

# Solaris フリーソフトウェア導入手順書 -Apache による Web サーバの構築-



---

**2010年7月**  
**富士通株式会社**

#### ■ 商標について

- ・ Sun, Sun Microsystems, Sun ロゴ, Solaris および Solaris に関連する商標及びロゴは、米国およびその他の国における米国 Sun Microsystems 社の商標または登録商標です。
- ・ すべての SPARC 商標は、米国 SPARC International, Inc.の登録商標です。SPARC 商標のついた製品は米国 Sun Microsystems, Inc.のが開発したアーキテクチャに基づくものです。
- ・ SPARC64 は、米国 SPARC International, Inc.のライセンスを受けて使用している同社の登録商標です。
- ・ UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。
- ・ そのほか、本書に記載されている会社名および製品名は、それぞれ各社の商標または登録商標です。

#### ■ 留意事項

- ・ 本書の内容は、改善のため事前連絡なしに変更することがあります。
- ・ 本書の内容は、細心の注意を払って制作致しましたが、本書中の誤字、情報の抜け、本書情報の使用に起因する運用結果に関しましては、責任を負いかねますので予めご了承願います。
- ・ 本書に記載されたデータの使用に起因する第三者の特許権およびその他の権利の侵害については、当社はその責を負いません。
- ・ 無断転載を禁じます。

## はじめに

本書は、Solaris 10 OSに標準でインストールされている Apache の構築手順を記載した資料です。また、本書ではイントラネット内で利用することを前提としています。

構築環境は SPARC Enterprise を対象としています。本手順書に記載の設定値（ホスト名、IP アドレス等）は参考例です。構築時にはシステム環境に応じて読み替えてください。

## 目 次

事前注意事項 .....	5
1. ApacheとWebサーバ .....	6
2. 構築前の各種確認 .....	7
3. Apache2 の構築 .....	10
3-1. 構成ファイルの編集 .....	10
3-2. サービスの起動 .....	11
3-3. ブラウザからの確認 .....	11
4. SMFサービスの管理方法 .....	12
5. Apache2 環境の詳細設定 .....	13
5-1. 構成ファイルの設定 .....	13
5-2. httpd.confファイルの内容 .....	14
5-3. サービスの再起動 .....	21
6. アクセス制御情報ファイルの編集とBasic認証 .....	22
7. SSL認証(Secure Socket Layer) .....	25
7-1. SSL環境の設定 .....	25
7-2. ブラウザからの確認 .....	28
8. バーチャルホストの設定 .....	29
8-1. NAMEベース .....	29
8-2. IPベース .....	30

## 事前注意事項

本手順書で解説している手順は必須パッチを適用した Solaris OS 環境で実施してください。Solaris OS に関する注意事項は下記を参照してください。

必要なパッチが適用されていないと、Apache に関するコマンドが正常に実施できないなど不具合が発生することがあります。

- ・システムに適用されているパッチを調べる方法

```
# showrev -p
```

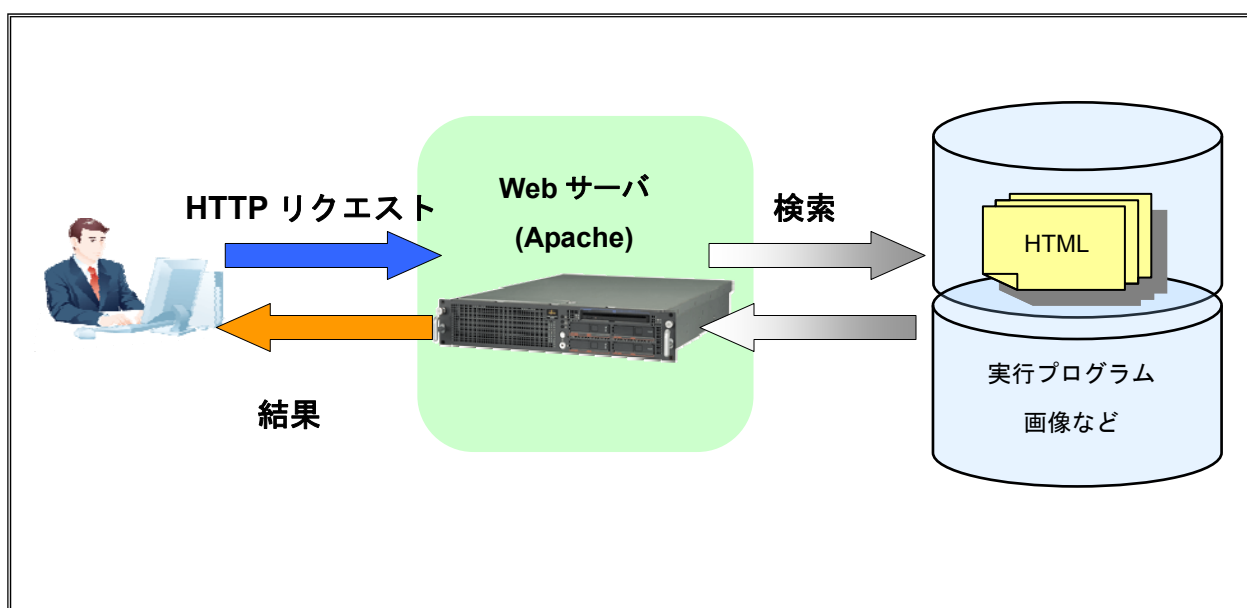
※一覧が表示されますので、パッチ ID を調べます。

## 1. Apache と Web サーバ

Apache とは、Web サービスを提供するための HTTP(ハイパーテキスト転送プロトコル)サーバソフトウェアです。

Apache を使用して Web サーバを構築することで、HTML 文書や画像などの情報を蓄積しておき、Web ブラウザなどのクライアントからの HTTP リクエストに応じて、これらの情報を送信することができます。(図:イメージ 1-1)

図:イメージ 1-1



## 2. 構築前の各種確認

Solaris 10 OS インストール時に、全体ディストリビューションと OEM サポート、全体ディストリビューションまたは、開発者システムサポートのいずれかを選択すると標準で Apache と Apache2 の 2 種類のバージョンがインストールされます。

