

管理者用設定ファイルリファレンス

Sun™ ONE Application Server

Version 7, Enterprise Edition

817-5549-10

2003 年 9 月

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.

Copyright © 2003 Sun Microsystems, Inc. All rights reserved.

このソフトウェアは SUN MICROSYSTEMS, INC. の機密情報と企業秘密を含んでいます。SUN MICROSYSTEMS, INC. の書面による許諾を受けることなく、このソフトウェアを使用、開示、複製することは禁じられています。U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms.

この配布には、第三者が開発したソフトウェアが含まれている可能性があります。

Sun、Sun Microsystems、Sun のロゴマーク、Java、Sun™ ONE、Java Coffee Cup のロゴマークおよび Sun™ ONE 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。

UNIX は、X/Open Company, Ltd が独占的にライセンスしている米国およびその他の国における登録商標です。

この製品は、米国の輸出規制に関する法規の適用および管理下にあり、また、米国以外の国の輸出および輸入規制に関する法規の制限を受ける場合があります。核、ミサイル、生物化学兵器もしくは原子力船に関連した使用またはかかる使用者への提供は、直接的にも間接的にも、禁止されています。このソフトウェアを、米国の輸出禁止国へ輸出または再輸出すること、および米国輸出制限対象リスト (輸出が禁止されている個人リスト、特別に指定された国籍者リストを含む) に指定された、法人、または団体に輸出または再輸出することは一切禁止されています。

目次

本書について	9
対象読者	9
マニュアルの使用方法	10
マニュアルの構成	12
マニュアルの表記規則	13
一般的な表記規則	13
ディレクトリ名の表記規則	14
製品サポート	15
第1章 サーバー動作の基本	17
設定ファイル	17
server.xml	18
init.conf	18
obj.conf	18
mime.types	19
動的再設定	19
設定ファイルの手動編集	20
クラスタの再設定	22
第2章 サーバー設定ファイル	23
sun-server_1_1.dtd ファイル	24
サブ要素	25
データ	25
属性	26
server.xml ファイル内の要素	26
一般的な要素	27

server	27
property	29
description	30
admin-service	30
server-instance	30
リスナーサービス要素	31
http-service	31
http-listener	32
ssl	35
mime	37
acl	37
virtual-server-class	38
virtual-server	39
http-qos	43
auth-db	44
iiop-service	45
orb	46
ssl-client-config	47
iiop-listener	47
コンテナ要素	49
web-container	49
ejb-container	50
mdb-container	53
可用性要素	56
availability-service	56
persistence-store	58
session-config	58
session-manager	59
manager-properties	60
store-properties	62
session-properties	64
J2EE サービス要素	66
jms-service	66
log-service	68
security-service	70
auth-realm	71
transaction-service	73
Java 設定要素	75
java-config	75
profiler	78
jvm-options	79
リソース要素	80
resources	80

custom-resource	81
external-jndi-resource	82
jdbc-resource	83
mail-resource	84
jms-resource	85
persistence-manager-factory-resource	87
jdbc-connection-pool	88
アプリケーション要素	92
applications	92
lifecycle-module	93
j2ee-application	94
ejb-module	95
web-module	96
connector-module	97
ユーザーデータベースの選択	98
Sun ONE LDAP スキーマ	99
収束ツリー	100
ドメインコンポーネント (dc) ツリー	100
変数	101
変数の形式	101
id 変数	102
その他の重要な変数	102
一般的な変数	102
send-cgi 変数	102
変数の評価	103
server.xml ファイルのサンプル	104
 第 3 章 init.conf の構文と使い方	109
Init 関数	111
サーバー情報	111
NetSiteRoot	111
TempDir	111
TempDirSecurity	112
User (UNIX のみ)	112
DNS 検索	113
AsyncDNS	113
DNS	113
スレッド、プロセス、および接続	114
ConnQueueSize	115
HeaderBufferSize	115
IOTimeout	115
KeepAliveThreads	116

KeepAliveTimeout	116
ListenQ	116
MaxKeepAliveConnections	116
PostThreadsEarly	117
RcvBufSize	117
RqThrottle	117
RqThrottleMin	117
SndBufSize	118
StackSize	118
StrictHttpHeaders	118
TerminateTimeout	118
ThreadIncrement	118
ネイティブのスレッドプール	119
NativePoolStackSize	119
NativePoolMaxThreads	120
NativePoolMinThreads	120
NativePoolQueueSize	120
CGI	121
CGIExpirationTimeout	121
CGIStubIdleTimeout	121
MaxCGIStubs	121
MinCGIStubs	122
エラー記録	122
ErrorLogDateFormat	122
LogFlushInterval	122
PidLog	122
ACL	124
ACLCacheLifetime	124
ACLUserCacheSize	124
ACLGroupCacheSize	124
セキュリティ	125
Security	125
SSLCacheEntries	126
SSLClientAuthDataLimit	126
SSLClientAuthTimeout	126
SSLSessionTimeout	126
SSL3SessionTimeout	127
チャンクされたエンコード	127
UseOutputStreamSize	127
ChunkedRequestBufferSize	128
ChunkedRequestTimeout	128
その他	129
ChildRestartCallback	129

HTTPVersion	129
MaxRqHeaders	129
ReentrantTimeFunctions (Solaris のみ)	130
Umask (UNIX のみ)	130
第 4 章 MIME タイプ	131
はじめに	132
MIME タイプの決定	132
Type の応答への影響	133
クライアントの行う MIME タイプの処理	134
MIME タイプファイルの構文	134
MIME タイプファイルの例	134
第 5 章 その他の設定ファイル	137
dbswitch.conf	138
配備記述子	140
generated.instance.acl	141
loadbalancer.xml	141
nsfc.conf	144
password.conf	146
server.policy	147
付録 A 日時の形式	149
付録 B サーバー設定要素のアルファベット順リスト	151
付録 C init.conf の指令のアルファベット順リスト	155
索引	159

本書について

本書では、`server.xml` や `init.conf`、`mime.types` を含めた Sun™ ONE Application Server 7, Enterprise Edition に使用する設定ファイルの目的を説明します。また、これらの設定ファイルに含まれる要素や指令の包括的なリストを提供します。

この章には次のトピックがあります。

- [対象読者](#)
- [マニュアルの使用法](#)
- [マニュアルの構成](#)
- [マニュアルの表記規則](#)
- [製品サポート](#)

対象読者

このマニュアルは、Sun ONE Application Server の管理および保守の担当者を対象にしています。

このマニュアルでは、次のトピックに精通していることを前提としています。

- Java™ 2 Platform, Enterprise Edition (J2EE™ platform) 仕様
- HTTP
- HTML
- XML
- Java プログラミング
- Java™ Servlet、JavaServer Pages™ (JSP™)、Enterprise JavaBeans™ (EJB™)、および Java™ Database Connectivity (JDBC™) 仕様に定義されている Java API
- リレーショナルデータベースの概念

マニュアルの使用法

Sun ONE Application Server 7, Enterprise Edition のマニュアルは、PDF 形式または HTML 形式でも入手できます。

次の表は、Sun ONE Application Server のマニュアルに記述されているタスクと概念を示しています。

表 1 Sun ONE Application Server マニュアルの概要

情報の内容	参照するマニュアル
ソフトウェアおよびマニュアルの最新情報	『リリースノート』
サポート対象のハードウェア、オペレーティングシステム、JDK、JDBC、RDBMS の一覧	『Platform Summary』
Sun ONE Application Server 7 の概要と、製品の各エディションで利用可能な機能	『製品の概要』
サーバーのアーキテクチャの図と説明、Sun ONE Application Server アーキテクチャの利点	『サーバーアーキテクチャの概要』
企業、開発者、および運用向けの、Sun ONE Application Server 7 の新機能	『新機能』
Sun ONE Application Server 7 製品の基本的な使用方法。サンプルアプリケーションのチュートリアルも含まれる	『入門ガイド』
Sun ONE Application Server ソフトウェアとそのコンポーネント (サンプルアプリケーション、管理インタフェース、高可用性コンポーネントなど) のインストール。高可用性の基本設定の実装手順が記載されている	『インストールガイド』
Sun ONE Application Server を確実にサイトに適合させるように配備するための、システムのニーズとエンタープライズの評価。アプリケーションサーバーを配備する際に注意する必要がある、一般的な問題と懸案事項も記載されている	『システム配備ガイド』
アプリケーションの設計者と開発者が、HTTP セッションの可用性を確保するための最適な方法	『Application Design Guidelines for Storing Session State』
J2EE コンポーネント (サーブレット、EJB™ (Enterprise JavaBeans™)、JSP™ (JavaServer Pages™) など) 向け Java オープンスタンダードモデルに準拠した Sun ONE Application Server 7 上で実行することを目的とした Java™ Platform, Enterprise Edition (J2EE™ プラットフォーム) アプリケーションの作成および実装。アプリケーション設計、開発ツール、セキュリティ、アセンブリ、配備、デバッグ、ライフサイクルモジュールの作成方法などについての一般的な情報が含まれる。Sun ONE Application Server のさまざまな用語について解説する用語集も含まれる	『Developer's Guide』

表 1 Sun ONE Application Server マニュアルの概要 (続き)

情報の内容	参照するマニュアル
Sun ONE Application Server 7 の Java™ Servlet および JSP (JavaServer Pages) 仕様に準拠した J2EE Web アプリケーションの作成および実装。Web アプリケーションプログラミングの概念とタスクの説明、サンプルコード、実装のヒント、関連資料の紹介など。結果キャッシュ機能、JSP のプリコンパイル、セッション管理、セキュリティ、配備、SHTML、CGI などが含まれる	『Developer's Guide to Web Applications』
Sun ONE Application Server 7 のエンタープライズ Bean 向け Java オープンスタンダードモデルに準拠した J2EE アプリケーションの作成および実装。Enterprise JavaBeans (EJB) プログラミングの概念とタスクの説明、サンプルコード、実装のヒント、関連資料の紹介など。コンテナ管理持続性、読み取り専用 Bean、エンタープライズ Bean に関連付けられた XML ファイルや DTD ファイルなどが含まれる	『Developer's Guide to Enterprise JavaBeans Technology』
Sun ONE Application Server 7 上で J2EE アプリケーションにアクセスする Application Client Container (ACC) の作成	『Developer's Guide to Clients』
Sun ONE Application Server 環境での Web サービスの作成	『Developer's Guide to Web Services』
Java™ Database Connectivity (JDBC™)、トランザクション、Java Naming and Directory Interface™ (JNDI)、Java™ Message Service (JMS)、および JavaMail™ API	『Developer's Guide to J2EE Services and APIs』
カスタム NSAPI プラグインの作成	『NSAPI Developer's Guide』
管理インタフェースとコマンド行インタフェースの両方からの Sun ONE Application Server サブシステムとコンポーネントの設定、管理、および配備の説明と手順。クラスタ管理、高可用性データベース、ロードバランス、およびセッションの持続性について説明する。Sun ONE Application Server のさまざまな用語について解説する用語集も含まれる	『管理者ガイド』
server.xml ファイルなどの Sun ONE Application Server の設定ファイルの編集	『管理者用設定ファイルリファレンス』
Sun ONE Application Server 運用環境のセキュリティの設定および管理。一般的なセキュリティ、証明書、および SSL/TLS 暗号化に関する情報など。HTTP サーバーベースのセキュリティについても説明	『セキュリティ管理者ガイド』
Sun ONE Application Server 7 用の J2EE™ CA (Connector Architecture) コネクタのサービスプロバイダ実装の設定と管理。管理ツール、プーリングモニター、JCA コネクタの配備、サンプルコネクタとサンプルアプリケーションなどについて説明する	『J2EE CA Service Provider Implementation Administrator's Guide』
新しい Sun ONE Application Server 7 プログラミングモデルへのアプリケーションの移行 (特に、iPlanet Application Server 6.x、Netscape Application Server 4.0 からの移行)。移行例も含まれる	『サーバーアプリケーションの移行および再配備』
Sun ONE Application Server のパフォーマンスを改善する方法と、なぜそうする必要があるか	『Performance Tuning Guide』

表 1 Sun ONE Application Server マニュアルの概要 (続き)

情報の内容	参照するマニュアル
Sun ONE Application Server に関する問題の解決	『Troubleshooting Guide』
Sun ONE Application Server 7 の実行中に表示される可能性のあるメッセージ。メッセージが生成された状態について、考えられる原因とガイドラインを説明する	『Error Message Reference』
マニュアルページに記載されている、Sun ONE Application Server で利用できるユーティリティコマンド	『Utility Reference Manual』
Sun™ Open Net Environment (Sun ONE) Message Queue ソフトウェアの使用	Sun ONE Message Queue については次の URL を参照： http://docs.sun.com/db/prod/s1.s1msggu?l=ja#hic

マニュアルの構成

このマニュアルは、次の章および付録で構成されています。

- 第 1 章「サーバー動作の基本」
この章では、Sun ONE Application Server を制御する主な設定ファイルを紹介し、設定ファイルを編集し有効にする方法について説明します。
- 第 2 章「サーバー設定ファイル」
この章では、サーバーの動作の大部分を制御する `server.xml` ファイルについて説明します。
- 第 3 章「`init.conf` の構文と使い方」
この章では、`init.conf` ファイルに指令を設定して、初期化時に Sun ONE Application Server を設定する方法について説明します。
- 第 4 章「MIME タイプ」
この章では、ファイル拡張子をファイルタイプにマップする MIME タイプファイルについて説明します。
- 第 5 章「その他の設定ファイル」
この章では、その他の重要な設定ファイルを一覧で表示し、その内容が一目で分かるようにしています。
- 付録 A「日時の形式」
この付録では、サーバーログで日時に使われる形式文字列について説明します。

- [付録 B「サーバー設定要素のアルファベット順リスト」](#)
[付録 C「init.conf の指令のアルファベット順リスト」](#)

これらの付録には、`server.xml` の要素と `init.conf` の指令を簡単に検索できるように、アルファベット順のリストが記載されています。

マニュアルの表記規則

この節では、このマニュアルの表記規則について説明します。

- [一般的な表記規則](#)
- [ディレクトリ名の表記規則](#)

一般的な表記規則

このマニュアルは、次の表記規則に従っています。

- **ファイルとディレクトリのパス**は、UNIX の形式で表記します (ディレクトリ名を「/」記号で区切って表記)。
- **URL** は次の書式で記述します。

`http://server.domain/path/file.html`

server はアプリケーションを実行するサーバー名、*domain* はユーザーのインターネットドメイン名、*path* はサーバー上のディレクトリの構造、*file* は個別のファイル名を示します。URL の斜体文字の部分は可変部分です。

- **フォント**は、次のように使い分けます。
 - モノスペースフォントは、コード例、コードリスト、API および言語要素 (関数名、クラス名など)、ファイル名、パス名、ディレクトリ名、および HTML タグに使用します。
 - 斜体文字はコード変数に使用します。
 - 斜体文字は、変数および可変部分、およびリテラルに使われる文字にも使用します。
 - 太字は、段落の先頭またはリテラルに使われる文字の強調に使用します。
- このマニュアルでは、ほとんどのプラットフォームの**インストールルートディレクトリ**を `install_dir` と記述します。例外については、[14 ページの「ディレクトリ名の表記規則」](#)を参照してください。

デフォルトでは、ほとんどのプラットフォームの `install_dir` は次の場所になります。

`/opt/SUNWappserver7`

上記のプラットフォームで *default_config_dir* および *install_config_dir* は、*install_dir* と同義です。例外と追加情報については、[14 ページの「ディレクトリ名の表記規則」](#)を参照してください。

- このマニュアルでは、**インスタンスルートディレクトリ**は、*instance_dir* と記述します。これは以下のパスの省略形式です。

default_config_dir/domains/*domain*/*instance*

- このマニュアルを通じて、特に明記のない限り、すべての **UNIX 固有の表記**は、Linux オペレーティングシステムにも適用されます。

ディレクトリ名の表記規則

Solaris 8 および 9 のインストールでは、アプリケーションサーバーのファイルはデフォルトで複数のルートディレクトリにまたがって保存されます。ここでは、これらのディレクトリについて説明します。

- **Solaris 8 および 9 のインストール**では、デフォルトのインストールディレクトリは次のように表記されます。
 - *install_dir* は /opt/SUNWappserver7 を示します。このディレクトリにはインストールイメージの静的な要素が保存されます。ユーティリティ、実行可能ファイル、およびアプリケーションサーバーを構成するライブラリは、すべてここに保存されます。
 - *default_config_dir* は /var/opt/SUNWappserver7/domains を示します。このディレクトリは、作成したドメインのデフォルトの保存場所です。
 - *install_config_dir* は /etc/opt/SUNWappserver7/config を示します。このディレクトリには、ライセンスなどのインストール全体に適用される設定情報や、このインストール用に設定した管理ドメインのマスタリストが保存されます。

製品サポート

製品またはマニュアルに関する一般的なフィードバックは、appserver-feedback@sun.com までお送りください。

ご使用のシステムに問題が発生した場合は、次のいずれかの方法でカスタマサポートにお問い合わせください。

- 次のオンラインサポート Web サイトをご利用ください。

<http://www.sun.com/supporttraining/>

- 保守契約を結んでいるお客様の場合は、専用ダイヤルをご利用ください。

サポートのご依頼の前に、次の情報を用意してください。サポート担当がお客様の問題を解決するために必要な情報です。

- 問題が発生した箇所や動作への影響など、問題の具体的な説明
- マシン機種、OS バージョン、および、問題の原因と思われるパッチやその他のソフトウェアなどの製品バージョン
- 問題を再現するための具体的な手順の説明
- エラーログやコアダンプ

サーバー動作の基本

Sun™ Open Net Environment (Sun ONE) Application Server 7, Enterprise Edition の設定および動作は設定ファイルで決まります。管理インタフェースまたはコマンド行を使用すると、これらの設定ファイルの設定を変更できます。

この章には次の節があります。

- [設定ファイル](#)
- [動的再設定](#)
- [設定ファイルの手動編集](#)
- [クラスタの再設定](#)

設定ファイル

Sun ONE Application Server の設定および動作は設定ファイルによって制御されます。設定ファイルは、*instance_dir/config/* ディレクトリにあります。このディレクトリには、異なるコンポーネントを制御するためのさまざまな設定ファイルがあります。設定ファイルの実際の数と名前は、サーバーで有効になっているコンポーネントまたはサーバーに読み込まれたコンポーネントによって異なります。

ただし、このディレクトリには、サーバーの動作に不可欠な 4 つの設定ファイルが常に含まれています。

- [server.xml](#) -- ほとんどのサーバー設定情報が含まれている
- [init.conf](#) -- グローバルサーバー初期化情報が含まれている
- [obj.conf](#) -- クライアントからの HTTP 要求を処理する命令が含まれている
- [mime.types](#) -- 要求されたリソースのコンテンツタイプを決定する情報が含まれている

dbswitch.conf および loadbalancer.xml などのその他の重要な設定ファイルについては、[第 5 章「その他の設定ファイル」](#)を参照してください。

server.xml

このファイルには、ほとんどのサーバー設定情報が含まれています。ファイルの書式と内容は、スキーマファイル sun-server_1_1.dtd に定義されます。サーバーがどのように sun-server_1_1.dtd および server.xml を使用するかについての詳細は、[第 2 章「サーバー設定ファイル」](#)を参照してください。

init.conf

このファイルでは、初期化時にサーバーを設定する変数の値を設定します。サーバーはこのファイルを参照し、起動時に設定を実行します。サーバーは、再起動されるまでこのファイルを再び参照することはありません。サーバーがどのように init.conf を使用するかについては、[第 3 章「init.conf の構文と使い方」](#)を参照してください。

obj.conf

このファイルには、クライアントからの HTTP 要求の処理方法、またネイティブサーバープラグインや CGI プログラムなどの Web サーバーのコンテンツの提供方法をサーバーに示す指示が含まれています。obj.conf ファイルには、要求応答プロセスの各段階でどのような処理を行うかを Sun ONE Application Server に指示する命令 (指令) が含まれています。サーバーは、クライアントからの要求を処理するときは必ず、このファイルで定義されている設定を参照します。

