

プログラマーズガイド

iPlanet Web Server, Enterprise Edition

Version 6.0

2001 年 5 月

Copyright © 2001, Sun Microsystems, Inc. All rights reserved. 継承部分については Copyright © 2001, Netscape Communications Corporation Inc.

Sun、Sun Microsystems、iPlanet、iPlanet のロゴマークは、米国およびその他の国における米国 Sun Microsystems, Inc.(以下、米国 Sun Microsystems 社とします)の商標もしくは登録商標です。

iPlanet および iPlanet のロゴマークは Sun | Netscape Alliance の商標です。

サン のロゴマーク および Solaris は、米国 Sun Microsystems 社の登録商標です。

Netscape および Netscape の N のロゴマークは、米国およびその他の国における Netscape Communications Corporation 社の登録商標です。その他の Netscape のロゴマーク、製品名、およびサービス名もまた、米国の Netscape Communications Corporation の商標であり、その他の国においても登録されている可能性があります。

本製品には Apache Software Foundation (<http://www.apache.org/>) で開発されたソフトウェアが含まれています。Copyright © 1999, The Apache Software Foundation. All rights reserved.

本製品にはカリフォルニア大学バークレイ校およびその貢献者によって開発されたソフトウェアが含まれています。Copyright © 1990, 1993, 1994, The Regents of the University California. All rights reserved.

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

Federal Acquisitions: Commercial Software-Government Users Subject to Standard License Terms and Conditions.

本書で説明されている製品は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。

Sun | Netscape Alliance の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法(外為法)に定められる戦略物資等(貨物または役務)に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典： iPlanet Web Server, Enterprise Edition Programmer's Guide

目次

このマニュアルについて	7
第1章 概要	9
構成ファイル	10
iPlanet Web Server 6.0 の API	10
サーバによる HTML タグの構文解析	11
サーバによるタグの構文解析を有効にする	11
参照先	12
CGI	12
CGI を有効にする	12
CGI プログラムの実行環境をカスタマイズする (UNIX のみ)	14
CGI プログラムをサーバに追加	19
Windows NT CGI プログラムとシェル CGI プログラム	19
Perl CGI プログラム	19
CGI 変数	19
参照先	21
Java サブレットと JavaServer Pages (JSP)	21
Java サブレットと JavaServer Pages を有効にする	22
サブレットおよび JavaServer Pages をサーバに追加する	23
参照先	23
NSAPI	24
NSAPI を有効にする	24
NSAPI プラグイン (SAF) をインストールする	25
参照先	25
アクセス制御 API	25
新しい認証サービスを登録	26
参照先	26
証明書マッピング API	27

参照先	27
API の要約	28
旧バージョンからの変更内容	29
iPlanet Web Server 3.x 以降の API の変更内容	29
iPlanet Web Server 4.0.x 以降の API の変更内容	29
iPlanet Web Server 4.1.x 以降の API の変更内容	30

第 2 章 構成ファイル	31
backups.conf	32
certmap.conf	34
cjava.properties	35
cluster.xml	36
contexts.properties	37
cron.conf	41
dbswitch.conf	42
iwsstats.xml	43
jvm12.conf	50
magnus.conf	53
Init 関数	54
指令	60
mime.types	67
ns-cron.conf	68
nsfc.conf	68
obj.conf	70
bucket パラメータ	71
AuthTrans 関数	72
NameTrans 関数	73
PathCheck 関数	75
ObjectType 関数	79
Service 関数	81
AddLog 関数	87
Error 関数	88
password.conf	88
rules.properties	89
server.xml	90
servers.lst	98
servlets.properties	99
web.xml	101
web-apps.xml	102

第 3 章 サーバによる HTML タグの構文解析	109
サーバサイドの HTML コマンドの使用方法	110

config	111
include	111
echo	112
fsize	112
flastmod	112
exec	112
サーバサイド HTML コマンドの環境変数	113
サブレットを埋め込む方法	114
サーバが構文解析するカスタム HTML タグの定義方法	116
実装方法	116
タグを実装する関数を定義する	117
初期化関数を記述して新しいタグを登録する	121
サーバに新しいタグを読み込む	121
付録 A iWS 4.x と 6.0 の設定の変更点	123
magnus.conf	123
obj.conf	127
contexts.properties	128
rules.properties	129
servlets.properties	129
索引	131

このマニュアルについて

このマニュアルは、iPlanet™ Web Server, Enterprise Edition 6.0 がサポートするさまざまな API やプログラミング技術についての情報を必要とする開発者にとって出発点となるものです。

このマニュアルでは API やプログラミング技術について簡単に説明し、詳細情報の参照先を示しています。一般的な個々の API やプログラミング技術については別のプログラマ用のマニュアルで説明されていますが、カスタマイズされたタグのサーバーによる構文解析を定義する API はこのマニュアルの第3章「サーバによる HTML タグの構文解析」で扱います。

このマニュアルには次の章があります。

- 第1章「概要」
サーバに同梱された API におけるバージョン 3.x から 6.0 までの変更点について説明します。また、サーバがサポートするさまざまな API やプログラミング技術についても簡単に説明し、詳細情報の参照先を示します。
- 第2章「構成ファイル」
iPlanet Web Server で使用される構成ファイルについて簡単に説明します。
- 第3章「サーバによる HTML タグの構文解析」
サーバが構文解析するタグの使用方法を説明して標準的なタグを一覧で示し、タグをユーザが独自に定義する方法について説明します。
- 付録 A 「iWS 4.x と 6.0 の設定の変更点」
iPlanet Web Server 4.x 以降の構成ファイルの変更点を簡単に説明します。

注

- このマニュアルでは、特に明記しない限り UNIX 固有の記述方式が Linux にも適用されます。
 - このマニュアルでは、パスの区切り記号に「/」を使用していますが、Microsoft Windows 環境では、画面上に円記号 (¥) が表示されることがあります。また、パスの先頭にドライブ名の指定が必要な場合もあります。詳細は、オペレーティングシステムのマニュアルを参照してください。
 - このマニュアルには、「バックスラッシュ (\)」という表記が含まれていますが、これは円記号 (¥) に読み換えてください。
-

概要

iPlanet Web Server 6.0 はさまざまなアプリケーションプログラミングインタフェース (API) とプログラミング技術をサポートしており、次のような機能を実現しています。

- クライアントの要求に応じて動的なコンテンツを生成する
- サーバの動作を変更し、拡張する
- サーバに保存された内容を変更する

この章では、サーバでサポートするさまざまな API とプログラミング技術について、その概要を説明します。それぞれの API やプログラミング技術の詳細については、このマニュアルの該当する章または個別のマニュアルを参照してください。

この章には次の節があります。

- 構成ファイル
- iPlanet Web Server 6.0 の API
- API の要約
- 旧バージョンからの変更内容

構成ファイル

サーバマネージャおよびクラスマネージャインタフェースを使用する代わりに、構成ファイルを編集することによって、iPlanet Web Server を構成することができます。構成ファイルの大半は、`server_root/https-server_id/config` ディレクトリにあります (`server_root` はサーバルート、`server_id` はサーバ id を示す)。たとえば、Windows NT の `C:\iPlanet\Servers\` に iPlanet Web Server がインストールされている場合、サーバの構成ファイル `myserver.com` は次のディレクトリにあります。

```
C:\iPlanet\Servers\https-myserver.com\config
```

主な構成ファイルは `magnus.conf`、`server.xml`、`obj.conf`、および `mime.types` ですが、この他にも構成ファイルがあります。構成ファイルの概要は、第 2 章「構成ファイル」を参照してください。

`magnus.conf`、`server.xml`、`obj.conf`、および `mime.types` ファイルの詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照してください。

iPlanet Web Server 4.x 以降の構成ファイルの変更点については、付録 A 「iWS 4.x と 6.0 の設定の変更点」を参照してください。

iPlanet Web Server 6.0 の API

この節では、iPlanet Web Server 6.0 でサポートするさまざまな API とプログラミング技術について概要を説明し、iPlanet Web Server 6.0 でその機能を活用する方法を説明します。さらに、それらの詳細情報の参照先を示します。

iPlanet Web Server で拡張や変更ができる主なカテゴリには、次のものがあります。

- 要求への応答 (またはその一部) を動的に作成する。このカテゴリに該当する API とプログラミング技術は次のとおり
 - サーバによる HTML タグの構文解析
 - CGI
 - Java サーブレットと JavaServer Pages (JSP)
- サーバプラグインを実装し、サーバの動作を変更する。サーバプラグインの多くは Netscape Server API (NSAPI) を使って記述される。サーバプラグインを記述するための API もある。たとえば、サーバリソースへのアクセスを制御するアクセス制御リスト API (ACLAPI) など

サーバの動作を変更する API は次のとおり。

- NSAPI
- アクセス制御 API

- 証明書マッピング API
- リソースとディレクトリの追加や削除、変更によって、サーバの内容を変更する。
変更のために、リモートファイルを手作業で編集する

サーバによる HTML タグの構文解析

iPlanet Web Server 6.0 では、サーバサイドのタグをユーザが独自に定義する C 言語の API を提供しています。定義したタグは、`config` や `include` などの標準的なサーバサイドのタグに追加して、HTML ファイル内で使用することができます。

サーバによるタグの構文解析を有効にする

サーバサイドのタグの構文解析は、クラスマネージャの「Content Management」タブにある「Parse HTML」ページで、有効または無効にすることができます。このページでは、サーバサイドの HTML タグの構文解析を無効にすることができます。また、`exec` タグが有効であるかどうかにかかわらず、構文解析を有効にすることもできます。このページでは、全ファイルを構文解析するか、または拡張子が `.shtml` のファイルだけを構文解析するかを指定することもできます。

`magnus.conf` の指令でサーバサイドのタグの構文解析を有効にする場合、Windows NT では次のようになります (`install_dir` はインストール先のディレクトリを示す)。

```
Init funcs="shtml_init,shtml_send"
shlib="install_dir/bin/https/bin/Shtml.dll" NativeThread="no"
fn="load-modules"
```

UNIX の場合、ファイルが `Shtml.so` であることを除けば指令は同じです。

`obj.conf` の指令でサーバサイドのタグの構文解析を有効にする場合は、次のようになります。

```
Service fn="shtml_send" type="magnus-internal/parsed-html"
method="(GET|HEAD)"
```

ファイルの拡張子が `.shtml` 以外のサーバサイドのタグに対して、構文解析を有効にするには、`mime.types` ファイルの適切な行に拡張子を追加します。たとえば次の行を `mime.types` に記述すると、`.shtml` または `.jbhtml` の拡張子を持つファイルでサーバサイドのタグが構文解析されます。

```
type=magnus-internal/parsed-html exts=shtml,jbhtml
```

参照先

サーバによるタグの構文解析の定義と使い方については、第 3 章「サーバによる HTML タグの構文解析」を参照してください。

CGI

CGI (Common Gateway Interface) はサーバ上で稼働するプログラムで、応答を作成して要求元のクライアントに返します。CGI プログラムは、C、C++、Java、Perl などのさまざまな言語で記述することができ、シェルスクリプトで記述することもできます。CGI は URL を呼び出すことによって実行されます。

iPlanet Web Server は、バージョン 1.1 の CGI 仕様に対応しています。

サーバでは、CGI のスクリプトやプログラムを実行するたびにプロセスを起動するので、このプログラミングメソッドは、サーバ側にかかるコストが高くなる方式です。

CGI を有効にする

iPlanet Web Server には CGI プログラムを識別するために、次の 2 つの方法が提供されています。

- CGI 用のディレクトリを指定する
サーバは、CGI ディレクトリ内のすべてのファイルを CGI プログラムと見なす
- CGI ファイルの拡張子を指定する
サーバは、特定の拡張子を持つすべてのファイルを CGI プログラムと見なす

CGI 用のディレクトリを指定する

CGI プログラムを含むディレクトリ (CGI プログラムだけが格納されている) を指定するには、クラスマネージャの「Programs」タブにある「CGI Directory」ページを使用します。サーバは、このディレクトリにあるすべてのファイルを CGI プログラムとして扱います。

各 CGI ディレクトリのファイル `obj.conf` には `NameTrans` 指令が記述されていて、`cgi` 名とそのディレクトリのリソースに対する要求とを関連付けています。この指令は、クラスマネージャインタフェースの CGI ディレクトリを指定すると `obj.conf` に自動的に追加されます。また、必要に応じて手動で `obj.conf` に追加することもできます。

たとえば次のような命令があると、`http://server-name/cgi-local` にあるリソースへのすべての要求は、`C:/iPlanet/Servers/docs/mycgi` のディレクトリにある CGI プログラムを呼び出すものであると解釈されます (`server-name` はサーバ名を示す)。

```
NameTrans fn="pfx2dir" from="/cgi-local"
dir="C:/iPlanet/Servers/docs/mycgi" name="cgi"
```

`obj.conf` ファイルには、次のような名前のオブジェクトが指定されている必要があります。

```
<Object name="cgi">
  ObjectType fn="force-type" type="magnus-internal/cgi"
  Service fn="send-cgi"
</Object>
```

このオブジェクトの記述は、`obj.conf` から削除しないでください。この記述を削除すると、クラスマネージャインタフェースで CGI ディレクトリを指定した場合でも、手動で `NameTrans` 指令を `obj.conf` に追加した場合でも、サーバはディレクトリを認識しなくなります。

CGI ファイルの拡張子を指定する

クラスマネージャの「Programs」タブにある「CGI File Type」ページで、ファイルのあるディレクトリに関わりなく特定の拡張子を持つすべてのファイルを CGI プログラムとして扱うようサーバに設定します。デフォルトの CGI 拡張子は、`.cgi`、`.bat` および `.exe` です。

CGI プログラムを示す拡張子を変更するには、`mime.types` の次の行で拡張子を変更します。`mime.types` を編集したら、サーバを必ず再起動します。

```
type=magnus-internal/cgi exts=cgi,exe,bat
```

該当する拡張子の付いているすべてのファイルを、サーバが CGI プログラムと見なす場合には、`obj.conf` ファイルに次のような `Service` 指令が記述されています。

```
Service fn="send-cgi" type="magnus-internal/cgi"
```

CGI プログラムの実行環境をカスタマイズする (UNIX のみ)

実行環境をカスタマイズする前に、`suid Cgistub` をインストールし、ルートとして実行する必要があります。

1. スーパーユーザとしてログインする

```
su
```

2. `Cgistub` の `private` ディレクトリを作成する (`server_root` はサーバルートを示す)

```
cd server_root/https-instance
```

```
mkdir private
```

3. `Cgistub` を `private` ディレクトリにコピーする

```
cd private
```

```
cp ../../bin/https/bin/Cgistub .
```

4. `private` の所有者をサーバのユーザに設定する

```
chown user .
```

5. `private` にパーミッションを設定する

```
chmod 500 .
```

6. `Cgistub` の所有者を `root` に設定する

```
chown root Cgistub
```

7. `Cgistub` にパーミッションを設定する

```
chmod 4711 Cgistub
```

8. `obj.conf` の `send-cgi SAF` への各参照に `user` パラメータを指定することができる。

```
Service fn="send-cgi" user="user" (user はユーザを示す)
```

変数の使用も可能。たとえば `server.xml` で、`VS` (仮想サーバ) 要素を次のような `VAR` サブ要素にすることもできる

```
<VAR user="user" />
```

これにより、`obj.conf` で次のような `send-cgi SAF` の行を記述することができる

```
Service fn="send-cgi" user="$user"
```

`send-cgi`、`server.xml`、および `obj.conf` の詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照。

9. サーバを再起動して変更を適用する

注 Cgistub は `server_root/https-instance/private` のディレクトリにインストールすることをお勧めします (`server_root` はサーバルート、`instance` はインスタンスを示す)。これ以外の場所にインストールする場合は、Cgistub へのパスを `magnus.conf` の `init-cgi` 関数に指定しておく必要があります。詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照してください。

注 `suid Cgistub` プログラムは、NFS マウントにはインストールできない場合があります。`suid Cgistub` を使用する場合は、サーバのインスタンスをローカルのファイルシステムにインストールする必要があります。

Cgistub には次のようなセキュリティ上の制限があります。

- CGI プログラムを実行するユーザには 100 以上の `uid` が必要。これにより、Cgistub を使用して第三者がルートとしてアクセスすることを防ぐ
- CGI プログラムは実行者であるユーザが所有し、所有者以外のユーザが書き込めない設定になっている必要がある。これにより、第三者がプログラムに入り込んでリモートで実行することが困難になる
- `cgistub` は UNIX の待機ソケットを作成し、アクセス権を 0700 に設定する

注 多種多様な UNIX マシン間でのソケットのアクセス権の制限については、考慮されていません。これは SunOS/Solaris の現行バージョンも同様です。Cgistub の不正な利用を防ぐには、`magnus.conf` の `TempDir` 指令を使って、サーバの一時ディレクトリをサーバのユーザだけがアクセスできる場所に変更します。詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照してください。

Cgistub をインストールすると、次の方法で実行環境をカスタマイズできます。

- 独自の CGI ディレクトリ、および仮想サーバの UNIX ユーザとグループを指定する
- 仮想サーバの Chroot ディレクトリを指定する

独自の CGI ディレクトリ、および仮想サーバの UNIX ユーザとグループを指定する

仮想サーバの CGI プログラムが他のユーザの作業に影響を及ぼさないようにするには、プログラムを独自のディレクトリに格納し、独自の UNIX ユーザとグループのアクセス権で実行する必要があります。

最初に、UNIX のユーザとグループを作成します。ユーザとグループの正確な作成手順は、オペレーティングシステムによって異なります。使用しているオペレーティングシステムのマニュアルを参照してください。

次に、以下の手順に従って仮想サーバの `cgi-bin` ディレクトリを作成します。

1. スーパーユーザとしてログインする

```
su
```

2. 仮想サーバのディレクトリに移動する (`vs_dir` は仮想サーバのディレクトリを示す)

```
cd vs_dir
```

3. `cgi-bin` ディレクトリを作成する (`user` はユーザを、`group` はグループを示す)

```
mkdir cgi-bin
```

```
chown user:group cgi-bin
```

```
chmod 755 cgi-bin
```

ここで、次のいずれかの方法で仮想サーバの CGI ディレクトリ、ユーザ、およびグループを設定できます。

- `obj.conf` ファイルにある `send-cgi Service SAF` の変数 `dir`、`user`、および `group` を使う (iPlanet Web Server の『NSAPI プログラマーズガイド』を参照)
- Virtual Server Manager の「Settings」タブにこの情報を入力する (iPlanet Web Server の『管理者ガイド』を参照)

仮想サーバの Chroot ディレクトリを指定する

セキュリティを更に向上させるには、前述したデータや仮想サーバのディレクトリ以外のディレクトリにあるデータには、CGI スクリプトからアクセスできないようにする必要があります。

最初に、`chroot` 環境を設定します。`chroot` 環境の正確な設定手順は、オペレーティングシステムによって異なります。使用しているオペレーティングシステムのマニュアルを参照してください。`ftpd` および `chroot` の `man` ページを最初に参照することをお勧めします。

Solaris 2.6 から 8 のバージョンでは次の手順に従ってください。

1. スーパーユーザとしてログインする

```
su
```


2. chroot ディレクトリに移動する。通常は、前の節で述べた *vs_dir* ディレクトリ
`cd chroot`
3. chroot ディレクトリに tmp を作成する
`mkdir tmp`
`chmod 1777 tmp`
4. chroot ディレクトリに dev を作成する
`mkdir dev`
`chmod 755 dev`
5. /dev/tcp を一覧表示し、出力結果のメジャー番号とマイナー番号を確認する。この例では、メジャー番号は 11、マイナー番号は 42
`ls -lL /dev/tcp`

```
crw-rw-rw-  1 root      sys          11, 42 Apr  9 1998 /dev/tcp
```
6. メジャー番号とマイナー番号を使用して tcp デバイスを作成する
`mknod dev/tcp c 11 42`
`chmod 666 dev/tcp`
7. 次のデバイスごとに、手順 5 と 6 を繰り返す (それぞれのデバイスには、固有のメジャー番号とマイナー番号がある)
`/dev/udp`
`/dev/ip`
`/dev/kmem`
`/dev/kstat`
`/dev/ksyms`
`/dev/mem`
`/dev/null`
`/dev/stderr`
`/dev/stdin`
`/dev/stdout`
`/dev/ticotsord`
`/dev/zero`
8. chroot ディレクトリの dev にあるデバイスにアクセス権を設定する
`chmod 666 dev/*`

9. chroot ディレクトリに lib と usr/lib を作成して設定する

```
mkdir usr
mkdir usr/lib
ln -s /usr/lib
ln /usr/lib/* usr/lib
```

このコマンドで作成されるメッセージは無視してください。

/usr/lib ディレクトリが別のファイルシステムにある場合は、最後のコマンドを次のように変更する

```
cp -rf /usr/lib/* usr/lib
```

10. chroot ディレクトリに bin と usr/bin を作成して設定する

```
mkdir usr/bin
ln -s /usr/bin
ln /usr/bin/* usr/bin
```

このコマンドで作成されるメッセージは無視してください。

/usr/bin ディレクトリが別のファイルシステムにある場合は、最後のコマンドを次のように変更する

```
cp -rf /usr/bin/* usr/bin
```

11. chroot ディレクトリに etc を作成して設定する

```
mkdir etc
ln /etc/passwd /etc/group /etc/netconfig etc
```

12. chroot 環境をテストする

```
chroot chroot bin/ls -l
```

出力結果の例を次に示します。

```
total 14
lrwxrwxrwx  1 root  other    8 Jan 13 03:32 bin -> /usr/bin
drwxr-xr-x  2 user  group   512 Jan 13 03:42 cgi-bin
drwxr-xr-x  2 root  other   512 Jan 13 03:28 dev
drwxr-xr-x  2 user  group   512 Jan 13 03:26 docs
drwxr-xr-x  2 root  other   512 Jan 13 03:33 etc
lrwxrwxrwx  1 root  other    8 Jan 13 03:30 lib -> /usr/lib
drwxr-xr-x  4 root  other   512 Jan 13 03:32 usr
```

次のいずれかの方法で仮想サーバの chroot ディレクトリを設定できます。

- obj.conf ファイルにある send-cgi Service SAF のパラメータ chroot を使用する (iPlanet Web Server の『NSAPI プログラマーズガイド』を参照)

- Virtual Server Manager の「Settings」タブにこの情報を入力する (iPlanet Web Server の『管理者ガイド』を参照)

CGI プログラムをサーバに追加

以下のいずれかを実行すると、CGI プログラムが iPlanet Web Server に追加されます。

- CGI 用のディレクトリが設定されている場合、その場所にプログラムファイルを格納する
- CGI ファイルタイプ認識機能が有効の場合は、サーバが CGI プログラムとして認識するファイル名を付けてドキュメントルート以下のディレクトリに格納する

UNIX では、プログラムファイルの実行権が必要です。

Windows NT CGI プログラムとシェル CGI プログラム

クラスマネージャインタフェースを使用して、Windows NT に CGI プログラムとシェル CGI プログラムをインストールする方法については、iPlanet Web Server の『管理者ガイド』を参照してください。

Perl CGI プログラム

Perl 5.6.x では、`-w` フラグのある CGI を実行することはできません。代わりに、次のコードをファイルに記述してください。

```
use warnings;
```

CGI 変数

CGI プログラムでは、サーバがセキュアモードで稼働している場合、標準の CGI 変数だけでなく表 1-1 の iPlanet Web Server CGI 変数を使用しても、クライアント証明書情報にアクセスできます。CLIENT_CERT 変数と REVOCATION 変数は、クライアント証明書ベースの認証が有効な場合にだけ使用できます。

表 1-1 CGI 変数

変数	説明
SERVER_URL	クライアントが要求したサーバの URL
HTTP_xxx	HTTP が要求する受信ヘッダー、xxx はヘッダー名
HTTPS	サーバがセキュアモードの場合は ON、それ以外は OFF
HTTPS_KEYSIZE	SSL ハンドシェイクのキーサイズ (サーバがセキュアモードの場合に利用可)

表 1-1 CGI 変数 (続き)

変数	説明
HTTPS_SECRETKEYSIZE	SSL ハンドシェークの秘密鍵部分のキーサイズ (サーバがセキュアモードの場合に利用可)
HTTPS_SESSIONID	接続のセッション ID (サーバがセキュアモードの場合に利用可)
CLIENT_CERT	クライアントが発行する証明書 (バイナリ DER 形式)
CLIENT_CERT_SUBJECT_DN	クライアント証明書のサブジェクトの識別名
CLIENT_CERT_SUBJECT_OU	クライアント証明書のサブジェクトの組織単位
CLIENT_CERT_SUBJECT_O	クライアント証明書のサブジェクトの組織
CLIENT_CERT_SUBJECT_C	クライアント証明書のサブジェクトの国
CLIENT_CERT_SUBJECT_L	クライアント証明書のサブジェクトの場所
CLIENT_CERT_SUBJECT_ST	クライアント証明書のサブジェクトの州
CLIENT_CERT_SUBJECT_E	クライアント証明書のサブジェクトの電子メール
CLIENT_CERT_SUBJECT_UID	クライアント証明書のサブジェクトの CN の UID
CLIENT_CERT_ISSUER_DN	クライアント証明書発行者の識別名
CLIENT_CERT_ISSUER_OU	クライアント証明書発行者の組織単位
CLIENT_CERT_ISSUER_O	クライアント証明書発行者の組織
CLIENT_CERT_ISSUER_C	クライアント証明書発行者の国
CLIENT_CERT_ISSUER_L	クライアント証明書発行者の場所
CLIENT_CERT_ISSUER_ST	クライアント証明書発行者の州
CLIENT_CERT_ISSUER_E	クライアント証明書発行者の電子メール
CLIENT_CERT_ISSUER_UID	クライアント証明書発行者の CN の UID
CLIENT_CERT_VALIDITY_START	証明書の開始日
CLIENT_CERT_VALIDITY_EXPIRES	証明書の有効期限
CLIENT_CERT_EXTENSION_xxx	証明書の拡張子、xxx は拡張子名
REVOCATION_METHOD	証明書の取り消し方法がある場合は、その方法名
REVOCATION_STATUS	証明書がある場合は、その取り消し状況

