServerName ディレクティブで複数の LDAP サーバが指定されている場合，LDAP サーバごとにポート番号を指定してください。ただし，LDAPServerName ディレクティブで指定したすべての LDAP サーバで同じポート番号を使用するときは複数指定する必要はありません。ポート番号を一つ指定するだけで，すべての LDAP サーバに適用されます。複数の LDAP サーバを指定する場合の指定例は LDAPServerName ディレクティブを参照してください。LDAP サーバの数とポート番号の数が一致しないと，エラーになります。

(b) 注意事項

LDAP サーバを利用したユーザ認証を使用するためには mod_hws_ldap モジュールの組み込みが必要です。LDAP サーバを利用したユーザ認証については，「4.5.4　ディレクトリサービスを利用したユーザ認証とアクセス制御」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>

(9) LDAPSetEnv　環境変数　属性

(a) 内容

LDAP サーバで認証する場合，認証されたユーザの DN によって識別されるエントリを構成する属性の値を環境変数の値として設定します。属性の値は，LDAP サーバから得られる文字コードで設定します。返送される文字コードについては，使用する LDAP サーバのマニュアルを参照してください。一つの属性に対して複数の値がある場合には，環境変数は設定されません。バイナリオプション (;binary) のある属性は指定できませ

ん。そのほかのオプションを持つ属性では，属性の値が文字列でないかぎり正常に設定されません。同じ環境変数を別の属性で定義した場合はエラーとなりませんが，環境変数の値は不定になります。

(b) 注意事項

LDAP サーバを利用したユーザ認証を使用するためには mod_hws_ldap モジュールの組み込みが必要です。LDAP サーバを利用したユーザ認証については，「4.5.4　ディレクトリサービスを利用したユーザ認証とアクセス制御」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(d) 上書き許可

FileInfo レベル

(10) LDAPTimeout　値

～ ((1 － 86400))《30》（単位：秒）

(a) 内容

ユーザ認証後，一つの LDAPRequire ディレクティブで指定したフィルタの検索処理の最大待ち時間を秒単位で指定します。次に示すどれかの場合に，ステータスコード 500 Internal Server Error になります。

- このディレクティブで指定した間応答がない場合
- LDAP サーバ自身がタイムアウトを通知した場合
- LDAP サーバがアクセスに失敗した場合

(b) 注意事項

LDAP サーバを利用したユーザ認証を使用するためには mod_hws_ldap モジュールの組み込みが必要です。LDAP サーバを利用したユーザ認証については，「4.5.4　ディレクトリサービスを利用したユーザ認証とアクセス制御」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>

(11) LDAPUnsetEnv　環境変数

(a) 内容

LDAPSetEnv ディレクティブで指定した環境変数を無効化します。

(b) 注意事項

LDAP サーバを利用したユーザ認証を使用するためには mod_hws_ldap モジュールの組

み込みが必要です。LDAP サーバを利用したユーザ認証については，「4.5.4　ディレクトリサービスを利用したユーザ認証とアクセス制御」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(d) 上書き許可

FileInfo レベル

(12) LimitRequestBody　リクエストボディサイズ

～ ((0 － 2147483647))《0》（単位：バイト）

(a) 内容

HTTP 通信によって，Web ブラウザが送信してくるリクエストをサーバが受信する場合のオブジェクトボディ（デ ータ）のサイズの上限を指定します。Web ブラウザから <FORM METHOD=POST ACTION=...> によるリクエストを送る場合などにオブジェクトボディが用いられます。上限値を設定しない場合は，0 を指定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(13) LimitRequestFields　ヘッダ数

～ ((0 － 32767))《100》

(a) 内容

HTTP 通信によって，Web ブラウザが送信してくるリクエストをサーバが受信する場合の HTTP ヘッダ数の上限を指定します。リクエストの HTTP ヘッダ数は，Web ブラウザやリクエストを中継するプロキシなどの仕様で変わります。上限値を設定しない場合は，0 を指定してください。

(b) 記述できる場所

httpsd.conf

(14) LimitRequestFieldsize　ヘッダサイズ

～ ((0 － 8190))《8190》（単位：バイト）

(a) 内容

HTTP 通信によって，Web ブラウザが送信してくるリクエストをサーバが受信する場合，一つの HTTP ヘッダの，サイズの上限を指定します。リクエストヘッダのサイズは Web ブラウザやリクエストを中継するプロキシなどの仕様で変わります。

(b) 記述できる場所

httpsd.conf

(15) LimitRequestLine　リクエストライン長

～ ((0 － 8190))《8190》（単位：バイト）

(a) 内容

HTTP 通信によって，Web ブラウザが送信してくるリクエストをサーバが受信する場合のリクエストライン（メソッド，問い合わせ文字列などを含む URI，HTTP バージョン）の長さの上限を指定します。Web ブラウザから <FORM METHOD=GET ACTION...> によるリクエストを送る場合などに問い合わせ文字列としてリクエストラインが用いられます。なお，リクエストラインとして Web ブラウザから何バイト送れるかは，Web ブラウザやリクエストを中継するプロキシなどの仕様で変わります。

(b) 記述できる場所

httpsd.conf

(16) Listen 〔IP アドレス :〕ポート番号

(a) 内容

リクエストを受け付ける IP アドレスおよびポート番号を指定します。Port ディレクティブと異なり，複数指定できます。Listen ディレクティブを指定すると，Port ディレクティブおよび BindAddress ディレクティブの指定は無視されます。

IP アドレスには IPv6 アドレスも指定できます。IPv6 アドレスを指定する場合は，IPv6 アドレスを [　] で囲んでください。ただし，IP アドレスを省略してポート番号だけを指定した場合は，IPv4 アドレスを使用したリクエストだけを受け付けます。このため，IPv6 アドレスを使用する場合は，必ず Listen ディレクティブに IPv6 アドレスを指定してください。

Listen ディレクティブの IP アドレスを変更してサーバを再起動する場合，サーバをいったん停止後，起動してください。コマンドなどで再起動を選択すると，サーバの起動に失敗する場合があります。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
Listen 80
Listen [fec0::123:4567:89ab:cdef]:8080
Listen [::]:80
```

(17) ListenBacklog　バックログ数

～ ((1 － 2147483647))《511》

(a) 内容

クライアントからの接続要求の最大の待ち行列数を指定します。この指定値はシステムコール listen( ) のバックログ数として設定されます。ただし，指定値の制限値や，実際の待ち行列数の最大値については OS によって異なるため，詳細は各 OS の listen( ) についてのマニュアルや，各 OS の TCP/IP 実装の詳細を説明しているドキュメントを参照してください。

(b) 記述できる場所

httpsd.conf

(18) LoadFile　ファイル名 〔ファイル名 …〕

(a) 内容

DSO によって組み込むモジュールが参照するコードがあるオブジェクトファイルまたはライブラリを指定します。ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

LoadModule ディレクティブでこのファイルを参照するモジュールを指定する場合，それらが httpsd.conf で使用される前に，このディレクティブを指定する必要があります。

(b) 記述できる場所

httpsd.conf

(19) LoadModule　module 構造体名　ライブラリファイル名

(a) 内容

Web サーバに動的に組み込むモジュールを指定します。ライブラリファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
LoadModule hws01_module libexec/mod_hws01.so
LoadModule hws02_module libexec/mod_hws02.so
```

モジュール hws01_module とモジュール hws02_module を組み込みます。

(20) LogFormat　" フォーマット " 〔ラベル名〕

～ <<"%h %l %u %t ¥"%r¥" %>s %b">>

(a) 内容

ログのフォーマットにラベル名を定義します。ここで定義したラベル名を CustomLog ディレクティブで指定できます。指定できるフォーマットは CustomLog ディレクティブを参照してください。なお，フォーマットに %A または %a を指定した場合，IPv6 アドレスも出力できます。また，フォーマットに %h または %V を指定した場合，IPv6 アドレスに対応したホスト名または IPv6 アドレスも出力できます。

ラベル名を付けない場合は，このディレクティブを複数指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
LogFormat "%h %l %u %t ¥"%r¥" %>s %b ¥"%{Referer}i¥"
¥"%{User-Agent}i¥"" combined
LogFormat "%h %l %u %t ¥"%r¥" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

(21) LogLevel {debug | info | notice | warn | error | crit | alert | emerg}

(a) 内容

エラーログに出力するエラーのレベルを指定します。指定したレベルの上位レベルのログを出力します。ただし，notice レベルのログはこの指定に関係なく出力されます。また，Hitachi Web Server 起動時など，レベル指定の解析終了前に出力されるメッセージは，この指定に関係なく出力される場合があります。

次にエラーレベルを上位順に示します。

レベル	意味
emerg	緊急メッセージ
alert	即時処理要求メッセージ
crit	致命的な状態のメッセージ
error	一般的エラーメッセージ
warn	警告レベルメッセージ
notice	標準的だが重要なメッセージ
info	インフォメーションメッセージ，外部モジュールと CGI プログラム実行時のモジュールトレース※
debug	デバッグレベルメッセージ，内部モジュールトレースおよび info 相当のモジュールトレース※

注※　モジュールトレースは，エラーログではなくリクエストログに出力するよう設定できます。

詳細は，「4.2.2(5) 各トレースの出力先」および「4.2.6 モジュールトレースの採取」を参照してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
LogLevel info
```

6.2.6 M，N，O，P，Q，R で始まるディレクティブ

(1) MaxClients 接続数 U

～((1 － 1024))《1024》

(a) 内容

同時に接続できるクライアントの最大数を指定します。

サーバを起動すると，StartServer ディレクティブで指定した数のプロセスが起動されリクエストを待ちます。多くのリクエストが同時に発生した場合，複数のプロセスでリクエストを処理することになります。リクエスト待ちの残りプロセス数が MinSpareServers ディレクティブで指定した数より少なくなると，徐々に新規プロセスを生成します。このとき，プロセス数がこのディレクティブで指定した数になるまでプロセスが生成されます。その後，リクエストの処理が終了しリクエスト待ちプロセスが増加すると，MaxSpareServers ディレクティブで指定した数までプロセスを終了させます。

プロセス数に関連するほかのディレクティブについては，「4.1 Hitachi Web Server の処理とディレクティブとの関係」を参照してください。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
MaxClients 150
```

(2) MaxKeepAliveRequests 接続数

～((0 － 2147483647))《100》

(a) 内容

KeepAlive 連続接続回数の上限を指定します。上限値を設定しない場合は 0 を指定します。KeepAlive はサーバプロセスが特定のクライアントに占有されるので，ほかのクライアントにもサービスの機会を与えるために上限を設けます。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
MaxKeepAliveRequests 100
```

(3) MaxRequestsPerChild　リクエスト処理回数

～((0 － 2147483647))《0》

(a) 内容

サーバプロセスのリクエスト処理回数を指定します。サーバプロセスは指定されたリクエスト処理回数だけ動作し，終了します。ユーザが作成したアプリケーションなどによるメモリリークによる障害を未然に防ぐ効果があります。なお，0 を指定すると，サーバプロセスのリクエスト処理回数の上限は設定されません。サーバプロセスは終了することなく，リクエストを待ち，処理します。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
MaxRequestsPerChild 10000
```

(4) MaxSpareServers　プロセス数

～((1 － 1024))《10》

(a) 内容

リクエスト待ち状態で稼働させておくサーバプロセスの最大数を指定します。プロセス数に関連するほかのディレクティブについては，「4.1　Hitachi Web Server の処理とディレクティブとの関係」を参照してください。

MinSpareServers 以下の値を設定した場合，MinSpareServers 指定値 +1 の値が仮定されます。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
MaxSpareServers 10
```

(5) MinSpareServers　プロセス数

～((1 － 1024))《5》

(a) 内容

リクエスト待ち状態で稼働しているサーバプロセスの最小数を指定します。サーバプロセス数がこの指定値より少なくなったら，新しいプロセスを生成します。プロセス数に関連するほかのディレクティブについては，「4.1　Hitachi Web Server の処理とディレクティブとの関係」を参照してください。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
MinSpareServers 5
```

(6) MultiviewsMatch　{NegotiatedOnly ｜ Handlers}

(a) 内容

コンテントネゴシエーションの対象となる拡張子の種類を指定します。

NegotiatedOnly：拡張子が文字セット，圧縮形式，言語コード，MIME タイプと関連づけられたものだけをコンテントネゴシエーションの対象にします。

Handlers：NegotiatedOnly を指定した場合の対象に加え，ハンドラと関連づけられた拡張子についてもコンテントネゴシエーションの対象にします。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
MultiviewsMatch Handlers
```

(7) NameVirtualHost　{IP **アドレス**｜ *}〔: **ポート番号**〕

(a) 内容

サーバ名に基づくバーチャルホストで使用する IP アドレスを指定します。IP アドレスには，IPv6 アドレスも指定できます。IPv6 アドレスを指定する場合は，IPv6 アドレスを [　] で囲んでください。

IP アドレスの代わりに * を指定すると，IPv4 アドレスを指定した NameVirtualHost ディレクティブや <VirtualHost> ブロックで使用していない IPv4 アドレスによるコネクションに対応して，サーバ名に基づくバーチャルホストを生成します。この指定はサーバ名に基づくバーチャルホストだけを使用している場合で，コンフィグファイルに IP アドレスを固定したくないときに便利です。

(b) 記述できる場所

httpsd.conf

(8) Options {+ | -} オプション 〔{+ | -} オプション …〕

～《All》

(a) 内容

ユーザが利用できる機能を制限する場合に指定します。

+：オプションで指定した機能の利用を許可します。

-：オプションで指定した機能の利用を禁止します。

オプション	機能
All	MultiViews，SymLinksIfOwnerMatch を除くすべてのオプションが有効です。
ExecCGI	CGI スクリプトの実行を許可します。
FollowSymLinks	シンボリックリンクをたどります。Windows 版では指定できません。
Indexes	URL にディレクトリが指定されたとき，DirectoryIndex ディレクティブで指定したファイル（デフォルトは index.html）がない場合，ディレクトリのインデクスを表示します。
MultiViews	Content-negotiated Multiviews をサポートします。
None	すべてのオプションで指定できる機能を無効にします。
SymLinksIfOwnerMatch	ファイルまたはディレクトリの所有者がシンボリックリンクの所有者と同じ場合だけ，リンクをたどります。Windows 版では指定できません。

注 +- を使用しないでこのディレクティブを複数指定すると，最後に指定したディレクティブだけが有効になります。

(例 1)

```
Options All
Options ExecCGI
```

このようにオプションに +- を指定しないディレクティブを 2 行指定した場合，ユーザは CGI スクリプトの実行機能だけが利用できます。ディレクトリインデクスなどの機能は利用できません。

(例 2)

httpsd.conf ファイルの指定

```
Options All
```

アクセスコントロールファイルの指定

```
Options ExecCGI
```

httpsd.conf ファイルの後にアクセスコントロールファイルが参照されるので，アクセスコントロールファイルがあるディレクトリでは CGI スクリプトの実行機能だけが利用できます。

（例 3）

```
Options Indexes ExecCGI
```

このように 1 行に +- を指定しないオプションを指定した場合は，指定した機能の両方を利用できます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Options レベル

(9) Order　指示子

～《deny,allow》

(a) 内容

Allow ディレクティブと Deny ディレクティブの指定の評価の順序を指定します。指示子に指定できるものを次に示します。先に評価されたものは，後に評価されるものに上書きされます。

指示子	意味
deny,allow	Deny ディレクティブの指定を，Allow ディレクティブの指定より先に評価
allow,deny	Allow ディレクティブの指定を，Deny ディレクティブの指定より先に評価
mutual-failure	Allow ディレクティブに指定され，Deny ディレクティブに指定されていないホストだけアクセスを許可

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

Limit レベル

(10) PassEnv　環境変数 〔環境変数 …〕

(a) 内容

CGI スクリプトに渡す任意の環境変数を指定できます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
PassEnv TMP
```

(11) PidFile　ファイル名

～《logs/httpd.pid》

(a) 内容

制御プロセス ID を格納するファイル名を指定します。ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

(b) 注意事項

Windows 版の場合，再起動時には，PidFile ディレクティブ指定値の変更は反映されません。PidFile ディレクティブ指定値を変更した場合は，いったん Web サーバを停止してから，再起動してください。

UNIX 版の場合，再起動時には，PidFile ディレクティブ指定値の変更は反映されません。PidFile ディレクティブ指定値を変更した場合は，いったん Web サーバを kill コマンドで停止してから，起動してください。停止時に httpsdctl ユティリティは使用できません。

(c) 記述できる場所

httpsd.conf

(d) 指定例

```
PidFile logs/httpd.pid
```

(12) Port　ポート番号

～ ((1 － 65535))《80》

(a) 内容

IPv4 アドレスを使用した Web ブラウザからの要求を受け付けるサーバのポート番号を指定します。

Port ディレクティブを指定しても，IPv6 アドレスを使用した Web ブラウザからの要求は受け付けません。IPv6 アドレスを使用する場合は，Listen ディレクティブで指定してください。その場合，IPv4 アドレスと併用するときは，IPv4 アドレスについても Listen ディレクティブを指定してください。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
Port 80
```

(13) ProxyErrorOverride {On | Off}

(a) 内容

バックエンドサーバからのステータスコードが 300 番台，400 番台または 500 番台の場合，レスポンスヘッダとレスポンスボディをオーバーライドします。その結果，リバースプロキシはバックエンドサーバからのレスポンスではなく，自身が生成したレスポンスをクライアントに返します。

On：バックエンドサーバからのステータスコードが 300 番台，400 番台または 500 番台の場合，レスポンスヘッダとレスポンスボディをオーバーライドします。

Off：レスポンスヘッダとレスポンスボディをオーバーライドしません。

(b) 注意事項

リバースプロキシを使用するためには mod_proxy モジュールおよび mod_proxy_http モジュールの組み込みが必要です。リバースプロキシの詳細は，「4.7　リバースプロキシの設定」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>

(d) 指定例

```
ProxyErrorOverride On
```

バックエンドサーバからのステータスコードが 300 番台，400 番台または 500 番台の場合，リバースプロキシが生成したレスポンスをクライアントに返します。

(14) ProxyPass **パス名** URL

(a) 内容

リバースプロキシを使用する場合，Web ブラウザからのリクエストとそれを転送するアドレスを指定します。

パス名：Web ブラウザからリバースプロキシへのリクエストを /（スラッシュ）から始まる URL で指定します。

URL：転送先となるバックエンドサーバの URL を "http:// ホスト名〔: ポート番号〕" を含む形で指定します。

URL には，IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

(b) 注意事項

リバースプロキシを使用するためには mod_proxy モジュールおよび mod_proxy_http モジュールの組み込みが必要です。リバースプロキシの詳細は，「4.7　リバースプロキシの設定」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>

(15) ProxyPassReverse　パス名　URL

(a) 内容

リバースプロキシを使用する場合，バックエンドサーバからのリダイレクトレスポンスの Location ヘッダで示す URL を変更します。Web ブラウザからのリダイレクトによるリクエストをリバースプロキシを通すリクエストにするために Location ヘッダをこのディレクティブの指定値に変更します。

パス名：リダイレクトのリクエスト先であるリバースプロキシのパス名を，/（スラッシュ）から始まる URL で指定します。

URL：変更対象となる Location ヘッダ中のバックエンドサーバの URL を "http:// ホスト名〔: ポート番号〕" を含む形で指定します。

URL には，IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。IPv6 アドレスにはさまざまな表記方法がありますので，指定値に注意してください。IPv6 アドレスの表記が指定値と一致しない場合，ディレクティブが有効になりません。IPv6 アドレスを指定する場合は，バックエンドサーバからの応答の Location ヘッダ値に含まれる IPv6 アドレスの表記を確認してください。

(b) 注意事項

リバースプロキシを使用するためには mod_proxy モジュールおよび mod_proxy_http モジュールの組み込みが必要です。リバースプロキシの詳細は，「4.7　リバースプロキシの設定」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>

(16) ProxyPreserveHost　{On ｜ Off}

(a) 内容

リバースプロキシを使用する場合，クライアントから受信した Host ヘッダの値をそのままバックエンドサーバに転送するかどうかを指定します。

On：クライアントから受信した Host ヘッダの値をそのままバックエンドサーバに転送

します。

Off：クライアントから受信した Host ヘッダの値を ProxyPass ディレクティブの指定値に従って変更して，バックエンドサーバに転送します。

(b) 注意事項

リバースプロキシを使用するためには mod_proxy モジュールおよび mod_proxy_http モジュールの組み込みが必要です。リバースプロキシの詳細は，「4.7　リバースプロキシの設定」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>

(d) 指定例

```
ProxyPreserveHost On
```

クライアントから受信した Host ヘッダの値をそのままバックエンドサーバに転送します。

(17) ProxyVia　{on ｜ off ｜ full ｜ block}

(a) 内容

このディレクティブはプロキシで Via ヘッダの使用を制御する場合に指定します。

on：Via ヘッダに自ホストの情報を追加します。すでにある情報は変更しません。

off：Via ヘッダに自ホストの情報を追加しません。すでにある情報は変更しません。

full：コメントとして自ホストのバージョンを付けた情報を Via ヘッダに追加します。すでにある情報は変更しません。

block：Via ヘッダに自ホストの情報を追加しません。リクエスト中の Via ヘッダは削除します。

(b) 注意事項

リバースプロキシを使用するためには mod_proxy モジュールおよび mod_proxy_http モジュールの組み込みが必要です。リバースプロキシの詳細は，「4.7　リバースプロキシの設定」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>

(18) QOSCookieDomain　ドメイン名

(a) 内容

流量制限機能に使用するクッキーが有効とされるドメインを指定します。この値は，

HWS 作成モードで使用され，ユーザ作成モードでは使用されません。複数のホストを設定している場合，このディレクティブを設定することでドメイン部分の共通するホスト間でクッキーを使用できるようになります。ドメイン名には，少なくとも "." が二つ含まれていなければなりません。

なお，IPv6 アドレスに対応したドメイン名も指定できます。

（例）

a.example.com と b.example.com の二つのホストを設定している場合，このディレクティブで .example.com と指定すると，二つのホストのどちらにアクセスしても優先度処理が行われます。

(b) 注意事項

流量制限機能を使用するためには mod_hws_qos モジュールの組み込みが必要です。流量制限機能については，「4.9　流量制限機能」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>

(19) QOSCookieExpires　値

～ ((0 － 86400))《300》（単位：秒）

(a) 内容

流量制限機能に使用するクッキーの有効時間を秒単位で指定します。この値は，HWS 作成モードで使用され，ユーザ作成モードでは使用されません。

(b) 注意事項

流量制限機能を使用するためには mod_hws_qos モジュールの組み込みが必要です。流量制限機能については，「4.9　流量制限機能」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Location>

(20) QOSCookieName　クッキー名　〔{hws ｜ user}〕

～《HWSCHK》

(a) 内容

流量制限機能に使用するクッキー名を指定します。クッキー名にセミコロン，コンマ，空白文字は使用できません。ホスト間および URL 間でそれぞれ異なるクッキーを利用したセション管理を行う場合は，別のクッキー名を指定する必要があります。

hws：Hitachi Web Server が作成するクッキーを用いて，セション管理を実施します。

これを HWS 作成モードと呼びます。

user：Hitachi Web Server 以外の外部モジュールなどで作成されたクッキーを用いて，セション管理を実施します。これをユーザ作成モードと呼びます。

(b) 注意事項

- 流量制限機能を使用するためには mod_hws_qos モジュールの組み込みが必要です。流量制限機能については，「4.9　流量制限機能」を参照してください。
- QOSCookieName ディレクティブを特定のブロックに指定した場合，上位に指定されている QOSCookieName ディレクティブは継承しません。

(例)

```
QOSCookieName Cookie1 hws
<Location /loc1>
   QOSCookieName Cookie2 user
</Location>
```

この場合，"/loc1" から始まるリクエストでは，クッキー名 Cookie2 の指定が有効になります。"/loc1" 以外から始まるリクエストでは，クッキー名 Cookie1 の指定が有効になります。

- QOSCookieName ディレクティブを複数指定する場合は，クッキー名を重複させないでください。重複している場合は，起動エラーになります。

(例)

```
QOSCookieName Cookie1 hws
QOSCookieName Cookie1 user
```

この場合，クッキー名が重複しているため起動エラーになります。

- HWS 作成モードの QOSCookieName ディレクティブを複数指定した場合は，後に指定した方が有効になります。

(例)

```
QOSCookieName Cookie1 hws
QOSCookieName Cookie2 hws
```

この場合，クッキー名 Cookie1 の指定は無効になり，Cookie2 の指定が有効になります。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Location>

(21) QOSCookieSecure {on | off}

(a) 内容

クライアントに対し，SSL によるアクセス時だけにクッキーを送信させるよう設定します。この値は，HWS 作成モードで使用され，ユーザ作成モードでは使用されません。クッキーの確認は SSL の暗号処理の終了後であることに注意してください。

on：SSL によるアクセス時だけ，クライアントにクッキーを送信させるよう設定します。

off：SSL 以外によるアクセス時にも，クライアントにクッキーを送信させるよう設定します。

（例）

SSL が有効であるホストと無効であるホストを設定している場合，このディレクティブを設定すると，SSL が有効なホストへのアクセスだけクッキーが送信されます。

（b） 注意事項

流量制限機能を使用するためには mod_hws_qos モジュールの組み込みが必要です。流量制限機能については，「4.9　流量制限機能」を参照してください。

（c） 記述できる場所

httpsd.conf，<VirtualHost>，<Location>

（22） QOSCookieServers　値

UNIX 版の場合

～ ((0 － MaxClients ディレクティブ指定値))《10》

Windows 版の場合

～ ((0 － ThreadsPerChild ディレクティブ指定値))《10》

（a） 内容

リクエスト待ち状態のサーバプロセス数が減少した場合に，クッキーを送信してきたリクエストだけを処理するときの，サーバプロセス数を指定します。

Windows 版の場合は，サーバスレッド数を指定します。

（b） 注意事項

流量制限機能を使用するためには mod_hws_qos モジュールの組み込みが必要です。流量制限機能については，「4.9　流量制限機能」を参照してください。

（c） 記述できる場所

httpsd.conf，<VirtualHost>，<Location>

（23） QOSRedirect　旧パス　新パス

（a） 内容

流量制限機能によって処理が拒否された場合に，クライアントからのリクエストを指定されたパスにリダイレクトさせるときに指定します。新パスには，" プロトコル名 :// ホスト名〔: ポート番号〕" を含む URL のパスを指定します。また，新パスに指定する

URL には，IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

旧パスでリクエストを受けた場合，ステータスコード 302 と Location ヘッダに新パスを設定したレスポンスを返します。レスポンスをカスタマイズすることはできません。

旧パス，新パスの指定については，Redirect ディレクティブを参照してください。

(b) 注意事項

流量制限機能を使用するためには mod_hws_qos モジュールの組み込みが必要です。流量制限機能については，「4.9　流量制限機能」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Location>

(24) QOSRejectionServers　値

UNIX 版の場合

～ ((0 － MaxClients ディレクティブ指定値))《1》

Windows 版の場合

～ ((0 － ThreadsPerChild ディレクティブ指定値))《1》

(a) 内容

リクエスト待ち状態のサーバプロセス数が減少し，受信したすべてのリクエストを拒否するようになるときの，サーバプロセス数を指定します。

Windows 版の場合は，サーバスレッド数を指定します。

(b) 注意事項

流量制限機能を使用するためには mod_hws_qos モジュールの組み込みが必要です。流量制限機能については，「4.9　流量制限機能」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Location>

(25) QOSResponse　{file［MIME タイプ］ファイル名｜message　テキスト}

(a) 内容

流量制限機能によって処理が拒否された場合に，503 ステータスコードとともに返送するコンテンツを指定します。コンテンツはサーバプロセス内にキャッシュされるため，変更する場合にはサーバの再起動が必要です。

file：指定したファイルを，指定した MIME タイプで返送します。MIME タイプを省略したときは "text/html" が設定されます。また，ファイル名には，絶対パスまたは

ServerRoot ディレクティブの指定値からの相対パスが指定できます。

message：指定したテキストを返送します。テキストは先頭に " を記述して文字列を指定します。MIME タイプには "text/html" が設定されます。

(b) 注意事項

流量制限機能を使用するためには mod_hws_qos モジュールの組み込みが必要です。流量制限機能については，「4.9　流量制限機能」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Location>

(d) 指定例

```
QOSResponse file "text/html; charset=ISO-8859-1" htdocs/busy.html
QOSResponse message "Server busy.
```

(26) ReadmeName　ファイル名

(a) 内容

ディレクトリインデクス表示時の Readme として付けるコメントを記述したファイルのファイル名（パス情報なし）を指定します。HTML またはプレーンテキストで記述できます。ただし，AddType ディレクティブまたは TypesConfig ディレクティブで指定したファイルで，MIME タイプが正しく定義されている必要があります。プレーンテキストでコメントを作成した場合，ディレクトリインデクス表示の HTML には <PRE> タグが追加されます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

Indexes レベル

(d) 指定例

```
ReadmeName README.html
```

ディレクトリ下の README.html ファイルの内容を表示します。

(27) Redirect 〔{permanent ｜ temp ｜ seeother ｜ gone ｜ステータスコード}〕 旧パス　新パス

(a) 内容

旧パスに対するクライアントからのリクエストを，新パスに再リクエスト（リダイレクト）する場合に指定します。

旧パスには，スラッシュから始まるリクエスト URL のパスを指定します。ただし，旧パ

スには，? 以降（問い合わせ文字列）を指定できません。

新パスには，" プロトコル名 :// ホスト名〔: ポート番号〕" を含む URL のパスを指定します。また，新パスに指定する URL には，IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

旧パスでリクエストを受けた場合，指定したステータスコードと Location ヘッダに新パスを設定した応答を返します。通常，300 番台のステータスコードを受けた Web ブラウザは，自動的に Location ヘッダに指定されたアドレスに対してリダイレクトします。

Redirect ディレクティブでは，特定のファイルへのリクエストを特定のファイルへリダイレクトするか，特定のディレクトリ下の，任意のパスへのリクエストを特定のディレクトリ下の，同名パスへリダイレクトする指定ができます。特定のディレクトリ下の，任意のパスへのリクエストを，特定のファイルへリダイレクトしたい場合は RedirectMatch ディレクティブを使用してください。

permanent：ステータスコード 301 Moved Permanently を応答します。

temp：ステータスコード 302 Found を応答します。

seeother：ステータスコード 303 See Other を応答します。

gone：ステータスコード 410 Gone を応答します。新パスは指定できません。

ステータスコード：指定したステータスコードを応答します。指定できる値は，「付録 A　ステータスコード」を参照してください。ただし，300 番台以外を指定する場合，新パスは指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
Redirect temp /index.html http://ホスト名:port番号/default.html
```

/index.html に対するリクエストを，ステータスコード 302 で "http:// ホスト名 :port 番号 /default.html" にリダイレクトします。

(28) RedirectMatch 〔{permanent ｜ temp ｜ seeother ｜ gone ｜ ステータスコード}〕 正規表現 新パス

(a) 内容

正規表現で記述した条件を満たすパスに対するクライアントからのリクエストを，新パスに再リクエスト（リダイレクト）する場合に指定します。

正規表現には，スラッシュから始まるリクエスト URL の旧パスを正規表現で指定します。ただし，旧パスには，? 以降（問い合わせ文字列）を指定できません。

新パスには，" プロトコル名 :// ホスト名〔: ポート番号〕" を含む URL のパスを指定します。また，新パスに指定する URL には，IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

正規表現で括弧（）を使用してグループ化している場合，その i 番目のグループの表現にマッチした文字列を，新パスで $i を使用して参照できます。i には 1 から 9 までの数字を指定します。正規表現で記述した条件を満たすパスへのリクエストを受信した場合に，指定したステータスコードと，新パスを設定した Location ヘッダを応答します。通常，300 番台のステータスコードを受けた Web ブラウザは，自動的に Location ヘッダに指定されたアドレスに対して再リクエスト（リダイレクト）します。

各ステータスコードの指定については，Redirect ディレクティブを参照してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

(例 1)

```
RedirectMatch ^/other/ http://www.example.com/
```

/other/ で始まるすべてのリクエストを，ステータスコード 302 で "http://www.example.com/" にリダイレクトします。

(例 2)

```
RedirectMatch permanent ^/old/(.*) http://www.example.com/new/$1
```

"/old/ ファイル名 " に対するリクエストを，ステータスコード 301 で "http://www.example.com/new/ ファイル名 " にリダイレクトします。

(29) RequestHeader {{set | append | add} ヘッダ　ヘッダ値〔env=〔!〕環境変数〕 | unset　ヘッダ}

(a) 内容

クライアントから受信したヘッダ値をカスタマイズする場合に指定します。

set：ヘッダを設定します。ヘッダがある場合は，指定したヘッダ値に書き換えます。

append：存在するヘッダにヘッダ値を追加します。存在するヘッダ値との間は，コンマで区切られます。ヘッダがない場合は，ヘッダを設定します。

add：ヘッダがあっても，別の行にヘッダを設定します。同じヘッダを複数行設定する場合に使用します。

unset：指定したヘッダがある場合，そのヘッダをすべて削除します。

env= 環境変数：指定した環境変数が設定されている場合に，RequestHeader ディレクティブで指定した内容を実行します。

env=! 環境変数：指定した環境変数が設定されていない場合に，RequestHeader ディレクティブで指定した内容を実行します。

ヘッダ値に空白がある場合は，"（引用符）で囲む必要があります。ヘッダ値は文字だけから成る文字列，フォーマット指示子を含む文字列または両方から成る文字列を指定できます。フォーマット指示子を次に示します。

フォーマット指示子	意味
%t	リクエストを受け取った時刻を，1970 年 1 月 1 日 0 時 0 分 0 秒 (GMT：Greenwich Mean Time) から経過した時間で表示する。単位はマイクロ秒。先頭には "t=" が付けられる。
%D	リクエスト処理に掛かった時間を表示する。単位はマイクロ秒。先頭には "D=" が付けられる。
%{env_name}e	環境変数 env_name の値。

(b) 注意事項

ヘッダカスタマイズ機能を使用するためには mod_headers モジュールの組み込みが必要です。ヘッダカスタマイズ機能については，「4.10　ヘッダカスタマイズ機能」を参照してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(d) 上書き許可

FileInfo レベル

(e) 指定例

```
RequestHeader set Host www.example.com
```

(30) Require {user ユーザ名 〔ユーザ名 …〕 | group グループ名 〔グループ名 …〕 | valid-user | file-owner | file-group}

(a) 内容

AuthName ディレクティブ，AuthType ディレクティブ，AuthUserFile ディレクティブ（または AuthGroupFile ディレクティブ）と一緒に指定し，アクセス制限を定義します。

user：AuthUserFile ディレクティブで指定したパスワードファイルに登録されているユーザのうち，ユーザ名で指定したユーザだけアクセスできます。

group：AuthGroupFile ディレクティブで指定したグループファイルに登録されているグループ名で指定したグループに属するユーザだけがアクセスできます。

valid-user：AuthUserFile ディレクティブで指定したパスワードファイルに登録されているすべてのユーザまたは LDAPRequire ディレクティブで指定したグループのユーザだけがアクセスできます。パスワードファイルと LDAPRequire ディレクティブの組み合わせはできません。組み合わせた場合の動作は保証しません。

file-owner：AuthUserFile ディレクティブで指定したパスワードファイルに登録されているユーザのうち，アクセス対象ファイルのシステムの所有ユーザと一致しているユーザだけがアクセスできます（Windows 版では指定できません）。

file-group：AuthGroupFile ディレクティブで指定したグループファイルに登録されているグループ名で指定したグループに属するユーザのうち，グループ名がアクセス対象ファイルのシステムの所有グループに一致しているユーザだけがアクセスできます（Windows 版では指定できません）。

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

AuthConfig レベル

6.2.7 S で始まるディレクティブ

(1) Satisfy {any | all}

(a) 内容

コンテンツへのアクセスが，ユーザ認証（AuthUserFile，Require ディレクティブなどを指定）とホスト名または IP アドレス（Allow from，Deny from ディレクティブなどを指定）の両方によって制限されている場合にその関係を設定します。

any：そのどちらかの条件を満たしていれば，コンテンツへのアクセスを許可します。

all：そのどちらの条件も満たさなければ，コンテンツへのアクセスを禁止します。

(b) 記述できる場所

<Directory>，.htaccess

(2) Script メソッド CGI スクリプト名

(a) 内容

指定されたメソッドによるリクエストがあった場合に CGI スクリプト名で示すスクリプトを実行します。

指定できるメソッド：GET，POST，PUT，DELETE

メソッドは大文字，小文字を区別します。

ただし，GET メソッドの場合，スクリプトは問い合わせ引数があるときだけ（例えば，/foo.html?bar）呼ばれます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>