すべての obj.conf ファイルは、instance_dir/config/ ディレクトリにあります。複数の仮想サーバーが obj.conf ファイルを共有するように設定されている場合を除き、仮想サーバーごとに 1 つの obj.conf ファイルがあります。このマニュアルで示す「obj.conf ファイル」とは常に、すべての obj.conf ファイル、または説明中の仮想サーバーの obj.conf ファイルのことを指します。

プレフィックスを持たない obj.conf ファイルは、各仮想サーバーの obj.conf ファイルを作成するときに Sun ONE Application Server が使用するテンプレートです。このファイルを編集しても既存の仮想サーバーには影響しません。ただし、それ以後に作成する仮想サーバーには影響します。

デフォルト設定では、アクティブな `obj.conf` ファイルには `virtual_server_name-obj.conf` という名前が付けられます。サーバーインスタンスのデフォルトの仮想サーバーにはインスタンスに基づいた名前が付けられるので、サーバーインスタンスの作成時に `obj.conf` ファイルには `instance_name-obj.conf` という名前が付けられます。

サーバーがどのように `obj.conf` を使用するかの詳細については、『Sun ONE Application Server Developer's Guide to NSAPI』を参照してください。

mime.types

このファイルは、ファイル拡張子を MIME タイプにマップして、要求されたリソースのコンテンツタイプをサーバーが判別できるようにします。たとえば、`.html` 拡張子を持つリソースの要求は、クライアントが HTML ファイルを要求していることを意味し、`.gif` 拡張子を持つリソースの要求は、クライアントが GIF フォーマットのイメージファイルを要求していることを意味します。サーバーがどのように `mime.types` を使用するかについては、[第 4 章「MIME タイプ」](#)を参照してください。

動的再設定

`server.xml` ファイルの `applications` 要素と `resources` 要素に変更を加えると、その変更内容はすぐに適用されます。たとえば、Java™ 2 Platform, Enterprise Edition (J2EE™ プラットフォーム) アプリケーションの配備、Java™ Database Connectivity (JDBC™) 接続プールの登録などの変更がこれに該当します。変更内容を適用するために、サーバーを再起動する必要はありません。

`loadbalancer.xml` ファイルの変更がすぐに適用されるように設定したい場合は、このファイルの `reload-poll-interval-in-seconds` プロパティを変更します。詳細については、[141 ページの「loadbalancer.xml」](#)を参照してください。

次のファイルに変更を加えた場合、変更内容を適用するためにサーバーを再起動する必要はありません。ただし、変更内容を明示的に適用する必要があります。

- `obj.conf`
- `mime.types`
- ACL ファイル
- `server.xml` (`applications` および `resources` を除くすべての要素)

変更内容を適用するには、次のように管理インタフェースを使います。

1. サーバーインスタンスのページに移動します。

2. 「General (一般)」タブを選択します。
3. 「Select the Apply Changes (変更の適用)」ボタンを選択します。

または、次のようにコマンド行を使います。

```
asadmin reconfig --user user [--password password] [--passwordfile password_file]
[--host machine] [--port port] [--discardmanualchanges=false |
--keepmanualchanges=false] instance_name
```

次に例を示します。

```
asadmin reconfig --user joeuser --password secret --keepmanualchanges=true
server1
```

設定を変更してその変更を適用すると、上記に示した動的に設定可能なファイル内のすべての情報を含む新しい設定がメモリーに読み込まれます。

注 設定の変更を適用すると、サーバーインスタンスがエラーメッセージを表示する場合があります。たとえば、セキュリティがオンになっているときに変更を適用すると、設定が正しくない旨のエラーが表示されることがあります。

このエラーメッセージが表示された場合は、「Instance restart is required (インスタンスの再起動が必要です)」または「Server restart needed (サーバーの再起動が必要です)」などの再起動メッセージが表示されたらただちにインスタンスを再起動します。再起動が遅れると、同じエラーメッセージが繰り返し表示されることがあります。

設定ファイルの手動編集

管理インタフェースまたはコマンド行からサーバーインスタンスに変更を加えた場合、変更内容はメモリーに格納されます。変更内容を適用すると、この情報が内部で設定ファイルに書き込まれます。

手動で設定ファイルを編集してサーバーインスタンスに変更を加えた場合、ユーザーが変更内容を適用するまで、サーバーインスタンスは変更内容を認識しません。

このため、変更内容を適用せずに2種類の変更を加えるときは、どちらか一方の変更内容を確保し、もう一方を破棄する必要があります。このように、設定ファイルを手動で編集する場合、変更内容が上書きされることがあります。

ただし、loadbalancer.xml ファイルは例外です。このファイルは必ず手動で編集します。詳細については、[141 ページの「loadbalancer.xml」](#)を参照してください。

次に、手動で設定ファイルを編集する際の注意事項を示します。

- 変更内容を適用せずに 2 種類の変更を加えるときは、次のいずれかの方法に従ってください。
 - 管理インタフェースを使用して変更を適用します。この場合、手動の変更と管理インタフェースによる変更のどちらかを選択する必要があります。
 - `asadmin reconfig` コマンドを使用して変更を適用します。この場合、`--keepmanualchanges` オプションまたは `--discardmanualchanges` オプションを選択します。どちらのオプションも選択しなかった場合、変更内容は破棄されます。次に例を示します。

```
asadmin reconfig --user joeuser --password secret --keepmanualchanges=true
server1
```

この問題を避けるためには、変更を加えたらすぐに変更内容を適用するようにします。

- 変更が必要な場合には、`instance_dir/config/` ディレクトリにある設定ファイルのみを編集します。ファイルをサーバーのバックアップディレクトリにコピーしたり、タイムスタンプファイルを削除したりしないでください。タイムスタンプファイルがあると、手動による変更の上書きを防ぐことができます。なお、手動で変更を加える前に、設定に間違いがあった場合に備えてバックアップファイルを作成しておくことをお勧めします。
- 管理インタフェースを使用して `server.xml` の設定を変更するたびに、`sun-server_1_1.dtd` ファイルと比較して検証されます。ファイルの相互参照のチェックに関する追加の検証が行われます。手動で `server.xml` を編集する場合、必要なすべての設定が、正確に設定されているかどうかは、ユーザーの責任です。
- `server.xml` の `applications` サブ要素または `resources` サブ要素を手動で変更した場合、変更内容はすぐに適用されます。変更内容を明示的に適用する必要はありません。この場合も、変更内容があとで上書きされることはありませんが、設定が完全かつ正確であることを十分に確認してください。
- 手動で変更を加える場合、サーバーを再起動する前に変更内容を適用する必要があります。
- 管理サーバーは、再起動前の適用されていない変更を記憶しています。このため、管理サーバーが認識しているサーバーインスタンスの設定が実際のサーバーインスタンスの設定とは異なる場合があります。この問題を避けるため、管理サーバーを再起動する前に必ずすべてのサーバーインスタンスに変更を適用してください。

クラスタの再設定

サーバーインスタンスのクラスタの設定を変更するには、`cladmin` コマンドを使用します。このコマンドでは、クラスタに対して次の操作を実行できます。

- すべてのサーバーインスタンスの起動または停止
- アプリケーションの配備または配備取り消し
- JDBC リソースおよび接続プールの作成または削除
- 高可用性 HTTP セッションの持続性の設定

詳細については、『Sun ONE Application Server 管理者ガイド』を参照してください。

サーバー設定ファイル

`server.xml` ファイルには、ほとんどのサーバー設定情報が含まれています。このファイルのエンコーディングは、一般的な UNIX のテキストエディタとの互換性を維持するため、UTF-8 になっています。`server.xml` ファイルは、`instance_dir/config` ディレクトリにあります。スキーマファイル `sun-server_1_1.dtd` は、`server.xml` ファイルの書式と内容を定義します。

この章には、`server.xml` および `sun-server_1_1.dtd` ファイルについて説明する次の節があります。

- [sun-server_1_1.dtd ファイル](#)
- [server.xml ファイル内の要素](#)
- [全般的な要素](#)
- [リスナーサービス要素](#)
- [コンテナ要素](#)
- [可用性要素](#)
- [J2EE サービス要素](#)
- [Java 設定要素](#)
- [リソース要素](#)
- [アプリケーション要素](#)
- [ユーザーデータベースの選択](#)
- [Sun ONE LDAP スキーマ](#)
- [変数](#)
- [server.xml ファイルのサンプル](#)

注 仮想サーバーとサーバーインスタンスは別のものです。各サーバーインスタンスは、1 つ以上の仮想サーバーを含み、完全に個別のサーバーです。

sun-server_1_1.dtd ファイル

sun-server_1_1.dtd ファイルは、server.xml ファイルの構造を定義します。つまり、server.xml ファイル内に記述できるさまざまな要素と、その要素が持つことのできる属性を定義しています。sun-server_1_1.dtd ファイルは、`install_dir/lib/dtds` ディレクトリにあります。

注 sun-server_1_1.dtd ファイルを編集しないでください。このファイルの内容は、Sun ONE Application Server のバージョンの改訂にともなって変更されます。

注 sun-server_1_1.dtd インタフェースは、不確定です。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。

DTD ファイルおよび XML の全般的な情報については、次のサイトにある XML 仕様を参照してください。

<http://www.w3.org/TR/REC-xml>

DTD ファイルに定義された各要素 (対応する XML ファイル内に置かれている場合もある) には、次の要素が含まれています。

- サブ要素
- データ
- 属性

サブ要素

要素にはサブ要素を含めることができます。たとえば、次のファイルコードは `iioop-listener` 要素を定義しています。

```
<!ELEMENT iioop-listener (ssl?, property*)>
```

この `ELEMENT` タグは、`iioop-listener` 要素に `ssl` と `property` 要素をこの順番で含めることができることを示しています。

次の表に、サブ要素のサフィックス文字 (省略可能) によって決まる必要指定数、つまり指定可能なサブ要素の数を示します。

表 2-1 サブ要素のサフィックスの必要規則

サブ要素のサフィックス	必要指定数
<i>element</i> *	このサブ要素を含まないか、1 個以上含めることができる
<i>element</i> ?	このサブ要素を含まないか、1 個含めることができる
<i>element</i> +	このサブ要素を 1 個以上含まなければならない
<i>element</i> (サフィックスなし)	このサブ要素を 1 個だけ含まなければならない

要素にほかの要素を含めることができない場合は、カッコで囲まれた要素名のリストの代わりに、`EMPTY` または `(#PCDATA)` が表示されます。

データ

要素の中には、サブ要素の代わりに文字データを含むものもあります。これらの要素は、次の形式で定義されます。

```
<!ELEMENT element-name (#PCDATA)>
```

次に例を示します。

```
<!ELEMENT description (#PCDATA)>
```

`server.xml` ファイル内では、データ要素内の空白スペースはデータの一部として扱われます。そのため、データ要素で区切られたデータの前後には余分な空白がないようにする必要があります。次に例を示します。

```
<description>shopping cart bean</description>
```

属性

ATTLIST タグを持つ要素には属性 (名前と値のペア) が含まれています。次に例を示します。

```
<!ATTLIST iiop-listener      id          CDATA      #REQUIRED
                                address     CDATA      #REQUIRED
                                port        CDATA      "3700"
                                enabled     %boolean;  "true">
```

iiop-listener 要素には、id、address、port、または enabled 属性を含めることができます。

#REQUIRED ラベルは、値を指定する必要があることを示します。#IMPLIED ラベルは、その属性の指定は省略可能であり、Sun ONE Application Server がデフォルト値を生成することを示します。可能な場合は、"true" などの明示的なデフォルト値が示されます。

属性宣言は、属性のタイプを指定します。たとえば、CDATA は文字データ、%boolean は事前定義された列挙型を表します。

server.xml ファイル内の要素

この節では、server.xml ファイル内の XML 要素について説明します。要素は次のグループに分類されます。

- [全般的な要素](#)
- [リスナーサービス要素](#)
- [コンテナ要素](#)
- [J2EE サービス要素](#)
- [Java 設定要素](#)
- [リソース要素](#)
- [アプリケーション要素](#)

注 特に指定しない限り、サブ要素は各**サブ要素欄**に示されている順に定義する必要があります。

server.xml の要素のアルファベット順のリストについては、[付録 B 「サーバー設定要素のアルファベット順リスト」](#)を参照してください。

全般的な要素

全般的な要素は次のとおりです。

- `server`
- `property`
- `description`
- `admin-service`
- `server-instance`

server

サーバーを定義します。これはルート要素であり、`server.xml` ファイル内には `server` 要素が 1 つだけ存在します。

サブ要素

次の表は、`server` 要素のサブ要素を示します。

表 2-2 `server` のサブ要素

要素	必要数	説明
<code>http-service</code>	1 個のみ	HTTP サービスを定義する
<code>iiop-service</code>	1 個のみ	IIOP サービスを定義する
<code>admin-service</code>	0 または 1 個	実装されていない
<code>web-container</code>	1 個のみ	Web コンテナを設定する
<code>ejb-container</code>	1 個のみ	Enterprise JavaBeans™ (EJB™) コンテナを設定する
<code>mdb-container</code>	1 個のみ	メッセージ駆動型 Beans (MDB) コンテナを設定する
<code>jms-service</code>	1 個のみ	Java™ Message Service (JMS) プロバイダを設定する
<code>log-service</code>	1 個のみ	システムログサービスを設定する
<code>security-service</code>	1 個のみ	J2EE セキュリティサービスで必要となる情報を定義する
<code>transaction-service</code>	1 個のみ	Java Transaction Service を設定する
<code>java-config</code>	1 個のみ	Java™ 仮想マシン (JVM™) の設定を含む
<code>resources</code>	1 個のみ	設定されたリソースを含む

表 2-2 server のサブ要素 (続き)

要素	必要数	説明
applications	1 個のみ	配備された J2EE アプリケーション、J2EE モジュール、およびライフサイクルモジュールを含む
availability-service	0 または 1 個	高可用性機能を有効にする
property	0 または 1 個 以上	プロパティまたは変数を指定する

属性

次の表は、server 要素の属性について説明しています。

表 2-3 server の属性

属性	デフォルト値	説明
name	なし	サーバーインスタンスの名前を指定する
locale	オペレーティングシステムのデフォルト	(省略可能) サーバーインスタンスの言語を指定する
log-root	<i>instance_dir</i> /logs	(省略可能) サーバーインスタンスのログファイルを保存する位置を指定する。サーバーログが保存されるディレクトリには、サーバーで実行されているどのユーザーアカウントも書き込める必要がある。ログについての詳細は、「 log-service 」の説明を参照
application-root	<i>instance_dir</i> /applications	(省略可能) このサーバーインスタンス内に配備されたアプリケーションの保存されている場所への絶対パスを指定する
session-store	<i>instance_dir</i> /session-store	(省略可能) 非活性化された Beans が格納される、ファイルシステム内のディレクトリを指定する

property

プロパティ (server.xml 内で定義されて、obj.conf 内で参照される変数) を指定します。変数については、[101 ページの「変数」](#)を参照してください。

property は、次のいずれかまたは両方の親要素に設定情報を追加します。

- Sun ONE Application Server の省略可能な要素
- Sun ONE Application Server が認識していない、LDAP サーバーや Java クラスなどのシステムまたはオブジェクトに必要な要素

たとえば、auth-realm 要素には、property サブ要素を含めることができます。

```
<auth-realm name="file"
  classname="com.iplanet.ias.security.auth.realm.file.FileRealm">
  <property name="file" value="instance_dir/config/keyfile"/>
  <property name="jaas-context" value="fileRealm"/>
</auth-realm>
```

auth-realm 要素が使用するプロパティは、auth-realm 要素の name 属性によって決まります。file レルムは、file および jaas-context プロパティを使用します。その他のレルムは、別のプロパティを使用します。

サブ要素

次の表は、property 要素のサブ要素を示します。

表 2-4 property のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む

属性

次の表は、property 要素の属性を示します。

表 2-5 property の属性

属性	デフォルト値	説明
name	なし	プロパティまたは変数の名前を指定する
value	なし	プロパティまたは変数の値を指定する

description

親の要素を説明するテキストを含む

サブ要素

なし

属性

なし

admin-service

この要素は実装されていないので、使用できません。

server-instance

この要素は実装されていないので、使用できません。

リスナーサービス要素

リスナーサービス要素は次のとおりです。

- `http-service`
- `http-listener`
- `ssl`
- `mime`
- `acl`
- `virtual-server-class`
- `virtual-server`
- `http-qos`
- `auth-db`
- `iiop-service`
- `iiop-listener`
- `orb`

http-service

HTTP サービスを定義します。

この要素の属性に定義されるサービス品質機能についての詳細は、『Sun ONE Application Server Performance Tuning Guide』を参照してください。

サブ要素

次の表は、`http-service` 要素のサブ要素を示します。

表 2-6 `http-service` のサブ要素

要素	必要数	説明
<code>http-listener</code>	0 または 1 個以上	HTTP 待機ソケットを定義する
<code>mime</code>	0 または 1 個以上	MIME タイプを定義する
<code>acl</code>	0 または 1 個以上	ACL ファイルを参照する
<code>virtual-server-class</code>	0 または 1 個以上	仮想サーバークラスを定義する
<code>http-qos</code>	0 または 1 個	サービス品質パラメータを定義する
<code>property</code>	0 または 1 個以上	プロパティまたは変数を指定する

注 `http-listener`、`mime`、`acl`、および `virtual-server-class` 要素はどのような順番で記述してもかまいませんが、`http-qos` および `property` 要素はそれぞれ最後から 2 番目と最後に記述する必要があります。

属性
次の表は、`http-service` 要素の属性を示します。

表 2-7 `http-service` の属性

属性	デフォルト値	説明
<code>qos-metrics-interval-in-seconds</code>	30	(省略可能) トラフィックを測定する間隔を指定する
<code>qos-recompute-time-interval-in-millis</code>	100	(省略可能) すべてのサーバーエンティティの帯域幅を再計算する間隔を指定する
<code>qos-enabled</code>	true	(省略可能) サービス品質機能を有効にして、サーバー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ー統計を表示したりする。有効な値は <code>yes</code> 、 <code>no</code> 、 <code>on</code> 、 <code>off</code> 、 <code>1</code> 、 <code>0</code> 、 <code>true</code> 、 <code>false</code>

http-listener

HTTP 待機ソケットを定義します。

注 管理インターフェースを使用して安全なリスナーを作成する場合、セキュリティは自動的に `init.conf` 内でグローバルにオンになります。
`sever.xml` で安全なリスナーを手動で作成する場合は、`init.conf` ファイルの `Security` 指令を編集してセキュリティをオンにする必要があります。

サブ要素
次の表は、`http-listener` 要素のサブ要素を示します。

表 2-8 http-listener のサブ要素

要素	必要数	説明
ssl	0 または 1 個	SSL パラメータを定義する

属性

次の表は、http-listener 要素の属性を示します。

表 2-9 http-listener の属性

属性	デフォルト値	説明
id	なし	一意のリスナー名。http-listener の名前は数字で始めることはできない
address	なし	リスナーの IP アドレス。ドットで区切った形式または IPv6 表記法で指定できる。すべての IP アドレスを待機する any (INADDR_ANY の) を指定できる。ホスト名も指定できる
port	なし	リスナーが待機するポート番号。有効な値は 1 ～ 65535。UNIX の場合、ポート 1 ～ 1024 で待機するソケットを作成するには、スーパーユーザー特権が必要。SSL リスナーをポート 443 で待機するように設定するのが標準
family	inet	(省略可能) ソケットファミリのタイプ。有効値は、「inet」、「inet6」、および「nca」。IPv6 リスナーには「inet6」を使用する。この値が inet6 の場合、サーバーログでは、IPv4 アドレスの先頭に ::ffff: が付く。Solaris Network Cache および Accelerator で使用する場合は、nca を指定する
acceptor-threads	1	(省略可能) リスナーのアクセプタスレッドの数。通常はマシン内のプロセッサ数に一致。有効な値は 1 ～ 1024
blocking-enabled	false	(省略可能) リスナーおよび受け入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用すると、ベンチマークの結果が向上する可能性がある。有効な値は on、off、yes、no、1、0、true、false