参照先

CGI プログラムの記述方法について数多くの情報があります。最初に次のサイトの「The Common Gateway Interface」を参照することをお勧めします。

<http://hoohoo.ncsa.uiuc.edu/cgi/overview.html>

Java サーブレットと JavaServer Pages (JSP)

iPlanet Web Server 6.0 は Java サーブレット仕様バージョン 2.2 (Web アプリケーションと WAR ファイルのサポートを含む) および JavaServer Pages (JSP) バージョン 1.1 をサポートしています。

Java サーブレットはサーバサイドの Java プログラムで、CGI プログラムとほぼ同じ方法で、クライアントの要求に応じて動的な内容を生成することができます。サーブレットには URL を呼び出してアクセスします。

サーブレットは米国 Sun Microsystems Inc. (以降、Sun とします) が作成した Servlet API を使用して作成します。iPlanet Web Server 6.0 には、Java サーブレットの開発と実行に必要なファイルがすべて含まれています。servlet.jar ファイルがアクセスできる Java コンパイラであれば、どのコンパイラでもサーブレットをコンパイルできます。servlet.jar ファイルはサーバのインストール先である次のディレクトリにあります。

```
/bin/https/jar
```

Servlet API の使い方は、次の Sun の Web サイトにある Java Servlet API のマニュアルを参照してください。

<http://java.sun.com/products/servlet/index.html>

JavaServer Page (JSP) は、HTML で作成されたページのように、Web ブラウザで表示できるページです。ただし HTML タグ以外にも Java コードを含む JSP タグや指令が使われているので、Web ページを作成する際にページに動的な内容を記述することができます。このような機能を追加すると、プロパティ値を表示したり簡単な条件を使用したりできます。

JavaServer Pages の使い方については、次の Sun の Web サイトにある JavaServer Pages のマニュアルを参照してください。

<http://java.sun.com/products/jsp/index.html>

Java サブレットと JavaServer Pages を有効にする

サブレットを有効にするにはサーバマネージャで「Java」タブを選択し、次に「Enable/Disable Servlets/JSP」タブを選択します。「Enable Java Globally」チェックボックスをオンにすると、サーバ全体でサブレットが有効になります。「Enable Java for Class」チェックボックスをオンにすると、単一の仮想サーバクラスのサブレットが有効になります。クラスのサブレットを有効にするには、Java 全体を有効にしておく必要があります。デフォルトでは Java 全体が有効であり、仮想サーバの各クラスも有効です。

JSP を有効にするには、enable=true である jsp-servlet 要素を web-apps.xml ファイルに指定して tools.jar を JVM クラスパスに追加する必要があります。コンパイルされていない JSP を実行する場合は Java Development Kit (JDK) もインストールする必要があります。

iPlanet Web Server 6.0 のインストール時に、同梱されている Java Runtime Environment (JRE) をインストールするか、または JDK へのパスを指定することができます。JDK は iPlanet Web Server に同梱されていませんが、次の米国 Sun Microsystems, Inc. の Web サイトから無料でダウンロードすることができます。

<http://java.sun.com/products/jdk/1.2/>

JRE を使うとサブレットやコンパイル前の JSP をサーバで実行することはできますが、コンパイルされていない JSP を実行するには JDK が必要です。

iPlanet Web Server 6.0 では、JDK 1.2 の正式バージョンを使う必要があります。詳細は、iPlanet Web Server の『サブレットに関するプログラマーズガイド』を参照してください。

iPlanet Web Server のインストール中に JRE をインストールする場合も JDK へのパスを指定する場合も、Administration Server の「Global Settings」タブにある「Configure JRE/JDK Paths」ページでいつでも JRE と JDK を切り替えて使うことができます。

magnus.conf ファイルには次のような Init 指令が設定されています。最初の指令でサブレットのライブラリを読み込み、サブレット関連の機能を iPlanet Web Server で利用できるようにします。他の 2 つの指令により、サブレットのエンジンが初期化されます。以下に示す shlib 値は Windows NT 用のものです。

```
Init shlib="d:/server_root/bin/https/bin/NSServletPlugin.dll"
funcs="NSServletEarlyInit,NSServletLateInit,NSServletNameTrans,N
SServletService" shlib_flags="(global|now)" fn="load-modules"
Init EarlyInit="yes" fn="NSServletEarlyInit"
Init LateInit="yes" fn="NSServletLateInit"
```

UNIX では `shlib` 値は次のようになります。

```
shlib="server_root/bin/https/lib/libNSServletPlugin.so"
```

`obj.conf` ファイルにはサーブレットに関連する他の指令もあり、サーブレットへの要求を処理する追加オブジェクトを定義します。

サーブレットおよび JavaServer Pages をサーバに追加する

次のいずれかの方法で、サーブレットと JSP を使って、クライアントにアクセスすることができます。

- Web アプリケーションにサーブレットをインクルードして導入する
- デフォルトの仮想サーバでサーブレットを構成する。iPlanet Web Server 4.x との下位互換用

詳細は、iPlanet Web Server の『サーブレットに関するプログラマーズガイド』を参照してください。

参照先

iPlanet Web Server 6.0 でサーブレットを使用する方法については、iPlanet Web Server の『サーブレットに関するプログラマーズガイド』を参照してください。

Servlet API を使ってサーブレットを作成する方法については、次の Sun の Web サイトにある Java Servlet API のマニュアルを参照してください。

<http://java.sun.com/products/servlet/index.html>

JSP の作成については、次の Sun の JavaServer Pages の Web サイトを参照してください。

<http://java.sun.com/products/jsp/index.html>

NSAPI

NSAPI (Netscape Server Application Programming Interface) とは、サーバに拡張機能を実装するための一連の C 関数のことです。このような拡張機能は、サーバプラグインとして知られています。

NSAPI を使って、iPlanet Web Server の機能を拡張するプラグインを作成することができます。NSAPI プラグインは 1 つまたは複数の Server Application Functions (SAF) を定義します。SAF を開発してカスタム認証やカスタムログ作成を実行し、iPlanet Web Server が要求を処理する方法を変更することができます。

ファイル `obj.conf` には (指令と呼ばれる) 命令があり、クライアントから受け取った要求を処理する方法をサーバに伝えます。それぞれの命令は、サーバ初期化中または要求処理のある段階で実行されます。それぞれの命令が Server Application Function (SAF) を呼び出します。

たとえば、要求メソッドが GET で、要求されたリソースのタイプが `text/html` の場合、次のような命令が呼び出されます。この命令が、トレーラ引数 `<H4>Served by 6.0</H4>` を持つ `append-trailer` 関数を呼び出します。`append-trailer` 関数は要求されたリソースだけをクライアントに返し、与えられたトレーラを追加します。次の例では HTML ファイルを返します。

```
Service method=GET type="text/html" fn=append-trailer
trailer="<H4><font color=green>Served by 6.0</font></H4>"
```

iPlanet Web Server 6.0 には、定義済みの SAF が同梱されています。また NSAPI 関数のライブラリも提供されているため、ユーザが独自の SAF を開発してサーバが要求を処理する方法を変更することもできます。

NSAPI を有効にする

NSAPI そのものは有効にしません。NSAPI を使用して Server Application Functions (SAF) を開発し、`obj.conf` ファイルで使用します。`obj.conf` はサーバの操作には欠かせないファイルです。このファイルがないと、要求の処理方法に関する命令を検索できないためサーバは動作できません。

SAF を新しく定義する場合は、`server_root/plugins/include` にあるヘッダー関数 `nsapi.h` をインクルードしてすべての NSAPI 関数にアクセスできるようにします。

NSAPI プラグイン (SAF) をインストールする

カスタマイズされた SAF を含む新しい NSAPI プラグインをサーバに読み込むには、`magnus.conf` に `Init` 指令を追加して、新しい SAF を共有ライブラリファイルに定義します。この指令は、次のような引数を取る `load-modules` 関数を呼び出す必要があります。

- `shlib` - 読み込む共有ライブラリ
- `funcs` - サーバで利用できる関数

参照先

次の項目については iPlanet Web Server の『NSAPI プログラマーズガイド』を参照してください。

- `obj.conf` の指令、およびサーバが要求を処理する際の処理方法の決定方法
- iPlanet Web Server 6.0 に同梱されている定義済みの SAF
- カスタム SAF が作成できる NSAPI 関数
- カスタム SAF の作成方法
- `load-modules` を呼び出す `magnus.conf` に `Init` 指令を追加してカスタム SAF を iPlanet Web Server に読み込む方法

アクセス制御 API

アクセス制御 API は、iPlanet Web Server 上で、誰が何に対するアクセス権を持つかをプログラムで制御する C 言語の API です。

アクセス制御リスト (ACL) では、誰がどのようなアクセス権をサーバ上のどのリソースに対して持っているかを決定します。それぞれの ACL にはアクセス制御エントリのリストがあります。たとえば次のアクセス制御エントリでは、`/private` で始まる URI を持つリソースに対しては誰もアクセスできません。

```
acl "uri=/private/*";
deny (all)
(user = "anyone");
```

アクセス制御リストを作成するには、サーバマネージャの「Preferences」タブにある「Restrict Access」ページを使います。サーバが使用する ACL を含むファイルを編集することもできます。

デフォルトのアクセス制御リストは `server_root/httpacl` ディレクトリにあります。デフォルトの ACL ファイルは `generated.https-server_id.acl` です。

`genwork.https-server_id.acl` と呼ばれるファイルもあります。これは、ユーザインタフェースを使って作業しているときに行なった変更を保存して適用するまでサーバで使用される作業用のコピーです。ACL ファイルを編集する場合、`genwork` ファイルで作業した後に `Server Manager` で変更を読み込んで適用することがあります。

iPlanet Web Server 6.0 では、複数の ACL ファイルを設定して参照できます。詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』にある `server.xml` ファイルについての説明を参照してください。

アクセス制御 API では、アクセス制御リスト (ACL) の操作、ACL ファイルの読み込みと書き込み、およびサーバにあるリソースの評価とアクセスのテストができます。

また、独自の認証属性も定義できます。たとえば電子メールのアドレスに基づいてユーザを認証したり、ユーザが参照するリソースの URL に基づいて認証することができます。たとえば次のようになります。

```
allow (read) referer="*www.acme.com*"
```

また、独自の認証方法やデータベースに基づいてクライアントを認証することもできます。

新しい認証サービスを登録

設定した認証属性をサーバが使用できるようにするには、NSAPI のプラグインである `Loadable Authentication Service (LAS)` を独自に定義する必要があります。サーバへの読み込みは通常の方法で行い、`magnus.conf` に次の指令を追加します。

- `load-modules` 関数を呼び出して共有ライブラリを読み込む `Init` 指令
- 初期化関数を呼び出す `Init` 指令

参照先

ACL API の使い方は、『`Access Control Programmer's Guide`』を参照してください。ACL ファイルを編集するための構文は、このマニュアルの「付録 A」を参照してください。

仮想サーバの ACL ファイルの設定については、iPlanet Web Server の『NSAPI プログラマーズガイド』にある `server.xml` ファイルについての説明を参照してください。

iPlanet Web Server 6.0 のアクセス制御 API への変更については、`server_root/plugins/include/nsacl/aclapi.h` ファイルのコメントを参照してください。

証明書マッピング API

証明書マッピング API は、証明書マッピングを管理するデータ構造と関数で構成されています。

ユーザがサーバにクライアント証明書を送信して iPlanet サーバへの認証を行う場合、サーバは証明書の情報を使ってユーザのエントリをユーザのディレクトリで検索します。

このプロセスの一部は、`certmap.conf` ファイルを編集して設定することができます。このファイルでは、次のような内容を指定します。

- ユーザのエントリがあるディレクトリをサーバが検索する方法
- ユーザの証明書とサーバが提示する証明書との間で、これらが一致していることを確認する手順を、サーバでさらに実行するかどうか

このファイルの詳細については、第 2 章「構成ファイル」を参照してください。

この「ディレクトリエントリの証明書」プロセスをプログラムで変更することもできます。iPlanet サーバにはこのプロセスを制御できる一連の API 関数(ここでは証明書マッピング API 関数と呼びます)が設定されています。独自の関数を作成して、ディレクトリで証明書サブジェクトのエントリを検索する方法をカスタマイズすることができます。

この API を使うには、Directory SDK のコピーが必要です。この SDK のコピーは次の Web サイトからダウンロードできます。

<http://developer.iplanet.com/>

参照先

証明書マッピング API の使い方は、『Certificate-Mapping Programmer's Guide』を参照してください。

API の要約

次の表に、iPlanet Web Server 6.0 で使用できる API を示します。

表 1-2 iPlanet Web Server 6.0 で使用できる API

API/ インタフェース/ プロトコル	言語	マニュアル
動的な内容を生成するインタフェース		
カスタム HTML タグのサーバによる構文解析	C	第 3 章「サーバによる HTML タグの構文解析」
Java サブレット	Java	iPlanet Web Server の『サブレットに関するプログラマーズガイド』
JavaServer Pages	HTML と追加の JSP タグ	iPlanet Web Server の『サブレットに関するプログラマーズガイド』
CGI (要求ごとに 1 プロセス)	C、C++、Perl、シェル、およびその他の言語	『The Common Gateway Interface』
サーバのプラグインを作成する API		
NSAPI (プロセス内共有オブジェクト /DLL)	C、C++	iPlanet Web Server の『NSAPI プログラマーズガイド』
アクセス制御 API	C、C++	『Access Control Programmer's Guide』
証明書マッピング API	C、C++	『Certificate-Mapping Programmer's Guide』

旧バージョンからの変更内容

iPlanet Web Server の旧バージョンからの変更内容は、次の節で簡単に説明します。

- iPlanet Web Server 3.x 以降の API の変更内容
- iPlanet Web Server 4.0.x 以降の API の変更内容
- iPlanet Web Server 4.1.x 以降の API の変更内容

構成ファイル固有の情報は、付録 A 「iWS 4.x と 6.0 の設定の変更点」を参照してください。

iPlanet Web Server 3.x 以降の API の変更内容

- 新しい API が追加され、カスタマイズされたタグのサーバによる構文解析が NSAPI プラグインとして定義された。詳細は第 3 章「サーバによる HTML タグの構文解析」を参照
- サーバサイド Java アプレット (HttpApplets) はサポートされていない。代わりに Java サブレットを使用
- エージェント API はサポートされていない
- NSAPI に新機能

iPlanet Web Server 4.0.x 以降の API の変更内容

- Java サブレットバージョン 2.2.1 と JavaServer Pages 1.1 をサポート
- HTTP/1.1 cookie をサポート
- CGI 変数の説明がこの章の「CGI 変数」の追加
- <SERVLET> タグを使って HTML ページの SSI としてサブレットを呼び出し可能。第 3 章「サーバによる HTML タグの構文解析」を参照
- NSAPI の新機能

iPlanet Web Server 4.1.x 以降の API の変更内容

- サブレットなどのプログラムで、サーバ全体ではなく仮想サーバを変更。(iPlanet Web Server 4.1 にあるようなプログラムを追加する場合、仮想サーバを 1 つだけ設定できる)
- Java Servlet 2.2 API で仕様説明されている Web アプリケーションに対応
- NSAPI の新機能。詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照
- 一部の構成ファイルの変更。詳細は付録 A 「iWS 4.x と 6.0 の設定の変更点」を参照
- アクセス制御 API の変更。詳細は、`server_root/plugins/include/nsacl/aclapi.h` ファイルのコメントを参照

構成ファイル

構成ファイルは、iPlanet Web Server の動作をコントロールします。この章では、構成ファイルごとに **目的**、**場所**、および**内容**または**構文** を説明し、ファイルで使用できる指令やパラメータをすべて表にして簡単に説明します。他のマニュアルで指令やパラメータを詳しく説明している場合は、**関連項目**に参照先を示します。

iPlanet Web Server 4.x 以降の構成ファイルの変更点については、付録 A 「iWS 4.x と 6.0 の設定の変更点」を参照してください。

次の設定ファイルをアルファベット順に説明します。

- backups.conf
- certmap.conf
- cjava.properties
- cluster.xml
- contexts.properties
- cron.conf
- dbswitch.conf
- iwsstats.xml
- jvm12.conf
- magnus.conf
- mime.types
- ns-cron.conf
- nsfc.conf
- obj.conf
- password.conf
- rules.properties

backups.conf

- server.xml
- servers.lst
- servlets.properties
- web.xml
- web-apps.xml

backups.conf

目的

構成ファイルのバックアップを追跡します。

場所

server_root/https-admserv/conf_bk

server_root/https-server_id/conf_bk

構文

file:path_to_backup:version:timestamp:original_path

...

backup_version_history

...

内容

backups.conf:Version 4.0

https-admserv.acl:httpsacl/genwork.https-admserv.acl:2:952103058:httpsacl/genwork.h
ttps-admserv.acl

magnus.conf:https-admserv/conf_bk/magnus.conf:2:952103070:https-admserv/config/ma
gnus.conf

obj.conf:https-admserv/conf_bk/obj.conf:2:952103060:https-admserv/config/obj.conf

mime.types:https-admserv/conf_bk/mime.types:2:952103060:https-admserv/config/mime
.types

jvm12.conf:https-admserv/conf_bk/jvm12.conf:2:952103068:https-admserv/config/jvm1
2.conf

servlets.properties:https-admserv/conf_bk/servlets.properties:2:952103068:https-a
dmserv/config/servlets.properties

contexts.properties:https-admserv/conf_bk/contexts.properties:2:952103068:https-a
dmserv/config/contexts.properties

rules.properties:https-admserv/conf_bk/rules.properties:2:952103068:https-admserv
/config/rules.properties

952103058:https-admserv.acl/1::


```

952103060:https-admserv.acl/1:magnus.conf/1::
952103060:https-admserv.acl/1:magnus.conf/1:obj.conf/1::
952103060:https-admserv.acl/1:magnus.conf/1:obj.conf/1:mime.types/1::
952103068:https-admserv.acl/1:magnus.conf/1:obj.conf/1:mime.types/1:jvm12.conf/1:
:
952103068:https-admserv.acl/1:magnus.conf/1:obj.conf/1:mime.types/1:jvm12.conf/1:
servlets.properties/1::
952103068:https-admserv.acl/1:magnus.conf/1:obj.conf/1:mime.types/1:jvm12.conf/1:
servlets.properties/1:
    contexts.properties/1::
952103068:https-admserv.acl/1:magnus.conf/1:obj.conf/1:mime.types/1:jvm12.conf/1:
servlets.properties/1:
    contexts.properties/1:rules.properties/1::
952103068:https-admserv.acl/2:magnus.conf/2:obj.conf/2:mime.types/2:jvm12.conf/2:
servlets.properties/2:
    contexts.properties/2:rules.properties/2::Added ExtraPath for
Java.--EOF--

```

表 2-1 backup.conf

項目	説明
<i>file</i>	バックアップを作成するファイル名。たとえば <code>server.xml</code> 、 <code>obj.conf</code> など
<i>path_to_backup</i>	バックアップファイルへのパス
<i>version</i>	ファイルのバージョン
<i>timestamp</i>	バックアップの時刻表示
<i>original_path</i>	バックアップしたファイルへのパス
<i>backup_version_history</i>	ファイルのバージョン履歴

certmap.conf

目的

name で指定した証明書を *issuerDN* で指定した LDAP エントリにマップする方法を設定します。

場所

server_root/bin/https/install/misc

server_root/userdb

構文

```
certmap name issuerDN
```

```
name:property1 [value1]
```

```
name:property2 [value2]
```

```
...
```

デフォルトの証明書は *default* と名付けられ、デフォルトの *issuerDN* は *default* と名付けられます。したがって、ファイルでは最初の *certmap* を、次のように定義する必要があります。

```
certmap default default
```

行頭に # を付けると、その行がコメントであることを表示します。

関連項目

iPlanet Web Server の『管理者ガイド』

表 2-2 certmap.conf

プロパティ	使用できる値	デフォルト値	説明
DNComps	「説明」を参照	コメントアウト	<p>ユーザエントリに証明書をマッピング中に LDAP 検索を行う場合のベース DN の作成に使用する。値は次のとおり</p> <ul style="list-style-type: none"> コメントアウトの場合は、証明書のユーザ DN を使用する 空白の場合は、LDAP ツリー全体を検索する (DN == suffix) 属性がコンマで区切られている場合は、DN を作成する

表 2-2 certmap.conf (続き)

プロパティ	使用できる値	デフォルト値	説明
FilterComps	「説明」を参照	コメントアウト	ユーザエントリに証明書をマッピング中に LDAP 検索を行う場合のフィルタの作成に使用する。値は次のとおり <ul style="list-style-type: none"> コメントアウトまたは空白の場合は、"objectclass=*" にフィルタを設定する 属性がコンマで区切られている場合は、フィルタを作成する
verifycert	on または off	off (コメントアウト)	証明書を検証するかどうかを指定する
CmapLdapAttr	LDAP 属性名	certSubjectDN (コメントアウト)	証明書の DN を持つ LDAP データベースの属性名を指定する
library	共有ライブラリまたは dll へのパス	なし	カスタム証明書マッピングコードへのライブラリパスを指定する
InitFn	初期化関数名	なし	library が参照する証明書マッピングコードの初期化関数を指定する

cjava.properties

目的

サブレットと JVM のエラーメッセージを定義します。

場所

`server_root/bin/https/res`

構文

`error = message`

エラーは編集できないので、ここでは扱いません。メッセージの編集はできますが、お勧めしません。

cluster.xml

目的

サーバのバックアップ用クラスタとサーバファームのフェイルオーバーを定義します。このファイルは、少なくとも1つのクラスタが定義されている場合にだけ存在します。

場所

`server_root/https-admserv/config`

構文

ほとんどのファイルには次のような基本的な XML 構文があり、要素が入れ子になっています。

```
<ELEMENT attribute="value" attribute="value" ... >
  <SUBELEMENT attribute="value" attribute="value" ... />
</ELEMENT>
```

表 2-3 では要素を太字で表して属性と区別しています。

関連項目

iPlanet Web Server の『管理者ガイド』

表 2-3 cluster.xml

要素 / 属性	使用できるサブ要素または値	説明
CLUSTER	MASTER	Web サーバのクラスタを定義する
id	テキスト文字列	クラスタの ID
MASTER	SLAVE	クラスタのマスターサーバを定義する
id	テキスト文字列	マスターの ID
hostname	通常は <code>server_id</code>	マスターのホスト名
adminport		マスターの管理ポート
instance	<code>https-server_id</code>	マスターのサーバインスタンス名
SLAVE	(なし)	クラスタのスレーブサーバを定義する
id	テキスト文字列	スレーブの ID
hostname	通常は <code>server_id</code>	スレーブのホスト名
adminport		スレーブの管理ポート
instance	<code>https-server_id</code>	スレーブのサーバインスタンス名

表 2-3 cluster.xml (続き)

要素 / 属性	使用できるサブ要素または値	説明
protocol	http, https	クライアントとの通信に使用するプロトコル
substitute	マスターまたはスレーブの id、または null	このサーバが停止した場合の代用サーバの ID

contexts.properties

目的

iPlanet Web Server 4.x との下位互換用です。サーブレットを構成する代わりに、web-apps.xml を使用することをお勧めします。

コンテキストを定義し、複数のサーブレットでデータを交換してお互いのフィールドにアクセスできるようにします。コンテキストは、仮想サーバの定義やコードの分離に便利です。デフォルトのコンテキストは global です。iPlanet Web Server 6.0 では、デフォルトの仮想サーバだけをサポートします。

場所

server_root/https-admserv/config

server_root/https-admserv/conf_bk

server_root/https-server_id/config

server_root/https-server_id/conf_bk

構文

context.context_name.property=value

表 2-4 に、プロパティと可能な値を示します。

関連項目

iPlanet Web Server の『サーブレットに関するプログラマーズガイド』

server.xml ファイルおよび web-apps.xml ファイル

付録 A 「iWS 4.x と 6.0 の設定の変更点」

次の Web サイトの Servlet 2.2 API 仕様

<http://java.sun.com/products/servlet/index.html>

表 2-4 contexts.properties

プロパティ	使用できる値	デフォルト値	説明
sessionmgr	セッションマネージャのオブジェクト	com.iplanet. server.http. session. IWSessionManager (すべてを1行に記述し、ダッシュは使用しない)	コンテキストのセッションマネージャ名。 MMapSessionManagerなどのセッションマネージャが、インスタンス化できるのは、サーバ内で1度だけ
sessionmgr.initArgs	コンマで区切った <i>name=value</i> の組み合わせ	セッションマネージャによって異なる	セッションマネージャ固有のパラメータのリスト。詳細は、iPlanet Web Server の『サーブレットに関するプログラマーズガイド』を参照
initArgs	コンマで区切った <i>name=value</i> の組み合わせ	initial=0	コンテキスト属性の追加リスト
respondCookieVersion	cookie のバージョン番号	0	特定の cookie バージョンに応答するかどうかをサーバに伝える
tempDir	パス	/tmp	一時ディレクトリに Servlet API 2.2 プロパティを設定する。スラッシュ (/) だけを使用する
reloadInterval	秒数	5	サーバが JSP を調べてサーブレットファイルの修正を開始するまでの時間。global コンテキストだけに適用する
bufferSize	バイト数	4096	HTTP 出力ストリームの初期バッファサイズ
docRoot	スラッシュを使って指定されるパス	Web サーバのドキュメントルート	コンテキストのドキュメントルート。docRoot が指定されていない場合、Web サーバのドキュメントルートは使用されない

表 2-4 contexts.properties (続き)