本手順書では、Apache2 の構築手順を解説します。

構築前に本手順書で使用する環境の確認を行います。

- Solaris 10 OS のバージョン
- Apache2 のパッケージの有無とインストール状態の確認
- Apache2 のバージョン確認
- httpd サービスの状態確認

### 1) Solaris 10 OS のバージョンの確認

```
# cat /etc/release
Solaris 10 5/09 s10s_u7wos_08 SPARC
Copyright 2009 Sun Microsystems, Inc. All Rights Reserved.
Use is subject to license terms.
Assembled 30 March 2009
```

### 2) Apache2 のパッケージを確認します。

Solaris 10 には Apache のパッケージが 2 種類インストールされています。

- Apache2 のパッケージ : SUNWapch2d、SUNWapch2r、SUNWapch2u
- Apache のパッケージ : SUNWapchd、SUNWapchr、SUNWapchu

```
# pkginfo -l SUNWapch2d
PKGINST: SUNWapch2d
NAME: Apache Web Server V2 Documentation
CATEGORY: system
ARCH: sparc
VERSION: 11.10.0,REV=2005.01.08.05.16
BASEDIR: /
VENDOR: Sun Microsystems, Inc.
DESC: The Apache HTTP Server Version 2 (documentation)
PSTAMP: sfw10-patch20090202141229
INSTDATE: 3月 09 2010 16:20
HOTLINE: Please contact your local service provider
STATUS: completely installed
FILES: 779 installed pathnames
        2 shared pathnames
        24 directories
        19396 blocks used (approx)
```

```
# pkginfo -l SUNWapch2r

PKGINST: SUNWapch2r
  NAME: Apache Web Server V2 (root)
CATEGORY: system
  ARCH: sparc
VERSION: 11.10.0,REV=2005.01.08.05.16
BASEDIR: /
VENDOR: Sun Microsystems, Inc.
  DESC: The Apache HTTP server program Version 2 (root components)
PSTAMP: sfw10-patch20090702141617
INSTDATE: 3月 09 2010 16:20
HOTLINE: Please contact your local service provider
STATUS: completely installed
FILES:    323 installed pathnames
          8 shared pathnames
          19 directories
          16 executables
          1629 blocks used (approx)
```

```
# pkginfo -l SUNWapch2u

PKGINST: SUNWapch2u
  NAME: Apache Web Server V2 (usr)
CATEGORY: system
  ARCH: sparc
VERSION: 11.10.0,REV=2005.01.08.05.16
BASEDIR: /
VENDOR: Sun Microsystems, Inc.
  DESC: The Apache HTTP Server Version 2 (usr components)
PSTAMP: sfw10-patch20090702141724
INSTDATE: 3月 09 2010 16:20
HOTLINE: Please contact your local service provider
STATUS: completely installed
FILES:    558 installed pathnames
          2 shared pathnames
          95 directories
          296 executables
          12231 blocks used (approx)
```

3) Apache2 のバージョンを確認します。

```
# /usr/apache2/bin/httpd -v
Server version: Apache/2.0.63
Server built:   Jan 18 2009 15:51:05
```



4) Apache2 のサービスの状態を確認します。

```
# svcs svc:/network/http:apache2
STATE          STIME    FMRI
disabled       Mar_10   svc:/network/http:apache2
```

《参考》 Apache2 は SMF によって管理されますが、Apache は従来の RC スクリプトで管理されます (SMF の STATE の値が legacy\_run と表示されます)。

```
# svcs -a
STATE          STIME    FMRI
:
legacy_run     Mar_10   irc:/etc/rc3_d/S50apache           ← 「Apache」のサービス
:
disabled       Mar_10   svc:/network/http:apache2         ← 「Apache2」のサービス
```

### 3. Apache2 の構築

#### 3-1. 構成ファイルの編集

Apache2 の構成ファイルは/etc/apache2 配下にサンプルが用意されていますので、サンプルファイルをコピーして編集を行います。

1) サンプルファイルから構成ファイルをコピーします。

```
# cp /etc/apache2/httpd.conf-example /etc/apache2/httpd.conf
```

2) 構成ファイルを編集します。

```
# vi /etc/apache2/httpd.conf
```

3) [ServerName]の設定

サーバが自分自身のホスト名を示す時に使われる名前を指定します。ServerName ホスト名[:ポート番号]で指定します。ポート番号を省略した場合は、80 番ポートが指定されたものとみなします。

```
342 # If your host doesn't have a registered DNS name, enter its IP address here..
343 # You will have to access it by its address anyway, and this will make
344 # redirections work in a sensible way.
345 #
346 ServerName websv.example.co.jp:80
347
348 #
349 # UseCanonicalName: Determines how Apache constructs self-referencing
350 # URLs and the SERVER_NAME and SERVER_PORT variables.
```

※上記はホスト名「websv.example.co.jp」の場合の例

4) [DocumentRoot]の設定

HTML ドキュメントを置くトップディレクトリを指定します。任意の場所に設定します。

```
358 # DocumentRoot: The directory out of which you will serve your
359 # documents. By default, all requests are taken from this directory, but
360 # symbolic links and aliases may be used to point to other locations.
361 #
362 DocumentRoot "/var/apache2/htdocs"
363
364 #
365 # Each directory to which Apache has access can be configured with respect
366 # to which services and features are allowed and/or disabled in that
367 # directory (and its subdirectories).
```

