WebOTX Web サーバ 運用ガイド

WebOTX Web サーバ運用ガイド

バージョン： 7.1

版数： 第四版

リリース： 2009 年 11 月

目次

1.はじめに

WebOTX Application Server では、Web サーバ層の提供機能として、Java ベースの内蔵 Web サーバと、Apache HTTP Server ベースの WebOTX Web サーバをバンドルしています。

本書では Apache HTTP Server ベースの WebOTX Web サーバを運用するための運用操作法について概要や具体的な設定項目や設定方法について記載しています。

対象読者

このマニュアルはWebOTX Application Serverを使って運用環境を構築するシステムエンジニア、日々の運用を行うオペレータを対象としています。

表記について

パス名表記

本書ではパス名の表記については特に OS を限定しない限りセパレータはスラッシュ'/'で統一しています。Windows 環境においては'¥'に置き換えてください。

環境変数表記

インストールディレクトリやドメインルートディレクトリなど環境によって値の異なるものについては環境変数を用いて表します。

${*env*} または $(*env*) で表しています。

例）
${AS_INSTALL} : インストールディレクトリ
${INSTANCE_ROOT}: ドメインルートディレクトリ

コマンド操作について

本書中では運用操作に用いるコマンドの詳細についての説明は省略しています。

コマンドの詳細は「運用管理コマンド」、「運用管理コマンドリファレンス」を参照してください。

2. 機能

ここでは、WebOTX Web サーバが提供する機能について説明します。

2.1.機能概要

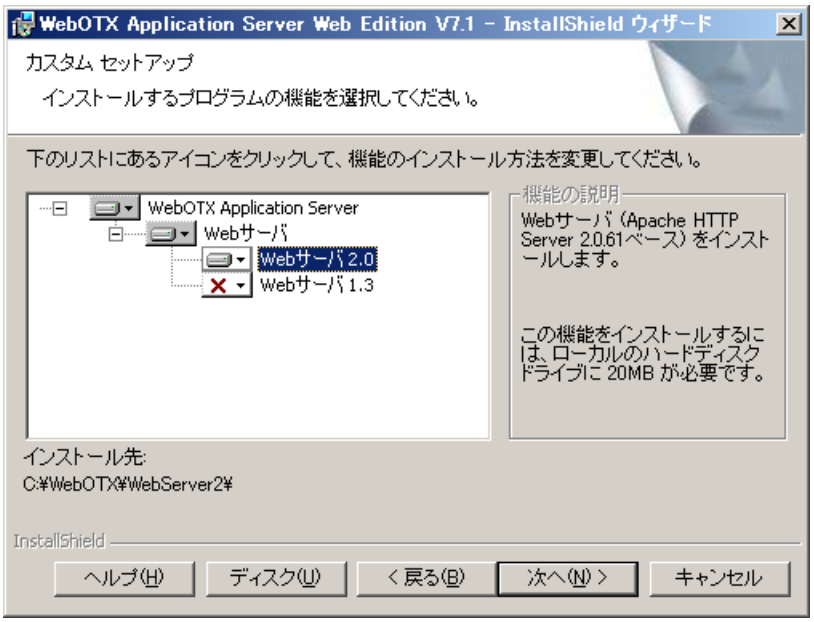
WebOTX Web サーバとは、WebOTX Application Server の Web サーバ層の機能を提供しており、デファクトスタンダードである Apache HTTP Server の次のバージョンをバンドルしています。(2007 年 12 月現在)

- Apache 1.3.39
- Apache 2.0.61

WebOTX Web サーバでは、Apache HTTP Server で提供される機能に加え、次の機能を提供しています。

- SSL 通信のサポート
- WebOTX Application Server 連携用モジュールの提供
- IPv6 環境での動作サポート (Apache 2.0 のみ)
- WebOTX 運用管理ツール／コマンドから定義情報の参照･更新 (WebOTX V7.11 以降)

利用者は、WebOTX のインストール時に、上記のどちらかのバージョンを選択して、WebOTX Web サーバをインストールすることができます。次の例は、Windows 版の WebOTX Application Server Web Edition のインストール時に「Apache 2.0 ベース」のインストールを選択している図となります。



なお、WebOTX Application Server では、本 Web サーバの他に、Java ベースの Web サーバを内蔵しており、そちらも利用可能です。さらに、Internet Information Services(IIS) や、Apache HTTP Server 1.3.39 以降、2.0.61 以降、Sun Java System Web Server 6.1、Sun ONE Web Server 6.0 以降と連携させて動作サポートしています。

2.2.基本機能

WebOTX Web サーバは、Apache HTTP Server が提供する Web サーバ動作に関する基本機能を提供します。

主に次の機能があります。

- HTTP/1.1　サポート
- Basic 認証/Digest 認証
- 仮想ホスト機能
- クライアントアクセス制御
- CGI スクリプト実行機能
- ログ出力機能　等

WebOTX Web サーバは、Apache HTTP Server が提供する Web サーバ動作に関する基本機能をすべて提供します。

2.3.SSL 通信機能

SSL (Secure Sockets Layer) は、公開鍵暗号方式を利用してデータの暗号化を行う。公開鍵と秘密鍵と呼ばれるキーの対を利用して、情報の暗号化と復号を行う。公開鍵は、特定のアルゴリズムを使用してデータを暗号化するためのものであり、他社に配布可能である。秘密鍵は、一般には配布せず、サーバ上に安全が保たれた状態で保管する必要がある。
SSL を使用してサイトにクライアントが接続すると、サーバは証明書の一部として公開キーとそれに付随する情報を送信し、クライアントが公開鍵暗号方式を利用してサーバの身元を確認する。

証明書は、認証局 (CA : Certificate Authority)によって発行された電子的はドキュメントであり、インターネット上で個人または企業の身元を保証するものであり、証明書にはサイトの公開鍵が含まれているため、クライアントはそれを利用して、サーバから送られてきたデータを復号できる。

WebOTX Web サーバでは、OpenSSL ライブラリを利用した mod_ssl モジュールを使用して、SSL2.0/3.0, TLS1.0 を利用して、かつ 128Bit 以上の暗号化方式をサポートしたセキュアな Web サイトを構築することができます。また、SSL クライアント認証機能も利用可能です。

クライアントが SSL を利用したセキュアなサイトにアクセスするには、次の形式の URL を指定します。

https://ホストアドレス[:ポート]/ホスト内資源アドレス

HTTPS 接続の場合、ポートは通常 443 が利用されます。ポート番号に 443 を利用する場合は省略が可能です。

2.4.LDAP 連携機能

LDAP (Lightweight Directory Access Protocol)サーバと連携して、HTTP 認証を LDAP エントリデータに登録されたユーザで行うきことができます。なお、この機能は、**Apache 2.0** でのみ提供します。

また、WebOTX Application Server では、LDAP サーバとして **EnterpriseDirectoryServer (EDS)** をバンドルしており、EDS に登録したユーザを利用して HTTP 認証を行うことができます。

2.5.IPv6 機能

IPv6 ネットワーク環境での動作をサポートします。この機能は、**Apache 2.0** でのみ提供します。
IPv6/IPv4 ネットワーク混在環境において、それぞれ別々の IP アドレス、ポートに対して待ち合わせが可能です。

待ち合わせ用のポート番号は、IPv6/IPv4 で同一にすることもできますし、別々に設定することもできます。

2.6.提供モジュール一覧

WebOTX Web サーバは、Apache HTTP Server で提供される次のモジュールを提供しています。デフォルトで組み込まれていないモジュールが提供する機能を利用する場合には、**LoadModule** 指示子により、モジュールのロードを行う必要があります。

Apache 2.0

モジュール	機能概要
(コアモジュール)	(デフォルトで組み込まれている機能。ロードする必要はありません)
http_core	サーバのコア機能を提供します。
Worker	(UNIX)UNIX 版の MPM モジュールは worker としています。Worker は、複数のスレッドを有するプロセスが複数個動作するモードです。クライアントから要求は、各スレッド上で受け付けを行い、処理を行います。
mod_access	クライアントのホスト名、IP アドレス、その他のクライアントのリクエストに基づいたアクセス制御機能を提供します。
mod_actions	メディアタイプやリクエストメソッドに応じて CGI スクリプトを実行する機能を提供します。
mod_alias	ホストファイルシステム上のいろいろな違う場所をドキュメントツリーにマップする機能と、URL のリダイレクトを行う機能を提供します。
mod_asis	自分用の HTTP ヘッダの書かれているファイルを送信します。
mod_auth	テキストファイルを用いたユーザ認証機能を提供します。
mod_autoindex	Unix の ls コマンドや Windows の dir シェルコマンドに似たディレクトリインデックスを生成します。
mod_cgid	外部 CGI デーモンを使用した CGO スクリプトを実行します。
mod_dir	URL に指定される「最後のスラッシュ」のリダイレクトと、ディレクトリのインデックスファイルを扱う機能を提供します。
mod_env	CGI スクリプト及び SSI ページに渡される環境変数を変更する機能を提供します。
mod_imap	サーバサイドのイメージマップを実行します。
mod_include	サーバがパースする html ドキュメント(Server Side Includes)
mod_log_config	サーバへのリクエストのロギングを行います。
mod_mime	リクエストされたファイルの拡張子とファイルの振る舞い(ハンドラとフィルタ)、内容(MIME タイプ、言語、文字セット、エンコーディング)とを関連付けます。
mod_negotiation	コンテントネゴシエーション機能を提供します。

mod_setenvif	リクエストの特徴に基づいた環境変数の設定を可能にします。				
mod_so	起動時や再起動時に実行コードとモジュールをサーバにロードします。				
mod_status	サーバの活動状況と性能に関する情報を提供します。				
mod_userdir	ユーザ専用のディレクトリを提供します。				
(オプションモジュール)	デフォルトで組み込まれていません。利用するには LoadModule 指示子により別途ロードする必要があります。 右欄は、**W**(Windows)、**H**(HP-UX)、**L**(Linux)、**S**(Solaris)を意味し、各 OS で提供しているモジュールに○をつけています。	**W**	**H**	**L**	**S**
mod_auth_digest	MD5 ダイジェスト認証を利用したユーザ認証機能を提供します。	○	○	○	○
mod_auth_ldap	LDAP ディレクトリに格納されたデータベースを利用して HTTP 基本認証を許可します。	○	○	○	○
mod_cache	URI をキーにしたコンテンツのキャッシュを行います。	○	―	―	―
mod_cern_meta	CERN httpd が使う追加の HTTP ヘッダ形式でメタ情報を指定できるようにします。	○	―	―	―
mod_cgi	CGI スクリプトを実行します。	○	―	―	―
mod_charset_lite	キャラクタセット	○	―	―	―
mod_dav	分散オーサリングとバージョン管理(WebDAV)機能を提供します。	○	○	○	○
mod_dav_fs	mod_dav のためのファイルシステムプロバイダを提供します。	○	○	○	○
mod_deflate	クライアントへ送られる前にコンテンツを圧縮します。	―	―	―	―
mod_disk_cache	URI をキーにしたコンテンツキャッシュストレージを管理します。	○	―	―	―
mod_dumpio	すべての I/O をエラーログにダンプします。	―	―	―	―
mod_echo	プロトコルモジュールの概要を示すための単純なエコーサーバを提供します。	―	―	―	―
mod_expires	ユーザの指定した基準に基づいた Expires と Cache-Control HTTP ヘッダの生成をします。	○	○	○	○
mod_ext_filter	レスポンスのボディをクライアントに送る前に外部プログラムで処理します。	○	○	○	○
mod_file_cache	メモリ内にファイルの静的なリストをキャッシュします。	○	―	―	―
mod_headers	HTTP リクエストヘッダとレスポンスヘッダをカスタマイズします。	○	○	○	○
mod_info	サーバの設定の包括的な概観を提供します。	○	○	○	○
mod_ldap (util_ldap)	LDAP 連携用モジュール。	○	○	○	○
mod_log_forensic	サーバに送られたリクエストを forensic ロギングします。	○	―	―	―
mod_logio	リクエスト毎に入力バイト数と出力バイト数をロギングします。	○	―	―	―
mod_mem_cache	URI をキーにしたコンテンツキャッシュします。	○	―	―	―
mod_mime_magic	ファイルの内容を読み込んで MIME タイプを決定します。	○	―	―	―

mod_proxy	HTTP/1.1 プロキシ/ゲートウェイサーバを提供します。	○	○	○	○
mod_proxy_connect	mod_proxy 関連モジュール。	○	○	○	○
mod_proxy_ftp	mod_proxy で FTP をサポートするモジュール。	○	○	○	○
mod_proxy_http	mod_proxy で HTTP をサポートするモジュール。	○	○	○	○
mod_rewrite	URL の書き換えを行うリライトエンジンを提供します。	○	○	○	○
mod_speling	ユーザが入力したであろう間違った URL を、大文字小文字の区別を無視することと一つ以下の綴り間違いを許容することで修正を試みます。	○	○	○	○
mod_ssl	SSL 通信用のモジュール。	○	○	○	○
mod_suexec	指定されたユーザとグループで CGI スクリプトを実行します。	－	－	－	－
mod_unique_id	それぞれのリクエストに対する一意な識別子の入った環境変数を提供します。	○	－	－	－
mod_usertrack	Cookie によりユーザの追跡を行います。	○	○	○	○
mod_version	バージョン依存の設定をします。	○	－	－	－
mod_vhost_alias	バーチャルホストに関する動的な設定を提供します。	○	○	○	○
mod_jk-20	Web コンテナと接続を行うコネクタモジュール。	○	○	○	○
mod_jk_om-20	**(WebOTX 独自)**マルチプロセス対応の Web コンテナと連携を行うためのコネクタモジュール。	○	○	○	○