(c) 指定例

```
Script POST /cgi-bin/search
```

(3) ScriptAlias URL ディレクトリ名

(a) 内容

Web ブラウザから URL で指定された CGI プログラム実行のリクエストに対して，実行する CGI プログラムのあるディレクトリ名を指定します。

ディレクトリ名は，絶対パスで指定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
ScriptAlias /cgi-bin/ "C:/Program Files/Hitachi/httpsd/cgi-bin/"
```

(4) ScriptAliasMatch 正規表現 新パス

(a) 内容

Web ブラウザから指定された CGI プログラム実行要求の URL が正規表現で記述した条件を満たす場合，指定した新パスの CGI プログラムを実行します。正規表現で括弧（）を使用してグループ化している場合，その i 番目のグループの表現にマッチした文字列を，新パスで $i を使用して参照できます。i には 1 から 9 までの数字を指定します。

新パスは，絶対パスで指定してください。また，新パスの文字として，'$' または '&' を含める場合は，その文字の前に '¥' を付加してください。なお，$i を指定する際には，'$' の前に '¥' を付加する必要はありません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
ScriptAliasMatch ^/cgi-bin/(.*) "C:/Program Files/Hitachi/httpsd/
cgi-bin/$1"
```

(5) ScriptInterpreterSource { registry | script }

(a) 内容

CGI スクリプトの実行に使用されるインタプリタを定義します。

registry：レジストリが検索され，拡張子に関連づけられているプログラムがインタプリタとして使用されます。

script：スクリプト内の #! 行で指定されたインタプリタが使用されます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(6) ScriptLog　ファイル名

(a) 内容

CGI スクリプトのエラーログ出力先のファイルを指定します。ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

UNIX の場合，指定するファイルは，User ディレクティブで指定したユーザの権限で書き込みができるようになっている必要があります。

(b) 記述できる場所

httpsd.conf

(7) ScriptLogBuffer　バッファ数

～ ((0 － 2147483647))《1024》（単位：バイト）

(a) 内容

PUT，POST メソッドによるリクエストのボディ部のログを採取する場合の最大値をバイト単位で指定します。ScriptLog ディレクティブでエラーログ出力先のファイルを指定した場合だけ，この指定は有効になります。

このディレクティブでの指定値分の領域が，リクエスト処理中に確保されます。そのため，大きい値を指定すると，メモリ確保失敗となって，Web サーバが終了する場合があります。デフォルト値または必要最小限の値を指定することを推奨します。

(b) 記述できる場所

httpsd.conf

(8) ScriptLogLength　ファイルサイズ

～ ((0 － 2147483647))《10385760》(単位：バイト)

(a) 内容

CGI スクリプトのエラーログファイルの最大サイズをバイト単位で指定します。
ScriptLog ディレクティブでエラーログ出力先のファイルを指定した場合だけ指定が有効になります。

(b) 記述できる場所

httpsd.conf

(9) ServerAdmin　E-Mail アドレス

(a) 内容

サーバ管理者の E-Mail アドレスを指定します。ServerSignature ディレクティブで E-Mail を指定する場合は，必ず指定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
ServerAdmin www-admin@server.example.com
```

(10) ServerAlias　ホスト名 〔ホスト名 …〕

(a) 内容

サーバ名に基づくバーチャルホストで使用するホスト名（ServerName）の別名を指定します。IPv6 アドレスに対応したホスト名も指定できます。

(b) 記述できる場所

<VirtualHost>

(11) ServerName　サーバ名〔: ポート番号〕

(a) 内容

Hitachi Web Server のサーバ名およびポート番号を指定します。ポート番号を省略した場合は，Port ディレクティブ指定値が設定されます。

サーバ名は，FQDN（完全修飾ドメイン名）または IP アドレスで指定します。また，サーバ名には，IPv6 アドレスまたは IPv6 アドレスに対応した FQDN も指定できます。

IPv6 アドレスを指定し，かつポート番号を指定する場合は，IPv6 アドレスを [ ] で囲んでください。

UseCanonicalName ディレクティブ指定値に従い，Redirect ディレクティブの指定，イメージマップの利用または末尾を /（スラッシュ）で閉じないディレクトリ指定のリクエストなど，Web サーバでリダイレクトが指示された場合のリダイレクト先として Location ヘッダに設定されクライアントに返信されるため，クライアントからアクセスできるサーバ名を指定しなければなりません。このディレクティブの指定は必須です。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
ServerName www.example.com
ServerName fec0::123:4567:89ab:cdef
ServerName [fec0::123:4567:89ab:cdef]
ServerName [fec0::123:4567:89ab:cdef]:8080
```

(12) ServerPath　パス名

(a) 内容

サーバ名に基づくバーチャルホストで，Host ヘッダの代わりにパス名を利用して各ホストに接続する場合に指定します。

(b) 記述できる場所

<VirtualHost>

(13) ServerRoot　ディレクトリ名

～《/opt/hitachi/httpsd》(UNIX 版)

～《インストール先ディレクトリ》(Windows 版)

(a) 内容

サーバのルートディレクトリを絶対パスで指定します。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
ServerRoot "C:/Program Files/Hitachi/httpsd"
```

(14) ServerSignature　{On ｜ Off ｜ Email}

(a) 内容

Web サーバが作成するエラーメッセージなどのコンテンツのフッタに署名するかどうか

を指定します。

On：ServerTokens ディレクティブに従った文字列（Hitachi Web Server やバージョン番号など）および UseCanonicalName ディレクティブ指定値に従ったサーバ名とポート番号を表示します。

```
Hitachi Web Server 03-00 at www.example.com Port 80
```

Off：コンテンツのフッタに署名を表示しません。

Email：On を指定した場合の表示に加え ServerAdmin ディレクティブの指定値を mailto タグで追加します。

なお，On を指定した場合，ServerName ディレクティブに指定した IPv6 アドレスまたは IPv6 アドレスに対応したホスト名を表示できます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 指定例

```
ServerSignature On
```

(15) ServerTokens {Minimal | OS | Full | ProductOnly}

(a) 内容

HTTP レスポンスヘッダの Server ヘッダのフォーマットを設定します。それぞれの設定による Server ヘッダの値を次に示します。OS 種別には，Unix または Win32 が設定されます。Server ヘッダの値がどのように利用されるかはクライアントの仕様によります。

Minimal：Hitachi Web Server **バージョン番号**

OS：Hitachi Web Server **バージョン番号** (OS **種別**)

Full：Hitachi Web Server **バージョン番号** (OS **種別**) **付加** PP **で設定された情報**

ProductOnly：Hitachi Web Server

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
ServerTokens Full
```

(16) SetEnv 環境変数 値

(a) 内容

CGI スクリプトに任意の環境変数を渡す場合に設定する環境変数の値を指定します。このディレクティブを複数指定する場合，同じ環境変数に異なる値は指定できません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
SetEnv MY_ENV myenv
```

(17) SetEnvIf　リクエスト値　正規表現　環境変数〔= 値〕〔環境変数〔= 値〕 …〕

(a) 内容

クライアントからのリクエストを基に環境変数を定義します。クライアントからのリクエスト値が正規表現で表した条件を満たす場合，指定した環境変数を設定します。設定する値のデフォルト値は 1 です。環境変数の前に ! が付いたときは，その環境変数の設定を解除します。

リクエスト値としては，HTTP リクエストヘッダか次の表に示す値を指定できます。先に指定された環境変数をリクエスト値として指定することで環境変数の検査ができます。ただし，この場合の環境変数は，HTTP リクエストヘッダにも次の表に示す指定値にも一致していない必要があります。

リクエスト値	意味
Remote_Addr	クライアントの IP アドレス
Remote_Host	クライアントのホスト名（リクエストに設定されている場合だけ）
Request_Protocol	リクエストのプロトコル（HTTP/1.1　など）
Request_Method	リクエストのメソッド名（GET，POST，HEAD など）
Request_URI	リクエストの URI
Server_Addr	リクエストを受信したサーバの IP アドレス

このディレクティブを複数指定する場合，同じリクエスト値は複数指定できません。

なお，リクエスト値に Remote_Host を指定した場合，正規表現には IPv6 アドレスに対応したホスト名も指定できます。また，IPv6 を使用した接続に対しては，Remote_Addr と Server_Addr のリクエスト値は使用できません。Remote_Addr と Server_Addr を使用したい場合は，HWSSetEnvIfIPv6 ディレクティブで設定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

(例 1)

```
SetEnvIf User-Agent "Mozilla.*" SETENVIF_USER_AGENT=Mozilla
```

(例 2)

```
SetEnvIf Request_URI "¥.(gif)|(jpg)$" request_is_image
```

(例 3)

IPv4 を使用した接続のうち，特定のクライアントに対して環境変数を設定する場合は，次のように指定します。

```
Listen 123.123.123.123:80
Listen [fec0::123:4567:89ab:cdef]:80
<VirtualHost 123.123.123.123:80>
        SetEnvIf Remote_Addr ^234¥.234¥.234¥.234$ IPV4_CLIENT
</VirtualHost>
```

(18) SetEnvIfNoCase　リクエスト値　正規表現　環境変数〔= 値〕〔環境変数〔= 値〕 …〕

(a) 内容

クライアントからのリクエストを基に環境変数を定義します。クライアントからのリクエスト値が正規表現で表した条件を満たす場合，指定した環境変数を設定します。設定する値のデフォルト値は 1 です。環境変数の前に ! が付いたときは，その環境変数の設定を解除します。

リクエスト値に指定できる値については，SetEnvIf ディレクティブを参照してください。

ただし，このディレクティブでは，正規表現の大文字，小文字の区別をしません。また，このディレクティブを複数指定する場合，同じリクエスト値は複数指定できません。

なお，リクエスト値に Remote_Host を指定した場合，正規表現には IPv6 アドレスに対応したホスト名も指定できます。また，IPv6 を使用した接続に対しては，Remote_Addr と Server_Addr のリクエスト値は使用できません。Remote_Addr と Server_Addr を使用したい場合は，HWSSetEnvIfIPv6 ディレクティブで設定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(19) SetHandler　ハンドラ名

(a) 内容

指定した <Directory> またはアクセスコントロールファイルの範囲すべてのリクエストをハンドラ名で指定したハンドラに関連づける場合，指定します。ハンドラ名として none を指定すると，それまでの SetHandler ディレクティブの設定が無効になります。

(b) 記述できる場所

<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(20) SSLBanCipher　暗号種別 〔暗号種別 …〕

(a) 内容

指定した暗号種別でのアクセスを拒否し，クライアントにステータスコード 403 Forbidden を応答します。暗号種別を次に示します。

暗号種別	鍵交換方式※3	認証方式	対称鍵暗号方式	暗号鍵サイズ (bit)	メッセージ認証アルゴリズム
EXP-DES-CBC-SHA※1	RSA(512bit)	RSA	DES	40	SHA
EXP-RC2-CBC-MD5	RSA(512bit)	RSA	RC2	40	MD5
EXP-RC4-MD5	RSA(512bit)	RSA	RC4	40	MD5
EXP-DES-56-SHA※1	RSA(512, 1024bit)	RSA	DES	56	SHA
EXP-RC4-56-SHA※1	RSA(512, 1024bit)	RSA	RC4	56	SHA
DES-CBC-MD5※2	RSA(512, 1024, 2048bit)	RSA	DES	56	MD5
DES-CBC-SHA※1	RSA(512, 1024, 2048bit)	RSA	DES	56	SHA
RC2-CBC-MD5※2	RSA(512, 1024, 2048bit)	RSA	RC2	128	MD5
RC4-MD5	RSA(512, 1024, 2048bit)	RSA	RC4	128	MD5
RC4-SHA※1	RSA(512, 1024, 2048bit)	RSA	RC4	128	SHA
AES128-SHA※1	RSA(512, 1024, 2048bit)	RSA	AES	128	SHA
DES-CBC3-MD5※2	RSA(512, 1024, 2048bit)	RSA	DES	168	MD5
DES-CBC3-SHA※1	RSA(512, 1024, 2048bit)	RSA	DES	168	SHA
AES256-SHA※1	RSA(512, 1024, 2048bit)	RSA	AES	256	SHA

注※ 1

SSLv2 では使用できません。

注※ 2

SSLv2 でだけ使用できます。

注※ 3

表中のビット長は，それぞれの暗号種別に対応する Web サーバの秘密鍵のビット長を表します。Web サーバの秘密鍵のビット長が，対応するビット長より長い場合には，Web サーバは対応するビット長を持つ一時的な公開鍵と秘密鍵を作成します。例えば，sslckey genrsa ユティリティで秘密鍵のビット長を 1024 として Web サーバの秘密鍵を作成した場合，暗号種別 EXP-RC4-MD5 を使用して通信する際には，512bit の一時鍵が作成されます。一時鍵はサーバプロセス生成時に作成されます。一時鍵の作成にはオーバヘッドを伴うので，注意してください。Windows 版では，一時鍵は基本的にサーバに一つだけ作成されますが，バーチャルホスト構成の場合には複数個作成されることがあります。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(21) SSLCACertificateFile　ファイル名

(a) 内容

SSL でサーバ認証およびクライアント認証する場合，CA（認証局）の公開鍵（PEM 形式）のファイル名を指定します。複数の証明書ファイルを連結させて，一つのファイルに複数の証明書が混在できます。

ファイル名は，絶対パスで指定してください。

- **サーバ認証時の利用**
  チェーンした CA で発行されたサーバ証明書を使用して運用する場合，チェーン CA の証明書を設定します。
- **クライアント認証時の利用**
  クライアント証明書を発行した CA の証明書を設定します。チェーンされたクライアント証明書の場合，チェーン CA の証明書も設定します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
SSLCACertificateFile "C:/Program Files/Hitachi/httpsd/conf/ssl/
cacert/anycert.pem"
```

(22) SSLCACertificatePath　ディレクトリ　U

(a) 内容

SSL でサーバ認証およびクライアント認証する場合，CA の証明書（PEM 形式）へのハッシュリンクを格納したディレクトリを指定します。ハッシュリンクの作成および運用方法については「5.2.6　ハッシュリンクの作成（UNIX 版）」を参照してください。

クライアント証明書の検証をする場合にハッシュリンクが必要になります。取得した CA の証明書は，特定のディレクトリでハッシュリンクを作成し，このディレクトリを SSLCACertificatePath ディレクティブに指定してください。

ディレクトリ名は，絶対パスで指定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
SSLCACertificatePath /opt/hitachi/httpsd/conf/ssl/cacerts
```

(23) SSLCacheServerPath　パス名　U

(a) 内容

SSL セション管理キャッシュサーバ gcache へのパス名を指定します。パス名は絶対パスまたは ServerRoot ディレクティブからの相対パスで指定できます。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
SSLCacheServerPath /opt/hitachi/httpsd/sbin/gcache
```

(24) SSLCacheServerPort　{ポート番号｜パス名}　U

～ ((ポート番号を指定する場合は 1 － 65535))

(a) 内容

Web サーバ本体と SSL セション管理キャッシュサーバ gcache との間でデータ交換するためのポート番号またはパス名を指定します。パス名は絶対パスまたは ServerRoot ディレクティブからの相対パスで指定できます。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
SSLCacheServerPort logs/gcache_port
```

(25) SSLCacheServerRunDir　パス名

～《ServerRoot ディレクティブ指定値》

(a) 内容

SSL セション管理キャッシュサーバ gcache が動作するパス名を指定します。gcache がコアダンプを出力するディレクトリを指定するために使用します。パス名は絶対パスまたは ServerRoot ディレクティブからの相対パスで指定できます。パス名に指定したディレクトリには，User ディレクティブに指定したユーザの読み込み権限，書き込み権限および実行権限が必要です。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
SSLCacheServerRunDir /opt/hitachi/httpsd/logs
```

(26) SSLCertificateFile　ファイル名

(a) 内容

SSL で認証する場合，Web サーバの証明書（PEM 形式）のファイル名を指定します。

ファイル名は，絶対パスで指定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
SSLCertificateFile "C:/Program Files/Hitachi/httpsd/conf/ssl/
server/httpsd.pem"
```

(27) SSLCertificateKeyFile　ファイル名

～《SSLCertificateFile ディレクティブ指定値》

(a) 内容

SSL で認証する場合，Web サーバの秘密鍵のファイル名を指定します。

ファイル名は，絶対パスで指定してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
SSLCertificateKeyFile "C:/Program Files/Hitachi/httpsd/conf/ssl/
server/httpsdkey.pem"
```

(28) SSLCertificateKeyPassword　パス名

(a) 内容

パスワード保護をされているサーバ秘密鍵のパスワードを格納しておくファイルのパス名を指定します。パスワード格納ファイルは，sslpasswd ユティリティによって作成します。パス名は絶対パスまたは ServerRoot ディレクティブからの相対パスで指定します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(29) SSLCRLAuthoritative　{On ｜ Off}

(a) 内容

SSL クライアント認証時に使用する CRL の次回発行日の扱いについて指定します。

On：

SSL クライアント認証時，クライアント証明書に対応する CRL の次回発行日を過ぎていた場合，Web サーバは認証に失敗したとして，クライアントとの接続を拒否します。CRL の正しい運用が必要です。

Off：

CRL の次回発行日を無視します。次回発行日を過ぎていても CRL は有効であると扱うため，CRL に登録されていなければ，クライアントは接続できます。セキュリティレベルは下がりますが，CRL を正しく運用しなかったときでも，最低限のセキュリティを維持して，サービスを継続できます。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
SSLCRLAuthoritative On
```

CRL の次回発行日を過ぎている場合，その CRL を発行した CA が発行した証明書を持つクライアントのアクセスはすべて拒否します。

(30) SSLCRLDERPath　パス名

(a) 内容

DER 形式の CRL を格納するディレクトリを絶対パスで指定します。指定したディレクトリに必要な CRL を格納して，Web サーバを起動または再起動すると，SSL でのクライアント認証時に CRL を適用できます。SSLCRLPEMPath ディレクティブで指定したディレクトリ内の CRL も含め，同じ CA から発行された CRL を複数格納している場合，発行日が最新の CRL を適用します。ディレクトリ内に DER 形式の CRL 以外のファイ

ルがある場合，Web サーバは起動しません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
SSLCRLDERPath "C:/Program Files/Hitachi/httpsd/conf/ssl/crl/DER"
```

DER 形式の CRL ファイルを格納しているディレクトリを指定します。

(31) SSLCRLPEMPath　パス名

(a) 内容

PEM 形式の CRL を格納するディレクトリを絶対パスで指定します。指定したディレクトリに必要な CRL を格納して，Web サーバを起動または再起動すると，SSL でのクライアント認証時に CRL を適用できます。SSLCRLDERPath ディレクティブで指定したディレクトリ内の CRL も含め，同じ CA から発行された CRL を複数格納している場合，発行日が最新の CRL を適用します。ディレクトリ内に PEM 形式の CRL 以外のファイルがある場合，Web サーバは起動しません。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
SSLCRLPEMPath "C:/Program Files/Hitachi/httpsd/conf/ssl/crl/PEM"
```

PEM 形式の CRL ファイルを格納しているディレクトリを指定します。

(32) SSLDenySSL

(a) 内容

SSL によるアクセスを禁止する場合に指定します。このディレクティブが指定されている場合，SSLEnable ディレクティブで SSL を有効にしても https によるアクセスがステータスコード 403 Forbidden で拒否されます。SSLRequireSSL ディレクティブの逆の動作をします。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(33) SSLDisable

(a) 内容

SSL を無効にします。デフォルト値は SSLEnable ディレクティブ（SSL を有効）です。バーチャルホストで特定のホストに対して SSL を無効にする場合などに指定します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(34) SSLEnable

(a) 内容

SSL を有効にします。SSLDisable ディレクティブを指定しないかぎり，デフォルトで有効になります。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(35) SSLExportCertChainDepth　値

～ ((0 － 9))《0》

(a) 内容

SSL クライアント認証をする場合，環境変数 SSL_CLIENT_CERT_CHAIN_n にクライアントの証明書を発行した CA からルート CA までの証明書を設定するときに指定します。指定した値が n の最大値になります。このディレクティブは SSLExportClientCertificates ディレクティブを指定している場合だけ有効になります。指定された数の CA 証明書が gcache サーバへキャッシュされるため，CGI または Servlet で必要な数だけをこのディレクティブに指定することでキャッシュを有効に利用できます。ただし，メモリの制限でキャッシュされた一部の証明書が削除されて取得できなかった場合は，取得できたものだけを環境変数に設定します。

0：

環境変数は設定しません。

1 ～ 9：

クライアントの証明書に近い方から順に番号が割り当てられ，環境変数を設定します。環境変数には DER 形式の証明書を Base64 エンコーディングした値を設定します。一つの証明書を Base64 エンコーディングした場合のバイト数は約 1KB です。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

「ルート CA －下位 CA －クライアント証明書」という証明書チェーンの場合を説明します。この場合，環境変数と証明書の対応は次のようになります。

環境変数	証明書
SSL_CLIENT_CERT	クライアント証明書
SSL_CLIENT_CERT_CHAIN_1	下位 CA の証明書
SSL_CLIENT_CERT_CHAIN_2	ルート CA の証明書

この環境変数と証明書チェーンをすべて取得するには，次のようにディレクティブを指定します。

```
SSLExportClientCertificates
SSLExportCertChainDepth 2       2以上の値を指定します
```

(36) SSLExportClientCertificates

(a) 内容

SSL クライアント認証をする場合，環境変数 SSL_CLIENT_CERT にクライアント証明書を設定するときに指定します。環境変数 SSL_CLIENT_CERT には DER 形式の証明書を Base64 エンコーディングした値を設定します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(37) SSLFakeBasicAuth

(a) 内容

SSL クライアント認証の機能と併せて，Web ブラウザでユーザ ID とパスワードを入力することなく，クライアント証明書の提示だけで Basic 認証をできるようにします。AuthUserFile ディレクティブで指定するファイルには X509 クライアント証明書の Subject とパスワードを記述します。パスワードは，次に示す値で常に固定とします（どちらも "password" を暗号化したもの）。

- UNIX 版："xxj31ZMTZzkVA"
- Windows 版："{SHA}W6ph5Mm5Pz8GgiULbPgzG37mj9g="

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

証明書の Subject フィールドの値

```
Subject:
C=JP,ST=Kanagawa,L=Yokohama-shi,O=HITACHI,OU=Software,CN=userna
me/Email=username@userhost
```

この場合 AuthUserFile ディレクティブで指定するファイルは次のように指定します。

UNIX 版の場合

```
/C=JP/ST=Kanagawa/L=Yokohama-shi/O=HITACHI/OU=Software/
CN=username/Email=username@userhost:xxj31ZMTZzkVA
```

Windows 版の場合

```
/C=JP/ST=Kanagawa/L=Yokohama-shi/O=HITACHI/OU=Software/
CN=username/
Email=username@userhost:{SHA}W6ph5Mm5Pz8GgiULbPgzG37mj9g=
```

LogFormat ディレクティブの u 指定では Subject がロギングされます。

認証に失敗した場合，ステータスコード 401 Authorization Required を応答します。

(38) SSLProtocol プロトコル名 〔プロトコル名 …〕

～《All》

(a) 内容

使用する SSL プロトコルのバージョンを指定します。

プロトコル名として設定できる値は次のとおりです。

SSLv2: SSL プロトコルバージョン 2 を使用する。

SSLv3: SSL プロトコルバージョン 3 を使用する。

TLSv1: TLS プロトコルバージョン 1 を使用する。

All: 上記すべてのプロトコルを使用する。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(39) SSLRequireCipher 暗号種別 〔暗号種別 …〕

(a) 内容

指定した暗号種別以外でのアクセスを拒否し，クライアントにステータスコード 403 Forbidden を応答します。指定できる暗号種別は，SSLBanCipher ディレクティブを参照してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(40) SSLRequiredCiphers　暗号種別 〔: 暗号種別 …〕

(a) 内容

SSL 通信で使用可能とする暗号種別を指定します。ディレクティブに指定した暗号種別とクライアントが使用できる暗号種別との間で一致するものがあれば，SSL 通信が確立され HTTP リクエストを受信します。一致するものがない場合は，SSL 通信は確立されないで HTTP リクエストを受信しません。指定できる暗号種別は，SSLBanCipher ディレクティブを参照してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
SSLRequiredCiphers
EXP-RC4-MD5:RC4-MD5:EXP-RC2-CBC-MD5:DES-CBC-SHA:DES-CBC3-SHA
```

(41) SSLRequireSSL

(a) 内容

SSL 以外によるアクセスを禁止する場合に指定します。このディレクティブが指定されている場合，SSLDisable ディレクティブで SSL を無効にしても http によるアクセスがステータスコード 403 Forbidden で拒否されます。異なるディレクティブの記述場所で，不用意に SSL を無効にしコンテンツを公開してしまうことを防止します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
<VirtualHost 172.17.40.10:443>
 SSLDisable
 ...
  <Directory /secure/dir>
    SSLRequireSSL
    ...
  </Directory>
</VirtualHost>
```

172.17.40.10 ホストの 443 ポートに対する http アクセスは，/secure/dir ディレクトリへのアクセスを除いてできます。/secure/dir ディレクトリへの http アクセスは，ステータスコード 403 Forbidden を応答します。

(42) SSLSessionCacheTimeout　値

～ ((1 － 2147483647))《3600》（単位：秒）

(a) 内容

Web サーバ内または SSL セション管理キャッシュサーバ gcache 内で保持されるセション ID などのデータの有効時間を秒単位で指定します。

(b) 注意事項

グリニッジ標準時（GMT）の 2038 年 1 月 19 日 3 時 14 分 7 秒を超えないように，有効時間を設定してください。

(c) 記述できる場所

httpsd.conf，<VirtualHost>

(d) 指定例

```
SSLSessionCacheTimeout 3600
```

(43) SSLSessionCacheSize　{サイズ｜ max}

～ ((0 － 2147483647))《16777216》（単位：バイト）

(a) 内容

UNIX 版の場合，SSL セションを管理するキャッシュサーバ gcache 内のメモリにキャッシュされるセション ID などのデータの，メモリサイズの上限値をバイト単位で指定します。0 を指定した場合，gcache サーバは起動しないで，セションキャッシュは実施されません。

Windows 版の場合，セションキャッシュサイズの上限値をバイト単位で指定します。0 を指定した場合，セションキャッシュは実施されません。

max を指定した場合，上限を設定しません。1 SSL セション当たり，サーバ認証だけする場合約 200 バイト，クライアント認証もする場合約 1 キロバイト使用します。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
SSLSessionCacheSize 1024
```

(44) SSLSessionCacheSizePerChild　{サイズ｜ max}　U

～ ((0 － 2147483647))《20480》（単位：バイト）

(a) 内容

サーバプロセス内のメモリにキャッシュされるセション ID などのデータの，メモリサイ

ズの上限値をバイト単位で指定します。max を指定した場合，上限を設定しません。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
SSLSessionCacheSizePerChild 1024
```

(45) SSLVerifyClient {0 | 1 | 2 | 3}

～《0》

(a) 内容

クライアント認証時の証明書に関する設定を指定します。

0：証明書の要求をしません。

1：クライアントは証明書を提示できます。運用テスト用。

2：クライアントは証明書を提示しなければなりません。

3：クライアントは証明書を提示できますが，サーバが鍵を持っている CA（認証局）が発行したものである必要はありません。運用テスト用。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
SSLVerifyClient 2
```

(46) SSLVerifyDepth 段階数

～ ((0 － 10))《0》

(a) 内容

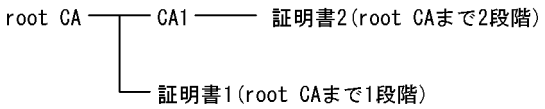
証明書のチェーンを何段階までたどるかを指定します。

クライアント認証に使用する CA 証明書のチェーンについて，認証チェックをする段階数を指定します。チェーンされた CA をどこまで信用するかを制限するために使用します。自己署名の証明書は認証しないため，段階数は 2 以上を指定します。例を次に示します。

(例)

条件

- CA1 は，root CA に署名されている。
- 証明書 1 は，root CA に署名されている。
- 証明書 2 は，CA1 に署名されている。



SSLVerifyDepth の指定

この場合，証明書 1，証明書 2 とも認証チェックをするためには，SSLVerifyDepth ディレクティブに 3 以上を指定します。また，証明書 1 は認証チェックし，証明書 2 は認証チェックをしないようにするには SSLVerifyDepth ディレクティブに 2 を指定します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
SSLVerifyDepth 10
```

(47) StartServers　プロセス数　U

～((0 － 1024))《5》

(a) 内容

Web サーバ起動時のサーバプロセス数を指定します。プロセス数に関連するほかのディレクティブについては，「4.1　Hitachi Web Server の処理とディレクティブとの関係」を参照してください。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
StartServers 5
```

6.2.8　T，U で始まるディレクティブ

(1) Timeout　時間

～((0 － 65535))《300》（単位：秒）

(a) 内容

次の待ち時間を秒単位で指定します。0 を指定した場合は，待ち時間が 0 秒になります。

- クライアントからのリクエスト受信（コネクション確立後，HTTP プロトコルの受信）中にデータを受信しなくなった場合の待ち時間
- クライアントへのレスポンス送信中にデータを送信できなくなった場合の待ち時間
- CGI プログラムへのリクエスト送信中にデータを送信できなくなった場合の待ち時間

- CGI プログラムへのリクエスト送信後からレスポンス受信までの待ち時間
- CGI プログラムからのレスポンス受信中にデータを受信しなくなった場合の待ち時間
- リバースプロキシを使用している場合の，バックエンドサーバへのリクエスト送信中にデータを送信できなくなった場合の待ち時間
- リバースプロキシを使用している場合の，バックエンドサーバへのリクエスト送信後からレスポンス受信までの待ち時間
- リバースプロキシを使用している場合の，バックエンドサーバからのレスポンス受信中にデータを受信しなくなった場合の待ち時間

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
Timeout 300
```

(2) ThreadsPerChild　スレッド数 W

～ ((1 － 1024))《40》

(a) 内容

サーバとして起動するスレッド数を指定します。指定したスレッド数はサーバの最大同時接続数を示します。

(b) 記述できる場所

httpsd.conf

(3) TraceEnable　{On ｜ Off ｜ extended}

(a) 内容

TRACE メソッドによるリクエストを拒否するかどうかを指定します。

On：TRACE メソッドによるリクエストを許可します。ただし，リクエストボディが付加されている場合は，413 Request Entity Too Large を応答します。

Off：TRACE メソッドによるリクエストを拒否します。TRACE メソッドによるリクエストの場合は，ステータスコード 403 Forbidden を応答します。

extended：TRACE メソッドによるリクエストを許可します。リクエストボディが付加されていても許可します。ただし，リバースプロキシ以外のリクエストボディサイズの上限は 64KB です。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
TraceEnable Off
```

(4) TransferLog {ファイル名|パイプ}

(a) 内容

ログを格納するファイルまたはログを出力するプログラムを指定します。ログのフォーマットはラベル名を指定しない LogFormat ディレクティブで指定できます。

LogFormat ディレクティブでログのフォーマットを指定する場合は，IPv6 アドレスや IPv6 アドレスに対応したホスト名も出力できます。指定できるフォーマットは CustomLog ディレクティブを参照してください。

LogFormat ディレクティブでフォーマットを指定しない場合は，標準のログフォーマットで出力します。

ファイル名：ログを格納するファイル名を指定します。ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

パイプ：標準入力からログ情報を受け取るプログラムを "| プログラム名 " のフォーマットで指定します。Windows 版での注意事項は，CustomLog ディレクティブを参照してください。

(b) 記述できる場所

httpsd.conf，<VirtualHost>

(c) 指定例

```
TransferLog "|¥"¥"C:/Program Files/Hitachi/httpsd/sbin/
rotatelogs.exe¥" ¥"C:/Program Files/Hitachi/httpsd/logs/access¥"
86400¥""
```

rotatelogs ユティリティを使用してログを 24 時間ごとに分割して採取します。

(5) TypesConfig ファイル名

～《conf/mime.types》

(a) 内容

ファイル拡張子とコンテントタイプ（MIME タイプ）の関係を定義する設定ファイルを指定します。ファイル名には，絶対パスまたは ServerRoot ディレクティブの指定値からの相対パスが指定できます。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
TypesConfig conf/mime.types
```

MIME タイプの設定ファイルは mime.types

(6) UnsetEnv 環境変数 〔環境変数 …〕

(a) 内容

CGI スクリプトに渡す環境変数から，SetEnv ディレクティブまたは PassEnv ディレクティブで指定した環境変数を削除する場合に指定します。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>，.htaccess

(c) 上書き許可

FileInfo レベル

(d) 指定例

```
UnsetEnv MY_ENV
```

(7) UseCanonicalName {On | Off | dns}

(a) 内容

サーバの正式な名前の生成方法を指定します。サーバの正式な名前は，自サーバを参照する URL や環境変数の SERVER_NAME と SERVER_PORT に設定されます。

On：サーバの正式な名前は，ServerName ディレクティブ指定値から作成され，自サーバを参照する URL や環境変数に設定されます。VirtualHost 指定時に IP アドレスを使用する場合は，VirtualHost ブロック内で ServerName を指定してください。ブロック内で ServerName を指定していない場合は，IP アドレスからホスト名を取得します。

Off：サーバの正式な名前は，Host ヘッダによってクライアントから与えられたホスト名称とポート番号から作成され，自サーバを参照する URL や環境変数に設定されます。ただし，Host ヘッダが与えられない場合は，ServerName ディレクティブ値と，実際のコネクションに使用されているポート番号から作成されます。

dns：Host ヘッダを持たない古いクライアントのためのオプションです。このオプション指定時には，サーバの正式な名前は，クライアントから与えられたサーバの IP アドレスから逆引きしたホスト名称および実際にコネクションに使用されているポート番号から作成され，自サーバを参照する URL や環境変数に設定されます。

なお，On，Off，dns すべての場合で，IPv6 アドレスに対応しています。

(b) 記述できる場所

httpsd.conf，<VirtualHost>，<Directory>

(8) User ユーザ名 U

～《#-1》

(a) 内容

サーバプロセスが動作するときのユーザ名を指定します。

(b) 記述できる場所

httpsd.conf

(c) 指定例

```
User nobody
```

(9) UserDir {ディレクトリ名 | disabled 〔ユーザ名 〔ユーザ名 …〕〕}

～《public_html》(UNIX 版)

～《disabled》(Windows 版)

(a) 内容

Web ブラウザからの /~ ユーザ名 / へのリクエストに対して公開するサーバ上の場所をディレクトリ名で指定します。disabled を指定すると，Web コンテンツを公開しないユーザを指定できます。

ディレクトリ名は，相対パスまたは絶対パスで指定します。

Windows 版では，絶対パスだけ有効です。

ディレクトリ名：

- 相対パスで指定した場合
  サーバ上にユーザ ID を持つユーザが，ユーザのホームディレクトリ下の Web コンテンツを公開する場合の場所を指定します。/~ ユーザ名 / へのリクエストがあった場合，" ユーザのホームディレクトリ / ディレクトリ名 " にアクセスします。
- 絶対パスで指定した場合
  ユーザディレクトリの場所を指定します。/~ ユーザ名 / へのリクエストがあった場合，" ディレクトリ名 / ユーザ名 " にアクセスします。

disabled：

Web ブラウザからの /~ ユーザ名 / へのリクエストに対して，Web コンテンツを公開しないユーザを指定します。指定されたユーザ名でのリクエストに対しては，アクセスするディレクトリ名を変換しません。ユーザ名の指定がない場合は，すべてのユーザについて disabled を指定したことになります。

(b) 注意事項

- 複数の UserDir ディレクティブでディレクトリ名を指定した場合，後から指定したものに上書きされます。
- disabled に指定するユーザ名は，複数の UserDir ディレクティブを用いて指定できます。

(c) 記述できる場所

httpsd.conf, <VirtualHost>

(d) 指定例

(例 1)

```
UserDir public_html
```

ユーザ user1 のホームディレクトリを /home/user1 とすると，リクエスト http:// ホスト名〔: ポート番号〕/~user1/index.html で，/home/user1/public_html/index.html にアクセスします。

(例 2)

```
UserDir /home
UserDir disabled user3
UserDir disabled user4 user5
```

リクエスト http:// ホスト名〔: ポート番号〕/~user1/index.html で，/home/user1/index. html にアクセスします。ただし，user3 は disabled が指定されているため，http:// ホスト名〔: ポート番号〕/~user3/index.html というリクエストで /home/user3/index.html にアクセスできません。user4，user5 についても user3 と同様です。

(例 3)