### 3-2. サービスの起動

1) Apache2 のサービスを起動します。

```
# svcadm enable svc:/network/http:apache2
```

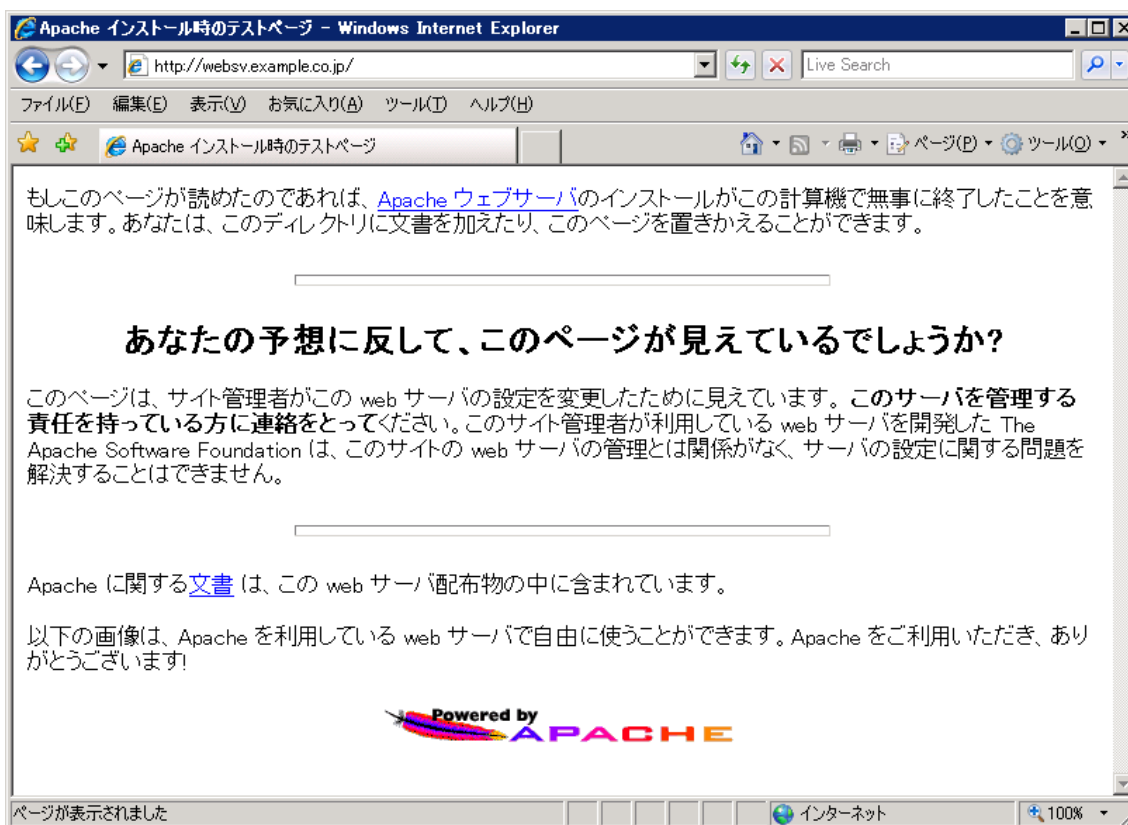
2) Apache2 のサービスを確認します。

```
# svcs svc:/network/http:apache2
STATE          STIME          FMRI
online         14:34:00      svc:/network/http:apache2
```

### 3-3. ブラウザからの確認

Web ブラウザから Web サーバへ接続できることを確認します。URL に下記を入力します。

http://ホスト名[:ポート番号] (ホスト名の代わりにIPアドレスでも可)



## 4. SMF サービスの管理方法

ここでは SMF 対応した Apache2 のサービス管理の方法についてまとめています。必要に応じて参照して下さい。

### 1) サービスの状態を確認する方法

```
# svcs svc:/network/http:apache2
STATE          STIME    FMRI
online         16:17:23 svc:/network/http:apache2
```

- STATE の主な表示は「online」（起動）、「disable」（停止）となります。
- サービス名は省略形[apache2]で指定することも可能です。以下のコマンドも同様です。

### 2) サービスの停止方法

```
# svcadm disable svc:/network/http:apache2
```

### 3) サービスの再起動方法

```
# svcadm restart svc:/network/http:apache2
```

### 4) サービスの起動方法

```
# svcadm enable svc:/network/http:apache2
```

## 5. Apache2 環境の詳細設定

### 5-1. 構成ファイルの設定

構成ファイル (/etc/apache2/httpd.conf) を編集することで Web サーバの動作環境の詳細設定が可能です。

以下に構成ファイルについて解説します。

	パラメータ名	意味
1	ServerRoot	サーバのベースディレクトリ指定
2	PidFile	PID の記録先ファイルの指定
3	IfModule	モジュールが存在するときに処理されるディレクティブを指定
4	Listen	リクエストを受け付ける IP アドレスとポート番号の設定
5	User/Group	httpd プロセスのユーザ名とグループ名の設定
6	ServerAdmin	通知先メールアドレスの設定
7	ServerName	自分自身を示すホスト名とポート番号の設定
8	UseCanonicalName	サーバが自身の名前とポートを決定する方法を設定
9	DocumentRoot	HTML ドキュメントの格納先の設定
10	Directory	指定のディレクトリ配下に適用されるディレクティブを指定
11	Options	ディレクトリに対して使用可能な機能を設定
12	AllowOverride	アクセス制御情報ファイル内で許可するディレクティブの種類を設定
13	DirectoryIndex	インデックス要求のデフォルトページの設定
14	AccessFileName	アクセス制御情報用に使用するファイルの指定
15	TypesConfig	MIME タイプ設定ファイルの位置を指定
16	DefaultType	デフォルトの MIME タイプの設定
17	HostnameLookups	名前解決の設定
18	ErrorLog	エラーログの格納先ファイルの設定
19	LogLevel	エラーログの詳細度の設定
20	LogFormat	アクセスログのフォーマット指定
21	CustomLog	ログの格納ファイルと出力フォーマットの設定
22	ServerTokens	http 応答ヘッダの設定
23	ServerSignature	サーバ情報の表示/非表示の設定
24	Alias	DocumentRoot 以外の場所にドキュメントを保存する設定

25	ScriptAlias	CGI スクリプトのパス設定
26	Include	http サーバ実行時に含むファイルの指定

## 5 - 2 .httpd.conf ファイルの内容

ここでは、/etc/apache2/httpd.conf の内容を設定内容毎に解説します。

### 1) [ ServerRoot ]

インストールされたサーバのベースディレクトリを指定します。

```
54 # Do NOT add a slash at the end of the directory path..
55 #
56 ServerRoot "/usr/apache2"
```

☞ ディレクトリパスの最後に"/ (スラッシュ) "を付加しないで下さい。

### 2) [ PidFile ]

サーバがプロセス ID (PID) を記録するファイルを指定します。

```
83 # PidFile: The file in which the server should record its process
84 # identification number when it starts.
85 #
86 <IfModule !mpm_netware.c>
87 PidFile /var/run/apache2/httpd.pid
88 </IfModule>
```