プロパティ	使用できる値	デフォルト値	説明
inputStreamLengthCheck	true、false	true	Content-Length のバイト数を読み込んだら、データの読み込みを停止するように ServletInputStream に指示する
outputStreamFlushTimer	秒数	0	最後のフラッシュ以降、指定した秒数が経過したらストリームにデータをフラッシュさせる。0 に設定すると、このプロパティは無視される
uri	URI	/	コンテキストのベースとなる URI の追加接頭辞
authdb	データベース名	default	承認データベース名。このデータベースは、USERDB 要素の database 属性にある server.xml ファイルと、dbswitch.conf ファイルでも定義する必要がある
classpath	パス		このコンテキストのグローバルクラスパス
singleClassLoader	true、false	false	コンテキストのすべてのサーブレットで単独のクラスローダーを使用するかどうかをサーブレットエンジンに指示する
serverName	サーバのインスタンス名		コンテキストでサーブレットを実行するサーバインスタンスを指定するために使用される
contentTypeIgnoreFromSSI	true、false	true	true の場合、SSI から呼び出される setContentType を無視する

表 2-4 contexts.properties (続き)

プロパティ	使用できる値	デフォルト値	説明
parameterEncoding	none、auto、responseCT、または utf8 や Shift_JIS などの特定の encoding	auto	<p>次の形式のパラメータを復号化する方法を Web サーバに指示する</p> <ul style="list-style-type: none"> • <i>encoding</i>: 特定の符号化を使用する • <i>none</i>: システムのデフォルトの符号化を使用する • <i>auto</i>: 次の順序で符号化の方式を試みる。 1) charset、2) parameter Encoding 属性、3) <i>j_encoding</i> などの隠し形式フィールド。これ以外の場合は <i>none</i> と同じ • <i>responseCT</i>: 応答がある場合はコンテンツの種類による符号化を試みる。それ以外の場合は <i>none</i> に同じ
isModifiedCheckAggressive	true、false	false	<p>サーブレットローダーが依存度を積極的に確認し、変更されたサーブレットを再読み込みするかどうかを判定する</p>

cron.conf

目的

サーバがログファイルのバックアップなどの保守アクティビティを、定期的に行うようにプログラムします。ns-cron.conf ファイルで、cron.conf ファイルの起動をコントロールします。

場所

`server_root/https-admserv/config`

構文

```
<Object name=name>
Command "command"
User user
Time nn:nn
Days day day ...
</Object>
```

次に、ログのローテーションを管理する cron.conf ファイルの例を示します。

```
<Object name=https-server_id_rotatelg0>
Command "server_root/bin/https/httpadmin/bin/rotlog https-server_id"
User LocalSystem
Time 03:00
Days Sun Mon Tue Wed Thu Fri Sat
</Object>
```

表 2-5 cron.conf

指令	使用できる値	説明
name		保守アクティビティのオブジェクト名
Command		保守アクティビティを行うコマンドまたはスクリプト。任意のコマンドまたは実行ファイル
User		システムユーザ名
Time	24 時制の時刻	アクティビティを実行する時刻
Days	Sun、Mon、Tue、 Wed、Thu、Fri、Sat	アクティビティを実行する曜日

dbswitch.conf

目的

iPlanet Web Server で使用する LDAP ディレクトリを指定します。

場所

server_root/userdb

構文

```
directory name LDAP_URL
name:property1 [value1]
name:property2 [value2]
...
```

このファイルのデフォルトの内容は次のとおりです。

```
directory default null:///none
```

SSL の匿名バインディングでは、ファイルを次のように編集します。

```
directory default ldaps://directory.netscape.com:636:/dc%3Dcom
```

SSL でない匿名バインディングでは、ファイルを次のように編集します。

```
directory default ldap://directory.netscape.com:389:/dc%3Dcom
```

関連項目

iPlanet Web Server の『NSAPI プログラマーズガイド』の第 8 章

表 2-6 dbswitch.conf

プロパティ	使用できる値	デフォルト値	説明
nssessions	正の整数	8	データベースに対する LDAP 接続数
dyngroups	off、on、recursive	on	ダイナミックグループの処理方法を決定。off の場合、ダイナミックグループはサポートされない。on の場合、ダイナミックグループはサポートされる。recursive の場合、ダイナミックグループに他のグループを入れることができる
binddn	有効な DN		データベース接続に使用する DN。binddn および bindpw の両方がない場合、匿名でバインドされる
bindpw			データベース接続に使用するパスワード。binddn および bindpw の両方がない場合、匿名でバインドされる

表 2-6 dbswitch.conf (続き)

プロパティ	使用できる値	デフォルト値	説明
dcsuffix	有効な DN (LDAP URL に関連)	(なし)	要求の仮想サーバに対するベース DN のデフォルト値がある場合、接続グループの <code>servername</code> 属性は、DC ツリーの <code>dcsuffix</code> DN から検索して決定される。それ以外の場合、ベース DN のデフォルト値は、LDAP URL の <code>base</code> DN 値となる server.xml ファイルにある USERDB 要素の <code>basedn</code> 属性は、この値を無視する
digestauth	off、on	off	データベースがダイジェスト認証を行うかどうかを指定する。on の場合、特別なディレクトリサーバプラグインが必要。このプラグインのインストール方法は、iPlanet Web Server の『管理者ガイド』を参照

iwsstats.xml

目的

サーバのパフォーマンスに関する統計をレポートします。obj.conf の stats-xml SAF で設定し、この SAF を使用する場合にだけ存在します。このファイルは読み取り専用で、変更できません。

場所

次のディレクトリで、動的に生成されます。

```
server_root/https-server_id/stats-xml/iwsstats.xml
```

表示場所は次のとおりです。

```
http://server_id:port/stats-xml/iwsstats.xml
```

構文

ファイルには次のような基本的な XML 構文があり、要素が入れ子になっています。

```
<ELEMENT attribute="value" attribute="value" ... >
  <SUBELEMENT attribute="value" attribute="value" ... />
</ELEMENT>
```

表 2-7 では要素を太字で表して属性と区別しています。

関連項目

iPlanet Web Server の『NSAPI プログラマーズガイド』の第 3 章

表 2-7 iwsstats.xml

要素 / 属性	サブ要素または値	説明
stats	server	最上位レベルの統計要素。stats-xml の統計情報はすべてこの要素に含まれている。
enabled	0 (off), 1 (on)	統計情報の収集が有効かどうかを示す (on)
versionMajor		統計形式のメジャーバージョン。このバージョンの iPlanet Web Server では、値は 1 に固定されている
versionMinor		統計形式のマイナーバージョン
server	connection-queue、 thread-pool、 profile、process、 virtual-server	サーバのインスタンスを記述する
id		サーバのインスタンス ID (たとえば https-www.iplanet.com)
versionServer		iPlanet Web Server のバージョンを表す文字列 (たとえば iPlanet-WebServer-Enterprise/6.0 B1-12/20/2000 13:56 (SunOS DOMESTIC))
timeStarted	00:00:00 1/1/1970 以後の秒	このサーバのインスタンスを開始した時刻
secondsRunning		このサーバのインスタンスを開始してから の秒数
ticksPerSecond		1 秒ごとの刻み数。この値はシステムに依存する
maxProcs		最大プロセス数
maxThreads		スレッド処理の最大要求数
maxVirtualServers		仮想サーバの最大数
flagProfilingEnabled	0 (off)、1 (on)	NSAPI のパフォーマンスプロファイルが有効 (on) かどうかを示す
flagVirtualServer Overflow	0 (no), 1 (yes)	maxVirtualServers を超える仮想サーバが設定されている (yes) かどうかを示す。この属性が 1 に設定されている場合は、すべての仮想サーバの統計が追跡されているわけではない

表 2-7 iwsstats.xml (続き)

要素 / 属性	サブ要素または値	説明
connection-queue	(なし)	接続キューを表す。この場合のキューは、要求が提供される前にキューに入ったものである。iPlanet Web Server 6.0 の接続キューは 1 つしかない。今後のバージョンでは複数の接続キューに対応する予定
id		接続キューの ID
thread-pool	(なし)	magnus.conf ファイルで定義されているスレッドプール
id		スレッドプールの ID
name		スレッドプールの記号名
profile	(なし)	magnus.conf ファイルで定義されている NSAPI パフォーマンスプロファイルのバケット
id		NSAPI パフォーマンスプロファイルのバケット ID
name		NSAPI パフォーマンスプロファイルのバケットの記号名
description		NSAPI パフォーマンスプロファイルのバケットの説明
process	connection-queue-bucket、 thread-pool-bucket、 dns-bucket、 keepalive-bucket、 cache-bucket、 thread	サーバインスタンスの単独サーバプロセス
pid		このプロセスを一意に識別するオペレーティングシステムのプロセス識別子
mode	unknown、active	このプロセスが有効な場合は active を表示する
timeStarted	00:00:00 1/1/1970 以後の秒	このプロセスを開始した時刻
countConfigurations		構成が読み込まれた回数。この情報が利用できない場合は 0
connection-queue-bucket	(なし)	特定の connection-queue に関連する統計を追跡する

表 2-7 iwsstats.xml (続き)

要素 / 属性	サブ要素または値	説明
connection-queue		connection-queue 要素の ID
countTotalConnections		新しく受け入れられた接続の総数
countQueued		現在キューに入っている接続の数
peakQueued		同時にキューに入っている最大接続数
maxQueued		キューに入れることができる最大接続数
countOverflows		キューが満杯で、接続をキューに入れられなかった回数
countTotalQueued		キューに入っている接続の総数。所定の接続が何回もキューに入ることがあり、countTotalQueued が countTotalConnections 以上になることがある
ticksTotalQueued	刻み数はシステムに依存する時間の単位。 ticksPerSecond を参照	キューの中で接続が必要とした、総刻み数
thread-pool-bucket	(なし)	特定の thread-pool に関連する統計を追跡する
thread-pool		thread-pool 要素の ID
countThreadsIdle		現在アイドル状態にあるスレッドを処理する要求数
countThreads		スレッド処理の要求数
maxThreads		現在あるスレッドを処理できる最大要求数
countQueued		このスレッドプールで処理するためにキューに入っている要求数
peakQueued		同時にキューに入っている最大要求数
maxQueued		キューに入れることができる最大要求数
dns-bucket	(なし)	DNS (Domain Name System) 統計を追跡する
flagCacheEnabled	0 (off), 1 (on)	DNS のキャッシュが有効 (on) かどうかを示す
countCacheEntries		現在キャッシュにある DNS エントリの数

表 2-7 iwsstats.xml (続き)

要素 / 属性	サブ要素または値	説明
maxCacheEntries		キャッシュが格納できる最大 DNS エントリ数
countCacheHits		DNS キャッシュ検索の成功回数
countCacheMisses		DNS キャッシュ検索の失敗回数
flagAsyncEnabled	0 (off), 1 (on)	非同期 DNS のキャッシュが有効 (on) かどうかを示す
countAsyncNameLookups		非同期 DNS 名前検索の総実行回数
countAsyncAddrLookups		非同期 DNS アドレス検索の総実行回数
countAsyncLookupsInProgress		現在実行中の非同期 DNS 検索数
keepalive-bucket	(なし)	keepalive (常時接続) 統計を追跡する
countConnections		現在 keepalive モードにある接続数
maxConnections		keepalive の最大同時接続数
countHits		keepalive モードの接続により有効な要求が作成される回数
countFlushes		サーバにより keepalive 接続が閉じられた回数
secondsTimeout		アイドル状態の keepalive 接続をサーバが閉じるまでの秒数
cache-bucket	(なし)	ファイルキャッシュ (NSFC) 統計を追跡する
flagEnabled	0 (off), 1 (on)	ファイルキャッシュが有効 (on) かどうかを示す
secondsMaxAge	秒数	ファイルキャッシュの最大エン트리期間
countEntries		現在ファイルキャッシュにあるエン트리数
maxEntries		ファイルキャッシュが同時に格納できる最大キャッシュエントリの数
countOpenEntries		オープンファイルに関連付けられているエン트리数
maxOpenEntries		ファイルキャッシュが同時に格納できる、オープンファイルに関連付けられている最大キャッシュエントリの数

表 2-7 iwsstats.xml (続き)

要素 / 属性	サブ要素または値	説明
sizeHeapCache	バイト数	キャッシュファイルの内容が使用するヒープ総量
maxHeapCacheSize	バイト数	ファイルキャッシュがキャッシュされたファイル内容に使用する最大ヒープ総量
sizeMmapCache	バイト数	メモリにマップされたファイルの内容が使用するアドレス空間の総量
maxMmapCacheSize	バイト数	メモリにマップされたファイルの内容が使用するファイルキャッシュのための、アドレス空間の総量の最大値
countHits		キャッシュエントリ検索の成功回数
countMisses		キャッシュエントリ検索の失敗回数
countInfoHits		ファイル情報検索の成功回数
countInfoMisses		ファイル情報検索の失敗回数
countContentHits		ファイル内容検索の成功回数
countContentMisses		ファイル内容検索の失敗回数
thread	request-bucket, profile-bucket	要求処理スレッドを表す
mode	unknown、idle、DNS、 request、 processing、 response、updating	スレッドの最新状態
timeStarted	00:00:00 1/1/1970 以後の 秒	このスレッドを開始した時刻
connection-queue		スレッドが提供する connection-queue の ID
virtual-server		スレッドが最後に提供した virtual-server の ID
virtual-server	request-bucket、 profile-bucket	仮想サーバを表す
id		仮想サーバの ID
mode	unknown、active	この仮想サーバが有効な場合は active を 表示する

表 2-7 iwsstats.xml (続き)

要素 / 属性	サブ要素または値	説明
hosts		この仮想サーバでサービスを受けるソフトウェアの仮想サーバホスト名 (たとえば www.foo.com foo.com foo.isp.com)
interfaces		仮想サーバが構成されているインタフェース (待機ソケット) (たとえば 192.168.1.2:80 192.168.1.2:443)
request-bucket	(なし)	要求に関連する統計を追跡する
method		最後にサービスを受けた要求のメソッド (たとえば GET)
uri		最後にサービスを受けた URI (たとえば /index.html)
countRequests		サービスを受けた要求数
countBytesReceived		受信バイト数。0 の場合は、この情報が利用できない
countBytesTransmitted		送信バイト数。0 の場合はこの情報が利用できない
rateBytesTransmitted	毎秒のバイト数	サーバで定義した間隔でデータを送信する速度。0 の場合はこの情報が利用できない
countOpenConnections		オープン接続数。0 の場合はこの情報が利用できない
count2xx		200 レベル応答の送信数
count3xx		300 レベル応答の送信数
count4xx		400 レベル応答の送信数
count5xx		500 レベル応答の送信数
countOther		200、300、400、500 レベル以外の応答の送信数
count200		200 応答の送信数
count302		302 応答の送信数
count304		304 応答の送信数
count400		400 応答の送信数
count401		401 応答の送信数
count403		403 応答の送信数

表 2-7 iwsstats.xml (続き)

要素 / 属性	サブ要素または値	説明
count404		404 応答の送信数
count503		503 応答の送信数
profile-bucket	(なし)	profile 要素に関連する統計を追跡する
profile		profile 要素の ID
countCalls		NSAPI SAF への呼び出し数
countRequests		処理された要求数
ticksDispatch	刻み数はシステム依存型の時間単位。 ticksPerSecond を参照	ディスパッチ要求に使用される刻み数
ticksFunction	刻み数はシステム依存型の時間単位。 ticksPerSecond を参照	NSAPI SAF で使用される刻み数

jvm12.conf

目的

Java 仮想マシンの設定を変更できます。

場所

server_root/https-admserv/config

server_root/https-admserv/conf_bk

server_root/https-server_id/config

server_root/https-server_id/conf_bk

構文

```
[JVMConfig]
```

```
setting=value
```

```
...
```

関連項目

iPlanet Web Server の『サーブレットに関するプログラマーズガイド』

表 2-8 jvm12.conf

設定	使用できる値	デフォルト値	説明
<i>variable</i>	任意の JVM 環境変数		JVM 環境変数は、次の例のように <code>jvm.conf</code> にインクルードして値を指定することができる (1 行に記述) <code>org.omg.CORBA.ORBClass=com.inprise.vbroker.orb.ORB</code>
<code>jvm.minHeapSize</code>		1048576 (1M バイト)	Java に割り当てられた最小ヒープサイズ Solaris では、この値を 3145278 (3M バイト) に変更する。HPUX では、この値を 4194304 (4M バイト) に変更する。他のオペレーティングシステムでは 1M バイトを推奨
<code>jvm.maxHeapSize</code>		16777216 (16M バイト)	Java に割り当てられた最大ヒープサイズ
<code>jvm.enableClassGC</code>	0 (off), 1 (on)	0	クラスのガーベジコレクションを有効または無効にする
<code>jvm.verboseMode</code>	0 (off), 1 (on)	0	JVM 冗長モードを有効または無効にする。on の場合は、クラスの読み込みなどの動作内容をコメントで記録する。コメントはエラーログに表示される
<code>jvm.enableDebug</code>	0 (off), 1 (on)	0	JVM でのリモートからのデバッグを有効または無効にする
<code>jvm.printErrors</code>	0 (off), 1 (ログファイルへのログの書き込み)、 2 (stderr への書き込み)	0	vfprintf 経由のエラーレポートを有効または無効にする
<code>jvm.option</code>			ベンダー JVM オプションが設定できる
<code>jvm.profiler</code>			プロファイラを指定。Intuitive Systems の optimizeit プロファイラを使用する場合は、OPTIDIR も設定する必要がある

表 2-8 jvm12.conf (続き)

設定	使用できる値	デフォルト値	説明
jvm.disableThreadRecycling	0 (off)、1 (on)	0	スレッドの再利用を有効または無効にする。on の場合、サーバは必ずグローバルスコープスレッドを作成してサーブレットを実行する。それ以外の場合、スレッド処理要求がグローバルスコープにない場合にだけグローバルスコープスレッドが作成される
jvm.serializeAttach	0 (off)、1 (on)	0	on の場合、JVM に接続するスレッドは直列化される。デフォルトの off の場合、スレッドは並列で JVM に接続する
jvm.stickyAttach	0 (off)、1 (on)	0	このパラメータを 1 に設定すると、スレッドは JVM に接続されていることを記憶し、その状態を維持する
jvm.trace		5	追跡レベルを決定する。サーブレットおよび JSP のデバッグでは、レベル 7 を推奨する。レベル 5 では、サーブレットエンジンのメッセージが表示される。レベル 6 では、サーブレットおよび JSP エンジンのメッセージが表示される。レベル 7 では、上記のほか、例外がブラウザに表示される
jvm.allowExit	0 (off)、1 (on)	0	プロセス終了を有効または無効にする
java.compiler		NONE	Java コンパイラを指定する。JIT (Just In Time) コンパイラのオンとオフを切り換えるオプションについては、JVM のマニュアルを参照。 jvm.enableDebug が on の場合は、NONE に設定する
OPTITDIR	パス	*	プロファイラが optimizeit の場合、プロファイラへのパスを指定する
nes.jsp.enableddebug	0 (off)、1 (on)	1	冗長 JSP コンパイル追跡を有効または無効にする

表 2-8 jvm12.conf (続き)

設定	使用できる値	デフォルト値	説明
jvm.include.CLASSPATH	0 (off)、1 (on)	1	jvm.classpath 設定に CLASSPATH 環境変数値を含めるかどうかを指定する
nes.jsp.forkjavac	0 (off)、1 (on)	0	on の場合、JSP の Java コンパイルが別のプロセスで実行される
jvm.serializeFirstRequest	0 (off)、1 (on)	1 は Linux、AIX、および Compaq (DEC)。 0 はその他のプラットフォーム。	on の場合、1 つの要求スレッドだけが読み込まれ、サーブレットオブジェクトを構築する。サーブレットを読み込んで初期化すると、同じサーブレットへの新しい要求が並列して発生する。この設定は Linux、AIX、および Compaq (DEC) では on にする必要がある
jvm.classpath	スラッシュのあるパスだけ		JVM に依存する JAR ファイルへのパスを指定。必要に応じて他のクラスパスの値も入力

* N :/App/IntuitiveSystems/OptimizeIt30D、 N は OptimizeIt がインストールされているドライブ

magnus.conf

目的

このファイルには、サーバ機能に影響を与えるグローバル変数を含みます。また、このファイルは、サーバ起動時は読み取り専用になっています。

場所

`server_root/https-admserv/config`

`server_root/https-admserv/conf_bk`

`server_root/https-server_id/config`

`server_root/https-server_id/conf_bk`

構文

Init 関数には次の構文があります。

```
Init fn=function param1="value1" ...paramN="valueN"
```

表 2-9 では関数を太字で表してパラメータと区別しています。

指令には次の構文があります。

directive value

関連項目

iPlanet Web Server の『NSAPI プログラマーズガイド』の第7章

付録 A 「iWS 4.x と 6.0 の設定の変更点」

Init 関数

表 2-9 magnus.conf Init 関数

関数 / パラメータ	使用できる値	デフォルト値	説明
cindex-init			拡張インデックスのデフォルト特性を変更する
opts	s	(なし)	(省略可) オプションを有効にする文字列。現在使用できるのは次のオプションだけ <ul style="list-style-type: none"> s を指定すると、ディレクトリに含まれるすべての HTML ファイルから、<TITLE> タグの部分をスキャンしてインデックスを生成し、説明フィールドに表示する。<TITLE> タグの部分は、ファイルの先頭から 255 文字以内であること
widths	コンマで区切られた文字数	列のタイトルを表示するために必要な最小数	(省略可) インデックス表示に使用する名前、最終変更日、サイズ、説明の列幅を、それぞれ指定する 最後の 3 つの値は、0 を設定すると、その列が非表示になる。名前の列は非表示にできない
timezone	GMT または local	local	(省略可、iPlanet Web Server 4.x の場合のみ) 最後に変更した時刻を、その地域の標準時またはグリニッジ標準時で示す

表 2-9 magnus.conf Init 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
format	UNIX 関数 strftime() の形式	%d - %b - %Y %H:%M	(省略可、iPlanet Web Server 4.x の場合のみ) 最後に変更した日 付の書式を決定
ignore	ワイルド カード形式	.*	(省略可) インデックスの生成中 にサーバが無視すべきファイル の名前を、ワイルドカードの形 式で指定する。ピリオド(.)で始 まるファイル名は常に無視され る
icon-uri		/mc-icons/	(省略可) ファイルアイコン (.gif ファイル) の URL 生成時 に、index-common 関数が使 用する URI 接頭辞を指定する。 icon-uri がデフォルトと異な る場合は、NameTrans 指令の pfx2dir 関数を変更してサー バがアイコンを検索できるよう にする必要がある
define-perf-bucket			パフォーマンスバケットを作成 すると、obj.conf で SAF のパ フォーマンスを測定できる (71 ページの「bucket パラメータ」を 参照)。この関数は、perf-init 関数が有効な場合にだけ機能す る
name			バケット名。たとえば cgi-bucket
description			バケットの測定内容の説明、た とえば CGI Stats
dns-cache-init			DNS キャッシュを設定する
cache-size	32 ~ 32768 (32K)	1024	(省略可) キャッシュに含まれる エントリの数を指定する
expire	1 ~ 31536000 秒 (1 年)	1200 秒 (20 分)	(省略可) キャッシュエントリの 有効期限を秒で指定する
flex-init			フレキシブルログ作成システム を初期化する

表 2-9 magnus.conf Init 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
<i>logFileName</i>	パス名または ファイル名		ログファイルへの絶対パスまたはサーバの logs ディレクトリに関連するファイル名。以下の例では、ログファイル名は access で、パスは /logdir/access access="/logdir/access"
<i>format.logFileName</i>			ログファイルのログエントリ形式を、それぞれ指定する。詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照
buffer-size	バイト数	8192	グローバルログバッファのサイズを指定する
num-buffers		1000	使用するログバッファの最大サイズを指定する
flex-rotate-init			ログのローテーションを有効にする
rotate-start	24 時間形式で時刻を表す 4 桁の文字列		ローテーションの開始時刻。たとえば、0900 は午前 9 時、2100 は午後 9 時を表す。
rotate-interval	分の値		ログローテーション間の経過時間を分で表す。
rotate-access	yes、no	yes	(省略可) common-log、flex-log、および record-useragent ログをローテーションさせるかどうか決定する
rotate-error	yes、no	yes	(省略可) エラーログをローテーションさせるかどうか決定する
rotate-callback	パス		(省略可) ユーザが供給したプログラムのファイル名を指定し、ログファイルローテーションを実行する。ローテーションをしたログファイルのローテーション後の名前を、ファイルのパラメータとしてプログラムに渡す

表 2-9 magnus.conf Init 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
init-cgi			CGI プログラムのデフォルト設定を変更する
timeout	秒数	300	(省略可) スクリプトの終了の前に、サーバが CGI からの出力を待つ秒数を指定する
cgistub-path			(省略可) CGI スタブバイナリへのパスを指定する。指定しない場合、iPlanet Web Server はサーバのインスタンスの config ディレクトリに関連するディレクトリを、次の順序で検索する。 ../private/Cgistub、次に ../../bin/https/bin/Cgistub
env-variable			suid Cgistub のインストールは、第 1 章「概要」を参照 (省略可) サーバが CGI の環境に組み込む環境変数の名前と値を指定する
init-clf			共通ログサブシステムを初期化する
logFileName	パス名または ファイル名		ログファイルへの絶対パスまたはサーバの logs ディレクトリに関連するファイル名を指定する
init-uhome			ユーザのホームディレクトリ情報を読み込む
pwfile			(省略可) /etc/passwd 以外のファイルへのファイルシステムの絶対パスを指定する。指定しない場合は、デフォルトの UNIX パス (/etc/passwd) が使用される
load-modules			サーバに共有ライブラリを読み込む