表 2-9 http-listener の属性 (続き)

属性	デフォルト値	説明
security-enabled	false	<p>(省略可能) リスナーが SSL を実行するかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false。SSL2 または SSL3 のオンとオフを切り替えたり、ssl 要素を使って暗号化方式を設定したりできる</p> <p>init.conf ファイル内の Security 設定は、サーバーインスタンスで証明書を利用できるようにすることで、SSL をグローバルに有効または無効にする。そのため、init.conf の Security が on になっていることが必要。そうでないと、server.xml の security-enabled は機能しない。詳細については、第 3 章「init.conf の構文と使い方」を参照</p>
default-virtual-server	なし	この特定のリスナー用の、デフォルトの仮想サーバーの id 属性
server-name	なし	<p>サーバーがクライアントに送信する URL のホスト名部分に指定する名前をサーバーに通知する。これは、サーバーが自動的に生成する URL には影響するが、サーバーに格納されているディレクトリやファイルの URL には影響しない。サーバーがエイリアスを使っている場合、この名前はエイリアス名である必要がある</p> <p>コロンとポート番号を追加すると、そのポートは、サーバーがクライアントへ送信する URL で使用される</p>
enabled	true	(省略可能) リスナーをアクティブにするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

警告 ブロックモードソケットは、実際の配備では使わないようにしてください。ブロックモードソケットを使うと動的な再設定ができなくなり、サーバーがサービスを利用できなくなることがあります。

ssl

SSL (Secure Socket Layer) パラメータを定義します。

http-listener 要素の security-enabled 属性が on に設定されている場合は、ssl 要素が必要です。

ssl 要素は、http-listener または iiop-listener 要素内でのみ使用できます。

サブ要素

なし

属性

次の表は、ssl 要素の属性を示します。

表 2-10 ssl の属性

属性	デフォルト値	説明
cert-nickname	なし	証明書データベースまたは PKCS#11 トークン内でのサーバー証明書のニックネーム。証明書では、名前の形式は <i>tokenname: nickname</i> 。この属性では、名前の <i>tokenname:</i> の部分は省略可能
ssl2-enabled	false	(省略可能) SSL2 を有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false SSL2 と SSL3 の両方が仮想サーバーで有効になっている場合、サーバーは初めに SSL3 暗号化方式を試行する。SSL3 暗号化方式が失敗すると、SSL2 暗号化方式を試行する
ssl2-ciphers	なし	(省略可能) SSL2 暗号化方式に使用する値をコンマで区切って指定したリスト。有効にするにはプレフィックス + を付け、無効にするには - を付ける。たとえば、+rc4 のようになる。有効な値は rc4、rc4export、rc2、rc2export、idea、des、desede3

表 2-10 ssl の属性 (続き)

属性	デフォルト値	説明
ssl3-enabled	true	(省略可能) SSL3 を有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false。デフォルトは true SSL2 と SSL3 の両方が仮想サーバーで有効になっている場合、サーバーは初めに SSL3 暗号化方式を試行する。SSL3 暗号化方式が失敗すると、SSL2 暗号化方式を試行する
ssl3-tls-ciphers	なし	(省略可能) SSL3 暗号化方式に使用する値をコンマで区切って指定したリスト。有効にするにはプレフィックス + を付け、無効にするには - を付ける。たとえば、+rsa_des_sha のようになる。指定できる SSL3 値は rsa_rc4_128_md5、rsa_3des_sha、rsa_des_sha、rsa_rc4_40_md5、rsa_rc2_40_md5、rsa_null_md5。指定できる TLS 値は rsa_des_56_sha、rsa_rc4_56_sha
tls-enabled	true	(省略可能) TLS を有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false
tls-rollback-enabled	true	(省略可能) TLS ロールバックを有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false。Microsoft Internet Explorer 5.0 および 5.5 では、TLS ロールバックが有効になっている必要がある。詳細については、『Sun ONE Application Server 管理者ガイド』を参照
client-auth-enabled	false	(省略可能) ACL ベースのアクセス制御とは別に、すべての要求で SSL3 クライアント認証を実行するかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

mime

MIME タイプを定義します。

要求されたリソースの MIME タイプをサーバーが決定するには、`obj.conf` ファイルの `ObjectType` セクションにある `type-by-extension` 指令を呼び出す方法が最も一般的です。`type-by-extension` 関数は、`server` 要素で `mime` 要素が定義されていない場合は機能しません。

注	<code>mime.types</code> インタフェースは不確定です。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。
---	---

サブ要素

なし

属性

次の表は、`mime` 要素の属性を示します。

表 2-11 `mime` の属性

属性	デフォルト値	説明
<code>id</code>	なし	MIME タイプのリストの内部名。仮想サーバーが使用する MIME タイプを定義するために、 <code>virtual-server</code> 要素で使用される。MIME タイプ名は数字で始めることはできない
<code>file</code>	なし	MIME タイプのファイルの名前。このファイルの書式設定の詳細については、 第 4 章「MIME タイプ」 を参照

acl

ACL ファイルを参照します。

注	<code>ACL</code> ファイルインタフェースは、不確定です。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。
---	---

サブ要素

なし

属性

次の表は、acl 要素の属性を示します。

表 2-12 acl の属性

属性	デフォルト値	説明
id	なし	ACL ファイルのリストの内部名。仮想サーバーで使用される ACL ファイルを定義するために、virtual-server 要素で使用される。ACL ファイル名は数字で始めることはできない
file	なし	ACL ファイル。各 ACL ファイルの名前は、一意であることが必要。ACL ファイルの書式については、『Sun ONE Application Server セキュリティ管理者ガイド』を参照

virtual-server-class

仮想サーバークラスを定義します。

サブ要素

次の表は、virtual-server-class 要素のサブ要素を示します。

表 2-13 virtual-server-class のサブ要素

要素	必要数	説明
virtual-server	0 または 1 個以上	仮想サーバーを定義する
http-qos	0 または 1 個	サービス品質パラメータを定義する
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、virtual-server-class 要素の属性を示します。

表 2-14 virtual-server-class の属性

属性	デフォルト値	説明
id	なし	仮想サーバークラスの ID。これは、特定の仮想サーバークラスの検索を可能にする一意の ID。仮想サーバークラス ID は数字で始めることはできない

表 2-14 virtual-server-class の属性 (続き)

属性	デフォルト値	説明
config-file	なし	(省略可能) この仮想サーバークラス用の obj.conf ファイルの名前。virtual-server 要素内でオーバーライドすることができる
default-object	デフォルト	(省略可能) obj.conf ファイルから読み込まれるどのオブジェクトがデフォルトなのかをサーバーに指示する。 virtual-server 要素内でオーバーライドすることができる デフォルトのオブジェクトは、仮想サーバー用のすべての名前変換 (NameTrans) 指令を持っているとみなされる。また、デフォルトオブジェクトに設定されているサーバーの動作は、仮想サーバークラス全体に影響する 存在しないオブジェクトを指定した場合、サーバーは、クライアントがドキュメントを取得しようとするまではエラーを報告しない。管理インタフェースでは、デフォルトが default という名前のオブジェクトであるとみなされる。管理インタフェースを使う (または使う予定の) 場合は、この規則に従う必要がある
accept-language	false	(省略可能) true の場合、サーバーは Accept-Language ヘッダーを解析し、クライアントが受け入れることのできる言語に基づいて適切な言語のバージョンを送信する。サーバーが複数の言語をサポートする場合にのみ、この値を true に設定する。 virtual-server 要素内でオーバーライドすることができる。 有効な値は on、off、yes、no、1、0、true、false。 virtual-server 要素内でオーバーライドすることができる
enabled	true	(省略可能) 仮想サーバークラスをアクティブにするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

virtual-server

仮想サーバーを定義します。仮想サーバー (仮想ホスト) は、特定の URL 用にコンテンツを提供する仮想 Web サーバーです。複数の仮想サーバーが、同一または異なったホスト名、ポート番号、IP アドレスなどを使ってコンテンツを提供できます。HTTP サービスは、受信する Web 要求を URL に基づいて異なった仮想サーバーに送信できます。

Sun ONE Application Server の最初のインストール時に、デフォルトの仮想サーバーが作成されます。作成する新しい [http-listener](#) ごとに、デフォルトの仮想サーバーを割り当てることもできます。

注	仮想サーバーとサーバーインスタンスは別のものです。各サーバーインスタンスは、1 つ以上の仮想サーバーを含み、完全に個別のサーバーです。
---	---

Sun ONE Application Server は、要求を処理する前に、リスナーを介して要求を受け取り、要求を正しい仮想サーバーに転送する必要があります。仮想サーバーは、次のように決定されます。

- リスナーにデフォルトの仮想サーバーのみが設定されている場合、その仮想サーバーが選択される
- リスナーに複数の仮想サーバーが設定されている場合、要求 Host ヘッダーは仮想サーバーの host 属性と比較される。Host ヘッダーが存在しないか、host 属性が一致しない場合、リスナーのデフォルトの仮想サーバーが選択される

仮想サーバーが SSL リスナーに設定されている場合、起動時にその hosts 属性が証明書のサブジェクトのパターンと比較されます。それらが一致しない場合、警告が生成され、サーバーログに書き込まれます。

サブ要素

次の表は、virtual-server 要素のサブ要素を示します。

表 2-15 virtual-server のサブ要素

要素	必要数	説明
http-qos	0 または 1 個	サービス品質パラメータを定義する
auth-db	0 または 1 個以上	J2EE アプリケーションに適用できない、仮想サーバー用のユーザーデータベースを定義する
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、virtual-server 要素の属性を示します。

表 2-16 virtual-server の属性

属性	デフォルト値	説明
id	なし	仮想サーバー ID。特定の仮想サーバーの検索を可能にする一意の ID。obj.conf ファイル内の変数 \$id として参照することもできる。仮想サーバー ID は数字で始めることはできない

表 2-16 virtual-server の属性 (続き)

属性	デフォルト値	説明
http-listeners	なし	(省略可能) 仮想サーバーが使用する接続を指定する、http-listener id のリスト。 各 id はコンマで区切られる。 http-listener の default-virtual-server ではない virtual-server の場合にのみ必要
default-web-module	なし	(省略可能) この仮想サーバー用のデフォルトの web-module で、この仮想サーバーに 配備されたその他の Web モジュールで解決 できないすべての要求に応答する この属性が空になっている場合、この仮想 サーバーに割り当てられた context-root 属性が空になっている web-module が使わ れる。この仮想サーバーに割り当てられた web-module で context-root 属性が空に なっているものがない場合、システムのデ フォルトの Web モジュールが使われる
config-file	<i>virtual_server_name</i> -obj.conf	(省略可能) この仮想サーバー用の obj.conf ファイルの名前。 virtual-server-class 要素の値をオー バーライドすることがある
default-object	デフォルト	(省略可能) obj.conf ファイルから読み込 まれるどのオブジェクトがデフォルトなの かをサーバーに指示する。 virtual-server-class 要素の値をオー バーライドすることがある デフォルトのオブジェクトは、仮想サー バー用のすべての名前変換 (NameTrans) 指 令を持っているとみなされる。また、デ フォルトオブジェクトで設定されている サーバーの動作は、サーバー全体に影響す る。デフォルト値は default 存在しないオブジェクトを指定した場合、 サーバーは、クライアントがドキュメント を取得しようとするまではエラーを報告し ない。管理インタフェースでは、デフォル トが default という名前のオブジェクトで あるとみなされる。管理インタフェースを 使う (または使う予定の) 場合は、この規則 に従う必要がある

表 2-16 virtual-server の属性 (続き)

属性	デフォルト値	説明
hosts	なし	仮想サーバーを選択するために Host 要求ヘッダーに指定できる値のリスト。各値は、コンマで区切られる。同じ http-listener に設定された各 virtual-server は、そのリスナーに対して一意の hosts 値を持っていることが必要
mime	なし	仮想サーバーが使用する、mime 要素の id
state	on	(省略可能) virtual-server をアクティブ (on) または非アクティブ (off、disabled) にするかどうかを決定する。デフォルトは on (アクティブ)。非アクティブの場合、virtual-server は要求を処理しない virtual-server が disabled になっている場合、グローバルサーバー管理者だけが on にすることができる
acls	なし	(省略可能) acl 要素の 1 つ以上の id 属性。複数指定する場合は、コンマで区切って指定する。仮想サーバーが使用する ACL ファイルを指定する
accept-language	false	(省略可能) true の場合、サーバーは Accept-Language ヘッダーを解析し、クライアントが受け入れることのできる言語に基づいて適切な言語のバージョンを送信する。サーバーが複数の言語をサポートする場合にのみ、この値を on に設定する。有効な値は on、off、yes、no、1、0、true、false。virtual-server-class 要素の値をオーバーライドすることがある
log-file	server 要素の log-root 属性で指定されたディレクトリにある server.log	(省略可能) この仮想サーバーのログメッセージを、サーバーログとは別のログファイルに書き込む。仮想サーバーログが保存されるファイルとディレクトリには、サーバーで実行されているどのユーザーアカウントも書き込めることが必要。ログについての詳細は、「log-service」の説明を参照

プロパティ

次の表は、virtual-server 要素のプロパティを示します。

表 2-17 virtual-server のプロパティ

プロパティ	デフォルト値	説明
sso-enabled	true	true の場合、同じレルムに設定されたこの仮想サーバー上の Web アプリケーションに対してシングルサインオンを有効にする。false の場合、この仮想サーバーに対してシングルサインオンは無効になり、ユーザーは仮想サーバー上のアプリケーションごとに個別に認証を行う必要がある。有効な値は on、off、yes、no、1、0、true、false
sso-max-inactive-seconds	300	クライアントが活動を停止後、何秒後にユーザーのシングルサインオンレコードをパージ可能にするかを指定する。シングルサインオンは同一仮想サーバー上の複数のアプリケーションに適用されるので、これらのアプリケーションのいずれかにアクセスすることでシングルサインオンの記録は有効なまま確保される。値を大きくすればユーザーのシングルサインオンの持続時間が長くなるが、サーバー上のメモリー消費量も増える
sso-reap-inaterval-seconds	60	有効期限が切れたシングルサインオンレコードのパージを行う間隔を指定する

http-qos

http-service、virtual-server-class、または virtual-server 要素のサービス品質パラメータを定義します。

[http-service](#) 要素の属性は、サービス品質機能をアクティブにします。

詳細については、『Sun ONE Application Server Performance Tuning Guide』を参照してください。

サブ要素

なし

属性

次の表は、http-qos 要素の属性を示します。

表 2-18 http-qos の属性

属性	デフォルト値	説明
bandwidth-limit	なし	enforce-bandwidth-limit が true の場合に必須。server、virtual-server-class、または virtual-server の帯域幅の最大値 (単位は秒あたりのバイト数)
enforce-bandwidth-limit	false	(省略可能) 帯域幅制限を適用するかどうかを指定する。有効な値は yes、no、on、off、1、0、true、false
connection-limit	なし	enforce-connection-limit が true の場合に必須。server、virtual-server-class、または virtual-server の最大同時接続数
enforce-connection-limit	false	(省略可能) 接続制限を適用するかどうかを指定する。有効な値は yes、no、on、off、1、0、true、false

auth-db

virtual-server が使用するユーザーデータベースを定義します。J2EE アプリケーションに適用できません。

指定された仮想サーバーのユーザーデータベースを選択する方法については、98 ページの「ユーザーデータベースの選択」を参照してください。

注	ユーザーデータベースはサーバー自体のセキュリティのみに適用します。J2EE アプリケーションおよびモジュールのセキュリティには関係ありません。詳細は、『Sun ONE Application Server セキュリティ管理者ガイド』を参照してください。
---	--

サブ要素

なし

属性

次の表は、auth-db 要素の属性を示します。

表 2-19 auth-db の属性

属性	デフォルト値	説明
id	なし	仮想サーバーの ACL ファイル内のユーザーデータベース名。ユーザーデータベース名は数字で始めることはできない
database	なし	dbswitch.conf ファイル内のユーザーデータベース名
basedn	なし	(省略可能) dbswitch.conf ファイル内の基本 DN 検索をオーバーライドする。ただし、basedn 値は、dbswitch.conf エントリからの基本 DN 値と相対関係になる
certmaps	なし	(省略可能) LDAP エントリマッピング(certmap.conf で定義されている)に使う証明書を指定する。存在しない場合、すべてのマッピングが使われる。certmap.conf 内のマッピングに基づくすべての検索は、virtual-server の最終の基本 DN と相対関係になる

iiop-service

IIOP サービスを定義します。

サブ要素

次の表は、iiop-service 要素のサブ要素を示します。

表 2-20 iiop-service のサブ要素

要素	必要数	説明
orb	1 個のみ	ORB を設定する
ssl-client-config	0 または 1 個	ORB の SSL パラメータを定義する
iiop-listener	0 または 1 個以上	IIOP 待機ソケットを定義する

属性

なし

orb

ORB を設定します。

送信接続の SSL を有効にするには、`ssl-client-config` サブ要素を親 `iiop-service` 要素に含めます。

サブ要素

次の表は、orb 要素のサブ要素を示します。

表 2-21 orb のサブ要素

要素	必要数	説明
<code>property</code>	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、orb 要素の属性を示します。

表 2-22 orb の属性

属性	デフォルト値	説明
<code>message-fragment-size</code>	1024	(省略可能) このバイト数より大きいサイズの GIOPv1.2 メッセージは断片化される
<code>steady-thread-pool-size</code>	10	(省略可能) ORB 内のワーカースレッドの最小数
<code>max-thread-pool-size</code>	200	(省略可能) ORB 内のワーカースレッドの最大数
<code>idle-thread-timeout-in-seconds</code>	300	(省略可能) この時間が経過すると、アイドル状態のワーカースレッドがプールから削除される
<code>max-connections</code>	1024	(省略可能) すべての IIOP リスナー上の受信接続の最大数。有効な値は整数
<code>log-level</code>	<code>log-service</code> 要素の <code>level</code> 属性の値	(省略可能) この要素がサーバーログに記録する、メッセージのタイプを制御する。詳細は、 <code>log-service</code> 要素の <code>level</code> 属性の説明を参照
<code>monitoring-enabled</code>	false	(省略可能) ORB の監視を有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

ssl-client-config

ORB が送信 SSL 接続を確立してクライアントとして動作するときの、SSL パラメータを定義します。

サブ要素

次の表は、ssl-client-config 要素のサブ要素を示します。

表 2-23 ssl-client-config のサブ要素

要素	必要数	説明
ssl	1 個のみ	SSL パラメータを定義する

属性

なし

iiop-listener

IIOP 待機ソケットを定義します。

このリスナーの SSL を有効にするには、ssl サブ要素を含めます。

注	管理インターフェースを使用して安全なリスナーを作成する場合、セキュリティは自動的に <code>init.conf</code> 内でグローバルにオンになります。 <code>sever.xml</code> で安全なリスナーを手動で作成する場合は、 <code>init.conf</code> ファイルの <code>Security</code> 指令を編集してセキュリティをオンにする必要があります。
---	---

サブ要素

次の表は、iiop-listener 要素のサブ要素を示します。

表 2-24 iiop-listener のサブ要素

要素	必要数	説明
ssl	0 または 1 個	SSL パラメータを定義する
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、iiop-listener 要素の属性を示します。

表 2-25 iiop-listener の属性

属性	デフォルト値	説明
id	なし	リスナー名。iiop-listener 名は数字で始めることはできない
address	なし	リスナーの IP アドレス。ドットで区切ったペア、または IPv6 表記で指定する
port	3700 (最初のサーバー インスタンス用)	(省略可能) リスナーを作成するポート番号。有効な値は 1 ～ 65535。UNIX の場合、ポート 1 ～ 1024 で待機するソケットを作成するには、スーパーユーザー特権が必要
enabled	true	(省略可能) リスナーをアクティブにするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