Apache 1.3

モジュール	概要
(基本モジュール)	(デフォルトで組み込まれています。別途ロードする必要はありません。)
Core	Apache のコアモジュール
mod_access	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を行う。
mod_actions	メディアタイプやリクエストメソッドによって CGI スクリプトを実行します。
mod_alias	ホストファイルシステムのドキュメントツリーへのマッピング及び URL のリダイレクションを行います。
mod_asis	HTTP ヘッダを含むファイルを送信します。
mod_auth	テキストファイル形式の認証ファイルを使用したユーザ認証機能を提供します。
mod_autoindex	自動的にディレクトリー覧を作成します。
mod_cgi	CGI スクリプトを実行します。
mod_dir	ディレクトリの取り扱いについての、基本的な機能を提供します。
mod_env	CGI スクリプトに渡す環境変数の操作を行います。
mod_imap	イメージマップファイルを取り扱う機能を提供します。

mod_include	SSIドキュメントを有効にします。
mod_log_config	標準的な書式により、リクエストログを記録します。
mod_mime	ファイルの拡張子を利用してドキュメントタイプの判定を行います。
mod_negotiation	コンテキストネゴシエーション機能を提供します。
mod_setenvif	クライアントの情報を元に環境変数をセットします。
mod_so	実行時に Apache のモジュールを動的読み込みする機能を提供します。
mod_status	サーバの稼動状況を表示します。
mod_userdir	ユーザのホームディレクトリにアクセスする機能を提供します。
（オプションモジュール）	（デフォルトで組み込まれていません。利用するには、LoadModule 指示子を利用してモジュールをロードする必要があります。）
mod_ssl	OpenSSL ライブラリを利用した SSL 通信機能を提供します。
mod_headers	リソースに任意の HTTP ヘッダを加えます。
mod_rewrite	正規表現を利用した URI からファイル名への強力なマッピング機能を提供します。
mod_usertrack	Cookie によりユーザの追跡を行います。
mod_jk	Web コンテナとの接続を行うコネクタモジュール。
mod_jk_om	**(WebOTX 独自)**マルチプロセス対応の Web コンテナとの接続を行うコネクタモジュール。
mod_webotx	**(WebOTX 独自)**WebOTX 画面テンプレート機能を用いたサーバアプリケーションを WebOTX Web サーバ上で実行するためのモジュール。

2.7.その他の機能

その他の機能の詳細は、以下の Apache HTTP Server の Web サイトを参照してください。

http://httpd.apache.org/docs/1.3/

http://httpd.apache.org/docs/2.0/

3. 定義情報

WebOTX Web サーバの 定義情報は、定義情報ファイル(httpd.conf)に格納され、Web サーバ起動時に読み込まれます。
定義情報を更新した場合には、更新した定義情報を反映するには、WebOTX Web サーバの再起動が必要となります。

3.1.定義情報ファイル

WebOTX Web サーバの定義情報は、**httpd.conf** ファイルに格納されます。

定義情報ファイルは、WebOTX の ドメイン毎に格納されるため、ドメイン毎に定義情報を変更する必要があります。

格納場所
＜domain フォルダ＞/config/WebServer/httpd.conf

SSL 通信用ライブラリをインストールした場合、SSL 設定に関する定義情報は、**ssl.conf** ファイルに格納されます。

SSL 通信で利用するポート番号や証明書/秘密鍵ファイルを変更する場合、このファイルに定義された情報を変更する必要があります。

格納場所
＜domain フォルダ＞/config/WebServer/ssl.conf

3.2.基本定義

ここでは、WebOTX Web サーバが動作するために設定する必要がある必要最低限の定義情報について説明します。

定義情報ファイルには、次の定義が含まれている必要があります。

- ServerRoot
- Listen (または Port)
- User / Group
- DocumentRoot
- ErrorLog
- LoadModule

3.2.1.ServerRoot

名前
ServerRoot

説明

WebOTX Web サーバが動作するために必要とするディレクトリを設定します。この値は既定値以外の値に変更することはありません。

書式

ServerRoot *directory-path*

既定値

（Windows）
ServerRoot “C:/WebOTX/domains/domain1”

（UNIX）
ServerRoot /opt/WebOTX/domains/domain1

3.2.2.Listen

名前
Listen (または Port)

説明
Web サーバが利用するリクエスト受付用のポート番号を指定します。Port 指示子は、Apache1.3 でのみ有効で、ポート番号のみを設定します。Listen 指示子は Apache1.3/2.0 共に設定可能で、ポート番号だけの設定と、IP アドレスとポート番号を一緒に設定することができます。なお、UNIX 版において、インストール時に「**WebOTX 運用ユーザで利用する**」を指定した場合、1024 以下のポート番号は指定できません。

書式
Listen [*IP アドレス*:]*port*
Port *port*

設定例
・ポート番号 8080 を利用する場合の設定
Listen 8080

・IPv4 アドレスと IPv6 アドレスで別々のポート番号を指定する場合の設定(Apache2.0 のみ)
Listen 0.0.0.0:8081
Listen [::]8082

3.2.3.User/Group

名前
User
Group

説明

UNIX 版においてのみ有効です。WebOTX Web サーバがリクエストに応答する際に用いるユーザ／グループ ID を指定します。WebOTX Application Server のインストールにおいて、「WebOTX 運用ユーザ」を利用する選択をした場合には、この設定値も「WebOTX 運用ユーザ」に変更する必要があります。

書式
User *unix-userid*
Group *unix-groupid*

設定例
User otxadmin
Group otxadm

3.2.4.DocumentRoot

名前
DocumentRoot

説明

WebOTX Web サーバにアクセスした場合のルートディレクトリを指定します。

書式

DocumentRoot *directory-path*

設定例

(Windows)

DocumentRoot “C:¥WebOTX¥domains¥domain1¥docroot”

(UNIX)

DocumentRoot /opt/WebOTX/domains/domain1/docroot

3.2.5.ErrorLog

名前

ErrorLog

説明

WebOTX Web サーバ内部で発生したエラーを記録するファイル名を指定します。

書式

ErrorLog *errorlog-path*

設定例

ErrorLog logs/WebServer/error_log

3.2.6.LoadModule

名前

LoadModule

説明

Apache のモジュールをロードし、使用モジュールリストに追加します。

書式

LoadModule *module-name module-path*

設定例

LoadModule ssl_module /opt/WebOTX/WebServer2/modules/mod_ssl.so

3.3.サーバ動作に関する設定

ここでは、動作プロセス数に関する設定を説明します。

Windows 版のサーバ動作は、1つの子サーバプロセス上で、複数のスレッドが動作し、各スレッド上でクライアントからのリクエストを受け付けます。

UNIX 版のサーバ動作は、Apache1.3 の場合、複数の子プロセスが動作し、各子プロセス上でクライアントからのリクエストを受け付けます。つまり、1リクエストに対して1プロセスが割り当てられます。Apache2.0 の場合は、複数の子プロセスが動作し、さらに、その子プロセス上で複数のスレッドが動作します。各子プロセス上のスレッドでクライアントからのリクエストを受け付けます。つまり、1リクエストに対して1スレッドが割り当てられます。

そのため、Windows と UNIX で定義する内容が異なります。また、Apache 1.3 と 2.0 でも定義する内容が異なります。

サーバ動作に関連する主な設定は以下の指示子となります。

- MaxClients
- ThreadsPerChild
- MinSpareThreads / MaxSpareThreads
- StartServers
- ServerLimit / ThreadLimit
- MaxRequestsPerChild
- MinSpareServers / MaxSpareServers

3.3.1.MaxClients

名前
MaxClients

説明

WebOTX Web サーバが処理できる最大同時接続コネクション数を設定します。クライアントは、この値を超えて同時に接続することはできません。UNIX 版の Apache 1.3 の場合、この値は、リクエストに応じるために起動される子プロセスの最大数となります。UNIX 版の Apache2.0 の場合、リクエストに応じることのできるスレッドの総数となります。Windows 版の場合、この設定値は無効です。

書式

MaxClients *number*

既定値
MaxClients 150

3.3.2.ThreadsPerChild

名前
ThreadsPerChild

説明

Windows と UNIX で意味が異なります。Windows の場合、プロセス内で動作するスレッド数となり、WebOTX Web サーバが処理できる最大同時接続コネクション数を意味します。
UNIX の Apache1.3 場合、この値は意味を持ちません。UNIX の Apache2.0 の場合、1つのプロセス内で動作するスレッド数を意味します。

書式

ThreadsPerChild *number*

既定値

(Windows Aapache2.0)
ThreadsPerChild 250

(Windows Apache1.3)
ThreadsPerChild 150

(UNIX Apache2.0)
ThreadsPerChild 25

3.3.3.MinSpareThreads / MaxSpareThreads

名前
MinSpareThreads
MaxSpareThreads

説明

(UNIX Apache2.0)アイドル状態であるスレッドの最小値、最大値を設定します。アイドル状態とは、リクエストを処理していない状態のtことです。アイドル状態では、この範囲内に総スレッド数が収まるように、起動しているプロセス数が調整されます。

書式

MinSpareThreads *number*
MaxSpareThreads *number*

既定値
MinSpareThreads 25
MaxSapareThreads 75

3.3.4.StartServers

名前
StartServers

説明
起動時に生成される子プロセス数を設定します。

書式
StartServers *number*

設定例/既定値
(UNIX Apache2.0)
StartServers 2
(UNIX Apache1.3)
StartServers 5

3.3.5.ServerLimit / ThreadLimit

名前
ServerLimit
ThreadLimit

説明
(UNIX Apache2.0) 子プロセスの上限値(ServerLimit)、子プロセス内で動作するスレッドの上限値(ThreadLimit)を設定します。

書式
ServerLimit *number*
ThreadLimit *number*

既定値
ServerLimit 16
ThreadLimit 64

3.3.6.MaxRequestsPerChild

名前
MaxReqeuetsPerChild

説明
個々の子サーバプロセスが稼働中に扱うリクエスト数の上限を設定します。ここで指定した数のリクエストを受け付けた後で、その子プロセスは終了します。0 を指定した場合、そのプロセスはリクエスト数の上限を超えたことにより終了することはありません。

書式
MaxRequestsPerChild *number*

設定例/既定値
MaxRequestsPerChild 0

3.3.7.MinSpareServers / MaxSpareServers

名前
MinSpareServers
MaxSpareServers

説明
(UNIX Apache1.3)アイドル状態である子プロセスが動作する最大／最小数を指定します。アイドル状態とは、リクエストを処理していない状態のことです。アイドル状態では、この範囲内に総プロセススレッド数が収まるように、起動しているプロセス数が調整されます。

書式
MinSpareServers *number*
MaxSpareServers *number*

設定例/既定値
MinSpareServers 5
MaxSpareServers 10

3.4.SSL

SSL 通信を行う場合、定義情報ファイルに定義する主な設定を説明します。SSL に関する設定は、ssl.conf ファイルに格納されています。ssl.conf ファイルは、SSL 通信用パッケージをインストールすることで追加されます。

- SSLEngine
- SSLCertificateFile
- SSLCertificateKeyFile
- SSLCACertificateFile
- SSLVerifyClient

3.4.1.SSLEngine

名前
SSLEngine

説明
SSL を有効にする。通常、<VirtualHost>内で使用され、特定の仮想ホストに対して SSL を有効にします。

書式
SSLEngine *On|Off*

設定例
```
<VirtualHost _default_:443>
  SSLEngine on
  …
</VirtualHost>
```

3.4.2.SSLCertificateFile

名前
SSLCertificateFile

説明
SSL で利用する証明書ファイルを設定します。

書式
SSLCertificateFile *path*

既定値
SSLCertificateFile /opt/WebOTX/WebServer2/conf/ssl.crt/sample.crt

3.4.3.SSLCertificateKeyFile

名前
SSLCertificateKeyFile

説明
SSL で利用する秘密鍵ファイルを設定します。

書式
SSLCertificateKeyFile *path*

既定値
SSLCertificateKeyFile /opt/WebOTX/WebServer2/conf/ssl.key/sample.key

3.4.4.SSLCACertificateKeyFile

名前
SSLCACertificateKeyFile

説明
クライアント認証を行う場合、SSL で利用する秘密鍵ファイルを設定します。

書式
SSLCertificateKeyFile *path*

既定値
SSLCertificateKeyFile /opt/WebOTX/WebServer2/conf/ssl.key/client.key

3.4.5.SSLVerifyClient

名前
SSLVerifyClient

説明

クライアント認証を行う場合、クライアントが証明書を SSL で利用する秘密鍵ファイルを設定します。

書式
SSLVerifyClient *level*

設定例
クライアント認証を行わない場合
SSLVerifyClient none
クライアントに有効な証明書を提示させる場合
SSLVerifyClient require

3.5.認証、アクセス制御

WebOTX Web サーバがサポートする基本認証機能とホストベースのクライアントのアクセス制御に関する設定を記述します。

- AuthType
- AuthName
- Order
- Allow/Deny

3.5.1.AuthType

名前
AuthType

説明
ユーザ認証の種類を設定します。Basic と Digest が設定可能です。

書式
AuthType Basic|Digest

既定値
AuthType Basic

3.5.2.AuthName

名前
AuthName

説明
HTTP 認証の認可領域の表示。ここで指定した文字列が、大部分のブラウザのパスワードダイアログに表示されます。

書式
AuthName *auth-domain*

設定例
AuthName “Input a username & password.”