```
UserDir disabled
```

すべてのユーザに対して disabled を指定します。

7 メッセージ

この章では，Hitachi Web Server で出力するメッセージについて説明します。

7.1 メッセージの形式

(1) 出力形式

メッセージは，時刻，エラーレベルとそれに続くメッセージテキストで構成されているものと，メッセージテキストだけで構成されているものがあります。形式を次に示します。

[時刻] [エラーレベル] メッセージテキスト
または
メッセージテキスト

(2) メッセージの記法

このマニュアルでは次の形式で説明しています。

メッセージテキスト

メッセージの説明文

エラーレベル：エラーログに出力するエラーのレベルを示します。エラーレベルはLogLevel ディレクティブで指定します。エラーレベルが「なし」のメッセージは，レベル設定のないメッセージで，メッセージテキストだけ出力します。

(S) システムの処置を示します。

(O) メッセージが出力されたときに，オペレータの取る処置を示します。

(3) メッセージの出力先

メッセージの出力先には次の 3 種類があります。

- 標準出力
- 標準エラー出力
- エラーログファイル

(4) 注意事項

notice レベルのメッセージは，LogLevel ディレクティブの指定に関係なく出力されます。

Hitachi Web Server 起動時など，レベル指定解析前には，LogLevel ディレクティブの指定に関係なくメッセージが出力される場合があります。

次に示すメッセージは一部を除き，記載していません。

- Hitachi Web Server の起動時に出力される，コンフィグファイルの文法エラーに伴うメッセージ
- Hitachi Web Server の起動後に出力される，エラーレベルが debug のメッセージ

- Hitachi Web Server の起動後に出力される，エラーレベルがないメッセージ

メッセージテキストの説明で「詳細情報」と記載している部分には，「(エラーコード) エラー文字列」などが出力されます。

7.2 メッセージ一覧

7.2.1 基本機能についてのメッセージ

(1) emerg レベルのメッセージ

詳細情報 : OpenEvent on **イベント名称** event

イベントのオープンに失敗しました。

エラーレベル：emerg

(S) Web サーバは停止，再起動処理を中断します。

(O) OpenEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : SetEvent on **イベント名称** event

イベントの通知に失敗しました。

エラーレベル：emerg

(S) Web サーバは停止，再起動処理を中断します。

(O) SetEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : couldn't grab the accept mutex

accept の排他獲得処理に失敗しました。

エラーレベル：emerg

(S) サーバプロセスを終了します。

(O) 詳細情報について見直してください。

詳細情報 : couldn't release the accept mutex

accept の排他解除処理に失敗しました。

エラーレベル：emerg

(S) サーバプロセスを終了します。

(O) 詳細情報について見直してください。

詳細情報 : Couldn't initialize cross-process lock in child

サーバプロセス間で使用する排他の初期化処理に失敗しました。

エラーレベル：emerg

(S) サーバプロセスを終了します。

(O) 詳細情報について見直してください。

詳細情報 : Couldn't create accept lock **ファイル名**

accept の排他環境生成に失敗しました。なお，ファイル名は出力されないことがあります。

エラーレベル：emerg

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Couldn't set permissions on cross-process lock; check User and Group directives

サーバプロセス間で使用する排他のパーミッション設定に失敗しました。

エラーレベル：emerg

(S) Web サーバは起動処理を中断します。

(O) User および Group ディレクティブ値を**詳細情報**について見直してください。

詳細情報 : apr_accept: giving up.

accept() 関数でエラー（ENETDOWN）が発生しました。

エラーレベル：emerg

(S) リクエストを受けたサーバプロセスを終了します。

(O)accept() 関数が返す**詳細情報**について見直してください。

（2） alert **レベルのメッセージ**

[client **クライアントアドレス**] **アクセスコントロールファイル名** : **エラーメッセージ**

アクセスコントロールファイルを解析中にエラーが発生しました。

エラーレベル：alert

(S) ステータスコード「500　Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) アクセスコントロールファイルを**エラーメッセージ**について見直してください。

---

no listening sockets available, shutting down

listen 状態のソケットがありません。

エラーレベル：alert

(S)Web サーバは起動処理を中断します。

(O)Listen，BindAddress ディレクティブまたは Port ディレクティブについて，ソケット生成が失敗する原因がないか見直してください。

---

httpsd: Could not determine the server's fully qualified domain name, using Web **サーバ名称** for ServerName

Web サーバの完全なドメイン名称を解決できなかったため，ServerName として Web **サーバ名称**を使用します。

エラーレベル：alert

(S)ServerName に Web **サーバ名称**を使用します。

Child **サーバプロセス** ID returned a Fatal error... Server is exiting!

サーバプロセスは，続行できないエラーのため終了しました。

エラーレベル：alert

(S)Web サーバは終了します。

(O) このメッセージのほかに出力されているメッセージのエラーの原因について見直してください。

詳細情報 : getpwuid: couldn't determine user name from uid **ユーザ** ID, you probably need to modify the User directive

パスワード・データベース内の**ユーザ** ID の検索処理でエラーが発生しました。User ディレクティブを見直す必要があります。

エラーレベル：alert

(S)Web サーバは起動処理を中断します。

(O)User ディレクティブでの指定値と getpwuid() 関数が返す**詳細情報**について見直して

ください。

詳細情報 : setgid: unable to set group id to Group **グループ** ID

グループ ID の設定に失敗しました。

エラーレベル：alert

(S) Web サーバは起動処理を中断します。

(O)setgid() 関数が返す**詳細情報**について見直してください。

詳細情報 : initgroups: unable to set groups for User **ユーザ名** and Group **グループ** ID

ユーザに関するグループアクセスリストの初期化に失敗しました。

エラーレベル：alert

(S) Web サーバは起動処理を中断します。

(O)initgroups() 関数が返す**詳細情報**について見直してください。

詳細情報 : setuid: unable to change to uid: **ユーザ** ID

ユーザ ID の設定に失敗しました。

エラーレベル：alert

(S) Web サーバは起動処理を中断します。

(O)setuid() 関数が返す**詳細情報**について見直してください。

詳細情報 : set dumpable failed - this child will not coredump after software errors

コアダンプの設定に失敗しました。サーバプロセスは通常コアダンプするようなエラーが発生してもコアを出力しません。

エラーレベル：alert

(S) Web サーバは起動処理を続行します。

(O)prctl() 関数が返す**詳細情報**について見直してください。

（3） crit **レベルのメッセージ**

[client **クライアントアドレス**] **詳細情報** : **アクセスコントロールファイル名** pcfg_openfile: unable to check htaccess file, ensure it is readable

アクセスコントロールファイルを読み込めません。

エラーレベル：crit

(S) ステータスコード「403　Forbidden」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報に示す原因についてアクセスコントロールファイルを見直してください。

詳細情報 : unable to replace stderr with error_log

標準エラー出力ファイル記述子をエラーログのファイル記述子に複製できません。

エラーレベル：crit

(S) Web サーバは起動処理を続行します。

(O) 詳細情報について見直してください。

詳細情報 : unable to replace stderr with /dev/null

標準エラー出力を /dev/null で置き換えられません。

エラーレベル：crit

(S) Web サーバは起動処理を続行します。

(O)freopen() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **プロセス** ID: Failed to create a max_requests event.

サーバプロセスは最大リクエスト数のためのイベント生成に失敗しました。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) CreateEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **プロセス** ID: _beginthreadex failed. Unable to create all worker threads. Created **生成したスレッド数** of the **ディレクティブ値** threads requested with the ThreadsPerChild configuration directive.

ThreadsPerChild ディレクティブに指定された数の Web サーバスレッドを生成中に，スレッド生成に失敗しました。

エラーレベル：crit

(S) サーバプロセスは制御プロセスに対して，Web サーバの停止を要求します。

(O)_beginthreadex() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **プロセス** ID: WAIT_FAILED -- shutting down server

サーバプロセスは終了，計画停止または最大リクエスト数のイベント待ちに失敗しました。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) WaitForMultipleObjects() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **サーバプロセス** ID: Unable to retrieve the ready event from the parent

サーバプロセスの開始イベントハンドルを制御プロセスから取得できませんでした。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) ReadFile() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **サーバプロセス** ID: Unable to retrieve the exit event from the parent

サーバプロセスの終了イベントハンドルを制御プロセスから取得できませんでした。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) ReadFile() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **サーバプロセス** ID: Unable to retrieve the start_mutex from the parent

排他用のハンドラを制御プロセスから取得できませんでした。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) ReadFile() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **サーバプロセス** ID: Unable to access the start_mutex from the parent

排他用のハンドラにアクセスできませんでした。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) **詳細情報**について見直してください。

詳細情報 : Child **サーバプロセス** ID: Unable to retrieve the scoreboard from the parent

スコアボード用のハンドラを制御プロセスから取得できませんでした。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O)ReadFile() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **サーバプロセス** ID: Unable to access the scoreboard from the parent

スコアボードにアクセスできませんでした。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O)MapViewOfFile() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **サーバプロセス** ID: Unable to reopen the scoreboard from the parent

スコアボードをオープンできませんでした。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) **詳細情報**について見直してください。

詳細情報 : Parent: Unable to duplicate the ready event handle for the child

サーバプロセスの開始イベントハンドルを複製できませんでした。

エラーレベル：crit

(S)Web サーバは起動処理を中断します。

(O)DuplicateHandle() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Unable to send the ready event handle to the child

サーバプロセスに開始イベントハンドルを送信できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Unable to duplicate the exit event handle for the child

サーバプロセスの終了イベントハンドルを複製できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) DuplicateHandle() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Unable to send the exit event handle to the child

サーバプロセスに終了イベントハンドルを送信できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Unable to retrieve the start mutex for the child

排他用のハンドルを取得できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Unable to duplicate the start mutex to the child

排他用のハンドルを複製できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) DuplicateHandle() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Unable to duplicate the start mutex to the child

排他用のハンドルを複製できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) DuplicateHandle() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Unable to send the start mutex to the child

サーバプロセスに排他用のハンドルを送信できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Unable to retrieve the scoreboard handle for the child

スコアボード用のハンドルを取得できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Unable to duplicate the scoreboard handle to the child

スコアボード用のハンドルを複製できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) DuplicateHandle() 関数が返す**詳細情報**について見直してください。

詳細情報 : Unable to send the scoreboard handle to the child

サーバプロセスにスコアボード用のハンドルを送信できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : setup_inherited_listeners: Unable to read socket data from parent.

制御プロセスからのソケットデータを読み込めません。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O)ReadFile() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **サーバプロセス** ID: setup_inherited_listeners(), WSASocket failed to open the inherited socket.

継承したソケットのオープンに失敗しました。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O)WSASocket() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: WSADuplicateSocket failed for socket fd **番号** .

制御プロセスはソケットの複製に失敗しました。

エラーレベル：crit

(S) サーバプロセス生成処理を中断します。

(O)WSADuplicateSocket() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Unable to write duplicated socket fd **番号** to the child.

制御プロセスはソケットを通してサーバプロセスへの書き込みに失敗しました。

エラーレベル：crit

(S) サーバプロセス生成処理を中断します。

(O) **詳細情報**について見直してください。

詳細情報 : Parent: Failed to get the current path

カレントパス情報の取得に失敗しました。

エラーレベル：crit

(S)Web サーバは起動処理を続行します。

(O) **詳細情報**について見直してください。

詳細情報 : Parent: Failed to get full path of **起動コマンド**

起動コマンドの絶対パス取得に失敗しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Unable to create child stdin pipe.

サーバプロセスに情報を送信するためのパイプ生成に失敗しました。

エラーレベル：crit

(S) サーバプロセス生成処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Unable to connect child stdout to NUL.

サーバプロセスから情報を取得するための NUL ハンドルへの接続に失敗しました。

エラーレベル：crit

(S) サーバプロセス生成処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Unable to connect child stderr.

サーバプロセスに設定する標準エラー出力に接続できませんでした。

エラーレベル：crit

(S) サーバプロセス生成処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Could not create ready event for child process

サーバプロセスの開始イベント生成に失敗しました。

エラーレベル：crit

(S) サーバプロセス生成処理を中断します。

(O) CreateEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Could not create exit event for child process

サーバプロセスの終了イベント生成に失敗しました。

エラーレベル：crit

(S) サーバプロセス生成処理を中断します。

(O) CreateEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Failed to create the child process.

サーバプロセスの生成に失敗しました。

エラーレベル：crit

(S) サーバプロセス生成処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : master_main: create child process failed. Exiting.

制御プロセスはサーバプロセスの生成に失敗しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : master_main: WaitForMultipleObjects WAIT_FAILED -- doing server shutdown

制御プロセスは停止，計画停止または再起動のイベント待ちに失敗しました。

エラーレベル：crit

(S) Web サーバは終了します。

(O) WaitForMultipleObjects() 関数が返す**詳細情報**について見直してください。

詳細情報 : Failed to get the full path of **起動コマンド**

起動コマンドの絶対パス取得に失敗しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : **サービス名称** : Unable to start the service manager.

サービス起動に失敗しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Cannot create shutdown event **イベント名称**

制御プロセスは停止のためのイベント生成に失敗しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) CreateEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Cannot create restart event **イベント名称**

制御プロセスは再起動のためのイベント生成に失敗しました。

エラーレベル：crit

(S) Web サーバは再起動処理を中断します。

(O) CreateEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : **サービス名称** : Failed to start the service process.

サービスの起動に失敗しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : make_sock: for address Web **サーバアドレス** : **ポート番号** , apr_socket_opt_set: (SO_KEEPALIVE)

ソケットレベルオプション SO_KEEPALIVE の設定時にエラーが発生しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O)setsockopt() 関数が返す**詳細情報**について見直してください。

詳細情報 : make_sock: for address Web **サーバアドレス** : **ポート番号** , apr_socket_opt_set: (SO_REUSEADDR)

ソケットレベルオプション SO_REUSEADDR の設定時にエラーが発生しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O)setsockopt() 関数が返す**詳細情報**について見直してください。

詳細情報 : alloc_listener: failed to set up sockaddr for IP **アドレス**

IP **アドレス**に関するソケット情報を取得できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) Listen ディレクティブまたは BindAddress ディレクティブで指定した IP **アドレス**について見直してください。

詳細情報 : alloc_listener: failed to get a socket for IP **アドレス**

ソケットの終端の作成時にエラーが発生しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O)socket() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **サーバプロセス** ID: Unable to retrieve the graceful exit event from the parent

サーバプロセスの計画停止イベントハンドルを制御プロセスから取得できませんでした。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) ReadFile() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Unable to duplicate the graceful exit event handle for the child

サーバプロセスの計画停止イベントハンドルを複製できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) DuplicateHandle() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Unable to send the graceful exit event handle to the child

サーバプロセスに計画停止イベントハンドルを送信できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: Could not create graceful exit event for child process

サーバプロセスの計画停止イベント生成に失敗しました。

エラーレベル：crit

(S) サーバプロセス生成処理を中断します。

(O) CreateEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Cannot create graceful stop event **イベント名称**

制御プロセスは計画停止のためのイベント生成に失敗しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) CreateEvent() 関数が返す**詳細情報**について見直してください。

[client **クライアントアドレス**] configuration error: couldn't **テキスト** : **リクエスト** URI **値**

リクエスト URI **値**にアクセスする場合に，コンフィグファイル内での設定にエラーがあるため，**テキスト**に示す事象を実行できません。

エラーレベル：crit

(S) ステータスコード「500　Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) テキストに示す内容についてコンフィグファイルを見直してください。

詳細情報 : Fatal error: unable to create global pool for use with by the scoreboard

スコアボードで使用する領域を確保できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Unable to create scoreboard (anonymous shared memory failure)

スコアボードで使用する共有メモリを確保できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

Invalid config file path **コンフィグファイル名**

コンフィグファイルのパスが不正です。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) コンフィグファイル名に示すパスについて見直してください。

詳細情報 : Invalid PID file path **プロセス** ID **格納ファイル名** , ignoring.

プロセス ID 格納ファイル名が不正です。

エラーレベル：crit

(S) Web サーバは起動処理を続行します。

(O) PidFile ディレクティブを**詳細情報**について見直してください。

詳細情報 : make_sock: could not bind to address Web **サーバアドレス** : **ポート番号**

ソケットへのアドレスバインド時にエラーが発生しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O)bind() 関数が返す**詳細情報**について見直してください。

Ouch! Out of memory in add_job()!

リクエスト受付処理中にメモリ確保に失敗しました。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) メモリ使用量について，システムの状態を確認してください。

詳細情報 : Could not open pipe-of-death.

サーバプロセスの終了要求で使用するパイプの生成に失敗しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O)pipe() 関数が返す**詳細情報**について見直してください。

詳細情報 : make_sock: for address Web **サーバアドレス** : **ポート番号** , apr_socket_opt_set: (IPV6_V6ONLY)

ソケットレベルオプション IPV6_V6ONLY の設定時にエラーが発生しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O)setsockopt() 関数が返す**詳細情報**について見直してください。

詳細情報 : An attempt to load the audit log library has failed.

監査ログ出力用ライブラリのロードに失敗しました。

エラーレベル：crit

(S) Web サーバは処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : An attempt to acquire the address of the audit log function has failed.

監査ログ出力用ライブラリの関数のアドレス取得に失敗しました。

エラーレベル：crit

(S) Web サーバは処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : An attempt to acquire the path of the audit log library has failed.

監査ログ出力用ライブラリのパスの取得に失敗しました。

エラーレベル：crit

(S) Web サーバは処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Child **サーバプロセス** ID: Unable to retrieve the parent process handle from the parent

制御プロセスのプロセスハンドルを制御プロセスから取得できませんでした。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) ReadFile() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Unable to duplicate the parent process handle for the child

制御プロセスのプロセスハンドルを複製できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) DuplicateHandle() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: Unable to send the parent process handle to the child

サーバプロセスに制御プロセスのプロセスハンドルを送信できませんでした。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : Parent: SetHandleInformation failed.

制御プロセスの標準入力，標準出力または標準エラー出力のプロパティ設定に失敗しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) SetHandleInformation() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **サーバプロセス** ID: SetHandleInformation failed.

サーバプロセスの標準入力，標準出力または標準エラー出力のプロパティ設定に失敗しました。

エラーレベル：crit

(S) サーバプロセスを終了します。

(O) SetHandleInformation() 関数が返す**詳細情報**について見直してください。

Parent: the child process exited before starting initialization.

サーバプロセスは初期化処理を開始する前に終了しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) このメッセージのほかに出力されているメッセージのエラーの原因について見直してください。

詳細情報 : Parent: WaitForMultipleObjects failed.

制御プロセスはサーバプロセスの初期化処理開始イベント待ちに失敗しました。

エラーレベル：crit

(S) Web サーバは起動処理を中断します。

(O) WaitForMultipleObject() 関数が返す**詳細情報**について見直してください。

---

the service control dispatcher terminated before completing initialization of the service.

サービス制御ディスパッチャスレッドはサービス初期化処理が完了する前に終了しました。

エラーレベル：crit

(S) サービスの起動処理を中断します。

(O) このメッセージのほかに出力されているメッセージのエラーの原因について見直してください。

---

service initialization wait timed out.

サービス起動プロセスのサービス初期化処理完了イベント待ちでタイムアウトが発生しました。

エラーレベル：crit

(S) サービスの起動処理を中断します。

(O) このメッセージのほかに出力されているメッセージのエラーの原因について見直してください。

詳細情報 : WaitForMultipleObjects failed.

サービス起動プロセスはサービス初期化処理完了イベント待ちに失敗しました。

エラーレベル：crit

(S) サービスの起動処理を中断します。

(O) WaitForMultipleObject() 関数が返す**詳細情報**について見直してください。

(4) error レベルのメッセージ

[client **クライアントアドレス**] Request exceeded the limit of **制限値** internal redirects due to probable configuration error. Use 'LimitInternalRecursion' to increase the limit if necessary. Use 'LogLevel debug' to get a backtrace.

リクエストの内部リダイレクトの回数が，**制限値** (10) に達しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 内部リダイレクトが不要に繰り返されないように，設定を確認してください。

[client **クライアントアドレス**] Request exceeded the limit of **制限値** subrequest nesting levels due to probable configuration error. Use 'LimitInternalRecursion' to increase the limit if necessary. Use 'LogLevel debug' to get a backtrace.

リクエストのサブリクエストのネスト回数が**制限値** (10) に達しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) サブリクエストが不要に繰り返されないように，設定を確認してください。

[client **クライアントアドレス**] Invalid URI in request **リクエスト**

リクエスト中の URI が不正です。

エラーレベル：error

(S) ステータスコード「400　Bad Request」をクライアントに返し，リクエスト処理を中断します。

(O) リクエスト中の URI を見直してください。

[client **クライアントアドレス**] File does not exist: **ファイル名**

要求したファイルが見つかりません。

エラーレベル：error

(S) ステータスコード「404　Not Found」をクライアントに返し，リクエスト処理を中断します。

(O) 要求したファイル名について見直してください。

[client **クライアントアドレス**] Attempt to serve directory: **ディレクトリ名**

ディレクトリ名への要求はできません。

エラーレベル：error

(S) ステータスコード「404　Not Found」をクライアントに返し，リクエスト処理を中断します。

(O) リクエスト中の URI を見直してください。

[client **クライアントアドレス**] This resource does not accept the **メソッド** method.

ソースに対して許可されていないメソッドでの要求でした。

エラーレベル：error

(S) ステータスコード「405　Method Not Allowed」をクライアントに返し，リクエスト処理を中断します。

(O) リクエスト中のメソッドを見直してください。

[client **クライアントアドレス**] **詳細情報** : file permissions deny server access: **ファイル名**

要求されたファイルへのアクセス権限がありません。

エラーレベル：error

(S) ステータスコード「403　Forbidden」をクライアントに返し，リクエスト処理を中断します。

(O) 要求されたファイルについて**詳細情報**に従い見直してください。

[client **クライアントアドレス**] Invalid method in request **リクエスト**

リクエスト中のメソッドが不正です。

エラーレベル：error

(S) ステータスコード「501　Method Not Implemented」をクライアントに返し，リクエスト処理を中断します。

(O) リクエスト中のメソッドを見直してください。

[client **クライアントアドレス**] **詳細情報** : core_output_filter: Error reading from bucket.

送信データの読み込み処理に失敗しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報について見直してください。

[client **クライアントアドレス**] **詳細情報** : Failed to read cgi file **ファイル名称** for testing

ファイルの読み込みに失敗しました。

エラーレベル：error

(S) スクリプト実行処理を中断します。

(O) 詳細情報について見直してください。

[client **クライアントアドレス**] **ファイル名称** is not executable; ensure interpreted scripts have "#!" first line

指定されたファイルは実行できません。

エラーレベル：error

(S)CGI プログラムの起動を中断します。

(O) 最初の行に "#!" が指定してあるか確認してください。

[client **クライアントアドレス**] **詳細情報** : reading request body failed

クライアントからのリクエストボディの受信に失敗しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) **詳細情報**に示す原因について見直してください。

詳細情報 : could not create **プロセス** ID **格納ファイル名**

プロセス ID 格納ファイルを作成できませんでした。

エラーレベル：error

(S)Web サーバは起動処理を中断します。

(O)PidFile ディレクティブで指定したファイルを**詳細情報**について見直してください。

---

httpsd: could not log pid to file **プロセス** ID **格納ファイル名**

プロセス ID **格納ファイル**へのプロセス ID の記録に失敗しました。

エラーレベル：error

(S)Web サーバは起動処理を中断します。

(O) このメッセージのほかに出力されているメッセージのエラーの原因について見直してください。

Too many errors in select loop. Child process exiting.

サーバプロセスでの select() エラー発生回数が 100 回を超えたため，サーバプロセスを終了します。

エラーレベル：error

(S) リクエスト受付処理を終了します。

(O) このメッセージのほかに出力されている select() 関数のエラーの原因について見直してください。

詳細情報 : accept: (client socket)

accept() 関数でエラーが発生しました。

エラーレベル：error

(S) リクエスト受付処理を続行します。

(O)accept() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **プロセス** ID: Failed to acquire the start_mutex. Process will exit.

サーバプロセス排他用のロック獲得に失敗しました。

エラーレベル：error

(S) サーバプロセスを終了します。

(O) WaitForSingleObject() 関数が返す**詳細情報**について見直してください。

詳細情報 : Child **プロセス** ID: Failure releasing the start mutex

サーバプロセス排他用のロック解放に失敗しました。

エラーレベル：error

(S) サーバプロセスは終了処理を続行します。

(O) ReleaseMutex() 関数が返す**詳細情報**について見直してください。

詳細情報 : set_listeners_noninheritable: SetHandleInformation failed.

オブジェクトハンドルのプロパティ設定に失敗しました。

エラーレベル：error

(S) サーバプロセスは初期化処理を続行します。

(O) SetHandleInformation() 関数が返す**詳細情報**について見直してください。

詳細情報 : master_main: WaitForMultipleObjects with INFINETE wait exited with WAIT_TIMEOUT

制御プロセスは停止，計画停止または再起動のイベント待ち中に，INFINETE 指定であるにもかかわらずタイムアウトが発生しました。

エラーレベル：error

(S) Web サーバは終了します。

(O) WaitForMultipleObjects() 関数が返す**詳細情報**について見直してください。

詳細情報 : ResetEvent(shutdown_event)

制御プロセスは停止イベントの解除に失敗しました。

エラーレベル：error

(S) Web サーバは停止処理を続行します。

(O) ResetEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: ResetEvent(restart_event) failed.

制御プロセスは再起動イベントの解除に失敗しました。

エラーレベル：error

(S) Web サーバは再起動処理を続行します。

(O) ResetEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: SetEvent for child process **スロット番号** failed

制御プロセスはサーバプロセスの再起動イベントの設定に失敗しました。

エラーレベル：error

(S) Web サーバは再起動処理を続行します。

(O) SetEvent() 関数が返す**詳細情報**について見直してください。

詳細情報 : Parent: SetEvent for child process **スロット番号** failed

制御プロセスはサーバプロセスの停止または計画停止イベントの設定に失敗しました。

エラーレベル：error

(S) Web サーバは停止または計画停止処理を続行します。

(O) SetEvent() 関数が返す**詳細情報**について見直してください。

Parent: child process exited with status **終了コード** -- Aborting.

サーバプロセスは**終了コード**で終了しました。Web サーバを停止します。

エラーレベル：error

(S) Web サーバは停止処理を開始します。

(O) 終了コードが示す原因について，見直してください。

サービス名称 :Service is already installed.

すでにインストールされているサービスをインストールしようとしました。

エラーレベル：error

(S) Web サーバはサービスインストール処理を中止します。

(O) サービスを確認してください。

詳細情報 : No installed service named " **サービス名称** ".

インストールされていないサービスを指定しました。

エラーレベル：error

(S) Web サーバは起動処理を中断します。

(O) サービスを確認してください。

詳細情報 : **サービス名称** : Unable to create the start_mutex.

排他用のハンドラ生成に失敗しました。

エラーレベル：error

(S) Web サーバは起動処理を中断します。

(O) CreateMutexW() 関数が返す**詳細情報**について見直してください。

詳細情報 : could not open transfer log file **ファイル名** .

ログファイルが開けません。

エラーレベル：error

(S) Web サーバの起動処理を終了します（Web サーバを起動しません）。

(O) TransferLog ディレクティブまたは CustomLog ディレクティブの設定を見直してください。

詳細情報 : invalid transfer log path **ファイル名** .

ログファイルが不正です。

エラーレベル：error

(S) Web サーバの起動処理を終了します（Web サーバを起動しません）。

(O) TransferLog ディレクティブまたは CustomLog ディレクティブの設定を見直してください。

[client **クライアントアドレス**] **詳細情報：** log writer isn't correctly setup

log writer が正しくセットアップされていません。

エラーレベル：error

(S) サーバは処理を続行します。

(O) **詳細情報**について見直してください。

詳細情報 : ResetEvent(graceful_stop_event)

制御プロセスは計画停止イベントの解除に失敗しました。

エラーレベル：error

(S)Web サーバは計画停止処理を続行します。

(O)ResetEvent() 関数が返す**詳細情報**について見直してください。

[client **クライアントアドレス**] request failed: URI too long (longer than **上限値**)

リクエスト URI が長過ぎるため，リクエスト処理を続行できません。

エラーレベル：error

(S) ステータスコード「414　Request-URI Too Large」をクライアントに返し，リクエスト処理を中断します。

(O) リクエストライン（メソッド，問い合わせ文字列などを含む URI，HTTP バージョン）で指定した文字列が LimitRequestLine ディレクティブ値を超えていないか見直してください。

[client **クライアントアドレス**] request failed: erroneous characters after protocol string: **リクエストライン**

リクエストラインに不正な HTTP プロトコル文字列が指定されています。

エラーレベル：error

(S) ステータスコード「400 Bad Request」をクライアントに返し，リクエスト処理を中断します。

(O) リクエストラインに文法エラーがあります。リクエストラインに指定された HTTP バージョンに，不正な文字列が指定されていないかどうかを確認してください。

[client **クライアントアドレス**] request failed: error reading the headers

リクエストヘッダにエラーがあるためリクエスト処理を続行できません。

エラーレベル：error

(S) ステータスコード「400　Bad Request」をクライアントに返し，リクエスト処理を中断します。

(O) リクエストヘッダに文法エラーがあります。ヘッダとして誤ったものを指定してい

ないかまたはリクエストヘッダの個数が LimitRequestFields ディレクティブ値を超えていないかを見直してください。

[client **クライアントアドレス**] client sent invalid HTTP/0.9 request: HEAD **リクエスト** URI **値**

クライアントは不当な HTTP/0.9 リクエスト（HEAD リクエスト URI 値）を送付したため，リクエスト処理を続行できません。

エラーレベル：error

(S) ステータスコード「400　Bad Request」をクライアントに返し，リクエスト処理を中断します。

(O)HEAD メソッドでのリクエストは HTTP プロトコル 1.0 以降で有効です。HTTP プロトコル文字列を含まないで，HEAD メソッドによるリクエストをしていないか確認してください。

[client **クライアントアドレス**] client sent HTTP/1.1 request without hostname (see RFC2616 section 14.23): **リクエスト** URI **値**

クライアントは HTTP/1.1 リクエストを HOST ヘッダを付けないで送付したため，リクエスト処理を続行できません。

エラーレベル：error

(S) ステータスコード「400　Bad Request」をクライアントに返し，リクエスト処理を中断します。

(O)HOST ヘッダを付けた HTTP/1.1 リクエストであるかを確認してください。

[client **クライアントアドレス**] need AuthType to note auth failure: **リクエスト** URI **値**

ユーザ認証する場合の AuthType ディレクティブで指定する認証タイプ名が指定されていません。

エラーレベル：error

(S) ステータスコード「401　Authorization Required」を返してリクエスト処理を終了します。

(O) ユーザ認証する場合には，AuthType ディレクティブを指定してください。

[client **クライアントアドレス**] need AuthName: **リクエスト** URI **値**

ユーザ認証する場合の AuthName ディレクティブで指定する realm 名が指定されていません。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) ユーザ認証する場合には，AuthName ディレクティブを指定してください。

[client **クライアントアドレス**] client used wrong authentication scheme: **リクエスト** URI **値**

ユーザ認証する場合の Authorization ヘッダの認証制御のタイプが不正です。

エラーレベル：error

(S) ステータスコード「401　Authorization Required」を返してリクエスト処理を終了します。

(O) Authorization ヘッダには Basic だけを指定できます。Authorization ヘッダでの指定値を見直してください。

[client **クライアントアドレス**] **詳細情報** : ap_content_length_filter: apr_bucket_read() failed

レスポンスデータの読み込み処理に失敗しました。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報について見直してください。

[client **クライアントアドレス**] **詳細情報** : dir_walk error, path_info **パス情報** is not relative to the filename path **ディレクトリ名** for uri **リクエスト** URI **値**

パス情報に示すパスが**ディレクトリ名**に示すディレクトリ以下に含まれていません。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報について見直してください。

[client **クライアントアドレス**] **詳細情報** : dir_walk error, could not determine the root path of filename **ファイル名** for uri **リクエスト** URI **値**

ファイル名に示すパスに対して，基準となるパスを決定できませんでした。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) **詳細情報**について見直してください。

[client **クライアントアドレス**] **詳細情報** : access to **リクエスト** URI **値** denied

リクエスト URI に示すファイルへのアクセスは，ファイルへの検索パーミッションがないために失敗しました。

エラーレベル：error

(S) ステータスコード「403 Forbidden」をクライアントに返し，リクエスト処理を中断します。

(O) アクセスファイルについて**詳細情報**に従い見直してください。

[client **クライアントアドレス**] **詳細情報** : access to **リクエスト** URI **値** failed

リクエスト URI に示すファイルへアクセスできません。

エラーレベル：error

(S) ステータスコード「403 Forbidden」をクライアントに返し，リクエスト処理を中断します。

(O) アクセスファイルについて**詳細情報**に従い見直してください。

[client **クライアントアドレス**] Forbidden: **アクセスファイル** doesn't point to a file or directory

アクセスファイルはファイルおよびディレクトリではありません。

エラーレベル：error

(S) ステータスコード「403 Forbidden」をクライアントに返し，リクエスト処理を中断します。

(O) アクセスファイルの属性を見直してください。

Access to file **ファイル名** denied by server: not a regular file

ファイルが標準ファイルではないため，ファイルへのアクセスは，Web サーバによって拒否されました。

エラーレベル：error

(S)Web サーバの起動処理を中断します。

(O) アクセスするファイルが標準ファイルかどうか見直してください。

[client **クライアントアドレス**] Premature end of script headers: CGI **プログラム**

CGI **プログラム**からクライアントへの出力データがありません。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」をクライアントに返し，リクエスト処理を中断します。CGI プログラムは実行を終了します。

(O)CGI プログラムについて見直してください。CGI プログラムは，最初に HTTP ヘッダを出力する必要があります。