コンテナ要素

コンテナ設定要素は次のとおりです。

- `web-container`
- `ejb-container`
- `mdb-container`

web-container

Web コンテナを設定します。

サブ要素

次の表は、web-container 要素のサブ要素を示します。

表 2-26 web-container のサブ要素

要素	必要数	説明
<code>session-config</code>	0 または 1 個	Web コンテナのセッション設定情報を指定する
<code>property</code>	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、web-container 要素の属性を示します。

表 2-27 web-container の属性

属性	デフォルト値	説明
<code>monitoring-enabled</code>	<code>false</code>	(省略可能) web コンテナの監視を有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false
<code>log-level</code>	<code>log-service</code> 要素の level 属性の値	(省略可能) この要素がサーバーログに記録する、メッセージのタイプを制御する。詳細は、 <code>log-service</code> 要素の level 属性の説明を参照 ServerContext.log メッセージは、デフォルトで INFO レベルのログに記録される

ejb-container

EJB コンテナを設定します。ステートレスセッション Bean は、プールに保持されます。ステートフルセッション Beans は、セッション親和性があり、キャッシュされます。データベースの主キーに関連付けられたエンティティ Bean もキャッシュされます。主キーにまだ関連付けられていないエンティティ Bean は、プールに保持されます。プーリングされたエンティティ Bean は、ejbCreate() とファインダーメソッドを実行するために使用されます。

サブ要素

次の表は、ejb-container 要素のサブ要素を示します。

表 2-28 ejb-container のサブ要素

要素	必要数	説明
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、ejb-container 要素の属性を示します。

表 2-29 ejb-container の属性

属性	デフォルト値	説明
steady-pool-size	32	(省略可能) プールに保持される Beans の初期数および最小数を指定する。0 以上で、max-pool-size 未満である必要がある Bean インスタンスは、プールから削除され、使用後に戻される。プールはこのサイズを保つために、定期的に補充またはクリーンアップされる ステートレスセッション Beans とエンティティ Beans に適用される

表 2-29 ejb-container の属性 (続き)

属性	デフォルト値	説明
pool-resize-quantity	16	<p>(省略可能) 次の条件を満たす Beans 数を指定する</p> <ul style="list-style-type: none"> • 利用できる Bean がプールにないときに要求を受信した場合に作成される (max-pool-size の制限を受ける) • pool-idle-timeout-in-seconds に指定された時間を過ぎ、クリーナスレッドが未使用のインスタンスを削除すると、削除される <p>0 以上で、max-pool-size 未満である必要がある。プールのサイズが steady-pool-size を下回ることはない</p> <p>ステートレスセッション Beans とエンティティ Beans に適用される</p>
max-pool-size	64	<p>(省略可能) クライアントの要求に対応するために作成できる Beans の最大数を指定する。値が 0 の場合は、プールのサイズに制限がないことを示す</p> <p>ステートレスセッション Beans とエンティティ Beans に適用される</p>
cache-resize-quantity	16	<p>(省略可能) 次の条件を満たす Beans 数を指定する</p> <ul style="list-style-type: none"> • 利用できる Beans がプールにないときに要求を受信した場合に作成される (max-cache-size の制限を受ける) • cache-idle-timeout-in-seconds に指定された時間を過ぎ、クリーナスレッドが未使用のインスタンスを削除するか、あるいはキャッシュサイズが max-cache-size を超える場合に非活性化される <p>1 以上で、max-cache-size 未満である必要がある</p> <p>ステートフルセッション Beans およびエンティティ Beans に適用される</p>

表 2-29 ejb-container の属性 (続き)

属性	デフォルト値	説明
max-cache-size	512	(省略可能) キャッシュに存在できる Beans の最大数を指定する。値が 0 の場合は、キャッシュのサイズに制限がないことを示す ステートフルセッション Beans およびエンティティ Beans に適用される
pool-idle-timeout-in-seconds	600	(省略可能) Bean がプールにアイドル状態のままでいられる最長時間を指定する。この時間を過ぎると、プールはこの Bean を削除する。0 の値を指定すると、アイドル状態の Beans がプールにいつまでも残ることができる ステートレスセッション Beans とエンティティ Beans に適用される
cache-idle-timeout-in-seconds	600	(省略可能) Bean がキャッシュにアイドル状態のままでいられる最長時間を指定する。この時間を過ぎると、コンテナはこの Bean を非活性化できる。値に 0 を指定すると、その Beans は非活性化の対象にならない ステートフルセッション Beans およびエンティティ Beans に適用される
removal-timeout-in-seconds	5400	(省略可能) Bean が、非活性化されたままでいられる時間 (セッションストアから削除されるまで) を指定する。0 の値を指定すると、コンテナは非活性化されている Beans を自動的に削除しない removal-timeout-in-seconds を cache-idle-timeout-in-seconds より短く、または同じに設定すると、Beans は非活性化されずにただちに削除される server 要素の session-store 属性によって、セッションストアの位置が決まる ステートフルセッション Beans およびエンティティ Beans に適用される

表 2-29 ejb-container の属性 (続き)

属性	デフォルト値	説明
victim-selection-policy	nru	(省略可能) エンティティ Beans またはステートフルセッション Beans を非活性化するための選択方法を指定する。有効な値は fifo、lru、および nru <ul style="list-style-type: none">• fifo を指定すると、最も古いインスタンスが選択される• lru を指定すると、最近アクセスしたインスタンスが選択される• nru を指定すると、最近使っていないインスタンスが選択される
commit-option	B	(省略可能) エンティティ Beans に使うコミットオプションを決定する。有効な値は B または C
log-level	log-service 要素の level 属性の値	(省略可能) この要素がサーバーログに記録する、メッセージのタイプを制御する。詳細は、log-service 要素の level 属性の説明を参照
monitoring-enabled	false	(省略可能) EJB コンテナの監視を有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

mdb-container

メッセージ駆動型 Beans (MDB) コンテナを設定します。

サブ要素

次の表は、mdb-container 要素のサブ要素を示します。

表 2-30 mdb-container のサブ要素

要素	必要数	説明
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、mdb-container 要素の属性を示します。

表 2-31 mdb-container の属性

属性	デフォルト値	説明
steady-pool-size	10	(省略可能) プールに保持される Beans の初期数および最小数を指定する
pool-resize-quantity	2	(省略可能) プールが空のときに要求を受信した場合に作成される Beans の数 (max-pool-size の制限を受ける)、または idle-timeout-in-seconds 以上の間アイドル状態の場合に削除する Beans の数を指定する
max-pool-size	60	(省略可能) クライアントの要求に対応するために作成できる Beans の最大数を指定する
idle-timeout-in-seconds	600	(省略可能) Bean がプールにアイドル状態のままではいられる最長時間を指定する。この時間が経過すると、Bean は削除される
log-level	log-service 要素の level 属性の値	(省略可能) この要素がサーバーログに記録する、メッセージのタイプを制御する。詳細は、log-service 要素の level 属性の説明を参照
monitoring-enabled	false	(省略可能) メッセージ駆動型 Beans (MDB) コンテナの監視を有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

プロパティ

次の表は、mdb-container 要素のプロパティを示します。

表 2-32 mdb-container のプロパティ

プロパティ	デフォルト値	説明
cmt-max-runtime-exceptions	1	コンテナ管理のトランザクションを使用するときに、メッセージ駆動型 Beans の onMessage() メソッドから最高何回 RuntimeExceptions がスローされるかを指定する。推奨されていない

表 2-32 mdb-container のプロパティ (続き)

プロパティ	デフォルト値	説明
reconnect-enabled	true	<p>true の場合、接続が切断されると、MDB コンテナが JMS プロバイダに自動的に再接続を試みる</p> <p>接続が切断されると、メッセージの処理段階によっては、onMessage() メソッドを完了させることができないか、トランザクションが JMS 例外によりロールバックされることがある。MDB コンテナが再度接続を確立すると、JMS メッセージの再配信セマンティックスが適用される</p>
reconnect-delay-in-seconds	60	再接続の試みの遅延間隔を指定する
reconnect-max-retries	60	再接続の試みの最大数を指定する

可用性要素

可用性要素は次のとおりです。

- [availability-service](#)
- [persistence-store](#)
- [session-config](#)
- [session-manager](#)
- [manager-properties](#)
- [store-properties](#)
- [session-properties](#)

availability-service

Sun ONE 高可用性データベース (HADB) の HTTP セッション持続や Web アプリケーションのシングルサインオンといった高可用性機能を有効にします。

サブ要素

次の表は、availability-service 要素のサブ要素を示します。

表 2-33 availability-service のサブ要素

要素	必要数	説明
persistence-store	0 または 1 個	HADB を設定する
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、availability-service 要素の属性を示します。

表 2-34 availability-service の属性

属性	デフォルト値	説明
availability-enabled	false	<p>(省略可能) true の場合、Sun ONE 高可用性データベース (HADB) の HTTP セッション持続や Web アプリケーションのシングルサインオンといった高可用性機能が有効になる</p> <p>有効な値は on、off、yes、no、1、0、true、false</p> <p>可用性の設定は、サーバーインスタンスに配備されたすべてのアプリケーションに無条件に適用される。HTTP セッション持続用に個別に設定されたものも対象となる。動作の一貫性のためには、クラスタ内のすべてのインスタンスに同一の可用性値を設定する</p> <p>可用性サービスの設定も、シングルサインオンが設定されている場合、その可用性に影響を及ぼす</p>

注	セッション持続の設定を行う場合、availability-enabled の値を true に設定するだけでは不十分です。その他の可用性要素も設定する必要があります。
---	---

可用性サービスが無効になっている場合、HTTP セッション持続の高可用性は得られません。つまり、`persistence-type=memory` となります。

可用性サービスが有効になっている場合、デフォルトですべてのアプリケーションに対して HTTP セッション持続の高可用性が有効になります。`server.xml` ファイルか `sun-web.xml` ファイルで特別な設定を加えていない場合、デフォルトのセッション持続設定は次のとおりです。

```
persistence-type=ha
persistenceFrequency=time-based
persistenceScope=session
```

`server.xml` 内にさまざまな可用性要素を設定することにより、すべてのアプリケーションについて、このデフォルト設定を上書きできます。`sun-web.xml` では、特定のアプリケーションについて、このデフォルト設定を上書きできます。詳細については、『Sun ONE Application Server Developer's Guide to Web Applications』を参照してください。

persistence-store

Sun ONE 高可用性データベース (HADB) を設定します。

サブ要素

次の表は、persistence-store 要素のサブ要素を示します。

表 2-35 persistence-store のサブ要素

要素	必要数	説明
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

なし

プロパティ

次の表は、persistence-store 要素のプロパティを示します。

表 2-36 persistence-store のプロパティ

プロパティ	デフォルト値	説明
store-pool-jndi-name	jdbc/hastore	セッション持続用として HADB 接続に使用される jdbc-resource の jndi-name を指定する HADB 用の接続プールならびに JDBC リソースの詳細な設定方法については、『Sun ONE Application Server インストールガイド』と『Sun ONE Application Server 管理者ガイド』を参照

session-config

Web コンテナ全体のセッション設定情報を指定します。個々の Web アプリケーションの sun-web.xml ファイル内の対応する要素を使って、これらの設定を上書きできます。

サブ要素

次の表は、session-config 要素のサブ要素を示します。

表 2-37 session-config のサブ要素

要素	必要数	説明
session-manager	0 または 1 個	セッションマネージャの設定情報を指定する

表 2-37 session-config のサブ要素 (続き)

要素	必要数	説明
<code>session-properties</code>	0 または 1 個	セッションのプロパティを指定する

属性
なし

session-manager

セッションマネージャの情報を指定します。

注 セッションマネージャインタフェースは不確定なインタフェースです。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。

サブ要素

次の表は、session-manager 要素のサブ要素を示します。

表 2-38 session-manager のサブ要素

要素	必要数	説明
<code>manager-properties</code>	0 または 1 個	セッションマネージャのプロパティを指定する
<code>store-properties</code>	0 または 1 個	セッションの持続性 (格納) のプロパティを指定する

属性

次の表は、session-manager 要素の属性を示します。

表 2-39 session-manager の属性

属性	デフォルト値	説明
<code>persistence-type</code>	<code>memory</code>	(省略可能) セッションの持続性メカニズムを指定する。有効な値は <code>memory</code> 、 <code>file</code> 、および <code>ha</code> セッションの持続性が必要な本稼働環境では、 <code>ha</code> を使用する <code>custom</code> 値は実装されないので使用できない

注

セッション持続の設定を行う場合、Web コンテナの `persistence-type` の値を `ha` に設定するだけでは不十分です。[availability-service](#) も有効にする必要があります。セッション持続の詳細は、『Sun ONE Application Server Developer's Guide to Web Applications』を参照してください。

manager-properties

セッションマネージャのプロパティを指定します。

サブ要素

次の表は、`manager-properties` 要素のサブ要素を示します。

表 2-40 `manager-properties` のサブ要素

要素	必要数	説明
property	0 または 1 個以上	名前と値を持つプロパティを指定する

属性

なし

プロパティ

次の表は、`manager-properties` 要素のプロパティを示します。

表 2-41 manager-properties のプロパティ

プロパティ名	デフォルト値	説明
reapIntervalSeconds	60	<p>期限切れのセッションをチェックする周期を秒単位で指定する</p> <p><code>session-manager</code> 要素の <code>persistence-type</code> 属性の値が <code>file</code> または <code>ha</code> の場合、<code>maxSessions</code> の値より大きくなるとセッションは非活性化される</p> <p><code>persistenceFrequency</code> の値が <code>time-based</code> に設定されている場合、アクティブなセッションはこの間隔で格納される</p> <p>セッションデータが変更される頻度より小さい値を設定する必要がある。たとえば、頻繁にアクセスされる Web サイト上のヒットカウンタサーブレットの場合、この値をできるだけ小さくする必要がある (1 秒)。そうしないと、サーバーを再起動するたびに最後の数回分のヒットが失われる可能性がある</p>
maxSessions	-1	<p>キャッシュ内に格納できるセッションの最大数を指定する。制限しない場合は -1 (デフォルト値)。その後、新しいセッションの作成を試行すると、<code>IllegalStateException</code> がスローされる</p> <p><code>session-manager</code> 要素の <code>persistence-type</code> 属性の値が <code>file</code> または <code>ha</code> の場合、この最大値を超えると、セッションマネージャによりセッションは非活性化され、持続ストアに格納される</p>
sessionFilename	なし。再起動すると、その前の状態は破棄される	<p>アプリケーションの再起動の間にセッション状態を保持するディレクトリの絶対パスまたは相対パスを指定する (状態の保持が可能な場合)。相対パスは、この Web アプリケーションの一時ディレクトリを基準とした場所を示す</p> <p><code>session-manager</code> 要素の <code>persistence-type</code> 属性が <code>memory</code> の場合にのみ設定可能</p>

表 2-41 manager-properties のプロパティ (続き)

プロパティ名	デフォルト値	説明
persistenceFrequency	web-method	セッションの状態が格納される頻度を指定する。有効な値は次のとおり <ul style="list-style-type: none">web-method - セッションの状態は、各 Web 要求の終了時、クライアントへ応答を返す前に格納される。このモードでは、障害が発生した場合にセッションの状態を完全に更新できる可能性が高いtime-based - セッションの状態は、reapIntervalSeconds で設定された頻度でバックグラウンドで格納される。このモードでは、セッションの状態を完全に更新できる可能性がやや低い。ただし、要求のたびに状態が格納されないので、パフォーマンスの大幅な向上が見込める <p>session-manager 要素の persistence-type 属性が ha の場合にのみ設定可能</p>

store-properties

セッションの持続性 (格納) のプロパティを指定します。

サブ要素

次の表は、store-properties 要素のサブ要素を示します。

表 2-42 store-properties のサブ要素

要素	必要数	説明
property	0 または 1 個以上	名前と値を持つプロパティを指定する

属性

なし

プロパティ

次の表は、store-properties 要素のプロパティを示します。

表 2-43 store-properties のプロパティ

プロパティ名	デフォルト値	説明
directory	<i>instance_dir</i> /generated /jsp/j2ee-apps/ <i>appname</i> / <i>appname_war</i>	個々のセッションファイルが書き込まれるディレクトリの絶対パスまたは相対パスを指定する。相対パスは、この Web アプリケーションの一時作業ディレクトリを基準とした場所を示す session-manager 要素の <code>persistence-type</code> 属性が <code>file</code> の場合にのみ設定可能
persistenceScope	セッション	格納されるセッションの状態の量を指定する。有効な値は次のとおり <ul style="list-style-type: none"> • <code>session</code> - 毎回セッションの状態全体が格納される。このモードでは、セッションデータを分散可能な Web アプリケーション用に正しく格納できる可能性が高い • <code>modified-session</code> - 変更された場合、セッションの状態全体が格納される。 <code>HttpSession.setAttribute()</code> または <code>HttpSession.removeAttribute()</code> が呼び出されると、セッションが変更されたと見なされる。属性の変更のたびに <code>setAttribute()</code> を呼び出す必要がある。この要件は J2EE 仕様には記載されていないが、このモードを正常に機能させるために必須 • <code>modified-attribute</code> - 変更されたセッション属性だけが格納される。このモードを正常に機能させるためには、この表の直後に説明されているガイドラインに従う必要がある session-manager 要素の <code>persistence-type</code> 属性が <code>ha</code> の場合にのみ設定可能

`persistenceScope` 格納プロパティの値が `modified-attribute` に設定されている場合、Web アプリケーションは次のガイドラインに従う必要があります。

- セッションの状態を変更するたびに `setAttribute()` を呼び出す
- 属性間にクロスリファレンスが存在しないことを確認する。各属性キーの下オブジェクトグラフは直列化され、個別に格納される。個々のキーの下オブジェクト間にオブジェクトクロスリファレンスが存在する場合、これらは正常に直列化または直列化解除されない

- 複数の属性、少なくとも読み取り専用属性と変更可能属性にセッションを分散させる

注	セッション持続範囲 <code>modified-attribute</code> は、本稼働環境で動作することが保証されていません。この持続範囲で、予想されるピーク負荷時の Web コンテナのパフォーマンスと安定性を評価する必要があります。例外が記録された場合、または応答時間が長すぎる場合は、この持続範囲を本稼働環境では使用しないでください。代わりに、 <code>session</code> または <code>modified-session</code> を使用してください。
---	--

session-properties

セッションのプロパティを指定します。

サブ要素

次の表は、`session-properties` 要素のサブ要素を示します。

表 2-44 <code>session-properties</code> のサブ要素		
要素	必要数	説明
<code>property</code>	0 または 1 個以上	名前と値を持つプロパティを指定する

属性
なし

プロパティ

次の表は、`session-properties` 要素のプロパティを示します。

表 2-45 session-properties のプロパティ

プロパティ名	デフォルト値	説明
timeoutSeconds	600	<p>この Web モジュールで作成されたすべてのセッションについて、非アクティブな状態が継続する最長時間のデフォルトの値を秒単位で指定する。0 またはそれ以下の値を指定した場合、この Web モジュールのセッションは期限切れになることがない</p> <p>web.xml ファイルに session-timeout 要素を指定した場合、session-timeout 値は、timeoutSeconds 値に優先して使用される。session-timeout も timeoutSeconds も指定しなかった場合、timeoutSeconds のデフォルトの値が使用される</p> <p>web.xml には session-timeout 要素を秒単位ではなく分単位で指定する</p>
enableCookies	true	true を指定すると、セッションの追跡に cookie が使用される
enableURLRewriting	true	URL の書き換えを有効にする。ブラウザが cookie を受け入れない場合に、URL の書き換えを通してセッションを追跡できる。ただし、サーブレットや JavaServer Pages™ (JSP™) ページで encodeURL または encodeRedirectURL の呼び出しを使用することも必要
idLengthBytes	128	Web モジュールのセッション ID の長さをバイト単位で指定する