3.5.3.Order

名前
Order

説明
デフォルトのアクセス可能な状態と、Allow と Deny が評価される順番を制御します。

書式
Order *ordering*

設定例
以下の例では、Allow from all の後で Deny from foo.domain.com が設定されるため、foo.domain.com 以外のクライアントからのアクセスを許可します。

```
Order Allow,Deny
Allow from all
Deny from foo.domain.com
```

以下の例では、Deny from foo.domain.com の後で Allow from all が設定されるため、すべてのクライアントからのアクセスを許可します。

```
Order Deny,Allow
Allow from all
Deny from foo.domain.com
```

3.5.4.Allow / Deny

名前
Allow
Deny

説明
サーバのある領域にアクセスできるホストを制御します。どのクライアントがサーバにアクセスできるかを制御します。
Allow はアクセスを許可するクライアントを、Deny はアクセスを拒否するクライアントを指定します。
すべてのクライアント(all)、ドメイン名(domain.com)、IP アドレス(12.34.56.78)、IP アドレスの一部(12.34)およびネットワーク/ネットワークマスクの対(12.34.0.0/255.255.0.0 または 12.34.0.0/16)を指定することができます。

書式
Allow from all|*host*|env=*env-variable*
Deny from all|*host*|env=*env-variable*

設定例
以下の例では、foo.domain.com サブドメインにあるクライアント以外の domain.com ドメインのすべてのクライアントからアクセスが許可されます。

```
Order Allow,Deny
Allow from apache.org
Deny from foo.apache.org
```

次の3つの設定は同じアドレス群に対するアクセス許可の設定となります。

```
Allow from 10.1
Allow from 10.1.0.0/255.255.0.0
Allow from 10.1.0.0/16
```

3.6.LDAP 連携

LDAP 連携する場合に必要となる定義情報を記載します。なお、LDAP 連携機能は Apache2.0 でのみ有効です。

- AuthLDAPUrl

3.6.1.AuthLDAPUrl

名前
AuthLDAPUrl

説明
LDAP 連携で使用する LDAP サーバの DN 情報を URL で指定します。
ldap://*host*:*port*/*basedn*?*attributes*?*scope*?*filter*

書式
AuthLDAPUrl *url*

設定例
ドメインの docroot に対するアクセスに対して LDAP 認証をかける場合、次の設定を行います。
<Directory /opt/WebOTX/domains/domain1/docroot>
AuthType Basic
AuthName “Input a username & password.”
AuthLDAPUrl ldap://ldap-server/dc=users,dc=webotx,o=NEC,c=JP?uid?sub
Require valid-user
</Directory>

連携する LDAP サーバ上に、DN 情報として、dc=users, dc=webotx, o=NEC, c=JP が設定されている場合、

3.7.ログ出力

WebOTX Web サーバが出力するログ情報に関する定義情報を記載します。

- CustomLog
- LogFormat
- LogLevel

3.7.1.CustomLog

名前
CustomLog

説明
ログファイルの名前と書式を設定します。

書式
CustomLog *file*|*pipe format*|*nickname*

既定値
CustomLog logs/WebServer/access_log common
CustomLog /opt/WebOTX/domains/domain1/logs/WebServer/ssl_request_log ¥
%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x ¥”%r¥” %b”

3.7.2.LogFormat

名前
LogFormat

説明
アクセスログに出力するログの出力書式を設定します。アクセスログ情報には、クライアントの IP アドレス、URL、送信バイト数、処理時間等を出力することができます。

書式
LogFormat *format nickname*

既定値

LogFormat “%h %l %u ¥”%r¥” %>s %b” common

%h　アクセスを行ったマシンのホスト名(または IP アドレス)が表示される。

%l　リモートログ名。通常は'-'が表示される。

%u　認証リモートユーザ名。認証処理を行った場合に、そのユーザ名が表示される。

%r　リクエストの最初の行。

%>s　ステータス。正常終了の場合、200 が出力される。

%b　HTTP ヘッダ以外のサーバからクライアントに送られたバイト数。

common　このログフォーマットを表すニックネーム。CustomLog 指示子に、このニックネームを指定することで
このニックネームで設定されたログフォーマット情報が出力される。

設定可能な値

指示子	説明
%a	リモート IP アドレス
%A	ローカル IP アドレス
%b	サーバからクライアントに送信されたバイト数(HTTP ヘッダ以外)。CLF 書式。すなわち、1 バイトも送信されなかった場合'-'が出力される。
%B	サーバからクライアントに送信されたバイト数(HTTP ヘッダ以外)。
%c	応答が終了したときに接続ステータス 'X' = 応答が終了する前に接続が異常終了 '+' = 応答が送られた後に接続を持続することが可能 '-' = 応答が送られた後に接続は切られる
%{FOOBAR}e	環境変数 FOOBAR の内容
%f	ファイル名
%h	リモートホスト名
%H	リクエストプロトコル
%{Foobar}i	サーバに送られたリクエストの Foobar: ヘッダの内容
%l	(もしあれば、identd からの)リモートログ名
%m	リクエストメソッド
%{Foobar}n	他のモジュールからのメモ “Foobar” の内容
%{Foobar}o	応答の Foobar: ヘッダの内容
%p	リクエストを扱っているサーバの正式なポート番号
%P	リクエストを扱った子プロセスのプロセス ID
%q	問い合わせ文字列(存在する場合は前に ? が追加される。そうでない場合は空文字列)
%r	リクエストの最初の行
%s	HTTP ステータスコード。内部でリダイレクトされたリクエストは、

	元々のリクエストのステータスが表示される。最後のステータスを表示する場合は %>S と設定する。正常終了の場合 200 が出力される。
%t	時刻。CLF の時刻の書式(標準の英語の書式)
%{format}t	Format で与えられた書式による時刻。
%T	リクエストを扱うのにかかった時間を 秒単位で切り捨て表示する。例えば、リクエストの処理時間が 1 秒未満の場合は '0' が表示される。
%u	リモートユーザ(auth による認証されたユーザ。ステータス(%s)が 401 の場合は意味がない可能性がある)
%U	リクエストされた URL パスで、クエリ文字列は含まない
%v	リクエストを扱っているサーバの正式な ServerName
%V	UseCanonicalName の設定によるサーバ名

3.7.3.LogLevel

名前
LogLevel

説明
エラーログに出力するログレベルを設定します。

書式
LogLevel *level*

既定値
LogLevel warn

なお、*level* には次のレベルを設定可能です。

レベル	説明	error.log への出力例
emerg	緊急 - システムが利用できない	Child cannot open lock file. Exiting (子プロセスがロックファイルを開けないために終了した)
alert	直ちに対処が必要	getpwuid: couldn't determine user name from uid (getpwuid: UID からユーザ名を特定できない)
crit	致命的な状態	socket: Failed to get a socket, exiting child (socket: ソケットの取得に失敗したため、子プロセスが終了した)
error	エラー	Permature end of script headers (スクリプトのヘッダが足りないままで終了した)
warn	警告	child process 1234 did not exit, sending anther SIGHUP

		(子プロセス 1234 が終了しないため、SIGHUP を再送した)
notice	重要な情報	httpd: caught SIGBUS, attempting to dump core in … (httpd: SIGBUS シグナルを受け、…へコアダンプを出力した)
Info	追加情報	Server seems busy, (you may need to increase MaxClients) … (サーバ負荷が高い、MaxClients の値を増やす必要があるかも)
debug	デバッグ	Opening config file… (設定ファイルを開いている…)

3.8.その他の定義情報

その他の定義情報の詳細については、次の URL を参照してください。

- http://httpd.apache.org/docs/1.3/mod/
- http://httpd.apache.org/docs/2.0/mod/

4. 運用

ここでは、WebOTX Web サーバの運用・操作方法について記載します。また、特定の機能を利用する場合の設定方法を記載します。

4.1.起動・停止

WebOTX Web サーバの起動・停止は、WebOTX Application Server のドメインの起動・停止に連動して動作します。

WebOTX のドメインの起動はサービスとして登録されていますので、特に Web サーバの起動・停止を意識する必要はありません。

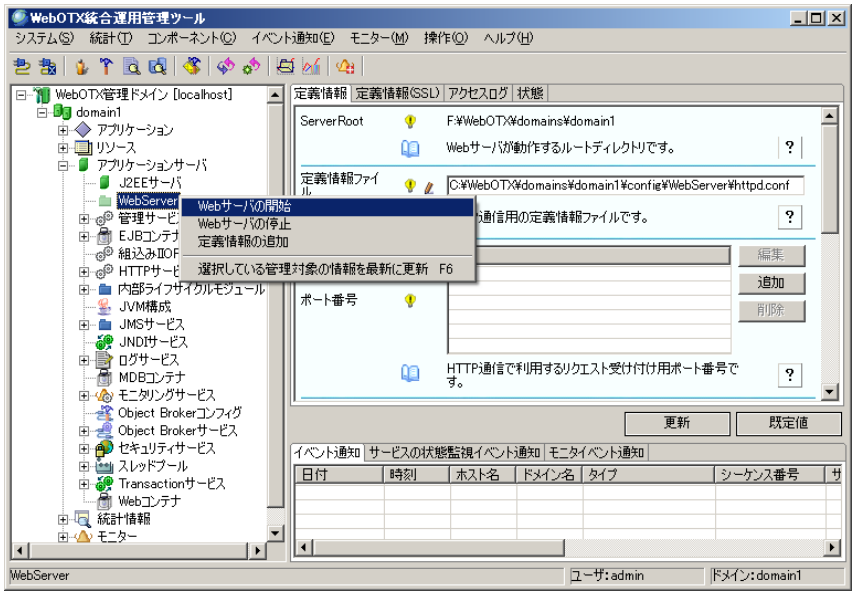
なお、WebOTX のドメインが起動している状態で、WebOTX Web サーバのみを単独で起動・停止を行う場合には、統合運用管理ツールから操作するか、次のコマンドを実行してください。

この方法は、WebOTX Web サーバの定義を変更した場合に有効です。

統合運用管理ツールからの操作

あらかじめ、統合運用管理ツールよりドメインと接続しておきます。

起動・停止方法



1. 「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「WebServer」を選択し、右クリックするか、あるいは、メニューバーの[操作] を選択します。

2. 表示されるメニューから「Web サーバの起動」を選択すると WebOTX Web サーバが起動します。また、「Web サーバの停止」を選択することで WebOTX Web サーバが停止します。