[client **クライアントアドレス**] malformed header from script. Bad header= **ヘッダ** : CGI **プログラム**

CGI **プログラム**からクライアントへ出力された HTTP ヘッダのフォーマットは不正です。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」をクライアントに返し，リクエスト処理を中断します。CGI プログラムは実行を終了します。

(O)CGI プログラムについて見直してください。CGI プログラムは，最初に HTTP ヘッダを出力する必要があります。出力された HTTP ヘッダのフォーマットが不正です。

詳細情報 : Cannot resolve host name **バーチャルホスト名** --- ignoring!

不正なバーチャルホストを指定しているため，指定を無視します。

エラーレベル：error

(S) 不正なバーチャルホストは無視して，Web サーバの起動処理を続行します。

(O) NameVirtualHost ディレクティブまたは VirtualHost ディレクティブで指定したホスト名について見直してください。

VirtualHost **ホスト名** : **ポート番号** -- mixing * ports and non-* ports with a NameVirtualHost address is not supported, proceeding with undefined results

VirtualHost ディレクティブと NameVirtualHost ディレクティブに *（アスタリスク）で指定したポート番号と * ではないポート番号が混在しています。実行結果は保証できません。

エラーレベル：error

(S) Web サーバは起動処理を続行します。

(O)VirtualHost ディレクティブと NameVirtualHost ディレクティブで指定するポート番号について，*(アスタリスク) の混在について見直してください。VirtualHost ディレクティブで指定するポート番号で * を指定した場合には，NameVirtualHost ディレクティブについてもポート番号には * を指定してください。

詳細情報 : Failed to resolve server name for IP **アドレス** (check DNS) -- or specify an explicit ServerName

VirtualHost ディレクティブで指定したホスト名の解決処理に失敗しました。

エラーレベル：error

(S)Web サーバはホスト名の解決処理に失敗したバーチャルホストについては無視して，起動処理を続行します。解決処理に失敗したバーチャルホストは，バーチャルホストとして認識されません。

(O)VirtualHost ディレクティブで指定したホスト名について見直してください。

[client **クライアントアドレス**] Client sent malformed Host header

クライアントは不正な Host ヘッダを送付しました。

エラーレベル：error

(S) ステータスコード「400　Bad Request」をクライアントに返し，リクエスト処理を中断します。

(O)Host ヘッダを見直してください。

[client **クライアントアドレス**] Invalid Content-Length

Content-Length ヘッダで指定した長さが不正です。

エラーレベル：error

(S) ステータスコード「400　Bad Request」をクライアントに返し，リクエスト処理を中断します。

(O)Content-Length ヘッダには数値（10 進数）だけが指定できます。指定値を見直してください。

[client **クライアントアドレス**] Requested content-length of **指定長** is larger than the configured limit of LimitRequestBody **ディレクティブ値**

Content-Length ヘッダでの指定長は，LimitRequestBody ディレクティブ値よりも大きい値を指定しています。

エラーレベル：error

(S) ステータスコード「413　Request Entity Too Large」をクライアントに返し，リクエスト処理を中断します。

(O) LimitRequestBody ディレクティブでの指定値を見直してください。リクエストボディサイズに上限値を設定しない場合には，0 を指定してください。

[client **クライアントアドレス**] Read content-length of **読み込みボディサイズ** is larger than the configured limit of LimitRequestBody **ディレクティブ値**

LimitRequestBody ディレクティブ値よりも大きいサイズのボディを読み込みました。

エラーレベル：error

(S) ステータスコード「413　Request Entity Too Large」をクライアントに返し，リクエスト処理を中断します。

(O) LimitRequestBody ディレクティブでの指定値を見直してください。リクエストボディサイズに上限値を設定しない場合には，0 を指定してください。

[client **クライアントアドレス**] Unknown Transfer-Encoding Transfer-Encoding **ヘッダ値**

Transfer-Encoding ヘッダでの指定値が不正です。

エラーレベル：error

(S) ステータスコード「501　Method Not Implemented」をクライアントに返し，リクエスト処理を中断します。

(O) Transfer-Encoding ヘッダに chunked 以外を指定しています。Transfer-Encoding ヘッダを見直してください。

[client **クライアントアドレス**] chunked Transfer-Encoding forbidden: **リクエスト** URI **値**

Transfer-Encoding ヘッダに chunked を指定することは禁止されています。

エラーレベル：error

(S) ステータスコード「411　Length Required」または「400　Bad Request」をクライアントに返し，リクエスト処理を中断します。

(O) Transfer-Encoding ヘッダに chunked 以外を指定しています。Transfer-Encoding ヘッダを見直してください。

[client **クライアントアドレス**] **メソッド** with body is not allowed for **リクエスト** URI **値**

ボディを伴うことを許可されていないメソッドに対して，ボディを伴って指定されています。

エラーレベル：error

(S) ステータスコード「413　Request Entity Too Large」をクライアントに返し，リクエスト処理を中断します。

(O) リクエストボディが付加された TRACE メソッドを許可する場合は，TraceEnable ディレクティブで extended を指定してください。ただし，リクエストボディサイズが 64KB を超える場合は受け付けられません。

[client **クライアントアドレス**] **詳細情報** : apr_brigade_partition() failed [**開始オフセット値または終了オフセット値** +1, **ファイルサイズ**]

バイトレンジオペレーション処理時にエラーが発生しました。

エラーレベル：error

(S) Web サーバはエラーの発生したバイトレンジオペレーションについてのデータはクライアントには送信しないで，次のバイトレンジオペレーション処理を実行します。

(O) 詳細情報に示す原因について見直してください。

[client **クライアントアドレス**] Invalid error redirection directive: **リダイレクト先**

ErrorDocument ディレクティブで指定するカスタマイズ方法に文法的にエラーがあります。

エラーレベル：error

(S) エラーステータス番号はカスタマイズしません。

(O) ErrorDocument ディレクティブでエラーメッセージをカスタマイズする場合には，テキスト，ローカル URL またはフル URL を指定してください。詳細は，ErrorDocument ディレクティブの説明を参照してください。

Unable to open logs

ログ出力先の設定またはソケット生成処理でエラーが発生しました。

エラーレベル：error

(S) Web サーバは起動処理を中断します。

(O) このメッセージのほかに出力されているメッセージのエラーの原因について見直してください。

Configuration Failed

コンフィグファイル読み込み後処理でエラーが発生しました。

エラーレベル：error

(S) Web サーバは起動処理を中断します。

(O) このメッセージのほかに出力されているメッセージのエラーの原因について見直してください。

詳細情報 : Failure registering service handler

サービス制御マネジャへのサービス制御要求処理関数の登録に失敗しました。

エラーレベル：error

(S) サービスの起動処理を中断します。

(O) RegisterServiceCtrlHandler() 関数が返す**詳細情報**について見直してください。

詳細情報 : Error starting service control dispatcher

サービス制御マネジャへのサービス制御ディスパッチャスレッドの設定に失敗しました。

エラーレベル：error

(S) サービスの起動処理を中断します。

(O) StartServiceCtrlDispatcher() 関数が返す**詳細情報**について見直してください。

詳細情報 : GetModuleFileName failed

Web サーバ実行ファイルのパス取得に失敗しました。

エラーレベル：error

(S) サービスの登録処理を中断します。

(O) GetModuleFileName() 関数が返す**詳細情報**について見直してください。

詳細情報 : Failed to open the WinNT service manager

ローカルコンピュータのサービス制御マネジャとの接続に失敗しました。

エラーレベル：error

(S) サービスのインストール，削除または起動などの Web サーバ操作処理を中断します。

(O) OpenSCManager() 関数が返す**詳細情報**について見直してください。

詳細情報 : Failed to open the WinNT service manager.

ローカルコンピュータのサービス制御マネジャとの接続に失敗しました。

エラーレベル：error

(S) サービスの削除処理を中断します。

(O) OpenSCManager() 関数が返す**詳細情報**について見直してください。

詳細情報 : OpenService failed

サービスのハンドルの取得に失敗しました。

エラーレベル：error

(S) サービスの構成パラメタ変更処理を中断します。

(O) OpenService() 関数が返す**詳細情報**について見直してください。

詳細情報 : ChangeServiceConfig failed

サービスの構成パラメタ変更に失敗しました。

エラーレベル：error

(S) サービスの構成パラメタ変更処理を中断します。

(O) ChangeServiceConfig() 関数が返す**詳細情報**について見直してください。

詳細情報 : Failed to create WinNT Service Profile

サービスの登録に失敗しました。

エラーレベル：error

(S) サービスの登録処理を中断します。

(O) CreateService() 関数が返す**詳細情報**について見直してください。

詳細情報 : **サービス名称** : Failed to store the ConfigArgs in the registry.

レジストリエントリ ConfigArgs の格納に失敗しました。

エラーレベル：error

(S) サービスのインストール，構成パラメタ変更処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : **サービス名称** : OpenService failed

サービスのハンドルの取得に失敗しました。

エラーレベル：error

(S) サービスの削除処理を中断します。

(O) OpenService() 関数が返す**詳細情報**について見直してください。

詳細情報 : **サービス名称** : Failed to delete the service.

サービスの削除に失敗しました。

エラーレベル：error

(S) サービスの削除処理を中断します。

(O) DeleteService() 関数が返す**詳細情報**について見直してください。

詳細情報 : **サービス名称** : Failed to open the service.

サービスのハンドルの取得に失敗しました。

エラーレベル：error

(S) サービスの起動処理を中断します。

(O) OpenService() 関数が返す**詳細情報**について見直してください。

Service **サービス名称** is already started!

すでに起動しているサービスを起動しようとしました。

エラーレベル：error

(S) サービスの起動処理を中断します。

(O) サービス名称に示すサービスがすでに起動していないかを確認してください。

詳細情報 : Failed to open the NT Service Manager

ローカルコンピュータのサービス制御マネジャとの接続に失敗しました。

エラーレベル：error

(S) サービスの停止，計画停止または再起動処理を中断します。

(O) OpenSCManager() 関数が返す**詳細情報**について見直してください。

詳細情報 : Failed to open the **サービス名称** Service

サービスのハンドルの取得に失敗しました。

エラーレベル：error

(S) サービスの停止，計画停止または再起動処理を中断します。

(O) OpenService() 関数が返す**詳細情報**について見直してください。

詳細情報 : Query of Service **サービス名称** failed

サービスのステータス情報の取得に失敗しました。

エラーレベル：error

(S) サービスの停止，計画停止または再起動処理を中断します。

(O) QueryServiceStatus() 関数が返す**詳細情報**について見直してください。

詳細情報 : make_sock: unable to listen for connections on address Web **サーバアドレス** : **ポート番号**

ソケット上での接続の受け入れ準備時にエラーが発生しました。

エラーレベル：error

(S) Web サーバは起動処理を中断します。

(O)listen() 関数が返す**詳細情報**について見直してください。

詳細情報 : The HWS trace could not open the log file **ファイル名** specified in the HWSTraceLogFile.

HWSTraceLogFile ディレクティブで指定した**ファイル名**に示すファイルを開けません。

エラーレベル：error

(S) 処理を中断します。

(O) 詳細情報に示す原因について見直してください。

詳細情報 : The HWS trace could not create the segment(size= **サイズ**).

共有メモリを確保できません。

エラーレベル：error

(S) 処理を中断します。

(O) 詳細情報に示す原因について見直してください。

詳細情報 : The HWS trace could not map the segment(id= **識別子**) for the child.

共有メモリを空間に割り当てられません。

エラーレベル：error

(S) 処理を中断します。

(O) 詳細情報に示す原因について見直してください。

詳細情報 : The HWS trace could not map the segment(id= **識別子**) for the parent.

共有メモリを空間に割り当てられません。

エラーレベル：error

(S) 処理を中断します。

(O) 詳細情報に示す原因について見直してください。

詳細情報 : The HWS trace could not open the ID file **ファイル名** specified in the HWSTraceIdFile.

HWSTraceIdFile ディレクティブで指定した**ファイル名**で示すファイルを開けません。

エラーレベル：error

(S) 処理を中断します。

(O) 詳細情報に示す原因について見直してください。

詳細情報 : The HWS trace could not write the log file **ファイル名** specified in the HWSTraceLogFile.

HWSTraceLogFile ディレクティブで指定したファイルへ，プロセス ID についての共有メモリの内容を出力することに失敗しました。

エラーレベル：error

(S) 出力処理を中断します。

詳細情報 : The HWS trace could not open the segment(id= **共有メモリ識別子**).

共有メモリのオープンに失敗しました。

エラーレベル：error

(S) 処理を中断します。

(O) 詳細情報に示す原因について見直してください。

[client **クライアントアドレス**] File does not exist: **ファイル名**

要求したファイルが見つかりません。

エラーレベル：error

(S) ステータスコード「404　Not Found」をクライアントに返し，リクエスト処理を中断します。

(O) 要求したファイル名について見直してください。

[client **クライアントアドレス**] **詳細情報** : Can't open directory for index: **ディレクトリ名**

ディレクトリインデクスをするディレクトリが開けません。

エラーレベル：error

(S) ステータスコード「403　Forbidden」を返してリクエスト処理を終了します。

(O) 詳細情報に示す原因について，システムの状態を確認してください。

[client **クライアントアドレス**] Directory index forbidden by rule: **ファイル名**

設定によって，ディレクトリインデクスを表示できません。

エラーレベル：error

(S) ステータスコード「403　Forbidden」を返してリクエスト処理を終了します。

(O) ディレクトリインデクスを許可する場合には，設定を変更してください。

[client **クライアントアドレス**] cannot redirect ' **リクエスト** URI' to ' **リダイレクト先** '; target is not a valid absoluteURI or abs_path

リダイレクト先にリダイレクトできません。設定されたリダイレクト先は，URL ではありません。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 設定されたリダイレクト先が URL となっていることを確認してください。

[client **クライアントアドレス**] **詳細情報** : couldn't create child process: **エラーコード ファイル名**

CGI 実行時にエラーが発生し，プログラム用のプロセスを生成できませんでした。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」を返してリクエスト処理を終了します。

(O) **詳細情報**に示す原因について，システムの状態を確認してください。

[client **クライアントアドレス**] **詳細情報** : don't know how to spawn child process: **ファイル名**

プロセスを生成できません。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」を返してリクエスト処理を終了します。

(O) **詳細情報**に示す原因について，システムの状態を確認してください。

[client **クライアントアドレス**] **詳細情報** : couldn't spawn child process: **ファイル名**

CGI プログラム用のプロセスを生成できませんでした。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」を返してリクエスト処理を終了します。

(O) **詳細情報**に示す原因について，システムの状態を確認してください。

[client **クライアントアドレス**] invalid base directive in map file: **リクエスト** URI **値**

イメージマップファイル中の base ディレクティブ指定が無効です。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」を返してリクエスト処理を終了します。

(O) マップファイルの base 設定を見直してください。

[client **クライアントアドレス**] invalid directory name in map file: **リクエスト** URI **値**

イメージマップファイル中で指定されているディレクトリ名が無効です。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」を返してリクエスト処理を終了します。

(O) イメージマップファイル内の指定値を見直してください。

[client **クライアントアドレス**] map file **リクエスト** URI **値** , line **行番号** syntax error: requires at least two fields

イメージマップファイルの**行番号**で示す行で構文エラーがあります。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」を返してリクエスト処理を終了します。

(O) エラーとなっている行の設定に対して，一つ以上の指定を追加してください。

[client **クライアントアドレス**] **詳細情報** : cannot read directory for multi: **ディレクトリ名**

コンテントネゴシエーションをしようとするディレクトリを開けません。

エラーレベル：error

(S) ステータスコード「403 Forbidden」を返してリクエスト処理を終了します。

(O) 詳細情報に示す原因について，システムの状態を確認してください。

[client **クライアントアドレス**] no acceptable variant: **ファイル名**

Web サーバによるコンテントネゴシエーションの結果，クライアントが受理できるタイプがありませんでした。

エラーレベル：error

(S) ステータスコード「406 Not Acceptable」を返してリクエスト処理を終了します。

[client **クライアントアドレス**] Negotiation: discovered file(s) matching request: **ファイル名** (None could be negotiated).

ファイル名にマッチするファイルの中に，コンテントネゴシエーション可能なものがありませんでした。

エラーレベル：error

(S) ステータスコード「404 Not Found」を返してリクエスト処理を終了します。

(O) コンテントネゴシエーションさせたいファイルのアクセス権限を見直してください。

[client **クライアントアドレス**] client denied by server configuration: **ファイル名**

ファイル名に示すファイルへのアクセスが Allow または Deny ディレクティブによって拒否されました。

エラーレベル：error

(S) ステータスコード「403　Forbidden」を返してリクエスト処理を終了します。

(O) アクセスを許可する場合には，設定を変更してください。

[client **クライアントアドレス**] **詳細情報** : Could not open password file: **ファイル名**

AuthUserFile ディレクティブで指定された**ファイル名**で示すパスワードファイルが開けません。

エラーレベル：error

(S)

- AuthAuthoritative ディレクティブの値が off の場合
  クライアントから送信されてきたユーザ名はパスワードファイルに未登録であったと解釈して，処理を続行します。
- そのほかの場合
  ステータスコード「401　Authorization Required」を返してリクエスト処理を終了します。

(O)AuthUserFile ディレクティブ指定値を見直してください。

[client **クライアントアドレス**] user **ユーザ名** not found: **リクエスト** URI **値**

リクエスト URI にアクセスしようとしたクライアントが送信してきたユーザ名が，アクセスできるユーザ名一覧の中に見つかりませんでした。

エラーレベル：error

(S) ステータスコード「401　Authorization Required」を返してリクエスト処理を終了します。

(O) ユーザのアクセスを許可する場合には，設定に追加してください。

[client **クライアントアドレス**] user **ユーザ名** : authentication failure for " **リクエスト** URI **値** ": Password Mismatch

リクエスト URI にアクセスしようとしたクライアントは認証エラーとなりました。

エラーレベル：error

(S) ステータスコード「401　Authorization Required」を返してリクエスト処理を終了します。

(O) パスワードを見直してください。

[client **クライアントアドレス**] access to **リクエスト** URI **値** failed, reason: unknown require directive: " **識別子** "

Require ディレクティブに対して不明な識別子が指定されていたため，正常に認証されませんでした。

エラーレベル：error

(S) ステータスコード「401　Authorization Required」を返してリクエスト処理を終了します。

(O) Require ディレクティブの指定を見直してください。

[client **クライアントアドレス**] access to **リクエスト** URI **値** failed, reason: user **ユーザ名** not allowed access

認証に失敗したため，ユーザはリクエスト URI にアクセスできません。

エラーレベル：error

(S) ステータスコード「401　Authorization Required」を返してリクエスト処理を終了します。

(O) ユーザのアクセスを許可する場合には，アクセス制御設定を変更してください。

[client **クライアントアドレス**] **詳細情報** : Could not open group file: **ファイル名**

AuthGroupFile ディレクティブで指定されたファイルを開けません。

エラーレベル：error

(S) リクエスト処理を続行します。

(O) AuthGroupFile ディレクティブの設定を見直してください。

詳細情報 : Invalid mime types config path **ファイル名**

TypesConfig ディレクティブで指定されたファイルのパスが不正です。

エラーレベル：error

(S) Web サーバの起動処理を終了します（Web サーバを起動しません）。

(O)TypesConfig ディレクティブの設定を見直してください。

詳細情報 : could not open mime types config file **ファイル名** .

TypesConfig ディレクティブで指定されたファイルが開けません。

エラーレベル：error

(S)Web サーバの起動処理を終了します（Web サーバを起動しません）。

(O)TypesConfig ディレクティブの設定を見直してください。

Pre-configuration failed

コンフィグファイル読み込み前処理でエラーが発生しました。

エラーレベル：error

(S)Web サーバは起動処理を中断します。

(O) このメッセージのほかに出力されているメッセージのエラーの原因について見直してください。

詳細情報 : unable to setup module trace: **関数名**

モジュールトレースを出力するための初期設定ができませんでした。

エラーレベル：error

(S)Web サーバは起動処理を中断します。

(O)_open_osfhandle() 関数または _fdopen 関数が返す詳細情報について見直してください。

詳細情報 : apr_poll: (listen)

select() 関数でエラーが発生しました。

エラーレベル：error

(S) サーバプロセスを終了します。

(O)select() 関数が返す詳細情報について見直してください。

---

server reached MaxClients setting, consider raising the MaxClients setting

サーバプロセス数は MaxClients ディレクティブ設定値に到達しました。MaxClients ディレクティブを増加させることを検討してください。

エラーレベル：error

(S) Web サーバの起動を続行します。MaxClients ディレクティブ指定値を超えてはサーバプロセスの生成はされません。

(O) MaxClients ディレクティブ指定値を見直してください。MinSpareServers ディレクティブで指定した数のアイドル状態のサーバプロセスを生成するためには，MaxClients ディレクティブでの指定値を増加させる必要があります。

---

child process **プロセス** ID still did not exit, sending a SIGKILL

プロセス ID で示すサーバプロセスが終了していないため，SIGKILL シグナルを送信します。

エラーレベル：error

(S) Web サーバの停止または再起動処理を続行します。

(O) サーバプロセスが終了するまでお待ちください。

---

could not make child process **プロセス** ID exit, attempting to continue anyway

プロセス ID で示すサーバプロセスを終了させることができませんでした。

エラーレベル：error

(S) 未終了のサーバプロセスを無視してサーバの停止または再起動処理を続行します。

(O) 未終了のサーバプロセスがあります。終了しない原因を調査し，終了させる必要があればシグナルを送付し強制的に終了させてください。

詳細情報 : apr_accept: (client socket)

accept() 関数でエラーが発生しました。

エラーレベル：error

(S) リクエストを受けたサーバプロセスを終了します。

(O)accept() 関数が返す**詳細情報**について見直してください。

詳細情報 : The HWS trace could not assign the segment(id= **共有メモリ識別子**).

共有メモリの割り当てができません。

エラーレベル：error

(S) 処理を中断します。

(O)shmat() 関数が返す**詳細情報**について見直してください。

詳細情報 : HWS trace shmctl() IPC_STAT could not educe the current value(id= **共有メモリ識別子**).

共有メモリ識別子に対する共有メモリの状態を抽出できません。

エラーレベル：error

(S) 処理を中断します。

(O)shmctl 関数が返す**詳細情報**について見直してください。

詳細情報 : HWS trace shmctl() IPC_SET could not set the value(id= **共有メモリ識別子**), you probably need to modify User or Group directives.

共有メモリ識別子に対する共有メモリの状態を設定できません。

エラーレベル：error

(S) 処理を中断します。

(O)User ディレクティブまたは Group ディレクティブで指定したユーザが，システムに登録されていない場合に発生します。User ディレクティブ，Group ディレクティブの指定値を見直してください。

[client **クライアントアドレス**] Symbolic link not allowed: **アクセスファイル**

アクセスファイルで示すファイルはシンボリックリンクをたどることを許可されていません。

エラーレベル：error

(S) ステータスコード「403　Forbidden」をクライアントに返し，リクエスト処理を中断します。

(O) Options ディレクティブの FollowSymLinks が有効であることを確認ください。FollowSymLinks が無効の場合には，SymLinksIfOwnerMatch が有効であるか確認してください。SymLinksIfOwnerMatch が有効である場合には，**アクセスファイル**またはディレクトリの所有者がシンボリックリンクの所有者と同じことを確認してください。

詳細情報 : socket: SetHandleInformation failed.(client **クライアントアドレス**)

ソケットのプロパティ設定に失敗しました。

エラーレベル：error

(S)Web サーバはリクエスト処理を中断します。

(O)SetHandleInformation() 関数が返す**詳細情報**について見直してください。

詳細情報 : **関数名** failed

関数名で示す関数でエラーが発生しました。

エラーレベル：error

(S)Web サーバは起動処理を中断します。

(O) 関数が返す**詳細情報**について見直してください。

---

request queue reached HWSMaxQueueSize setting, consider raising the HWSMaxQueueSize setting

リクエストキューに保持されているリクエストの数が，HWSMaxQueueSize ディレクティブの設定値に到達しました。HWSMaxQueueSize ディレクティブの設定値を増加させることを検討してください。

エラーレベル：error

(S) 処理を続行します。キューサイズを超えたクライアントからのリクエストの接続は Web サーバ側で切断されます。

(O)HWSMaxQueueSize ディレクティブの設定値を見直してください。

(5) warn レベルのメッセージ

---

module **モジュール名称** is already loaded, skipping

モジュールはすでに Web サーバに組み込まれているため，処理を行いません。

エラーレベル：warn

(S)Web サーバは起動処理を続行します。

(O)LoadModule ディレクティブの記述内容を見直してください。

---

pid file **プロセス** ID **格納ファイル名** overwritten -- Unclean shutdown of previous server run?

プロセス ID を格納しているファイルを上書きしました。前回 Web サーバが正常にシャットダウンされなかったおそれがあります。Web サーバが正常にシャットダウンされた場合は，プロセス ID 格納ファイルは消去されます。

エラーレベル：warn

(S)Web サーバは起動処理を続行します。

(O) 未終了の Web サーバがないかどうか確認してください。PidFile ディレクティブで指定したファイルには，Web サーバの起動時に制御プロセスのプロセス ID が格納されます。複数の Web サーバで PidFile ディレクティブで指定したファイルが重複した場合にも，このメッセージは出力されます。複数の Web サーバで共用していないか確認してください。

詳細情報 : apr_socket_opt_set: (TCP_NODELAY)

IP プロトコルレベルのオプション TCP_NODELAY 設定時にエラーが発生しました。

エラーレベル：warn

(S)Web サーバは TCP_NODELAY オプションは設定しないで，起動処理を続行します。Nagle バッファリングアルゴリズムは有効です。

(O)setsockopt() 関数が返す**詳細情報**について見直してください。

詳細情報 : getsockname failed

getsockname() に失敗しました。

エラーレベル：warn

(S) サーバスレッドはリクエスト待ち状態となります。

(O)getsockname() 関数が返す**詳細情報**について見直してください。

詳細情報 : getpeername failed

getpeername() に失敗しました。

エラーレベル：warn

(S) サーバスレッドはクライアントのソケット情報を設定しないで，リクエスト処理を続行します。

(O)getpeername() 関数が返す**詳細情報**について見直してください。

Server ran out of threads to serve requests. Consider raising the ThreadsPerChild setting

リクエストを処理するサーバスレッドが不足しています。

エラーレベル：warn

(S) リクエスト処理を続行します。

(O)ThreadsPerChild ディレクティブ値の増加を検討してください。

詳細情報 : No installed ConfigArgs for the service " **サービス名称** ", using Server defaults.

サービス名称で示すサービスのレジストリエントリ ConfigArgs の内容を取得できないため，デフォルトで起動します。

エラーレベル：warn

(S)Web サーバは起動処理を続行します。

(O) **サービス名称**のレジストリキー下に，レジストリエントリ ConfigArgs があるかを確認してください。

NameVirtualHost **ホスト名** : **ポート番号** has no VirtualHosts

NameVirtualHost ディレクティブが定義されたホスト名，ポート番号は VirtualHost ディレクティブで使用されていません。

エラーレベル：warn

(S)NameVirtualHost ディレクティブの指定を無視し，Web サーバ起動処理を続行します。

(O) 不要な NameVirtualHost ディレクティブは削除してください。NameVirtualHost ディレクティブは，同一のホスト名，ポート番号に対して複数の VirtualHost ディレクティブを定義する場合に指定してください。

---

default VirtualHost overlap on port **ポート番号** , the first has precedence

デフォルトバーチャルホストはポート番号についてオーバラップしました。最初に指定したバーチャルホストが有効になります。

エラーレベル：warn

(S) 最初に指定したバーチャルホスト以外は無視し，Web サーバ起動処理を続行します。

(O) デフォルトバーチャルホスト (VirtualHost ディレクティブで *，255.255.255.255 または _default_ を指定する) は，同じポートについて複数指定できません。デフォルトバーチャルホストの指定について見直してください。

VirtualHost **ホスト名** : **ポート番号** overlaps with VirtualHost **ホスト名** : **ポート番号** , the first has precedence, perhaps you need a NameVirtualHost directive

ホスト名：ポート番号で指定されたバーチャルホストはオーバラップしました。最初に指定したバーチャルホストが有効になります。

エラーレベル：warn

(S) 最初に指定したバーチャルホスト以外は無視し，Web サーバ起動処理を続行します。

(O) ホスト名：ポート番号に同じバーチャルホストを指定する場合には，該当するホスト名，ポート番号について NameVirtualHost ディレクティブを定義してください。

---

httpsd: gethostname() failed to determine ServerName

gethostname() によって，ServerName を決定しようとしましたが失敗しました。

エラーレベル：warn

(S) Web サーバのアドレスは 127.0.0.1 とします。

(O) ホスト名が取得できない原因について見直してください。

PassEnv variable **環境変数** was undefined

PassEnv ディレクティブで指定した**環境変数**は環境変数として定義されていません。

エラーレベル：warn

(S) 処理を続行します。

(O) 環境変数を設定してください。

The **ディレクティブ** directive in **コンフィグファイル** at line **行番号** will probably never match because it overlaps an earlier **ディレクティブ** .

同一の URL に適用可能なディレクティブを重複して指定しているため，後の指定は適用されません。

エラーレベル：warn

(S) Web サーバは起動処理を続行します。

(O) 表示されている行のディレクティブの設定を見直してください。

詳細情報 : Cannot get media type from ' **メディアタイプ** '

MIME のメディアタイプが正しく取得できません。

エラーレベル：warn

(S) リクエスト処理を続行します。

(O)MIME の設定を見直してください。

詳細情報 : Cannot get media parameter.

MIME のメディアパラメータが正しく取得できません。

エラーレベル：warn

(S) リクエスト処理を続行します。

(O)MIME の設定を見直してください。

詳細情報 : Cannot get media subtype.

MIME のメディアサブタイプが正しく取得できません。

エラーレベル：warn

(S) リクエスト処理を続行します。

(O)MIME の設定を見直してください。

詳細情報 : mod_mime: analyze_ct: cannot get media type from ' **メディアタイプ** '

MIME のメディアタイプが正しく取得できません。

エラーレベル：warn

(S) リクエスト処理を続行します。

(O)MIME の設定を見直してください。

詳細情報 : sigaction(**シグナル**)

シグナルに対するアクションの変更時にエラーが発生しました。
シグナルの種類を次に示します。
SIGSEGV SIGBUS SIGABRT SIGILL SIGTERM SIGINT
SIGXCPU SIGXFSZ SIGPIPE SIGHUP SIGUSR1 SIGUSR2

エラーレベル：warn

(S) Web サーバは起動処理を続行します。

(O)sigaction() 関数が返す詳細情報について見直してください。

---

long lost child came home! (pid **プロセス** ID)

graceful によるサーバの再起動後，制御プロセスが監視対象以外のプロセスを終了しました。

エラーレベル：warn

(S) 制御プロセスはスコアボードファイルを使用してサーバプロセスの状態を管理しています。graceful による再起動後に，終了したプロセスが管理しているプロセスに該当しない場合，このメッセージを出力します。

(O) 終了したプロセスが制御プロセスの監視対象外である原因について見直してください。例えば，gcache サーバは，制御プロセスの監視対象外のプロセスです。サーバプロセスは制御プロセスの監視対象であり，サーバプロセスに対してこのメッセージが出力された場合には，何らかの原因でサーバプロセスが監視対象から外れたことになります。

詳細情報 : killpg SIGTERM

Web サーバを終了させるために制御プロセスは killpg() 関数によってサーバプロセスに対してシグナル SIGTERM を送信しましたがエラーが発生しました。

エラーレベル：warn

(S) 制御プロセスは，数秒後にサーバプロセスそれぞれに対して SIGTERM シグナルを送信しサーバプロセス終了処理を続行します。

(O)killpg() 関数が返す**詳細情報**について見直してください。

詳細情報 : apr_socket_opt_set: (TCP_NODELAY)

IP プロトコルレベルのオプション TCP_NODELAY 設定時にエラーが発生しました。

エラーレベル：warn

(S)TCP_NODELAY オプションは設定しないで Web サーバの起動処理を続行します。Nagle バッファリングアルゴリズムは有効です。

(O)setsockopt() 関数が返す**詳細情報**について見直してください。

詳細情報 : write pipe_of_death

サーバプロセスの終了要求で使用するパイプへの書き込みに失敗しました。

エラーレベル：warn

(S) サーバプロセス終了要求処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : get socket to connect to listener

サーバプロセスの終了要求で使用するソケット生成に失敗しました。

エラーレベル：warn

(S) サーバプロセス終了要求処理を中断します。

(O)socket() 関数が返す**詳細情報**について見直してください。

詳細情報 : set timeout on socket to connect to listener

サーバプロセスの終了要求で使用するソケットのタイムアウト設定に失敗しました。

エラーレベル：warn

(S) サーバプロセス終了要求処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : connect to listener on Web **サーバアドレス** : **ポート番号**

サーバプロセス終了要求処理中，Web サーバへの接続に失敗しました。

エラーレベル：warn

(S) サーバプロセス終了要求処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : The HWS trace could not obtain the shared memory identifier from the file **ファイル名** specified in the HWSTraceIdFile.

HWSTraceIdFile ディレクティブで指定した**ファイル名**で示すファイルから共有メモリ識別子を取り出せません。

エラーレベル：warn

(S) 処理を続行します。

(O) 詳細情報で示す原因について見直してください。

The format of the shared memory identifier specified in the HWSTraceIdFile is invalid.

HWSTraceIdFile ディレクティブで指定したファイルの共有メモリ識別子はフォーマットが不正です。

エラーレベル：warn

(S) 処理を続行します。

(O) 共有メモリ識別子について見直してください。

詳細情報 : HWS trace shmctl() IPC_RMID could not remove the shared memory segment(id= **共有メモリ識別子**).

共有メモリを削除できません。

エラーレベル：warn

(S) 処理を続行します。

(O) 詳細情報で示す原因について見直してください。

詳細情報 : killpg SIGUSR1

graceful による再起動または計画停止させるために制御プロセスは killpg() 関数によってサーバプロセスに対してシグナル SIGUSR1 を送信しましたがエラーが発生しました。

エラーレベル：warn

(S) 制御プロセスは，再起動または計画停止処理を続行します。

(O)killpg() 関数が返す**詳細情報**について見直してください。

[client **クライアントアドレス**] authenticated user ' **ユーザ名称** ' not a member of any groups, so 'file-group' requirement cannot succeed for file ' **ファイル名称** '

認証ユーザはどのグループのメンバでもないため，file-group 要求は失敗しました。

エラーレベル：warn

(S) リクエスト処理を続行します。

詳細情報 : killpg SIGHUP

Web サーバを再起動させるために制御プロセスは killpg() 関数によってサーバプロセスに対してシグナル SIGHUP を送信しましたがエラーが発生しました。

エラーレベル：warn

(S) 制御プロセスは，数秒後にサーバプロセスそれぞれに対して SIGTERM シグナルを送信し，サーバプロセス終了処理を続行します。

(O)killpg() 関数が返す**詳細情報**について見直してください。

MaxClients can't be changed by restart. Original was used

MaxClients ディレクティブは，graceful または restart による再起動時には変更できません。

エラーレベル：warn

(S) Web サーバは起動時の MaxClients ディレクティブ値を設定し，再起動処理を続行します。

(O) 再起動時には，MaxClients ディレクティブ値を変更しないでください。

---

child process **プロセス** ID still did not exit, sending a SIGTERM

プロセス ID で示すサーバプロセスが終了していないため，SIGTERM シグナルを送信します。

エラーレベル：warn

(S) Web サーバの停止または再起動処理を続行します。

(O) サーバプロセスに対して SIGTERM シグナルがブロックされていないか確認し，サーバプロセスが終了するまでお待ちください。

(6) notice レベルのメッセージ

---

cannot use a full URL in a 401 ErrorDocument directive --- ignoring!

ErrorDocument ディレクティブでエラーステータスコードに 401 を指定したときにはフル URL 指定はできません。

エラーレベル：notice

(S) Web サーバは起動処理を続行します。

(O) エラーステータスコードに 401 を指定したときは，テキストまたはローカル URL を指定してください。

Child **プロセス** ID: Listening on port **ポート番号** .

ポート番号に示すポートへのリクエストを受け付けます。

エラーレベル：notice

(S) リクエスト受付処理を開始します。

Child **プロセス** ID: Acquired the start mutex.

サーバプロセス排他用のロックを獲得しました。

エラーレベル：notice

(S) サーバプロセスは起動処理を続行します。

Child **プロセス** ID: Starting **スレッド数** worker threads

スレッド数に示す数のサーバスレッドを生成します。

エラーレベル：notice

(S) サーバプロセスはサーバスレッド生成を開始します。

Child **プロセス** ID: Exit event signaled. Child process is ending.

サーバプロセスに終了イベントが通知されました。

エラーレベル：notice

(S) サーバプロセスの終了処理を開始します。

Child **プロセス** ID: Released the start mutex

サーバプロセス排他用のロックを解放しました。

エラーレベル：notice

(S) サーバプロセスは終了処理を続行します。

Child **プロセス** ID: Waiting for **生成サーバスレッド数** worker threads to exit.

サーバプロセスはサーバスレッドの終了待ち状態です。

エラーレベル：notice

(S) サーバプロセスはサーバスレッドの終了を待ちます。

Child **プロセス** ID: Terminating **未終了のサーバスレッド数** threads that failed to exit.

終了しないサーバスレッドを強制的に終了させます。

エラーレベル：notice

(S) サーバプロセスは終了処理を続行します。

Child **プロセス** ID: All worker threads have exited.

すべてのサーバスレッドが終了しました。

エラーレベル：notice

(S) サーバプロセスは終了処理を続行します。

Parent: Created child process **プロセス** ID

サーバプロセスを生成しました。

エラーレベル：notice

(S) Web サーバは起動処理を続行します。

Parent: Received shutdown signal -- Shutting down the server.

制御プロセスに停止イベントが通知されました。

エラーレベル：notice

(S) Web サーバは停止処理を開始します。

Parent: Received restart signal -- Restarting the server.

制御プロセスに再起動イベントが通知されました。

エラーレベル：notice

(S) Web サーバは再起動処理を開始します。

Parent: child process exited with status **終了コード** -- Restarting.

サーバプロセスは**終了コード**で終了しました。サーバプロセスを再起動します。

エラーレベル：notice

(S) Web サーバは再起動処理を開始します。

Parent: Child process exited successfully.

サーバプロセスは正常に終了しました。

エラーレベル：notice

(S) Web サーバは停止または計画停止処理を続行します。

Parent: Forcing termination of child process **ハンドル番号**

制御プロセスが設定した停止または計画停止イベントで終了しないサーバプロセスを強制的に終了させます。

エラーレベル：notice

(S) Web サーバは停止または計画停止処理を続行します。

(O) サーバプロセスが終了しない要因を見直してください。

Child プロセス ID: Child process is running

サーバプロセスの起動処理が開始されました。

エラーレベル：notice

(S) サーバプロセスは起動処理を続行します。

Child プロセス ID: Child process is exiting

サーバプロセスを終了します。

エラーレベル：notice

(S) サーバプロセスは終了処理を続行します。

サーバ名 configured -- resuming normal operations

Web サーバの起動が開始されました。

エラーレベル：notice

(S)Web サーバは起動処理を続行します。

Server built: Web サーバの構築された時刻

起動された Web サーバの構築時刻を示します。

エラーレベル：notice

(S)Web サーバの構築時刻を出力し，起動処理を続行します。

Child サーバプロセス ID: Graceful exit event signaled. Child process is ending.

サーバプロセスに計画停止イベントが通知されました。

エラーレベル：notice

(S) サーバプロセスの計画停止処理を開始します。

Parent: Received graceful stop signal -- Shutting down the server gracefully.

制御プロセスに計画停止イベントが通知されました。

エラーレベル：notice

(S)Web サーバは計画停止処理を開始します。

[client **クライアントアドレス**] {child process **プロセス** ID|server thread **スレッド** ID}:forcing termination of request " **リクエストライン** "

計画停止要求受付から強制停止時間が経過したため，**リクエストライン**に示すリクエスト処理を中断しました。

エラーレベル：notice

(S)Web サーバはリクエスト処理を中断して，停止処理を続行します。

---

caught **シグナル** , shutting down

Web サーバを停止しました。
シグナルの種類を次に示します。
SIGTERM SIGUSR2 SIGXCPU SIGXFSZ

エラーレベル：notice

(S)Web サーバを停止します。

Graceful restart requested, doing restart

Web サーバに対して graceful による再起動が要求されました。

エラーレベル：notice

(S)graceful による再起動処理を開始します。

SIGHUP received. Attempting to restart

Web サーバはシグナル SIGHUP による再起動要求を受信しました。

エラーレベル：notice

(S)Web サーバは再起動処理を開始します。

Graceful stop requested, doing graceful stop

Web サーバに対して計画停止が要求されました。

エラーレベル：notice

(S) 計画停止処理を開始します。

---

child pid **プロセス** ID exit signal **シグナル意味** (**シグナル番号**)〔, possible coredump in **コア出力先ディレクトリ**〕

サーバプロセスは，**シグナル番号**で示すシグナルで終了しました。出力可能である場合には，**コア出力先ディレクトリ**にコアを出力します。

エラーレベル：notice

(S) Web サーバは処理を続行します。

(O) 保守員に連絡してください。

---

seg fault or similar nasty error detected in the parent process

制御プロセスを終了させるシグナルを受信しました。

エラーレベル：notice

(S) 制御プロセスを終了します。

(O) 保守員に連絡してください。

Child **サーバプロセス** ID: Parent process exited. Child process is ending.

制御プロセスの終了を検知しました。

エラーレベル：notice

(S) サーバプロセスの終了処理を開始します。

Parent **制御プロセス** ID: Using config file " **コンフィグファイル名** "

制御プロセスのプロセス ID および起動処理で使用したコンフィグファイル名を示します。

エラーレベル：notice

(S) Web サーバは起動処理を続行します。

The server started by non-root user. User and Group directives is not used.

一般ユーザで起動しているため，User ディレクティブと Group ディレクティブは使用しません。