J2EE サービス要素

J2EE サービス要素は次のとおりです。

- `.jms-service`
- `log-service`
- `security-service`
- `auth-realm`
- `transaction-service`

jms-service

Sun ONE Application Server で管理される、組み込み Java Message Service (JMS) を設定します。

サブ要素

次の表は、`jms-service` 要素のサブ要素を示します。

表 2-46 `jms-service` のサブ要素

要素	必要数	説明
<code>property</code>	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、`jms-service` 要素の属性を示します。

表 2-47 `jms-service` の属性

属性	デフォルト値	説明
<code>port</code>	7676	(省略可能) JMS プロバイダが使うポート番号を指定する
<code>admin-user-name</code>	admin	JMS プロバイダの管理者ユーザー名を指定する
<code>admin-password</code>	admin	JMS プロバイダの管理者パスワードを指定する
<code>init-timeout-in-seconds</code>	60	(省略可能) 起動時にサーバーインスタンスが、対応する JMS インスタンスの応答を待つ時間を指定する。応答がない場合、起動は中断される。0 に設定すると、サーバーインスタンスはいつまでも待機する
<code>start-args</code>	なし	(省略可能) 対応する JMS インスタンスを起動するための引数の文字列を指定する

表 2-47 jms-service の属性 (続き)

属性	デフォルト値	説明
log-level	log-service 要素の level 属性の値	(省略可能) この要素がサーバーログに記録する、メッセージの タイプを制御する。詳細は、log-service 要素の level 属性 の説明を参照
enabled	true	(省略可能) true に設定すると、Sun ONE Applicatin Server イン スタンスが JMS プロバイダの起動と停止を行う。false に設 定すると、Sun ONE Application Servr インスタンスは JMS プ ロバイダの起動と終了を行わない (JMS プロバイダが、使われて いないか、Sun ONE Applicatin Server から独立して管理されて いるため) 有効な値は on、off、yes、no、1、0、true、false

プロパティ

次の表は、jms-service 要素のプロパティを示します。

表 2-48 jms-service のプロパティ

プロパティ	デフォルト値	説明
instance-name	domain_instance	ドメインとサーバーインスタンス名をつなげた、完全な Sun™ Open Net Environment (Sun ONE) Message Queue ブローカインスタンス名を指定する。例： domain1_server1
instance-name-suffix	なし	完全な Sun ONE Message Queue ブローカインスタンス 名に追加するサフィックスを指定する。サフィックスと インスタンス名は、下線文字 () で区切られる。たとえ ば、インスタンス名が domain1_server1 である場合 に、サフィックス xyz を付加すると、インスタンス名 は domain1_server1_xyz になる
append-version	false	true の場合、下線文字 () を前に付けて、メジャーお よびマイナーバージョン番号を完全な Sun ONE Message Queue ブローカインスタンス名に付加する。 たとえば、インスタンス名が domain1_server1 であ る場合に、バージョン番号を付加すると、インスタンス 名は domain1_server1_7_0 になる

log-service

次のログファイルを含む、システムログサービスを設定します。

- **サーバーログファイル**にはデフォルトの仮想サーバーからのメッセージが格納されます。その他の設定されている仮想サーバーからのメッセージもここに格納されます。ただし、log-file 属性が **virtual-server-class** または **virtual-server** 要素内で明示的に指定されている場合を除きます。デフォルトの名前は `server.log` です。
- **アクセスログファイル**にはデフォルトの仮想サーバーからの HTTP アクセスメッセージが格納されます。デフォルトの名前は `access.log` です。アクセスログを設定するには、`init.conf` および `obj.conf` ファイルのサーバーアプリケーションの関数を使います。詳細は、『Sun ONE Application Server Developer's Guide to NSAPI』を参照してください。
- **トランザクションログファイル**にはデフォルトの仮想サーバーからのトランザクションメッセージが格納されます。これらのファイルが格納されているディレクトリのデフォルト名は `tx` です。
- **仮想サーバーログファイル**には、log-file 属性が明示的に指定された、**virtual-server-class** または **virtual-server** 要素からのメッセージが格納されます。

サブ要素

次の表は、log-service 要素のサブ要素を示します。

表 2-49 log-service のサブ要素

要素	必要数	説明
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、log-service 要素の属性を示します。

表 2-50 log-service の属性

属性	デフォルト値	説明
file	server.log server 要素の log-root 属性で指定したディレクトリに存在する	(省略可能) サーバーログの名前または場所をオーバーライドする。サーバーログが保存されるファイルとディレクトリには、サーバーを実行しているユーザーアカウントも書き込めることが必要 絶対パスを指定すると、この値が server 要素の log-root 属性をオーバーライドする 相対パスを指定する場合、相対パスは server 要素の log-root 属性に相対的になる。log-root 値を指定しない場合は、このパスは instance_dir/config に相対的になる
level	INFO	(省略可能) 別の要素によってサーバーログに記録されるメッセージのデフォルトタイプを制御する。多くのその他の要素が、log-level 属性のこのデフォルトをオーバーライドできる。有効な値 (降順) は次のとおり。FINEST、FINER、FINE、CONFIG、INFO、WARNING、SEVERE、ALERT、FATAL 各値は、それよりも低いレベルの値のすべてのメッセージをログに記録する。たとえば、FINEST ではすべてのメッセージが記録され、FATAL では FATAL メッセージだけが記録される。デフォルト値は SEVERE であり、SEVERE、ALERT、FATAL のすべてのメッセージがログに記録される
log-stdout	true	(省略可能) true の場合、stdout 出力がサーバーログにリダイレクトされる。有効な値は on、off、yes、no、1、0、true、false
log-stderr	true	(省略可能) true の場合、stderr 出力がサーバーログにリダイレクトされる。有効な値は on、off、yes、no、1、0、true、false
echo-log-messages-to-stderr	true	(省略可能) true の場合、サーバーログに加えてログメッセージが stderr に送信される。有効な値は on、off、yes、no、1、0、true、false
log-virtual-server-id	false	(省略可能) true の場合、仮想サーバー ID が仮想サーバーログに表示される。これは、複数の virtual-server 要素が同じログファイルを共有している場合に有用。有効な値は on、off、yes、no、1、0、true、false
use-system-logging	false	(省略可能) true の場合、UNIX syslog サービスを使ってログの作成と管理が行われる。有効な値は on、off、yes、no、1、0、true、false

security-service

J2EE セキュリティサービスで必要となるパラメータと設定情報を定義します。

サブ要素

次の表は、security-service 要素のサブ要素を示します。

表 2-51 security-service のサブ要素

要素	必要数	説明
auth-realm	1 個以上	認証のレルムを定義する
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、security-service 要素の属性を示します。

表 2-52 security-service の属性

属性	デフォルト値	説明
default-realm	file	(省略可能) このサーバーインスタンスのアクティブな認証レルム (auth-realm name 属性) を指定する
default-principal	なし	(省略可能) 主体がない場合に、必要に応じて、デフォルトのセキュリティコンテキストの ID として使われる。通常のサーバー操作では、この属性を設定する必要はない
default-principal-password	なし	(省略可能) デフォルトの主体のパスワード。通常のサーバー操作では、この属性を設定する必要はない
anonymous-role	ANYONE	(省略可能) デフォルトまたは匿名のロールの名前として使われる。匿名のロールは常にすべての主体に割り当てられる。このロール値を J2EE 配備記述子を使って、誰にでもアクセス権を与えることができる

表 2-52 security-service の属性 (続き)

属性	デフォルト値	説明
audit-enabled	false	(省略可能) true の場合、監査情報を提供するために追加のアクセスログが実行される。有効な値は on、off、yes、no、1、0、true、false 監査情報には、次の内容が含まれる <ul style="list-style-type: none"> • 認証の成功および失敗イベント • サブレットおよび EJB へのアクセスの許可と拒否
log-level	log-service 要素の level 属性の値	(省略可能) この要素がサーバーログに記録する、メッセージのタイプを制御する。詳細は、log-service 要素の level 属性の説明を参照

auth-realm

認証のレルムを定義します。

認証のレルムには、プロバイダ固有のプロパティが必要となり、このプロパティは特定の実装のニーズによって異なります。

レルムを定義する方法の詳細については、『Sun ONE Application Server Developer's Guide』を参照してください。

次に、デフォルトの file レルムの例を示します。

```
<auth-realm name="file"
  classname="com.ipplanet.ias.security.auth.realm.file.FileRealm">
  <property name="file" value="instance_dir/config/keyfile"/>
  <property name="jaas-context" value="fileRealm"/>
</auth-realm>
```

auth-realm 要素が使用するプロパティは、auth-realm 要素の name 属性によって決まります。file レルムは、file および jaas-context プロパティを使用します。その他のレルムは、別のプロパティを使用します。

サブ要素

次の表は、auth-realm 要素のサブ要素を示します。

表 2-53 auth-realm のサブ要素

要素	必要数	説明
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、auth-realm 要素の属性を示します。

表 2-54 auth-realm の属性

属性	デフォルト値	説明
name	なし	このレルムの名前を指定する
classname	なし	このレルムを実装する Java クラスを指定する

プロパティ

Sun ONE Application Server が提供する標準のレルムには、必須および省略可能なプロパティがあります。カスタムレルムのプロパティは、標準レルムのプロパティとは異なるものを使用できます。

次の表は、auth-realm 要素のプロパティを示します。

表 2-55 auth-realm のプロパティ

プロパティ	レルム	説明
jaas-context	file, ldap, solaris	JAAS (Java Authentication and Authorization Service) コンテキストを指定する
file	file	ユーザー名を保存するファイルを指定する。デフォルトは、instance_dir/config/keyfile である
assign-groups	Certificate	(省略可能) このプロパティを設定すると、その値はグループ名をコンマで区切ったリストの形になる。有効な証明書を持つすべてのクライアントには、これらのグループのメンバーシップが割り当てられ、Web コンテナと EJB コンテナの承認に使用される
directory	ldap	LDAP URL をサーバーに指定する

表 2-55 auth-realm のプロパティ (続き)

プロパティ	レルム	説明
base-dn	ldap	ユーザーデータの位置の LDAP 基本 DN を指定する。ツリー範囲検索が実行されるため、この基本 DN は、ユーザーデータのレベルより上に置かれる。検索ツリーが小さければ小さいほど、パフォーマンスは良くなる
search-filter	ldap	(省略可能) ユーザーを検索するために使う検索フィルタを指定する。デフォルトは、uid=%s (%s はサブジェクト名に展開される)
group-base-dn	ldap	(省略可能) グループデータの位置の基本 DN を指定する。デフォルトでは、base-dn と同じ。ただし、必要に応じて変更できる
group-search-filter	ldap	(省略可能) ユーザーのグループメンバーシップを検索するために使う検索フィルタを指定する。デフォルトは、uniquemember=%d (%d はユーザー要素 DN に展開される)
group-target	ldap	(省略可能) グループ名エントリを含む LDAP の属性名を指定する。デフォルトは、CN
search-bind-dn	ldap	(省略可能) search-filter 検索を実行するためにディレクトリに対する認証に使うオプションの DN を指定する。匿名検索を実行できないディレクトリだけで必要とされる
search-bind-password	ldap	(省略可能) search-bind-dn に指定された DN の LDAP パスワードを指定する

transaction-service

Java Transaction Service (JTS) を設定します。

サブ要素

次の表は、transaction-service 要素のサブ要素を示します。

表 2-56 transaction-service のサブ要素

要素	必要数	説明
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、transaction-service 要素の属性を示します。

表 2-57 transaction-service の属性

属性	デフォルト値	説明
automatic-recovery	false	(省略可能) true の場合、サーバーインスタンスは起動時にトランザクションの復旧を実行する。有効な値は on、off、yes、no、1、0、true、false
timeout-in-seconds	0	(省略可能) トランザクションを中止するまでの時間を指定する。0 に設定すると、トランザクションはタイムアウトにならない
tx-log-dir	server 要素の log-root 属性で指定したディレクトリ	(省略可能) トランザクションログディレクトリの位置をオーバーライドする。トランザクションログが保存されるディレクトリには、サーバーで実行されているどのユーザーカウントも書き込めることが必要。ログについての詳細は、「log-service」の説明を参照
heuristic-decision	ロールバック	(省略可能) 復旧中に、トランザクションの結果をログから決定できない場合、このプロパティが結果を決定する。有効な値は rollback および commit
keypoint-interval	2048	(省略可能) ログ内でのキーポイント処理間のトランザクションの数を指定する。キーポイント処理は、トランザクションログファイルを圧縮してサイズを小さくする。この属性の値を大きくすると (たとえば 4096)、トランザクションログファイルが大きくなるが、キーポイント処理が少なくなり、パフォーマンスが向上する可能性がある。小さい値 (たとえば 100) にするとログファイルは小さくなるが、キーポイント処理の頻度が増えるためパフォーマンスが少し低下する
log-level	log-service 要素の level 属性の値	(省略可能) この要素がサーバーログに記録する、メッセージのタイプを制御する。詳細は、log-service 要素の level 属性の説明を参照
monitoring-enabled	false	(省略可能) JTS の監視を有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false。デフォルトは false

プロパティ

次の表は、transaction-service 要素のプロパティを示します。

表 2-58 transaction-service のプロパティ

プロパティ	デフォルト値	説明
oracle-xa-recovery-wor karound	false	true の場合、Oracle XA Resource の回避策がトランザク ションの復旧に使用される。有効な値は on、off、yes、 no、1、0、true、false
sybase-xa-recovery-wor karound	false	true の場合、Sybase XA Resource の回避策がトランザク ションの復旧に使用される。有効な値は on、off、yes、 no、1、0、true、false
disable-distributed-tr ansaction-logging	false	true の場合、トランザクションのログへの記録を無効に するので、パフォーマンスが向上する。有効な値は on、 off、yes、no、1、0、true、false automatic-recovery 属性を true に設定した場合は、 このプロパティは無視される
xaresource-txn-timeout	使用される XAResource に固有	XAResource タイムアウトを変更する。XAResource の デフォルトのタイムアウトによりトランザクションが中止 されることがあるので、この値を変更することが望ましい

Java 設定要素

Java 設定要素は次のとおりです。

- [java-config](#)
- [profiler](#)
- [jvm-options](#)

java-config

Java Virtual Machine (JVM) 設定パラメータを指定します。

サブ要素

次の表は、java-config 要素のサブ要素を示します。

表 2-59 java-config のサブ要素

要素	必要数	説明
profiler	0 または 1 個	Sun ONE Application Server 用にプロファイ ラを設定する
jvm-options	0 または 1 個以上	JVM コマンド行オプションを含む

表 2-59 java-config のサブ要素 (続き)

要素	必要数	説明
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、java-config 要素の属性を示します。

表 2-60 java-config の属性

属性	デフォルト値	説明
java-home	なし	JDK がインストールされているディレクトリのパス
debug-enabled	false	(省略可能) true の場合、サーバーは JPDA ベースのデバッガに接続できるようにデバッグモードで起動する。有効な値は on、off、yes、no、1、0、true、false
debug-options	-Xdebug -Xrunjdpw:transport=dt_socket,server=y,suspend=n	(省略可能) JPDA (Java Platform Debugger Architecture) オプションを指定する。指定可能なデバッグオプションのリストは、次のサイトを参照 http://java.sun.com/products/jpda/doc/conninv.html#Invocation デバックの詳細は、『Sun ONE Application Server Developer's Guide』を参照
rmic-options	-iiop -poa -alwaysgenerate -keepgenerated -g	(省略可能) アプリケーションの配備時に RMI コンパイラに渡すオプションを指定する。 -keepgenerated オプションを指定すると、スタブとタイ用に生成されるソースが保存される rmic コマンドの詳細は、『Sun ONE Application Server Enterprise Java Beans 開発者ガイド』を参照
javac-options	-g	(省略可能) アプリケーションの配備時に Java コンパイラに渡すオプションを指定する

表 2-60 java-config の属性 (続き)

属性	デフォルト値	説明
classpath-prefix	なし	(省略可能) システムクラスパスのプレフィックスを指定する。XML パーサークラスなどのシステムクラスをオーバーライドしたい場合にだけ、システムクラスパスにプレフィックスを付ける。この属性を使用する際には注意が必要である
server-classpath	なし	(省略可能) サーバーが起動された環境のクラスパスを指定する。このクラスパスには、 <code>System.getProperty("java.class.path")</code> を使ってアクセスできる
classpath-suffix	なし	(省略可能) システムクラスパスのサフィックスを指定する
native-library-path-prefix	なし	(省略可能) ネイティブライブラリパスのプレフィックスを指定する ネイティブライブラリパスとは、Application Server のネイティブ共有ライブラリの相対パス、標準の JRE ネイティブライブラリパス、シェル環境設定 (UNIX の <code>LD_LIBRARY_PATH</code>)、および profiler 要素に指定されたパスを、自動的に連結して作成したパスのことである。これは組み合わせによって作成されるため、サーバーの設定には明示的に表示されない
native-library-path-suffix	なし	(省略可能) ネイティブライブラリパスのサフィックスを指定する
bytecode-preprocessors		(省略可能) それぞれに <code>com.sun.appserv.BytecodePreprocessor</code> インタフェースを実装する必要があるクラス名をコンマで区切って指定する。指定された各プリプロセッサクラスは、指定された順に呼び出される

表 2-60 java-config の属性 (続き)

属性	デフォルト値	説明
env-classpath-ignored	true	(省略可能) false の場合、CLASSPATH 環境変数が読み取られ、Sun ONE Application Server のクラスパスに付加される。 CLASSPATH 環境変数は、一番最後の、classpath-suffix の後に追加される 開発環境では、この値を false に設定することが必要。本稼動環境では、この値を true に設定して、環境変数による影響を避ける必要がある 有効な値は on、off、yes、no、1、0、true、false

profiler

Sun ONE Application Server 用にプロファイラを設定します。プロファイラの詳細は、『Sun ONE Application Server Developer's Guide』を参照してください。

サブ要素

次の表は、profiler 要素のサブ要素を示します。

表 2-61 profiler のサブ要素

要素	必要数	説明
jvm-options	0 または 1 個以上	プロファイラ固有の JVM コマンド行オプションを含む
property	0 または 1 個以上	プロパティまたは変数を指定する

注	profiler 要素のサブ要素は任意の順序で記述できます。
---	--------------------------------

属性

次の表は、profiler 要素の属性を示します。

表 2-62 profiler の属性

属性	デフォルト値	説明
name	なし	プロファイラの名前を指定する
classpath	なし	(省略可能) プロファイラのクラスパスを指定する
native-library-path	なし	(省略可能) プロファイラのネイティブライブラリパスを指定する
enabled	true	(省略可能) プロファイラを有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

jvm-options

次のような JVM コマンド行オプションを含みます。

```
<jvm-options>-Xdebug -Xmx128m</jvm-options>
```

使用可能なオプションについては、次のサイトを参照してください。

<http://java.sun.com/docs/hotspot/VMOptions.html>

サブ要素

なし

属性

なし

リソース要素

リソース要素は次のとおりです。

- resources
- custom-resource
- external-jndi-resource
- jdbc-resource
- mail-resource
- jms-resource
- persistence-manager-factory-resource
- jdbc-connection-pool

resources

データベース接続、JavaMail™ セッションなどの設定されたリソースが含まれています。

注	リソースごとに Java Naming and Directory Interface™ (JNDI) 名を指定する必要があります。JNDI 内のその他のエンタープライズリソース名との衝突や移植性の問題を回避するため、Sun ONE Application Server アプリケーション内のすべての名前を、文字列 java:comp/env で始める必要があります。
---	--

サブ要素

次の表は、resources 要素のサブ要素を示します。

表 2-63 resources のサブ要素

要素	必要数	説明
custom-resource	0 または 1 個以上	カスタムリソースを定義する
external-jndi-resource	0 または 1 個以上	外部 JNDI リポジトリに配備されているリソースを定義する
jdbc-resource	0 または 1 個以上	JDBC (Java Database Connectivity) リソースを定義する
mail-resource	0 または 1 個以上	JavaMail リソースを定義する
jms-resource	0 または 1 個以上	JMS リソースを定義する

表 2-63 resources のサブ要素 (続き)