運用管理コマンド(otxadmin)からの操作

あらかじめ、otxadmin コマンドを起動し、ドメインにログインしておきます。

otxadmin>login --user admin --password adminadmin --port 6212

起動・停止方法

1. WebOTX Web サーバを起動するには、以下のコマンドを実行します。
   otxadmin>invoke server.WebServer.start

2. WebOTX Web サーバを停止するには、以下のコマンドを実行します。
   otxadmin>invoke server.WebServer.stop

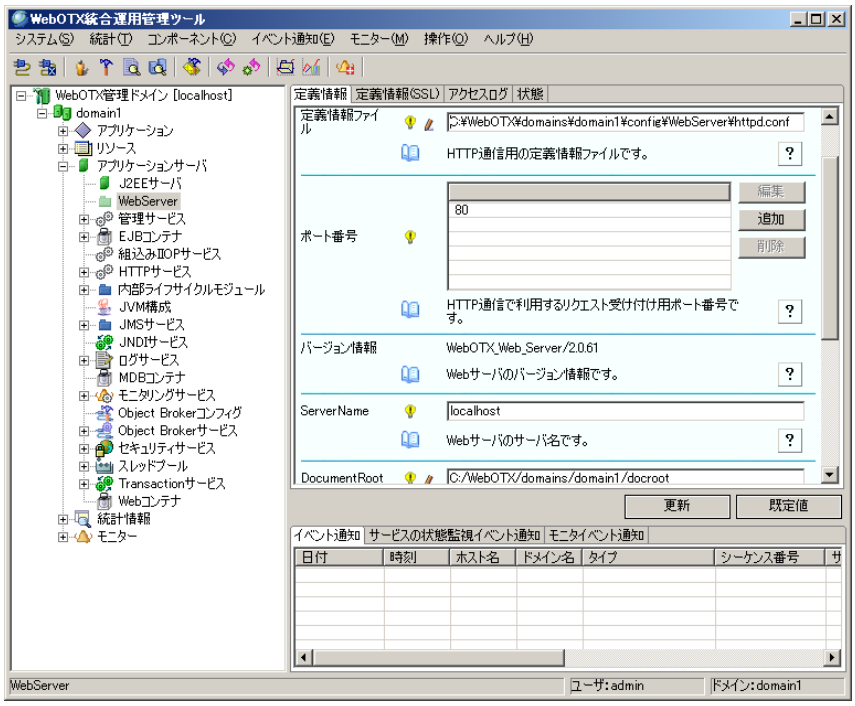
4.2.定義情報の参照

WebOTX Web サーバの定義情報のうち、動作に必要な一部の定義情報は、WebOTX の統合運用管理ツールおよび運用管理コマンドから参照することができます。

統合運用管理ツールからの操作

あらかじめ、統合運用管理ツールよりドメインと接続しておきます。

定義情報の参照



1. 「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「WebServer」を選択します。

2. 「**定義情報**」タブ、「**定義情報(SSL)**」タブ、「**アクセスログ**」タブを選択することで、各情報が各項目の情報を参照します。

運用管理コマンド(otxadmin)からの操作

あらかじめ、otxadmin コマンドを起動し、ドメインにログインしておきます。

otxadmin>login --user admin --password adminadmin --port 6212

定義情報の参照

1. WebOTX Web サーバの定義情報を取得するには、以下のコマンドを実行します。コマンドから参照可能な定義情報の一覧については、以下の表を参照してください。

   otxadmin>get server.WebServer.***

WebOTX 運用管理コマンド(ツール)から参照できる定義情報の一覧

統合運用管理ツールでの属性名	Server.WebServer.*	説明
ポート番号	Port	**Listen** 指示の設定値を取得します。WebOTX Web サーバが待ち合わせを行うポート番号を表します。
バージョン情報	Version	Web サーバのバージョン情報を表示します。この情報は httpd.conf には定義されていません。
ServerName	ServerName	**ServerName** 指示子の設定値を取得します。
DocumentRoot	DocumentRoot	**DocumentRoot** 指示子の設定値を取得します。ブラウザから見えるメインのドキュメントツリーになるディレクトリを表します。
ErrorLog	ErrorLog	**ErrorLog** 指示子の設定値を取得します。Web サーバのエラーログの出力先を表します。
LogLevel	LogLevel	**LogLevel** 指示子の設定値を取得します。Web サーバのエラーログへの出力レベルを表します。
最大同時接続数	MaxClients	UNIX 版の **MaxClients** 指示子、あるいは、Windows 版の **ThreadsPerChild** 指示子の設定値を取得します。最大同時接続数を表します。
SSL(HTTPS) 通信の使用の有無	security-enabled	SSL 通信を利用しているかのどうかの情報です。運用管理ツールでチェックされている(コマンドで true が返却された)場合、SSL 通信を利用します。運用管理ツールでチェックされていない(コマンドで false が返却された)場合、SSL 通信は利用していません。なお、この情報は httpd.conf に定義されていません。
HTTPS 通信用ポート番号	ssl-port	ssl.conf 内の **Listen** 指示子の設定値を取得します。SSL 通信用のポート番号を表します。
アクセスログ出力先と出力フォーマット	AccessLog	**CustomLog** 指示子の設定値を取得します。アクセスログの出力先と、出力するフォーマット情報のニックネーム値を表します。

「リクエスト処理時間(秒))情報の出力	AccesslogTat	**LogFormat** 指示子に リクエスト処理時間(%T)が設定されているかどうかの情報を取得します。運用管理ツールでチェックされている(コマンドで true が返却された)場合、設定されています。運用管理ツールでチェックされていない(コマンドで false が返却された)場合、設定されていません。
アクセスログのローテーション	Rotatelog	**CustomLog** 指示子にローテーション出力が設定されているかどうかの情報です。運用管理ツールでチェックされている(コマンドで true が返却された)場合、ローテーション出力が設定されています。運用管理ツールでチェックされていない(コマンドで false が返却された)場合、ローテーション出力は設定されていません。
ローテーション間隔	RotationTime	上記のアクセスログのローテーションが設定されている場合、そのローテーション時間を秒単位で表示します。既定値は 864000 秒(= 24 時間)です。

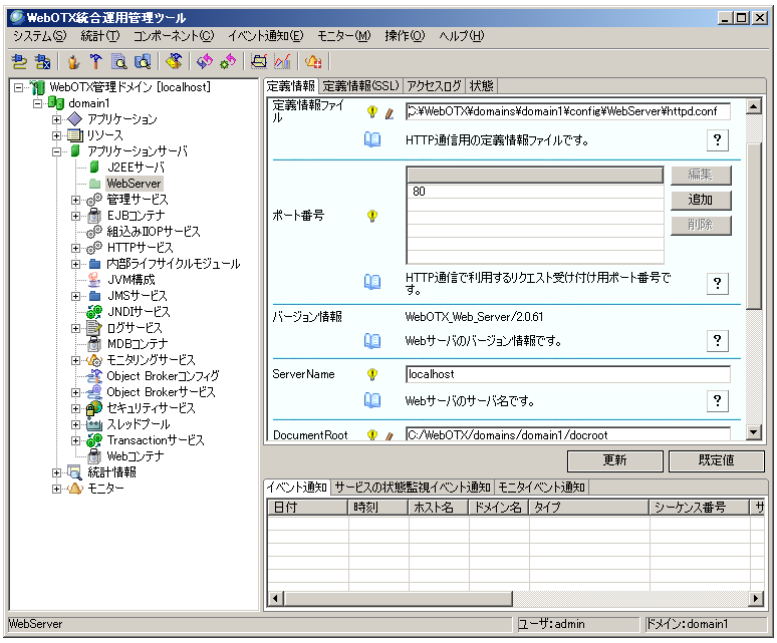
4.3.定義情報の更新

(**WebOTX V7.11 以降**) WebOTX Web サーバの定義情報のうち、動作に必要な一部の定義情報は、統合運用管理ツールおよび WebOTX のコマンドから更新することができます。

統合運用管理ツールからの操作

あらかじめ、統合運用管理ツールよりドメインと接続しておきます。

定義情報の更新



1. 「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「WebServer」を選択します。

2. 「**定義情報**」タブ、「**定義情報(SSL)**」タブ、「**アクセスログ**」タブの各項目を更新します。
   例えば、ポート番号を **8080** に変更する場合には、現在定義されているポート番号の項目(上記の例の場合、80 の項目)を選択し、「編集」のボタンを押下して、出力されたダイアログに **8080** を設定します。

3. 「更新」ボタンを押下することで、定義情報ファイルの情報が更新されます。

4. Web サーバの再起動または WebOTX の再起動を行うことで、更新された情報が反映されます。

運用管理コマンド(otxadmin)からの操作

あらかじめ、otxadmin コマンドを起動し、ドメインにログインしておきます。

otxadmin>login --user admin --password adminadmin --port 6212

定義情報の更新

1. WebOTX Web サーバの定義情報を更新するには、以下のコマンドを実行します。更新できる定義情報については、以下の表を参照してください。

   otxadmin>set server.WebServer.＊=*xxxx*

   例えば、ポート番号を **8080** に変更する場合には、以下のコマンドを実行します。

   otxadmin>set server.WebServer.port=8080

2. Web サーバの再起動または WebOTX の再起動を行うことで、更新された情報が反映されます。

WebOTX 運用管理コマンド(ツール)から更新できる定義情報の一覧

統合運用管理ツールでの属性名	Server.WebServer.*	説明
ポート番号	Port	**Listen** 指示子を設定します。WebOTX Web サーバが待ち合わせを行うポート番号を設定します。[*IP アドレス:*]*ポート番号* の形式での設定も可能です。複数の設定も可能です。
ServerName	ServerName	**ServerName** 指示子を設定します。
DocumentRoot	DocumentRoot	**DocumentRoot** 指示子を設定します。ブラウザから見えるメインのドキュメントツリーになるディレクトリを設定します。
ErrorLog	ErrorLog	**ErrorLog** 指示子を設定します。エラーログの出力先を設定します。
LogLevel	LogLevel	**LogLevel** 指示子を設定します。エラーログに出力するログレベルを設定します。
最大同時接続数	MaxClients	UNIX 版の **MaxClients** 指示子、あるいは、Windows 版の **ThreadsPerChild** 指示子を設定します。最大同時接続数を設定します。

SSL(HTTPS) 通信の使用の有無	security-enabled	SSL 通信を利用するかを指定します。SSL 通信を利用する場合、true を、利用しない場合には false を設定します。
HTTPS 通信用ポート番号	ssl-port	ssl.conf ファイル内の **Listen** 指示子を設定します。SSL 通信用で利用するポート番号を設定します。
アクセスログと出力フォーマット	AccessLog	**CustomLog** 指示子を設定します。アクセスログの出力先ファイル名と **LogFormat** 指示子で設定しているニックネームを設定します。
「リクエスト処理時間(秒)」の出力	AccesslogTat	**LogFormat** 指示子に リクエスト処理時間(”%T”)を追加します。つまり、アクセスログに「リクエスト処理時間(秒)」の情報を出力します。出力する場合には true を、出力しない場合には false を設定します。
アクセスログのローテーション	Rotatelog	**CustomLog** 指示子にローテーション設定を行い、アクセスログをローテーション出力するかどうかを設定します。ローテーション出力を行う場合 true を、行わない場合は false を設定します。
ローテーション間隔	RotationTime	アクセスログのローテーション設定を行う場合、ローテーション時間を秒単位で設定します。既定値は 86400(秒)、つまり 24 時間です。

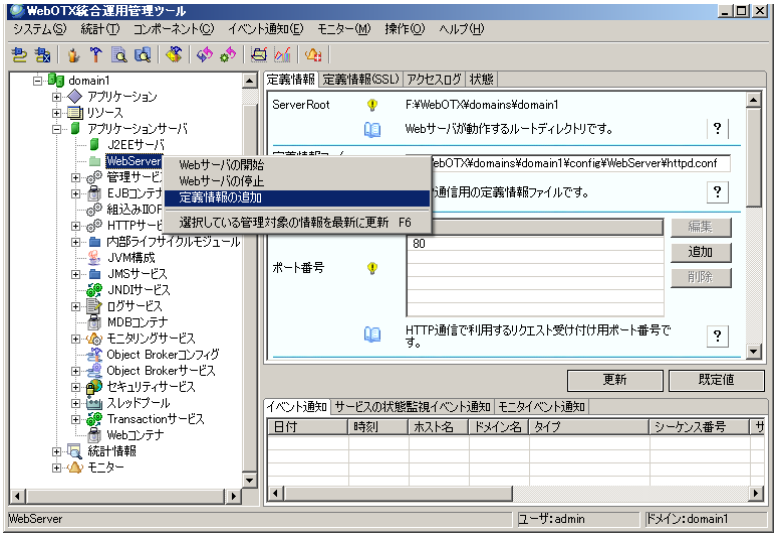
4.4.定義情報の追加

(**WebOTX V7.11 以降**) WebOTX Web サーバの定義情報のうち、定義情報に設定されていない情報を、統合運用管理ツールおよび WebOTX のコマンドから追加することができます。

統合運用管理ツールからの操作

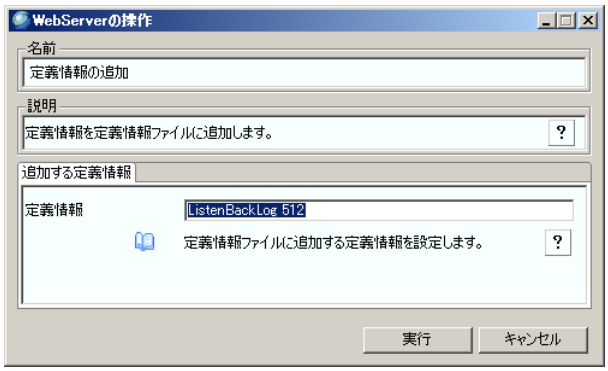
あらかじめ、統合運用管理ツールよりドメインと接続しておきます。

定義情報の追加



1. 「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「WebServer」を選択し、右クリックするか、あるいは、メニューバーの[操作] を選択します。

2. 表示されるメニューから「定義情報の追加」を選択すると、「定義情報の追加」ダイアログが表示されます。



3. 「追加する定義情報」に、追加する定義情報を設定します。定義情報は、<指示子> <設定値> の形式で指定する必要があります。なお、<指示子>だけの定義を設定する場合には、<指示子>の後に半角スペースを設定してください。

4. 「実行」のボタンを押下することで、設定した定義情報が、定義情報ファイルに追加されます。

5. Web サーバの再起動または WebOTX の再起動を行うことで、追加された情報が反映されます。

運用管理コマンド(otxadmin)からの操作

あらかじめ、otxadmin コマンドを起動し、ドメインにログインしておきます。

otxadmin>login --user admin --password adminadmin --port 6212

定義情報の追加

1. WebTX Web サーバの定義情報を更新するには、**server.WebServer.setDirective** コマンドを実行します。
   なお、追加する定義情報は、<指示子> <設定値>の形式で、ダブルコーテーションで囲む必要があります。
   例えば、**ListenBackLog 512** という定義情報を追加するには、次のように指定します。
   otxadmin>invoke server.WebServer.setDirective “ListenBackLog 512”

   また、**Win32DisableAcceptEx** のような<指示子>だけの定義を追加する場合には、<指示子>の後に半角スペースを付けて次のように指定します。
   otxadmin>invoke server.WebServer.setDirective “Win32DisableAcceptEx ”