### 3) [ IfModule ]

<IfModule>から</IfModule>に囲まれるセクションは、モジュールが存在するときに処理されるディレクティブを指定するために利用します。

### 4) [ Listen ]

Apache が外部からリクエストを受け付ける IP アドレスとポート番号を指定します。

```
216 # Listen: Allows you to bind Apache to specific IP addresses and/or
217 # ports, instead of the default. See also the <VirtualHost>
218 # directive.
219 #
220 # Change this to Listen on specific IP addresses as shown below to
221 # prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
222 #
223 #Listen 12.34.56.78:80
224
225 Listen 80
```

## 5) [ User ] と [ Group ]

httpd サーバプロセスの所有者となるユーザ名とグループ名を指定します。

```

314 # User/Group: The name (or #number) of the user/group to run httpd as.
315 # . On SCO (ODT 3) use "User nouser" and "Group nogroup".
316 # . On HPUX you may not be able to use shared memory as nobody, and the
317 #   suggested workaround is to create a user www and use that user.
318 # NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
319 # when the value of (unsigned)Group is above 60000;
320 # don't use Group #-1 on these systems!
321 #
322 User webservd
323 Group webservd

```

## 6) [ ServerAdmin ]

httpd サーバに障害が発生した場合の通知先メールアドレスを指定します。

```

328 # ServerAdmin: Your address, where problems with the server should be
329 # e-mailed. This address appears on some server-generated pages, such
330 # as error documents. e.g. admin@your-domain.com
331 #
332 ServerAdmin you@yourhost.com

```

## 7) [ ServerName ]

サーバが自分自身を示すときに使うホスト名とポートを指定します。

例 : webserv.example.co.jp:80

```

342 # If your host doesn't have a registered DNS name, enter its IP address here.
343 # You will have to access it by its address anyway, and this will make
344 # redirections work in a sensible way.
345 #
346 ServerName 127.0.0.1

```

## 8) [ UseCanonicalName ]

サーバが自分自身の名前とポート番号を決定する方法を設定します。

ServerName ディレクティブで指定された値を使って、Apache HTTP サーバが自身を参照するように設定します。off に設定すると、代わりにサーバは自身を参照するときにクライアントの要求で使われる値を使用します。

```

349 # UseCanonicalName: Determines how Apache constructs self-referencing
350 # URLs and the SERVER_NAME and SERVER_PORT variables.
351 # When set "Off", Apache will use the Hostname and Port supplied
352 # by the client. When set "On", Apache will use the value of the
353 # ServerName directive.
354 #
355 UseCanonicalName Off

```

## 9) [DocumentRoot]

HTML ドキュメントを置くためのトップディレクトリを指定します。

```
358 # DocumentRoot: The directory out of which you will serve your
359 # documents. By default, all requests are taken from this directory, but
360 # symbolic links and aliases may be used to point to other locations.
361 #
362 DocumentRoot "/var/apache2/htdocs"
```

👉 ディレクトリパスの最後に"/"を付加しないで下さい。

## 10) [Directory]

特定ディレクトリとそのサブディレクトリだけに適用する 設定ディレクティブのグループを囲むのに使用されます。

## 11) [Options]

ディレクトリに対して使用可能な機能を設定します。

## 12) [AllowOverride]

アクセス制御情報ファイル (AccessFileName ディレクティブで指定したファイル名) で許可するディレクティブの種類を設定します。本ディレクティブに None を指摘した場合、対象ディレクトリ配下でアクセス制御情報ファイルは無効です。

```
365 # Each directory to which Apache has access can be configured with respect
366 # to which services and features are allowed and/or disabled in that
367 # directory (and its subdirectories).
368 #
369 # First, we configure the "default" to be a very restrictive set of
370 # features.
371 #
372 <Directory />
373     Options FollowSymLinks
374     AllowOverride None
375 </Directory>
```

## 13) [DirectoryIndex]

クライアントが HTML ファイルを指定しないでアクセスした場合、ここで指定したファイルにアクセスします。複数指定する場合はスペースで区切ります。優先順位は左からです。

```
445 # The index.html.var file (a type-map) is used to deliver content-
446 # negotiated documents. The MultiViews Option can be used for the
447 # same purpose, but it is much slower.
448 #
449 DirectoryIndex index.html index.html.var
```



## 1 4) [AccessFileName]

アクセス制御情報用に使用するファイルを指定します。

```
452 # AccessFileName: The name of the file to look for in each directory
453 # for additional configuration directives. See also the AllowOverride
454 # directive.
455 #
456 AccessFileName .htaccess
```

## 1 5) [TypesConfig]

MIME タイプマッピング（ファイル名拡張子から ContentType へ）のデフォルト一覧を設定するファイルを指定します。

```
468 # TypesConfig describes where the mime.types file (or equivalent) is
469 # to be found.
470 #
471 TypesConfig /etc/apache2/mime.types
```

## 1 6) [DefaultType]

MIME タイプを決定できない場合に使用される、デフォルト ContentType を設定します。

```
474 # DefaultType is the default MIME type the server will use for a document
475 # if it cannot otherwise determine one, such as from filename extensions.
476 # If your server contains mostly text or HTML documents, "text/plain" is
477 # a good value. If most of your content is binary, such as applications
478 # or images, you may want to use "application/octet-stream" instead to
479 # keep browsers from trying to display binary files as though they are
480 # text.
481 #
482 DefaultType text/plain
```

 MIME は、メッセージに何が含まれているかを知らせるものであり、これによりメッセージの内容は適切な方法で表示されます。

## 1 7) [HostnameLookups]

リクエストのあったクライアント IP アドレスからホスト名を逆引きするかどうかの指定をします。"On"の場合は毎回逆引きをしてログファイルにホスト名をロギングし、"Off"の場合は IP アドレスのままになります。"Double"の場合は逆引きしたホスト名をさらに正引きし、一致するかどうかをチェックします。

```
494 # HostnameLookups: Log the names of clients or just their IP addresses
495 # e.g., www.apache.org (on) or 204.62.129.132 (off).
496 # The default is off because it'd be overall better for the net if people
497 # had to knowingly turn this feature on, since enabling it means that
498 # each client request will result in AT LEAST one lookup request to the
499 # nameserver.
500 #
501 HostnameLookups Off
```