表 2-9 magnus.conf Init 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
shlib			共有ライブラリ、ダイナミックリンクライブラリへの絶対パス、またはサーバの構成ディレクトリに関連するファイル名を指定する
funcs	コンマ区切りのリスト、スペースなし		他の Init 指令や Service 指令で使用できるようにした、共有ライブラリやダイナミックリンクライブラリの関数名のリスト。関数名には、アンダースコア () の代わりにダッシュ (-) を使用することもできる
NativeThread	yes、no	yes	(省略可) 使用するスレッドモデルを指定する。no に設定すると、ライブラリのルーチンにはユーザレベルのスレッドが使用される。yes に設定すると、カーネルレベルのスレッドが有効になる
pool			カスタムのスレッドプール名。thread-pool-init で指定する
nt-console-init			NT コンソールを有効にする。NT コンソールは、標準出力とエラーストリームを表示するコマンドラインシェルである
stderr	console		エラーメッセージを NT コンソールで表示させる
stdout	console		出力を NT コンソールで表示させる
perf-init			パフォーマンスバケットを使って、システムのパフォーマンス測定を有効にする
disable	true、false	true	true の場合は関数を無効にする
pool-init			プールされたメモリーの割り当てを設定する

表 2-9 magnus.conf Init 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
free-size	1048576 バイト以下		(省略可) フリーブロックリストの最大バイトサイズ
disable	true、false	false	(省略可) true の場合は、プールされたメモリの使用を無効にする
register-http-method			新しい HTTP メソッドの登録によって、HTTP プロトコルを拡張する
methods	コマンドで区切られたリスト		登録中のメソッド名
stats-init			XML 形式によるパフォーマンス統計のレポートを有効にする
profiling	yes、no	no	バケットを使って NSAPI パフォーマンスのプロファイリングを有効にする。perf-init で有効にすることもできる
update-interval	1 以上	5	サーバ内で統計の更新をする間隔 (秒)
virtual-servers	1 以上	1000	統計を追跡する仮想サーバの最大数。構成されている仮想サーバの数よりも大きい数字を設定しておく必要がある
thread-pool-init			追加のスレッドプールを構成する
name			スレッドプール名
maxthreads			プールの最大スレッド数
minthreads			プールの最小スレッド数
queueSize	バイト数		プールのキューサイズ
stackSize	バイト数		内蔵 (カーネル) スレッドプールにある各スレッドのスタックサイズ

指令

表 2-10 magnus.conf 指令

指令	使用できる値	デフォルト値	説明
ACLCacheLifetime	任意の秒数	120	キャッシュエントリの有効期間を秒数で決定する。キャッシュのエントリが参照されるたびに期間が計算され、ACLCacheLifetime と照合される。期間が ACLCacheLifetime を超えているかこれと同じ場合、エントリは使用されない この値が 0 に設定されると、キャッシュは使用されない
ACLUserCacheSize		200	ユーザキャッシュのユーザ数を決定する
ACLGroupCacheSize		4	単独の UID/ キャッシュエントリでキャッシュするグループ ID の数を決定する
AdminLanguage	en (英語)、fr (フランス語)、de (ドイツ語)、ja (日本語)	en	サーバマネージャの言語を指定する
AsyncDNS	on、off	off	非同期 DNS の許可の有無を指定する
CGIExpirationTimeout	任意の秒数	300 (5 分) を推奨	CGI プロセスを実行してから終了するまでの最大秒数を指定する
CGIStubIdleTimeout	任意の秒数	30	この指令で設定した秒数にわたってアイドル状態にある CGIStub プロセスを、サーバが終了させる。ただし、プロセスの数が MinCGIStubs の値まで減少すると、サーバはそれ以上のプロセスを終了させない
CGIWaitPid	on、off	on	(UNIX の場合のみ) SIGCHLD 信号に対するアクションを、信号に対するシステムのデフォルトアクションにする。SHTML エンジン、exec cmd 子プロセスで明示的に待機させる
ChildRestartCallback	on、off、yes、no、true、false	no	サーバの再起動時またはシャットダウン時に、daemon_atrestart 関数を使って登録された NSAPI 関数に対して、強制的なコールバックをかける

表 2-10 magnus.conf 指令 (続き)

指令	使用できる値	デフォルト値	説明
ChunkedRequestBufferSize	任意のバイト数	8192	分割されていない要求データに対する、デフォルトのバッファサイズを決定する
ChunkedRequestTimeout	任意の秒数	60 (1分)	分割されていない要求データに対する、デフォルトのタイムアウトを決定する
ClientLanguage	en (英語)、fr (フランス語)、de (ドイツ語)、ja (日本語)	ja	クライアントメッセージの言語を指定する (File Not Found など)
ConnQueueSize	任意の接続数	5000	Web サーバに設定できる、サービス予定はあるが未処理の接続数を指定する
DefaultCharSet	有効な文字セット名	iso-8859-1	サーバのデフォルトの文字セットを指定する。クライアントの応答と管理には、デフォルトの言語が使用される
DefaultLanguage	en (英語)、fr (フランス語)、de (ドイツ語)、ja (日本語)	ja	サーバのデフォルト言語を指定する。クライアントの応答と管理には、デフォルトの言語が使用される
DNS	on、off	on	サーバにアクセスするクライアントの DNS 検索を、サーバが実行するかどうか指定する
ErrorLog	パス	(なし)	サーバのエラーを記録するディレクトリを指定する
ErrorLogDateFormat	C ライブラリ関数 strftime はマニュアルを参照	%d/%b/%Y:%H:%M:%S	エラーログのデータ書式
ExtraPath	パス	(なし)	PATH 環境変数に、指定したディレクトリ名を設定する。Windows NT で Java を設定する場合に使用する。デフォルト値はないので、値を設定する必要がある

表 2-10 magnus.conf 指令 (続き)

指令	使用できる値	デフォルト値	説明
HeaderBufferSize	任意の バイト数	8192 (8 K バイ ト)	クライアントの要求データを読み取る要求処理スレッドごとに使用されるバッファのサイズ(バイト)。要求処理スレッドの最大数は RqThrottle 設定で制御する
HTTPVersion	<i>m.n</i> ; <i>m</i> はバージョンのメ ジャー番号、 <i>n</i> はマイナー番号	1.1	サーバで使用する現在の HTTP バージョン
IOTimeout	任意の秒数	ハードウェア 暗号化デバイスを使用しないサーバでは 30、使用する サーバでは 300	クライアントから届くデータをサーバが待つ秒数を指定する。タイムアウトまでにデータが届かない場合は、接続が閉じられる
KeepAliveThreads	任意の スレッド数	1	keep-alive サブシステムのスレッド数を指定する。システムが実装しているプロセッサ数の数倍であることが望ましい
KeepAliveTimeout	300 秒、最大	30	サーバが HTTP Keep-Alive 接続またはクライアントとサーバの間の常時接続を開いておく時間の最大値を決定する
KernelThreads	0 (off)、1 (on)	0 (off)	on の場合、サーバではカーネルレベルのスレッドだけが使用され、ユーザレベルのスレッドは使用されない。off の場合、ユーザレベルのスレッドだけが使用される
ListenQ	範囲はプラットフォームに よって 異なる	4096 (AIX)、 200 (NT)、128 (その他)	サーバソケットの入力接続数を指定する
LogFlushInterval	任意の秒数	30	ログフラッシュスレッドのログフラッシュ間隔を秒単位で決定する
LogVerbose	on、off	off	on の場合、デフォルトではログに記録されないサーバメッセージを含めて、すべてログに記録する

表 2-10 magnus.conf 指令 (続き)

指令	使用できる値	デフォルト値	説明
LogVsId	on、off	off	仮想サーバ ID をエラーログに表示するかどうかを決定する。複数の仮想サーバが同じログファイルを共有する場合は、LogVsId を有効にする必要がある
MaxCGIStubs	任意の CGI スタブ数	10	サーバが生成できる CGIStub プロセスの最大数を制御する。これは同時に実行できる CGIStub プロセスの最大値であり、保留中の要求の最大値ではない
MaxKeepAliveConnections	0 - 32768	256	サーバが同時に開くことができる Keep-Alive 接続と常時接続の最大値を指定する
MaxProcs	任意のプロセス数	1	(UNIX の場合のみ) サーバが同時に実行できるプロセスの最大数を指定する
MaxRqHeaders	0 - 32	32	要求の最大ヘッダ行数を指定する
MinCGIStubs	MaxCGIStubs より小さい任意の数	2	デフォルトで開始するプロセスの数を制御する
MtaHost	有効な電子メールアドレス	(なし)	サーバのエージェントが使用する SMTP メールサーバを指定する。この値を指定すると、レポートをメールのアドレスに送信できるようになる
NativePoolMaxThreads	任意のスレッド数	128	内蔵 (カーネル) スレッドプールにあるスレッドの最大数を決定する
NativePoolMinThreads	任意のスレッド数	1	内蔵 (カーネル) スレッドプールにあるスレッドの最小数を決定する
NativePoolQueueSize	負の数以外の数	0	スレッドプールのキューで待機できるスレッド数を決定する
NativePoolStackSize	負の数以外の数	0	内蔵 (カーネル) スレッドプールにある各スレッドのスタックサイズを決定する
NetSiteRoot	パス	(なし)	サーバのインスタンスが検索できる最上位ディレクトリへの絶対パス名を指定する。デフォルト値はないので、値を設定する必要がある

表 2-10 magnus.conf 指令 (続き)

指令	使用できる値	デフォルト値	説明
PidLog	ファイルへの有効なパス	(なし)	基本サーバプロセスのプロセス ID (pid) を記録するファイルを指定する
PostThreadsEarly	1 (on), 0 (off)	0 (off)	on の場合、接続の確立後で要求への応答を送信する前に、スレッドの最小数がソケットで利用できるかどうかを確認する
RcvBufSize	範囲はプラットフォームフォームによって異なる	0 (プラットフォーム固有のデフォルトを使用)	サーバのソケットでの受信バッファのサイズを制御する
RqThrottle	任意の要求数	512	サーバが各ソケットで同時に処理できる、同時要求処理スレッドの最大数を指定する
RqThrottleMin	RqThrottle より小さい任意の数	48	サーバ起動時に作成される要求処理スレッドの数を指定する。サーバへの負荷が増すと、要求処理スレッドの数も増え、RqThrottle スレッドの最大値まで作成される
Security	on、off	off	証明書をサーバのインスタンスで利用できるようにして、SSL をグローバルに有効または無効にする。 仮想サーバで SSL を使用するには、on にする必要はある
ServerConfigurationFile	ファイル名	server.xml	仮想サーバを指定するファイル名
ServerID	文字列	(なし)	https-boots.mcom.com のように、サーバ ID を指定する
#ServerRoot	パス	(なし)	サーバルートを指定する。この指令はインストール時に設定され、コメントアウトされる。他の指令とは異なり、サーバはこの指令を # で始めるよう求める。この指令は変更しないこと
SndBufSize	範囲はプラットフォームフォームによって異なる	0 (プラットフォーム固有のデフォルトを使用)	サーバのソケットでの送信バッファのサイズを制御する
SSL3SessionTimeout	5 - 86400	86400 (24 時間)	キャッシュされた SSL3 セッションが無効になるまでの秒数

表 2-10 magnus.conf 指令 (続き)

指令	使用できる値	デフォルト値	説明
SSLCacheEntries	負の数以外の整数	10000 (0 を指定すると使用される)	キャッシュできる SSL のセッション数を指定する。上限はない
SSLClientAuthDataLimit	バイト数	1048576 (1M バイト)	クライアントの証明書ハンドシェイクフェーズでバッファされるアプリケーションデータの最大容量を指定する
SSLClientAuthTimeout	任意の秒数	60	クライアントの証明書ハンドシェイクフェーズがタイムアウトしたあとの秒数を指定する
SSLSessionTimeout	5 - 100	100	キャッシュされた SSL2 セッションが無効になるまでの秒数を指定する
StackSize	バイト数	最も好ましいマシン固有のスタックサイズ	要求処理スレッドごとに最大スタックサイズを決定する
StrictHttpHeaders	on、off	off	on の場合、重複したヘッダーがある接続を拒否する
TempDir	パス	/tmp (UNIX) TEMP (Windows NT の環境変数)	サーバが一時ファイルに使用するディレクトリを指定する。UNIX では、サーバを実行するユーザがこのディレクトリを所有し、書き込める必要がある
TempDirSecurity	on、off	on	TempDir ディレクトリのセキュリティをサーバで確認するかどうかを決定する。UNIX では、TempDirSecurity off を指定すると、サーバは /tmp を一時ディレクトリとして使用できる
TerminateTimeout	任意の秒数	30	サーバが停止するまでに、すべての既存の接続が終了するのをサーバが待つ時間を秒単位で指定する
ThreadIncrement	任意のスレッド数	10	サーバの負荷増大に対処するために作成された、追加要求処理スレッドまたは新規要求処理スレッドの数

表 2-10 magnus.conf 指令 (続き)

指令	使用できる値	デフォルト値	説明
Umask	UNIX の標準的な umask 値	(なし)	UNIX の場合のみ、NSAPI の System_fopenWA() 関数および System_fopenRW() 関数を使って umask 値を指定し、別のモードでファイルを開く
UseNativePoll	1 (on), 0 (off)	1 (on)	1 (on) に設定すると、プラットフォーム固有のポーリングインタフェースが使用される。0 (off) に設定すると、KeepAlive サブシステムで NSPR のポーリングインタフェースが使用される
UseOutputStreamSize	任意のバイト数	8192 (8 K バイト)	NSAPI の net_read 関数および netbuf_grab 関数のデフォルトの出力ストリームバッファサイズを決定する
User	ログイン名、8 文字以下	(なし)	(Windows NT の場合) サーバを実行するユーザアカウントを指定し、サーバのシステム機能の制限や有効化を設定する (UNIX の場合) サーバが superuser または root で起動している場合、サーバは指定したポートにバインドされ、ユーザ ID は User 指令で指定したユーザアカウントに切り替えられる。この指令は、サーバが root で起動していない場合は無視される
WinCGITimeout	任意の秒数	60	この値より長い WinCGI プロセスはタイムアウトになると終了する

mime.types

目的

標準の MIME タイプをファイル拡張子にマップします。仮想サーバにはそれぞれ独自の mime.types ファイルを設定することができます。

場所

```
server_root/https-admserv/config
server_root/https-admserv/conf_bk
server_root/https-server_id/config
server_root/https-server_id/conf_bk
server_root/bin/https/install/misc
```

構文

```
type=type/subtype exts=ext1,ext2,...
...
enc=subtype exts=ext1,ext2,...
...
```

関連項目

iPlanet Web Server の『NSAPI プログラマーズガイド』

表 2-11 mime.types

指令	使用できる値	説明
<i>type</i>	application、 image、text、 video、audio、 perf、x-world、 x-conference、 magnus-internal	コンテンツの基本タイプ
<i>subtype</i>		コンテンツに固有のタイプ。たとえば text/html では、サブタイプは html
<i>ext1, ext2, ...</i>	コンマで区切られた ファイル拡張子のリ スト	タイプのファイル拡張子。たとえば text/html のファイル拡張子は htm または html

ns-cron.conf

目的

cron.conf ファイルを有効または無効にします。

場所

server_root/https-admserv/config

内容

ConfFile *server_root*/https-admserv/config/cron.conf

Dir /tmp

Status on

表 2-12 ns-cron.conf

指令	使用できる値	デフォルト値	説明
ConfFile	パス		cron.conf ファイルの場所
Dir	パス		一時ディレクトリの場所
Status	on,off	on	cron.conf ファイルの状態。on は有効、off は無効

nsfc.conf

目的

ファイルのキャッシュパラメータを設定します。このファイルは、デフォルトのキャッシュパラメータが変更された場合にだけ存在します。

場所

server_root/https-admserv/config

構文

parameter=value

関連項目

『Performance Tuning, Sizing, and Scaling Guide for iPlanet Web Server』

表 2-13 nsfc.conf

パラメータ	使用できる値	デフォルト値	説明
FileCacheEnable	on、off	on	ファイルのキャッシュを有効にする
CacheFileContent	on、off	on	ファイル内容のキャッシュと、MediumFileSizeLimit よりも小さい (TransmitFiles が on の場合は SmallFileSizeLimit よりも小さい) ファイル情報のキャッシュを有効にする
MaxAge	秒数	30	有効なキャッシュエントリの最大期間。一度キャッシュされたファイルのキャッシュ情報の使用可能な期間を設定する。MaxAge よりも以前のエントリは、同じファイルの新しいエントリに置き換えられる
MediumFileSizeLimit	使用可能なメモリに制限される	537600 (525K)	(UNIX の場合のみ) メモリマップにキャッシュされるファイルの最大ファイルサイズ (TransmitFiles が off の場合)
MediumFileSpace	使用可能なメモリに制限される	10485760 (10 M)	メモリマップにキャッシュされる全ファイルのファイルサイズ総計 (TransmitFiles が off の場合)
SmallFileSizeLimit	使用可能なメモリに制限される	2048 (2K)	(UNIX の場合のみ) メモリに読み込める最大ファイルサイズ
SmallFileSpace	使用可能なメモリに制限される	1048576 (UNIX、1 M)、0 (NT)	メモリに読み込めるファイルサイズの総計
TransmitFiles	on、off	on (NT)、off (UNIX)	TransmitFile システムコールの使用を有効にする。IRIX、Compaq、Solaris、Linux ではサポートされていない
MaxFiles		1024	ファイルキャッシュの最大ファイル数
HashInitSize	使用可能なメモリに制限される	0	ハッシュバケットの初期数。0 の場合、ハッシュバケットの数は $2 * \text{MaxFiles} + 1$ として動的に決定される
CopyFiles	on、off	on	(NT の場合のみ) ファイルを一時ディレクトリにコピーして、共有違反を防止する
TempDir	パス	system_temp/se rver_id	CopyFiles が on の場合、ファイルキャッシュの一時ディレクトリを指定する

obj.conf

目的

要求に対するクライアントの応答を決定します。仮想サーバにはそれぞれ独自の obj.conf ファイルを設定することができます。

場所

`server_root/https-admserv/config`

`server_root/https-admserv/conf_bk`

`server_root/https-server_id/config`

`server_root/https-server_id/conf_bk`

構文

```
directive fn=function param1="value1" ...paramN="valueN"
```

object タグの指令は、name/name または root/ppath 照合によって NameTrans 指令がタグへのフロー制御をリダイレクトする場合にだけ実行されます。object タグは次の構文を取ります。

```
NameTrans fn=function name="name" |root="path"
```

```
<object name="name" |ppath="path">
directive1
directive2
...
</object>
```

注 iPlanet Web Server 6.0 の Init 指令の記述は、magnus.conf ファイルに移動しました。

表 2-15 から表 2-21 では、関数を**太字**で表してパラメータと区別しています。

関連項目

iPlanet Web Server の『NSAPI プログラマーズガイド』の第 2 章および第 3 章

独自の関数を作成する方法は、iPlanet Web Server の『NSAPI プログラマーズガイド』の第 4 章～第 6 章を参照。

付録 A 「iWS 4.x と 6.0 の設定の変更点」

表 2-14 obj.conf

指令	説明
AuthTrans	HTTP 要求で提供される認証情報 (通常は Authorization ヘッダーで送信される) を検証し、ユーザまたはグループに変換する
NameTrans	論理 URL の要求で指定した URL を、要求されたリソースの物理ファイルシステム上のパスに変換する。別のサイトにリダイレクトすることもできる
PathCheck	NameTrans の手順で決定した物理パスのテストを実行する。通常は、このテストでパスの妥当性を検証し、要求されたリソースにクライアントがアクセスできるかどうかを確認する
ObjectType	要求されたリソースの MIME (Multi-purpose Internet Mail Encoding) タイプを決定する
Service	応答を生成し、クライアントに送信する。これには、HTTP 結果状況の設定、content-type や content-length などの応答ヘッダーの設定、応答データの作成と送信が含まれる
AddLog	エントリをログファイルに追加して、トランザクション情報を記録する
Error	直前の指令を実行した結果生じる HTTP エラーを処理する。通常、サーバは問題点および考えられる解決法を記述したカスタム HTML ドキュメントをユーザに送信してエラー処理を行う

bucket パラメータ

iPlanet Web Server では、次のパフォーマンスバケットがあらかじめ定義されています。

- default-bucket は、ユーザが定義したバケットや組み込みバケットには関連付けられていない関数の統計を記録する
- all-requests バケットは、.perf の NSAPI SAF 統計をすべて記録する。これには default-bucket の統計も含まれる

magnus.conf ファイルにパフォーマンスバケットを追加して定義することもできます (perf-init 関数および define-perf-bucket 関数を参照してください)。

obj.conf 内で、関数に bucket=cache-bucket のような bucket=bucket-name パラメータを指定することによって追加された、任意の SAF に対して、そのパフォーマンスを測定できます。bucket はあらゆる obj.conf 関数のパラメータなので、一覧表は省略します。

パフォーマンス統計は、service-dump Service 関数を使って表示します。

または、stats-xml Service 関数を使ってパフォーマンス統計を生成することもできます。バケットの使用は任意です。

パフォーマンスバケットの詳細は、『Performance Tuning, Sizing, and Scaling Guide for iPlanet Web Server』を参照してください。

AuthTrans 関数

表 2-15 obj.conf AuthTrans 関数

関数 / パラメータ	使用できる値	デフォルト値	説明
basic-auth			カスタム関数を呼び出してユーザ名とパスワードを確認する。必要に応じて、ユーザのグループを決定する
auth-type	basic	basic	使用する認証の種類を指定する
userdb			(省略可) ユーザ検証で使うユーザデータベースの絶対パスとファイル名を指定する。このパラメータは user 関数に渡される
userfn			承認を検証するユーザカスタム関数名。この関数は、load-modules であらかじめ読み込んでおく必要がある
groupdb			(省略可) ユーザデータベースの絶対パスとファイル名を指定する。このパラメータは group 関数に渡される
groupfn			(省略可) グループカスタム関数名。load-modules であらかじめ読み込んでおく必要がある
basic-ncsa			NCSA-style またはシステムの DBM データベースに対するユーザ名とパスワードを検証する。必要に応じて、ユーザのグループを決定する
auth-type	basic	basic	使用する認証の種類を指定する
dbm			(省略可) ユーザデータベースの絶対パスとファイル名をサーバのネイティブ形式で指定する。このパラメータを使用する場合は、userfile パラメータと一緒に使用しないこと
userfile			(省略可) ユーザデータベースの絶対パス名を NCSA スタイルの HTTPD ユーザファイル形式で指定する。この形式には name:password 形式を使用する行があり、password は暗号化されている。このパラメータを使用する場合は、dbm を使用しないこと。

表 2-15 obj.conf AuthTrans 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
grpfile			(省略可) 使用する NCSA スタイルの HTTPD グループ ファイルを指定する。グループファイルの各行は <code>group:user1 user2 ... userN</code> の形式で構成されており、各ユーザはスペースで区切られている
get-sslid			現在の SSL セッションに対して一意の文字列を検索し、検出した文字列を <code>Session->client</code> パラメータブロックに <code>ssl-id</code> 変数として格納する
qos-handler			仮想サーバ、仮想サーバクラス、グローバルサーバに対するサービス統計の現在の品質を検証して統計を記録し、エラーを返して QOS パラメータを実行する。正しく動作させるには、この関数を AuthTrans 関数として default オブジェクトに最初に設定する必要がある

NameTrans 関数

表 2-16 obj.conf NameTrans 関数

関数 / パラメータ	使用できる値	デフォルト値	説明
assign-name			名前付きオブジェクトで指令を処理するようサーバに指示する
from			影響を受けるパスを指定するワイルドカード形式
name			追加の名前付きオブジェクトを obj.conf に指定し、そのオブジェクトの指令をこの要求に適用する
find-pathinfo-forward	値は無視される		(省略可) サーバ関数 assign-name がデフォルトでパスの最後から逆方向に検索するのとは対照的に、 <code>ntrans-base</code> の直後のパスで PATHINFO を順方向に検索する
nostat			(省略可) 可能な限り、指定された URL でサーバが <code>stat</code> を実行しないようにする。NSAPI プラグイン URL のように <code>virtual-path</code> のパスがシステムに存在していない場合にだけ <code>nostat</code> を使用して、このような URL での不必要な <code>stat</code> の実行を避けてパフォーマンスを向上させる
document-root			要求されたリソースの <code>http://server-name/</code> の部分をドキュメントのルートディレクトリに置き換えて、ファイルシステムのパスを URL に変換する

表 2-16 obj.conf NameTrans 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
root		server_root/ docs	サーバのルートドキュメントディレクトリへのファイルシステムのパス
home-page			サーバのルートホームページ (/) への要求を、特定のファイルに変換する
path			ホームページファイルへのパスとファイル名。path がスラッシュ (/) で始まる場合、ファイルへの絶対パスとみなされる
pfx2dir			特定の接頭辞で始まる任意の URL をファイルシステムのディレクトリに変換し、場合によっては追加された名前付きオブジェクトの指令を有効にする
from			変換する URI 接頭辞。スラッシュ (/) で終了しないこと
dir			接頭辞の変換先になるローカルファイルシステムのディレクトリパス。スラッシュ (/) で終了しないこと
name			(省略可) 追加の名前付きオブジェクトを obj.conf に指定し、そのオブジェクトの指令をこの要求に適用する
find-pathinfo-forward	値は無視される		(省略可) サーバ関数 pfx2dir がデフォルトでパスの最後から逆方向に検索するのとは対照的に、ntrans-base の直後のパスで PATHINFO を順方向に検索する
redirect			クライアントを別の URL にリダイレクトする
from			要求された URI の接頭辞を指定し、照合する
url			(たいていは省略可) 完全な URL を指定し、クライアントに返す。このパラメータを使用する場合は、url-prefix を使用しないこと (逆の場合も同様)
url-prefix			(たいていは省略可) 新しい URL 接頭辞を指定し、クライアントに返す。from 接頭辞は、この URL 接頭辞でそのまま置き換えられる。このパラメータを使用する場合は、url を使用しないこと (逆の場合も同様)
escape	yes, no	yes	(省略可) URL を送信する前に util_uri_escape を実行するようにサーバに指示するフラグ