要素	必要数	説明
<code>persistence-manager-factory-resource</code>	0 または 1 個以上	CMP 用の持続マネージャファクトリリソースを定義する
<code>jdbc-connection-pool</code>	0 または 1 個以上	JDBC 接続プールの作成に必要なプロパティを定義する

注 resources 要素のサブ要素は任意の順序で記述できます。

属性

なし

custom-resource

カスタムサーバー全体のリソースオブジェクトファクトリを指定する、カスタムリソースを定義します。これらのオブジェクトファクトリは、`javax.naming.spi.ObjectFactory` インタフェースを実装しています。

サブ要素

次の表は、`custom-resource` 要素のサブ要素を示します。

表 2-64 custom-resource のサブ要素

要素	必要数	説明
<code>description</code>	0 または 1 個	この要素を説明するテキストを含む
<code>property</code>	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、`custom-resource` 要素の属性を示します。

表 2-65 custom-resource の属性

属性	デフォルト値	説明
<code>jndi-name</code>	なし	リソースの JNDI 名を指定する
<code>res-type</code>	なし	リソースの完全指定されたタイプを指定する

表 2-65 custom-resource の属性 (続き)

属性	デフォルト値	説明
factory-class	なし	javax.naming.spi.ObjectFactory を実装する、ユーザーが記述したファクトリクラスの完全指定名を指定する
enabled	true	(省略可能) リソースを実行時に有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

external-jndi-resource

外部 JNDI リポジトリに配備されているリソースを定義します。たとえば、汎用 Java オブジェクトを LDAP サーバーに格納します。外部 JNDI ファクトリは、javax.naming.spi.InitialContextFactory インタフェースを実装する必要があります。

サブ要素

次の表は、external-jndi-resource 要素のサブ要素を示します。

表 2-66 external-jndi-resource のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、external-jndi-resource 要素の属性を示します。

表 2-67 external-jndi-resource の属性

属性	デフォルト値	説明
jndi-name	なし	リソースの JNDI 名を指定する
jndi-lookup-name	なし	リソースの JNDI 検索名を指定する
res-type	なし	リソースの完全指定されたタイプを指定する

表 2-67 external-jndi-resource の属性 (続き)

属性	デフォルト値	説明
factory-class	なし	javax.naming.spi.InitialContextFactory を実装する、ファクトリクラスの完全指定名を指定する JNDI の詳細は、『Sun ONE Application Server Developer's Guide to J2EE Services and APIs』を参照
enabled	true	(省略可能) リソースを実行時に有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

jdbc-resource

JDBC (javax.sql.DataSource) リソースを定義します。

サブ要素

次の表は、jdbc-resource 要素のサブ要素を示します。

表 2-68 jdbc-resource のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、jdbc-resource 要素の属性を示します。

表 2-69 jdbc-resource の属性

属性	デフォルト値	説明
jndi-name	なし	リソースの JNDI 名を指定する
pool-name	なし	jdbc-connection-pool 要素で定義されている、関連付けられた JDBC 接続プールの name を指定する
enabled	true	(省略可能) リソースを実行時に有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

mail-resource

JavaMail (javax.mail.Session) リソースを定義します。

サブ要素

次の表は、mail-resource 要素のサブ要素を示します。

表 2-70 mail-resource のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、mail-resource 要素の属性を示します。

表 2-71 mail-resource の属性

属性	デフォルト値	説明
jndi-name	なし	リソースの JNDI 名を指定する
store-protocol	imap	(省略可能) ストアプロトコルサービスを指定する。このサービスは、メールサーバーへの接続、メッセージの取得、およびフォルダへのメッセージの保存を実行する。値の例は <code>imap</code> および <code>pop3</code>
store-protocol-class	<code>com.sun.mail.imap.IMAPStore</code>	(省略可能) ストア用のサービスプロバイダ実装クラスを指定する
transport-protocol	smtp	(省略可能) メッセージを送信するトランスポートプロトコルサービスを指定する
transport-protocol-class	<code>com.sun.mail.smtp.SMTPTransport</code>	(省略可能) トランスポート用のサービスプロバイダ実装クラスを指定する
host	なし	メールサーバーのホスト名
user	なし	メールサーバーのユーザー名
from	なし	メッセージの送信者を示すためにメールサーバーが使う電子メールアドレス
Debug	false	(省略可能) リソースのデバッグを有効にするかどうかを決定する。有効な値は <code>on</code> 、 <code>off</code> 、 <code>yes</code> 、 <code>no</code> 、 <code>1</code> 、 <code>0</code> 、 <code>true</code> 、 <code>false</code>

表 2-71 mail-resource の属性 (続き)

属性	デフォルト値	説明
enabled	true	(省略可能) リソースを実行時に有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

プロパティ

mail-resource 要素のプロパティを設定してから、後で JavaMail Session オブジェクトから設定したプロパティを取得できます。すべてのプロパティ名は、mail- というプレフィックスで始める必要があります。Sun ONE Application Server は、プロパティ名内のダッシュ (-) をピリオド (.) に変換し、プロパティを MailConfiguration および JavaMail Session オブジェクトに保存します。プロパティの名前が mail- で始まらない場合は、プロパティは無視されます。

たとえば、JavaMail Session オブジェクト内に mail.password プロパティを定義したい場合は、まず次のように server.xml を編集します。

```
...
<mail-resource jndi-name="mail/Session" ...>
  <property name="mail-password" value="adminadmin"/>
</mail-resource>
...
```

JavaMail Session オブジェクトを取得後、次のようにして、値 adminadmin を得るために mail.password プロパティを取得できます。

```
String password = session.getProperty("mail.password");
```

jms-resource

JMS (Java Message Service) リソースを定義します。

サブ要素

次の表は、jms-resource 要素のサブ要素を示します。

表 2-72 jms-resource のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、`jms-resource` 要素の属性を示します。

表 2-73 `jms-resource` の属性

属性	デフォルト値	説明
<code>jndi-name</code>	なし	リソースの JNDI 名を指定する
<code>res-type</code>	なし	リソースの完全修飾されたタイプを指定する。次のいずれかのタイプを指定できる <code>javax.jms.Topic</code> <code>javax.jms.Queue</code> <code>javax.jms.TopicConnectionFactory</code> <code>javax.jms.QueueConnectionFactory</code>
<code>enabled</code>	<code>true</code>	(省略可能) リソースを実行時に有効にするかどうかを決定する。有効な値は <code>on</code> 、 <code>off</code> 、 <code>yes</code> 、 <code>no</code> 、 <code>1</code> 、 <code>0</code> 、 <code>true</code> 、 <code>false</code>

プロパティ

次の表は、`jms-resource` 要素のプロパティを示します。

表 2-74 `jms-resource` のプロパティ

プロパティ	デフォルト値	説明
<code>imqDestinationName</code>	なし	この JMS リソースに関連付けられた物理的な JMS 送信先名を指定する。 <code>javax.jms.Topic</code> または <code>javax.jms.Queue</code> の <code>res-type</code> で <code>jms-resource</code> 要素のプロパティを指定する必要がある 『Sun ONE Message Queue 管理者ガイド』には、このプロパティのデフォルトの値が示されているが、この値は Sun ONE Application Server 環境に適用しない
<code>imqBrokerHostName</code>	Sun ONE Application Server インスタンスと同じホスト名 (localhost)	JMS サービス (Sun ONE Message Queue ブローカ) が実行されているホストの名前を指定する。 <code>res-type</code> が <code>javax.jms.TopicConnectionFactory</code> または <code>javax.jms.QueueConnectionFactory</code> の <code>jms-resource</code> 要素用

表 2-74 jms-resource のプロパティ (続き)

プロパティ	デフォルト値	説明
imqBrokerHostPort	jms-service 要素の port 属性	JMS サービス (Sun ONE Message Queue ブローカ) が実行されているポートを指定する。res-type が javax.jjms.TopicConnectionFactory または javax.jms.QueueConnectionFactory の jms-resource 要素用
imqConfiguredClientID	なし	QueueConnectionFactory および TopicConnectionFactory クラスそれぞれに対応する createQueueConnection および createTopicConnection JMS API を使用して作成した Connection に関連付けられた JMS クライアント ID を指定する res-type が javax.jjms.TopicConnectionFactory または javax.jms.QueueConnectionFactory の jms-resource 要素用 永続的なサブスクリプション名は一意であり、クライアント ID の範囲内でのみ有効である。永続的サブスクリプを作成またはアクティブにし直すには、接続には有効なクライアント ID が必要である。JMS 仕様によると、クライアント ID が一意であり、特定のクライアント ID は一度に 1 つのアクティブな接続にしか使用できない

persistence-manager-factory-resource

コンテナ管理持続性 (CMP) 用の持続マネージャファクトリリソースを定義します。

サブ要素

次の表は、persistence-manager-factory-resource 要素のサブ要素を示します。

表 2-75 persistence-manager-factory-resource のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、persistence-manager-factory-resource 要素の属性を示します。

表 2-76 persistence-manager-factory-resource の属性

属性	デフォルト値	説明
jndi-name	なし	リソースの JNDI 名を指定する
factory-class	com.sun.jdo.spi. persistence. support.sqlstore .impl. PersistenceManag erFactoryImpl	(省略可能) ファクトリクラスの名前を指定する。この属性はサードパーティ製の CMP 持続マネージャファクトリをサポートしている。サードパーティの CMP 実装が必要とする名前を使用する。この属性は、組み込み CMP 実装には使わないこと
jdbc-resource-jndi-name	なし	(省略可能) データベース接続が取得される jdbc-resource を指定する。これは、既存の jdbc-resource の jndi-name であることが必要
enabled	true	(省略可能) リソースを実行時に有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

jdbc-connection-pool

JDBC 接続プールの作成に必要なプロパティを定義します。

ヒント	プール定義を作成し、次にコピー、ペースト、および編集して、複数の JDBC データソースを設定することができます。
-----	---

サブ要素

次の表は、jdbc-connection-pool 要素のサブ要素を示します。

表 2-77 jdbc-connection-pool のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、jdbc-connection-pool 要素の属性を示します。

表 2-78 jdbc-connection-pool の属性

属性	デフォルト値	説明
name	なし	接続プールの名前を指定する。 <code>jdbc-resource</code> 要素の <code>pool-name</code> 属性は、この <code>name</code> に指定された値を参照する
datasource-classname	なし	ベンダーが提供するデータソースに関連付けられたクラス名を指定する。このクラスは、 <code>java.sql.DataSource</code> 、 <code>java.sql.XADataSource</code> 、あるいはその両方を実装していることが必要
res-type	<code>javax.sql.DataSource</code>	(省略可能) データソースクラスが実装するインタフェースを指定する。この属性の値として、 <code>javax.sql.DataSource</code> または <code>javax.sql.XADataSource</code> を指定することができる。値がこれらのインタフェースのいずれでもない場合は、デフォルトが使用される。この属性に有効な値が指定されていても、指定されたインタフェースがデータソースクラスに実装されていない場合は、エラーが発生する
steady-pool-size	8	(省略可能) プールに保持される接続の初期数および最小数を指定する
max-pool-size	32	(省略可能) クライアントの要求に対応するために作成できる接続の最大数を指定する
max-wait-time-in-millis	60000	(省略可能) 呼び出し側が接続を取得するまで待機する時間をミリ秒単位で指定する。0 の場合、呼び出し側はリソースが利用できるようになるまで、またはエラーが発生するまで待機し続ける
pool-resize-quantity	2	(省略可能) 既存の接続の数が <code>steady-pool-size</code> (<code>max-pool-size</code> による制限を受ける) を超える場合に破棄される接続の数を指定する。これは、 <code>idle-time-out-in-seconds</code> 間隔で定期的に行われる。アイドル状態の接続は、 <code>idle-time-out-in-seconds</code> の間使用されない接続である
idle-timeout-in-seconds	300	(省略可能) プールで接続がアイドル状態のままでいられる最長時間を指定する。この時間を過ぎると、プールはこの接続を閉じる
transaction-isolation-level	デフォルトの JDBC ドライバ遮断レベル	(省略可能) プールされたデータベース接続のトランザクション遮断レベルを指定する。有効な値は <code>read-uncommitted</code> 、 <code>read-committed</code> 、 <code>repeatable-read</code> 、 <code>serializable</code> プールされた接続の遮断レベルをアプリケーションでプログラム的に変更すると、プールのしくみに支障をきたす場合があり、エラーが発生することがある。詳細については、 <code>is-isolation-level-guaranteed</code> を参照

表 2-78 jdbc-connection-pool の属性 (続き)

属性	デフォルト値	説明
is-isolation-level-guaranteed	true	(省略可能) transaction-isolation-level が明示的に設定されているときに限り適用される。true の場合、プールから取得されたすべての接続は希望する遮断レベルを持つことができる。これは、いくつかの JDBC ドライバのパフォーマンスに影響を与えることがある。ホストされているアプリケーションが変更された遮断レベルで接続を返さないことが確かな場合は、この属性を false に設定できる
is-connection-validation-required	false	(省略可能) アプリケーションに対して接続が確立される前に接続を検証する必要があるかどうかを指定する。リソースの検証が失敗すると破棄され、新しいリソースが作成されて返される。有効な値は on、off、yes、no、1、0、true、false
connection-validation-method	auto-commit	(省略可能) 有効な値は次のとおり <ul style="list-style-type: none">• auto-commit (デフォルト)。Connection.setAutoCommit (Connection.getAutoCommit ()) を使用する• meta-data。Connection.getMetaData () を使用する• table。validation-table-name 属性で指定されているテーブルでクエリを実行する
validation-table-name	なし	(省略可能) 接続を検証するクエリを実行するために使用するテーブル名を指定する。connection-validation-type が table に設定されている場合に限り、このパラメータは必須である
fail-all-connections	false	(省略可能) true の場合、1 つでも検証が失敗するとプールのすべての接続が閉じられる。is-connection-validation-required が true に設定されている場合に限り、このパラメータは必須である。有効な値は on、off、yes、no、1、0、true、false

プロパティ

ほとんどの JDBC 2.0 ドライバは、標準のプロパティリストを使って、ユーザー、パスワード、およびその他のリソース設定情報を指定することができます。Sun ONE Application Server ではプロパティはオプションですが、中にはほとんどのデータベースで必要なプロパティもあります。詳細は、『JDBC 2.0 Standard Extension API』の第 5.3 節を参照してください。

プロパティが指定されているときは、setName(value) メソッドを使って、datasource-classname 属性で指定されたベンダーのデータソースクラスに渡されます。

コンテナ管理認証が指定されていて、`default-resource-principal` がアプリケーション配備記述子内にない場合、`user` および `password` プロパティがデフォルトの主体として使われます。

次の表は、`jdbc-connection-pool` 要素の共通プロパティを示します。

表 2-79 `jdbc-connection-pool` プロパティ

プロパティ	説明
<code>user</code>	接続プールのユーザー名を指定する
<code>password</code>	接続プールのパスワードを指定する
<code>databaseName</code>	接続プールのデータベースを指定する
<code>serverName</code>	接続プールのデータベースサーバーを指定する
<code>port</code>	データベースサーバーが要求を待機するポートを指定する
<code>networkProtocol</code>	通信プロトコルを指定する
<code>roleName</code>	最初の SQL ロール名を指定する
<code>datasourceName</code>	基礎となる <code>XADataSource</code> を指定するか、または、接続プールが実行されている場合は <code>ConnectionPoolDataSource</code> を指定する
<code>description</code>	説明テキストを指定する
<code>url</code>	接続プールの URL を指定する。標準のプロパティではないが、広く使用される

アプリケーション要素

Application 要素は次のとおりです。

- applications
- lifecycle-module
- j2ee-application
- web-module
- ejb-module
- connector-module

applications

配備された J2EE アプリケーション、J2EE モジュール、およびライフサイクルモジュールを含みます。

サブ要素

次の表は、applications 要素のサブ要素を示します。

表 2-80 applications のサブ要素		
要素	必要数	説明
lifecycle-module	0 または 1 個以上	配備されたライフサイクルモジュールを指定する
j2ee-application	0 または 1 個以上	配備された J2EE アプリケーションを指定する
ejb-module	0 または 1 個以上	配備された EJB モジュールを指定する
web-module	0 または 1 個以上	配備された Web モジュールを指定する
connector-module	0 または 1 個以上	配備されたコネクタモジュールを指定する

注	applications 要素のサブ要素は任意の順序で記述できます。
---	------------------------------------

属性

次の表は、applications 要素の属性を示します。

表 2-81 applications の属性

属性	デフォルト値	説明
dynamic-reload-enabled	false	(省略可能) 動的再読み込みを有効にするかどうかを指定する。この属性は、開発環境では true に設定し、本稼動環境では false に設定することが必要である。有効な値は on、off、yes、no、1、0、true、false
dynamic-reload-poll-interval-in-seconds	2	(省略可能) アプリケーションとモジュールのコードが変更されていないかを確認し、動的再読み込みする間隔を指定する

lifecycle-module

配備されたライフサイクルモジュールを指定します。ライフサイクルモジュールの詳細は、『Sun ONE Application Server Developer's Guide』を参照してください。

サブ要素

次の表は、lifecycle-module 要素のサブ要素を示します。

表 2-82 lifecycle-module のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む
property	0 または 1 個以上	プロパティまたは変数を指定する

属性

次の表は、lifecycle-module 要素の属性を示します。

表 2-83 lifecycle-module の属性

属性	デフォルト値	説明
name	なし	ライフサイクルモジュールの名前
class-name	なし	ライフサイクルモジュールのクラスファイルの完全指定名。 com.sun.appserv.server.LifecycleListener インタフェースを実装していることが必要である

表 2-83 lifecycle-module の属性 (続き)

属性	デフォルト値	説明
classpath	server 要素の application-root 属性の値	(省略可能) ライフサイクルモジュールのクラスパス。モジュールの場所を指定する
load-order	なし	(省略可能) 起動時にどのライフサイクルモジュールを読み込むかどうかを決定する。モジュールに指定された数値が小さいほど、早く読み込まれる。値の範囲は、101 から オペレーティングシステムの MAXINT まで。1 から 100 までの値は予約されている
is-failure-fatal	false	(省略可能) ライフサイクルモジュールが失敗したときにサーバーをシャットダウンするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false
enabled	true	(省略可能) ライフサイクルモジュールを有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

j2ee-application

配備された J2EE アプリケーションを指定します。

サブ要素

次の表は、j2ee-application 要素のサブ要素を示します。

表 2-84 j2ee-application のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む

属性

次の表は、j2ee-application 要素の属性を示します。

表 2-85 j2ee-application の属性

属性	デフォルト値	説明
name	なし	アプリケーションの名前
location	なし	Sun ONE Application Server ファイルシステムでのアプリケーションの位置

表 2-85 j2ee-application の属性 (続き)

属性	デフォルト値	説明
virtual-servers	すべての仮想サーバー	(省略可能) このアプリケーション内の Web モジュールが配備される仮想サーバー
enabled	true	(省略可能) アプリケーションを有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

ejb-module

配備された EJB モジュールを指定します。

サブ要素

次の表は、ejb-module 要素のサブ要素を示します。

表 2-86 ejb-module のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む

属性

次の表は、ejb-module 要素の属性を示します。

表 2-87 ejb-module の属性

属性	デフォルト値	説明
name	なし	EJB モジュールの名前
location	なし	Sun ONE Application Server ファイルシステムでの EJB モジュールの位置
enabled	true	(省略可能) EJB モジュールを有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

web-module

配備された Web モジュールを指定します。

サブ要素

次の表は、web-module 要素のサブ要素を示します。

表 2-88 web-module のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む

属性

次の表は、web-module 要素の属性を示します。

表 2-89 web-module の属性

属性	デフォルト値	説明
name	なし	Web モジュールの名前
context-root	空文字列	<p>Web モジュールをインストールするコンテキストルート (先頭に / が付かないコンテキストパス)。Java™ Servlet 2.3 仕様書の第 5.4 節を参照</p> <p>この属性が空文字列 (デフォルト値) の場合、この Web モジュールが、virtual-servers 属性で指定される virtual-server 要素のデフォルトの Web モジュールになる</p> <p>仮想サーバーに割り当てられた複数の Web モジュールが空のコンテキストルートを持っている場合、いずれか 1 つが読み込まれ、残りは次のようなエラーメッセージを生成する。 Virtual server [<i>virtual_server</i>] already has a web module loaded at [/]</p> <p>ロードバランスの設定を行う場合、Web モジュールのコンテキストルートはクラスタ内で一意でなければならない。ロードバランスの詳細については、『Sun ONE Application Server 管理者ガイド』を参照</p>