2. Web サーバの再起動または WebOTX の再起動を行うことで、更新された情報が反映されます。

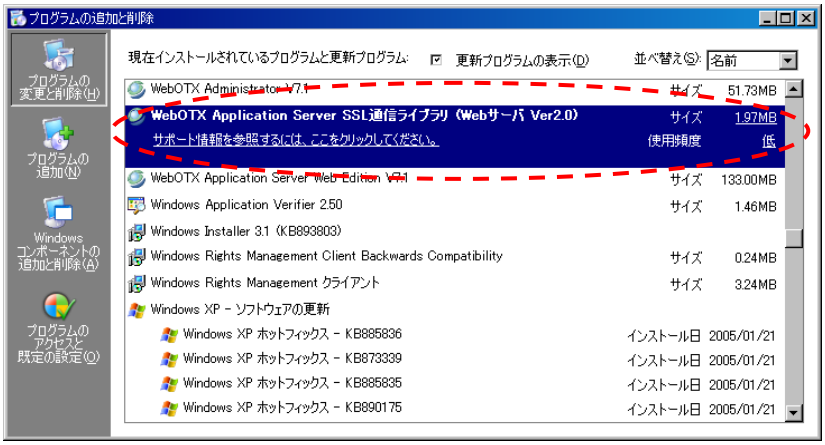
本節で説明している定義情報の追加処理は、Web サーバの定義情報ファイル(httpd.conf ファイル)に対してのみ有効です。
SSL 通信用の定義情報ファイル(ssl.conf)に対して、定義情報の追加処理を行うことはできません。

4.5.SSL 設定方法

WebOTX Web サーバは、OpenSSL ライブラリを利用した mod_ssl モジュールと連携することで、SSL プロトコルを利用した HTTPS 通信を実現することができます。HTTPS 通信を行うためには、次の設定が必要です。

1. SSL 通信用ライブラリのインストール

Windows 版の場合、WebOTX Web サーバのインストールを選択することで、SSL 通信用ライブラリも同時にインストールされます。SSL 通信用ライブラリがインストールされているかの確認は、「アプリケーションの追加と削除」から「**SSL 通信用ライブラリ(Web サーバ Ver1.3)**」または「**SSL 通信用ライブラリ(Web サーバ Ver2.0)**」がインストールされているかを確認してください。



UNIX版の場合、WebOTX のメディアから 次のパッケージを別途インストールします。

プラットフォーム	バージョン	インストールパッケージ
HP-UX(IPF)	1.3	/MODSSL/HP_UX/MODSSL
	2.0	/MODSSL/HP_UX/MODSSL2
Linux (x86)	1.3	/MODSSL/LINUX/modssl-2.8.xx.xx-1.i386.rpm
	2.0	/MODSSL/LINUX/modssl2-2.00.xx.xx-1.i386.rpm
Linux (x64)	1.3	/MODSSL/LINUX/modssl-2.8.xx.xx-1.i386.rpm (※)
	2.0	/MODSSL/LINUX/modssl2-2.00.xx.xx-1.x86_64.rpm
Solaris	1.3	/MODSSL/SUN/MODSSL
	2.0	/MODSSL/SUN/MODSSL2

（※）Linux(x86) と同一パッケージです。

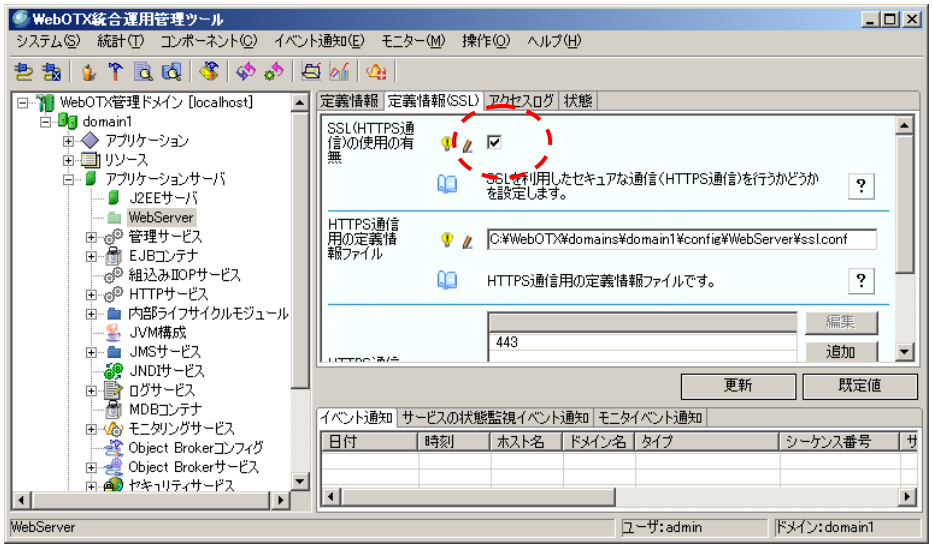
2. SSL 設定の有効化

SSL 通信用ライブラリをインストール後、SSL 通信機能を有効にするために、WebOTX Application Server の設定変更を行う必要があります。次の手順により、設定変更を行ってください。

統合運用管理ツールからの操作

あらかじめ、統合運用管理ツールよりドメインと接続しておきます。

SSL 通信の有効化



1. 「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「WebServer」を選択し、「**定義情報(SSL)**」タブの「**SSL(HTTPS 通信)の使用の有無**」をチェックします。

2. 「更新」ボタンを押下すると、SSL 設定が有効になります。SSL で利用するポート番号を変更する場合、「HTTPS 通信用の定義情報ファイル」の項目で表示されるファイルを編集してください。
   または、「HTTPS 通信用のポート番号」の項目を更新します。

3. WebOTX Web サーバを再起動することにより、SSL 設定が有効になります。

運用管理コマンド(otxadmin)からの操作

あらかじめ、otxadmin コマンドを起動し、ドメインにログインしておきます。

otxadmin>login --user admin --password adminadmin --port 6212

SSL 通信の有効化

1. WebOTX Web サーバの SSL 通信を有効化するには、以下のコマンドを実行します。
   otxadmin>set server.WebServer.security-enabled=true

2. SSL 通信用のポート番号を変更するには、以下のコマンドを実行します。
   例えば、8443 に変更する場合、次のコマンドを実行します。
   otxadmin>set server.WebServer.ssl-port=8443

3. WebOTX Web サーバを再起動します。
   otxadmin>invoke server.WebServer.stop
   otxadmin>invoke server.WebServer.start

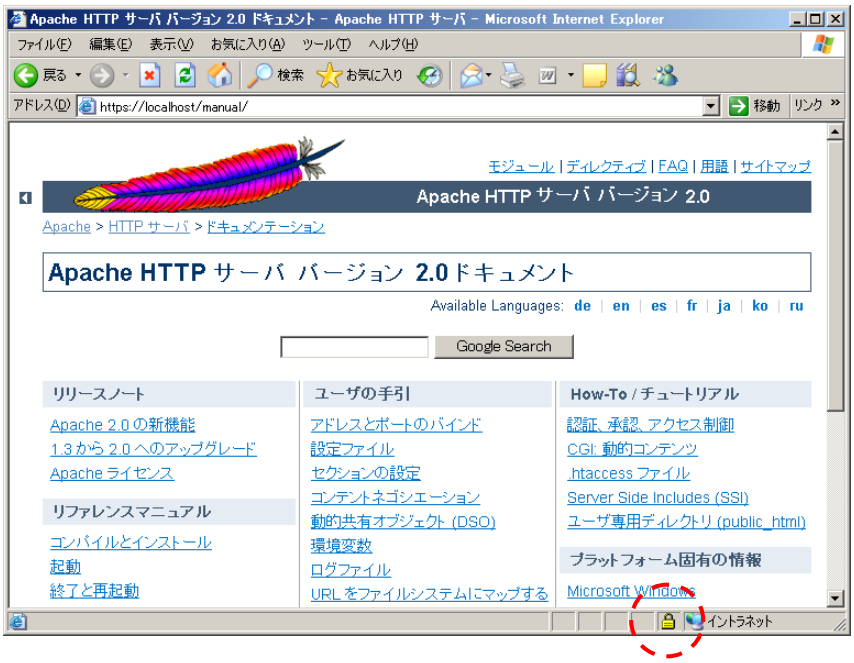
3. HTTPS 通信の接続確認

WebOTX Web サーバでは、SSL 通信用ライブラリをインストールすることで、HTTPS 接続評価用の証明書ファイルと秘密鍵ファイルが同時にインストールされます。したがって、インストール直後でもローカルマシンのブラウザから SSL での接続確認ができます。

1. ブラウザから次の URL を指定します。SSL 接続用のポート番号を変更している場合には、そのポート番号も指定してください。別マシンから接続確認を行う場合には、WebOTX をインストールしたホスト名を指定してください。

   https://localhost/manual/

2. 例えば、Apache2.0 を利用している場合、次のような画面が表示されれば、SSL で接続できたことが確認できます。
   また、ブラウザのステータスバーに SSL 接続中であることを意味する「鍵」マークが表示されていることを確認してください。

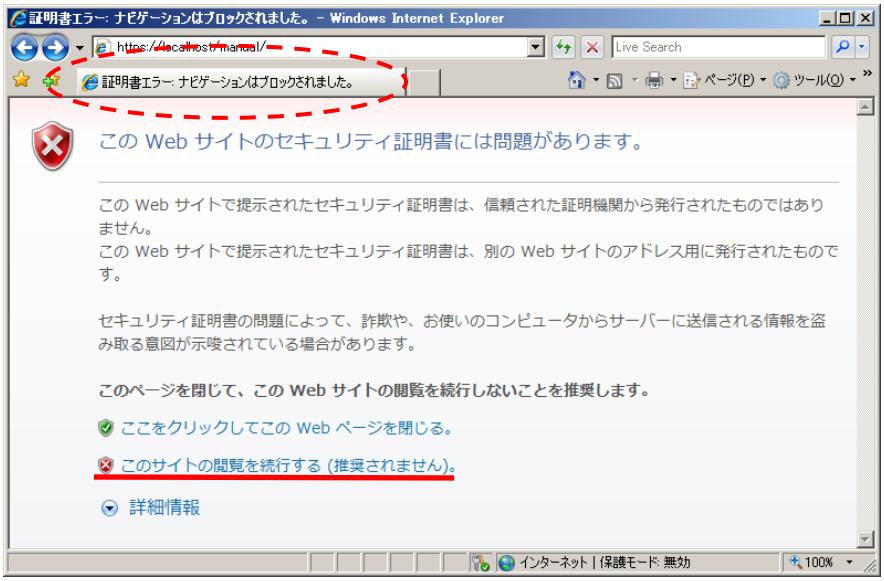


↑「鍵」マーク

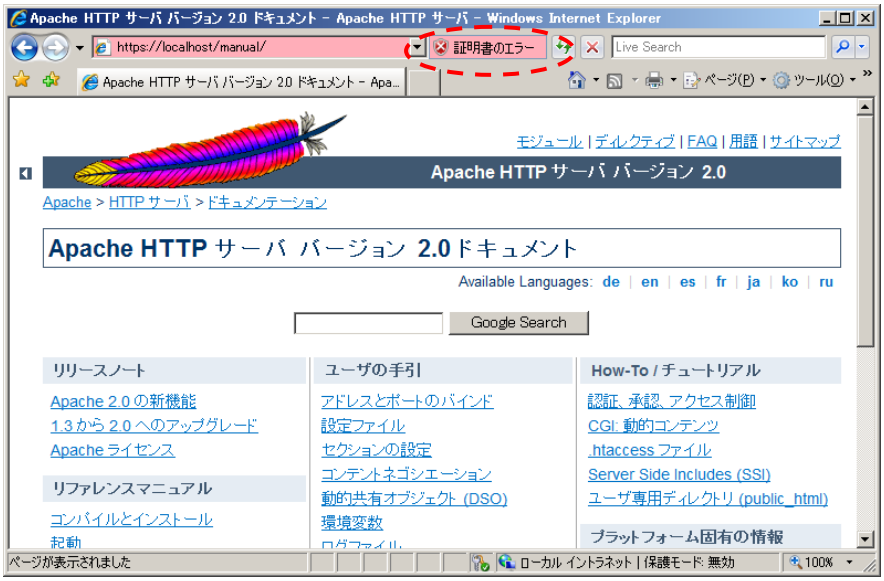
3. ブラウザに表示される「鍵」マークをクリックすることで、WebOTX Web サーバの SSL 通信機能で利用している証明書情報を参照することができます。ただし、WebOTX　Web サーバの SSL 通信ライブラリがインストールする本証明書は、接続確認用の自己署名証明書であるため、以下のように「**信頼された証明機関がこの証明書を確認できません**」と表示されます。「OK」ボタンを押下して証明書のダイアログを終了させてください。

4. なお、Internet Explorer 7 （IE 7）を利用した場合、次の画面(IE7 での HTTPS 接続画面−①)が表示されます。これは、IE 7 で証明書のチェックが厳しくなったために出力される情報であり、SSL での接続ができないという訳ではありません。**「このサイトの閲覧を続行する(推奨されません)。」**を選択すると、さらに次の画面(IE7 での HTTPS 接続画面−②)が表示され、アドレスバーに「証明書エラー」と表示されます。
本件は、信頼された証明機関から発行された正しい証明書を利用することで解決します。
次節に示す手順により、正しい証明書を入手してください。

IE7 での HTTPS 接続画面−①



IE7 での HTTPS 接続画面−②



4. 証明書の取得

次に示す手順は、CA 機関に対して証明書の発行を要求する手順の一例です。
この例では、Linux 上の OpenSSL コマンドを利用して、秘密鍵の生成と証明書署名要求の生成を行い、CA 機関に送付して証明書を取得し、WebOTX Web サーバへ設定を行うまでの手順を記載します。詳細については、各 CA 機関での証明書の取得方法(Apache の場合)を参照してください。

(ア) 秘密鍵の生成

/usr/local/openssl/private に 秘密鍵ファイル(server.key)を生成します。
キー生成のために、ランダムな情報が含まれている file1～file3 をあらかじめ用意しておいてください。

>openssl genrsa –rand file1:file2:file3 1024 –out /usr/local/openssl/private/server.key