エラーレベル：notice

(S) Web サーバは起動処理を続行します。

（7） info レベルのメッセージ

[client **クライアントアドレス**] **詳細情報** : core_output_filter: writing data to the network

クライアントへのデータ送信処理中にエラーが発生しました。

エラーレベル：info

(S) Web サーバからクライアントへのデータ送信処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : apr_socket_addr_get(APR_LOCAL)

Web サーバのソケット情報を取得できませんでした。

エラーレベル：info

(S) Web サーバはリクエスト処理を中断します。

(O)getsockname() 関数が返す**詳細情報**について見直してください。

詳細情報 :apr_socket_addr_get(APR_REMOTE)

クライアントのソケット情報を取得できませんでした。

エラーレベル：info

(S) Web サーバはリクエスト処理を中断します。

(O)getpeername() 関数が返す**詳細情報**について見直してください。

No ExecCGI or Open verb found for files of type ' **拡張子** '.

レジストリキー ExecCGI，Open を検索しましたが，スクリプトの拡張子に関連づけられているインタプリタは見つかりませんでした。

エラーレベル：info

(S)"#!" 行に指定されたインタプリタを用いてスクリプト実行処理を続行します。

(O) スクリプトの拡張子とそれに関連づけられているプログラムについて見直してください。

詳細情報 : select failed

サーバプロセスの select() でエラーが発生しました。

エラーレベル：info

(S) リクエスト受付処理を続行します。

(O)select() 関数が返す**詳細情報**について見直してください。

Parent: Duplicating socket fd **番号** and sending it to child process **プロセス** ID

制御プロセスはソケットを複製し，サーバプロセスに送付します。

エラーレベル：info

(S) Web サーバは起動処理を続行します。

Using ConfigArgs of the installed service " **サービス名称** ".

サービス名称で示すサービスのレジストリエントリ ConfigArgs を起動オプションに使用します。

エラーレベル：info

(S) Web サーバは起動処理を続行します。

---

removed PID file **ファイル名** (pid= **制御プロセス** ID)

制御プロセス ID 格納ファイルを削除しました。

エラーレベル：info

(S) Web サーバは停止処理を続行します。

No ConfigArgs registered for **サービス名称** , perhaps this service is not installed?

サービス名称に示すサービスのレジストリエントリ ConfigArgs がありません。

エラーレベル：info

(S) Web サーバは起動処理を続行します。

(O) サービスがインストールされているか確認してください。

[client **クライアントアドレス**] client sent an unrecognized expectation value of Expect: **ヘッダ値**

クライアントは，Expect ヘッダ値にサポートしていない値を指定してリクエストを送付したため，リクエスト処理を続行できません。

エラーレベル：info

(S) ステータスコード「417　Expectation Failed」をクライアントに返し，リクエスト処理を中断します。

(O)Expect ヘッダ値には 100-continue だけが指定できます。Expect ヘッダを見直してください。

The HWS trace created shared memory segment # **共有メモリ識別子** .

内部トレース用の共有メモリを確保しました。

エラーレベル：info

(S) 処理を続行します。

The HWS trace output the log of the process (pid= **プロセス** ID) into the file **ファイル名** specified in the HWSTraceLogFile.

HWSTraceLogFile ディレクティブで指定した**ファイル名**に示すファイルへ，**プロセス** ID に示すプロセスの共有メモリの内容を出力しました。

エラーレベル：info

(S) ファイルへの出力を正常に終了して続行します。

Child **プロセス** ID: Accept thread exiting.

リクエスト受付スレッドを終了します。

エラーレベル：info

(S) サーバプロセスは終了処理を続行します。

---

server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers), spawning **生成予定のサーバプロセス数** children, there are **存在するアイドル状態のサーバプロセス数** idle, and **存在するサーバプロセス数** total children

Web サーバはリクエスト処理でビジー状態です。StartServers ディレクティブ値を増加させるかまたは Min/MaxSpareServers ディレクティブ値を見直す必要があります。生成予定のサーバプロセス数は，**生成予定のサーバプロセス数**に示す数です。現状のアイドル状態のサーバプロセス数は，**存在するアイドル状態のサーバプロセス数**に示す数です。現状の全体のサーバプロセス数は，**存在するサーバプロセス数**に示す数です。

エラーレベル：info

(S)Web サーバの起動を続行します。MinSpareServers ディレクティブで指定した数のアイドルサーバに到達するまで，1，2，4，8，16，32 個ずつ生成します。生成する数が

8 個以上になるとこのメッセージが出力されます。

[client **クライアントアドレス**] no AuthGroupFile, so 'file-group' requirement cannot succeed for file ' **ファイル名** '

AuthGroupFile が指定されていないため，file-group 要求は成功しません。

エラーレベル：info

(S) リクエスト処理を続行します。

[info] **詳細情報** : Socket Input: timed out (**接続元** IP **アドレス** : **ポート番号** --> **接続先** IP **アドレス** : **ポート番号**)(**サーバプロセス** ID)

クライアントまたはバックエンドサーバとの受信処理でタイムアウトしました。次のどれかの待ち時間を経過しました。

- クライアントからのリクエスト受信（コネクション確立後，HTTP プロトコルの受信）中にデータを受信しなくなった場合の待ち時間
- リバースプロキシを使用している場合の，バックエンドサーバへのリクエスト送信後からレスポンス受信までの待ち時間
- リバースプロキシを使用している場合の，バックエンドサーバからのレスポンス受信中にデータを受信しなくなった場合の待ち時間

エラーレベル：info

(S) 処理を続行します。

(O) Timeout ディレクティブで指定した時間を経過したため，このメッセージが出力されました。Timeout ディレクティブでの指定値を見直してください。

[info] **詳細情報** : Socket Output: timed out (**接続先** IP **アドレス** : **ポート番号** <-- **接続元** IP **アドレス** : **ポート番号**)(**サーバプロセス** ID)

クライアントまたはバックエンドサーバとの送信処理でタイムアウトしました。次のどちらかの待ち時間を経過しました。

- クライアントからのリクエスト受信（コネクション確立後，HTTP プロトコルの受信）中にデータを受信しなくなった場合の待ち時間
- リバースプロキシを使用している場合の，バックエンドサーバへのリクエスト送信後からレスポンス受信までの待ち時間
- リバースプロキシを使用している場合の，バックエンドサーバからのレスポンス受信中にデータを受信しなくなった場合の待ち時間

エラーレベル：info

(S) 処理を続行します。

(O)Timeout ディレクティブで指定した時間を経過したため，このメッセージが出力されました。Timeout ディレクティブでの指定値を見直してください。

[client **クライアントアドレス**] The server did not send a status-code **ステータスコード** response because the server already sent an error response to the client.

すでにエラーレスポンスをクライアントに送信しているため，**ステータスコード**のレスポンスは送信しません。

エラーレベル：info

(S) 処理を続行します。

(8) レベルなしのメッセージ

---

httpsd: module " **モジュール名** " is not compatible with this version (found **モジュールのバージョン番号** , need **互換性のあるバージョン番号**).

追加しようとしたモジュールは，サーバとの互換性がありません。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O) **互換性のあるバージョン番号**に示したバージョンのモジュールを使用してください。

Please contact the vendor for the correct version.

サーバと互換性のあるバージョンのモジュールについてベンダに問い合わせてください。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O) このメッセージのほかに出力されているメッセージのモジュールについて，サーバと互換性のあるバージョンのモジュールを入手してください。

---

httpsd: could not open document config file **コンフィグファイル名**

コンフィグファイル名に示すファイルを開けませんでした。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O) **コンフィグファイル名**について見直してください。

Syntax error on line **行番号** of **ファイル名** :

エラーメッセージ

ファイル名，**行番号**に示す位置に構文エラーがあります。

エラーレベル：なし

(S) Web サーバは起動処理を中断します。

(O) 構文エラーとなった行を**エラーメッセージ**について見直してください。

Warning: DocumentRoot [**ディレクトリ名**] does not exist

DocumentRoot ディレクティブで指定されたディレクトリが見つかりません。

エラーレベル：なし

(S) Web サーバは起動処理を続行します。

(O)<VirtualHost> ブロック内で指定した DocumentRoot ディレクティブについて見直してください。

Cannot resolve host name **ホスト名**

BindAddress ディレクティブの指定値が不正です。

エラーレベル：なし

(S) Web サーバは起動処理を中断します。

(O) BindAddress ディレクティブで指定した IP アドレスについて見直してください。

Host **ホスト名** has multiple addresses --- you must choose one explicitly.

BindAddress ディレクティブで指定したホスト名から一つの IP アドレスを特定できませんでした。

エラーレベル：なし

(S) Web サーバは起動処理を中断します。

(O) BindAddress ディレクティブは IP アドレスで指定してください

詳細情報 : Couldn't start ErrorLog process

エラーログを出力するためのプロセス生成に失敗しました。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O)ErrorLog ディレクティブを**詳細情報**について見直してください。

詳細情報 : Couldn't start RequestLog process

リクエストログを出力するためのプロセス生成に失敗しました。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O)HWSRequestLog ディレクティブを**詳細情報**について見直してください。

詳細情報 : httpsd: Invalid error log path **ファイル名** .

ErrorLog ディレクティブで指定したファイルパスが不正です。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O)ErrorLog ディレクティブを**詳細情報**について見直してください。

詳細情報 : httpsd: Invalid request log path **ファイル名** .

HWSRequestLog ディレクティブで指定したファイルパスが不正です。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O)HWSRequestLog ディレクティブを**詳細情報**について見直してください。

詳細情報 : httpsd: could not open error log file **ファイル名** .

エラーログファイルをオープンできませんでした。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O)ErrorLog ディレクティブを**詳細情報**について見直してください。

詳細情報 : httpsd: could not open request log file **ファイル名** .

リクエストログファイルをオープンできませんでした。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O)HWSRequestLog ディレクティブを**詳細情報**について見直してください。

---

piped_log_spawn: unable to setup child process ' **ログ出力プログラム** ': **エラーメッセージ**

ログを出力するためのプロセスの属性初期設定ができませんでした。

エラーレベル：なし

(S) ログ出力用のプロセス生成処理を中断します。

(O) **エラーメッセージ**について見直してください。

---

unable to start piped log program ' **ログ出力プログラム** ': **エラーメッセージ**

ログ出力プログラムによるログ出力を開始できませんでした。

エラーレベル：なし

(S) ログ出力用のプロセス生成処理を中断します。

(O) **エラーメッセージ**について見直してください。

---

piped log program ' **ログ出力プログラム** ' failed unexpectedly

ログ出力プログラムに予期しないエラーが発生しました。

エラーレベル：なし

(S) ログ出力用のプロセスを再生成します。

---

piped_log_maintenance: unable to respawn ' **ログ出力プログラム** ': **エラーメッセージ**

ログ出力用のプロセス再生成処理に失敗しました。

エラーレベル：なし

(S) ログ出力用のプロセス生成処理を中断します。

(O) **エラーメッセージ**について見直してください。

Ouch! malloc failed in **詳細情報**

メモリの確保に失敗しました。

エラーレベル：なし

(S) メモリが不足している状態です。Web サーバプロセスは終了します。

(O) メモリ使用量について，システムの状態を確認してください。

WARNING: ThreadsPerChild of **ディレクティブ値** exceeds ThreadLimit value of **上限値** threads, lowering ThreadsPerChild to **上限値** . To increase, please see the ThreadLimit directive.

ThreadsPerChild ディレクティブ値が上限値を超えています。

エラーレベル：なし

(S)Web サーバは ThreadsPerChild ディレクティブ値に**上限値**を設定して，起動処理を続行します。

(O)ThreadsPerChild ディレクティブ値を見直してください。

WARNING: Require ThreadsPerChild > 0, setting to 1

ThreadsPerChild ディレクティブに不正な値が指定されています。

エラーレベル：なし

(S)Web サーバは ThreadsPerChild ディレクティブ値に 1 を設定して，起動処理を続行します。

(O)ThreadsPerChild ディレクティブ値を見直してください。

WARNING: detected MinSpareServers set to non-positive.

Resetting to 1 to avoid almost certain Server failure.

Please read the documentation.

MinSpareServers ディレクティブに不正な値が指定されています。

エラーレベル：なし

(S)Web サーバは MinSpareServers ディレクティブ値に 1 を設定して，起動処理を続行します。

(O)MinSpareServers ディレクティブ値を見直してください。

WARNING: MaxClients of **ディレクティブ値** exceeds compile time limit of **上限値** servers,

lowering MaxClients to **上限値** .

MaxClients **ディレクティブ値**が**上限値**を超えています。

エラーレベル：なし

(S)Web サーバは MaxClients **ディレクティブ値**に**上限値**を設定して，起動処理を続行します。

(O)MaxClients **ディレクティブ値**を見直してください。

WARNING: Require MaxClients > 0, setting to 1

MaxClients ディレクティブに不正な値が指定されています。

エラーレベル：なし

(S)Web サーバは MaxClients ディレクティブ値に 1 を設定して，起動処理を続行します。

(O)MaxClients ディレクティブ値を見直してください。

---

httpsd: bad user name **ユーザ名**

指定されたユーザはシステムに登録されていません。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O) システムに登録された**ユーザ名**を指定してください。

---

httpsd: bad group name **グループ名**

指定されたグループはシステムに登録されていません。

エラーレベル：なし

(S)Web サーバは起動処理を中断します。

(O) システムに登録された**グループ名**を指定してください。

詳細情報 : sending signal to server

Web サーバへのシグナル送信に失敗しました。

エラーレベル：なし

(S)Web サーバへのシグナル送信処理を中断します。

(O)kill() 関数が返す**詳細情報**について見直してください。

詳細情報 : Error retrieving pid file PID **格納ファイル名**

PID 格納ファイルの検索に失敗しました。

エラーレベル：なし

(S)Web サーバへのシグナル送信処理を中断します。

(O) **詳細情報**について見直してください。

Ouch! **関数名** 1 failed : (**エラーコード**) **詳細情報** (**関数名** 2)(**プロセス** ID)

関数名 1 の実行に失敗しました。

エラーレベル：なし

(S)Web サーバのプロセスは終了します。

(O) **関数名** 1 が返す**詳細情報**について見直してください。エラーの原因の一つとして，Windows が内部的に使用しているデスクトップヒープの不足が考えられます。

7.2.2 SSL についてのメッセージ

---

allocate error

SSL 処理に必要なメモリの確保に失敗しました。

エラーレベル：error

(S)SSL による接続ができません。

(O) システムリソースの使用状況を確認してください。

[client **クライアント** IP **アドレス**] [port **クライアントポート番号**] allocate error

SSL 処理に必要なメモリの確保に失敗しました。

エラーレベル：error

(S)SSL による接続ができません。

(O) システムリソースの使用状況を確認してください。

Attempt to reinitialise SSL for server **ホスト名**

ホストの設定を再初期化しようとしました。

エラーレベル：crit

(S)Web サーバを起動しません。

(O) バーチャルホスト内の SSL の設定を確認してください（少なくとも一つ以上の SSL 関連ディレクティブは設定しなければなりません）。

Bad password for the private key

SSLCertificateKeyPassword ディレクティブで指定されたパスワードファイルから正しいパスワードを読み込めませんでした。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) 正しいパスワードをパスワードファイルに設定してください。

Can't open certificate file Web **サーバ証明書ファイル** , nor **内部生成パス名**

証明書を読み込めません。

エラーレベル：crit

(S) Web サーバを起動しません。

(O)SSLCertificateFile ディレクティブの設定値を確認してください。

Could not get lastUpdate field in CRL: **ファイル名**

CRL の発行日が取得できませんでした。

エラーレベル：crit

(S) Web サーバを起動しません。

(O)CRL が正しく作成されているか，正しくダウンロードされたか確認してください。

Could not load the certificate file.

サーバ証明書ファイルの読み込みに失敗しました。

エラーレベル：crit

(S) Web サーバを起動しません。

(O)SSLCertificateFile ディレクティブの設定値を確認してください。

Could not read the private key file

サーバ秘密鍵ファイルが読み込めませんでした。

エラーレベル：なし

(S) 処理を終了します。

(O)SSLCertificateKeyFile ディレクティブの設定値を確認してください。

Could not setup a new lock.

ロック処理の初期化に失敗しました。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) システムリソースの使用状況を確認してください。

CRL expired, but CRL check passed: issuer= **発行者名** ,serial= **シリアル番号**

CRL の次回発行日を過ぎていますが，CRL にはクライアント証明書のシリアル番号の記載がなく，また，SSLCRLAuthoritative ディレクティブに Off が設定されているため，クライアントのアクセスを許可しました。

エラーレベル：warn

(S) 処理を継続します。

(O)CRL を更新してください。

CRL expired, but serial number was found in CRL: issuer= **発行者名** ,serial= **シリアル番号**

CRL の次回発行日を過ぎていますが，CRL にクライアント証明書のシリアル番号の記載があったため，SSL ハンドシェイクに失敗しました。

エラーレベル：error

(S)SSL ハンドシェイクに失敗したため，アクセスを拒否します。

(O)CRL を更新してください。

CRL expired: issuer= **発行者名**

CRL の次回発行日を過ぎていたため，SSL ハンドシェイクに失敗しました。

エラーレベル：error

(S)SSL ハンドシェイクに失敗したため，アクセスを拒否します。

(O)CRL を更新してください。

CRL expired: **ファイル名**

次回発行日を過ぎた CRL を読み込みました。

エラーレベル：warn

(S) 処理を続行します。クライアント認証の際には，SSLCRLAuthoritative ディレクティブ設定値に従い処理します。

(O) 新規 CRL を取得してください。

CRL is a duplicate: **ファイル名** will not be used in server **ホスト名** : **ポート番号**

一つの CA から発行された（同じ Subject を持つ）CRL を複数読み込みましたので，表示された CRL ファイルは使用しません。

エラーレベル：warn

(S) 処理を続行します。

(O) 一つの CA から発行された（同じ Subject を持つ）CRL は，ディレクトリ内に一つだけ格納してください。

CRL is not a valid type: **ファイル名**

SSLCRLDERPath または SSLCRLPEMPath ディレクティブで指定されたディレクトリ内から予期しないファイルを読み込みました。

エラーレベル：crit

(S)Web サーバを起動しません。

(O)CRL 以外のファイルはディレクトリ内に格納できません。CRL のファイルフォーマットを確認して，必要であれば形式変換するかまたは適切なディレクトリに格納してください。

CRL is not valid: issuer= **発行者名**

CRL が有効ではありません。

エラーレベル：error

(S)SSL ハンドシェイクに失敗したため，アクセスを拒否します。

(O)CRL が正しく作成されているか確認してください。

CRL is not yet valid: issuer= **発行者名**

CRL の発行日が現在時刻より前であったため，クライアント証明書の認証に失敗しました。

エラーレベル：error

(S)SSL ハンドシェイクに失敗したため，アクセスを拒否します。

(O) システムの時刻設定を見直してください。

CRL is not yet valid: **ファイル名**

CRL の発行日が，現在時刻より後に設定されていました。

エラーレベル：warn

(S) 処理を続行します。

(O) システムの時刻設定を見直してください。

CRL verify error: issuer= **発行者名**

CRL の署名検証に失敗しました。

エラーレベル：error

(S) クライアント証明書の認証に失敗します。

(O) 正しい CRL を読み込んでいることを確認してください。

---

data set error

SSL の初期化処理に失敗しました。

エラーレベル：error

(S)SSL による接続ができません。

(O) システムリソースの使用状況を確認してください。

[client **クライアントアドレス**] [port **クライアントポート番号**] data set error

SSL の初期化処理に失敗しました。

エラーレベル：error

(S)SSL による接続ができません。

(O) システムリソースの使用状況を確認してください。

Depth of certificate chain (CA **証明書の深さ**) exceeded SSLExportCertChainDepth limit: subject=(CA **証明書の** subject)

証明書検証に成功しましたが，SSLExportCertChainDepth ディレクティブ設定値が 1 以上であり，かつその設定値を超えた証明書チェーンをクライアントが送ってきたため，

ディレクティブ設定値で環境変数設定とキャッシュへの格納を打ち切りました。
エラーメッセージは，設定値を超えたすべての CA 証明書に対して出力されます。

(例) 証明書チェーンが（クライアント証明書を除き）3 段階であり，SSLExportCertChainDepth ディレクティブの設定値が 1 のときには，上記エラーメッセージは 2 回出力されます。

エラーレベル：warn

(S) 処理を継続します。

Error reading server certificate file **ファイル名**

証明書を読み込めません。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) 証明書の形式が適切であるかどうか確認してください。

---

error setting verify locations

SSLCACertificateFile または SSLCACertificatePath ディレクティブで指定したパス名を設定できません。

エラーレベル：crit

(S) Web サーバを起動しません。

(O)SSLCACertificateFile または SSLCACertificatePath ディレクティブの設定を確認してください。

Failed to stack CRL in ReadCRL()

データ格納処理に失敗しました。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) Web サーバを再起動してください。

Malloc error in GetCertificateAndKey()

メモリ確保に失敗しました。

エラーレベル：crit

(S)Web サーバを起動しません。

(O) システムリソースの使用状況を確認してください。

Malloc error in GetPrivateKey()

メモリ確保に失敗しました。

エラーレベル：crit

(S)Web サーバを起動しません。

(O) システムリソースの使用状況を確認してください。

Malloc error in SetupLock()

メモリ確保に失敗しました。

エラーレベル：crit

(S)Web サーバを起動しません。

(O) システムリソースの使用状況を確認してください。

---

malloc failed in CRLCheck()

処理に必要なメモリ確保に失敗しました。

エラーレベル：error

(S) クライアント証明書の認証に失敗します。

(O) システムのメモリ使用量を確認してください。

---

malloc failed in GetCRL()

処理に必要なメモリ確保に失敗しました。

エラーレベル：crit

(S)Web サーバを起動しません。

(O) システムのメモリ使用量を確認してください。

---

malloc failed in ReadCRL()

処理に必要なメモリ確保に失敗しました。

エラーレベル：crit

(S)Web サーバを起動しません。

(O) システムのメモリ使用量を確認してください。

No client certificate

クライアント証明書が送信されていません。

エラーレベル：error

(S)SSL リクエスト処理を終了します。

[client **クライアントアドレス**] [port **クライアントポート番号**] No client certificate

クライアント証明書が送信されていません。

エラーレベル：error

(S)SSL リクエスト処理を終了します。

No SSL Certificate set for server **ホスト名** : **ポート**

Web サーバ証明書が設定されていません。

エラーレベル：crit

(S)Web サーバを起動しません。

(O)SSLCertificateFile ディレクティブを設定します。

Required SSLCacheServerPath missing. gcache will not be started.

SSLCacheServerPath ディレクティブが指定されていないため，gcache サーバを起動できません。

エラーレベル：error

(S)gcache サーバを起動しません。

(O) セション管理機能を使用するときは，SSLCacheServerPath ディレクティブを指定してください。

Required SSLCacheServerPort missing. gcache will not be started.

SSLCacheServerPort ディレクティブが指定されていないため，gcache サーバを起動できません。

エラーレベル：error

(S)gcache サーバを起動しません。

(O) セション管理機能を使用するときは，SSLCacheServerPort ディレクティブを指定してください。

Required SSLCertificateKeyPassword missing

SSLCertificateKeyPassword ディレクティブが設定されていません。

エラーレベル：crit

(S)Web サーバを起動しません。

(O) SSLCertificateKeyPassword ディレクティブを設定してください。

Serial number was found in CRL: issuer= **発行者名** ,serial= **シリアル番号**

CRL にクライアント証明書のシリアル番号の記載があったため，SSL ハンドシェイクに失敗しました。

エラーレベル：error

(S)SSL ハンドシェイクに失敗したため，アクセスを拒否します。

Set error in GetCertificateAndKey()

SSL の初期化処理に失敗しました。

エラーレベル：crit

(S)Web サーバを起動しません。

(O) システムリソースの使用状況を確認してください。

SSLExportCertChainDepth is outside the appropriate range

SSLExportCertChainDepth ディレクティブで，指定できない値が設定されました。

エラーレベル：なし

(S)Web サーバを起動しません。

(O) ディレクティブ指定値を見直してください。

[client **クライアントアドレス**] **詳細情報** : SSL handshake interrupted by system: client port **ポート番号**

SSL ハンドシェイク処理が正しく終了しませんでした。

エラーレベル：info

(S)SSL リクエスト処理を終了します。

[client **クライアントアドレス**] **詳細情報** : SSL handshake interrupted by system: client port **ポート番号** (SSL **ハンドシェイク処理時間**)(**エラーナンバー値**)(**サーバプロセス** ID):SSL **ハンドシェイク処理の状態**

SSL ハンドシェイク処理が正しく終了しませんでした。

エラーレベル：info

(S)SSL リクエスト処理を終了します。

[client **クライアントアドレス**] SSL library error **エラー番号** in handshake

SSL ハンドシェイク処理が正しく終了しませんでした。

エラーレベル：error

(S)SSL リクエスト処理を終了します。

[client **クライアントアドレス**] [port **クライアントポート番号**] SSL library error **エラー番号** in handshake(SSL **ハンドシェイク処理時間**)(**エラーナンバー値**)(**サーバプロセス** ID):SSL **ハンドシェイク処理の状態**

SSL ハンドシェイク処理が正しく終了しませんでした。

エラーレベル：error

(S)SSL リクエスト処理を終了します。

SSL Library Error: **詳細情報**

SSL ライブラリでエラーが発生しました。

エラーレベル：crit または error

(S) Web サーバ起動中の場合は，起動処理を中断します。SSL リクエスト処理中の場合は，SSL リクエスト処理を中断します。

(O) 詳細情報について見直してください。

SSLSessionCacheSize is outside the appropriate range

SSLSessionCacheSize ディレクティブで，指定できない値が設定されました。

エラーレベル：なし

(S)Web サーバを起動しません。

(O) ディレクティブの指定値を見直してください。

SSLSessionCacheSizePerChild is outside the appropriate range

SSLSessionCacheSizePerChild ディレクティブで，指定できない値が設定されました。

エラーレベル：なし

(S)Web サーバを起動しません。

(O) ディレクティブの指定値を見直してください。

SSLSessionCacheTimeout not set

SSLSessionCacheTimeout ディレクティブの値が設定されていません。

エラーレベル：crit

(S)Web サーバを起動しません。

(O)SSLSessionCacheTimeout ディレクティブを設定してください。

The private key doesn't match the public key

SSLCertificateFile ディレクティブおよび SSLCertificateKeyFile ディレクティブで指定した，Web サーバ秘密鍵と Web サーバ証明書の対応が不正です。

エラーレベル：crit

(S)Web サーバを起動しません。

(O) 秘密鍵と証明書が正しいペアで設定されているかどうか確認してください。

---

unable to set certificate

SSLCertificateFile ディレクティブで指定した Web サーバ証明書を正しく設定できません。

エラーレベル：crit

(S) サーバを起動しません。

(O)SSLCertificateFile ディレクティブで指定した Web サーバ証明書が正しいフォーマットであるかどうか確認してください。

unable to set ciphers

SSLRequiredCiphers ディレクティブで指定した暗号種別が設定できません。

エラーレベル：crit

(S) Web サーバを起動しません。

(O)SSLRequiredCiphers ディレクティブの指定値を確認してください。

unable to set private key

SSLCertificateKeyFile または SSLCertificateFile ディレクティブで指定した Web サーバ秘密鍵が正しく設定できません。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) 秘密鍵のフォーマットが正しいかどうか，また，秘密鍵と証明書が正しいペアで設定されているかどうか確認してください。

Verify depth exceeded

SSLVerifyDepth ディレクティブの設定値より深い階層に位置するクライアント証明書が送られてきたため，検証に失敗しました。

エラーレベル：error

(S)SSL リクエスト処理を終了します。

(O)SSLVerifyDepth ディレクティブの設定値を確認してください。クライアント証明書を受け付けないときは対処する必要はありません。

[client **クライアントアドレス**] [port **クライアントポート番号**] Verify depth exceeded

SSLVerifyDepth ディレクティブの設定値より深い階層に位置するクライアント証明書が送られてきたため，検証に失敗しました。

エラーレベル：error

(S)SSL リクエスト処理を終了します。

(O)SSLVerifyDepth ディレクティブの設定値を確認してください。クライアント証明書を受け付けないときは対処する必要はありません。

verify error

クライアント証明書の検証に失敗し，かつその証明書の issuer が取得できません。

エラーレベル：error

(S)SSL リクエスト処理を終了します。

(O)SSLCertificateFile および SSLCertificatePath ディレクティブで設定されている CA 証明書を確認してください。そのクライアント証明書を受け付けない場合には，対処する必要はありません。

[client **クライアントアドレス**] [port **クライアントポート番号**] verify error

クライアント証明書の検証に失敗し，かつその証明書の issuer が取得できません。

エラーレベル：error

(S)SSL リクエスト処理を終了します。

(O)SSLCertificateFile および SSLCertificatePath ディレクティブで設定されている CA 証明書を確認してください。そのクライアント証明書を受け付けない場合には，対処する必要はありません。

---

verify error:num= **値** : **エラーメッセージ**

署名検証に失敗しました。

エラーレベル：error

(S)SSL リクエスト処理を終了します。

(O) 対応する CA 証明書を読み込んでください。そのクライアント証明書を受け付けない場合には，対処する必要はありません。

[client **クライアントアドレス**] [port **クライアントポート番号**] verify error:num= **値** : **エラーメッセージ**

署名検証に失敗しました。

エラーレベル：error

(S)SSL リクエスト処理を終了します。

(O) 対応する CA 証明書を読み込んでください。そのクライアント証明書を受け付けない場合には，対処する必要はありません。

詳細情報 : Can't open directory **ディレクトリ名**

SSLCACertificatePath ディレクティブで指定されたディレクトリが開けません。

エラーレベル：error

(S) Web サーバを起動しません。

(O) 詳細情報で示す原因について，ディレクトリがあるかどうかまたはパーミッションなどを確認してください。

詳細情報 : Can't open key file **ファイル名**

秘密鍵ファイルを読み込めません。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) 詳細情報で示す原因について，SSLCertificateKeyFile ディレクティブの指定を確認してください。

---

access to **ファイル名** failed for **ホスト名** , reason: Cipher **暗号種別** is not on the permitted list

SSLRequireCipher ディレクティブで指定されていない暗号種別を使用してアクセスしました。

エラーレベル：error

(S) ステータスコード「403 Forbidden」を返してリクエスト処理を終了します。

---

access to **ファイル名** failed for **ホスト名** , reason: Cipher **暗号種別** is forbidden

SSLBanCipher ディレクティブで指定した**暗号種別**を使用してアクセスしました。

エラーレベル：error

(S) ステータスコード「403 Forbidden」を返してリクエスト処理を終了します。

詳細情報 : Could not create a new mutex

ロック処理の初期化に失敗しました。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) 詳細情報を基に対応してください。

詳細情報 : Could not open CRL directory for DER format: **ディレクトリ名**

SSLCRLDERPath ディレクティブで指定されたディレクトリが開けません。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) 詳細情報で示す原因について，対応してください。

詳細情報 : Could not open CRL directory for PEM format: **ディレクトリ名**

SSLCRLPEMPath ディレクティブで指定されたディレクトリが開けません。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) 詳細情報で示す原因について，対応してください。

詳細情報 : Could not open CRL file: **ファイル名**

CRL ファイルが開けません。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) 詳細情報で示す原因について，対応してください。

詳細情報 : Could not Read password file.

SSLCertificateKeyPassword ディレクティブで指定されたパスワードファイルが読み込めませんでした。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) 詳細情報を基に対応してください。

詳細情報 : couldn't change working directory: **ディレクトリ**

gcache サーバが動作する**ディレクトリ**の設定に失敗しました。

エラーレベル：なし

(S)gcache サーバは起動しませんが，Web サーバは起動処理を続行します。

(O)SSLCacheServerRunDir ディレクティブの設定を，chdir() 関数が返す**詳細情報**について見直してください。

詳細情報 : Error reading private key file **ファイル名** :

秘密鍵を読み込めません。

エラーレベル：crit

(S) Web サーバを起動しません。

(O) 詳細情報で示す原因について，SSLCertificateKeyFile ディレクティブの指定値を確認してください。

---

access to **ファイル名** failed for **ホスト名** , reason: SSL denied

SSLDenySSL ディレクティブで指定したディレクトリに SSL を使用してアクセスしました。

エラーレベル：error

(S) ステータスコード「403 Forbidden」を返してリクエスト処理を終了します。

---

access to **ファイル名** failed for **ホスト名** , reason: SSL required

SSLRequireSSL ディレクティブで指定したディレクトリに SSL を使用しないでアクセスしました。

エラーレベル：error

(S) ステータスコード「403 Forbidden」を返してリクエスト処理を終了します。

詳細情報 : unable to exec gcache: **ファイル名**

ファイル名に示された gcache サーバの起動に失敗しました。

エラーレベル：なし

(S)gcache サーバは起動しませんが，Web サーバは起動処理を続行します。

(O)SSLCacheServerPath ディレクティブの設定を，execl() 関数が返す**詳細情報**について見直してください。

詳細情報 : unable to spawn gcache process

gcache サーバの起動に失敗しました。

エラーレベル：crit

(S)gcache サーバは起動しませんが，Web サーバは起動処理を続行します。

(O)fork() 関数が返す**詳細情報**について見直してください。

7.2.3 リバースプロキシについてのメッセージ

[client **クライアントアドレス**] proxy: Max-Forwards has reached zero - proxy loop? returned by URI

Max-Forwards リクエストヘッダの値が 0 になりました。

エラーレベル：error

(S) ステータスコード「502 Proxy Error」をクライアントに返し，リクエスト処理を中断します。

(O) Max-Forwards リクエストヘッダを使用する場合は，TRACE メソッドまたは OPTIONS メソッドを使用してください。

[client **クライアントアドレス**] proxy: No protocol handler was valid for the URL **パス名** . If you are using a DSO version of mod_proxy, make sure the proxy submodules are included in the configuration using LoadModule.

ProxyPass ディレクティブで指定したスキーマが不正です。

エラーレベル：warn

(S) ステータスコード「403 Forbidden」をクライアントに返し，リクエスト処理を中断します。

(O) ProxyPass ディレクティブの転送先のスキーマを見直してください。

詳細情報 : proxy: HTTP: error creating fam **アドレスファミリ** socket for target **ホスト名**

ソケットの作成に失敗しました。

エラーレベル：error

(S) ステータスコード「502 Bad Gateway」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : proxy: HTTP: attempt to connect to IP **アドレス** : **ポート番号** (**ホスト名**) failed

リバースプロキシはリモートの Web サーバへの接続に失敗しました。

エラーレベル：error

(S) ステータスコード「502 Bad Gateway」をクライアントに返し，リクエスト処理を中断します。

(O) **詳細情報**について見直してください。

[client **クライアントアドレス**] proxy: TRACE forbidden by server configuration

TraceEnable ディレクティブの設定により，TRACE メソッドによるリクエストを拒否します。

エラーレベル：error

(S) ステータスコード「403 Forbidden」をクライアントに返し，リクエスト処理を中断します。

(O) アクセスを許可する場合は TraceEnable ディレクティブの設定を見直してください。

[client **クライアントアドレス**] proxy: TRACE with request body is not allowed

TraceEnable ディレクティブの設定により，リクエストボディが付加されている TRACE メソッドによるリクエストを拒否します。

エラーレベル：error

(S) ステータスコード「413 Request Entity Too Large」をクライアントに返し，リクエスト処理を中断します。

(O) アクセスを許可する場合は TraceEnable ディレクティブの設定を見直してください。

[client **クライアントアドレス**] error parsing URL URL: **詳細情報**

リバースプロキシによる URL の解析中にエラーが発生しました。

エラーレベル：error

(S) ステータスコード「400 Bad Request」をクライアントに返し，リクエスト処理を中断します。

(O)ProxyPass ディレクティブで指定している URL に関して**詳細情報**に示す原因について見直してください。

[client **クライアントアドレス**] proxy: URI cannot be parsed: **転送先** URI returned by **リクエスト** URI

ProxyPass ディレクティブで指定した転送先の URI のフォーマットが不正です。

エラーレベル：error

(S) ステータスコード「400 Proxy Error」をクライアントに返し，リクエスト処理を中断します。

(O)ProxyPass ディレクティブで指定した転送先の URI を見直してください。

[client **クライアントアドレス**] proxy: DNS lookup failure for: **ホスト名** returned by URI

ProxyPass ディレクティブで指定した転送先のホスト名の DNS ルックアップに失敗しました。

エラーレベル：error

(S) ステータスコード「502 Proxy Error」をクライアントに返し，リクエスト処理を中断します。

(O)ProxyPass ディレクティブで指定した転送先のホスト名について見直してください。

---

proxy: previous connection is closed, creating a new connection.

以前のコネクションはクローズしているため，新しいコネクションを作成します。

エラーレベル：info

(S) 処理を続行します。新しいソケットを作成します。

---

proxy: failed to enable ssl support for IP **アドレス** : **ポート番号** (**ホスト名**)

リバースプロキシはリモートの Web サーバとのコネクションに SSL を使用できません。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O)Web サーバのアクセス方法に関する設定を見直してください。

[client **クライアントアドレス**] proxy: no HTTP 0.9 request (with no host line) on incoming request and preserve host set forcing hostname to be **ホスト名** for uri URI

クライアントから受信したリクエストヘッダに Host ヘッダが含まれていません。

エラーレベル：warn

(S) リバースプロキシが作成した Host ヘッダをリモートの Web サーバへ送信して処理を続行します。

(O) クライアントの設定を見直してください。

[client **クライアントアドレス**] proxy: error reading status line from remote server **ホスト名**

リバースプロキシはリモートの Web サーバからのステータスラインの読み込みに失敗しました。

エラーレベル：error

(S) ステータスコード「502 Proxy Error」をクライアントに返し，リクエスト処理を中断します。

(O) リモートの Web サーバが送信したレスポンスを見直してください。

[client **クライアントアドレス**] proxy: Error reading from remote server returned by URI

リバースプロキシはリモートの Web サーバからのステータスラインの読み込みに失敗しました。

エラーレベル：error

(S) ステータスコード「502 Proxy Error」をクライアントに返し，リクエスト処理を中断します。

(O) リモートの Web サーバが送信したレスポンスを見直してください。

[client **クライアントアドレス**] proxy: error reading response header from remote server **ホスト名**

リバースプロキシはリモートの Web サーバからのレスポンスヘッダの読み込みに失敗しました。

エラーレベル：error

(S) ステータスコード「502 Bad Gateway」をクライアントに返し，リクエスト処理を中断します。

(O) リモートの Web サーバが送信したレスポンスを見直してください。

[client **クライアントアドレス**] **詳細情報** : proxy: error reading response body from remote server **ホスト名**

リバースプロキシはリモートの Web サーバからのレスポンスボディの読み込みに失敗しました。

エラーレベル：error

(S) クライアントおよびリモートの Web サーバとの接続を切断します。

(O) リモートの Web サーバが送信したレスポンスを見直してください。

[client **クライアントアドレス**] proxy: Corrupt status line returned by remote server: **文字列** returned by URI

リモートの Web サーバが不正な HTTP ヘッダを送信しました。

エラーレベル：error

(S) ステータスコード「502 Proxy Error」をクライアントに返し，リクエスト処理を中断します。

(O) リモートの Web サーバが送信したレスポンスを見直してください。

---

proxy: bad HTTP **メジャーバージョン** . **マイナーバージョン** header returned by URI (**メソッド**)

リモートの Web サーバが不正な HTTP ヘッダを送信しました。

エラーレベル：warn

(S) ステータスコード「502 Bad Gateway」をクライアントに返し，リクエスト処理を中断します。

(O) リモートの Web サーバの設定を見直してください。

詳細情報 : proxy: pass request body failed to IP **アドレス** : **ポート番号** (**ホスト名**)

リバースプロキシはリモートの Web サーバへのリクエストボディの送信に失敗しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報に示す原因について見直してください。

---

proxy: client **クライアントアドレス** given Content-Length did not match number of body bytes read

クライアントが送信したリクエストボディのサイズが Content-Length ヘッダ値と異なります。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) Content-Length リクエストヘッダ値が正しいかどうか見直してください。

詳細情報 : proxy: search for temporary directory failed

リバースプロキシはリクエストボディを一時的に格納するディレクトリの検索に失敗しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報に示す原因について見直してください。

詳細情報 : proxy: creation of temporary file in directory **ディレクトリ名** failed

リバースプロキシはリクエストボディを一時的に格納するファイルの作成に失敗しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報に示す原因について見直してください。

詳細情報 : proxy: write to temporary file **ファイル名** failed

リバースプロキシはリクエストボディを一時的に格納するファイルへの書き込みに失敗しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報に示す原因について見直してください。

---

proxy: **転送コーディング値** Transfer-Encoding is not supported

クライアントが送信した Transfer-Encoding ヘッダに，サポートしていない転送コーディング値が設定されています。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) Transfer-Encoding ヘッダに使用する転送コーディング値は chunked を使用してください。