表 2-89 web-module の属性 (続き)

属性	デフォルト値	説明
location	なし	.war ファイルの内容を抽出したディレクトリへの完全修飾パスまたは相対パス。相対パスの場合、次のディレクトリを基準として場所を表す <i>instance_dir/applications/j2ee-modules/</i>
virtual-servers	すべての仮想サーバー	(省略可能) Web モジュールが配備される virtual-server 要素
enabled	true	(省略可能) Web モジュールを有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

connector-module

配備されたコネクタモジュールを指定します。

サブ要素

次の表は、connector-module 要素のサブ要素を示します。

表 2-90 connector-module のサブ要素

要素	必要数	説明
description	0 または 1 個	この要素を説明するテキストを含む

属性

次の表は、connector-module 要素の属性を示します。

表 2-91 connector-module の属性

属性	デフォルト値	説明
name	なし	コネクタモジュールの名前
location	なし	Sun ONE Application Server ファイルシステムでのコネクタモジュールの位置
enabled	true	(省略可能) コネクタモジュールを有効にするかどうかを決定する。有効な値は on、off、yes、no、1、0、true、false

ユーザーデータベースの選択

注 ユーザーデータベースはサーバー自体のセキュリティのみに適用します。J2EE アプリケーションおよびモジュールのセキュリティには関係ありません。

server.xml の auth-db 要素は、次のように親 virtual-server 要素のユーザーデータベースを選択します。

- auth-db 要素の id 属性は、ACL ファイルの database 属性にマップする
- auth-db 要素の database 属性は、dbswitch.conf エントリにマップする

ACL ファイルと dbswitch.conf ファイル間のこのレイヤは、サーバー管理者に、仮想サーバーの管理者とユーザーがアクセスするデータベースを完全に制御する権限を与えます。

注 dbswitch.conf インタフェースは、不確定です。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。

dbswitch.conf ファイルは、次のように LDAP データベースの検索ツリーのルートを確認します。

- dbswitch.conf にある LDAP URL の基本 DN は、すべての詳細な DN 仕様のルートオブジェクトを定義します。最新のインストールでは、基本 DN が空になっていることがあります。それは、最終の基本 DN が、ほかの方法、つまり DC ツリー検索または auth-db 要素の明示的な basedn 値のいずれかで決定されるためです。
- LDAP データベースの dbswitch.conf 属性である dcsuffix は、DC ツリーのルートを定義します。このルートは、LDAP URL の基本 DN と相対関係になります。データベースがスキーマに準拠している場合、dcsuffix を使うことができます。スキーマ準拠の要件は、[99 ページの「Sun ONE LDAP スキーマ」](#)のリストに記載されています。

仮想サーバー用のユーザーデータベースは、次のように選択されます。

- virtual-server に auth-db サブ要素がない場合、ユーザーベースまたはグループベースの ACL は失敗します。

- 仮想サーバーの ACL 定義に database 属性が存在しないときは、virtual-server は default の id 属性を伴う auth-db サブ要素を持っている必要があります。auth-db の database 属性は、dbswitch.conf のデータベースを指します。database 属性が存在しない場合は、default が使われます。
- LDAP データベースがスキーマに準拠している場合、クライアントの提供する Host ヘッダーと一致する virtual-server 要素の host 属性の DC ツリー検索を使って、アクセスの基本 DN が計算されます。host 属性が 1 つも一致しない場合、親 http-listener の server-name 属性が使われます。DC ツリー検索は dcsuffix DN に基づきます。検索結果には、基本 DN を含む inetDomainBaseDN 属性が含まれている必要があります。この基本 DN は、そのまま取得され、いずれの基本 DN 値とも関連がありません。
- auth-db 要素の basedn 属性が存在せず、データベースがスキーマに準拠していない場合、Application Server の前のバージョンでの場合と同様に、アクセスは dbswitch.conf エントリの基本 DN と関連して行われます。

Sun ONE LDAP スキーマ

LDAP データベースがこの節で説明した要件を満たす場合、dbswitch.conf ファイルにある dcsuffix 属性を使うことができます。

`o=isp` のように ISP エントリをルートとするサブツリーは、収束ツリーと呼ばれます。このツリーには、ISP が提供する、組織 (顧客) に関連するすべてのディレクトリデータが含まれます。

`o=internet` をルートとするサブツリーは、ドメインコンポーネントツリーまたは dc ツリーと呼ばれます。このサブツリーは、顧客ドメインのエントリを持つ簡単な DNS ツリーを含んでいます。これらのエントリは、そのドメインのデータがある、収束ツリーの適切な位置へのリンクになっています。

ディレクトリツリーは、ルートが 1 つだけの場合 (推奨) と、異なる 2 つのルート (収束ツリー用と dc ツリー用) がある場合があります。ルートが 1 つというのは、たとえば、`o=root` の下に `o=isp` と `o=internet` がある場合です。

収束ツリー

収束ツリーの最上位レベルには、組織のエントリが、各顧客 (または組織) ごとに 1 つ、そして ISP 自体に 1 つ必要です。

各組織の下には、2 つの `organizationalUnit` エントリ (`ou=People` および `ou=Groups`) が必要です。組織のデバイスデータを格納する場合、3 番目の `ou=Devices` が表示されます。

各ユーザーエントリは、指定した組織内で一意の `uid` 値を持っている必要があります。このサブツリーの下の名前空間をさまざまな `ou` エントリに区分し、ユーザーエントリを `ou=eng` や `ou=corp` のように便利なグループにまとめることができます。その場合でも、ユーザーの `uid` 値は `People` サブツリー全体で一意である必要があります。

収束ツリーのユーザーエントリは、`inetOrgPerson` タイプです。`cn`、`sn`、および `uid` 属性が必要です。`uid` 属性は、有効な電子メール名 (特に、RFC822 で定義されている有効なローカルパート) である必要があります。`cn` には名前のイニシャル `sn` が含まれている必要があります。ユーザーエントリの `RDN` は、`uid` 値に一致している必要があります。ユーザーエントリを有効またはサービスで有効と見なされるようにするには、`AUXILIARY` クラスの `inetUser` を含める必要があります。

ユーザーエントリは、アカウントの管理目的に使われる `AUXILIARY` クラス `inetSubscriber` も含みます。`inetUseStatus` 属性がエントリにあり、その値が `inactive` または `deleted` の場合、エントリは無視されます。

グループは `Groups` サブツリーの下に配備され、`groupOfUniqueNames` タイプの `LDAP` エントリで構成されます。

ドメインコンポーネント (dc) ツリー

`dc` ツリーには、階層構造の `domain` エントリが含まれていて、それぞれが `DNS` 名コンポーネントになっています。

ユーザーのドメイン名を表すエントリは、`LDAP` の `AUXILIARY` クラス `inetDomain` でオーバーレイされます。たとえば、2 つの `LDAP` エントリ

`dc=customer1,dc=com,o=Internet,o=root` および

`dc=customer2,d=com,o=Internet,o=root` は `inetDomain` クラスを含みますが、`dc=com,o=Internet,o=root` は含みません。後者は、ツリーに構造を持たせるためにのみ存在します。

inetDomain 属性を含むエントリは、仮想ドメインと呼ばれます。これらのエントリは、最上位レベルの組織エントリの DN が入力されている inetDomainBaseDN 属性を持っている必要があります。最上位レベルの組織エントリは収束ツリーでこのドメインのデータが格納されている場所です。たとえば、`dc=cust2,dc=com,o=Internet,o=root` の仮想ドメインエントリは、`o=Cust2,O=root` の値を持つ inetDomainBaseDN 属性を含みます。

inetDomainStatus 属性がエントリにあり、その値が `inactive` または `deleted` の場合、エントリは無視されます。

変数

変数の中には、`obj.conf` ファイルで使うために `server.xml` で定義されているものがあります。次のファイルコードは `docroot` 変数を定義します。

```
<property name="docroot" value="/server/docs/class2/acme" />
```

`docroot` 変数を使うと、異なるドキュメントルートディレクトリを異なる仮想サーバーに割り当てることができます。この変数は、`obj.conf` ファイルで使われます。次に例を示します。

```
NameTrans fn=document-root root="$docroot"
```

この `docroot` 変数を使うと、同じ仮想サーバークラス内で異なるドキュメントルートを異なる仮想サーバーに定義することができます。

注	変数は、 <code>http-service</code> 、 <code>virtual-server-class</code> 、または <code>virtual-server</code> レベルで、 <code>server.xml</code> ファイルに定義する必要があります。 <code>http-service</code> または <code>virtual-server-class</code> レベルでデフォルト値を持つ変数を定義し、 <code>virtual-server</code> レベルでそれらの変数をオーバーライドすることをお勧めします。
---	---

変数の形式

次の正規表現と一致するとき、変数は `obj.conf` にあります。

```
¥$ [A-Za-z] [A-Za-z0-9_]*
```

この表現は、\$ の後に 1 つまたは複数のアルファベットおよび数値文字が続いていることを表します。`{property}` のような区切り形式はサポートされていません。正規の \$ 文字を取得するには、\$\$ を使って変数を置き換えます。

id 変数

特殊な変数 `id` は、`virtual-server` 要素内で常に利用でき、`id` 属性の値を参照します。この変数は事前に定義されているのでオーバーライドすることはできません。`id` 属性は仮想サーバーを一意に識別します。次に例を示します。

```
<property name=docroot value="/export/$id" />
```

親 `virtual-server` を含む `id` 属性が `myserver` の場合、`docroot` 変数の値は `/export/myserver` に設定されます。

その他の重要な変数

次の変数は、Sun ONE Application Server 設定のさまざまな部分で使われます。`$id` 変数とは異なり、これらの変数はサーバーで事前に定義されていないので、オーバーライドすることができます。

一般的な変数

次の表は、一般的な `server.xml` 変数を示しています。

表 2-92 一般的な変数

変数	説明
<code>docroot</code>	仮想サーバーのドキュメントルート。通常は、 <code>obj.conf</code> ファイル内の <code>document-root</code> 関数へのパラメータとして評価される
<code>accesslog</code>	仮想サーバーのアクセスログファイル

send-cgi 変数

次の表は、`obj.conf` ファイルの `send-cgi` 関数で使われる `server.xml` 変数の一覧です。

表 2-93 send-cgi 変数

変数	説明
<code>user</code>	<code>user</code> CGI パラメータの値
<code>group</code>	<code>group</code> CGI パラメータの値
<code>chroot</code>	<code>chroot</code> CGI パラメータの値

表 2-93 send-cgi 変数 (続き)

変数	説明
dir	dir CGI パラメータの値
nice	nice CGI パラメータの値

send-cgi 関数の詳細は、『Sun ONE Application Server Developer's Guide to NSAPI』を参照してください。

変数の評価

変数は各仮想サーバー固有のオブジェクトを生成するときに評価されます。評価は再帰的に行われます。変数値にはその他の変数を含めることができます。次に例を示します。

```
...
<virtual-server-class>
    ...
    <virtual-server ...>
        ...
        <property name=docroot value="$docrootbase/nonjava/$id" />
    </virtual-server>
</virtual-server ...>
    ...
    <property name=docroot value="$docrootbase/java/$id" />
</virtual-server>
    ...
    <property name=docrootbase value="/export" />
</virtual-server-class>
...
```

サブ要素内の変数は、親要素内の変数をオーバーライドします。たとえば、仮想サーバークラスの変数を設定し、その変数を、各仮想サーバー内の同じ変数の定義でオーバーライドすることができます。

server.xml ファイルのサンプル

Sun ONE Application Server の最初のインストール時には、server.xml ファイルは次のようになっています。

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- server_7_0_dtd version: 1.1.2.39 -->

<!DOCTYPE server PUBLIC "-//Sun Microsystems Inc.//DTD Sun ONE Application Server
7.0//EN" "file:///C:/Sun/AppServer7/lib/dtds/sun-server_1_1.dtd">

<server name="server1" log-root="C:/Sun/AppServer7/domains/domain1/server1/logs"
application-root="C:/Sun/AppServer7/domains/domain1/server1/applications"
session-store="C:/Sun/AppServer7/domains/domain1/server1/session-store">

    <http-service qos-enabled="false">

        <http-listener id="http-listener-1" address="0.0.0.0" port="1024"
acceptor-threads="1" blocking-enabled="false" security-enabled="false"
default-virtual-server="server1" server-name="goliath" enabled="true"/>

        <mime id="mime1" file="mime.types"/>

        <acl id="acl1"
file="C:/Sun/AppServer7/domains/domain1/server1/config/generated.server1.acl"/>

        <virtual-server-class id="defaultclass" config-file="obj.conf"
default-object="default" accept-language="false" enabled="true">

            <virtual-server id="server1" http-listeners="http-listener-1"
hosts="goliath" mime="mime1" state="on" accept-language="false">

                <auth-db id="default" database="default"/>

                <property name="dir" value=""/>

                <property name="nice" value=""/>

                <property name="user" value=""/>

                <property name="group" value=""/>

                <property name="chroot" value=""/>

                <property name="docroot"
value="C:/Sun/AppServer7/domains/domain1/server1/docroot"/>

                <property name="accesslog"
value="C:/Sun/AppServer7/domains/domain1/server1/logs/access"/>

            </virtual-server>

        </virtual-server-class>

    </http-service>
```



```

    <iiop-service>
        <orb message-fragment-size="1024" steady-thread-pool-size="10"
max-thread-pool-size="200" idle-thread-timeout-in-seconds="300"
max-connections="1024" monitoring-enabled="false"/>

        <iiop-listener id="orb-listener-1" address="goliath" port="3700"
enabled="true">

            </iiop-listener>
        </iiop-service>

    <web-container monitoring-enabled="false" >

        </web-container>

    <ejb-container steady-pool-size="32" pool-resize-quantity="16"
max-pool-size="64" cache-resize-quantity="32" max-cache-size="512"
pool-idle-timeout-in-seconds="600" cache-idle-timeout-in-seconds="600"
removal-timeout-in-seconds="5400" victim-selection-policy="nru"
commit-option="B" monitoring-enabled="false">

        </ejb-container>

    <mdb-container steady-pool-size="10" pool-resize-quantity="2"
max-pool-size="60" idle-timeout-in-seconds="600" monitoring-enabled="false">

        </mdb-container>

    <jms-service port="7676" admin-user-name="admin" admin-password="admin"
init-timeout-in-seconds="30" enabled="true">

        </jms-service>

    <log-service file="C:/Sun/AppServer7/domains/domain1/server1/logs/server.log"
level="INFO" log-stdout="true" log-stderr="true"
echo-log-messages-to-stderr="true" create-console="true"
log-virtual-server-id="false" use-system-logging="false">

        </log-service>

    <security-service default-realm="file" anonymous-role="ANYONE"
audit-enabled="false">

        <auth-realm name="file"
classname="com.iplanet.ias.security.auth.realm.file.FileRealm">

            <property name="file"
value="C:/Sun/AppServer7/domains/domain1/server1/config/keyfile"/>

            <property name="jaas-context" value="fileRealm"/>

        </auth-realm>
    </security-service>

```

```

    <auth-realm name="ldap"
classname="com.ipplanet.ias.security.auth.realm.ldap.LDAPRealm">
        <property name="directory" value="ldap://localhost:389"/>
        <property name="base-dn" value="o=isp"/>
        <property name="jaas-context" value="ldapRealm"/>
    </auth-realm>

    <auth-realm name="certificate"
classname="com.ipplanet.ias.security.auth.realm.certificate.CertificateRealm">
    </auth-realm>

    <!--
    <auth-realm name="solaris"
classname="com.ipplanet.ias.security.auth.realm.solaris.SolarisRealm">
        <property name="jaas-context" value="solarisRealm"/>
    </auth-realm>
    -->

</security-service>

<transaction-service automatic-recovery="false" timeout-in-seconds="0"
tx-log-dir="C:/Sun/AppServer7/domains/domain1/server1/logs"
heuristic-decision="rollback" keypoint-interval="2048"
monitoring-enabled="false">

</transaction-service>

<java-config java-home="C:/Sun/AppServer7/jdk"
server-classpath="C:/Sun/AppServer7/lib/appserv-rt.jar;C:/Sun/AppServer7/jdk/lib/
tools.jar;C:/Sun/AppServer7/lib/appserv-ext.jar;C:/Sun/AppServer7/lib/appserv-cmp
.jar;C:/Sun/AppServer7/lib/appserv-ideplugin.jar;C:\Sun\%AppServer7%\imq\lib\imq.ja
r;C:\Sun\%AppServer7%\imq\lib\jaxm-api.jar;C:\Sun\%AppServer7%\imq\lib\imqadmin.jar;C:
\Sun\%AppServer7%\imq\lib\imqutil.jar;C:/Sun/AppServer7/lib/appserv-admin.jar;C:\Sun
\%AppServer7%\share\lib\jaxrpc-impl.jar;C:\Sun\%AppServer7%\share\lib\jaxrpc-api.jar;C:
\%Sun\%AppServer7%\share\lib\jaxr-impl.jar;C:\Sun\%AppServer7%\share\lib\jaxr-api.jar;
C:\Sun\%AppServer7%\share\lib\jhall.jar;C:\Sun\%AppServer7%\share\lib\activation.jar;C:
\%Sun\%AppServer7%\share\lib\mail.jar;C:\Sun\%AppServer7%\share\lib\saa-j-api.jar;C:\Su
n\%AppServer7%\share\lib\saa-j-impl.jar;C:\Sun\%AppServer7%\share\lib\commons-logging.j
ar;C:\Sun\%AppServer7%\imq\lib\fscontext.jar;C:\Sun\%AppServer7%\imq\lib/providerutil.
jar;C:/Sun/AppServer7/lib/appserv-jstl.jar"
classpath-suffix="C:/Sun/AppServer7/pointbase/client_tools/lib/pbclient42RE.jar"
env-classpath-ignored="true" debug-options="-Xdebug
-Xrunjdwp:transport=dt_socket,server=y,suspend=n" debug-enabled="false"
javac-options="-g">

```

```

        <jvm-options>-Dorg.xml.sax.parser=org.xml.sax.helpers.XMLReaderAdapter
    </jvm-options>

        <jvm-options>-Dorg.xml.sax.driver=org.apache.crimson.parser.XMLReaderImpl
    </jvm-options>

    <jvm-options>-Djava.security.policy=C:/Sun/AppServer7/domains/domain1/server1/conf
fig/server.policy</jvm-options>

    <jvm-options>-Djava.security.auth.login.config=C:/Sun/AppServer7/domains/domain1/
server1/config/login.conf</jvm-options>

    <jvm-options>-Dcom.sun.jdo.api.persistence.model.multipleClassLoaders=reload
    </jvm-options>

    <jvm-options>-Djava.util.logging.manager=com.ipplanet.ias.server.logging.ServerLog
Manager</jvm-options>

    <jvm-options>-Dcom.sun.aas.configRoot=C:/Sun/AppServer7¥config</jvm-options>
        <jvm-options>-Dcom.sun.aas.imqLib=C:¥Sun¥AppServer7¥imq¥lib</jvm-options>
        <jvm-options>-Dcom.sun.aas.imqBin=C:¥Sun¥AppServer7¥imq¥bin</jvm-options>

    <jvm-options>-Dcom.sun.aas.webServicesLib=C:¥Sun¥AppServer7¥share¥lib</jvm-option
s>