## 1 8) [ErrorLog]

エラーログの格納先ファイルを指定します。

```
523 # ErrorLog: The location of the error log file.
524 # If you do not specify an ErrorLog directive within a <VirtualHost>
525 # container, error messages relating to that virtual host will be
526 # logged here. If you *do* define an error logfile for a <VirtualHost>
527 # container, that host's errors will be logged there and not here.
528 #
529 ErrorLog /var/apache2/logs/error_log
```

## 1 9) [LogLevel]

エラーログの詳細度を指定します。

```
532 # LogLevel: Control the number of messages logged to the error_log.
533 # Possible values include: debug, info, notice, warn, error, crit,
534 # alert, emerg.
535 #
536 LogLevel warn
```

レベル	説明	備考
emerg	緊急 - システムが利用できない	
alert	直ちに対処が必要	
crit	致命的な状態	
error	エラー	
warn	警告	デフォルトレベル
notice	普通だが、重要な情報	
info	追加情報	
debug	デバッグメッセージ	

<http://www.apache.org/>より引用

## 2 0) [ LogFormat ]

アクセスログのフォーマットと、フォーマットに対するニックネームを指定します。ニックネームは CustomLog ディレクティブの引数として使用できます。以下に、フォーマットに指定できる主な "%" ディレクティブを示します。

%h リモートホストの情報  
 %l クライアントから報告される識別情報(条件を満たすとき取得可能)  
 %u ローカルで認証されたユーザ名(HTTP 基本認証のとき取得可能)  
 %t 時刻  
 %r リクエストの内容(先頭のみ)  
 %s サーバから返されるステータス (リダイレクト先のステータスを表示するには %>s)  
 %b リクエストに対して送信されたバイト数

```
539 # The following directives define some format nicknames for use with
540 # a CustomLog directive (see below).
541 #
542 LogFormat "%h %l %u %t %r" %>s %b "%[Referer]i" "%[User-Agent]i" combined
543 LogFormat "%h %l %u %t %r" %>s %b" common
544 LogFormat "%[Referer]i -> %U" referer
545 LogFormat "%[User-agent]i" agent
```

## 2 1) [ CustomLog ]

アクセスログの格納先ファイルと出力フォーマット（または、LogFormat ディレクティブで設定したフォーマットに対するニックネーム）を指定します。

```
551 # The location and format of the access logfile (Common Logfile Format).
552 # If you do not define any access logfiles within a <VirtualHost>
553 # container, they will be logged here. Contrariwise, if you *do*
554 # define per-<VirtualHost> access logfiles, transactions will be
555 # logged therein and *not* in this file.
556 #
557 CustomLog /var/apache2/logs/access_log common
```

## 2 2) [ ServerTokens ]

クライアントに送り返す Server 応答ヘッダに、サーバの一般的な OS 種別や、コンパイルされて組み込まれているモジュールの情報を含めるかどうかを指定します。

```
573 # ServerTokens
574 # This directive configures what you return as the Server HTTP response
575 # Header. The default is 'Full' which sends information about the OS-Type
576 # and compiled in modules.
577 # Set to one of: Full | OS | Minor | Minimal | Major | Prod
578 # where Full conveys the most information, and Prod the least.
```

```
579 #
580 ServerTokens Full
```

### 2 3) [ServerSignature]

サーバが作成するエラーページや FTP ファイルリストページなどの最下行に、上の ServerTokens で指定した情報を表示するかどうかの設定を行ないます。

“On”→表示する。

“Off”→表示しない。

```
583 # Optionally add a line containing the server version and virtual host
584 # name to server-generated pages (internal error documents, FTP directory
585 # listings, mod_status and mod_info output etc., but not CGI generated
586 # documents or custom error documents).
587 # Set to "EMail" to also include a mailto: link to the ServerAdmin.
588 # Set to one of: On | Off | EMail
589 #
590 ServerSignature On
```

### 2 4) [Alias]

エイリアスの指定をします。クライアントに公開するコンテンツは DocumentRoot 配下のディレクトリに設置する必要がありますが、エイリアスを使うことで別のディレクトリにあるファイルを DocumentRoot の配下に配置されているように見せかけることができます。

```
602 # We include the /icons/ alias for FancyIndexed directory listings. If you
603 # do not use FancyIndexing, you may comment this out.
604 #
605 Alias /icons/ "/var/apache2/icons/"
```

### 2 5) [ScriptAlias]

CGI スクリプトのパスを指定します。CGI のファイルに対して要求があった場合にはファイルの内容を返す代わりに要求があった CGI をサーバ側で実行し、その結果をブラウザに返します。通常このディレクトリには、CGI などの実行形式ファイルを配置します。

```
636 # ScriptAlias: This controls which directories contain server scripts.
637 # ScriptAliases are essentially the same as Aliases, except that
638 # documents in the realname directory are treated as applications and
639 # run by the server when requested rather than as documents sent to the client.
640 # The same rules about trailing "/" apply to ScriptAlias directives as to
641 # Alias.
642 #
643 ScriptAlias /cgi-bin/ "/var/apache2/cgi-bin/"
```

## 2 6) [Include]

http サーバ実行時に他の設定ファイルを含むようにすることができます。

```
1052 # Bring in additional module-specific configurations
1053 #
1054 <IfModule mod_ssl.c>
1055     Include /etc/apache2/ssl.conf
1056 </IfModule>
```

## 5-3. サービスの再起動

1) 構成ファイルを変更した場合、サービスを再起動して変更を反映させます

```
# svcadm restart svc:/network/http:apache2
```

2) サービスの状態を確認します

```
# svcs svc:/network/http:apache2
STATE      STIME      FMRI
online     10:10:53   svc:/network/http:apache2
```

※ STIME の値が更新されることを確認します

## 6. アクセス制御情報ファイルの編集と Basic 認証

Basic 認証とは、Web サイトの閲覧に使う http プロトコルが備える、最も基本的なユーザ認証方式です。アクセスの制限された Web ページにアクセスしようとする、Web ブラウザでユーザ名とパスワードの入力を求め、サーバでアクセスを許可しているユーザに一致すると、ページを閲覧することができる仕組みです。