詳細情報 : proxy: prefetch request body failed to **ホスト名** from **クライアントアドレス** (**クライアントホスト名**)

リバースプロキシはクライアントからのリクエストボディの受信に失敗しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報に示す原因について見直してください。

詳細情報 : proxy: pass request body failed to IP **アドレス** : **ポート番号** (**ホスト名**) from **クライアントアドレス** (**クライアントホスト名**)

リバースプロキシはリクエストボディの送信に失敗しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報に示す原因について見直してください。

[client **クライアントアドレス**] proxy fin wait timed out.

リモートの Web サーバからの FIN パケット待ち処理がタイムアウトしました。

エラーレベル：info

(S) リバースプロキシから FIN パケットを送ることによって，リモートの Web サーバとの接続をクローズして処理を続行します。

(O) リモートの Web サーバのコネクションに関する設定を見直してください。

詳細情報 : proxy: processing prefetched request body failed to **ホスト名** from **クライアントアドレス** (**クライアントホスト名**)

リバースプロキシはクライアントからのリクエストボディの受信に失敗しました。

エラーレベル：error

(S) ステータスコード「500 Internal Server Error」をクライアントに返し，リクエスト処理を中断します。

(O) 詳細情報に示す原因について見直してください。

[client **クライアントアドレス**] proxy: Could not get proxypass entry

バックエンドサーバから読み込んだ Set-Cookie ヘッダの変換に使用する ProxyPass ディレクティブの指定値を取得できませんでした。

エラーレベル：error

(S) バックエンドサーバから読み込んだ Set-Cookie ヘッダを変換しないで，リクエスト処理を続行します。

[client **クライアントアドレス**] proxy: Too many (**上限回数**) interim responses from origin server returned by URI

リモートの Web サーバが**上限回数**を超えるステータスコード 100 を送信しました。

エラーレベル：error

(S) ステータスコード「502 Proxy Error」をクライアントに返し，リクエスト処理を中断します。

(O) リモートの Web サーバが送信したレスポンスを見直してください。

7.2.4 流量制限機能についてのメッセージ

詳細情報 : Could not open file: **ファイル名**

QOSResponse ディレクティブで指定したファイルがオープンできません。

エラーレベル：crit

(S) サーバが起動しません。

(O) 詳細情報を基にディレクティブ設定値を見直してください。

詳細情報 : Could not read file: **ファイル名**

QOSResponse ディレクティブで指定したファイルが読み込めません。

エラーレベル：crit

(S) サーバが起動しません。

(O) 詳細情報を基に対応してください。

This file is too large: **ファイル名**

QOSResponse ディレクティブで指定したファイルが大き過ぎて読み込めません。

エラーレベル：crit

(S) サーバが起動しません。

(O)4,294,967,295 バイトよりも小さなファイルを指定してください。

QOSCookieServers is out of range

QOSCookieServers ディレクティブの指定範囲が不正です。

エラーレベル：crit

(S) サーバが起動しません。

(O) ディレクティブ設定値を見直してください。

QOSRejectionServers is out of range

QOSRejectionServers ディレクティブの指定範囲が不正です。

エラーレベル：crit

(S) サーバが起動しません。

(O) ディレクティブ設定値を見直してください。

[client **クライアントアドレス**] **詳細情報** : client stopped connection before send QOSResponse completed

クライアントからプロトコルボディデータ受信中または Web サーバからクライアントへデータ送信中，クライアントからコネクションが切断されました。

エラーレベル：info

(S) Web サーバからクライアントへのデータ送信処理を中断します。

Request rejected (service temporarily unavailable)

リクエスト処理が拒否されました。

エラーレベル：info

(S) リクエスト処理を中断します。

7.2.5 静的コンテンツキャッシュ機能についてのメッセージ

詳細情報 : Error creating mutex

mutex 作成時にエラーが発生しました。

エラーレベル：crit

(S) Web サーバを停止します。

(O) 詳細情報に示す原因について見直してください。

---

malloc failure in mod_hws_cache

ファイルのキャッシュ処理中にメモリ確保エラーが発生しました。

エラーレベル：warn

(S) ファイルはキャッシュされませんが処理を続行します。

HWSContentCacheMaxFileSize exceeded HWSContentCacheSize: reduced to HWSContentCacheSize **設定値**

HWSContentCacheMaxFileSize ディレクティブ値が，キャッシュ可能なメモリサイズを超えています。HWSContentCacheSize ディレクティブ値まで減少させました。

エラーレベル：notice

(S) 処理を続行します。

[client **クライアントアドレス**] **詳細情報** : client stopped connection before send cache completed

クライアントからプロトコルボディデータを受信中または Web サーバからクライアントへデータを送信中，クライアントからコネクションが切断されました。

エラーレベル：info

(S) Web サーバからクライアントへのデータ送信処理を中断します。

---

cache error: reason(file open error)

ファイルのオープンに失敗したためキャッシュできませんでした。

エラーレベル：info

(S) ファイルはキャッシュされませんが処理を続行します。

cache error: reason(file read error)

ファイルの読み込みに失敗したためキャッシュできませんでした。

エラーレベル：info

(S) ファイルはキャッシュされませんが処理を続行します。

cache error: reason(file size unstable)

ファイルサイズが確定できないためキャッシュできませんでした。

エラーレベル：info

(S) ファイルはキャッシュされませんが処理を続行します。

7.2.6 LDAP 連携機能についてのメッセージ

[client **クライアントアドレス**] An attempt to bind user **ユーザ** ID to the LDAP server failed: **詳細情報**

LDAP サーバでのユーザ認証に失敗しました。失敗した理由は次のどれかです。

- 無効なユーザ名やパスワードが指定されました。
- ユーザ名やパスワードをエンコード（BER-encoding）するときにエラーが発生しました。
- LDAP サーバが要求を受信できなかったまたは LDAP サーバへの接続ができなくなりました。
- メモリが不足しています。
- LDAP サーバから戻された処理結果を取得するときにエラーが発生しました。
- エンコード済みの処理結果をデコードするときにエラーが発生しました。

詳細情報に従い，エラーに対応してください。

エラーレベル：error

(S) ユーザ情報が LDAP サーバ内で見つからない場合はステータスコード「401 Authorization Required」で処理を中断します。それ以外のエラーならば，ステータスコード「500　Internal Server Error」で処理を中断します。

(O) 詳細情報に従い，LDAP サーバに登録されているユーザ情報と入力したユーザ情報に問題がないかどうか確認してください。

[client **クライアントアドレス**] An attempt to initialize the LDAP server session failed.

LDAP サーバとのセッションの初期化に失敗しました。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」を返して処理を中断します。

(O)LDAPServerName/LDAPServerPort ディレクティブに誤りがないかどうか確認してください。

[client **クライアントアドレス**] An attempt to terminate the LDAP server session failed.

LDAP サーバとのセションの終了に失敗しました。

エラーレベル：error

(S) 処理を続行します。

(O)LDAP サーバをアクセスするための資源の解放を失敗しているため，多発するようであれば，Hitachi Web Server を再起動してください。

[client **クライアントアドレス**] An LDAP search for user **ユーザ** ID failed: **詳細情報**

検索フィルタに従いユーザの情報を検索しましたが，エラーが発生して検索を完了できませんでした。エラーの詳細を**詳細情報**に示します。失敗する原因として次に示す原因が考えられます。

- 無効な検索フィルタが指定されました。
- 検索フィルタをエンコードするときにエラーが発生しました。
- LDAP サーバとの通信路に問題が発生しました。
- メモリの割り当てに失敗しました。
- LDAP サーバから戻された処理結果にエラーが見つかりました。
- 検索フィルタで検索したエンコード済みの処理結果をデコードできませんでした。
- LDAP のプロトコルバージョンが異なります。
- 検索フィルタを解析およびエンコード時にエラーが発生しました。
- 制限時間を検索処理が超過しました。
- LDAP サーバ内でユーザ情報を見つけられませんでした。

上記以外にもエラーの原因があることがありますので**詳細情報**について見直してください。

エラーレベル：error

(S) ユーザ情報が LDAP サーバ内で見つからない場合は，ステータスコード「401 Authorization Required」で処理を中断します。それ以外のエラーならば，ステータスコード「500　Internal Server Error」で処理を中断します。

(O) 詳細情報に示す原因について見直してください。

[client **クライアントアドレス**] Authentication failed

パスワードが入力されていない，ユーザ名が 255 文字を超えているまたはユーザ名の取得に失敗しました。

エラーレベル：error

(S) ステータスコード「401　Authorization Required」で処理を中断します。

(O) ユーザ名とパスワードを正しく入力してください。

[client **クライアントアドレス**] The first entry could not be acquired.

認証に成功しましたが，エントリの取得に失敗しました。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」で処理を中断します。

(O)LDAP サーバのスキーマに問題がないかどうか見直してください。

[client **クライアントアドレス**] The invalid DN DN **値** was found in LDAPBaseDN.

無効な DN を LDAPBaseDN ディレクティブに指定しています。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」で処理を中断します。

(O)LDAPServerName ディレクティブ，LDAPServerPort ディレクティブとの対応を見直してください。

[client **クライアントアドレス**] The invalid LDAP server LDAP **サーバ名** was found in LDAPServerName.

無効な LDAP サーバ名を LDAPServerName ディレクティブに指定しています。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」で処理を中断します。

(O)LDAPServerPort ディレクティブ，LDAPBaseDN ディレクティブとの対応を見直してください。

[client **クライアントアドレス**] The invalid port number **ポート番号** was found in LDAPServerPort.

無効なポート番号を LDAPServerPort ディレクティブに指定しています。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」で処理を中断します。

(O)LDAPServerName ディレクティブ，LDAPBaseDN ディレクティブとの対応を見直してください。

[client **クライアントアドレス**] The session option **オプション** could not be set.

セションのオプション設定に失敗しました。

- オプションが DEREF の場合
  LDAPRequire ディレクティブの検索フィルタの処理中に別名の機能を使わないように設定しましたが，失敗しました。
- オプションが PROTOCOL_VERSION の場合
  LDAP サーバにアクセスするためのプロトコルバージョンの設定に失敗しました。
- オプションが REFERRALS の場合
  クライアントがリフェラルに従わないように設定しましたが，失敗しました。

エラーレベル：error

(S) ステータスコード「500　Internal Server Error」で処理を中断します。

(O)LDAPServerName ディレクティブで指定した日立ディレクトリサーバ V2 であるか確認してください。

[client **クライアントアドレス**] The user **ユーザ** ID lacks LDAPRequire filter permission.

LDAPRequire ディレクティブで指定された検索フィルタに一致するものはありませんでした。

エラーレベル：error

(S)LDAPNoEntryStatus ディレクティブに Authorization を指定した場合は，ステータスコード「401　Authorization Required」で処理を中断します。LDAPNoEntryStatus ディレクティブに Forbidden を指定した場合は，ステータスコード「403　Forbidden」で処理を中断します。

(O) 問題があるならば，検索フィルタを再度見直してください。

[client **クライアントアドレス**] The value of attribute **属性** could not be acquired: **詳細情報**

属性値の取得に失敗しました。

エラーレベル：error

(S) 処理を続行します。文字列値の取得に失敗した属性を LDAPSetEnv ディレクティブに指定している場合は，その属性値を設定しようとした環境変数は設定されません。

(O) **詳細情報**に示す原因について見直してください。

LDAP: A separator (%) could not be found.

LDAPRequire ディレクティブに指定された DN 属性のセパレータ (%) がありません。

エラーレベル：なし

(S)Hitachi Web Server の起動を中止します。

(O)DN 属性を正しく指定して Hitachi Web Server を再起動してください。

LDAP: No DN was specified for LDAPBaseDN.

LDAPBaseDN ディレクティブに DN 値が指定されていません。

エラーレベル：なし

(S)Hitachi Web Server の起動を中止します。

(O)DN 値を正しく指定して Hitachi Web Server を再起動してください。

LDAP: No IP address or hostname was specified for LDAPServerName.

LDAPServerName ディレクティブにホスト名または IP アドレスが指定されていません。

エラーレベル：なし

(S)Hitachi Web Server の起動を中止します。

(O)LDAPServerName ディレクティブを正しく指定して Hitachi Web Server を再起動してください。

LDAP: No port number was specified for LDAPServerPort.

LDAPServerPort ディレクティブにポート番号が指定されていません。

エラーレベル：なし

(S)Hitachi Web Server の起動を中止します。

(O)LDAPServerPort ディレクティブを正しく指定して Hitachi Web Server を再起動してください。

LDAP: The specified DN is invalid.

LDAPBaseDN ディレクティブに指定された値が正しくありません。

エラーレベル：なし

(S) Hitachi Web Server の起動を中止します。

(O)LDAPBaseDN ディレクティブを正しく指定して Hitachi Web Server を再起動してください。

LDAP: The specified IP address is invalid.

LDAPServerName ディレクティブに指定された IP アドレスが不正です。

エラーレベル：なし

(S) Hitachi Web Server の起動を中止します。

(O)IP アドレスを正しく指定して Hitachi Web Server を再起動してください。

LDAP: The specified LDAPRequire value is invalid.

LDAPRequire ディレクティブに指定された値が正しくありません。

エラーレベル：なし

(S) Hitachi Web Server の起動を中止します。

(O)LDAPRequire ディレクティブを正しく指定して Hitachi Web Server を再起動してください。

LDAP: The specified LDAPTimeout exceeds the maximum limit, and has been changed to 86400 seconds.

LDAPTimeout ディレクティブに指定された検索処理の最大待ち時間（秒単位）が最大値を超えているため，86400 秒を設定します。

エラーレベル：warn

(S) 処理を続行します。

(O) フィルタの検索処理の最大待ち時間について見直してください。

LDAP: The specified port number is invalid.

LDAPServerPort ディレクティブに指定されたポート番号に数字以外の文字が含まれています。

エラーレベル：なし

(S) Hitachi Web Server の起動を中止します。

(O) ポート番号を正しく指定して Hitachi Web Server を再起動してください。

LDAP: The specified port number is out of range (range = 1 to 65535).

LDAPServerPort ディレクティブに指定されたポート番号が範囲外です。

エラーレベル：なし

(S) Hitachi Web Server の起動を中止します。

(O) ポート番号を正しく指定して Hitachi Web Server を再起動してください。

LDAP: The specified timeout is invalid.

LDAPTimeout ディレクティブに指定されたフィルタの検索処理の最大待ち時間（秒単位）に数字以外の文字が含まれています。

エラーレベル：なし

(S) Hitachi Web Server の起動を中止します。

(O) フィルタの検索処理の最大待ち時間を正しく指定して Hitachi Web Server を再起動してください。

LDAP: The specified timeout is out of range (range = 1 to 86400).

LDAPTimeout ディレクティブに指定された検索処理の最大待ち時間（秒単位）が範囲外です。

エラーレベル：なし

(S) Hitachi Web Server の起動を中止します。

(O) フィルタの検索処理の最大待ち時間を正しく指定して Hitachi Web Server を再起動してください。

LDAP: The status code specified in LDAPNoEntryStatus is invalid.

LDAPNoEntryStatus ディレクティブに指定された値が正しくありません。

エラーレベル：なし

(S) Hitachi Web Server の起動を中止します。

(O)LDAPNoEntryStatus ディレクティブを正しく指定して Hitachi Web Server を再起動してください。

7.2.7 ユティリティについてのメッセージ

（1） crldownload ユティリティ

An attempt to download the CRL file failed: **理由**

CRL ファイルのダウンロードに失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) 理由に沿った対応をしてください。

An attempt to download the CRL file **ファイル名** failed.

CRL のダウンロードに失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) 同時に出力されるほかのエラーメッセージを参照してください。

An attempt to initialize the ldap server session failed.

LDAP サーバとのセションの初期化に失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O)LDAP サーバ名やポート番号に誤りがないかどうか確認してください。

Could not locate **ファイル名** API: **理由**

LDAP ライブラリの処理を失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) 理由に沿った対応をしてください。

---

invalid option.

オプションが不正です。

エラーレベル：なし

(S) 処理を中断します。

(O) オプションを見直してください。

Memory could not be allocated.

メモリの確保ができません。

エラーレベル：なし

(S) 処理を中断します。

(O) メモリの使用状態を見直してください。

The CRL file **ファイル名** could not be opened: **詳細情報**

設定された CRL ファイルが開けません。

エラーレベル：なし

(S) 処理を中断します。

(O) **詳細情報**に示す原因について対応してください。

The CRL **ファイル名** was downloaded successfully.

CRL のダウンロードに成功しました。

エラーレベル：なし

(S) 処理を続行します。

The library file **ライブラリファイル名** could not be opened.

LDAP ライブラリが読み込めません。

エラーレベル：なし

(S) 処理を中断します。

(O) ライブラリがあるかどうかおよびディレクトリに読み出し許可が設定されているかどうかを見直してください。

The port number is out of range.

ポート番号の指定値が，指定できない値（1-65535 以外）です。

エラーレベル：なし

(S) 処理を中断します。

(O)-p オプションの指定値を見直してください。

The **オプション** option was not specified.

必要なオプション（-L，-b および -o）が指定されていません。

エラーレベル：なし

(S) 処理を中断します。

(O)-L，-b および -o オプションを指定してください。

（2） hwsserveredit ユティリティ

---

hwsserveredit: completed

ユティリティの処理が完了しました。または，-check 実行時，サーバ環境は構築されていました。

エラーレベル：なし

(S) 処理を終了します。

---

hwsserveredit: cannot {create|delete|open|read|write|close} **ファイル名 関数名** : **詳細情報**

ファイルの操作に失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) **詳細情報**に示す内容について見直してください。

---

hwsserveredit: malloc failed

メモリの確保に失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) メモリ使用量について，システムの状態を確認してください。

---

hwsserveredit: Service {add|delete} failed **関数名** : **詳細情報**

サービスの登録または削除処理に失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) 詳細情報に示す内容について見直してください。

---

hwsserveredit: uncompleted

-check 実行時，サーバ環境は構築されていませんでした。

エラーレベル：なし

(S) 処理を終了します。

(O) サーバ名に対するリソースが作成されているか確認してください。

詳細情報 : **関数名 レジストリキー**

レジストリに対する操作で失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) 詳細情報に示す内容について見直してください。

Registry does not contain key **レジストリキー** after creation

レジストリのオープンに失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) レジストリキーが存在するか確認してください。

詳細情報 : GetModuleFileName failed

実行ファイルのパス取得に失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) GetModuleFileName() 関数が返す**詳細情報**について見直してください。

詳細情報 : An attempt to load the audit log library has failed.

監査ログ出力用ライブラリのロードに失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : An attempt to acquire the address of the audit log function has failed.

監査ログ出力用ライブラリの関数のアドレス取得に失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) 詳細情報について見直してください。

詳細情報 : An attempt to acquire the path of the audit log library has failed.

監査ログ出力用ライブラリのパスの取得に失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) 詳細情報について見直してください。

(3) hwstraceinfo ユティリティ

---

hwstraceinfo: An open logfile error occurred.

出力ファイルを開けません。

エラーレベル：なし

(S) 処理を中断します。

(O) 出力ファイルについて見直してください。

---

hwstraceinfo: An open shared memory error(**詳細情報**) occurred.

共有メモリの参照に失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) 詳細情報に示す内容について見直してください。

---

hwstraceinfo: A map shared memory error(**詳細情報**) occurred.

共有メモリのマッピングに失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) **詳細情報**に示す内容について見直してください。

---

hwstraceinfo: A shmat error occurred.

共有メモリの割り当てができません。

エラーレベル：なし

(S) 処理を中断します。

(O) 共有メモリ識別子について見直してください。

---

hwstraceinfo: A shmctl ID removal error occurred.

共有メモリ識別子を削除できません。

エラーレベル：なし

(S) 処理を中断します。

(O) 共有メモリ識別子について見直してください。

---

hwstraceinfo: A write error occurred.

共有メモリの内容をファイル出力することに失敗しました。

エラーレベル：なし

(S) 処理を中断します。

(O) 出力ファイルについて見直してください。

---

hwstraceinfo: The shmid removal completed.

共有メモリ識別子を削除しました。

エラーレベル：なし

(S)hwstraceinfo コマンド処理を終了します。

---

hwstraceinfo: The trace output completed.

共有メモリの内容をファイルに出力しました。

エラーレベル：なし

(S)hwstraceinfo コマンド処理を終了します。

(4) rotatelogs ユティリティ

Rotation time must be > 0

ログ分割時間間隔が不正です。

エラーレベル：なし

(S) 処理を中断します。

(O) ログ分割時間の間隔について見直してください。

The number of files must be >= 1 and <=256

-fnum に指定した値が不正です。

エラーレベル：なし

(S) 処理を中断します。

(O)-fnum に指定した値について見直してください。

The offset minutes from UTC must be >= -1439 and <=1439

-diff に指定した値が不正です。

エラーレベル：なし

(S) 処理を中断します。

(O)-diff に指定した値について見直してください。

---

file path is too long.

分割ログファイル名の長さが長過ぎます。

エラーレベル：なし

(S) 処理を中断します。

(O) 分割ログファイルのプリフィックス長について見直してください。

(5) rotatelogs2 ユティリティ

The size(KB) of file must be >= 1 and <= 2097151

ログファイルサイズが不正です。

エラーレベル：なし

(S) 処理を中断します。

(O) ログファイルのサイズについて見直してください。

The number of files must be >= 1 and <= 256

ログファイル個数が不正です。

エラーレベル：なし

(S) 処理を中断します。

(O) ログファイルの個数について見直してください。

---

file path is too long.

ログファイル名の長さが長過ぎます。

エラーレベル：なし

(S) 処理を中断します。

(O) ログファイルのプリフィックス長について見直してください。

（6） sslpasswd ユティリティ

Could not create the password file.

パスワードファイルが作成できませんでした。

エラーレベル：なし

(S) 処理を終了します。

(O) 不正なサーバ秘密鍵ファイルを読み込んだまたはパスワードが長過ぎる可能性があります。これらについて確認してください。

詳細情報 : Could not open private key file.

サーバ秘密鍵ファイルがオープンできませんでした。

エラーレベル：なし

(S) 処理を終了します。

(O) **詳細情報**を基に対応してください。

Could not read the appropriate private key file.

サーバ秘密鍵ファイルが適切ではありませんでした。

エラーレベル：なし

(S) 処理を終了します。

(O) パスワード付きのサーバ秘密鍵を指定してください。

詳細情報 : Could not open the password file.

パスワードファイルをオープンできませんでした。

エラーレベル：なし

(S) 処理を終了します。

(O) 詳細情報を基に対応してください。

詳細情報 : Could not write the password file.

パスワードファイルを作成できませんでした。

エラーレベル：なし

(S) 処理を終了します。

(O) 詳細情報を基に対応してください。

付録

付録 A　ステータスコード

Hitachi Web Server が Web ブラウザに返送するステータスコードを次に示します。ステータスコードを Web ブラウザに返送する際には，ステータスコードに応じて自動生成するエラーメッセージを charset=ISO-8859-1 の HTML として同時に返送します。

表 A-1　ステータスコード一覧

ステータスコード	内容
100 Continue	クライアントは，リクエストを継続可能です。
200 OK	正常に終了しました。
204 No Content	リクエストは正常に終了しましたが，返すリソースはありません。ImapDefault nocontent ディレクティブの指定によって，発生します。
206 Partial Content	部分的なリソースを返します。 クライアントの Range ヘッダを用いた Partial GET リクエストの応答として，部分的なコンテンツを返す場合に発生します。
300 Multiple Choices	複数ページの利用が可能です。
301 Moved Permanently	リソースが恒久的に移動しました。 最後をスラッシュで閉じないディレクトリに対するリクエスト http:// **ホスト名**〔: **ポート番号**〕/ **ディレクトリ名**や，Redirect permanent ディレクティブの指定によって，発生します。
302 Found	リソースが一時的に移動しました。 Redirect temp ディレクティブの指定によって，発生します。
303 See Other	リソースが移動しました。 Redirect seeother ディレクティブの指定によって，発生します。
304 Not Modified	リクエストしたコンテンツが変更されていません。
400 Bad Request	リクエストにシンタックスエラーがあります。 ヘッダとして誤ったものを指定した場合，リクエストヘッダの個数が LimitRequestFields ディレクティブの値を超えた場合または HTTP/1.1 で Host ヘッダがなかった場合などに発生します。
401 Authorization Required	リソースにアクセスするためには，認証が必要です。AuthName ディレクティブまたは AuthUserFile ディレクティブなどでアクセスを制御した場合に発生します。
403 Forbidden	リソースへのアクセスが禁じられています。 アクセス制御によって，アクセスが拒否された場合または実行権限のない CGI プログラムの実行要求をした場合などに発生します。
404 Not Found	リソースが見つかりません。 サーバ上にはないファイルをリクエストした場合などに発生します。
405 Method Not Allowed	許可されていないメソッドを使用しました。
406 Not Acceptable	クライアントが Accept ヘッダで指定したタイプに応じたレスポンスを返せません。

ステータスコード	内容
408 Request Time-out	リクエストがタイムアウトになりました。
410 Gone	リソースが恒久的に利用できません。 Redirect gone ディレクティブの指定によって，発生します。
411 Length Required	クライアントは Content-Length ヘッダを指定する必要があります。
412 Precondition Failed	クライアントの If-Unmodified-Since ヘッダまたは If-Matched ヘッダなどで指定した条件が一致しません。
413 Request Entity Too Large	リクエストボディサイズが大き過ぎて，サーバで処理できません。 リクエストボディの長さが，LimitRequestBody ディレクティブで指定した長さよりも長い場合に発生します。
414 Request-URI Too Large	リクエスト URI が大き過ぎて，サーバで処理できません。 問い合わせ文字列などを含む URI などの長さが， LimitRequestLine ディレクティブで指定した長さよりも長い場合に発生します。
416 Requested Range Not Satisfiable	Range ヘッダでの指定範囲は，該当リソースの範囲を超えています。次の条件がすべて成立する場合に出力されます。 • リクエストが Range ヘッダフィールドを含む。 • フィールドの範囲指定値が，選ばれたリソースの現在の範囲に重なっていない。 • リクエストに If-Range リクエストヘッダフィールドを含んでいない。
417 Expectation Failed	Expect リクエストヘッダフィールドの拡張が受け入れられませんでした。
500 Internal Server Error	Web サーバ上でエラーが発生しました。 CGI プログラムの問題や，アクセス制御ファイル (.htaccess) のエラーなどの場合に発生します。詳細な情報は，エラーログに出力されます。
501 Method Not Implemented	サポートされていないメソッドの要求です。
502 Bad Gateway	プロキシサーバが不正な要求を受け取りました。
503 Service Temporarily Unavailable	サーバは過負荷状態であるため，現在リクエスト処理できません。
506 Variant Also Negotiates	サーバに内部配置上のエラーがあります。

注

表 A-1 および表 A-1 以外のステータスコードが Hitachi Web Server と連携した CGI プログラムなどの上位プログラムから出力されることがあります。その場合は，それぞれのプログラムのマニュアルを参照してください。

リバースプロキシを使用している場合には，400 Bad Request，403 Forbidden，502 Bad Gateway は，400 Proxy Error，403 Proxy Error，502 Proxy Error となる場合もあります。

付録 B　CGI プログラムに渡す環境変数

Web サーバが CGI プログラムに渡す環境変数の一覧を表 B-1，表 B-2 および表 B-5 に，SSL_SERVER_ 要素の例，SSL_SERVER_I_ 要素の例を，表 B-3，表 B-4 に示します。プラットフォーム，クライアントの設定，リクエストの形，Web サーバのディレクティブの設定などによって，ここで記載されている環境変数が設定されない場合や，記載していない環境変数が設定される場合もあります。表の中のサーバ名，ドメイン名，メールアドレスなどはすべて架空の値です。

表 B-1　環境変数一覧

環境変数名	内容	例
AUTH_TYPE	ユーザ認証をする場合の認証タイプ	Basic
COMSPEC	コマンドプロンプトの実行可能ファイル	C:¥WINNT¥system32¥cmd.exe
CONTENT_LENGTH	クライアントからのリクエストが POST の場合の，データのバイト数	20
CONTENT_TYPE	クライアントからのリクエストが POST の場合のコンテンツタイプ	application/ x-www-form-urlencoded
DOCUMENT_ROOT	DocumentRoot ディレクティブ指定値	C:/Program Files/Hitachi/ httpsd/htdocs
GATEWAY_INTERFACE	CGI バージョン	CGI/1.1
HTTP_ACCEPT	クライアントが示した Accept ヘッダの値	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
HTTP_ACCEPT_CHARSET	クライアントが示した Accept-Charset ヘッダの値	Shift_JIS,*,utf-8
HTTP_ACCEPT_ENCODING	クライアントが示した Accept-Encoding ヘッダの値	gzip
HTTP_ACCEPT_LANGUAGE	クライアントが示した Accept-Language ヘッダの値	ja,fr, en,it
HTTP_CONNECTION	クライアントが示した Connection ヘッダの値	Keep-Alive
HTTP_HOST	クライアントが示した Host ヘッダの値	www.hws.hitachi.co.jp:8080
HTTP_PRAGMA	クライアントが示した Pragma ヘッダの値	no-cache
HTTP_REFERER	クライアントが示した Referer ヘッダの値	http:// www.hws.hitachi.co.jp:8080/ test.html

環境変数名	内容	例
HTTP_USER_AGENT	クライアントが示した User-Agent ヘッダの値	Mozilla/4.73 [ja] (WinNT; U)
PATH	Web サーバ上の PATH 情報	C:¥WINNT¥system32;C:¥WINNT;C:¥WINNT¥System32¥Wbem
PATH_INFO	URL のうち CGI スクリプトより後ろの部分	/dir1/file1
PATH_TRANSLATED	ファイルシステムに変換された PATH_INFO の値	C:¥Program Files¥Hitachi¥httpsd¥htdocs¥dir1¥file1
QUERY_STRING	クライアントから送信された Query String（問い合わせ文字列）	query1=a&query2=b
REMOTE_ADDR	クライアントのアドレス	172.17.xx.xx
REMOTE_HOST	クライアントのホスト名（HostnameLookups が Off 以外でホスト名が解決された場合）	hostxxx
REMOTE_IDENT	クライアントの ID（IdentityCheck ディレクティブ参照）	unknown
REMOTE_PORT	クライアントのポート番号	2298
REMOTE_USER	認証されたリクエストの場合の認証ユーザ名	Userxxx
REQUEST_METHOD	クライアントから送信された HTTP メソッド	GET
REQUEST_URI	クライアントから送信されたリクエスト URI	/cgi-bin/test-cgi?query1=a&query2=b
SCRIPT_FILENAME	リクエストされた CGI スクリプトのファイル名	C:/Program Files/Hitachi/httpsd/cgi-bin/test-cgi
SCRIPT_NAME	リクエストされた CGI スクリプトの URI	/cgi-bin/test-cgi
SERVER_ADDR	Web サーバの IP アドレス	172.17.xx.xx
SERVER_ADMIN	ServerAdmin ディレクティブ指定値	www-admin@server.example.com
SERVER_NAME	Web サーバのホスト名（UseCanonicalName ディレクティブ参照）	www.hws.hitachi.co.jp
SERVER_PORT	Web サーバのポート名（UseCanonicalName ディレクティブ参照）	8080
SERVER_PROTOCOL	クライアントが示した HTTP バージョン	HTTP/1.0

環境変数名	内容	例
SERVER_SIGNATURE	Web サーバの署名（HTML タグを含む）（ServerSignature ディレクティブ参照）	<ADDRESS>Hitachi Web Server 03-00 at www.example.com Port 8080</ADDRESS>
SERVER_SOFTWARE	Web サーバのプログラム名	Hitachi Web Server 03-00
SYSTEMROOT	システムディレクトリ	C:¥WINNT
TZ	Web サーバのタイムゾーン	JST-9
WINDIR	システムディレクトリ	C:¥WINNT

表 B-2　SSL 通信時の環境変数一覧

環境変数名	内容	例
HTTPS	セキュア通信を示します。	on
HTTPS_CIPHER	SSL 暗号種別	RC4-MD5
HTTPS_KEYSIZE	対称鍵暗号の鍵のビット数	128
HTTPS_SECRETKEYSIZE	対称鍵暗号の鍵のビット数のうち，有効なビット数	128
SSL_CIPHER	SSL 暗号種別（HTTPS_CIPHER と同じ）	RC4-MD5
SSL_PROTOCOL_VERSION	SSL プロトコルバージョン	SSLv3
SSL_SERVER_DN	SSL サーバ証明書の subject の Distinguish Name	/C=JP/ST=Kanagawa/ L=Yokohama-shi/O=HITACHI/ OU=WebSite/ CN=www.hws.hitachi.co.jp/ Email=www-admin@hws.hitachi.co.jp
SSL_SERVER_ **要素**	SSL サーバ証明書の subject の Distinguish Name の各要素	SSL_SERVER_DN が上記の例の場合を表 B-3 に示します。
SSL_SERVER_I_DN	SSL サーバ証明書の issuer の Distinguish Name	/C=JP/ST=Kanagawa/ L=Yokohama-shi/ O=LOCAL-CA/OU=ca1/ CN=ca1.hitachi.co.jp/ Email=ca-admin@ca1.hitachi.co.jp
SSL_SERVER_I_ **要素**	SSL サーバ証明書の issuer の Distinguish Name の各要素	SSL_SERVER__I_DN が上記の例の場合を表 B-4 に示します。
SSL_SESSION_ID	SSL セション ID（16 進数）	F968F8D7075B76587F35931DC594D3E3
SSL_SSLC_VERSION	SSLC のバージョン	SSL-C 2.7.0.1 15-Mar-2006

表 B-3　SSL_SERVER_ 要素の例

環境変数名	内容	例
SSL_SERVER_C	SSL サーバ証明書の subject（Web サーバ）の Country Name	JP
SSL_SERVER_CN	SSL サーバ証明書の subject の Common Name	www.hws.hitachi.co.jp
SSL_SERVER_EMAIL	SSL サーバ証明書の subject の E-Mail アドレス	www-admin@hws.hitachi.co.jp
SSL_SERVER_L	SSL サーバ証明書の subject の Locality Name	Yokohama-shi
SSL_SERVER_O	SSL サーバ証明書の subject の Organization Name	HITACHI,Ltd.
SSL_SERVER_OU	SSL サーバ証明書の subject の Organization Unit Name	WebSite
SSL_SERVER_ST	SSL サーバ証明書の subject の State Name	Kanagawa

表 B-4　SSL_SERVER_I_ 要素の例

環境変数名	内容	例
SSL_SERVER_I_C	SSL サーバ証明書の issuer（発行者）の Country Name	JP
SSL_SERVER_I_CN	SSL サーバ証明書の issuer の Common Name	ca1.hitachi.co.jp
SSL_SERVER_I_EMAIL	SSL サーバ証明書の issuer の E-Mail アドレス	ca-admin@ca1.hitachi.co.jp
SSL_SERVER_I_L	SSL サーバ証明書の issuer の Locality Name	Yokohama-shi
SSL_SERVER_I_O	SSL サーバ証明書の issuer の Organization Name	LOCAL-CA
SSL_SERVER_I_OU	SSL サーバ証明書の issuer の Organization Unit Name	ca1
SSL_SERVER_I_ST	SSL サーバ証明書の issuer の State Name	Kanagawa

表 B-5　SSL クライアント認証時の環境変数一覧

環境変数名	内容	例
SSL_CLIENT_CERT	SSL クライアント証明書（DER-BASE64 形式）SSLExportClientCertificates ディレクティブの設定が必要です。	"MIIDrTCCAxagAwIBAgIBAjANBgkqhkiG9w0BAQQFADCBlzELMAkGA1UEBhMCSlAx..."
SSL_CLIENT_CERT_n	SSL クライアント証明書を発行した CA からルート CA までの CA 証明書（n はチェーン数を示す正の整数）（DER-BASE64 形式）SSLExportCertChainDepth ディレクティブの設定が必要です。	"MIIDrTCCAxagAwIBAgIBAjANBgkqhkiG9w0BAQQFADCBlzELMAkGA1UEBhMCSlAx..."
SSL_CLIENT_DN	SSL クライアント証明書の subject の Distinguish Name	/C=JP/ST=Kanagawa/L=Yokohama/O=Hitachi/OU=soft/CN=c_name/Email=c_name@soft.hitachi.co.jp
SSL_CLIENT_ **要素**	SSL クライアント証明書の subject の Distinguish Name の各要素	SSL_CLIENT_DN が上記の例の場合を表 B-6 に示します。
SSL_CLIENT_I_DN	SSL クライアント証明書の issuer の Distinguish Name	/C=JP/ST=Kanagawa/L=Yokohama-shi/O=LOCAL-CA/OU=ca1/CN=ca1.hitachi.co.jp/Email=ca-admin@ca1.hitachi.co.jp
SSL_CLIENT_I_ **要素**	SSL クライアント証明書の issuer の Distinguish Name の各要素	SSL_CLIENT_I_DN が上記の例の場合を表 B-7 に示します。

表 B-6　SSL_CLIENT_ 要素の例

環境変数名	内容	例
SSL_CLIENT_C	SSL クライアント証明書の subject の Country Name	JP
SSL_CLIENT_CN	SSL クライアント証明書の subject の Common Name	c_name
SSL_CLIENT_EMAIL	SSL クライアント証明書の subject の E-Mail アドレス	c_name@soft.hitachi.co.jp
SSL_CLIENT_L	SSL クライアント証明書の subject の Locality Name	Yokohama
SSL_CLIENT_O	SSL クライアント証明書の subject の Organization Name	Hitachi

環境変数名	内容	例
SSL_CLIENT_OU	SSL クライアント証明書の subject の Organization Unit Name	soft
SSL_CLIENT_ST	SSL クライアント証明書の subject の State Name	Kanagawa

表 B-7　SSL_CLIENT_I_ 要素の例

環境変数名	内容	例
SSL_CLIENT_I_C	SSL クライアント証明書の issuer の Country Name	JP
SSL_CLIENT_I_CN	SSL クライアント証明書の issuer の Common Name	ca1.hitachi.co.jp
SSL_CLIENT_I_EMAIL	SSL クライアント証明書の issuer の E-Mail アドレス	ca-admin@ca1.hitachi.co.jp
SSL_CLIENT_I_L	SSL クライアント証明書の issuer の Locality Name	Yokohama-shi
SSL_CLIENT_I_O	SSL クライアント証明書の issuer の Organization Name	LOCAL-CA
SSL_CLIENT_I_OU	SSL クライアント証明書の issuer の Organization Unit Name	ca1
SSL_CLIENT_I_ST	SSL クライアント証明書の issuer の State Name	Kanagawa

付録 C　高信頼化システム監視機能 HA モニタによるシステム監視（クラスタリングシステムの運用）

高信頼化システム監視機能 HA モニタは，システムの信頼性，稼働率の向上を目的として，サーバプログラム（以降，サーバと略します）を含めたシステムの切り替えを実現するソフトウェアです。

Hitachi Web Server は，HA モニタを使用したクラスタリングシステムで運用できます。なお，HA モニタの詳細については，マニュアル「高信頼化システム監視機能　HA モニタ」を参照してください。また，Hitachi Web Server の前提プログラム（OS など）や関連プログラム（CGI プログラムなど）をクラスタリングシステム構成で運用する場合の詳細については，各プログラムのマニュアルを参照してください。

HA モニタを使用すると，ハードウェア障害およびソフトウェア異常停止によるコンテンツ配信の停止時間を最小限にとどめた Web サーバ運用ができます。また，サービスを停止しないで，ソフトウェアの管理，保守およびバージョンアップができます。

対象となる主な障害

HA モニタが対象とする障害（HA モニタが検出する障害）は，サーバに発生するサーバ障害と，系に発生する系障害とに分けられます。系とは，業務処理に必要なハードウェアのほか，実行するプログラムや通信機器も含めたシステム全体の総称です。HA モニタの対象となる主な障害には，次のものがあります。

● サーバ障害
- サーバの論理エラー
- リソース（ディスク装置など）の障害

● 系障害
- 系のハードウェア障害または電源断
- カーネルの障害
- HA モニタの障害
- 監視パスの障害
- 系のスローダウン

付録 C.1　ハードウェア構成例と HA モニタの動作概要

HA モニタは，監視対象のシステム（以降，**現用系**と呼ぶ）に障害が発生すると，予備のシステム（以降，**予備系**と呼ぶ）に処理を切り替えて業務処理を続行します。この動作を，系切り替え機能といいます。

次に系切り替えと LAN アダプタを二重化した場合について，それぞれのハードウェア構成例と動作の概要を示します。

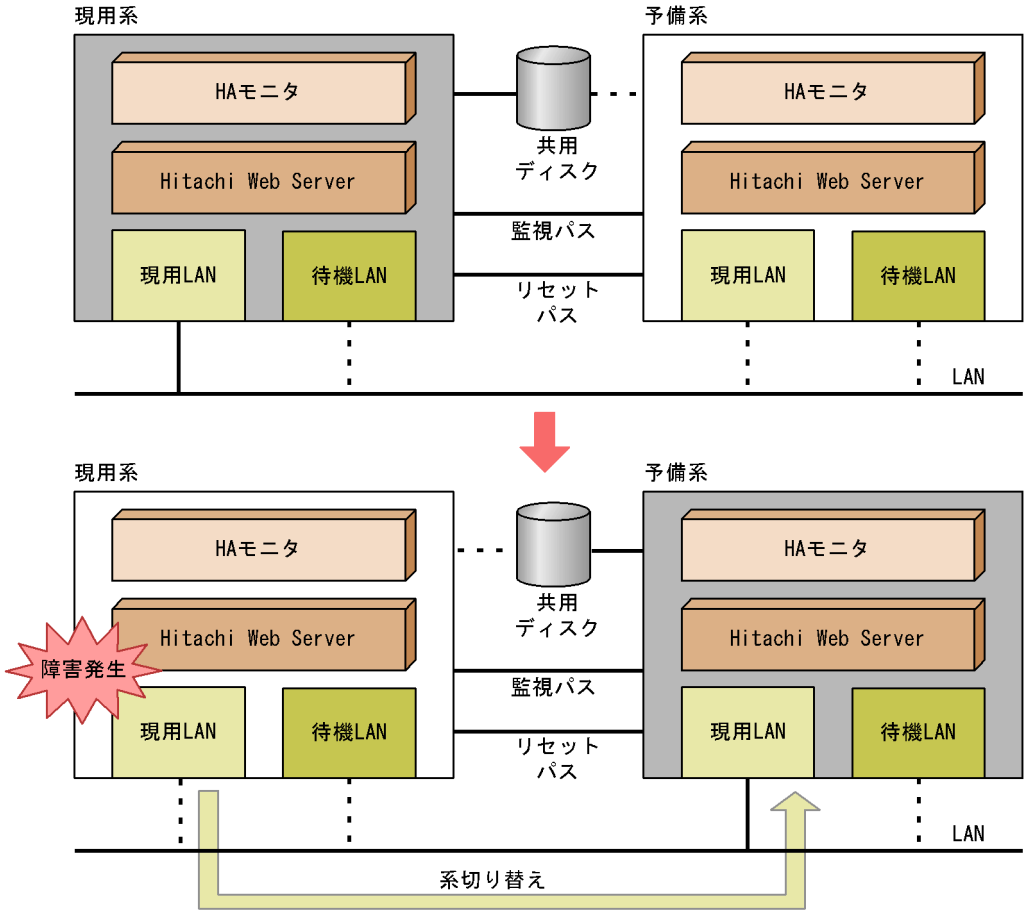
（1） 1：1 系切り替えの構成例

現用系と予備系が 1：1 に対応している構成の例について示します。

現用系と予備系の 2 台のサーバ構成で，サーバがサービスを提供するための LAN，互いの系を監視するための監視パスおよび実行系で障害が発生した場合にリセット指示をするためのリセットパスによって，2 台のシステム間が接続されています。ディスク記憶装置は，ノード間で共有します。

現用系に障害が発生すると，HA モニタは現用系を停止させて系切り替えをします。共有ディスクは，予備系のシステムの方にマウントされます。1：1 系切り替えの構成例を次に示します。

図 C-1　1：1 系切り替えの構成例



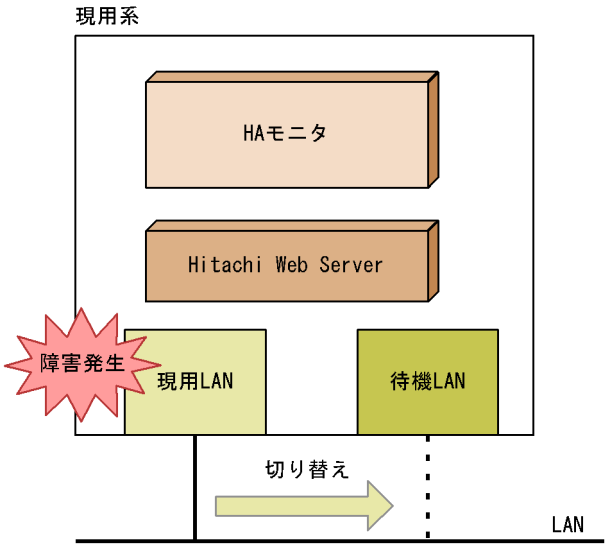
（2） LAN アダプタを二重化した場合の構成例

HP-UX HA モニタでは，系内での LAN アダプタの二重化も制御できます。サーバが使

用する LAN アダプタを現用，予備の組み合わせで二重化として定義することによって，HA モニタは一定間隔で LAN の活性状態を調査し，現用 LAN アダプタが障害になった場合に自動的に予備系の LAN アダプタに切り替えます。

LAN アダプタを二重化した場合の構成例を次に示します。

図 C-2　LAN アダプタを二重化した場合の構成例



付録 C.2　Hitachi Web Server の設定

HA モニタに適用するための，Hitachi Web Server の設定手順を次に示します。

1. Hitachi Web Server を各系のローカルディスクにそれぞれインストールします。
2. Hitachi Web Server のコンフィグファイルを作成し，各系に配布します。

設定に当たっては，次の点に注意してください。

（1）バーチャルホストの場合

系切り替えの結果，クライアントに返送されるサーバ名が変更される場合があります。このため，バーチャルホストでも ServerName ディレクティブは必ず設定してください。

（2）IP アドレスの指定

IP アドレスを指定するディレクティブ（<VirtualHost>，BindAddress，Listen，NameVirtualHost）では，物理 IP アドレスではなく，論理 IP アドレス（エイリアス IP アドレス）を使用してください。

（3）コンフィグファイルの文法チェック

HA モニタの起動の前には，"/opt/hitachi/httpsd/sbin/httpsdctl configtest" を実行し，サーバ設定が正しいことを確認してください。

（4）コンフィグファイルの変更

コマンド "httpsdctl restart" または "httpsdctl graceful" をコマンドラインから直接実行すれば，HA モニタ使用中に Hitachi Web Server の設定を変更できます。変更内容は，ほかの系にも反映する必要があります。

（5）CRL を使用して運用している場合

CRL を用いて運用しているときは，予備系の場合でも，現用系と同様の CRL を設定する必要があります。

付録 C.3　監視コマンドの作成

HA モニタでは，HA モニタとインタフェースを持たないサーバについては，サーバ障害を HA モニタに通知するためのプログラムを HA モニタに登録する必要があります。したがって，HA モニタとのインタフェースを持たない Hitachi Web Server を監視対象とするためには，Hitachi Web Server の動作を監視するコマンドを作成してください。監視対象としない場合はコマンドを作成する必要はありません。

Hitachi Web Server では，実行コマンドと実際にサービスするプロセスが異なります。HA モニタの監視対象とするためには，実際のプロセスを監視するコマンドを作成してください。

次に「Hitachi Web Server に障害が発生して停止したと同時にその実行を終了する」という処理をするスクリプトの例を示します。

（例）

PidFile ディレクティブで指定したファイルに記録されているプロセス ID が実行中かどうかを，5 秒おきに監視するシェルスクリプト（httpsd_monitor）です。