生成された秘密鍵ファイルにアクセス権の設定を行います。

>chmod 400 /usr/local/openssl/private/server.key
>chmod 700 /usr/local/openssl/private

(イ) 証明書署名要求の生成

証明書著名要求(CSR)ファイルを生成し、CA 機関に送付します。

>openssl req –new –key server.key –out server.csr

(ウ) 証明書ファイルの取得

CA 機関から返信された証明書ファイル(server.crt)を /use/local/openssl/certs に格納し、アクセス権を設定します。

>chmod 400 /usr/local/openssl/certs/server.crt
>chmod 700 /usr/local/openssl/certs

(エ) 証明書と秘密鍵の設定

証明書ファイルと秘密鍵ファイルを、WebOTX Web サーバに設定します。
/opt/WebOTX/domains/domain1/conf/WebServer/ssl.conf の SSLCertificateFile 指示子に入手した証明書ファイルを、**SSLCertificateKeyFile** 指示子に秘密鍵ファイルを設定してください。
(ssl.conf)

```
SSLCertificateFile        /usr/local/openssl/certs/server.crt
SSLCertificateKeyFile   /user/local/openssl/private/server.key
```

(オ) パスフレーズの取得設定

秘密鍵作成時にパスフレーズを設定している場合、証明書にアクセスするためにパスフレーズの読み込み処理を設定しておく必要があります。**SSLPassPhraseDialog** 指示子を参照し、パスフレーズの設定を行ってください。また、パスフレーズの読み込み処理を行うスクリプト(例えば、次の pass.sh のようなシェルスクリプト)等をあらかじめ用意しておく必要があります。なお、Windows の場合には、パスフレーズなしで秘密鍵を作成してください。
(ssl.conf)

```
SSLPassPhraseDialog exec:/usr/local/openssl/private/pass.sh
```

<</usr/local/openssl/private/pass.sh(※) の内容>>

```
#!/bin/sh
echo “passphrase”
exit 0
```

(※)pass.sh はアクセス権を設定しておく必要があります。

>chmod 500 /usr/local/private/pass.sh

(カ) Web サーバの再起

WebOTX Web サーバまたはドメインの再起動を行います。

4.6.ログファイルのローテーション

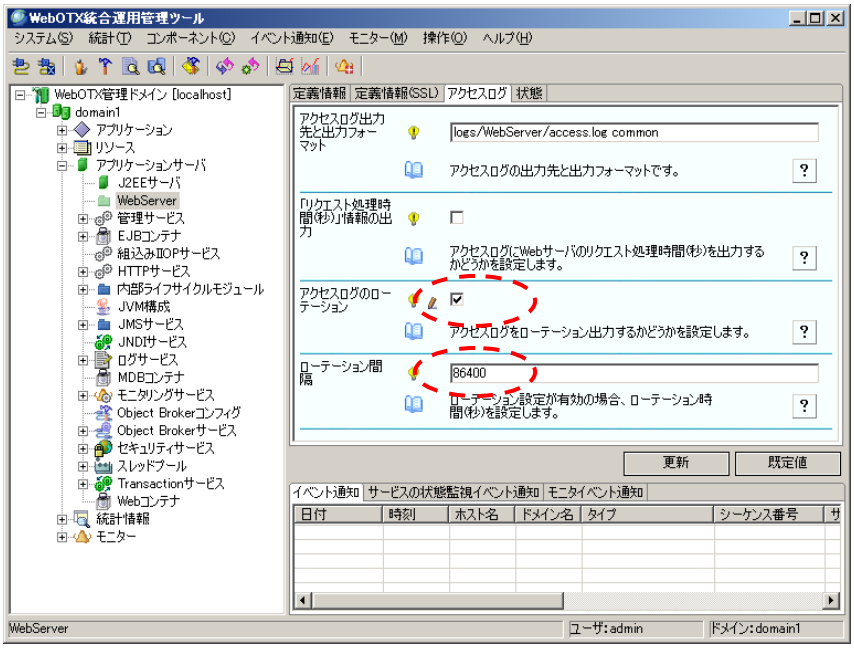
WebOTX Web サーバの出力するログファイルには、クライアントからのアクセス状況を出力する **access.log** と、Web サーバ本体側の動作に関連した情報を出力する **error.log** があります。

既定値の設定のままで WebOTX Web サーバを長時間動作させたままにすると、**access_log** に出力されるログ情報が蓄積されてディスク領域を大きく占有する場合があります。これを解消するために、access.log ファイルを時間でローテーションすることが考えられます。次の例では、access.log ファイルを 24 時間(86400 秒)でローテーション(1 日毎に access.log ファイルを作成)させる設定方法について記載します。
なお、統合運用管理ツール／運用管理コマンドからの操作は、WebOTX V7.11 以降から提供されている機能です。

統合運用管理ツールからの操作

あらかじめ、統合運用管理ツールよりドメインと接続しておきます。

アクセスログファイルのローテーション



1. 「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「WebServer」を選択し、「**アクセスログ**」タブの「**アクセスログのローテーション**」をチェックします。

2. 「**ローテーション間隔**」にローテーション時間を設定します。

3. 「更新」ボタンを押下することで、設定内容が定義情報ファイルに反映されます。

4. Web サーバを再起動することにより、設定内容が反映されます。

運用管理コマンド(otxadmin)からの操作

あらかじめ、otxadmin コマンドを起動し、ドメインにログインしておきます。

```
otxadmin>login --user admin --password adminadmin --port 6212
```

アクセスログファイルのローテーション

1. WebOTX Web サーバのアクセスログのローテーション設定を有効にするには、以下のコマンドを実行します。
   otxadmin>set server.WebServer.Rotatelog=true

2. ローテーション時間は既定値で 24 時間(=86400 秒)が設定されますが、ローテーション時間を変更するには、以下のコマンドを実行します。
   例えば、1週間(=604800 秒)でローテーションさせる場合は、次のコマンドを実行します。
   otxadmin>set server.WebServer.RotationTime=604800

3. 設定内容を反映するには、Web サーバの再起動が必要です。

上記の設定により、定義情報ファイルに次の設定が追加されます。
なお、統合運用管理ツール／運用管理コマンドからの操作ができない場合には、定義情報ファイルを直接編集し、次の設定を行ってください。
(UNIX:Apache2.0)

```
CustomLog “|/opt/WebOTX/WebServer2/bin/rotatelogs ¥
   /opt/WebOTX/domains/domain1/logs/WebServer/access_log 86400” common
```

(Windows:Apache2.0)

```
CustomLog “|C:/WebOTX/WebServer2/bin/rotatelogs.exe ¥
   C:/WebOTX/domains/domain1/logs/WebServer/access.log 86400” common
```

上記の設定により、Web サーバの再起動を実施することで、次のログファイルが順次生成されます。

access_log.1089207300

access_log.1083293700

access_log.1083380100

…

生成されたファイルのうち、小さい数字のものは過去のログとなりますので、ファイルの移動／削除等が可能となります。

なお、SSL 通信用の定義情報ファイル(ssl.conf)に定義されている **ssl_request_log** ファイルに対してローテーション設定を行う場合には、直接 ssl.conf ファイルを編集し、次の設定を行ってください。

(UNIX:Apache2.0)

```
CustomLog “|/opt/WebOTX/WebServer2/bin/rotatelogs ¥
   /opt/WebOTX/domains/domain1/logs/WebServer/ssl_request_log 86400” ¥
   “%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x ¥”%r¥” %b”
```

(Windows:Apache2.0)

```
CustomLog “|C:/WebOTX/WebServer2/bin/rotatelogs.exe ¥
   C:/WebOTX/domains/domain1/logs/WebServer/ssl_request.log 86400” ¥
   “%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x ¥”%r¥” %b”
```

4.7.アクセスログへリクエスト処理時間の出力

アクセスログに リクエスト処理時間の情報を出力することで、Web サーバがそのリクエストを受け付けて、レスポンスを返却するまでの時間を出力することができ、例えば、どのリクエスト(コンテンツ)に対する処理に時間がかかっているかを調査するのに役立つ場合があります。

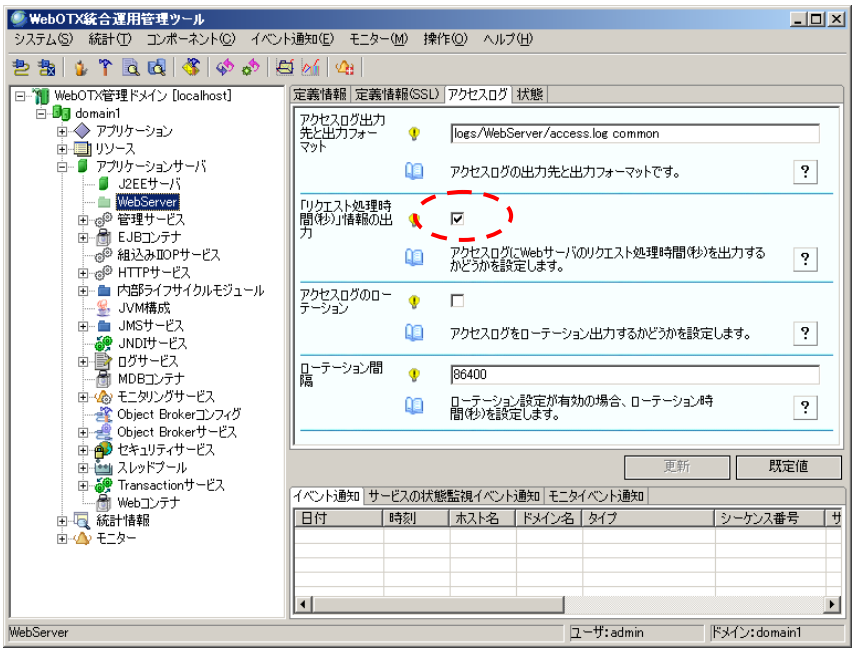
次の例では、access.log ファイルにリクエスト処理時間を出力する設定方法について記載します。

なお、統合運用管理ツール／運用管理コマンドからの操作は、WebOTX V7.11 以降から提供されている機能です。

統合運用管理ツールからの操作

あらかじめ、統合運用管理ツールよりドメインと接続しておきます。

リクエスト処理時間の情報出力



1. 「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「WebServer」を選択し、「**アクセスログ**」タブの「**リクエスト処理時間(秒)情報の出力**」をチェックします。

2. 「更新」ボタンを押下することで、設定内容が定義情報ファイルに反映されます。

3. Web サーバを再起動することにより、設定内容が反映されます。

運用管理コマンド(otxadmin)からの操作

あらかじめ、otxadmin コマンドを起動し、ドメインにログインしておきます。

otxadmin>login --user admin --password adminadmin --port 6212

リクエスト処理時間の情報出力

1. WebOTX Web サーバのアクセスログのローテーション設定を有効にするには、以下のコマンドを実行します。
   otxadmin>set server.WebServer.AccesslogTat=true

2. 設定内容を反映するには、Web サーバの再起動が必要です。

上記の設定により、定義情報ファイルに次の設定が追加されます。
なお、統合運用管理ツール／運用管理コマンドからの操作ができない場合には、定義情報ファイルを直接編集し、LogFormat 指示子に %T を追加してください。

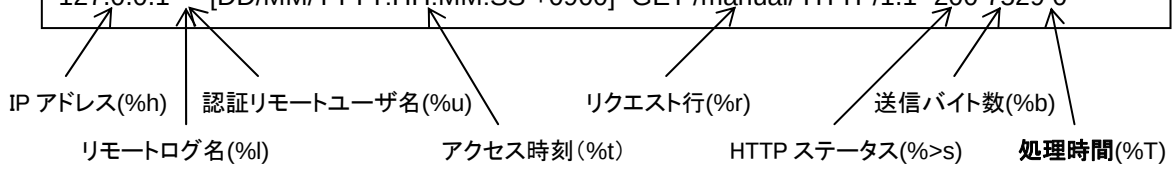
```
LogFormat   “%h %l %u %t ¥”%r¥” %>s %b %T” common
```

この設定により、アクセスログには次のログ情報が出力されます。最後の項目がリクエスト処理時間(秒)となります。

なお、**1 秒未満でリクエスト処理が完了した場合には、0 が表示されます。**

（アクセスログの出力内容例）

```
127.0.0.1 - - [DD/MM/YYYY:HH:MM:SS +0900] “GET / HTTP/1.1” 403 339 0
127.0.0.1 - - [DD/MM/YYYY:HH:MM:SS +0900] “GET /manual/ HTTP/1.1” 200 7529 0
```



4.8.最大同時接続数の変更方法

最大同時接続数を超えた場合、次のメッセージが error_log に出力されます。

（UNIX） **[error] server reached MaxClients setting, consider raising the MaxClients setting**