Basic 認証はアクセス制御情報ファイル (AccessFileName ディレクティブで指定したファイル名 (デフォルトは .htaccess)) を用いて、特定ユーザにアクセス権を設定します。アクセス制御情報ファイルで設定した内容は、そのファイルを設置したディレクトリと、その配下の全てのファイルに対して有効となります。

### 1) アクセス制御情報ファイルの作成と編集 (例)

```
# vi /export/www/.htaccess
AuthUserFile /export/www/.htpasswd
AuthGroupFile /dev/null
AuthName "Please enter your ID & password"
AuthType Basic
Require valid-user
```

#### ➤ AuthUserFile

ユーザ認証のための、ユーザとパスワードの一覧ファイル (上記では、/export/www/.htpasswd) を指定します。ファイル名を絶対パスで指定しない場合、ServerRoot ディレクティブで指定したディレクトリからの相対パスとして扱われます。

#### ➤ AuthGroupFile

ユーザ認証のための、グループファイルを設定します。上記では設定なし (/dev/null)。Require ディレクティブで group を宣言したときのみ必要です。ファイル名を絶対パスで指定しない場合、ServerRoot ディレクティブで指定したディレクトリからの相対パスとして扱われます。

#### ➤ AuthName

ユーザ認証の際にクライアント側に表示させる文字列を設定します。上の例では、"Please enter your ID & password"の部分です。任意に設定することが可能です。

#### ➤ AuthType

ユーザ認証機構の種別を指定します。type には、Basic (ベーシック認証) と Digest (MD5 ダイジェスト認証) の設定が可能です。上記では Basic 認証を設定しています。

➤ Require

ユーザ認証の際に、認証させるユーザを指定します。第一引数により、パスワード認証の機会を与えるメンバーを指定します。

• require user

ユーザ単位で認証の機会を与えます。パスワードファイルに登録されているユーザのうち、設定ユーザのみに認証させ他のユーザは拒否します。

• require group

グループ単位で認証の機会を与えます。パスワードファイルに登録されているユーザのうち、設定グループに属しているユーザにのみ認証させ他は拒否します。AuthGroupFile で事前にグループ定義を行う必要があります。

• require valid-user

全てのユーザに、認証の機会を与えます。

➤ Satisfy (任意)

ユーザ認証とホスト制限に関するアクセスポリシーを定義します。

• Satisfy all

クライアントがホスト制限を通過し、そしてユーザ認証を通過することの、両方を要求します。  
(デフォルト設定)

• Satisfy any

ホスト制限を通過するか、またはユーザ認証を通過するかの、どちらかがクライアントに要求されます。

## 2) パスワードファイルの作成と認証ユーザの登録

htpasswd コマンドを実行してパスワードファイルを作成します。

**書式** : /usr/apache2/bin/htpasswd [-c] [パスワードファイル名] [登録ユーザ名]

実行例) .htpasswd ファイルに secret ユーザを登録する

```
/usr/apache2/bin/htpasswd -c /export/www/.htpasswd secret
New password:XXXXXX
Re-type new password:XXXXXX
Adding password for user secret
```

※初回作成時のみ「-c」オプションを付けて実行します。

実行例) guest ユーザを追加登録する

```
# /usr/apache2/bin/htpasswd /export/www/.htpasswd guest
New password:XXXXXX
Re-type new password:XXXXXX
Adding password for user guest
```

## 3) パスワードファイルの確認

```
# ls -la /export/www/.htpasswd
-rw-r--r--  1 root    root          41 10月 15日 16:13 /export/www/.htpasswd

# cat /export/www/.htpasswd
secret:hM/N3U1P4Hbko
guest:LMHim.4f2C0tA
```

※パスワードは暗号化されて記載されます。



## 7. SSL 認証(Secure Socket Layer)

SSL は HTTP や FTP など送受信するネットワーク上のデータを暗号化して、安全な通信を行うためのプロトコルです。

### 7-1.SSL 環境の設定


Solaris 10 OS 標準機能である openssl を使用して SSL 環境を設定する手順を説明します。本手順書では、自己署名の手順を解説します。

#### 1) 環境変数 PATH の設定

```
# PATH=$PATH:/usr/sfw/bin
# export PATH
```

#### 2) Apache が読み込むサーバ証明書や秘密鍵を格納するためのディレクトリを作成します。

```
# mkdir /etc/apache2/ssl.crt
# mkdir /etc/apache2/ssl.key
```

 ディレクトリ名は/etc/apache2/ssl.conf 内で定義されています。ディレクトリ名を変更したい場合は/etc/apache2/ssl.conf 内のパスも変更する必要があります。

#### 3) パスフレーズを作成します。

```
# openssl genrsa -des3 -out /etc/apache2/ssl.key/server.key 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
Enter pass phrase for /etc/apache2/ssl.key/server.key:XXXXXXXX

    ←4文字以上のパスフレーズを入力(任意)
Verifying - Enter pass phrase for /etc/apache2/ssl.key/server.key:XXXXXXXX

    ←パスフレーズを再入力
```

#### 4) CSR の発行

```
# openssl req -new -x509 -days 365 -key /etc/apache2/ssl.key/server.key -out /etc/apache2/ssl.crt/s
erver.crt
Enter pass phrase for /etc/apache2/ssl.key/server.key:          ←3)で入力したパスフレーズを入力
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
_____
```

Country Name (2 letter code) [US]:JP	←JP(日本)を入力
State or Province Name (full name) [Some-State]:TOKYO	←TOKYO(東京)を入力(任意)
Locality Name (eg, city) []:SHINAGAWA	←SHINAGAWA(品川)を入力(任意)
Organization Name (eg, company) [Unconfigured OpenSSL Installation]:FUJITSU	←FUJITSUを入力(任意)
Organizational Unit Name (eg, section) []:INFRA	←INFRAを入力(任意)
Common Name (eg, YOUR name) []:websv	←ホスト名を入力
Email Address []:websv@example.co.jp	←メールアドレスを入力(任意)

## 5) RSA パラメータ生成機能を使用してパスフレーズを取り除く。

```
# openssl rsa -in /etc/apache2/ssl.key/server.key -out /etc/apache2/ssl.key/server.key
Enter pass phrase for /etc/apache2/ssl.key/server.key:XXXXXXX ←3)で入力したパスフレーズ
を入力
writing RSA key
```