表 2-16 obj.conf NameTrans 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
strip-params			パスに埋め込まれているセミコロンで区切られたパラメータを削除する。たとえば、 <code>/dir1;param1/dir2</code> の URI は <code>/dir1/dir2</code> となる。この関数を使用する場合は、最初に表示されている NameTrans 指令であること
unix-home			ユーザのホームディレクトリの URL を、指定したディレクトリに変換する
from			変換する URL 接頭辞。通常は「/～」
subdir			ユーザのホームディレクトリにあるサブディレクトリで、ユーザの Web ドキュメントが格納されている
pwfile			(省略可) <code>/etc/passwd</code> と異なる場合の、パスワードファイルの完全なパスとファイル名
name			(省略可) 追加の名前付きオブジェクトを指定し、そのオブジェクトの指令をこの要求に適用する

PathCheck 関数

表 2-17 obj.conf PathCheck 関数

関数 / パラメータ	使用できる値	デフォルト値	説明
cert2user			クライアント証明書を使って認証済みユーザを決定する
userdb			証明書を取得するユーザデータベースを指定する
makefrombasic			ユーザとの証明書のマッピングを行うように関数に指示する
require	0 または 1	1	ユーザ名に証明書をマッピングできない場合、戻り値を決定する。require=0 の場合、関数は REQ_NOACTION を返し、要求処理は引き続き実行される。require が 0 でない場合、関数は REQ_ABORTED を返してプロトコルの状態を 403 FORBIDDEN に設定する
method			この関数が適用される HTTP メソッドに、ワイルドカードのパターンを指定する。method がいない場合、関数は任意のメソッドに適用される

表 2-17 obj.conf PathCheck 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
check-acl			アクセス制御リストを確認して承認する
acl			アクセス制御リストの名前
shexp			(省略可) ACL を適用するパスを指定するワイルドカードのパターン
bong-file			(省略可) ACL がアクセスを拒否した場合に送信するファイルのパス名
deny-existence			リソースが検索されなかったことを示す
path			(省略可) ファイルシステムパスを非表示にする場合のワイルドカードのパターン。パスが一致しない場合、関数は何もせずに REQ_NOACTION を返す。パスがない場合、一致しているとみなされる
bong-file			(省略可) 「not found」の代わりに送信される、応答メッセージが格納されたファイルを指定する。ファイルシステムの絶対パスを指定する
find-index			ディレクトリが要求された場合、デフォルトのファイルを検出する
index-names	コンマ区切りリスト		検索するインデックスファイル名のリスト。スペースは、ファイル名の一部になっている場合にだけ使用する。コンマの前後にはスペースを使用しない。ファイルシステムで大文字と小文字が区別される場合は、このリストでも大文字と小文字を区別する
find-links			特定のファイルシステムにリンクしているディレクトリへのアクセスを拒否する
disable	h、s、o		無効にするリンクの文字列は次のとおり <ul style="list-style-type: none"> • h はハードリンク • s はソフトリンク • o は、ユーザがリンク先を所有している場合だけ、ユーザのホームディレクトリのシンボリックリンクが使用できる
dir			確認を開始するディレクトリ。絶対パスを指定する場合は、そのパスやサブディレクトリへの要求のシンボリックリンクを確認する。パスの一部を指定する場合は、そのパスを含む要求のシンボリックリンクを確認する

表 2-17 obj.conf PathCheck 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
find-pathinfo			ファイル名に続く外部パスで PATH_INFO CGI 環境変数を検出する
find-pathinfo-forward	値は無視される		(省略可) サーバ関数 find-pathinfo がデフォルトでパスの最後から逆方向に検索するのとは対照的に、ntrans-base の直後のパスで PATHINFO を順方向に検索する
get-client-cert			SSL3 セッションから、承認済みクライアント証明書を取得する
dorequest	0 または 1	0、 dorequest がない場合	実際に証明書を取得しようとするのか、確認用にテストするだけなのかを設定する <ul style="list-style-type: none"> • 1 に設定されていると、サーバにまだクライアント証明書がない場合、SSL3 ハンドシェイクを再実行してクライアント証明書を取得するよう関数に指示する。通常はクライアントがユーザにダイアログボックスを提示して、クライアント証明書を選択してもらう • 0 の場合、サーバにはまだクライアント証明書がなければ、SSL3 ハンドシェイクを再実行しないよう関数に指示する
require	0 または 1	1、require がない場合	クライアント証明書が取得できない場合、HTTP 要求を終了するかどうかを設定する <ul style="list-style-type: none"> • 1 の場合、dorequest の処理後にクライアント証明書がなければ、HTTP 要求を終了させるよう関数に指示する。この場合、HTTP の状態は PROTOCOL_FORBIDDEN に設定され、関数は REQ_ABORTED を返す • 0 の場合、dorequest の処理後にクライアント証明書がなければ、REQ_NOACTION を返すよう関数に指示する
method			(省略可) この関数が適用される HTTP メソッドに、ワイルドカードのパターンを指定する。method がない場合、関数はすべての要求に適用される
load-config			要求されたパスのファイルから、外部構成情報を検索して読み込む
file		.nsconfig	(省略可) 要求されたリソースに適用されるアクセス規則を含む、ダイナミック構成ファイルの名前

表 2-17 obj.conf PathCheck 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
disable-types			(省略可) magnus-internal/cgi のような、ベースディレクトリを無効にするワイルドカードのパターンのタイプを指定する。これらのタイプに一致するリソースの要求は中止される
descend			(省略可) このパラメータがある場合は、このディレクトリのサブディレクトリで、サーバがダイナミック構成ファイルを検索するように指定する
basedir	パス	要求されたリソースの URL を物理パス名に変換した結果	(省略可) ベースディレクトリを指定する。これは、要求が load-config 関数を呼び出すときの最上位のディレクトリで、サーバはこのディレクトリから、構成ファイルの検索を開始する
nt-uri-clean			安全でないパス名を持つ要求へのアクセスを拒否し、「not found」と表示する
ntcgicheck			指定した拡張子を持つ CGI ファイルを検索する
extension			交換用ファイルの拡張子
require-auth			未承認ユーザやグループのアクセスを拒否する
path			(省略可) この関数が実行されるローカルファイルシステムのワイルドカードパス。パスがない場合、関数はすべてのパスに適用される
auth-type	basic	basic	使用される HTTP 認証のタイプで、AuthTrans での前回の authorization 関数の auth-type と一致する必要がある
realm			ブラウザに送信される文字列で、ユーザ名とパスワードが要求されるセキュリティ保護された領域 (またはレルム) を表す
auth-user			(省略可) アクセスが許可されているユーザのワイルドカードリストを指定する。パラメータがない場合は、authorization 関数で認証された任意のユーザがアクセスできる
auth-group			(省略可) アクセスが許可されているグループのワイルドカードリストを指定する
set-virtual-index			ディレクトリの仮想インデックスを指定する
virtual-index			ユーザが入力する URI のインデックスとなるコンテンツジェネレータの URI

表 2-17 obj.conf PathCheck 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
from	コンマ区切りリスト		(省略可) この virtual-index を適用する URI のリスト。 from が指定されていない場合、virtual-index が常に適用される
ssl-check			秘密鍵のサイズを確認する
secret-keysize			(省略可) 秘密鍵で要求される最小ビット数
bong-file			(省略可) 制限と一致しない場合に提供されるファイル名 (URI ではない)
ssl-logout			サーバの SSL セッションのキャッシュにある、現在の SSL セッションを無効にする
unix-uri-clean			安全でないパス名を持つ要求のアクセスを拒否し、「not found」と表示する

ObjectType 関数

表 2-18 obj.conf ObjectType 関数

関数 / パラメータ	使用できる値	デフォルト値	説明
force-type			応答の content-type ヘッダーを特定のタイプに設定する
type			(省略可) マッチング要求に割り当てられたタイプ (content-type ヘッダー)
enc			(省略可) マッチング要求に割り当てられたエンコーディング (content-encoding ヘッダー)
lang			(省略可) マッチング要求に割り当てられた言語 (content-language ヘッダー)
charset			(省略可) rq->srvhdrs の magnus-charset パラメータに設定された文字。ブラウザが Accept-charset ヘッダーを送信した場合や User-agent が mozilla/1.1 以降の場合は、content-type に「; charset=charset」を追加する。ここでは charset は rq->srvhdrs の magnus-charset パラメータの値

表 2-18 obj.conf ObjectType 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
set-default-type			
			クライアントに送信する応答に charset、content-encoding、および content-language をデフォルトで定義できる
enc			(省略可) マッチング要求に割り当てられたエンコーディング (content-encoding ヘッダー)
lang			(省略可) マッチング要求に割り当てられた言語 (content-language ヘッダー)
charset			(省略可) rq->srvhdrs の magnus-charset パラメータに設定された文字。ブラウザが Accept-charset ヘッダーを送信した場合や User-agent が mozilla/1.1 以降の場合は、content-type に「; charset=charset」を追加する。ここで、charset は rq->srvhdrs の magnus-charset パラメータの値
shtml-hacktype			
			.htm および .html ファイルをサーバが解析する html コマンドで構文解析するように要求する
exec-hack	値は無視される		(UNIX の場合のみ、省略可) このパラメータがある場合に、実行ビットが有効な時にのみ content-type を変更するようにサーバに指示する
type-by-exp			
			応答の content-type ヘッダーを、要求されたパスに基づいて設定する
exp			この関数を適用するパスのワイルドカードのパターン
type			(省略可) マッチング要求に割り当てられたタイプ (content-typee ヘッダー)
enc			(省略可) マッチング要求に割り当てられたエンコーディング (content-encoding ヘッダー)
lang			(省略可) マッチング要求に割り当てられた言語 (content-language ヘッダー)
charset			(省略可) rq->srvhdrs の magnus-charset パラメータに設定された文字。ブラウザが Accept-charset ヘッダーを送信した場合や User-agent が mozilla/1.1 以降の場合は、content-type に「; charset=charset」を追加する。ここで、charset は rq->srvhdrs の magnus-charset パラメータの値

表 2-18 obj.conf ObjectType 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
<code>type-by-extension</code>			応答の <code>content-type</code> ヘッダーを、ファイル拡張子と MIME タイプのデータベースに基づいて設定する

Service 関数

表 2-19 obj.conf Service 関数

関数 / パラメータ	使用できる値	デフォルト値	説明
共通 Service パラメータ			リストの最初にある 7 つのパラメータは、すべての Service 関数に共通する。重複をしないよう一度だけ指定する
<code>type</code>			(省略可) この関数を実行する MIME タイプのワイルドカードのパターンを指定する。 <code>magnus-internal/*</code> MIME タイプは、Service 関数を選択して実行するためにだけ使用する
<code>method</code>			(省略可) この関数を実行する HTTP メソッドのワイルドカードのパターンを指定する。共通の HTTP メソッドは、GET、HEAD、および POST である
<code>query</code>			(省略可) この関数を実行する照会文字列のワイルドカードのパターンを指定する
<code>UseOutputStreamSize</code>	バイト数	8192	(省略可) NSAPI の <code>net_read</code> 関数および <code>netbuf_grab</code> 関数のデフォルトの出力ストリームのバッファサイズを決定する
<code>flushTimer</code>	ミリ秒数	3000	(省略可) バッファリングが有効な書き込み操作が可能な最大時間を決定する。このあと行われる書き込み操作までの間隔がアプリケーションの <code>flushTimer</code> の値よりも大きい場合、その後バッファは使用できなくなる
<code>ChunkedRequestBufferSize</code>	バイト数	8192	(省略可) 分割されていない要求データに対するデフォルトのバッファサイズを決定する

表 2-19 obj.conf Service 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
ChunkedRequestTimeout	秒数	60	(省略可) 分割されていない要求データに対するデフォルトのタイムアウトを秒で決定する
add-footer			ファイル名または URL で指定したフッターを HTML ファイルに追加する
file			(省略可) フッターを含むファイルのパス名。file または uri のいずれかを指定する
uri			(省略可) フッターを含むリソースを指定する URL。file または uri のいずれかを指定する
NSIntAbsFilePath	yes または no		(省略可) file パラメータを指定すると、NSIntAbsFilePath パラメータはファイル名を絶対パスで指定する (yes) のか相対パスで指定する (no) のかを決定する
add-header			ファイル名または URL で指定したヘッダーを HTML ファイルの先頭に追加する
file			(省略可) ヘッダーを含むファイルのパス名。file または uri のいずれかを指定する
uri			(省略可) ヘッダーを含むリソースを指定する URL。file または uri のいずれかを指定する
NSIntAbsFilePath	yes または no		(省略可) file パラメータを指定すると、NSIntAbsFilePath パラメータはファイル名が絶対パスで指定する (yes) のか相対パスで指定する (no) のかを決定する
append-trailer			HTML ファイルの末尾にテキストを追加する

表 2-19 obj.conf Service 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
<code>trailer</code>			HTML ドキュメントに追加するテキスト。文字列は、送信する前に <code>util_uri_unescape</code> でエスケープを解除する。テキストには HTML タグを指定することができ、エスケープを解除して日付を挿入したあと、512 文字まで使用できる。文字列 <code>:LASTMOD:</code> はファイルを最後に変更した日付で置き換えられるが、これを使う場合は <code>timefmt</code> で時刻の書式も指定しておく必要がある
<code>timefmt</code>			(省略可) <code>:LASTMOD:</code> の時刻書式文字列。 <code>timefmt</code> がない場合、 <code>:LASTMOD:</code> を時刻で置き換えることはできない。書式の詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照
<code>imagemap</code>			サーバサイドのイメージマップを処理する
<code>index-common</code>			要求されたディレクトリに、ファイルとディレクトリの拡張リストを生成する
<code>header</code>			(省略可) ディレクトリの冒頭にあるパス (インデックスが生成されるディレクトリへの相対パス) およびファイル名 (HTML またはプレーンテキスト) で、ディレクトリの内容を表す
<code>readme</code>			(省略可) ディレクトリのリストに追加するパス (インデックスが生成されるディレクトリの相対パス) およびファイル名 (HTML またはプレーンテキスト)
<code>index-simple</code>			要求されたディレクトリに、ファイルとディレクトリのシンプルリストを生成する
<code>key-toosmall</code>			クライアントに、与えられた証明書の鍵のサイズが小さすぎて受理されないことを示す
<code>list-dir</code>			ディレクトリの内容を表示する。要求 <code>method</code> は INDEX である必要がある
<code>make-dir</code>			ディレクトリを作成する。要求 <code>method</code> は MKDIR である必要がある

表 2-19 obj.conf Service 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
query-handler			HTML ISINDEX タグを扱う
path			実行する CGI プログラムの絶対パスとファイル名
remove-dir			空のディレクトリを削除する。要求 method は RMDIR である必要がある
remove-file			ファイルを削除する。要求 method は DELETE である必要がある
rename-file			ファイル名を変更する。要求 method は MOVE である必要がある
send-cgi			環境変数を設定し、CGI プログラムを起動してクライアントに応答を送信する
user			(UNIX の場合のみ) CGI プログラムを実行するユーザ名
group			(UNIX の場合のみ) CGI プログラムを実行するグループ名
chroot			(UNIX の場合のみ) chroot を実行する前のディレクトリ。magnus.conf で定義した Chroot に関連している
dir			(UNIX の場合のみ) chroot の実行後、chdir を実行する前のディレクトリ
rlimit_as			(UNIX の場合のみ) CGI プログラムのバイト単位の最大アドレス空間。現在の制限値 (弱い制限値) および最大制限値 (強い制限値) をコンマで区切って指定できる。弱い制限値を最初に指定する必要がある。制限値が 1 つだけの場合は、どちらもこの値に設定される
rlimit_core			(UNIX の場合のみ) CGI プログラムのコアファイルの最大サイズ。値が 0 の場合、書き込みコアは無効になる。現在の制限値 (弱い制限値) および最大制限値 (強い制限値) をコンマで区切って指定できる。弱い制限値を最初に指定する必要がある。制限値が 1 つだけの場合は、どちらもこの値に設定される

表 2-19 obj.conf Service 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
<code>rlimit_nofile</code>			(UNIX の場合のみ) CGI プログラムのファイル記述子の最大数。現在の制限値 (弱い制限値) および最大制限値 (強い制限値) をコンマで区切って指定できる。弱い制限値を最初に指定する必要がある。制限値が 1 つだけの場合は、どちらもこの値に設定される
<code>nice</code>			(UNIX の場合のみ) インクリメントを適用して、サーバに対する CGI プログラムの優先度を決定する。通常、サーバは nice 値 0 で動作し、nice 値は、0 (CGI プログラムがサーバと同じ優先度で動作する) から 19 (CGI プログラムがサーバよりも低い優先度で動作する) までの間で増分する
<code>send-file</code>			クライアントにローカルファイルを送信する。この指令は、要求メソッドが、GET、HEAD、または POST であり、タイプが <code>magnus-internal/</code> で始まっていない場合に呼び出される
<code>nocache</code>		値は無視される	(省略可) サーバが統計ファイル要求への応答をキャッシュしないようにする。たとえば、特定のディレクトリにあるファイルがキャッシュされないように指定することができるので、頻繁に変更されるファイルがあるディレクトリでは有用である
<code>send-range</code>			クライアントに送信するファイルのサイズ (バイト数)
<code>send-shellcgi</code>			(Windows NT の場合のみ) 環境変数を設定し、シェル CGI プログラムを起動してクライアントに応答を送信する
<code>send-wincgi</code>			(Windows NT の場合のみ) 環境変数を設定し、WinCGI プログラムを起動してクライアントに応答を送信する

表 2-19 obj.conf Service 関数 (続き)

関数 / パラメータ	使用できる値	デフォルト値	説明
service-dump			収集されたパフォーマンスバケットのデータに基づいて、パフォーマンスレポートを作成する (71 ページの「bucket パラメータ」を参照)。mime.types ファイルには次の行を指定する必要がある。 type=perf exts=perf。レポートを表示するには、ブラウザで次のアドレスをクリックする。 http://server_id:port/.perf
type	perf		レポートの MIME タイプを指定する
shtml_send			HTML ドキュメントの埋め込みコマンドをスキャンして、構文解析を実行する。このコマンドではサーバから情報を取得したり、他のファイルの内容を取り込んだり、CGI プログラムを実行したりすることがある。shtml_send 関数は、Shtml プラグイン (UNIX では libShtml.so、Windows NT では libShtml.dll) が読み込まれている場合にだけ利用できる
ShtmlMaxDepth		10	入れ子が許可されているインクルードが可能な最大の深さ
addCgiInitVars	yes、no	no	(UNIX の場合のみ) パラメータが存在し、yes である場合、init-cgi SAF に定義されている環境変数を SHTML exec タグで実行された任意のコマンドの環境に追加する
stats-xml			パフォーマンスレポートを XML 形式で作成する。magnus.conf の stats-init 関数を使ってこの関数を初期化し、次に NameTrans 関数を使って stats-xml 関数への要求を出す必要がある。レポートの作成場所は次のとおり。 http://server_id:port/stats-xml/iwsstats.xml 。関連する DTD ファイルは次の場所。 http://server_id:port/stats-xml/iwsstats.dtd
upload-file			ファイルをアップロードして保存する。要求メソッドは PUT である必要がある

AddLog 関数

表 2-20 obj.conf AddLog 関数

関数 / パラメータ	使用できる値	デフォルト値	説明
common-log			要求に関する情報を共通のログ形式で記録する
name			(省略可) ログファイル名を指定する。ログファイル名は、 <code>init-clf Init</code> 関数のパラメータとして指定されている名前にする必要がある。名前が指定されていない場合は、ログのエントリは、すべてグローバルログファイルに記録される
iponly	値は無視される		(省略可) DNS 名を検索してログに記録するのではなく、リモートクライアントの IP アドレスをログに記録するようサーバに指示する。 <code>magnus.conf</code> ファイルで DNS がオフになっていると、パフォーマンスが向上する
flex-log			要求に関する情報を、フレキシブルで構成可能な形式で記録する
name			(省略可) ログファイル名を指定する。ログファイル名は、 <code>init-clf Init</code> 関数のパラメータとして指定されている名前にする必要がある。名前がない場合は、ログのエントリは、すべてグローバルログファイルに記録される
iponly	値は無視される		(省略可) DNS 名を検索してログに記録するのではなく、リモートクライアントの IP アドレスをログに記録するようサーバに指示する。 <code>magnus.conf</code> ファイルで DNS がオフになっていると、パフォーマンスが向上する
record-useragent			クライアントの IP アドレスとユーザーエージェントのヘッダーを記録する
name			(省略可) ログファイル名を指定する。ログファイル名は、 <code>init-clf Init</code> 関数のパラメータとして指定されている名前にする必要がある。名前が指定されていない場合は、ログのエントリは、すべてグローバルログファイルに記録される

Error 関数

表 2-21 obj.conf Error 関数

関数 / パラメータ	説明
send-error	特定の HTTP 応答状態の代わりに、HTML ファイルをクライアントに送信する
path	クライアントに送信する HTML ファイルを、ファイルシステムの絶対パスで指定する。ファイル名や実際のファイルタイプにかかわらず、ファイルは text/html 形式で送信される。ファイルを指定しない場合、サーバはデフォルトの簡単なエラーページを送信する
reason	(省略可) 理由を表す文字列 ("Unauthorized" または "Forbidden" など) のいずれかのテキスト。文字列中の大文字と小文字は、区別されない
code	(省略可) 401 や 407 など、HTTP の応答状態を表す 3 桁の数字。HTTP の仕様に基づく任意の HTTP の応答状態コードまたは理由を表す文字列
qos-error	エラーを引き起こしたサービス制限の品質、および QOS 統計の値を示すエラーページを返す
code	(省略可) 401 や 407 など、HTTP の応答状態を表す 3 桁の数字。HTTP の仕様に基づく任意の HTTP の応答状態コードまたは理由を表す文字列。推奨値は 503

password.conf

目的

デフォルトでは、Web サーバが起動する前に管理者は鍵データベースのパスワードを入力するように求められます。Web サーバが自動的に再起動できるようにするには、password.conf ファイルにパスワードを保存しておく必要があります。このファイルと鍵データベースに障害が発生しないようにするため、使用するシステムが十分にセキュリティ保護されていることを確認してください。

場所

`server_root/https-admserv/config`

`server_root/https-server_id/config`

デフォルトでは、このファイルは存在しません。必要に応じて作成する必要があります。

構文

`PKCS#11_module_name:password`

サーバに同梱されている内部 PKCS#11 ソフトウェア暗号化モジュールを使う場合、次のように入力します。

```
Communicator_Cert_DB:password
```

別の PKCS#11 モジュールを使ってハードウェアを暗号化したり高速化する場合は、このモジュールの名前とパスワードを次のように指定する必要があります。

```
internal:password
```

関連項目

iPlanet Web Server の『管理者ガイド』

rules.properties

目的

iPlanet Web Server 4.x との下位互換用です。代わりに web.xml を使ってサーブレットを構成することをお勧めします。

サーブレットの仮想パス変換を定義します。iPlanet Web Server 6.0 では、デフォルトの仮想サーバだけをサポートします。

場所

```
server_root/https-admserv/config
```

```
server_root/https-admserv/conf_bk
```

```
server_root/https-server_id/config
```

```
server_root/https-server_id/conf_bk
```

構文

```
virtual_path=servlet_name
```

URL `http://server_id/virtual_path` は、`servlets.properties` ファイルで `servlet_name` で指定されたサーブレットを呼び出します。

`virtual_path` は正規表現になります。たとえば次の式では、`/my/xxx.foo` のような要求が URL にあった場合は必ず `wasp` サーブレットを実行するようサーバに指示します。

```
@.*[.]foo$=wasp
```

関連項目

iPlanet Web Server の『サーブレットに関するプログラマーズガイド』

web.xml ファイル

付録 A 「iWS 4.x と 6.0 の設定の変更点」

次の Web サイトの Servlet 2.2 API 仕様書

<http://java.sun.com/products/servlet/index.html>

server.xml

目的

待機ソケットと仮想サーバを定義します。

場所

`server_root/https-admserv/config`

`server_root/https-admserv/conf_bk`

`server_root/https-server_id/config`

`server_root/https-server_id/conf_bk`

構文

ファイルには次のような基本的な XML 構文があり、要素は入れ子になっています。

```
<ELEMENT attribute="value" attribute="value" ... >
  <SUBELEMENT attribute="value" attribute="value" ... />
</ELEMENT>
```

表 2-22 では要素 (ELEMENT) を太字にして属性 (attribute) と区別します。またデフォルト値は指定された属性が存在しないことを前提にしています。

関連項目

iPlanet Web Server の『NSAPI プログラマーズガイド』の第 8 章

表 2-22 server.xml

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
SERVER	VARS 、 LS 、 MIME 、 ACLFILE 、 VSCLASS 、 QOSPARAMS		サーバを定義する。サブ要素は、表示された順に定義する必要がある

表 2-22 server.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
gosactive	yes、no、on、off、 1、0	no	サービス機能の品質を有効にする。 サーバエンティティに制限を設定 したり、帯域幅や接続に関するサー バ統計を表示できる
gosmetricsinterval	秒数	30	(省略可)トラフィックの測定間隔
gosrecomputeinterval	ミリ秒数	100	(省略可)すべてのサーバエンティ ティの帯域幅を再計算する期間
legacyls			古いバージョンであるアプリケー ション(4.x)の待機ソケットのid 属性。LSには CONNECTIONGROUPが1つだけ含 まれ、そのdefaultvsであるVS を1つだけ構成する必要がある。 古いバージョンのアプリケーション は、すべてこの仮想サーバ上で実 行する必要がある
VARS	(サブ要素なし。共通 に定義されている変 数は、docroot、 adminusers、 webapps_file、 webapps_enable、 accesslog、user、 group、chroot、 dir、 およびnice)		server.xmlで値を設定して obj.confで参照できる変数を定 義する。デフォルトで定義されて いる変数はないが、最もよく定義さ れる変数は、obj.confの document-root関数で使用され るdocrootである。詳細は、iPlanet Web Serverの『NSAPIプログラ マーズガイド』の第8章を参照
LS	(なし)		待機ソケットを定義する
id			待機ソケットの内部名。vs要素で 使用され、仮想サーバに接続する待 機ソケットを定義する
ip	ドット付きペアまた はIPv6表記のIPアド レス。INADDR_ANY では0.0.0.0で表記 することもできる		待機ソケットのIPアドレス。複数 のCONNECTIONGROUPが待機ソ ケットに構成されている場合は、 0.0.0.0用の待機ソケットを構成 する必要がある