```
#!/bin/sh
#################################################################
### ALL RIGHTS RESERVED. COPYRIGHT (C) 2000, 2002, HITACHI,LTD.
#################################################################
HWSIDFILE=/opt/hitachi/httpsd/logs/httpd.pid
HWSITIME=5

if [ ! -e $HWSIDFILE ]
then
    exit 1
fi

HWSID=`cat $HWSIDFILE`
if [ x$HWSID = "x" ]
then
    exit 1
fi

while true
do
   STATUS=`ps -p $HWSID | grep $HWSID | awk '{print $1}' `
   if [ x$STATUS = "x" ]
   then
     break
   fi
   sleep $HWSITIME
done

exit 0
```

（1）注意事項

Hitachi Web Server では，リクエストを処理するためのプロセス群を制御するプロセスが一つあります（「4.1　Hitachi Web Server の処理とディレクティブとの関係」参照）。例に示したスクリプト httpsd_monitor では，その制御プロセスが動作しているかどうかを監視します。リクエスト処理のためのプロセス群の動作は監視しません。

付録 C.4　HA モニタの設定

Hitachi Web Server と，必要であれば関連プログラムを HA モニタに設定します。ここで説明していない内容および詳細説明については，マニュアル「高信頼化システム監視機能　HA モニタ」を参照してください。

Hitachi Web Server の HA モニタへの設定手順を次に示します。

1. HA モニタの環境設定をします。
2. 必要に応じて，Hitachi Web Server を監視するためのスクリプト，開始スクリプトお

よび停止スクリプトを作成し，各系に配布します。

3. Hitachi Web Server に対応した，HA モニタの環境設定をします。
4. HA モニタを起動し，HA モニタのコマンドで Hitachi Web Server を起動します。

（1）起動および停止スクリプトの作成

HA モニタが Hitachi Web Server を起動および停止するためには，起動および停止スクリプトを作成し登録する必要があります。

(a) 開始スクリプト例

```
#!/bin/sh
/opt/hitachi/httpsd/sbin/httpsdctl start
```

(b) 停止スクリプト例

```
#!/bin/sh
/opt/hitachi/httpsd/sbin/httpsdctl stop
```

（2）Hitachi Web Server に対応した HA モニタの環境設定

HA モニタ環境設定用ディレクトリ下にある，servers というファイルにサーバ対応の環境を設定します。ここで，開始スクリプト，停止スクリプトおよび監視コマンドについて設定します。

(a) 環境設定例

サーバ対応の環境設定例を次に示します。各オペランド，設定の詳細については，マニュアル「高信頼化システム監視機能　HA モニタ」を参照してください。

```
/* サーバ対応環境設定例 (現用系の例) */
      server  name                 /home/work/hws-start.sh, /* 開始スクリプト */
              alias                 HWS,
              acttype               monitor,
              termcommand          /home/work/hws-stop.sh,  /* 停止スクリプト */
              switchtype            switch,
              initial               online,                 /* 現用系の設定   */
              patrolcommand        /home/work/httpsd_monitor, /* 監視コマンド */
              servexec_retry        2,
              waitserv_exec         yes;
```

（3）注意事項

系切り替えが発生した場合，通常の HTTP 接続，SSL を使用した接続はすべて切断され，予備系には引き継がれません。クライアントは，再接続してください。

付録 D　MC/ServiceGuard によるシステム監視（クラスタリングシステムの運用）

MC/ServiceGuard は，クラスタリングシステムを構築する Hewlett-Packard 社のソフトウェアです。Hitachi Web Server は，MC/ServiceGuard を使用したクラスタリングシステムで運用できます。なお，MC/ServiceGuard の詳細については，MC/ServiceGuard のマニュアルを参照してください。また，Hitachi Web Server の前提プログラム（OS など）や関連プログラム（CGI プログラムなど）をクラスタリングシステム構成で運用する場合の詳細については，各プログラムのマニュアルを参照してください。

MC/ServiceGuard を利用すると，ハードウェア障害およびソフトウェア異常停止によるコンテンツ配信の停止時間を最小限にとどめた Web サーバ運用ができます。また，サービスを停止しないで，ソフトウェアの管理，保守およびバージョンアップもできます。

対象となる主な障害

MC/ServiceGuard の対象となる主な障害には，次のものがあります。

- LAN 障害
- リソース（システム演算処理装置，ディスク，インタフェース）障害
- ソフトウェアの異常停止

付録 D.1　ハードウェア構成例と MC/ServiceGuard の動作概要

MC/ServiceGuard は，監視対象のシステム（以降，**1 次系**とする）に障害が発生すると，予備のシステム（以降，**待機系**とする）に処理を切り替えてサービスを続行します。この動作を，フェイルオーバといいます。ローカルノードでの運用と複数ノードでの運用で，フェイルオーバの動作が異なります。

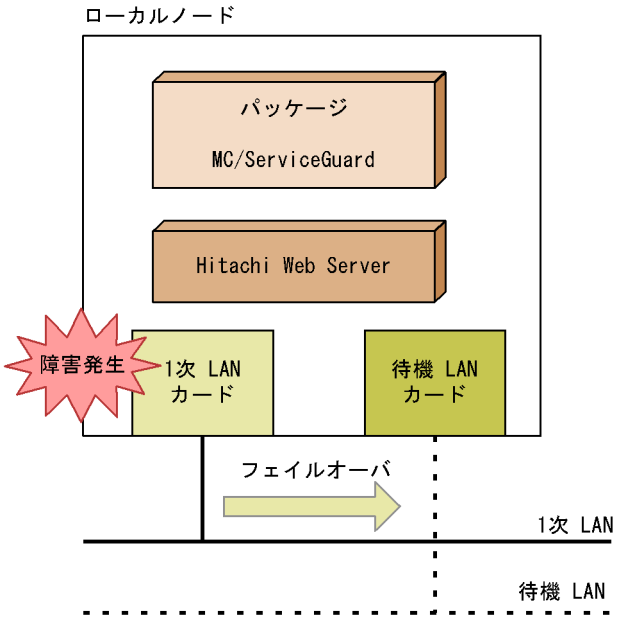
それぞれのハードウェア構成例とフェイルオーバの概要を次に示します。

（1）ローカルノードの運用の例

LAN は二重化していて，一方を 1 次 LAN，他方を待機 LAN としているシステムで，双方の LAN に対応する LAN カードを接続している場合の例を示します。

この場合，1 次系 LAN カードに障害が発生すると，同一ノード上の待機系 LAN カードに接続が切り替えられます。ローカルノードの運用の例を次に示します。

図 D-1　ローカルノードの運用の例



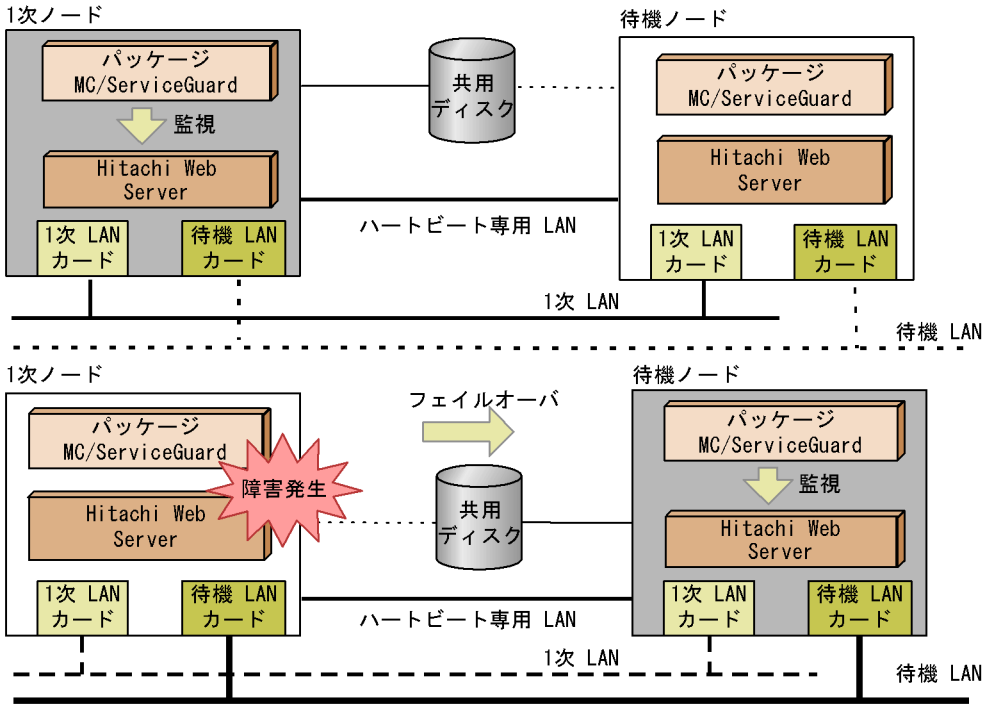
（2） 複数のノードの運用の例

LAN を二重化した場合の例について示します。

一方を 1 次 LAN，他方を待機 LAN としているシステムで，双方の LAN に対応する LAN カードをそれぞれのノードに接続します。さらに，ハートビート専用 LAN または RS232 通信を使用したハートビートラインを設け，1 次 LAN と合わせハートビート用の通信線を 2 重構成とします。ディスク記憶装置は，ノード間で共有します。

1 次ノードに障害が発生し，他ノードへのフェイルオーバが必要だと判断すると，1 次系システムを停止させ，待機系システム上で再度パッケージを起動してサービスを継続させます。共有ディスクは，待機系システムの方にマウントされます。複数ノードの運用の例を次に示します。

図 D-2　複数ノードの運用の例（MC/ServiceGuard）



付録 D.2　Hitachi Web Server の設定

MC/ServiceGuard に適用するための，Hitachi Web Server の設定手順を次に示します。

1. Hitachi Web Server を各ノードのローカルディスクにそれぞれインストールします。
2. 必要に応じて，Hitachi Web Server を監視するためのスクリプトを作成します。
3. MC/ServiceGuard の設定をします。
4. Hitachi Web Server のコンフィグファイル，作成した監視スクリプトおよび MC/ServiceGuard のパッケージ制御スクリプトを各ノードに配布します。
5. MC/ServiceGuard を起動します。

環境設定については，次の点に注意してください。

(1) バーチャルホストの場合

フェイルオーバの結果，クライアントに返送されるサーバ名が変化する場合があります。このため，バーチャルホストでも ServerName ディレクティブは必ず設定してください。

(2) IP アドレスの指定

IP アドレスを指定するディレクティブ（<VirtualHost>，BindAddress，Listen，NameVirtualHost）では，定常 IP アドレス（他ノードに移動できない IP アドレス）で

はなく，再配置できる IP アドレス（パッケージに対して与えられ，他ノードに移動できる IP アドレス）を使用してください。

（3）コンフィグファイルの文法チェック

MC/ServiceGuard の起動の前には，"/opt/hitachi/httpsd/sbin/httpsdctl configtest" を実行し，サーバ設定が正しいことを確認してください。

（4）コンフィグファイルの変更

コマンド "httpsdctl restart" または "httpsdctl graceful" をコマンドラインから直接実行すれば，MC/ServiceGuard 使用中に Hitachi Web Server の設定を変更できます。変更内容は，ほかのノードにも反映する必要があります。

（5）CRL を使用して運用している場合

待機ノードにも，1 次ノードと同様の CRL を設定する必要があります。

付録 D.3　監視スクリプトの作成

MC/ServiceGuard では，ソフトウェアを監視対象とするためには，各実行コマンドが実際のサービスの名前であることと，そのプロセスが実際にサービスが終了するまで動作している必要があります。

Hitachi Web Server では，実行コマンドと実際にサービスするプロセスが異なります。MC/ServiceGuard の監視対象とするためには，実際のプロセスを監視するスクリプトを作成してください。

ただし，監視対象としない場合や，ローカルノードだけの運用の場合，すなわち他ノードにフェイルオーバをさせない場合には，スクリプトを作成する必要はありません。

Hitachi Web Server の動作を監視するためのシェルスクリプトを作成します。「Hitachi Web Server に障害が発生して停止したと同時にその実行を終了する」という処理をするスクリプトの例を示します。

（例）

PidFile ディレクティブで指定したファイルに記録されているプロセス ID が実行中かどうかを，5 秒おきに監視するシェルスクリプト（httpsd_monitor）です。

PidFile ディレクティブ指定値を絶対パスで表したものを引数に指定します。

```
#!/bin/sh
################################################################
### P-1B41-E171    Hitachi Web Server
### ALL RIGHTS RESERVED, COPYRIGHT (C) 2000, HITACHI,LTD.
################################################################
HWSITIME=5
if [ $# -ne 1 ]
then
    exit 1
fi

HWSIDFILE=$1

if [ ! -e $HWSIDFILE ]
then
    exit 1
fi

HWSID=`cat $HWSIDFILE`
if [ x$HWSID = "x" ]
then
    exit 1
fi

while true
do
   STATUS=`ps -p $HWSID | grep $HWSID | awk '{print $1}' `
   if [ x$STATUS = "x" ]
   then
     break
   fi
   sleep $HWSITIME
done

exit 0
```

（1） 注意事項

Hitachi Web Server では，リクエストを処理するためのプロセス群を制御するプロセスが一つあります（「4.1　Hitachi Web Server の処理とディレクティブとの関係」参照）。例に示したスクリプト httpsd_monitor では，その制御プロセスが動作しているかどうかを監視します。リクエスト処理のためのプロセス群の動作は監視しません。

付録 D.4　MC/ServiceGuard の設定

Hitachi Web Server と，必要であれば関連プログラムをパッケージに定義します。ここで説明していない内容および詳細説明については，MC/ServiceGuard のマニュアルを参照してください。

(1) クラスタの構成およびパッケージの構成

クラスタの構成およびパッケージの構成例を次に示します。

(a) クラスタ構成例

```
CLUSTER_NAME cluster1
FIRST_CLUSTER_LOCK_VG /dev/vg01
NODE_NAME original_node
NETWORK_INTERFACE lan0
HEARTBEAT_IP 172.16.1.1
FIRST_CLUSTER_LOCK_PV /dev/dsk/c1t2d0
NODE_NAME adoptive_node
NETWORK_INTERFACE lan0
HEARTBEAT_IP 172.16.1.2
FIRST_CLUSTER_LOCK_PV /dev/dsk/c1t2d0
HEARTBEAT_INTERVAL 1000000
NODE_TIMEOUT 2000000
AUTO_START_TIMEOUT 600000000
NETWORK_POLLING_INTERVAL 2000000
MAX_CONFIGURED_PACKAGES 10
VOLUME_GROUP /dev/vg01
```

(b) パッケージ構成例

```
PACKAGE_NAME HitachiWebServer
FAILOVER_POLICY CONFIGURED_NODE
FAILBACK_POLICY MANUAL
NODE_NAME original_node
NODE_NAME adoptive_node
RUN_SCRIPT /etc/cmcluster/HitachiWebServer/control.sh
RUN_SCRIPT_TIMEOUT NO_TIMEOUT
HALT_SCRIPT /etc/cmcluster/HitachiWebServer/control.sh
HALT_SCRIPT_TIMEOUT NO_TIMEOUT
SERVICE_NAME httpsd_check
SERVICE_FAIL_FAST_ENABLED NO
SERVICE_HALT_TIMEOUT 300
SUBNET 172.16.1.0
PKG_SWITCHING_ENABLED YES
NET_SWITCHING_ENABLED YES
NODE_FAIL_FAST_ENABLED NO
```

(2) パッケージ制御スクリプトの記述

Hitachi Web Server を監視する場合のパッケージ制御スクリプトの記述について説明します。次に示す説明以外は，個々のシステムに応じて設定してください。

(a) スクリプトの登録

作成したシェルスクリプト httpsd_monitor を，MC/ServiceGuard が監視するサービスとして登録します。次の例では，httpsd_monitor が /opt/hitachi/httpsd/bin に格納されていることを前提としています。

なお，Hitachi Web Server のコンフィグファイルに指定した PidFile ディレクティブの値と，httpsd_monitor の引数には，同じ値を指定します。Hitachi Web Server を監視対象としない場合，SERVER_NAME と SERVER_CMD は登録する必要はありません。

```
PATH=/sbin:/usr/bin:/usr/sbin:/etc:/bin
VGCHANGE="vgchange -a e"
VG[0]=/dev/vg01
LV[0]=/dev/vg01/lvol1
FS[0]=/MCSG
FS_MOUNT_OPT[0]="-o rw"
IP[0]=172.16.1.3
SUBNET[0]=172.16.1.0
SERVICE_NAME[0]="httpsd_check"
SERVICE_CMD[0]="/opt/hitachi/httpsd/bin/httpsd_monitor
                /opt/hitachi/httpsd/logs/httpd.pid"
SERVICE_RESTART[0]="-r 0"
```

(b) 関数の定義

パッケージ制御スクリプト内の関数 customer_defined_run_cmds（パッケージ起動時）と，関数 customer_defined_halt_cmds（パッケージ停止時）内に，Hitachi Web Server を起動または終了させる処理を記述してください。

（起動）

```
function customer_defined_run_cmds
{
# ADD customer defined run commands.
: # do nothing instruction, because a function must contain some
command.
  /opt/hitachi/httpsd/sbin/httpsdctl start
  test_return 51
}
```

（終了）