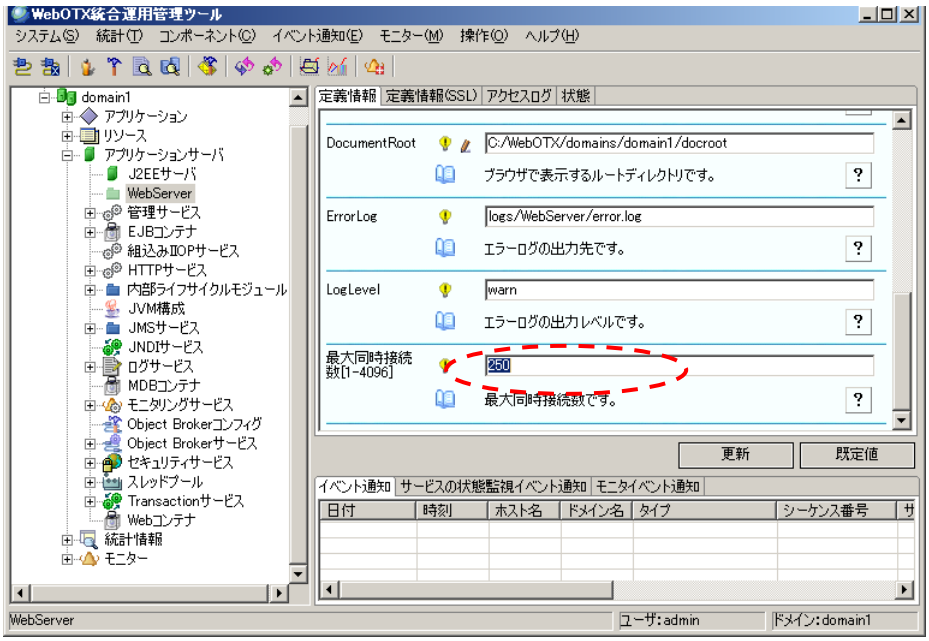
（Windows）**[warn] Server ran out of threads to serve requests. Consider raising the ThreadsPerChild setting**

最大同時接続数を増やすには、定義情報の次の設定を変更する必要があります。
なお、統合運用管理ツール／運用管理コマンドからの操作は、WebOTX V7.11 以降から提供されている機能です。

統合運用管理ツールからの操作

あらかじめ、統合運用管理ツールよりドメインと接続しておきます。

最大同時接続数の変更



1. 「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「WebServer」を選択し、「**定義情報**」タブの「**最大同時接続数**」の値を変更します。

2. 「更新」ボタンを押下することで、設定内容が定義情報ファイルに反映されます。

3. Web サーバを再起動することで、設定内容が反映されます。

運用管理コマンド(otxadmin)からの操作

あらかじめ、otxadmin コマンドを起動し、ドメインにログインしておきます。

otxadmin>login --user admin --password adminadmin --port 6212

最大同時接続数の変更

1. WebOTX Web サーバのアクセスログのローテーション設定を有効にするには、以下のコマンドを実行します。
   otxadmin>set server.WebServer.MaxClients=250

2. Web サーバを再起動することで、設定内容が反映されます。

なお、最大同時接続数の値を変更した場合は、次の設定も合わせて変更してください。

- **Web コンテナの最大プロセッサ数**
  (server.http-service.http-listener.ajp-listener-1.max-processors)

- **プラグインモジュールの最大リクエスト処理数**
  (ドメインの config/WebCont/workers.properties ファイルに定義する workers.ajp13.cachesize)

なお、定義情報ファイルを直接編集する場合には、次の設定を変更します。

- Windows の場合
  ThreadsPerChild 指示子を変更します。この値は、子プロセス内で起動するスレッド数となります。
  Windows 版では、クライアントから受け付けた1つのリクエストを1つのスレッド上で処理します。
  最大で 4096 まで設定可能です。既定値は、Apache1.3 で 50、Apache2.0 で 250 です。

- UNIX Apache2.0 の場合
  MaxClients 指示子を変更します。**MaxClients** を変更する場合、**ThreadsPerChild/ServerLimit/ThreadLimit** の各値を調整します。この値は、リクエストに応答できる全プロセス中の総スレッド数の最大値となります。
  UNIX 版の Apache2.0 では、複数のスレッドが動作するプロセスが複数個動作し、クライアントから受け付けた1つのリクエストを1つのスレッド上で処理します。既定値は 150 です。
  なお **ServerLimit** の既定値は 16、最大で 20000、**ThreadLimit** の既定値は 64、最大で 15000 まで設定可能です。
  また、リクエスト処理中のプロセス数の最大値は、**MaxClients** を **ThreadsPerChild** で割った値となります。

- UNIX Apache1.3 の場合
  MaxClients 指示子を変更します。この値は、リクエストに応答するために起動される子プロセスの最大数となります。
  UNIX 版の Apache1.3 では、クライアントから受け付けた1つのリクエストを1つの子プロセス上で処理します。
  最大で 4096 まで設定可能です。既定値は 150 です。

4.9.特定クライアントに対するアクセス制限

特定のクライアンに対してアクセス制限をかける場合、次の設定を追加します。定義情報ファイルを直接編集してください。

例えば、次の設定例では、特定のフォルダごとにアクセスを許可するクライアントを設定しています。

http://server/aaa にアクセスできるクライアントは yourdomain.com に属するマシンに限定し、http://server/bbb にアクセスできるクライアントは anotherdomain.com に属するマシンに限定しています。

```
<Directory /opt/WebOTX/domains/domain1/docroot/aaa>
   Order Deny,Allow
   Deny form all
   Allow from yourdomain.com
</Directory>
<Directory /opt/WebOTX/domains/domain1/docroot/bbb>
   Order Deny,Allow
   Deny form all
   Allow from anotherdomain.com
</Directory>
```

次の設定例では、特定のフォルダに対してアクセスを拒否するクライアントを設定しています。
http://server/ccc にアクセスできるクライアントは、ccc.domain.com 以外に属するクライアントとなります。ccc.domain.com に属するクライアントは http://server/ccc にアクセスできません。

```
<Directory /opt/WebOTX/domains/domain1/docroot/ccc>
   Order Allow,Deny
   Allow from all
   Deny from ccc.domain.com
</Directory>
```

4.10.LDAP 連携

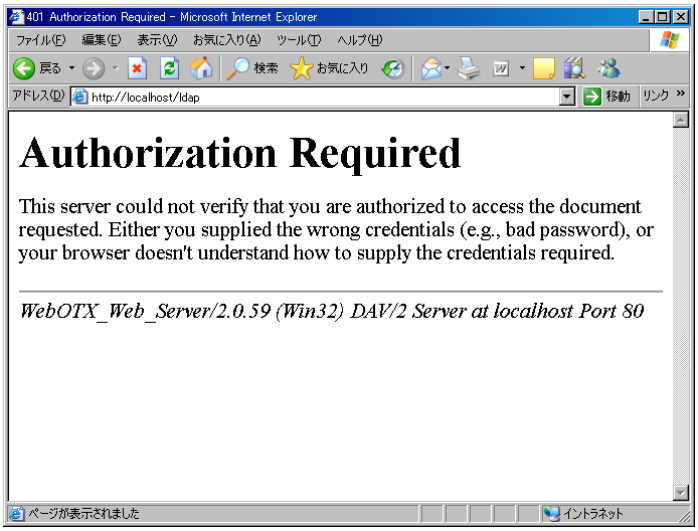
(Apache2.0) WebOTX Web サーバは、WebOTX Application Server にバンドルされている Enterprise Directory Server(EDS)と連携動作が可能であり、EDS に登録されたエントリ情報を、HTTP 認証に利用することができます。
定義情報ファイル(httpd.conf) において、次の設定を追加します。

```
LoadModule ldap_module “${AS_INSTALL}/WebServer2/modules/mod_ldap.so”
LoadModule auth_ldap_module “${AS_INSTALL}/WebServer2/modules/mod_auth_ldap.so”

<Directory /opt/WebOTX/domains/domain1/docroot>
   AuthType Basic
   AuthName “Enter username/password.”
   AuthLDAPUrl ldap://ldap-server:ldap-port/dc=users,dc=webotx,o=NEC,c=JP?uid?sub
   Require valid-user
</Directory>
```

上記設定により、ブラウザから http://server/ に対してアクセスが行われた場合に、次のダイアログが出力されます。
ここで LDAP サーバに登録されたユーザ／パスワードを入力することで、ブラウザからのアクセスが可能となります。

認証に失敗した場合には、次のメッセージ(HTTP ステータスコード 401)がブラウザに出力されます。



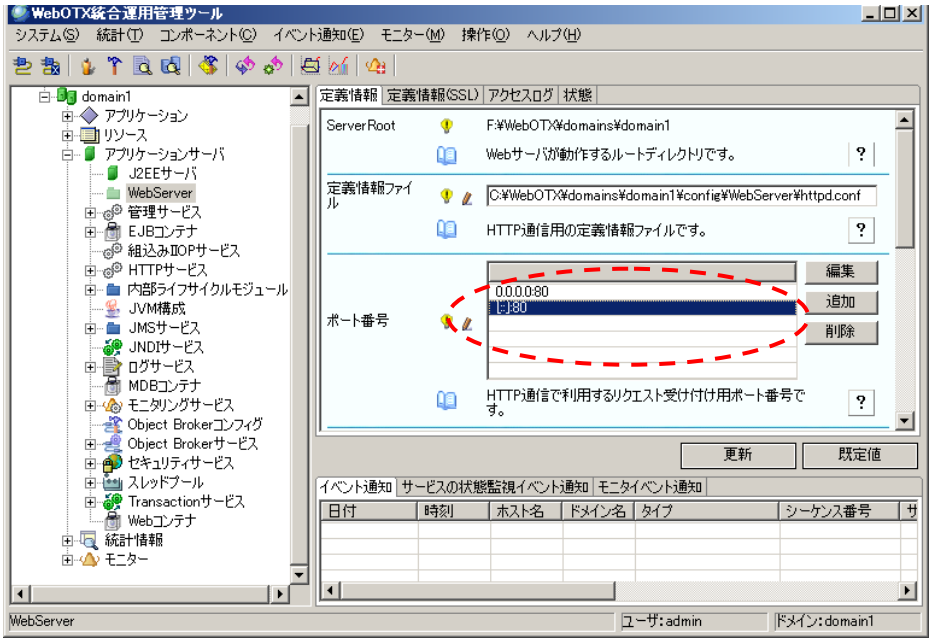
4.11.IPv6/IPv4 混在環境での設定

(Apache2.0) Windows マシンに、IPv4 と IPv6 のそれぞれの IP アドレスが設定されている環境において、それぞれの IP アドレスに対して Web サーバでアクセス受付を行う場合、**Listen** 指示子を利用して、IPv4 と IPv6 のそれぞれの IP アドレスとポート番号を設定してください。
なお、統合運用管理ツール／運用管理コマンドからの操作は、WebOTX V7.11 以降から提供されている機能です。

統合運用管理ツールからの操作

あらかじめ、統合運用管理ツールよりドメインと接続しておきます。

ポート番号の設定



1. 「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「WebServer」を選択し、「**定義情報**」タブの「**ポート番号**」の値を更新します。
   (Windows マシンの IPv4/IPv6 混在環境でポート番号 80 を有効にする場合)
   0.0.0.0:80
   [::]:80

2. 「更新」ボタンを押下することで、設定内容が定義情報ファイルに反映されます。

3. Web サーバを再起動することで、設定内容が反映されます。

運用管理コマンド(otxadmin)からの操作

あらかじめ、otxadmin コマンドを起動し、ドメインにログインしておきます。

otxadmin>login --user admin --password adminadmin --port 6212

ポート番号の設定

1. WebOTX Web サーバのアクセスログのローテーション設定を有効にするには、以下のコマンドを実行します。
   otxadmin>set server.WebServer.port=0.0.0.0:80,[::]:80

2. Web サーバを再起動することで、設定内容が反映されます。

上記の操作を行うことで、定義情報ファイルには、次の設定が反映されます。

```
#Listen 80
Listen 0.0.0.0:80
Listen [::]:80
```

なお、それぞれ IP アドレスに対して指定したポート番号で受付可能状態になっているかを確認するには、**netstat** コマンド等を利用し、設定したポート番号が LISTENING 状態となっていることを確認してください。
次の例では、IPv4 および IPv6 のそれぞれのアドレスに対してポート番号 80 が LISTENING 状態(リクエスト受付可能状態)になっていることを意味します。

```
>netstart –an
   Proto   Local Address          Foreign Address        State
   TCP      0.0.0.0:80             0.0.0.0:0              LISTENING
   …
   TCP      [::]:80                [::]:0                 LISTENING      0
   …
```

4.12.環境変数の設定

UNIX 版において、LoadModule 指示子を利用してモジュールの動的ロードを行う場合、モジュールが利用するライブラリをロードするために、あらかじめ環境変数(LD_LIBRARY_PATH/SHLIB_PATH 等)に登録しておく必要な場合があります。

この場合、次のファイルに必要となる環境変数を追加してください。

(Apache 2.0)
${AS_INSTALL}/WebServer2/bin/envvars
<<envvars の内容>>

```
LD_LIBRARY_PATH="xxx:/opt/WebOTX/WebServer2/lib:$LD_LIBRARY_PATH”
export LD_LIBRARY_PATH
```

(Apache 1.3)
${INSTANCE_ROOT}/domain1/bin/apachectl
<<apachectl の内容>>

```
SHLIB_PATH=”xxx:$SHLIB_PATH”
export SHLIB_PATH
```

4.13.起動待ち合わせ時間の設定

WebOTX Web サーバは、WebOTX Application Server の起動と連動しており、WebOTX Application Server の起動と同時に WebOTX Web サーバに対して、アライブチェックモニタ機能が動作します。

WebOTX Web サーバの起動タイミングとアライブチェックモニタの開始タイミングによっては、Web サーバが完全に起動する前に、アライブチェックモニタ機能が動作するため、「Web サーバが起動していない」というログが出力される場合があります。