## 6) Apache 用認証鍵の確認

```
# ls -l /etc/apache2/ssl.key/server.key
-rw-r--r-- 1 root root 891 Mar 16 15:10 /etc/apache2/ssl.key/server.key

# cat /etc/apache2/ssl.key/server.key
-----BEGIN RSA PRIVATE KEY-----
MIICXwIBAAKBgQC6iT7powhl3Xa0bVwQPQRyNvvoZ2NsZY4DwvmyfyNpe16M52oG
4JkR3R2AmmxgVu92cJ2kDnmmgvokuR8dKZnN/S8C4dv6Jz9vdGRkfRLUNkVV0XI
UXIXqTNZGbrOQI4IQX2erIbR+wPanYlasVFhoR+NxAzt7DYdfdjWJ6B6MQIDAQAB
AoGBAKxIMEv3IL18YrNXwczZni6i69yFCMXHsf1b8tVM26r2/LFIR6O97ERCLuL
ZIdCI96/OkmYePC6Ta/3Lm5vyrVq9eDo8psa7JDzDD5qp9ehe131+yMmNTFSQ7gZ
adToYEiIMLGYgS11LLshftmGyHFS4TLSSF5TtJHEY6VZ7DxBakEA2ppJVVstbTfC
XP97Zc8DMMIIGuY4uc2zuX9sioDID/Ja/PnHSUyuVlzC4VxizySmC5IqjW7DckEW
x4IETKLROQJBANpym3x3Z29I/0WcvDsd1STdqVYUOx0Bc4yc0UM0FNw0NVze3Nnj
MjHpwISNP/71w1kM1PFEIuACGw5FN9pZiLkCQQDBSEymCvodaYHG7HXUzLbLeiaV
JB3f2wsf5pS32cOu1+CvcMJOAyaX2ZIEIGIGYhfFSTrLJZO/VIA0OqwP/2xAkEA
pA58jY7pBoJpDju4leFCySwb18Tb3F3RrOG9G26K2BHxnazWSmEhunCnfJ6UWGO0
cTV82zG66cNLFH2Rg8/IKQJBAJ2uAUcs5c7huRLZnb4sRXIWaREG9+MO32K+Ja3I
fLbzqkFBIQKu58rMMxN2adI0+eO26gKrtDnRnXghQTF/FP4M=
-----END RSA PRIVATE KEY-----
```

## 7) /etc/apache2/ssl.conf の編集

```
# vi /etc/apache2/ssl.conf
:
# General setup for the virtual host
DocumentRoot "/var/apache2/htdocs"
ServerName websv:443 ←[ホスト名]:443 を設定します
ServerAdmin root@wevsv.example.co.jp ←管理者メールアドレスを設定
ErrorLog /var/apache2/logs/error_log
TransferLog /var/apache2/logs/access_log
```

## 8) Apache2 のサービス構成 httpd/ssl プロパティを true に設定

```
# svccfg -s apache2
svc:/network/http:apache2> setprop httpd/ssl = boolean: true
```

←プロパティに true を設定します

## 9) 変更後のプロパティ値を確認

```
svc:/network/http:apache2> listprop httpd/ssl
httpd/ssl    boolean true
```

←true が設定されていること

```
svc:/network/http:apache2> end
```

## 10) サービス構成の再読み込み

```
# svcadm refresh svc:/network/http:apache2
```

## 11) サービスの再起動

```
# svcadm restart svc:/network/http:apache2
```

## 12) サービスの状態確認

```
# svcs apache2
STATE          STIME      FMRI
online         16:20:10  svc:/network/http:apache2
```

←STATE が online になっていること

## 13) Apache の SSL 受付確認

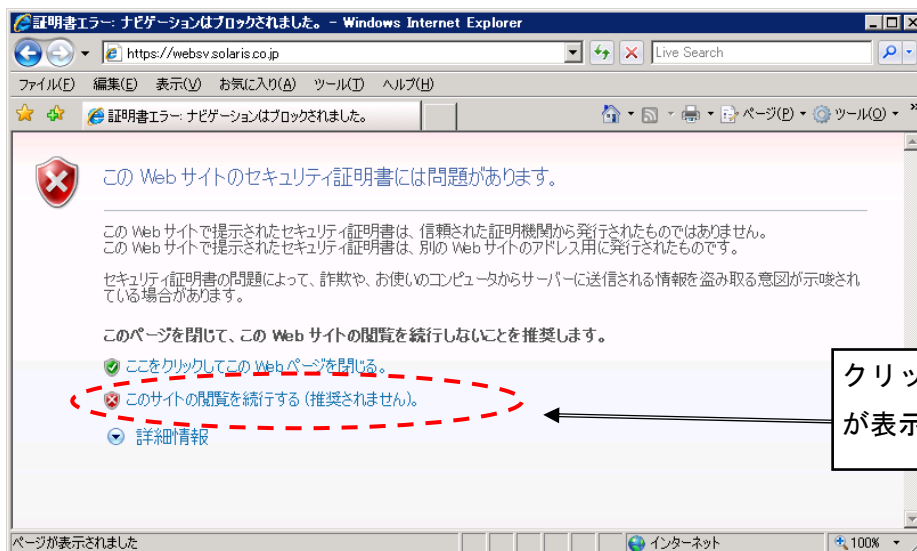
```
# ps -ef | grep http
root 4990 4988 0 15:14:45 pts/2      0:00 grep http
webservd 4903 4861 0 15:14:05 ?          0:00 /usr/apache2/bin/httpd -k start -DSSL
webservd 4901 4861 0 15:14:05 ?          0:00 /usr/apache2/bin/httpd -k start -DSSL
root 4861 4361 0 15:14:03 ?          0:01 /usr/apache2/bin/httpd -k start -DSSL
webservd 4902 4861 0 15:14:05 ?          0:00 /usr/apache2/bin/httpd -k start -DSSL
webservd 4900 4861 0 15:14:05 ?          0:00 /usr/apache2/bin/httpd -k start -DSSL
webservd 4904 4861 0 15:14:05 ?          0:00 /usr/apache2/bin/httpd -k start -DSSL
```