```
function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
: # do nothing instruction, because a function must contain some
command.
  /opt/hitachi/httpsd/sbin/httpsdctl stop
test_return 52
}
```

(3) 注意事項

ほかのノードへのフェイルオーバが発生した場合，通常の HTTP 接続，SSL を使用した接続はすべて切断され，待機ノードには引き継がれません。クライアントは，再接続してください。

付録 E　HACMP for AIX によるシステム監視（クラスタ・マルチプロセッシングの運用）

主幹業務の計算プラットフォームを構築するために IBM 社が開発したツールが HACMP for AIX ソフトウェアです。HACMP for AIX には，ハイ・アベイラビリティー（HA）およびクラスタ・マルチプロセッシング（CMP）という二つの主要なコンポーネントがあります。Hitachi Web Server は，HACMP for AIX を使用したクラスタ・マルチプロセッシングで運用できます。なお，HACMP for AIX の詳細については，HACMP for AIX のマニュアルを参照してください。また，Hitachi Web Server の前提プログラム（OS など）や関連プログラム（CGI プログラムなど）をクラスタ・マルチプロセッシング構成で運用する場合の詳細については，各プログラムのマニュアルを参照してください。

HACMP for AIX を利用すると，ハードウェア障害およびソフトウェア異常停止によるコンテンツ配信の停止時間を最小限にとどめた Web サーバ運用ができます。また，サービスを停止しないで，ソフトウェアの管理，保守およびバージョンアップもできます。

対象となる主な障害

HACMP for AIX の対象となる主な障害には，次のものがあります。

- LAN 障害
- リソース（システム演算処理装置，ディスク，インタフェース）障害
- ソフトウェアの異常停止

付録 E.1　ハードウェア構成例と HACMP for AIX の動作概要

HACMP for AIX は，監視対象のノードに障害が発生すると，予備のノードに処理を切り替えてサービスを続行します。この動作を，テイクオーバといいます。テイクオーバには，次に示す切り替え機能があります。

- ノード
- アプリケーション
- ネットワークおよびネットワーク・アダプタ
- ディスクおよびディスク・アダプタ

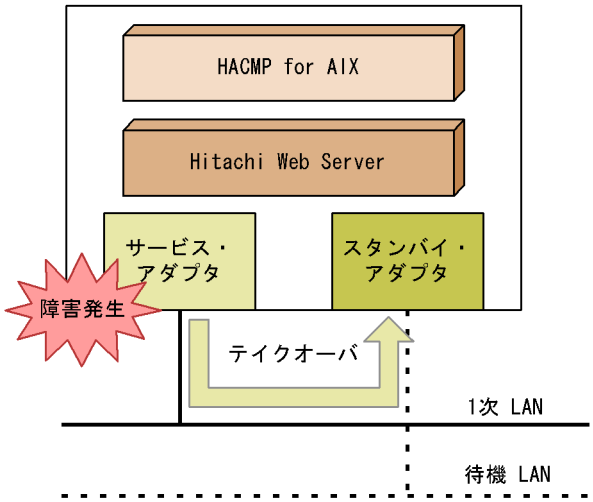
次にネットワークの障害例とアプリケーションの障害例を示します。

（1）ネットワークの障害例

LAN アダプタを二重化するときは，一方をアプリケーションのサービスをするために稼働させるアダプタ（サービス・アダプタ）とし，他方をサービス・アダプタをバックアップするアダプタ（スタンバイ・アダプタ）として定義します。

この構成では，サービス・アダプタ側に障害が発生すると，同一ノード上のスタンバイ・アダプタに接続が切り替えられます。アダプタの二重化について，次に示します。

図 E-1　アダプタの二重化



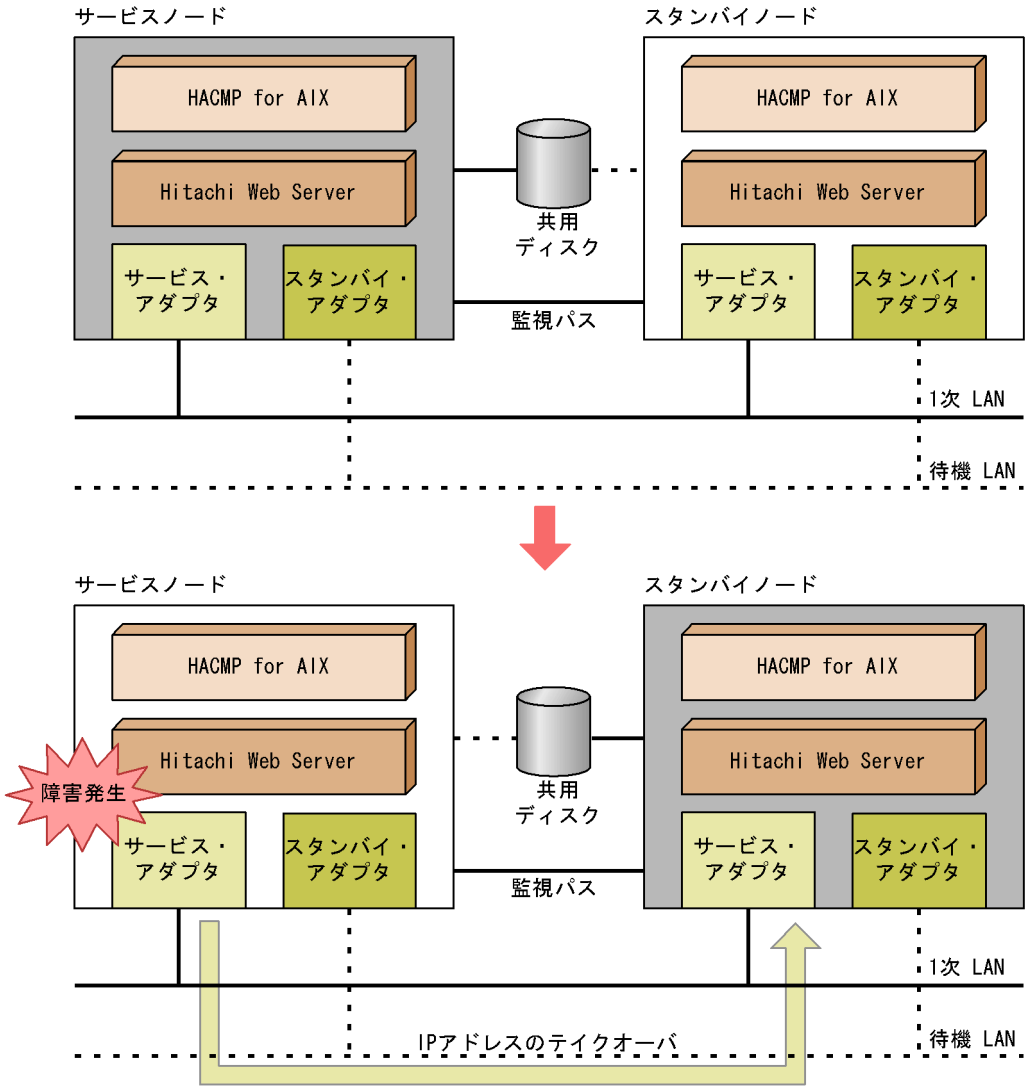
(2) アプリケーションの障害例

ノードを二重化した場合の例について示します。

稼働ノード（サービス・ノード）とバックアップ用ノード（スタンバイ・ノード）の 2 台のサーバ構成で，LAN を二重化し，双方の監視用パスとして RS232 通信を使用して 2 台のノード間を接続する構成とします。ディスク記憶装置は，ノード間で共有します。

サービス・ノードに障害が発生し，他ノードへのフォールオーバが必要だと判断すると，サービス・ノードを除去して，スタンバイ・ノード上でサービスを継続させます。共有ディスクは，スタンバイ・ノードで継続的に利用できます。複数ノードの運用の例を，次に示します。

図 E-2　複数ノードの運用の例（HACMP for AIX）



付録 E.2　Hitachi Web Server の設定

HACMP for AIX に適用するための，Hitachi Web Server の設定手順を次に示します。

1. Hitachi Web Server を各ノードのローカルディスクにそれぞれインストールします。
2. サービス・ノード上で Hitachi Web Server のコンフィグファイル，開始スクリプト，停止スクリプトおよびモニタメソッドを作成します。
3. 必要に応じて，コンフィグファイル，開始スクリプト，停止スクリプトおよびモニタメソッドをスタンバイ・ノードへ配布します。
4. HACMP for AIX で Hitachi Web Server 用のアプリケーションサーバを定義します。

5. HACMP for AIX で Hitachi Web Server 用のアプリケーションモニタを定義します。
6. HACMP for AIX の定義を完成し，クラスタ定義をすべてのノードで同期化します。
7. クラスタサービスを開始します。

環境設定については，次の点に注意してください。

（1） コンフィグファイルの文法チェック

クラスタサービスの開始前には，"/opt/hitachi/httpsd/sbin/httpsdctl　configtest" を実行し，サーバ設定が正しいことを確認してください。

（2） コンフィグファイルの変更

コマンド "httpsdctl restart" または "httpsdctl graceful" をコマンドラインから直接実行すれば，HACMP for AIX で使用中に Hitachi Web Server の設定を変更できます。変更内容は，ほかのノードにも反映する必要があります。

（3） CRL を用いて運用している場合

CRL を用いて運用しているときは，スタンバイノード上でも，サービスノードと同様の CRL を設定する必要があります。

付録 E.3　監視スクリプトの作成

HACMP for AIX で，Hitachi Web Server を監視対象とするためには，Hitachi Web Server の動作をモニタするスクリプトを作成して，モニタメソッドに登録する必要があります。このスクリプトは，Hitachi Web Server が正常であれば 0 を戻し，問題を検出したときに 0 以外の値を戻す必要があります。

Hitachi Web Server では，実行コマンドと実際にサービスするプロセスが異なります。HACMP for AIX の監視対象とするためには，実際のプロセスを監視するスクリプトを作成してください。

ただし，監視対象としない場合や，ローカルノードだけの運用の場合にはスクリプトを作成する必要はありません。

Hitachi Web Server の動作を監視するためのシェルスクリプトを作成します。「Hitachi Web Server が正常であれば 0 を戻し，問題を検出したときに 0 以外の値を戻す」という処理をするスクリプトの例を示します。

（例）

PidFile ディレクティブで指定したファイルに記録されているプロセス ID が実行中であれば 0 を戻し，実行されていなければ 1 を戻すスクリプトです。

```
#!/bin/sh
################################################################
### P-1M41-E171    Hitachi Web Server
### ALL RIGHTS RESERVED, COPYRIGHT (C) 2001, HITACHI,LTD.
################################################################


HWSIDFILE=/opt/hitachi/httpsd/logs/httpd.pid
if [ ! -e $HWSIDFILE ]
then
        exit 1
fi

HWSID=`cat $HWSIDFILE`
if [ x$HWSID = "x" ]
then
        exit 1
fi

STATUS=`ps -p $HWSID | grep $HWSID | awk '{print $1}'`

if [ x$STATUS = "x" ]
then
        exit 1
else
        exit 0
fi
```

（1）注意事項

Hitachi Web Server では，リクエストを処理するためのプロセス群を制御するプロセスが一つあります（「4.1　Hitachi Web Server の処理とディレクティブとの関係」を参照してください）。例に示したスクリプトでは，その制御プロセスが動作しているかどうかを監視します。リクエスト処理のためのプロセス群の動作は監視しません。

付録 E.4　HACMP for AIX の設定

Hitachi Web Server と，必要であれば関連プログラムをパッケージに定義します。ここで説明していない内容および詳細説明については，HACMP for AIX のマニュアルを参照してください。

（1）Hitachi Web Server のアプリケーション・サーバ登録方法

HACMP for AIX で Hitachi Web Server を管理するためには，Hitachi Web Server をアプリケーション・サーバとして登録する必要があります。

SMIT を使い「アプリケーション・サーバの追加（Add an Application Server）」画面を選び，Hitachi Web Server 用のサーバ名，開始スクリプトおよび停止スクリプトを登録

します。

(a) 開始スクリプト例

```
#!/bin/sh
/opt/hitachi/httpsd/sbin/httpsdctl start
```

(b) 停止スクリプト例

```
#!/bin/sh
/opt/hitachi/httpsd/sbin/httpsdctl stop
```

(2) Hitachi Web Server のモニタ方法

HACMP for AIX で Hitachi Web Server を監視するためには，監視スクリプトを登録する必要があります。運用を考慮してスクリプトを作成します。

HACMP/ES（拡張スケーラブル機能）を使った場合は SMIT を使い「ユーザ定義アプリケーション・モニタの追加（Add Custom Application Monitor）」画面を選び，Hitachi Web Server 用の監視スクリプトを登録できます。

(3) 注意事項

ほかのノードへのテイクオーバが発生した場合，通常の HTTP 接続，SSL を使用した接続はすべて切断され，待機ノードには引き継がれません。クライアントは，再接続してください。

付録 F　Microsoft サーバクラスタによるシステム監視

Microsoft サーバクラスタは Microsoft 社のソフトウェアです。Hitachi Web Server は，Microsoft サーバクラスタを使用し，クラスタサービスを実行し運用できます。なお，Microsoft サーバクラスタの詳細については，Microsoft サーバクラスタのマニュアルを参照してください。

サーバクラスタの対象となる主な障害には，次のものがあります。

- LAN 障害
- リソース（システム演算処理装置，ディスク，インタフェース）障害
- ソフトウェアの異常停止

サーバクラスタは，監視対象のシステム（以降，**1 次系**とする）に障害が発生すると，予備のシステム（以降，**待機系**とする）に処理を切り替えてサービスを続行します。この動作を，フェイルオーバといいます。

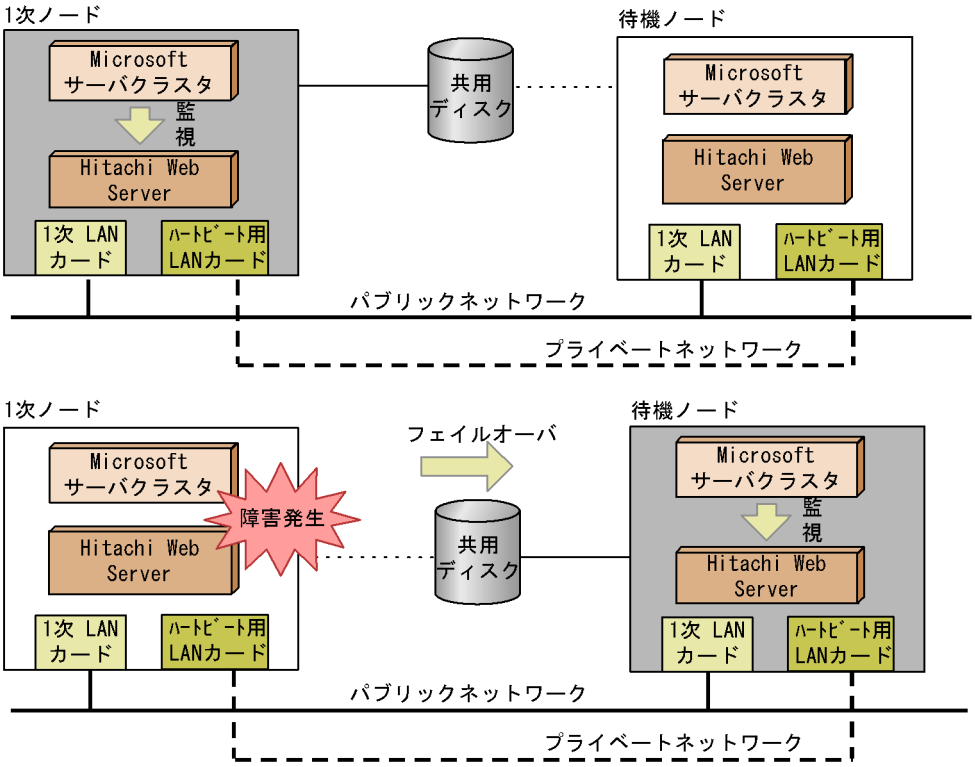
Windows Server 2008 でご使用になる場合は，用語を次のように読み替えてご使用ください。

本書およびマニュアルに記載の用語	Windows Server 2008 よりの用語
Microsoft Cluster Service または，Microsoft サーバクラスタ	Windows Server Failover Cluster
クラスタアドミニストレータ	フェールオーバー クラスタ管理
グループまたは，リソースグループ	クラスタ化されたサービスやアプリケーション

付録 F.1　運用の例

1 次ノードに障害が発生し，他ノードへのフェイルオーバが必要だと判断すると，1 次系システムを停止させ，待機系システム上で起動してサービスを継続させます。共有ディスクは，待機系システムの方にマウントされます。複数ノードの運用の例を次に示します。

図 F-1　複数ノードの運用の例（Microsoft サーバクラスタ）



付録 F.2　Hitachi Web Server の設定

サーバクラスタに適用するための，Hitachi Web Server の設定手順を次に示します。

1. Hitachi Web Server を各ノードのローカルディスクにそれぞれインストールします。
2. 各ノードで，Hitachi Web Server を Windows のサービスとして登録します。登録するサービス名は各ノードで共通にします。Hitachi Web Server をインストールしたときに登録される，サービス名 "Hitachi Web Server" のサービスをクラスタ化したい場合は，この手順は不要です。
3. サーバクラスタの設定をします。
4. Hitachi Web Server のコンフィグファイルを各ノードに配布します。
5. サーバクラスタのグループをオンラインにします。

環境設定については，次の点に注意してください。

(1) バーチャルホストの場合

フェイルオーバの結果，クライアントに返送されるサーバ名が変化する場合があります。このため，バーチャルホストでも ServerName ディレクティブは必ず設定してください。

（2） IP アドレスの指定

IP アドレスを指定するディレクティブ（<VirtualHost>，BindAddress，Listen，NameVirtualHost）では，LAN カードに指定された IP アドレス（他ノードに移動できない IP アドレス）ではなく，再配置できる IP アドレス（リソースモジュールに対して与えられ，他ノードに移動できる IP アドレス）を使用してください。

（3） コンフィグファイルの文法チェック

サーバクラスタの起動の前には，インストールしたディレクトリで "httpsd -t" を実行し，サーバ設定が正しいことを確認してください。Hitachi Web Server のコンフィグファイルの中で，IP アドレスや記憶域など，クラスタのグループに属しているリソースを参照している場合，確認しようとしているノードにグループを移動してから，” httpsd -t ” を実行してください。

（4） コンフィグファイルの変更

サービスがオンライン中のノードで，httpsd コマンドやスタートメニューにより Hitachi Web Server のサービスを再起動すれば，サーバクラスタで使用中のサービスをオフラインにすることなく Hitachi Web Server の設定を変更できます。コンフィグファイルの変更内容は，ほかのノードにも反映する必要があります。

（5） CRL を使用して運用している場合

待機ノードにも，1 次ノードと同様の CRL を設定する必要があります。

（6） リソースの種類

Hitachi Web Server をリソースで指定する場合，汎用アプリケーションではなく汎用サービスで指定してください。汎用アプリケーションでは，Hitachi Web Server は正常に動作しません。

付録 F.3　サーバクラスタの設定

（1） Microsoft サーバクラスタの設定（Windows Server 2003 の場合）

Microsoft サーバクラスタの設定では，クラスタ管理ソフトウェアの「クラスタアドミニストレータ」を使用します。ツールの詳細については，Microsoft 社のドキュメントを参照してください。

クラスタアドミニストレータを使用してグループを作成します。

そのグループに属する汎用サービス（Hitachi Web Server のサービス），ネットワーク，IP アドレス，物理ディスクなど，フェイルオーバ時にノード間を移動するリソース類を作成します。汎用サービスを作成する際，起動パラメータの設定値は何も入力しないでください。

（2）Windows Server Failover Cluster の設定（Windows Server 2008 の場合）

Windows Server Failover Cluster の設定では，クラスタ管理ソフトウェアの「フェールオーバー クラスタ管理」および cluster コマンドを使用します。これらの詳細については，Microsoft 社のドキュメントを参照してください。

次に示す手順で設定します。

1. 「フェールオーバー クラスタ管理」を使用して，Hitachi Web Server のクラスタ化されたサービスを作成します。クラスタ化されたサービスに属する汎用サービス (Hitachi Web Server のサービス)，クライアントアクセスポイント (名前・IP アドレス)，記憶域など，フェイルオーバ時にノード間を移動するリソース類を追加します。各項目のプロパティを表示し，リソースの依存関係やその他クラスタに関する設定を実施します。
2. 管理者として実行したコマンドプロンプトを開きます。
3. コマンドプロンプトから，以下のコマンドを実行します。
   cluster △ res △ " リソース名 " △ /priv △ StartupParameters=""
   △は半角スペースを表しています。リソース名には，Hitachi Web Server の汎用サービスのリソース名を指定します。Hitachi Web Server の汎用サービスのリソース名は「フェールオーバー クラスタ管理」から確認してください。
4. 「フェールオーバー クラスタ管理」から，Hitachi Web Server の汎用サービスのプロパティを開き，「セットアップ パラメータ」の値が空白になっていることを確認します。

付録 G　バージョン 03-00 以降への移行方法

バージョン 03-00 より前の Hitachi Web Server をバージョン 03-00 以降へ移行する手順を次に示します。

1. Hitachi Web Server を上書きインストールします。
2. 次の設定を見直します。

   ErrorDocument ディレクティブにテキストを指定している場合

   バージョン 03-00 より前では，先頭に " を記述して文字列を指定しましたが，
   バージョン 03-00 以降は，文字列を "" で囲ってください。
   (例)
   バージョン 03-00 より前での指定方法

   ```
   ErrorDocument 500 "Server Error.
   ```

   バージョン 03-00 以降での指定方法

   ```
   ErrorDocument 500 "Server Error."
   ```

   リバースプロキシを使用している場合

   バージョン 03-00 より前では，Windows 版では mod_proxy.so，UNIX 版では libproxy.so だけをロードしていましたが，バージョン 03-00 以降は，mod_proxy.so および mod_proxy_http.so の二つをロードしてください。UNIX 版の場合は必ず下記の例に示す順序でロードしてください。
   (例)
   Windows 版
   バージョン 03-00 より前での指定方法

   ```
   LoadModule proxy_module modules/mod_proxy.so
   ```

   バージョン 03-00 以降での指定方法

   ```
   LoadModule proxy_module modules/mod_proxy.so
   LoadModule proxy_http_module modules/mod_proxy_http.so
   ```

   UNIX 版
   バージョン 03-00 より前での指定方法

   ```
   LoadModule proxy_module libexec/libproxy.so
   ```

   バージョン 03-00 以降での指定方法

   ```
   LoadModule proxy_module libexec/mod_proxy.so
   LoadModule proxy_http_module libexec/mod_proxy_http.so
   ```

   ShmemUIDisUser ディレクティブを使用している場合

   内部処理の変更に伴い，バージョン 03-00 以降はこのディレクティブは設定不要となりましたので，
   ディレクティブ指定を削除してください。

3. TransferLog ディレクティブ，CustomLog ディレクティブ，ErrorLog ディレクティブに対し，rotatelogs または rotatelogs2 以外のプログラムをパイプ指定している場合

(Windows 版だけ)

- ログ情報に含める改行コードが LF から CRLF に変更されました。そのためプログラムの見直しが必要となる場合があります。
- ディレクティブにプログラムを指定するときは，プログラムの拡張子（例えば .exe）を含める必要があります。

  (例) ログを出力するユーザ作成プログラム writelogs.exe を指定する場合

  バージョン 03-00 より前での指定方法

  CustomLog "|¥"¥"C:/proprietary/writelogs¥" プログラム引数 ¥""

  バージョン 03-00 以降での指定方法

  CustomLog "|¥"¥"C:/proprietary/writelogs.exe¥" プログラム引数 ¥""

4. Hitachi Web Server を起動します。

付録 H　用語解説

< 英字 >

AES

Advanced Encryption Standard の略です。2000 年に米国標準技術局（NIST）が，DES に代わる政府の標準暗号方式として採用した対称鍵暗号の一方式です。

Base64

E-Mail などでバイナリデータを送信する場合に利用されるエンコード方式です。

CA

Certification Authority の略です。SSL を使用するための証明書を発行する認証局です。

CGI プログラム

Common Gateway Interface の略です。Web サーバとサーバ上で動作するプログラムとのインタフェースをとるプログラムです。

CRL

Certificate Revocation List の略です。鍵の漏洩などで鍵の信頼性が失われ，無効となった証明書のリストです。

CSR

Certificate Signing Request の略です。Web サーバサイトの証明書を発行してもらうために，認証局（CA）に提出する証明書発行要求ファイルのことです。

DER

証明書，秘密鍵，CSR および CRL ファイルの形式です。バイナリ形式のファイルです。

DES

Data Encryption Standard の略です。1977 年に米国商務省標準局が，政府の標準暗号方式として公表した，対称鍵暗号の一方式です。

DNS

Domain Name System の略です。TCP/IP システムの場合に，インターネットやイントラネットで使われる分散名前管理システムです。DNS を利用すると，Web で目的のサイトを探すときに，172.17.40.100 などのわかりにくい数字（IP アドレス）ではなく，サイトを表すわかりやすい名前で指定できます。
DNS サーバは，ホスト名のルックアップ要求に答えて，IP アドレスからホスト名への変換やホスト名から IP アドレスへの変換をします。

DSO

Dynamic Shared Object の略です。DSO によって，Web サーバの機能を動的に拡張できます。LoadModule ディレクティブに指定されたモジュールは DSO によって Web サーバに組み込まれます。

FQDN

Fully Qualified Domain Name（完全修飾ドメイン名）の略です。ホスト名＋ドメイン名による記述形式です。同一ホスト上に複数のサービスを稼働させる場合にネームサーバ上で別名定義することが多くあります。そのとき，www や news などのホスト名を使用して，FQDN を定義します。一般的には FQDN のことをホスト名と呼ぶ場合もあります。

HACMP for AIX

主幹業務の計算プラットフォームを構築するために IBM 社が開発したツールです。HACMP for AIX には，ハイ・アベイラビリティー（HA）およびクラスタ・マルチプロセッシング（CMP）という二つの主要なコンポーネントがあります。Hitachi Web Server は，HACMP for AIX を使用したクラスタ・マルチプロセッシングで運用できます。

HTTP

HyperText Transfer Protocol の略です。Web サーバと Web ブラウザ間の通信で使用するプロトコルです。

httpsd.conf ファイル

このファイルに Web サーバの環境を設定するディレクティブを定義します。

KeepAlive

一度のコネクションで複数のリクエストを処理する機能です。この機能を使用すると，コネクションの確立，解放のオーバヘッドがなくなり，レスポンスが良くなります。

LDAP

Lightweight Directory Access Protocol の略です。ISO 標準である X.500 を簡略化したディレクトリ・アクセスのためのプロトコルです。インターネットやイントラネットなどの TCP/IP ネットワークで，ディレクトリデータベースにアクセスするためのプロトコルです。

MC/ServiceGuard

MC/ServiceGuard は，Hewlett-Packard 社のソフトウェア製品です。MC/ServiceGuard を使用すると，ハードウェア障害またはソフトウェア障害が発生した場合でも，予備の待機系ノードに処理を切り替え (フェイルオーバ) て，サービスの中断を最小限に抑えられます。

MD5

Message Digest Algorithm 5 の略です。元になる長いメッセージから一方向性ハッシュ関数を使い，固定長のパターンを生成する技術の一方式です。デジタル署名などに用いられます。MD5 は RFC1321 で規定されています。

mime.types ファイル

コンテンツのファイル拡張子とコンテントタイプ（MIME タイプ）の関連づけを定義するファイルです。TypesConfig ディレクティブで異なるファイル名も指定できます。

NIS

Network Information Service の略です。Sun Microsystems 社が開発し UNIX に導入した，ネットワーク管理システムです。

PEM

証明書，秘密鍵，CSR および CRL ファイルの形式です。DER 形式のファイルを Base64 エンコード処理し，テキスト形式にしたファイルです。
PEM 形式の証明書ファイルでは，データの前後に "-----BEGIN CERTIFICATE-----"，"-----END CERTIFICATE-----" というタグが付きます。
秘密鍵では，データの前後に "-----BEGIN RSA PRIVATE KEY-----"，"-----END RSA PRIVATE KEY-----" というタグが付きます。
CSR では，データの前後に "-----BEGIN CERTIFICATE REQUEST-----"，"-----END CERTIFICATE REQUEST-----" というタグが付きます。
CRL では，データの前後に "-----BEGIN X509 CRL-----"，"-----END X509 CRL-----" というタグが付きます。

QOS

ユーザに提供するネットワークの通信品質を制御する技術の総称です。

RC2

RSA Security, Inc. で開発された対称鍵暗号の一方式です。

RC4

RSA Security, Inc. で開発された対称鍵暗号の一方式です。

RFC

Request for Comments の略です。インターネット上でのプロトコル標準などを含んでいる一連の文書またはその配布形式です。

RSA 暗号

Rivest-Shamir-Adleman Scheme の略です。リベスト，シャミアおよびエイドルマンが 1978 年に発明した公開鍵暗号の一方式です。

SCM

Service Control Manager の略です。サービスを管理するシステムです。SCM は判明しているサービスのリストをレジストリによって管理して，コンピュータの起動時にそれらを自動的に開始するかまたはユーザから要求されたときに開始します。サービスとして動作するプログラムは通常の EXE（実行可能）ファイルですが，SCM と正常に通信するためのインタフェースを確立するために，Microsoft の提供する必要な関数を，必要な手順に従って実施しなければなりません。

SHA

Secure Hash Algorithm の略です。元になる長いメッセージから一方向性ハッシュ関数を使い，固定長のパターンを生成する技術の一方式です。デジタル署名などに用いられます。1995 年に米国標準技術局 (NIST) によってアメリカ政府の標準ハッシュ関数として採用されました。

SSL

Secure Sockets Layer の略です。TCP 層の上位層です。クライアントと Web サーバの間で証明書による認証，鍵交換，暗号化およびメッセージ認証をします。

sslc.cnf ファイル

SSL 関連のユティリティを使用する場合の環境設定をするファイルです。

SSL バージョン 2

Netscape Communications 社によって開発された SSL の初期バージョンです。

SSL バージョン 3

SSL バージョン 2 に対して，大幅に改良したバージョンです。Netscape Communications 社によって開発されています。

TLS バージョン 1

SSL バージョン 3 を改訂し，IETF（The Internet Engineering Task Force）で標準化されているプロトコルです。

URI

Universal Resource Identifier の略です。インターネットのどこに，どんな情報があるかを位置づけるものです。RFC1630 で規定されています。URI は一般的概念で，URL の上位集合です。

URL

Uniform Resource Locator の略です。インターネットのどこに，どんな情報があるかを位置づけるものです。

Web サーバ

WWW 環境を構築するソフトウェアの一つで，Web ブラウザからの要求に対して，HTML 文書，画像，音声，動画などを送信します。

Web ブラウザ

WWW 環境を構築するソフトウェアの一つです。Web サーバに HTML 文書や画像などを要求し，画面上に表示します。

WWW

World Wide Web の略です。世界中の Web サーバとリンクして情報を利用できる，大規模情報システムです。

< ア行 >

アクセスコントロールファイル

ディレクトリのアクセス制御情報を定義したファイルのことです。このファイルは，アクセス制御するディレクトリの下に，AccessFileName ディレクティブに指定したファイル名と同じ名前で作成します。

暗号鍵サイズ

対称鍵暗号で使用される鍵のビット長です。

イメージマップ

画像（画像のファイル）に複数のリンクを定義できます。その定義個所をクリックすると，ほかの URL の情報を入手できます。

< カ行 >

鍵交換方式
クライアントと Web サーバ間で，鍵を送受信する際に使用する公開鍵暗号方式です。

クッキー
Web ブラウザからインターネットを介してサーバにアクセスしたときに，サーバからクライアント側に情報ファイルを送り，保存できる機能のことです。

計画停止
実行中のサーバプロセスまたはサーバスレッドを，実行終了後に停止する方法です。httpsdctl ユティリティの gracefulstop オプションまたは httpsd コマンドの -k gracefulstop オプションの指定によって，Hitachi Web Server は計画停止します。

公開鍵
公開鍵暗号方式で使用する鍵です。通信相手に公開するための鍵です。

公開鍵暗号方式
暗号方式の一つで，データを暗号化するための鍵と，復号するための鍵が異なる暗号方式です。

コンピュータ名
ネットワーク上のコンピュータに固有の NetBIOS 名です。コンピュータ名には，最小 1 文字，最大 15 文字まで指定できます。

コンフィグファイル
Web サーバおよび sslccert ユティリティの実行環境を定義しているファイルです。httpsd.conf，mime.types（TypesConfig ディレクティブでファイル名の変更可），アクセスコントロールファイル（.htaccess），sslc.cnf ファイル，および Include ディレクティブで指定したファイルを指します。

< サ行 >

サービス
Windows システムで，ユーザがログインしたときとは別に，システムを起動したときに自動的に開始するプロセスを作成して処理する機能です。メールを監視する Messenger サービス，at コマンドで登録されたタスクを一定時間ごとに実行する Schedule サービスなどがあります。

< タ行 >

対称鍵
対称鍵暗号方式で使用する鍵のことです。

対称鍵暗号方式
暗号方式の一つで，データを暗号化するための鍵と，復号するための鍵が同じである暗号方式です。

ディレクティブ

Web サーバの実行環境を定義するパラメタです。httpsd.conf ファイル，アクセスコントロールファイルに定義します。

ディレクトリインデクス

ディレクトリのファイル名一覧を出力する機能です。

トリプル DES（DES3）

安全性を高めるため DES を 3 段並べて鍵を長くした対称鍵暗号の一方式です。

< ナ行 >

内部リダイレクト

リダイレクトとは，クライアントからリダイレクト先に再リクエストすることです。これに対して，内部リダイレクトは，クライアントに再リクエストさせることなく，Web サーバでリダイレクト先に直接アクセスして結果を返すことを指します。例えば，ErrorDocument ディレクティブでローカル URL を指定した場合，エラーステータス番号に一致すると，Web サーバは内部的にローカル URL にアクセスし結果を返します。

認証方式

署名をするときに使用するアルゴリズムです。

< ハ行 >

バーチャルホスト

１台のサーバマシンを複数台のサーバマシンに見せる機能です。１台のサーバマシンに対して，DNS サーバで複数の名前を定義する方法（サーバ名に基づくバーチャルホスト）と，１台のサーバマシンに複数のネットワークインタフェースを設定または IP アドレスのエイリアスを指定する方法（IP アドレスに基づくバーチャルホスト）があります。

秘密鍵

公開鍵暗号方式で使用する鍵です。通信相手には公開しないで，送信側だけで保持する鍵のことです。

ホスト名のルックアップ

DNS サーバ，NIS サーバ，/etc/hosts ファイルなどを検索して，IP アドレスとホスト名との対応付けを解決することです。OS の機能（gethostbyname,gethostbyaddr など）を使用します。ホスト名から IP アドレスを検索することを**正引き**，IP アドレスからホスト名を検索することを**逆引き**といいます。

< マ行 >

メッセージ認証アルゴリズム

それぞれのメッセージに固有なハッシュ値を生成するためのアルゴリズムです。

< ラ行 >

リクエストログ

モジュールトレース，リクエストトレースおよび I/O フィルタトレースの総称です。

モジュールトレース

モジュールの各関数の実行時および CGI プログラムの実行時に採取されるトレースです。

リクエストトレース

次のときに採取されるトレースです。

・リクエスト処理開始時

・リクエスト処理完了時

・KeepAlive 接続の場合，次のリクエストラインの受信完了時

・リクエスト処理開始からリクエストライン受信完了前のコネクション切断時

I/O フィルタトレース

モジュールが実装している入出力フィルタ関数の実行時に採取されるトレースです。

索引

記号

A

B

C

D

E

F

G

H

I

S

T

U

W

あ

い

う

え

か

き

く

け

こ

さ

し

す

せ

そ

た

て

と

な

に

は

ひ

ふ

へ

ほ

め

も

ゆ

り

ろ

ソフトウェアマニュアルのサービス ご案内

１．マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

URL　http://www.hitachi.co.jp/soft/manual/

ホームページのメニューは次のとおりです。

■マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ，マニュアル名称，資料番号のいずれかから検索できます。
■CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
■マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
■オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
■サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
■ご意見・お問い合わせ	マニュアルに関するご意見，ご要望をお寄せください。

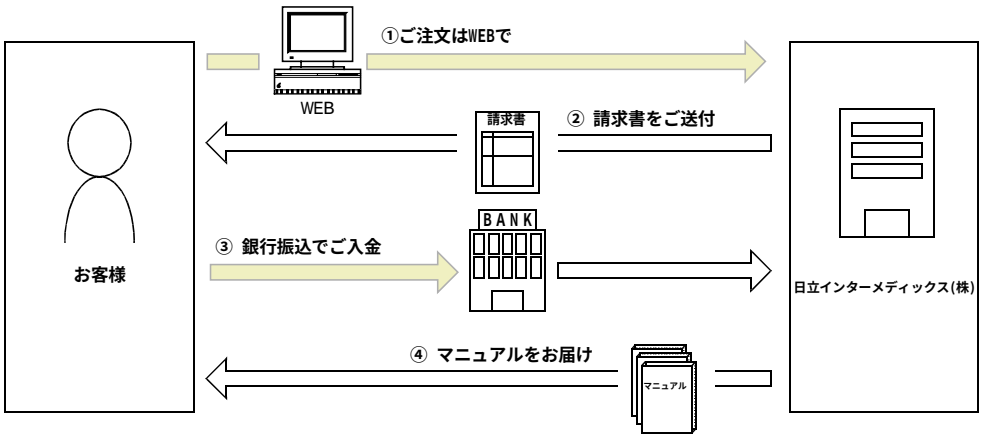
２．インターネットでのマニュアル公開

2 種類のマニュアル公開サービスを実施しています。

(1) マニュアル情報ホームページ「オンラインマニュアル」での公開
　製品をよりご理解いただくためのご参考として，一部製品のマニュアルを公開しています。

(2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開
　ソフトウェアサポートサービスご契約のお客様向けにマニュアルを公開しています。公開しているマニュアルの一覧，本サービスの対象となる契約の種別などはマニュアル情報ホームページの「サポートサービス」をご参照ください。

３．マニュアルのご注文



① マニュアル情報ホームページの「マニュアルのご購入」にアクセスし，お申し込み方法をご確認のうえ WEB からご注文ください。ご注文先は日立インターメディックス(株)となります。

② ご注文いただいたマニュアルについて請求書をお送りします。

③ 請求書の金額を指定銀行へ振り込んでください。

④ 入金確認後 7 日以内にお届けします。在庫切れの場合は，納期を別途ご案内いたします。