この場合、WebOTX Application Server の JavaVM のオプションに次の設定を行うことで、Web サーバ起動後にアライブチェックモニタ機能を開始する時間（待ち合わせ時間）を秒単位で指定することができます。

あらかじめ、otxadmin コマンドを起動し、ドメインにログインしておきます。

otxadmin>login --user admin --password adminadmin --port 6212

otxadmin>create-jvm-options –Dwebotx.webserver.startup_wait_count=xxx（秒単位）

4.14.WebOTX Web サーバの起動に失敗した場合の対処

WebOTX Web サーバの起動／停止は、WebOTX Application Server のドメイン起動／停止に連動していますが、ポートの重複や定義情報の設定ミス等により、起動に失敗する場合があります。

WebOTX Web サーバの起動に失敗した場合、次のファイルにエラーメッセージが出力されますので、その内容を確認し、エラー発生箇所を修正し、WebOTX Web サーバの再起動を行ってください。

エラー出力先）

/opt/WebOTX/domains/domain1/logs/webotx_agent.log
/opt/WebOTX/domains/domain1/logs/WebServer/webotx_websv.log

エラーメッセージ内容）

OTX05230002: execute ExecException occurred
Error: com.nec.webotx.enterprise.util.ExecException: abnormal sub process termination:
Detailed Message: *Error Message*

または

OTX05230002: コマンドの実行(execute)で例外(ExecException)が発生しました。
(com.nec.webotx.enterprise.syste.webserver)
Error: com.nec.webotx.enterprise.util.ExecException: abnormal sub process termination:
Detailed Message: *Error Message*

Error Message には起動に失敗した原因を意味するメッセージが出力されます。

Web サーバの起動に失敗する主な原因は次のことが考えられます。

- ポート番号の重複
  => netstat –an コマンドを実行し、Web サーバで利用するポート番号が、他プロセスで利用しているポート番号と重複していないかを確認します。
- 定義情報の不正
  => <INSTANCE_ROOT>/bin/apachectl(.bat) configtest コマンドを実行し、定義情報に問題がないかを確認します。
- 必要ライブラリの不足
  => UNIX の場合、ldd httpd を実行し、必要ライブラリが存在するかを確認します。

=> LoadModule 指示子で ロードしているモジュールが存在するか、そのモジュールがリンクしているライブラリへのパスが有効になっているか(環境変数に登録されているか)を確認します。

失敗原因についての詳細については、「運用編」－「障害解析」－「機能別」－「Web サーバ(Apache HTTP Server ベース)」を参照してください。

5. 注意・制限事項

ここでは、WebOTX Web サーバの注意・制限事項を記載します。

5.1.64 ビット OS での提供バイナリ

Apache 2.0 では、64 ビット OS で **64 ビットバイナリ**を提供します。
UNIX Apache 1.3 では、32/64 ビット OS に係わらず **32 ビットバイナリ**を提供します。

プラットフォーム	バージョン	バイナリ
Windows (x64)	1.3	Windows(x64)専用 64 ビットバイナリ。
	2.0	Windows(x64)専用 64 ビットバイナリ
HP-UX (IPF)	1.3	HP-UX(IPF)専用 32 ビットバイナリ
	2.0	HP-UX(IPF)専用 64 ビットバイナリ
Linux (x64)	1.3	Linux(x86)共通 32 ビットバイナリ
	2.0	Linux(x64)専用 64 ビットバイナリ

Apache HTTP Server 用の連携モジュールを、WebOTX 上で動作させる場合には、32 ビット用か 64 ビット用を確認し、ビット数が一致するモジュールを利用してください。

5.2.追加・変更インストール

すでに WebOTX Application Server をインストール済みの環境に、「Web サーバ」機能の追加(または 1.3 から 2.0 へのバージョン変更)を行う場合には、WebOTX メディアを用意して、メディアからインストーラを起動してください。

また、既存のドメイン環境に対して「Web サーバ」機能の追加／変更を行う場合には、追加／変更インストールを実行する前に、次の運用管理コマンドを実行してドメインの削除を実行してください。

ドメインの情報をそのまま利用する場合には、ドメインの定義情報のバックアップを行ってください。

```
>otxadmin delete-domain domain1
```

Windows 版の場合、追加/変更インストールを実行することにより、ドメインの再作成を行うか聞かれますので、再作成を行ってください。
UNIX 版の場合、追加/変更インストール実行後に、ant コマンドを実行して、ドメインの再作成を行ってください。
ドメインの再作成方法については、運用編を参照してください。

5.3.複数行の定義情報の更新・追加

定義情報の指示子によっては、複数行の設定が必要なものがあります。このような定義情報の更新・追加をする場合には、直接、定義情報ファイルの更新をしてください。

例えば、次に示すような**コンテナ指示子**の設定を行う場合には、直接、定義情報ファイルを更新してください。

指示子	説明
<Directory>	指定されたディレクトリに対する各種設定を行います。
<DirectoryMatch>	<Directory>と同様です。ディレクトリ情報に正規表現が利用できます。
<Files>	指定されたファイルに対する各種設定を行います。
<FilesMatch>	<Files>と同様です。ファイル情報に正規表現が利用できます。
<Localtion>	指定されたロケーション(URL 情報)に対する各種設定を行います。
<LocationMatch>	<Location>と同様です。ロケーション情報に正規表現が利用できます。
<VirtualHost>	仮想ホストに対して各種設定を行います。
<IfModule>	指定されたモジュールに対する各種設定を行います。

SSL 通信用のポート番号を複数設定する場合、それぞれのポート番号に対して、<VirtualHost>の設定がそれぞれ必要となります。SSL 通信用のポート番号を複数設定する場合には、SSL 通信用の定義情報ファイル(ssl.conf)を直接編集してください。

運用管理ツール/コマンドからの定義情報の追加処理は、定義情報ファイル(httpd.conf)に対してのみ行われます。SSL 通信用の定義情報ファイル(ssl.conf)に対して運用管理ツール／コマンドから定義情報の追加処理はできません。SSL 通信用の定義情報ファイル(ssl.conf)に定義されている情報を更新／追加する場合は、エディタ等を利用して直接編集してください。

5.4.ディレクトリー覧表示機能の無効化

Webサーバの定義情報ファイル(httpd.conf)において、**ディレクトリー覧表示機能を「無効」（ディレクトリリスティングを禁止)**に設定しています。そのため、ブラウザからWebサーバのDocumentRootディレクトリ(つまり http://localhost/)や、DirectoryIndex指示子で設定しているファイル(index.html)が存在しないディレクトリにアクセスすると、以下の**Forbidden** メッセージ(HTTPステータスコード 403)が返却されます。
Webサーバは正常起動していますので、本メッセージが出力されることに**問題はありません。**

なお、Webサーバへのアクセスに対して正常終了(HTTPステータス 200)が返却されることの確認を行う場合に、ブラウザから次のURLを指定し、Webサーバのマニュアルページが表示されることを確認してください。

http://localhost/manual/

Web サーバのディレクトリー覧表示機能を有効にするには、Web サーバの定義情報ファイル(httpd.conf ファイル)において、<Directory>指示子内で設定されている Options 指示子に Indexes オプションを追加します。<Directory>指示子は複数存在しますので、それぞれに設定する必要があります。

```
<Directory “/opt/WebOTX/domains/domain1/docroot”>
    Optios Indexes FollowSymLinks MultiView
    …
</Directory>
```

ただし、ディレクトリー覧表示機能を有効に設定することは、Web サーバのセキュリティ対策上、問題となる場合があります。本設定を有効に変更する場合には、十分な注意が必要です。

5.5.Windows 版の注意・制限事項

Windows 版の WebOTX Web サーバに関する注意・制限事項を記載します。

5.5.1.サービス名

Web サーバのインストールを行うと、次のサービスが自動的に登録されます。このサービスは、WebOTX のドメインの起動／停止と連動しているため、個別にサービスの起動属性等を変更する必要はありません。

バージョン	サービス名
1.3	WebOTX WebServer *domain 名*
2.0	WebOTX WebServer2 *domain 名*

なお、WebOTX のアンインストール時に上記サービスが削除されない場合があります。
この場合には、次のサービスのレジストリ情報を削除してください。

¥¥HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentConnnnrolSet¥Services 配下
¥WebOTXWebServer*domain 名* キー または
¥WebOTXWebServer2*domain 名* キー

5.5.2.Windows ファイアウォールの設定

Windows Server 2003 サービスパック 1 以降 および Windows Server 2003 R2 環境では、「Windows ファイアウォール」機能が標準実装され、デフォルトでは「無効」となっていますが、「有効」に設定した場合、ポートのブロッキングが発生し、動作に影響があります。

「コントロールパネル」－「Windows ファイアウォール」において、「例外」タブを選択し、「プログラムの追加」または「ポートの追加」を実行して、例外設定を行ってください。

○　プログラムの追加

パス
<WebOTX_Dir>¥WebServer¥bin¥apache.exe
<WebOTX_Dir>¥WebServer2¥bin¥apache.exe

○　ポートの追加

名前	ポート	TCP/UDP
HTTP	*80* (※)	TCP
HTTPS	*443*(※)	TCP

（※）インストール時に指定した、または、Web サーバが利用するポート番号を設定してください。

5.5.3.Windows Server 2003 インストール時の注意事項

Windows Server 2003 マシンにおいて、インストール時に「Web サーバ 1.3」を選択した場合、インストール時や新規ドメイン作成の途中で、イベントログ情報と WebOTX のログ情報に、次の警告／エラーメッセージが出力される場合があります。

WebOTX では、ドメインの作成時に情報設定のために、ドメインの作成とドメインの起動を行いますが、Web サーバのサービス起動は行わないために、ドメイン停止時に「Web サーバのサービスは起動していない」という意味の次の各メッセージが出力されます。なお、**本メッセージが出力されても動作上、特に問題はありません。**

警告メッセージ(イベントログ)
OTX05230002: コマンドの実行(execute)で例外(ExecException)が発生しました。(com.nec.webotx.enterprise.syste.webserver) Error: com.nec.webotx.enterprise.util.ExecException: abnormal sub process termination: Detailed Message: The Process Output: **The WebOTX WebServer domain1 service is not started.**
OTX01205061:例外 :(com.nec.webotx.enterprise.system.core) Error: com.nec.webotx.appserv.server.ServerLifecycleException: com.nec.webotx.enterprise.util.ExecException: abnormal sub process termination: Detailed Message: The Process Output: **The WebOTX WebServer domain1 service is not started.**
OTX01205107: サービス“WebServerService”を停止することができません！(com.nec.webotx.enterprise.system.core)

エラーメッセージ(イベントログ)

OTX05210004: WebServer Lifecycle Shutdown で例外が発生しました。

(com.nec.webotx.enterprise.syste.webserver)

Error: com.nec.webotx.enterprise.webserver.WebServerRuntimeException:

com.nec.webotx.enterprise.util.ExecException: abnormal sub process termination:

Detailed Message: The Process Output: **The WebOTX WebServer domain1 service is not started.**

5.6.UNIX 版の注意・制限事項

UNIX 版の WebOTX Web サーバに関する注意・制限事項を記載します。

5.6.1.必要パッケージ

Linux 版を利用する場合、次のパッケージが必要となります。

- compat-db-4.0.14-5 パッケージ
- compat-libcom_err パッケージ

5.6.2.WebOTX 運用ユーザ利用時の注意事項

UNIX 版の場合、インストール時に「WebOTX 運用ユーザ」を設定した場合、次の制限があります。

- 利用できるポート番号が OS によって制限されるため、1024 以下のポート番号は利用できません。

5.6.3.Solaris 版の注意事項

- SSL 通信を有効にして Web サーバを起動した場合、(**<VirtualHost _default_:*ssl-port*>** 指示子を利用している場合)、次のエラーメッセージが出力され、Web サーバの起動に失敗する場合があります。

  OTX05230002: コマンドの実行(execute)で例外(ExecException)が発生しました。
  com.nec.webotx.enterprise.util.ExecException: abnormal subprocess termination:
  Detail Message: [xx mm dd hh:mm:ss yyyy] [crit] [xx mm dd hh:mm:ss yyyy] file vhost.c, line 190, assertion “rv == APR_SUCCESS” failed
  異常終了 – コアダンプしました。

  このメッセージは、Solaris マシンで自マシンの IP アドレス解決に失敗した場合に出力されるメッセージです。
  該当 Solaris マシンの TCP/IP 設定が正しく設定されているかを確認してください。
  また、/etc/nsswitch.conf に ″hosts: files dns ..″ の設定が含まれているかを確認してください。

  なお、OS 側の設定変更ができない場合には、ssl.conf に定義されている <VirtualHost _default_:*ssl-port*> の設定を <VirtualHost *:*ssl-port*> に変更してください。

- リクエスト処理中に以下のメッセージがログに出力される場合があります。

  (45)Deadlock situation detected/avoided: apr_proc_mutex_lock failed. Attempting to shutdown process gracefully.

子プロセスが、リクエストを取得するために、シリアライズ化のための排他制御処理(ロック処理)を実行しますが、その処理に失敗した場合に本メッセージが出力されます。ロック処理に失敗した子プロセスは、現在処理中のリクエストを完了させた後、再起動を行います。

本メッセージが出力された場合の影響は、リクエスト処理の遅延が発生する可能性があります。リクエストの遅延やエラーメッセージを出力させないようにするには、httpd.conf に以下の定義を追加してください。

AcceptMutex pthread または posixsem