表 2-22 server.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
port	1 - 65535		待機ソケットを作成するポート番号。UNIX では、ポート 1 - 1024 で待機するソケットを作成するにはスーパーユーザ特権が必要である。ポート 443 で待機する SSL 待機ソケットの構成を推奨する。異なる IP アドレスが同じポートを使うことはできない
security	on、off、yes、no、 1、0	no	(省略可) 待機ソケットが SSL を実行するかどうかを決定する。SSL2 または SSL3 のオンとオフを切り換え、CONNECTIONGROUP オブジェクトにある SSLPARAMS オブジェクトを使って符号化方式を設定できる
acceptorthreads	1 - 1024	1	(省略可) 待機ソケットの受け入れスレッド数
family	inet、inet6、nca	inet	(省略可) ソケットファミリのタイプ。IPv6 待機ソケットで使う値は inet6 。inet6 の値を使う場合は、IPv4 のアドレスにはログファイルの接頭辞 ::ffff: が付く。Solaris Network Cache and Accelerator (NCA) を使う nca を指定する
blocking	on、off、yes、no、 1、0	no	(省略可) 待機ソケットと適用されたソケットをブロックモードにするかどうかを決定する。ブロックモードを使うとベンチマークの得点が高くなる可能性がある。本稼動の環境では no に設定する
CONNECTIONGROUP	SSLPARAMS		MIME タイプを定義する
id			接続グループの内部名。vs 要素で使用され、仮想サーバが使用する接続を定義する

表 2-22 server.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
matchingip	ドット付きペアまたは IPv6 表記の IP アドレス、または default の値。INADDR_ANY では 0.0.0.0 で表記することはできない		関連付けられた仮想サーバで使う IP アドレス。含まれている LS に ip=0.0.0.0 が設定されていない場合は default にする必要がある 含まれている LS に ip=0.0.0.0 が設定されていれば、特定の IP アドレスか default のいずれかを指定できる。この場合、default とは、ほかの LS または CONNECTIONGROUP 要素で指定されていない任意の IP アドレスを意味する
defaultvs			この接続グループに対するデフォルトの仮想サーバの id 属性
servername			クライアントに送信する URL のホスト名の部分に入力する内容をサーバに指示する。コロンとポート番号を追加した場合、サーバがクライアントに送信する URL にそのポートが使われる
SSLPARAMS	(なし)		接続グループの SSL パラメータを定義する。SSLPARAMS 要素が内部で必要となる。security 属性が on に設定されている待機ソケットに指定される CONNECTIONGROUP 要素は内部でだけ使用できる
servercertnickname			証明書データベースにあるサーバ証明書のニックネームまたは PKCS#11 のトークン。証明書の name 形式に <i>tokenname:nickname</i> 。tokenname: を指定する場合、この属性の名前の部分を省略することができる
ssl2	on、off、yes、no、1、0	no	(省略可) SSL2 を有効にするかどうかを決定する
ssl2ciphers	rc4、rc4export、rc2、rc2export、idea、des、desede3	なし	(省略可) スペースで区切られた SSL2 符号化方式リストで、接頭辞が + であれば有効、- であれば無効。たとえば +rc4 のようになる

表 2-22 server.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
ssl3	on、off、yes、no、 1、0	yes	(省略可) SSL3 を有効にするかどうかを決定する
ssl3tlsciphers	rsa_rc4_128_md5、 rsa3des_sha、 rsa_des_sha、 rsa_rc4_40_md5、 rsa_rc2_40_md5、 rsa_null_md5、 rsa_des_56_sha、 rsa_rc4_56_sha	なし	(省略可) スペースで区切られた SSL3 および TLS 符号化方式リストで、接頭辞が + であれば有効、- であれば無効。たとえば +rsa_des_sha のようになる
tls	on、off、yes、no、 1、0	no	(省略可) TLS を有効にするかどうかを決定する
tlscrollback	on、off、yes、no、 1、0	on	(省略可) TLS ロールバックを有効にするかどうかを決定する
clientauth	on、off、yes、no、 1、0	no	(省略可) ACL ベースのアクセス制御とは別に、要求ごとに SSL3 クライアント認証を実行するかどうかを決定する
MIME	(なし)		MIME タイプを定義する
id			MIME タイプのリストの内部名。vs 要素で使用され、仮想サーバが使用する MIME タイプを定義する
file			MIME タイプのファイル名。このファイルの形式については、iPlanet Web Server の『NSAPI プログラマーズガイド』の「付録 B」を参照
ACLFILE	(なし)		1 つまたは複数の ACL ファイルへの参照
id			ACL ファイルリストの内部名。vs 要素で使用され、仮想サーバが使用する ACL ファイルを定義する

表 2-22 server.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
file			<p>スペースで区切られた ACL ファイルリスト。それぞれの ACL ファイルには一意のファイル名が必要。ACL ファイル形式については、iPlanet Web Server の『管理者ガイド』を参照</p> <p>デフォルトの ACL ファイル名は generated.https-server_id.ac l で、ファイルは server_root/server_id/httpacl ディレクトリにある。このファイルを使うには、server.xml で参照する必要がある</p>
VSCLASS	VARs、VS、 QOSPARAMS		<p>仮想サーバクラスを定義する。サブ要素は、表示された順に定義する必要がある</p>
id			<p>仮想サーバクラスの ID。特定の仮想サーバクラスを検索できる一意の ID</p>
objectfile			<p>仮想サーバのこのクラスの obj.conf ファイル名。VS 要素で無効にすることはできない</p>
rootobject		default	<p>(省略可) obj.conf ファイルから読み込まれたオブジェクトのどれをデフォルトにするのかをサーバに指示する。デフォルトのオブジェクトには、仮想サーバに対する名前変換(NameTrans) 指令がすべて記述されているものとする。サーバマネージャは、default で指定されたオブジェクトがデフォルトであるとみなす</p>

表 2-22 server.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
acceptlanguage	on、off	off	(省略可) on の場合、サーバで Accept-Language ヘッダーを構文解析して、クライアントが使用する言語に基づいて適切な言語バージョンを送信する。サーバが複数の言語をサポートしている場合にだけこの値を on に設定しておく必要がある。VS 要素で無効にすることができる
VS	VARs, QOSPARAMS、 USERDB		仮想サーバを定義する。サブ要素は、表示された順に定義する必要がある
id	文字で始める必要がある		仮想サーバの ID。特定の仮想サーバを検索できる一意の ID である。obj.conf ファイルでは変数 \$id として扱うこともできる
connections			(省略可) 仮想サーバが使う接続を指定する CONNECTIONGROUP id のスペースで区切られたリスト。CONNECTIONGROUP の defaultvs でない VS についてだけ必要
urlhosts			URLHost 要求ヘッダーにおいて、現在の仮想サーバを選択するために使用できる値を、スペースで区切って表示したリスト。同じ CONNECTIONGROUP に設定された VS ごとに、そのグループに一意の値が必要
mime			仮想サーバで使用する MIME 要素の id
state	on、off、disable	on	(省略可) VS がアクティブ (on) であるかアクティブでない (off、disable) かを決定する。アクティブでない場合、VS は要求を実行しない VS が disable の場合は、グローバルサーバ管理者だけが on に変更できる

表 2-22 server.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
aclids			(省略可) スペースで区切られた、ACLFILE 要素の 1 つまたは複数の id 属性。仮想サーバで使う ACL ファイルを指定する
errorlog			(省略可) 仮想サーバ専用のエラーメッセージのログファイルを指定する
acceptlanguage	on、off	off	(省略可) on の場合、サーバで Accept-Language ヘッダーを構文解析して、クライアントが使用する言語に基づいて適切な言語バージョンを送信する。サーバが複数の言語をサポートしている場合にだけ、この値を on に設定しておく必要がある
QOSPARAMS	(なし)		SERVER、VSCLASS、または VS のサービスの品質パラメータを定義する
maxbps	毎秒のバイト数		(省略可) SERVER、VSCLASS、または VS の帯域幅制限の最大値
enforcebandwidth	yes、no、on、off、 1、0	no	帯域幅制限を有効にするかどうかを指定する
maxconn	接続数		(省略可) SERVER、VSCLASS、または VS の同時接続の最大数
enforceconnections	yes、no、on、off、 1、0	no	接続の制限を有効にするかどうかを指定する
USERDB	(なし)		仮想サーバで使うユーザデータベースを定義する
id			仮想サーバの ACL ファイルにあるユーザデータベース名
database			dbswitch.conf ファイルのユーザデータベース名
basedn			(省略可) dbswitch.conf ファイルのベース DN 検索を優先する

servers.lst

表 2-22 server.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
certmaps			(省略可) certmap.conf で定義された LDAP エントリマッピングで使う証明書を指定する。この属性がない場合は、すべてのマッピングを使う。certmap.conf のマッピングに基づくすべての検索は、VS の最終ベース DN に関連する

servers.lst

目的

Administration Server で管理される iPlanet Server を一覧表示します。このファイルは変更しないでください。

場所

`server_root/https-admserv/config`

構文

`protocol:server`

表 2-23 servers.lst

指令	使用できる値	デフォルト値	説明
<code>protocol</code>	http、https	https	通信プロトコル
<code>server</code>	iPlanet Server の名前	Web Server, Enterprise Edition	Administration Server で管理する iPlanet Server

servlets.properties

目的

iPlanet Web Server 4.x との下位互換用です。サーブレットを構成する代わりに `web-apps.xml` を使用することをお勧めします。

グローバルサーブレット設定とシステムのサーブレットリストを定義します。iPlanet Web Server 6.0 では、デフォルトの仮想サーバだけをサポートします。

場所

`server_root/https-admserv/config`

`server_root/https-admserv/conf_bk`

`server_root/https-server_id/config`

`server_root/https-server_id/conf_bk`

構文

`servlets.property=value`

`servlet.servlet_name.property=value`

表 2-24 で説明されている `servlet` プロパティは、名前付きサーブレットだけにあてはまります。表 2-25 で説明されている `servlets` プロパティは、すべてのサーブレットにあてはまります。

関連項目

iPlanet Web Server の『サーブレットに関するプログラマーズガイド』

`server.xml` ファイルおよび `web-apps.xml` ファイル

付録 A 「iWS 4.x と 6.0 の設定の変更点」

次の Web サイトの Servlet 2.2 API 仕様書

<http://java.sun.com/products/servlet/index.html>

表 2-24 `servlets.properties` の個別 (サーブレット) プロパティ

プロパティ	使用できる値	デフォルト値	説明
<code>code</code>	クラスまたはクラスファイル名		サーブレットのクラス名またはクラスファイル名。 .class 拡張子は省略可
<code>context</code>	コンテキスト名		サーブレットが属するコンテキスト。コンテキストの設定は <code>contexts.properties</code> ファイルで変更する

表 2-24 servlets.properties の個別 (サブレット) プロパティ

プロパティ	使用できる値	デフォルト値	説明
classpath	URL またはパスで、スラッシュ (/) だけを使う		クラスのあるディレクトリへの URL やパス。または (URL 以外の) ディレクトリまたは jar ファイルのリストへの URL やパス。たとえば、CLASSPATH 環境変数
initArgs	コンマで区切られた <i>name=value</i> の組み合わせ		Servlet API 呼び出しを使ってサブレットでアクセスできる <i>name=value</i> の組み合わせリスト
startup	true、false	true	Web サーバの起動時にサブレットを自動的に起動するかどうかを決定する

表 2-25 servlets.properties の一般的な (サブレット) プロパティ

プロパティ	使用できる値	デフォルト値	説明
config.docRoot	スラッシュのあるパス	Web サーバのドキュメントルート	すべてのサブレットのドキュメントルート。docRoot が指定されていない場合、Web サーバのドキュメントルートが使用される
config.realPathFromRequest	true、false	false	true の場合は、サブレットの docRoot に基づいて getRealPath を計算する。false の場合は、標準の NSAPI 手順で試行する
config.respondCookieVersion	cookie のバージョン番号	0	特定の cookie バージョンに回答するかどうかをサーバに指示する
config.sessionExpireOnClose	true、false	false	ユーザがブラウザを終了すると、セッションの cookie も有効期限が切れるようにサーバに指示する

表 2-25 `servlets.properties` の一般的な (サブレット) プロパティ

プロパティ	使用できる値	デフォルト値	説明
<code>sessionmgr</code>	セッションマネージャのオブジェクト	<code>com.iplanet.server.http.session.IWSSessionManager</code> (すべてを 1 行に記述し、ダッシュは使用しない)	サブレットのセッションマネージャ名。 <code>MMapSessionManager</code> などのセッションマネージャは、サーバ内でインスタンス化できるのは 1 度だけ
<code>config.reloadInterval</code>	秒数	5	サーバが、JSP およびサブレットファイルが修正されているかどうかを確認する時間間隔
<code>config.bufferSize</code>	バイト数	4096	HTTP 出力ストリームの初期バッファサイズ
<code>startup</code>	<code>true</code> 、 <code>false</code>	<code>true</code>	Web サーバの起動時に、すべてのサブレットを自動的に起動するかどうかを決定する。 <code>servlet.startup</code> プロパティを代わりに使うことを推奨する

web.xml

目的

サブレット、URL マッピング、セキュリティ制限などの Web アプリケーションを定義します。Web アプリケーションにはそれぞれ独自の `web.xml` ファイルがあります。

場所

アプリケーション固有の場所をユーザが定義します。

構文

次の Web サイトで .DTD ファイルを参照してください。

`http://java.sun.com/j2ee/dtds/web-app_2_2.dtd`

または

`server_root/bin/https/dtds/web-app_2_2.dtd`

関連項目

iPlanet Web Server の『サブレットに関するプログラマーズガイド』

次の Web サイトの Servlet 2.2 API 仕様書

<http://java.sun.com/products/servlet/index.html>

次の Web サイトの JSP 1.1 仕様書

<http://java.sun.com/products/jsp/download.html>

ファイルの要素や属性のリストは、Servlet 2.2 API の仕様で示されているので、ここでは扱いません。

web-apps.xml

目的

仮想サーバが管理する Web アプリケーションを定義します。仮想サーバには、それぞれ独自の web-apps.xml ファイルを設定することができます。

場所

デフォルトファイルの場所だけを示します。

`server_root/https-server_id/config`

構文

ほとんどのファイルには、次のような基本的な XML 構文があり、要素が入れ子になっています。

```
<ELEMENT attribute="value" attribute="value" ... >
  <SUBELEMENT attribute="value" attribute="value" ... />
</ELEMENT>
```

表 2-26 では、要素を太字にして属性と区別します。また、デフォルト値は、指定された属性が存在しないことを前提にしています。

関連項目

iPlanet Web Server の『サーブレットに関するプログラマーズガイド』

次の Web サイトの Servlet 2.2 API 仕様書

<http://java.sun.com/products/servlet/index.html>

次の Web サイトの JSP 1.1 仕様書

<http://java.sun.com/products/jsp/download.html>

表 2-26 web-apps.xml

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
auth-native	(なし)		(省略可) 承認とロールのマッピング用に、固有のネイティブユーザ / グループのデータベースを設定する。この要素が指定されていない場合は、ネイティブのデフォルト認証データベースで承認する
authdb			ネイティブ認証データベース
class-loader	(なし)		Web アプリケーションのクラスローダー
classpath			クラスローダーが使用するクラスパス
delegate	true、false	false	仮想サーバやシステムのクラスローダーを最初に呼び出してクラスを読み込むように指定する
reload-interval	秒数	30	Web アプリケーションが修正されているかどうかを、サーバが確認する時間間隔
description	(なし)		パラメータの説明。 init-param 要素で使用する。 iPlanet Web Server はこの要素を無視する

表 2-26 web-apps.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
filter 、 filter-mapping			Servlet 2.3 の Filter API を実装する。web-app 要素の中で使用される iPlanet Web Server 6.0 は web.xml ファイルで Servlet 2.2 API だけをサポートするが、この機能は web-apps.xml ファイルで利用できる filter および filter-mapping については、そのファイルの位置を除き Servlet 2.3 で説明されている。詳細は次のサイトを参照 http://java.sun.com/products/servlet/index.html
form-login-session	session-manager		仮想サーバのすべての Web アプリケーションを対象に、形式ベースのシングルサインオン認証を設定する。この要素がない場合は、仮想サーバのデフォルトレベルのセッションマネージャが使用される
cookie-name	cookie 名	iwsformloginid	セッション ID を追跡する cookie 名
timeOut	秒数	600 (10 分)	セッションのタイムアウト

表 2-26 web-apps.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
init-param	param-name 、 param-value 、 description		含まれている要素の初期化パラメータを指定する。 init-paramの属性は、含まれている要素が参照するオブジェクトによって異なる たとえば、含まれている要素が session-manager で、セッションマネージャが IWSessionManager の場合、init-param の属性は IWSessionManager の初期化パラメータである
jsp-servlet	init-param		JSP のコンパイル動作を設定する。初期化パラメータ use-precompiled を true に設定して、仮想サーバの JSP がすべてコンパイル済みであることを iPlanet Web Server に指示する。jsp-servlet 初期化パラメータの詳細は、iPlanet Web Server の『サーブレットに関するプログラマーズガイド』を参照
enable	true、false	true	JSP を有効にする
interval	秒数	0	フラッシュ間隔
param-name	(なし)		パラメータ名。init-param 要素の中で使用される
param-value	(なし)		パラメータ値。init-param 要素の中で使用される
parameter-encoding	(なし)		次の形式のパラメータに対する復号化の方法を、Web サーバに伝える

表 2-26 web-apps.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
enc	none、auto、または utf8 や Shift_JIS などの特定の <i>encoding</i>	auto	<ul style="list-style-type: none"> • <i>encoding</i>: 特定のエンコーディングを使用する • none: システムのデフォルトのエンコーディングを使用する • auto: 各エンコーディング方式を次の順序で試みる。1) charset、2) parameter Encoding 属性、3) form-hint-field で定義された隠し形式フィールド。これ以外の場合は none と同じ
form-hint-field		j_encoding	エンコーディングを指定する形式の隠しフィールド名
response-buffer	(なし)		HTTP サーブレットの応答バッファの初期サイズとデフォルトのサイズを設定する
flush-timeout	秒数	0	最後のフラッシュ以降、指定した秒数が経過したらストリームにデータをフラッシュさせる。0 (デフォルト) または負の数に設定されている場合、バッファに空き容量がなくなるまで、出力ストリームはフラッシュしない
size	バイト数	8192	バッファサイズ
response-cookie	(なし)		特定の cookie バージョンで応答するようサーバに指示する
version	cookie のバージョン番号	0	cookie のバージョン
role-mapping	(なし)		web.xml の role-name 値を LDAP のユーザまたはグループにマップする
map-to	user、group	group	web.xml の role-name 値を LDAP のユーザまたはグループにマップするかどうかを指定する

表 2-26 web-apps.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
session-cookie	(なし)		セッション cookie のパラメータを指定する
domain	ドメイン名	(なし)	この属性がある場合、値のタグが cookie に付けられる。デフォルト値はない
is-secure	true、false	false	true に設定すると、要求がセキュリティ保護された接続で届いた場合、サーバはセッション cookie において secure 属性を送信する。デフォルトは false
session-manager	init-param		Web アプリケーションのセッションマネージャ。各セッションマネージャの初期化パラメータは iPlanet Web Server の『サーバーレットに関するプログラマーズガイド』を参照
class			セッションマネージャのクラス
session-tracking	(なし)		セッション追跡のメソッドを決定する
use-cookies	true、false	true	true の場合はセッション追跡の cookie を使う
use-url-rewriting	true、false	true	true の場合はセッション追跡の URL 書き換えを使う
tempdir	(なし)		Web アプリケーションで使用される一時ディレクトリ
dir			一時ディレクトリ

表 2-26 web-apps.xml (続き)

要素 / 属性	使用できるサブ要素 または値	デフォルト値	説明
vs	auth-native、 class-loader、 form-login- session、 jsp-servlet、 parameter- encoding、 response-buffer 、 response-cookie 、role-mapping、 session-manager 、 session-trackin g、 session-cookie、 tempdir、 web-app		web-apps.xml ファイルの最上位要素。web-app 以外のサブ要素をすべての Web アプリケーションのデフォルトに設定する
web-app	auth-native、 class-loader、 filter、 filter-mapping、 jsp-servlet、 parameter- encoding、 response-buffer 、 response-cookie 、role-mapping、 session-manager 、 session-trackin g、 session-cookie、 tempdir		Web アプリケーション。Web アプリケーションは WAR ファイルに格納されていて、サーブレット、JSP、HTML ページ、クラスファイル、および他のアプリケーションリソースを含むことができる web-app 要素のサブ要素は、その Web アプリケーションに含まれる vs 要素の対応するサブ要素よりも優先する
dir			Web アプリケーションの内容が保存されているディレクトリ
uri			クライアントが Web アプリケーションへのアクセスに使う URI。この URI では正規表現も使用できる

サーバによる HTML タグの構文解析

HTML ファイルには、サーバで実行されるタグを含めることができます。iPlanet Web Server 6.0 は標準のサーバサイドのタグをサポートしていますが、さらにサーブレットを使用することで、ユーザ独自のサーバサイドのタグを定義することもできます。

この章では次の項目について説明します。

- サーバサイドの HTML コマンドの使用方法
- サーブレットを埋め込む方法
- サーバが構文解析するカスタム HTML タグの定義方法

注 サーバがサーバサイドのタグを解析するのは、サーバ側で構文解析を有効にしている場合だけです。クラスマネージャインタフェースの「Content Management」タブにある「Parse HTML」ページで、サーバサイドのタグの構文解析を有効または無効にします（クラスマネージャを表示するには、サーバマネージャの「Virtual Server Class」タブで「Manage Classes」ページを選択し、リストからクラスを選択して「Manage」ボタンをクリックします）。

構文解析を有効にしたら、magnus.conf ファイルに次の指令が追加されていることを確認します（ネイティブスレッドがオフになっていることを確認してください）。

```
Init funcs="shtml_init,shtml_send"  
shlib="install_dir/bin/https/bin/Shtml.dll" NativeThreads="no"  
fn="load-modules"
```

iPlanet Web Server 6.0 に、NativeThread="no" を設定する必要があります。また、これらの関数は shtml.dll (UNIX では libShtml.so) から作成されます。このファイルは Windows NT では `install_dir/bin/https/bin` に、UNIX では `install_dir/bin/https/lib` にあります。

また、次の指令が `obj.conf` ファイルに追加されていることも確認してください。

```
<Object name="default">
...
...
Service fn="shtml_send" type="magnus-internal/parsed-html"
method="(GET|HEAD)"
...
</Object>
```

サーバサイドの HTML コマンドの使用法

この節では、サーバが構文解析するタグを HTML ファイルに指定するための HTML コマンドを説明します。これらのコマンドは HTML ファイルに埋め込まれ、組み込み SAF `parse-html` で処理されます。

サーバはそれぞれのコマンドを、コマンドとその属性で決定したデータと置き換えます。コマンドの形式は次のとおりです。

```
<!--#command attribute1 attribute2 <Body>... -->
```

それぞれの *attribute* の形式は、次のような名前 (*name*) と値 (*value*) の組み合わせです。

```
name="value"
```

コマンド (*command*) と属性名 (*attribute*) は小文字で表します。

コマンドは HTML のコメントの中に「隠されて」いるので、サーバが構文解析を実行しない場合は無視されてしまいます。サーバサイドの標準的なコマンドは次のとおりです。

- `config`
- `include`
- `echo`
- `fsize`
- `flastmod`

- `exec`

config

`config` コマンドは、ほかのコマンドの書式を初期化します。

- `errmsg` 属性は、ファイルの構文解析中にエラーが発生したとき、クライアントに送信されるメッセージを定義する。この処理で発生したエラーも、エラーログファイルに記録される
- `timefmt` 属性は、`flastmod` コマンドの日付書式を指定する。この属性では、`util_strftime` 関数と同じ書式指定文字が使用される。デフォルトの時刻書式は次のとおり：`"%A, %d-%b-%y %T"`

時刻書式については、iPlanet Web Server の『NSAPI プログラマーズガイド』の付録「時刻書式」を参照

- `sizefmt` 属性では、`filesize` コマンドのファイルサイズ書式を指定する。次のいずれかの値を取る
 - `bytes`、12,345,678 の書式で全体のファイルサイズを示す
 - `abbrev`、ファイルサイズを K バイトまたは M バイトで示す (デフォルト)

例：

```
<!--#config timefmt="%r %a %b %e, %Y" sizefmt="abbrev"-->
```

この例では、データ書式を 08:23:15 AM Wed Apr 15, 1996 のように設定し、ファイルサイズの書式を、ファイルで使用される文字を使用し、K バイトまたは M バイトの単位で表示するように設定します。

include

`include` コマンドは、構文解析済みのファイルに、別のファイルの内容を挿入します。別の解析済みファイルをインクルードし、さらに別のファイルにインクルードすることによって、入れ子にすることもできます。ただし、サーバが構文解析されたファイルのあるディレクトリへのアクセス制御を行なっている場合、クライアントが構文解析済みのドキュメントを要求するには、入れ子になっているファイルへのアクセス権も必要です。

iPlanet Web Server 6.0 では、`virtual` 属性の付いた `include` コマンドを使って、CGI プログラムファイルを組み込むことができます。CGI プログラムを実行するには、`exec` コマンドも使う必要があります。

- `virtual` 属性はサーバ上のファイルの URI

- `file` 属性は、現在のディレクトリを起点とする相対パス名。このパス名には `../` などの要素を含めることはできない。また、絶対パスを指定することもできない

例：

```
<!--#include file="bottle.gif"-->
```

echo

`echo` コマンドで、環境変数の値を挿入します。`var` 属性では、挿入する環境変数を指定します。変数がない場合は、「(none)」が挿入されます。環境変数のリストについては、113 ページの「サーバサイド HTML コマンドの環境変数」を参照してください。