※プロセス名の最後に「-DSSL」が付いていれば SSL で Apache が起動しています。

## 7-2. ブラウザからの確認

Web ブラウザから Web サーバへ SSL を使用して接続できることを確認します。URL に下記を入力します。

https://ホスト名



クリックすると Apache の初期画面が表示されます。

## 8. バーチャルホストの設定

バーチャルホスト (Virtualhost) とは 1 台のサーバ上に複数の Web サイトを構成する機能です。バーチャルホストの運用方法には、NAME ベースと IP ベースの 2 種類の方法があります。両方式とも DNS サーバの設定が必要です。

### ➤ NAME ベース

NAME ベースのバーチャルホストは、Web ブラウザが Web サーバに対して送るホスト名を基にして応答するホストを決定する方式です。1 つの IP アドレスで、複数の Web サイトを構成することが可能です。

### ➤ IP ベース

IP ベースのバーチャルホストは、IP アドレスでホストを区別する方式です。1 つのサーバ上に複数の IP アドレスを設定して、複数の Web サイトを構成することが可能です。

### 8-1 .NAME ベース

設定例)

	ドメイン 1	ドメイン 2
ドメイン名	example.co.jp	sparc.co.jp
IP アドレス	10.20.112.202	10.20.112.202
DocumentRoot	/export/www	/var/apache2/htdocs
ServerAdmin	websv@example.co.jp	websv@sparc.co.jp

#### 1) httpd.conf の編集

バーチャルホストのサンプル設定の部分はコメントアウトされているため、解除して使用します。サーバ名 (ServerName) とドキュメントルート (DocumentRoot) を編集します。

```
# vi /etc/apache2/httpd.conf
:
NameVirtualHost 10.20.112.202:80          ←サーバのホスト名[or IP アドレス]を指定

# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for requests without a known
# server name.

<VirtualHost 10.20.112.202:80>
    ServerAdmin websv@example.co.jp      ←1つめの管理者メールアドレスを指定
    DocumentRoot /export/www            ←1つめのドキュメントルートを指定
    ServerName websv.example.co.jp      ←1つめのホスト名を指定
    ErrorLog /var/apache2/logs/dummy-host.example.com-error_log
    CustomLog /var/apache2/logs/dummy-host.example.com-access_log common
</VirtualHost>
```

```

<VirtualHost 10.20.112.202:80>
  ServerAdmin websv@sparc.co.jp           ←2つめの管理者メールアドレスを指定
  DocumentRoot /var/apache2/htdocs       ←2つめのドキュメントルート指定
  ServerName websv2.sparc.co.jp         ←2つめのホスト名(別名)を指定
  ErrorLog /var/apache2/logs/dummy-host.example.com-error_log
  CustomLog /var/apache2/logs/dummy-host.example.com-access_log common
</VirtualHost>

```

## 2) サービスの再起動

Apache2 のサービスを再起動します。

```
# svcadm restart svc:/network/http:apache2
```

## 3) サービスの状態確認

```
# svcs svc:/network/http:apache2
STATE          STIME      FMRI
online         13:43:46  svc:/network/http:apache2
```

### ※DNS サーバの設定

NAME ベースのバーチャルホストを使用する場合は DNS サーバに登録されている Web サーバのホスト名に別名 (CNAME) を設定します。

設定例)

```
websv.solaris.co.jp. IN  A      10.20.112.202
websv2.sparc.co.jp.  IN  CNAME  websv.solaris.co.jp.
```

## 8-2 .IP ベース

設定例)

	ドメイン1	ドメイン2
ドメイン名	example.co.jp	sparc.co.jp
IP アドレス	10.20.112.202	10.20.112.203
DocumentRoot	/export/www	/var/apache2/htdocs
ServerAdmin	websv@example.co.jp	websv@sparc.co.jp

### 1) httpd.conf の編集

バーチャルホストのサンプル設定はコメントアウトされているため、解除して使用します。サーバ名とドキュメントルートを編集します。

```
# vi /etc/apache2/httpd.conf
:
NameVirtualHost 10.20.112.202:80
                                ←7-1.NAME ベースで指定済みのためここでは修正の必要はありません

# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for requests without a known
# server name.

<VirtualHost 10.20.112.202:80>
    ServerAdmin websv@example.co.jp           ←1つめの管理者メールアドレスを指定
    DocumentRoot /export/www                  ←1つめのドキュメントルート指定
    ServerName websv.example.co.jp           ←1つめのホスト名を指定
    ErrorLog /var/apache2/logs/dummy-host.example.com-error_log
    CustomLog /var/apache2/logs/dummy-host.example.com-access_log common
</VirtualHost>

<VirtualHost 10.20.112.203:80>
    ServerAdmin websv@sparc.co.jp             ←2つめの管理者メールアドレスを指定
    DocumentRoot /var/apache2/htdocs          ←2つめのドキュメントルート指定
    ServerName websv2.sparc.co.jp            ←2つめのホスト名を指定
    ErrorLog /var/apache2/logs/dummy-host.example.com-error_log
    CustomLog /var/apache2/logs/dummy-host.example.com-access_log common
</VirtualHost>
```

## 2) サービスの再起動

```
# svcadm restart svc:/network/http:apache2
```

## 3) サービスの状態確認

```
# svcs svc:/network/http:apache2
STATE      STIME    FMRI
online     15:43:58 svc:/network/http:apache2
```

## ※DNS サーバの設定

IP ベースのバーチャルホストを使用する場合は、DNS サーバに各 IP アドレスに対応したホスト名が正引き、逆引きともに設定されている必要があります。

設定例)

example.co.jp ドメインの正引き用

```
websv IN      A      10.20.112.202
```

sparc.co.jp ドメインの正引き用

<b>websv</b>	<b>IN</b>	<b>A</b>	<b>10.20.112.203</b>
逆引き用			
<b>202</b>	<b>IN</b>	<b>PTR</b>	<b>websv.example.co.jp.</b>
<b>203</b>	<b>IN</b>	<b>PTR</b>	<b>websv.sparc.co.jp.</b>

以上、『Solaris フリーソフトウェア導入手順書 –Apache 編–』終了