例：

```
<!--#echo var="DATE_GMT"-->
```

filesize

`filesize` コマンドでファイルのサイズを送信します。属性は `include` コマンドの属性 (`virtual` および `file`) と同じです。ファイルサイズ書式は、`config` コマンドの `sizefmt` 属性で決定します。

例：

```
<!--#filesize file="bottle.gif"-->
```

flastmod

`flastmod` コマンドで、ファイルを最後に変更した日付をプリントします。属性は `include` コマンドの属性 (`virtual` および `file`) と同じです。時刻書式は、`config` コマンドの `timefmt` 属性で決定します。

例：

```
<!--#flastmod file="bottle.gif"-->
```

exec

`exec` コマンドで、シェルコマンドまたは CGI プログラムを実行します。

- `cmd` は UNIX 専用の属性で、`/bin/sh` を使ってコマンドを実行する。コマンドには任意の特別環境変数を指定することができる

- `cgi` 属性で CGI プログラムを実行し、構文解析済みファイルに実行結果を出力することができる

例：

```
<!--#exec cgi="workit.pl"-->
```

サーバサイド HTML コマンドの環境変数

CGI で使用する通常の変数以外に、構文解析済みコマンドには次の変数を指定できます。

- `DOCUMENT_NAME`
構文解析済みファイルのファイル名
- `DOCUMENT_URI`
構文解析済みファイルへの仮想パス (たとえば `/shtml/test.shtml`)
- `QUERY_STRING_UNESCAPED`
「`¥`」でエスケープされたシェル独自の特殊文字を含む、クライアントが送信した検索照会に対して、その結果からエスケープ文字を消したもの
- `DATE_LOCAL`
現在の日付とローカル時刻
- `DATE_GMT`
グリニッジ標準時で表した現在の日付と時刻
- `LAST_MODIFIED`
ファイルが最後に変更された日付

サーブレットを埋め込む方法

iPlanet Web Server 6.0 は、Java Web Server と同様に、<SERVLET> タグをサポートしています。このタグによって、SHTML ファイルにサーブレット出力を埋め込むことができます。この操作を有効にするために、設定を変更する必要はありません。SSI とサーブレットの両方が有効な場合、<SERVLET> が有効になります。

<SERVLET> タグ構文は、他の SSI コマンドのタグとは少し異なります。次のように、<APPLET> タグ構文に似ています。

```
<servlet name=name code=code codebase=path iParam1=v1 iParam2=v2>
<param name=param1 value=v3>
<param name=param2 value=v4>
.
.
.</servlet>
```

サーブレットが Web アプリケーションの一部である場合、code パラメータが必要です。他のパラメータは無視されます。code パラメータには、次の値が必要になります。

- Web アプリケーションの web.xml ファイルで定義された url-pattern 要素の値。web.xml については、次の Web サイトの「Servlet 2.2 API specification」を参照。
<http://java.sun.com/products/servlet/index.html>
- Web アプリケーションの web-apps.xml ファイルで定義された uri 属性の値。web-apps.xml については、iPlanet Web Server の『サーブレットに関するプログラマーズガイド』を参照。

たとえば、次の構文を SHTML ファイルに記述した場合、

```
<servlet name=pparams code="/PrintApp/PrintParams">
</servlet>
```

web-apps.xml ファイルには、次の構文を追加する必要があります。

```
<web-app uri="/PrintApp"
dir="/iws60/https-server.iplanet.com/acme.com/webapps/PrintApp"/>
```

また、web.xml ファイルには、次の構文も追加する必要があります。

```
<servlet>
  <servlet-name> pparams </servlet-name>
  <servlet-class> PrintPackage.PrintParams </servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name> pparams </servlet-name>
  <url-pattern> /PrintParams </url-pattern>
</servlet-mapping>
```

さらに、web.xml ファイルに、任意のサーブレット初期化パラメータも追加する必要があります。

古いバージョンの (iPlanet Web Server 4.x) サーブレットに対しては、code パラメータでサーブレットの .class ファイルを指定するので、このパラメータが必要になります。codebase パラメータは、サーブレットが `servlets.properties` ファイルに定義されておらず、.class ファイルが `<SERVLET>` タグを含む HTML ファイルと同じディレクトリにない場合に必要になります。古いバージョンのサーブレットはデフォルトの仮想サーバに設定する必要があり、web.xml ファイルは必要ありません。

サーブレットの作成については、iPlanet Web Server の『サーブレットに関するプログラマーズガイド』を参照してください。

サーバが構文解析するカスタム HTML タグの定義方法

iPlanet Web Server 6.0 では、ユーザが独自にサーバサイドのタグを定義できます。たとえば、タグ HELLO を定義して「Hello World!」とプリントする関数を呼び出すことができます。このためには、hello.shtml ファイルに、次のコードを指定します。

```
<html>
<head>
<title>shtml custom tag example</title>
</head>
<body>
<!--#HELLO-->
</body>
</html>
```

ブラウザでこのコードが表示されると、HELLO タグが検出されるたびに関数が呼び出されます。

実装方法

カスタムタグのサーバによる構文解析を定義する手順は次のとおりです。

1. タグを実装する関数を定義する

タグ実行関数を定義します。タグの読み込みと読み込み解除時、およびページの読み込みと読み込み解除時に呼び出す他の関数も定義します。

2. 初期化関数を記述して新しいタグを登録する

shtml_add_tag 関数を使って、タグを登録する初期化関数を記述します。

3. サーバに新しいタグを読み込む

タグを実装する関数を定義する

C 言語で書かれた関数を定義し、タグに実装するため、NSAPI を使用します。

- ヘッダー `shtml_public.h` をインクルードする。これは `install_dir/plugins/include/shtml` ディレクトリにある
- `shtml` 共有ライブラリにリンクする。Windows NT では、`shtml.dll` は `install_dir/bin/https/bin` にある。UNIX プラットフォームでは、`libshtml.so` または `.sl` は `install_dir/bin/https/lib` にある

`ShtmlTagExecuteFunc` は、実際のタグハンドラです。通常の NSAPI `pblock`、`Session`、および `Request` 変数で呼び出されます。さらに、そのタグのタグ読み込み関数やページ読み込み関数 (定義されている場合) を実行した結果から作成された `TagUserData` が渡されます。

タグ実行関数のシグニチャは次のとおりです。

```
typedef int (*ShtmlTagExecuteFunc)(pblock*, Session*, Request*,
TagUserData, TagUserData);
```

タグ実行関数の本文を記述して出力結果を生成し、`.shtml` ページのタグを置き換えます。`net_write` 関数を使って通常の NSAPI の方法で行います。指定したバッファから、指定したソケットに指定したバイト数が書き込まれます。

NSAPI プラグインの記述については、iPlanet Web Server の『NSAPI プログラマーズガイド』の第 4 章を参照してください。

`net_write` および他の NSAPI 関数については、iPlanet Web Server の『NSAPI プログラマーズガイド』の第 5 章を参照してください。

タグの実行関数は、サーバが `obj.conf` で次の命令に進むべきかどうかを示す `int` を返す必要があります。次のいずれかになります。

- `REQ_PROCEED` - 正常に実行された
- `REQ_NOACTION` - 変化なし
- `REQ_ABORTED` - エラーが発生した
- `REQ_EXIT` - 接続が切れた

タグには、次の関数を定義する必要があります。

- `ShtmlTagInstanceLoad`

タグを含むページが構文解析されるときに呼び出されます。ブラウザのキャッシュでページを取り出す場合は呼び出されません。基本的にはコンストラクタとして機能し、実行関数が呼び出されると、結果が必ずキャッシュされて `ShtmlTagExecuteFunc` に渡されます。

- ShtmlTagInstanceUnload
基本的にはデストラクタで、ShtmlTagInstanceLoad 関数で作成されたものをすべて消去します。ShtmlTagInstanceLoad 関数から返された結果を渡します。
- ShtmlTagPageLoadFunc
タグを含むページが実行されると、ページがブラウザのキャッシュにあるかどうかに関わりなく呼び出されます。これにより、同じページに同じタグがある場合に情報の持続性が保たれます。
- ShtmlTagPageUnLoadFn
タグを含むページが実行された後にこの関数が呼び出されます。ShtmlTagPageLoadFunc で行われた任意の割り当てを解除してから、ShtmlTagPageLoadFunc から戻された結果が渡されます。

この関数のシグニチャは次のとおりです。

```
#define TagUserData void*
typedef TagUserData (*ShtmlTagInstanceLoad) (
    const char* tag, pblock*, const char*, size_t);
typedef void (*ShtmlTagInstanceUnload) (TagUserData);
typedef int (*ShtmlTagExecuteFunc) (
    pblock*, Session*, Request*, TagUserData, TagUserData);
typedef TagUserData (*ShtmlTagPageLoadFunc) (
    pblock* pb, Session*, Request*);
typedef void (*ShtmlTagPageUnLoadFunc) (TagUserData);
```

HELLO タグを実装するコードは次のとおりです。

```
/*
 * mytag.c: #HELLO SSI 呼び出しを実装する NSAPI 関数
 *
 */

#include "nsapi.h"
#include "shtml/shtml_public.h"

/* 関数 : mytag_con
 *
 * 説明 : ShtmlTagInstanceLoad 関数
 */
#ifdef __cplusplus
```

```
extern "C"
#endif
TagUserData
mytag_con(const char* tag, pblock* pb, const char* c1, size_t t1)
{
    return NULL;
}

/* 関数 : mytag_des
 *
 * 説明 : ShtmlTagInstanceUnload
 */
#ifdef __cplusplus
extern "C"
#endif
void
mytag_des(TagUserData v1)
{
}

/* 関数 : mytag_load
 *
 * 説明 : ShtmlTagPageLoadFunc
 */
#ifdef __cplusplus
extern "C"
#endif
TagUserData
mytag_load(pblock *pb, Session *sn, Request *rq)
{
    return NULL;
}

/* 関数 : mytag_unload
 *
 * 説明 : ShtmlTagPageUnloadFunc
 */
#
#ifdef __cplusplus
extern "C"
#endif
void
mytag_unload(TagUserData v2)
{
}
```

```
/* 関数 : mytag
 *
 * 説明 : ShtmlTagExecuteFunc
 */
#ifdef __cplusplus
extern "C"
#endif
int
mytag(pblock* pb, Session* sn, Request* rq, TagUserData t1,
TagUserData t2)
{
    char* buf;
    int length;
    char* client;
    buf = (char *) MALLOC(100*sizeof(char));
    length = util_sprintf(buf, "<h1>Hello World! </h1>", client);
    if (net_write(sn->csd, buf, length) == IO_ERROR)
    {
        FREE(buf);
        return REQ_ABORTED;
    }
    FREE(buf);
    return REQ_PROCEED;
}

/* 関数 : mytag_init
 *
 * 説明 : 初期化関数 shtml_add_tag() を呼び出し、
 * 新しいタグを読み込む
 */

#
#ifdef __cplusplus
extern "C"
#endif
int
mytag_init(pblock* pb, Session* sn, Request* rq)
{
    int retVal = 0;
    // 注意 : shtml_add_tag() 関数には以下のすべての引数が必要
    retVal = shtml_add_tag("HELLO", mytag_con, mytag_des, mytag,
mytag_load, mytag_unload);

    return retVal;
}
/* end mytag.c */
```


初期化関数を記述して新しいタグを登録する

新しいタグを定義する共有ライブラリの初期化関数に、`shtml_add_tag` を使ってタグを登録します。シグニチャは次のとおりです。

```
NSAPI_PUBLIC int shtml_add_tag (
    const char* tag,
    ShtmlTagInstanceLoad ctor,
    ShtmlTagInstanceUnload dtor,
    ShtmlTagExecuteFunc execFn,
    ShtmlTagPageLoadFunc pageLoadFn,
    ShtmlTagPageUnLoadFunc pageUnLoadFn);
```

`tag` および `execFn` 以外は、いずれの引数も `NULL` を返すことができます。

サーバに新しいタグを読み込む

新しいタグを定義する共有ライブラリを作成したら、NSAPI プラグインの通常の方法で iPlanet Web Server にライブラリを読み込みます。つまり、次の指令を構成ファイル `magnus.conf` に追加します。

1. `fn` パラメータが `load-modules` で `shlib` パラメータが読み込む共有ライブラリである `Init` 指令を追加します。たとえば、共有オブジェクト `install_dir/hello.so` にタグをコンパイルすると、次のようになります。


```
Init funcs="mytag,mytag_init" shlib="install_dir/hello.so"
fn="load-modules"
```
2. `fn` パラメータが共有ライブラリの初期化関数であり、`shtml_add_tag` を使ってタグがライブラリに登録される場合、別の `Init` 指令を追加します。たとえば次のようになります。


```
Init fn="mytag_init"
```

サーバが構文解析するカスタム HTML タグの定義方法

iWS 4.x と 6.0 の設定の変更点

この章では、iPlanet Web Server の構成ファイルについて、前のバージョン以降の主な変更点を簡単に説明します。ここでは、次の 4.x のファイルを取り上げます。

- magnus.conf
- obj.conf
- contexts.properties
- rules.properties
- servlets.properties

magnus.conf

magnus.conf ファイルでは、他の構成ファイルへの指令が廃止されました。また、他の構成ファイルからの指令を受けるようになり、まったく新しい指令も追加されました。表 A-1 に変更点を示します。

表 A-1 magnus.conf の変更点

4.x の指令	6.0 の指令	コメント
AccelFileCache	(なし)	ファイルキャッシュアクセラレータが不要になったために廃止
AcceptLanguage	(なし)	server.xml の VSCLASS 要素と VS 要素を参照
AcceptTimeout	(なし)	IOTimeout にほぼ等しい
ACLFile	(なし)	server.xml の ACLFILE 要素を参照

表 A-1 magnus.conf の変更点 (続き)

4.x の指令	6.0 の指令	コメント
Address	(なし)	server.xml の LS 要素と VS 要素を参照
AdminLanguage	AdminLanguage	(変更なし)
AsyncDNS	AsyncDNS	(変更なし)
BlockingListenSockets	(なし)	仮想サーバの実装により廃止
CGIExpirationTimeout	CGIExpirationTimeout	(変更なし)
CGIStubIdleTimeout	CGIStubIdleTimeout	(変更なし)
CGIWaitPid	CGIWaitPid	(変更なし)
ChildRestartCallback	ChildRestartCallback	(変更なし)
Chroot	Chroot	(変更なし)
ChunkedRequestBufferSize	ChunkedRequestBufferSize	(変更なし)
ChunkedRequestTimeout	ChunkedRequestTimeout	(変更なし)
Ciphers	(なし)	server.xml の SSLPARAMS 要素を参照
ClientLanguage	ClientLanguage	(変更なし)
Concurrency	(なし)	仮想サーバの実装により廃止
(なし)	ConnQueueSize	(新規)
DaemonStats	(なし)	新しいパフォーマンス統計システムにより廃止
DefaultCharSet	DefaultCharSet	(変更なし)
DefaultLanguage	DefaultLanguage	(変更なし)
DNS	DNS	(変更なし)
ErrorLog	ErrorLog	(変更なし)
ErrorLogDateFormat	ErrorLogDateFormat	(変更なし)
ExtraPath	ExtraPath	(変更なし)
HeaderBufferSize	HeaderBufferSize	(変更なし)
HTTPVersion	HTTPVersion	(変更なし)
(なし)	IOTimeout	(新規)

表 A-1 magnus.conf の変更点 (続き)

4.x の指令	6.0 の指令	コメント
(なし)	obj.conf の Init 関数	廃止された関数の cache-init および load-types を除くすべての関数は存続 (load-types については server.xml ファイルの MIME 要素を参照)
KeepAliveThreads	KeepAliveThreads	(変更なし)
KeepAliveTimeout	KeepAliveTimeout	(変更なし)
KernelThreads	KernelThreads	(変更なし)
ListenQ	ListenQ	(変更なし)
LoadObjects	(なし)	server.xml の VSCLASS 要素を参照
LogFlushInterval	LogFlushInterval	(変更なし)
LogVerbose	LogVerbose	(変更なし)
MaxCGIStubs	MaxCGIStubs	(変更なし)
MaxKeepAliveConnections	MaxKeepAliveConnections	(変更なし)
MaxProcs	MaxProcs	(変更なし)
MaxRqHeaders	MaxRqHeaders	(変更なし)
MaxThreads	(なし)	新しいスレッド処理システムにより廃止
MinCGIStubs	MinCGIStubs	(変更なし)
MinProcs	(なし)	新しいスレッド処理システムにより廃止
MinThreads	(なし)	新しいスレッド処理システムにより廃止
MtaHost	MtaHost	(変更なし)
NativePoolMaxThreads	NativePoolMaxThreads	(変更なし)
NativePoolMinThreads	NativePoolMinThreads	(変更なし)
NativePoolQueueSize	NativePoolQueueSize	(変更なし)
NativePoolStackSize	NativePoolStackSize	(変更なし)
NetSiteRoot	NetSiteRoot	(変更なし)
PidLog	PidLog	(変更なし)

表 A-1 magnus.conf の変更点 (続き)

4.x の指令	6.0 の指令	コメント
Port	(なし)	server.xml の LS 要素を参照
PostThreadsEarly	PostThreadsEarly	(変更なし)
RcvBufSize	RcvBufSize	(変更なし)
RootObject	(なし)	server.xml の VSCLASS 要素を参照
RqThrottle	RqThrottle	(変更なし)
RqThrottleMinPerSocket	(なし)	server.xml の LS 要素を参照
(なし)	RqThrottleMin	(新規)
Security	Security	(変更なし)
ServerCert	(なし)	server.xml の SSLPARAMS 要素を参照
(なし)	ServerConfigurationFile	(新規)
ServerID	ServerID	(変更なし)
ServerKey	(なし)	server.xml の SSLPARAMS 要素を参照
ServerName	(なし)	server.xml の VS 要素を参照
#ServerRoot	#ServerRoot	(変更なし)
SndBufSize	SndBufSize	(変更なし)
SSL2	(なし)	server.xml の SSLPARAMS 要素を参照
SSL3	(なし)	server.xml の SSLPARAMS 要素を参照
SSL3Ciphers	(なし)	server.xml の SSLPARAMS 要素を参照
SSL3SessionTimeout	SSL3SessionTimeout	(変更なし)
SSLCacheEntries	SSLCacheEntries	(変更なし)
SSLClientAuth	(なし)	server.xml の SSLPARAMS 要素を参照
SSLClientAuthDataLimit	SSLClientAuthDataLimit	(変更なし)
SSLClientAuthTimeout	SSLClientAuthTimeout	(変更なし)
SSLSessionTimeout	SSLSessionTimeout	(変更なし)

表 A-1 magnus.conf の変更点 (続き)

4.x の指令	6.0 の指令	コメント
StackSize	StackSize	(変更なし)
StrictHttpHeaders	StrictHttpHeaders	(変更なし)
TerminateTimeout	TerminateTimeout	(変更なし)
ThreadIncrement	ThreadIncrement	(変更なし)
Umask	Umask	(変更なし)
UseNativePoll	UseNativePoll	(変更なし)
UseOutputStreamSize	UseOutputStreamSize	(変更なし)
User	User	(変更なし)
VirtualServerFile	(なし)	仮想サーバの実装により廃止
WincgiTimeout	WincgiTimeout	(変更なし)

obj.conf

obj.conf では magnus.conf への Init 指令が廃止されました。また、新しい指令とパラメータが追加されました。表 A-2 に変更点の概要を示します。ここでは、新しい指令と変更された指令だけを示します。

表 A-2 obj.conf changes

4.x の指令	6.0 の指令	コメント
Init 関数	(なし)	廃止された関数の cache-init および load-types を除くすべての関数は magnus.conf に移動 (load-types については server.xml ファイルの MIME 要素を参照)
(なし)	AuthTrans fn=qos-handler	(新規)
Service fn=parse-html	Service fn=shtml_send	
Service fn=send-cgi	Service fn=send-cgi	新規パラメータを追加
(なし)	Service fn=stats-xml	(新規)
(なし)	Error fn=qos-error	(新規)

contexts.properties

contexts.properties ファイルは、このバージョンでもデフォルトの仮想サーバでサポートされています。他の仮想サーバでは、同じ関数のほとんどが web-apps.xml ファイルに格納されています。

contexts.properties 関数には server.xml ファイルに格納されているものもあります。server.xml ファイルの詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照してください。

表 A-3 に、servlets.properties ファイルと web-apps.xml ファイルにある同等の関数を示します。

表 A-3 contexts.properties と web-apps.xml の対応

contexts.properties プロパティ	web-apps.xml の要素または属性	コメント
sessionmgr	session-manager 要素	
sessionmgr.initArgs	session-manager 要素の init-param サブ要素	
initArgs	要素をサポートする init-param サブ要素	
respondCookieVersion	response-cookie 要素の version 属性	
tempDir	tempdir 要素	
reloadInterval	class-loader 要素の reload-interval 属性	
bufferSize	response-buffer 要素の size 属性	
docRoot	(なし)	仮想サーバごとに server.xml ファイルで指定される
inputStreamLengthCheck	(なし)	Web アプリケーションのサポート により廃止
outputStreamFlushTimer	response-buffer 要素の flush-timeout 属性	
uri	web-app 要素の uri 属性	
authdb	auth-native 要素の authdb 属 性	
classpath	class-loader 要素の classpath 属性	

表 A-3 contexts.properties と web-apps.xml の対応 (続き)

contexts.properties プロパティ	web-apps.xml の要素または属性	コメント
singleClassLoader	(なし)	それぞれの Web アプリケーションにデフォルトで単独のクラスローダーがあるために廃止
serverName	(なし)	仮想サーバごとに server.xml ファイルで指定される
contentTypeIgnoreFromSSI	(なし)	Web アプリケーションのサポートにより廃止
parameterEncoding	parameter-encoding 要素	
isModifiedCheckAggressive	(なし)	Web アプリケーションのサポートにより廃止

rules.properties

rules.properties ファイルの関数は、このバージョンでは web.xml ファイルの servlet-mapping 要素で扱います。詳細は、次に示す Web サイトの「Servlet 2.2 API specification」を参照してください。

<http://java.sun.com/products/servlet/index.html>

servlets.properties

servlets.properties ファイルはこのバージョンでもデフォルトの仮想サーバでサポートされています。他の仮想サーバでは、同じ関数のほとんどが web-apps.xml ファイルに格納されています。

servlets.properties 関数には server.xml ファイルに格納されているものもあります。server.xml ファイルの詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照してください。

servlets.properties 関数には web.xml ファイルに格納されているものもあります。詳細は、次に示す Web サイトの「Servlet 2.2 API specification」を参照してください。

<http://java.sun.com/products/servlet/index.html>

表 A-4 および表 A-5 に、servlets.properties ファイルと web-apps.xml ファイルにある同等の関数を示します。

表 A-4 各(サーブレット)プロパティでの servlets.properties と web-apps.xml の対応

servlets.properties プロパティ	web-apps.xml の要素または属性	コメント
code	(なし)	web.xml ファイルの servlet-class 要素で指定される
context	(なし)	Web アプリケーションでサポートしているために廃止
classpath	class-loader 要素の classpath 属性	
initArgs	(なし)	<SERVLET> タグを使用。サーブレット固有の初期化パラメータを取る
startup	(なし)	web.xml ファイルの load-on-startup 要素で指定される

表 A-5 (サーブレット)プロパティ全般での servlets.properties と web-apps.xml の対応

servlets.properties プロパティ	web-apps.xml の要素または属性	コメント
config.docRoot	(なし)	仮想サーバごとに server.xml ファイルで指定される
config.realPathFromRequest	(なし)	Servlet 2.2 API では推奨されない
config.respondCookieVersion	response-cookie 要素の version 属性	
config.sessionExpireOnClose	(なし)	この方法によるセッションの有効期限の追跡は、必要なくなった
sessionmgr	session-manager 要素	
config.reloadInterval	class-loader 要素の reload-interval 属性	
config.bufferSize	response-buffer 要素の size 属性	
startup	(なし)	web.xml ファイルの load-on-startup 要素で指定される。サーブレットのグローバルな起動関数はない

A

- abbrev、sizefmt 属性の値, 111
- AccelFileCache 指令, 123
- AcceptLanguage 指令, 123
- acceptlanguage 属性, 96, 97
- acceptorthreads 属性, 92
- AcceptTimeout 指令, 123
- Access Control Programmer's Guide, 26
- ACLCacheLifetime 指令, 60
- ACLFile 指令, 123
- ACLFILE 要素, 94
- ACLGroupCacheSize 指令, 60
- aclids 属性, 97
- ACLUserCacheSize 指令, 60
- acl パラメータ, 76
- ACL ファイル
 - 編集する, 26
- addCgiInitVars パラメータ, 86
- add-footer 関数, 82
- add-header 関数, 82
- AddLog 関数, 87
- AddLog 指令, 71
- Address 指令, 124
- AdminLanguage 指令, 60, 124
- adminport 属性, 36
- API
 - CGI, 12
 - iPlanet Web Server 4.1, 10
 - iPlanet Web Server 4.1 の変更内容, 29
 - Java サブレットと JSP, 21
 - NSAPI, 24
 - アクセス制御, 25
 - サーバによる HTML タグの構文解析, 11
 - 要約, 28
- API 参照
 - JSP, 23
- append-trailer 関数, 82
- assign-name 関数, 73
- AsyncDNS 指令, 60, 124
- authdb 属性, 103
- authdb プロパティ, 39, 128
- auth-group パラメータ, 78
- auth-native 属性, 103
- AuthTrans 関数, 72
- AuthTrans 指令, 71
- auth-type パラメータ, 72, 78
- auth-user パラメータ, 78

B

backups.conf, 32
basedir パラメータ, 78
basedn 属性, 97
basic-auth 関数, 72
basic-ncsa 関数, 72
binddn プロパティ, 42
bindpw プロパティ, 42
BlockingListenSockets 指令, 124
blocking 属性, 92
bong-file パラメータ, 76, 79
bucket パラメータ, 71
buffer-size パラメータ, 56
bufferSize プロパティ, 38, 128
bytes、sizefmt 属性の値, 111

C

cache-bucket 要素, 47
CacheFileContent パラメータ, 69
cache-size パラメータ, 55
cert2user 関数, 75
certmap.conf, 34
certmaps 属性, 98
CGI, 12
 CGI ディレクトリを指定する, 12
 CGI プログラムをサーバに追加, 19
 Web サイト, 21
 参照先, 21
 ファイルの拡張子を指定する, 13
 変数, 19
 有効にする, 12
CGIExpirationTimeout 指令, 60, 124
Cgistub, 14
CGIStubIdleTimeout 指令, 60, 124
cgistub-path パラメータ, 57
CGIWaitPid 指令, 60, 124
charset パラメータ, 79, 80

check-acl 関数, 76
ChildRestartCallback 指令, 60, 124
Chroot 指令, 124
chroot パラメータ, 84
ChunkedRequestBufferSize 指令, 61, 124
ChunkedRequestBufferSize パラメータ, 81
ChunkedRequestTimeout 指令, 61, 124
ChunkedRequestTimeout パラメータ, 82
cindex-init 関数, 54
Ciphers 指令, 124
cjava.properties, 35
class-loader 要素, 103
classpath
 JVM, 53
classpath 属性, 103
classpath プロパティ, 39, 100, 128, 130
class 属性, 107
clientauth 属性, 94
ClientLanguage 指令, 61, 124
cluster.xml, 36
CLUSTER 要素, 36
CmapLdapAttr プロパティ, 35
code パラメータ, 88
code プロパティ, 99, 130
Command 指令, 41
Common Gateway Interface の Web サイト, 21
common-log 関数, 87
Concurrency 指令, 124
ConfFile 指令, 68
config
 サーバサイドの HTML コマンド, 111
config.bufferSize プロパティ, 101, 130
config.docRoot プロパティ, 100, 130
config.realPathFromRequest プロパティ, 100, 130
config.reloadInterval プロパティ, 101, 130
config.respondCookieVersion プロパティ, 100, 130
config.sessionExpireOnClose プロパティ, 100, 130
config コマンドの errmsg 属性, 111
config コマンドの sizefmt 属性, 111

config ディレクトリ, 10
場所, 10
CONNECTIONGROUP 要素, 92
connection-queue-bucket 要素, 45
connection-queue 属性, 46, 48
connection-queue 要素, 45
connections 属性, 96
ConnQueueSize 指令, 61, 124
contentTypeIgnoreFromSSI プロパティ, 39, 129
contexts.properties, 37
変更点, 128
context プロパティ, 99, 130
cookie-name 属性, 104
CopyFiles パラメータ, 69
count200 属性, 49
count2xx 属性, 49
count302 属性, 49
count304 属性, 49
count3xx 属性, 49
count400 属性, 49
count401 属性, 49
count403 属性, 49
count404 属性, 50
count4xx 属性, 49
count503 属性, 50
count5xx 属性, 49
countAsyncAddrLookups 属性, 47
countAsyncLookupsInProgress 属性, 47
countAsyncNameLookups 属性, 47
countBytesReceived 属性, 49
countBytesTransmitted 属性, 49
countCacheEntries 属性, 46
countCacheHits 属性, 47
countCacheMisses 属性, 47
countCalls 属性, 50
countConfigurations 属性, 45
countConnections 属性, 47
countContentHits 属性, 48

countContentMisses 属性, 48
countEntries 属性, 47
countFlushes 属性, 47
countHits 属性, 47, 48
countInfoHits 属性, 48
countInfoMisses 属性, 48
countMisses 属性, 48
countOpenConnections 属性, 49
countOpenEntries 属性, 47
countOther 属性, 49
countOverflows 属性, 46
countQueued 属性, 46
countRequests 属性, 49, 50
countThreadsIdle 属性, 46
countThreads 属性, 46
countTotalConnections 属性, 46
countTotalQueued 属性, 46
cron.conf, 41

D

DaemonStats 指令, 124
database 属性, 97
DATE_GMT
サーバサイドの HTML コマンドの変数, 113
DATE_LOCAL
サーバサイドの HTML コマンドの変数, 113
Days 指令, 41
dbm パラメータ, 72
dbswitch.conf, 42
dcsuffix プロパティ, 43
DefaultCharSet 指令, 61, 124
DefaultLanguage 指令, 61, 124
defaultvs 属性, 93
define-perf-bucket 関数, 55
delegate 属性, 103
deny-existence 関数, 76
descend パラメータ, 78

description 属性, 45
description パラメータ, 55
description 要素, 103
digestauth プロパティ, 43
Dir 指令, 68
dir 属性, 107, 108
dir パラメータ, 74, 76, 84
disable-types パラメータ, 78
disable パラメータ, 58, 59, 76
DNComps プロパティ, 34
dns-bucket 要素, 46
dns-cache-init 関数, 55
DNS 指令, 61, 124
docRoot プロパティ, 38, 128
DOCUMENT_NAME
 サーバサイドの HTML コマンドの変数, 113
DOCUMENT_URI
 サーバサイドの HTML コマンドの変数, 113
document-root 関数, 73
domain 属性, 107
dorequest パラメータ, 77
dyngroups プロパティ, 42

E

echo
 サーバサイドの HTML コマンド, 112
enabled 属性, 44
enable 属性, 105
enc 属性, 106
enc パラメータ, 79, 80
enforcebandwidth 属性, 97
enforceconnections 属性, 97
ErrorLogDateFormat 指令, 61, 124
ErrorLog 指令, 61, 124
errorlog 属性, 97
Error 関数, 88
Error 指令, 71

escape パラメータ, 74
exec
 サーバサイドの HTML コマンド, 11, 112
exec-hack パラメータ, 80
exec コマンドの cgi 属性, 113
exec コマンドの cmd 属性, 112
expire パラメータ, 55
exp パラメータ, 80
extension パラメータ, 78
ExtraPath 指令, 61, 124

F

family 属性, 92
FileCacheEnable パラメータ, 69
file 属性, 94, 95
file パラメータ, 77, 82
FilterComps プロパティ, 35
filter-mapping 要素, 104
filter 要素, 104
find-index 関数, 76
find-links 関数, 76
find-pathinfo-forward パラメータ, 73, 74, 77
find-pathinfo 関数, 77
flagAsyncEnabled 属性, 47
flagCacheEnabled 属性, 46
flagEnabled 属性, 47
flagProfilingEnabled 属性, 44
flagVirtualServerOverflow 属性, 44
flastmod
 timefmt 属性に影響される, 111
 サーバサイドの HTML コマンド, 112
flex-init 関数, 55
flex-log 関数, 87
flex-rotate-init 関数, 56
flush-timeout 属性, 106
flushTimer パラメータ, 81
force-type 関数, 79

format パラメータ, 55, 56
form-hint-field 属性, 106
form-login-session 要素, 104
free-size パラメータ, 59
from パラメータ, 73, 74, 75, 79
fsize
 サーバサイドの HTML コマンド, 112
funcs パラメータ, 58

G

get-client-cert 関数, 77
get-sslid 関数, 73
groupdb パラメータ, 72
groupfn パラメータ, 72
group パラメータ, 84
grpfile パラメータ, 73

H

HashInitSize パラメータ, 69
HeaderBufferSize 指令, 62, 124
header パラメータ, 83
home-page 関数, 74
hostname 属性, 36
hosts 属性, 49
HTML タグ
 サーバによる構文解析, 11
 サーバによるコマンド解析, 109
HttpApplets, 29
HTTPVersion 指令, 62, 124

I

icon-uri パラメータ, 55
id 属性

ACLFILE 要素, 94
CLUSTER 要素, 36
CONNECTIONGROUP 要素, 92
connection-queue 要素, 45
LS 要素, 91
MASTER 要素, 36
MIME 要素, 94
profile 要素, 45
SLAVE 要素, 36
thread-pool 要素, 45
USERDB 要素, 97
virtual-server 要素, 48
VSCLASS 要素, 95
VS 要素, 96
 サーバ要素, 44
ignore パラメータ, 55
imagemap 関数, 83
include
 サーバサイドの HTML コマンド, 111
include コマンドの file 属性, 112
include コマンドの virtual 属性, 111
index-common 関数, 83
index-names パラメータ, 76
index-simple 関数, 83
initArgs プロパティ, 38, 100, 128, 130
init-cgi 関数, 57
init-clf 関数, 57
InitFn プロパティ, 35
init-param 要素, 105
init-uhome 関数, 57
Init 関数, 54, 125, 127
inputStreamLengthCheck プロパティ, 39, 128
instance 属性, 36
interfaces 属性, 49
interval 属性, 105
IOTimeout 指令, 62
iponly パラメータ, 87
ip 属性, 91
isModifiedCheckAggressive プロパティ, 40, 129
is-secure 属性, 107
iwsstats.xml, 43

J

Java

- JDK のバージョン, 22
- JSP, 21
- servlet.jar, 21
- サーバサイドのアプレット, 29
- サーブレット, 21

java.compiler 設定, 52

Java Development Kit

- JDK を参照

Java Runtime Environment

- JRE を参照

JavaServer Pages

- JSP を参照

Java Servlet API のマニュアル, 23

JDK

- インストール, 22
- 切り替え, 22
- 設定, 22
- バージョン, 22

JRE

- インストール, 22
- 切り替え, 22
- 設定, 22

JSP, 21

- API 参照, 23
- インストールする, 23
- 参照先, 23
- 有効にする, 22

jsp-servlet 要素, 22, 105

jvm12.conf, 50

jvm.allowExit 設定, 52

jvm.classpath 設定, 53

jvm.disableThreadRecycling 設定, 52

jvm.enableClassGC 設定, 51

jvm.enableDebug 設定, 51

jvm.include.CLASSPATH 設定, 53

jvm.maxHeapSize 設定, 51

jvm.minHeapSize 設定, 51

jvm.option 設定, 51

jvm.printErrors 設定, 51

jvm.serializeAttach 設定, 52

jvm.serializeFirstRequest 設定, 53

jvm.stickyAttach 設定, 52

jvm.trace 設定, 52

jvm.verboseMode 設定, 51

K

keepalive-bucket 要素属性, 47

KeepAliveThreads 指令, 62, 125

KeepAliveTimeout 指令, 62, 125

KernelThreads 指令, 62, 125

key-toosmall 関数, 83

L

lang パラメータ, 79, 80

LAST_MODIFIED

サーバサイドの HTML コマンドの変数, 113

legacyls 属性, 91

library プロパティ, 35

list-dir 関数, 83

ListenQ 指令, 62, 125

load-config 関数, 77

load-modules 関数, 57

LoadObjects 指令, 125

LogFlushInterval 指令, 62, 125

LogVerbose 指令, 62, 125

LogVsId 指令, 63

LS 要素, 91

M

magnus.conf, 53

NSAPI プラグインをインストールする, 25

新しい認証サービスを追加する, 26

サーバによるタグの構文解析の有効化, 11

- サブレットを有効にする, 22
- 変更点, 123
- make-dir 関数, 83
- makefrombasic パラメータ, 75
- map-to 属性, 106
- MASTER 要素, 36
- matchingip 属性, 93
- MaxAge パラメータ, 69
- maxbps 属性, 97
- maxCacheEntries 属性, 47
- MaxCGIStubs 指令, 63, 125
- maxConnections 属性, 47
- maxconn 属性, 97
- maxEntries 属性, 47
- MaxFiles パラメータ, 69
- maxHeapCacheSize 属性, 48
- MaxKeepAliveConnections 指令, 63, 125
- maxMmapCacheSize 属性, 48
- maxOpenEntries 属性, 47
- MaxProcs 指令, 63, 125
- maxProcs 属性, 44
- maxQueued 属性, 46
- MaxRqHeaders 指令, 63, 125
- MaxThreads 指令, 125
- maxThreads 属性, 44, 46
- maxthreads パラメータ, 59
- maxVirtualServers 属性, 44
- MediumFileSizeLimit パラメータ, 69
- MediumFileSpace パラメータ, 69
- methods 関数, 59
- method 属性, 49
- method パラメータ, 75, 77, 81
- mime.types, 67
 - CGI 拡張子を指定する, 13
- mime 属性, 96
- MIME 要素, 94
- MinCGIStubs 指令, 63, 125
- MinProcs 指令, 125

- MinThreads 指令, 125
- minthreads パラメータ, 59
- mode 属性, 45, 48
- MtaHost 指令, 63, 125

N

- NameTrans 関数, 73
- NameTrans 指令, 71
- name 指令, 41
- name 属性, 45
- name パラメータ
 - assign-name 関数, 73
 - common-log 関数, 87
 - define-perf-bucket 関数, 55
 - flex-log 関数, 87
 - pfx2dir 関数, 74
 - record-useragent 関数, 87
 - thread-pool-init 関数, 59
 - unix-home 関数, 75
- NativePoolMaxThreads 指令, 63, 125
- NativePoolMinThreads 指令, 63, 125
- NativePoolQueueSize 指令, 63, 125
- NativePoolStackSize 指令, 63, 125
- NativeThreads, 11, 109
- NativeThread パラメータ, 58
- nes.jsp.enableddebug 設定, 52
- nes.jsp.forkjavac 設定, 53
- net_write
 - NSAPI 関数, 117
- NetSiteRoot 指令, 63, 125
- nice パラメータ, 85
- nocache パラメータ, 85
- noostat パラメータ, 73
- NSAPI, 10, 24
 - 参照先, 25
 - プラグインをインストールする, 25
 - ヘッダーファイル, 24
 - 有効にする, 24
- nsapi.h, 24

NSAPI (Netscape Server Application Programming Interface)

NSAPI を参照

ns-cron.conf, 68

nsessions プロパティ, 42

nsfc.conf, 68

NSIntAbsFilePath パラメータ, 82

ntgcicheck 関数, 78

nt-console-init 関数, 58

ntrans-base, 73, 74, 77

nt-uri-clean 関数, 78

num-buffers パラメータ, 56

O

obj.conf, 70

CGI 拡張子を指定する, 13

CGI ディレクトリを指定する, 13

NSAPI を使って SAF を記述する, 24

サーバによるタグの構文解析の有効化, 11
指令, 24

変更点, 127

objectfile 属性, 95

ObjectType 関数, 79

ObjectType 指令, 71

OPTITDIR 設定, 52

opts パラメータ, 54

outputStreamFlushTimer プロパティ, 39, 128

P

parameterEncoding プロパティ, 40, 129

parameter-encoding 要素, 105

param-name 要素, 105

param-value 要素, 105

parse-html SAF, 110

parse-html 関数, 127

password.conf, 88

PathCheck 関数, 75

PathCheck 指令, 71

path パラメータ

deny-existence 関数, 76

home-page 関数, 74

query-handler 関数, 84

require-auth 関数, 78

send-error 関数, 88

peakQueued 属性, 46

perf-init 関数, 58

pfx2dir 関数, 74

PidLog 指令, 64, 125

pid 属性, 45

pool-init 関数, 58

pool パラメータ, 58

Port 指令, 126

port 属性, 92

PostThreadsEarly 指令, 64, 126

process 要素, 45

profile-bucket 要素, 50

profile 属性, 50

profile 要素, 45

profiling パラメータ, 59

protocol 属性, 37

pwfile パラメータ, 57, 75

Q

qosactive 属性, 91

qos-error 関数, 88, 127

qos-handler 関数, 73, 127

qosmetricsinterval 属性, 91

QOSPARAMS 要素, 97

qosrecomputeinterval 属性, 91

QUERY_STRING_UNESCAPED

サーバサイドの HTML コマンドの変数, 113

query-handler 関数, 84

query パラメータ, 81

queueSize パラメータ, 59

R

rateBytesTransmitted 属性, 49
RcvBufSize 指令, 64, 126
readme パラメータ, 83
realm パラメータ, 78
reason パラメータ, 88
record-useragent 関数, 87
redirect 関数, 74
register-http-method 関数, 59
reload-interval 属性, 103
reloadInterval プロパティ, 38, 128
remove-dir 関数, 84
remove-file 関数, 84
rename-file 関数, 84
request-bucket 要素, 49
require-auth 関数, 78
require パラメータ, 75, 77
respondCookieVersion プロパティ, 38, 128
response-buffer 要素, 106
response-cookie 要素, 106
rlimit_as パラメータ, 84
rlimit_core パラメータ, 84
rlimit_nofile パラメータ, 85
role-mapping 要素, 106
RootObject 指令, 126
rootobject 属性, 95
root パラメータ, 74
rotate-access パラメータ, 56
rotate-callback パラメータ, 56
rotate-error パラメータ, 56
rotate-interval パラメータ, 56
rotate-start パラメータ, 56
RqThrottleMinPerSocket 指令, 64
RqThrottle 指令, 64, 126

rules.properties, 89
変更点, 129

S

SAF, 24
インストールする, 25
参照先, 25
secondsMaxAge 属性, 47
secondsRunning 属性, 44
secondsTimeout 属性, 47
secret-keysize パラメータ, 79
Security 指令, 64, 126
security 属性, 92
send-cgi 関数, 84, 127
send-error 関数, 88
send-file 関数, 85
send-range 関数, 85
send-shellcgi 関数, 85
send-wincgi 関数, 85
Server Application Functions
SAF を参照
servercertnickname 属性, 93
ServerCert 指令, 126
ServerID 指令, 64, 126
ServerKey 指令, 126
ServerName 指令, 126
servername 属性, 93
serverName プロパティ, 39, 129
ServerRoot 指令, 64, 126
servers.lst, 98
server.xml, 90
SERVER 要素, 90
server 要素, 44
service-dump 関数, 86
Service 関数, 81
Service 指令, 71
Service パラメータ, 81

Servlet.jar, 21
servlets.properties, 99
 変更点, 129
<SERVLET> タグ, 114
session-cookie 要素, 107
session-manager 要素, 107
sessionmgr プロパティ, 38, 128
sessionmgr プロパティ, 38, 101, 128, 130
session-tracking 要素, 107
set-default-type 関数, 80
set-virtual-index 関数, 78
shexp パラメータ, 76
shlib パラメータ, 58
SHTML, 114
shtml_add_tag, 116
 サーバサイドのカスタムタグを登録する関数
 , 121
shtml_init, 11, 109
shtml_public.h, 117
shtml_send, 11, 109
shtml_send 関数, 86, 127
shtml.dll, 117
shtml-hacktypem 関数, 80
ShtmlMaxDepth パラメータ, 86
shtml.so, 117
ShtmlTagExecuteFunc
 サーバサイドのタグを定義する関数, 117
ShtmlTagInstanceLoad
 サーバサイドのタグを定義する関数, 117
ShtmlTagInstanceUnload
 サーバサイドのタグを定義する関数, 118
ShtmlTagPageLoadFunc
 サーバサイドのタグを定義する関数, 118
ShtmlTagPageUnLoadFn
 サーバサイドのタグを定義する関数, 118
singleClassLoader プロパティ, 39, 129
sizeHeapCache 属性, 48
sizeMmapCache 属性, 48
size 属性, 106
SLAVE 要素, 36
SmallFileSizeLimit パラメータ, 69

SmallFileSpace パラメータ, 69
SndBufSize 指令, 64, 126
SSI, 114
ssl2ciphers 属性, 93
SSL2 指令, 126
ssl2 属性, 93
SSL3Ciphers 指令, 126
SSL3SessionTimeout 指令, 64, 126
ssl3tlsciphers 属性, 94
SSL3 指令, 126
ssl3 属性, 94
SSLCacheEntries 指令, 65, 126
ssl-check 関数, 79
SSLClientAuthDataLimit 指令, 65, 126
SSLClientAuthTimeout 指令, 65, 126
SSLClientAuth 指令, 126
ssl-logout 関数, 79
SSLPARAMS 要素, 93
SSLSessionTimeout 指令, 65, 126
StackSize 指令, 65, 127
stackSize パラメータ, 59
startup プロパティ, 100, 101, 130
state 属性, 96
stats-init 関数, 59
stats-xml 関数, 86, 127
stats 要素, 44
Status 指令, 68
stderr パラメータ, 58
stdout パラメータ, 58
StrictHttpHeaders 指令, 65, 127
strip-params 関数, 75
subdir パラメータ, 75
substitute 属性, 37

T

TagUserData

- サーバサイドのカスタムタグのデータ構造
 - , 117, 118
- TempDirSecurity 指令, 65
- TempDir 指令, 65
- TempDir パラメータ, 69
- tempDir プロパティ, 38, 128
- tempdir 要素, 107
- TerminateTimeout 指令, 65, 127
- ThreadIncrement 指令, 65, 127
- thread-pool-bucket 要素, 46
- thread-pool-init 関数, 59
- thread-pool 属性, 46
- thread-pool 要素, 45
- thread 要素, 48
- ticksDispatch 属性, 50
- ticksFunction 属性, 50
- ticksPerSecond 属性, 44
- ticksTotalQueued 属性, 46
- timefmt タグ, 111
- timefmt パラメータ, 83
- timeOut 属性, 104
- timeout パラメータ, 57
- timeStarted 属性, 44, 45, 48
- timezone パラメータ, 54
- Time 指令, 41
- tlsrollback 属性, 94
- tls 属性, 94
- trailer パラメータ, 83
- TransmitFiles パラメータ, 69
- type-by-exp 関数, 80
- type-by-extension 関数, 81
- type パラメータ, 79, 80, 81, 86

U

- Umask 指令, 66, 127
- UNIX
 - サーバの制約, 124

- unix-home 関数, 75
- unix-uri-clean 関数, 79
- update-interval パラメータ, 59
- upload-file 関数, 86
- uri 属性, 49, 108
- uri パラメータ, 82
- uri プロパティ, 39, 128
- URL
 - ファイルパスに変換, 71
- urlhosts 属性, 96
- url-prefix パラメータ, 74
- url パラメータ, 74
- use-cookies 属性, 107
- UseNativePoll 指令, 66, 127
- UseOutputStreamSize 指令, 66, 127
- UseOutputStreamSize パラメータ, 81
- userdb パラメータ, 72, 75
- USERDB 要素, 97
- userfile パラメータ, 72
- userfn パラメータ, 72
- User 指令, 41, 66, 127
- user パラメータ, 84
- use-url-rewriting 属性, 107

V

- VARS 要素, 91
- verifycert プロパティ, 35
- versionMajor 属性, 44
- versionMinor 属性, 44
- versionServer 属性, 44
- version 属性, 106
- virtual-index パラメータ, 78
- VirtualServerFile 指令, 64, 127
- virtual-servers パラメータ, 59
- virtual-server 属性, 48
- virtual-server 要素, 48

VSCLASS 要素, 95

VS 要素, 96

vs 要素, 108

W

web-apps.xml, 102

web-app 要素, 108

web.xml, 101

widths パラメータ, 54

WincgiTimeout 指令, 66, 127

あ

アクセス制御 API, 25

新しい認証サービスを読み込む, 26

参照先, 26, 27

い

インストール

JRE または JDK, 22

インストールする

JSP, 23

サーブレット, 23

プラグイン (SAF), 25

え

エージェント API, 29

エラー

カスタマイズされたメッセージを送信, 88

か

概要, 9

拡張

サーバ機能, 10

拡張する

サーバ, 24

環境変数

サーバサイドの HTML コマンド, 113

き

キャッシュ

統計ファイル, 85

く

クライアント証明書 CGI 変数, 19

こ

構成ファイル, 10

場所, 10

このマニュアルについて, 7

コンパイルする

Java サーブレット, 21

さ

サーバ

制約, 124

サーバ拡張機能, 24

サーバサイドの HTML コマンド, 109

config, 111

echo, 112

exec, 112

flastmod, 112

fsize, 112

- include, 111
 - 環境変数, 113
 - 形式, 110
 - サーバサイドのタグを新規作成, 116
 - 使用方法, 110
- サーバサイドの HTML タグ
 - 作成, 116
- サーバサイドのアプレット, 29
- サーバサイドのカスタム HTML タグ
 - 初期化関数, 121
 - 定義, 116
 - 読み込み, 121
- サーバサイドのタグ, 11
 - exec, 11
 - 構文解析を有効にする, 11
- サーバによる HTML タグの構文解析, 11
- サーバの動作
 - 変更, 10
- サーバプラグイン, 24
- サブレット
 - HTML ファイルに埋め込む, 114
 - インストールする, 23
 - コンパイルする, 21
 - 作成の詳細, 23
 - 参照先, 23
 - 特定のタイプのファイルを特定のサブレットに振り向ける, 89
 - 米国 Sun Microsystems, Inc. の API マニュアル, 21
 - 有効にする, 22
- 作成
 - サーバサイドのカスタムタグ, 116
 - 動的な内容, 10
- 作成する
 - サブレット, 23

し

- 紹介, 9
- 指令, 24

せ

- セキュリティ
 - サーバの制約, 124
- 設定
 - JDK, 22
 - JRE, 22

た

- タグ
 - サーバによる HTML 構文解析, 109
- タグ実行関数
 - サーバサイドのカスタムタグ, 116

て

- 定義
 - サーバサイドのカスタム HTML タグ, 116
- 定義する
 - サブレット, 23
- ディレクトリ
 - CGI, 12

な

- 内容
 - サーバでの変更, 11
 - 動的に作成, 10
- 内容を動的に作成, 10

は

- はじめに, 7

ふ

ファイルの拡張子
CGI, 13

プラグイン
インストールする, 25
作成, 24
参照先, 25

へ

ヘッダーファイル
nsapi.h, 24
shtml_public.h, 117

変更
API, 29
サーバの動作, 10

編集する
ACL ファイル, 26

変数
CGI, 19
サーバサイドの HTML コマンド, 113

も

戻りコード
REQ_ABORTED, 117
REQ_EXIT, 117
REQ_NOACTION, 117
REQ_PROCEED, 117

ゆ

有効にする
CGI, 12
JSP, 22
NSAPI, 24
サーバサイドのタグ, 11
サーブレット, 22

よ

要約
API, 28

読み込み
サーバサイドのカスタム HTML タグ, 121

読み込みモジュール SAF, 25, 121

読み込む
NSAPI プラグイン, 25
新しい認証サービス, 26