        <jvm-options>-Dsun.rmi.dgc.server.gcInterval=300000</jvm-options>

    <jvm-options> -Xrs -Xms128m -Xmx256m </jvm-options>

</java-config>

<resources>

</resources>

    <applications dynamic-reload-enabled="false"
dynamic-reload-poll-interval-in-seconds="2">

        </applications>

</server>

```

server.xml ファイルのサンプル

init.conf の構文と使い方

Sun ONE Application Server は起動時に、*instance_dir/config* ディレクトリの *init.conf* というファイルを検索して、サーバーインスタンスの動作と設定に関するグローバル変数セットを設定します。Sun ONE Application Server は *init.conf* で定義した指令をすべて実行します。

Init 関数を除く *init.conf* の指令では変数と値を 1 つずつ指定します。たとえば次のように指定します。

TempDir /tmp

指令の順序は重要ではありません。

注	ブール値の指令には次のような等価の値があります。on/yes/true と off/no/false。
---	---

注	<i>init.conf</i> ファイルを編集した場合、サーバーを再起動して変更内容を反映させる必要があります。
---	---

注	<i>init.conf</i> インタフェースは、不確定です。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。
---	---

この章では、Sun ONE Application Server 7 の `init.conf` で指定できるグローバル設定について示します。

この章には次の節があります。

- [Init 関数](#)
- [サーバー情報](#)
- [DNS 検索](#)
- [スレッド、プロセス、および接続](#)
- [ネイティブのスレッドプール](#)
- [CGI](#)
- [エラー記録](#)
- [ACL](#)
- [セキュリティ](#)
- [チャンクされたエンコード](#)
- [その他](#)

指令のアルファベット順のリストは、付録 C 「`init.conf` の指令のアルファベット順リスト」を参照してください。

注	ほとんどのファイルキャッシュ機能は <code>nsfc.conf</code> という設定ファイルで制御されます。 <code>nsfc.conf</code> については、 144 ページの「nsfc.conf」 と『Sun ONE Application Server Performance Tuning Guide』を参照してください。
---	---

Init 関数

Init 関数は、サーバーモジュールとプラグインを読み込んで初期化し、ログファイルを初期化します。これらの関数の詳細については、『Sun ONE Application Server Developer's Guide to NSAPI』を参照してください。

サーバー情報

この節では、サーバーに関する情報を指定する `init.conf` の指令について示します。次の指令があります。

- [NetSiteRoot](#)
- [TempDir](#)
- [TempDirSecurity](#)
- [User \(UNIX のみ\)](#)

NetSiteRoot

サーバーの `bin` ディレクトリと `lib` ディレクトリがある最上位ディレクトリへの絶対パス名を指定します。デフォルト値はないので値を指定する必要があります。

構文

`NetSiteRoot path`

TempDir

サーバーが一時ファイルとして使用するローカルボリュームのディレクトリを指定します。UNIX では、サーバーを実行するユーザーがそのディレクトリを所有し、書き込める必要があります。「[User \(UNIX のみ\)](#)」指令および「[TempDirSecurity](#)」指令も参照してください。

構文

`TempDir path`

デフォルト値

`/tmp (UNIX)`

TempDirSecurity

TempDir ディレクトリのセキュリティをサーバーがチェックするかどうかを指定します。UNIX では TempDirSecurity をオフ (TempDirSecurity off) にすると、サーバーは /tmp を一時ディレクトリとして使用します。

警告

UNIX では TempDirSecurity off を指定したり、/tmp を一時ディレクトリとして使用しないでください。/tmp を一時ディレクトリとして使用すると、セキュリティ上のリスクが大きくなります。

構文

TempDirSecurity [on|off]

デフォルト値

on

User (UNIX のみ)

User 指令ではサーバーの UNIX ユーザーアカウントを指定します。サーバーをスーパーユーザーまたはルートユーザーとして起動すると、指定したポートがバインドされ、ユーザー ID が User 指令で指定したユーザーアカウントに変更されます。サーバーを root で起動しない場合、この指令は無視されます。指定するユーザーアカウントには、サーバールートおよびサブディレクトリへの読み取りアクセス権を与える必要があります。ユーザーアカウントには、ログディレクトリへの書き込みアクセス権と CGI プログラムの実行権限を与える必要があります。ユーザーアカウントには設定ファイルに対する書き込み権を与えないようにします。これは、たとえ第三者がサーバーに侵入しても、設定ファイルを変更してマシンへアクセスすることを防ぐためです。nobody ユーザーを使用すると、十分なセキュリティが得られなくなる可能性があります。

構文

User *name*

name は 8 文字以下の UNIX ユーザーアカウントのログイン名です。

デフォルト値

User 指令がなければ、サーバーは起動時のユーザーアカウントで実行します。

例

User http

User server

User nobody

DNS 検索

この節では、DNS 検索に関係がある `init.conf` の指令を示します。次の指令があります。

- [AsyncDNS](#)
- [DNS](#)

AsyncDNS

非同期 DNS を使用するかどうかを指定します。この指令を有効にするには、DNS 指令を `on` に設定する必要があります。値は `on` または `off` です。DNS が有効であれば、非同期 DNS を有効にするとサーバーのパフォーマンスが向上します。

デフォルト値

デフォルトは `off` です。

DNS

DNS 指令では、サーバーにアクセスするクライアント上でサーバーが DNS 検索を実行するかどうかを指定します。クライアントがサーバーに接続すると、サーバーはクライアントの IP アドレスを認識しますが、クライアントのホスト名を認識しません (たとえばサーバーはクライアントを `198.95.251.30` として認識できてもホスト名の `www.a.com` を認識できません)。サーバーはクライアントの IP アドレスからホスト名を解決し、アクセスコントロール、CGI、エラーレポート、アクセスログなどを実行します。

サーバーが 1 日に応答する要求が多ければ、ホスト名の解決を停止して DNS サーバーや NIS サーバーの負荷を減らすこともできます。

構文

DNS [`on|off`]

デフォルト値

DNS ホスト名の解決は、デフォルトで `off` になっています。

例

DNS on

スレッド、プロセス、および接続

Sun ONE Application Server 7 では、待機ソケットの受け入れスレッドは接続すると接続キューで待機します。次にセッションスレッドがキューから接続を受け取って要求を処理します。セッションスレッドは、必要に応じてさらに多くのセッションスレッドを要求の最後に登録します。新しいスレッドは、接続キューが次のような状態の場合に追加されます。

- 新しい接続が返されるたびに、キューで待機している接続数 (接続のバックログ) と作成済みセッションスレッドの数が比較されます。待機している接続の数が作成済みスレッドの数よりも多ければ、次の要求完了時にスレッドがさらに追加されます。
- 前のバックログが追跡されます。バックログが時間の経過とともに増えていて、その増え方が `ThreadIncrement` の値よりも大きく、セッションスレッドからバックログを引いた数が `ThreadIncrement` の値よりも小さければ、`ThreadIncrement` で指定したスレッド数が新たに追加されます。
- 新しいセッションスレッドを追加するプロセスが、`RqThrottle` によって厳密に制限されます。
- ベンチマーク負荷の開始などによってバックログが急に増えても、作成されるスレッドが多くならないようにするために、16 回または 32 回ごとの接続時にスレッドが必要かどうかが決まります。この回数は既に存在するセッションスレッドの数によって決まります。

この節では、スレッド、プロセスおよび接続数とそれぞれのタイムアウトに関係がある `init.conf` の指令を示します。次の指令があります。

- [ConnQueueSize](#)
- [HeaderBufferSize](#)
- [IOTimeout](#)
- [KeepAliveThreads](#)
- [KeepAliveTimeout](#)
- [ListenQ](#)
- [MaxKeepAliveConnections](#)
- [PostThreadsEarly](#)
- [RcvBufSize](#)
- [RqThrottle](#)
- [RqThrottleMin](#)
- [SndBufSize](#)

- [StackSize](#)
- [StrictHttpHeaders](#)
- [TerminateTimeout](#)
- [ThreadIncrement](#)

ネイティブスレッドのプールを制御する指令については、[119 ページ](#)の「[ネイティブのスレッドプール](#)」も参照してください。

ConnQueueSize

アプリケーションサーバーが持てる未処理の接続数 (サービスが実行されていない接続数) を指定します。この値は、オペレーティングシステムで規定されている 1 プロセス当たりのオープンファイル記述子の最大値を超えるように設定する必要があります。

デフォルト値

デフォルト値は 5000 です。

HeaderBufferSize

クライアントからの要求データを読み取るために、各要求処理スレッドが使用するバッファのサイズ (バイト単位)。要求処理スレッドの最大値は [RqThrottle](#) で設定します。

デフォルト値

デフォルト値は 8192 (8 KB) です。

IOTimeout

クライアントから到着するデータをサーバーが待機する秒数を指定します。タイムアウトとなる前にデータが到着しなかった場合、接続は終了します。デフォルトの 30 秒より短く設定すると、スレッドを早く解放できます。ただし、接続速度が遅いユーザーの接続も切断してしまうことになります。

構文

IOTimeout *seconds*

デフォルト値

ハードウェア暗号化デバイスを使用しない場合は 30 秒、暗号化デバイスを使用する場合は 300 秒です。

KeepAliveThreads

この指令では、キープアライブサブシステムのスレッド数を指定します。推奨値は、システム上のプロセッサ数の数倍の値です。たとえば、CPU が 2 つあるシステムでは、キープアライブのスレッド数は 2 または 4 にします。この値は、キープアライブ接続の最大許容数 (**MaxKeepAliveConnections**) も考慮して設定する必要があります。

デフォルト値

1

KeepAliveTimeout

この指令では、サーバーが HTTP キープアライブ接続を開いている最大時間、またはクライアントとサーバー間の接続が持続している最大時間を指定します。以前のバージョンでは、サーバーがクライアント要求を処理している間はキープアライブ機能によってクライアントとサーバー接続を開いたままにしておくことができました。デフォルトの接続は、持続接続 (サーバーが接続を閉じるまでは開いたまま) であるか、または **KeepAliveTimeout** で指定した時間よりも長く接続が開いたままになっています。

接続がキープアライブサブシステムに移ると、タイムアウトへの秒読みが開始されます。タイムアウトまでに接続でアクティビティがない場合は、接続が閉じられます。

デフォルト値

デフォルト値は 30 秒です。最大値は 300 秒 (5 分) です。

ListenQ

待機ソケットで保留される接続の最大数を指定します。バックログキューがいっぱいになっている待機ソケットがタイムアウトになると接続は失敗します。

デフォルト値

デフォルト値は 128

MaxKeepAliveConnections

サーバーが同時に開くことのできるキープアライブ接続および持続的な接続の最大数を指定します。値の範囲は 0 ～ 32768 です。

デフォルト値

256

PostThreadsEarly

この指令が 1 (オン) に設定されていると、接続を受け入れてから要求に対する応答を送信するまでの間、待機ソケットの最小スレッド数が有効かどうかをサーバーがチェックします。データベース接続のように処理に時間のかかる要求をサーバーが扱う場合に、この指令を使用します。

デフォルト値

0 (オフ)

RcvBufSize

ソケットが使用する受信バッファのサイズをバイト単位で指定します。使用できる値は、オペレーティングシステムによって決まります。

デフォルト値

デフォルト値はオペレーティングシステムによって決まります。通常は 4096 (4K)、または 8192 (8K) がデフォルトです。

RqThrottle

サーバーが同時に扱える要求処理スレッドの最大数を指定します。それぞれの要求は、各スレッドで実行されます。

サーバーの設定やパフォーマンスのチューニングについては、『Sun ONE Application Server Performance Tuning Guide』で詳しく説明されています。

デフォルト値

128

RqThrottleMin

サーバー起動時に作成される要求処理スレッドの数を指定します。サーバーの負荷が増すにつれて、作成される要求処理スレッドが多くなります。RqThrottle スレッドの最大数まで作成できます。

デフォルト値

48

SndBufSize

ソケットが使用する送信バッファのサイズをバイト単位で指定します。

デフォルト値

デフォルト値はオペレーティングシステムによって決まります。通常は 4096 (4K)、または 8192 (8K) がデフォルトです。

StackSize

各要求処理スレッドの最大スタックサイズを指定します。

デフォルト値

各マシンに適したマシン固有のスタックサイズです。

StrictHttpHeaders

厳密な HTTP ヘッダーチェックを制御します。厳密な HTTP ヘッダーチェックがオンの場合、サーバーは不適切にヘッダーが重複している接続を拒否します。

構文

StrictHttpHeaders [on|off]

デフォルト値

off

TerminateTimeout

サーバーがシャットダウンするまでに、既存のすべての接続が切断するのを待機する時間を指定します。

デフォルト値

30 秒

ThreadIncrement

サーバー負荷の増加に対応するための追加または新規要求処理スレッド数です。要求処理キューに入っている保留接続数がアイドル状態の要求処理数より大きい場合などに追加されます。

サーバーの起動時に RqThrottleMin の数だけ要求処理スレッドが作成されます。負荷が増すと RqThrottle 指令による要求処理スレッドが作成されるまで ThreadIncrement 指令による追加要求処理スレッドが作成されます。

デフォルト値

デフォルト値は 10 です。

ネイティブのスレッドプール

この節では、ネイティブのスレッドプールのサイズを制御する指令を示します。このスレッドプールは、ネイティブの OS レベルのスレッドからのみ構成されています。すべてのスレッドは OS レベルであるため、通常、UNIX ではネイティブのプールは使用されません。UNIX でネイティブプールを使用するとコンテキストの切り替えが必要になり、パフォーマンスオーバーヘッドが少し増えますが、このネイティブプールを `stickyAttach` 効果のローカライズ、リソースの制御と管理、プラグインの単独スレッド動作のエミュレートなどの目的に使用できます。

`stickyAttach` は、`server.xml` ファイルの `java-config` 要素のプロパティとして次のように指定できます。

```
<java-config>
...
  <property name="stickyAttach" value="1" >
</java-config>
```

次の指令があります。

- [NativePoolStackSize](#)
- [NativePoolMaxThreads](#)
- [NativePoolMinThreads](#)
- [NativePoolQueueSize](#)

NativePoolStackSize

ネイティブスレッドプールのスレッドごとにスタックサイズを指定します。

デフォルト値

0 (オペレーティングシステム固有のデフォルト値)

NativePoolMaxThreads

ネイティブスレッドプールの最大スレッド数を指定します。

デフォルト値

128

NativePoolMinThreads

ネイティブスレッドプールの最小スレッド数を指定します。

デフォルト値

1

NativePoolQueueSize

スレッドプールのキューに待機できるスレッド数を指定します。プールのスレッドがすべてビジー状態の場合、次の要求処理スレッドはネイティブプールのスレッドが空くまでキューで待機します。キューが満杯の場合は次の要求処理スレッドがキューに入ろうとしても拒否され、クライアントにはビジー応答が返されます。その後、スレッドは解放され、キューで待機状態になっていた要求を処理します。

デフォルト値

0 (キューのサイズは無制限)

CGI

この節では、CGI プログラムの要求と関係がある `init.conf` の指令を示します。次の指令があります。

- [CGIExpirationTimeout](#)
- [CGIStubIdleTimeout](#)
- [MaxCGIStubs](#)
- [MinCGIStubs](#)

CGIExpirationTimeout

この指令では、CGI プロセスが終了される前に実行できる最大秒数を指定します。

`CGIExpirationTimeout` の値はあまり小さくしないようにします。対話型の CGI では 300 秒 (5 分) が適当です。ただし処理に時間がかかる CGI の場合は、CGI プログラムが処理に必要とする最大持続時間を指定します。値を 0 に設定すると CGI の有効期限がなくなり、CGI の処理時間は無制限になります。

デフォルト値

0 (無制限)

CGIStubIdleTimeout

指令で設定した秒数の間、アイドル状態にある `CGIStub` プロセスをサーバーで終了します。プロセスの数が `MinCGIStubs` になると、サーバーはそれ以上にプロセスを終了しなくなります。

デフォルト値

30

MaxCGIStubs

サーバーが生成できる `CGIStub` プロセスの最大数を制御します。この値は、同時に実行できる `CGIStub` プロセス数の最大値であり、保留中の要求の最大数とは異なります。デフォルト値はほとんどのシステムに適合します。この値を大きくしすぎると実際のスループットが減少します。

デフォルト値

10

MinCGIStubs

デフォルトで起動するプロセスの数を制御します。最初の CGIStub プロセスは CGI プログラムが呼び出されると起動します。init.conf ファイルに init-cgi 指令があれば、最小数の CGIStub プロセスが起動時に生成されます。この値は MaxCGIStubs の値より小さくする必要があります。

デフォルト値

2

エラー記録

この節では、エラーログに関係がある指令を示します。次の指令があります。

- [ErrorLogDateFormat](#)
- [LogFlushInterval](#)
- [PidLog](#)

ErrorLogDateFormat

ErrorLogDateFormat 指令では、サーバーログで使用する日付形式を指定します。

構文

ErrorLogDateFormat *format*

format には[付録 A 「日時の形式」](#)の形式がすべて使用できます。

デフォルト値

%d/%b/%Y:%H:%M:%S

LogFlushInterval

この指令では、ログフラッシュスレッドの間隔を秒単位で指定します。

デフォルト値

30

PidLog

PidLog では、ベースサーバープロセスのプロセス ID (pid) を記録するファイルを指定します。サーバーサポートプログラムによっては、このログがサーバールート logs/pid にあると見なす場合もあります。

サーバーをシャットダウンするには、`-TERM` シグナルを使って `pid` ログファイルに表示されているベースサーバープロセスを終了します。設定ファイルを再び読み取ってログファイルを再度開くようサーバーに指示するには、`kill` を `-HUP` シグナルとともに使用します。

サーバーのユーザーアカウントを使って `PidLog` ファイルに書き込めない場合、サーバーにはプロセス ID を記録する場所がありません。プロセス ID が記録できなければサーバーは起動しません。

構文

`PidLog file`

file はプロセス ID が保存されているファイルへの絶対パスとそのファイル名です。

デフォルト値

デフォルトはありません。

例

```
PidLog /var/ns-server/logs/pid
```

```
PidLog /tmp/ns-server.pid
```

ACL

この節では、アクセス制御リスト (ACL) に関連する `init.conf` の指令を示します。次の指令があります。

- [ACLCacheLifetime](#)
- [ACLUserCacheSize](#)
- [ACLGroupCacheSize](#)

ACLCacheLifetime

`ACLCacheLifetime` は、キャッシュエントリが無効になるまでの時間を秒数で指定します。キャッシュのエントリが参照されるたびに、経過時間が計算され `ACLCacheLifetime` と照合されます。経過時間が `ACLCacheLifetime` 以上の場合、このエントリは使用されません。この値を 0 にすると、キャッシュが無効になります。

この値を大きくすると、LDAP エントリを変更した場合に Sun ONE Application Server の再起動が必要になることがあります。たとえば 120 秒に設定すると、Sun ONE Application Server は 2 分間 LDAP サーバーと同期が取れなくなる可能性があります。LDAP が頻繁に変更されない場合は、大きな値を設定できます。

デフォルト値
120

ACLUserCacheSize

`ACLUserCacheSize` は、ユーザーキャッシュのユーザー数を指定します。

デフォルト値
200

ACLGroupCacheSize

`ACLGroupCacheSize` は、1 つの UID/ キャッシュエントリに対してキャッシュできるグループ ID の数を指定します。

デフォルト値
4

セキュリティ

この節では、Sun ONE Application Server のサーバーアクセスとセキュリティに関する `init.conf` の指令を示します。次の指令があります。

- [Security](#)
- [SSLCacheEntries](#)
- [SSLClientAuthDataLimit](#)
- [SSLClientAuthTimeout](#)
- [SSLSessionTimeout](#)
- [SSL3SessionTimeout](#)

Security

`Security` 指令で SSL をグローバルに有効または無効にするには、サーバーインスタンスへの証明書を有効にします。仮想サーバーで SSL を使用する場合は `on` にしておきます。 `on` に設定した場合、ユーザーは次の入力をするよう要求されます。

- サーバーの秘密キーが含まれている信頼データベースへのパスワード
- インストールされた暗号化されたハードウェアに必要な PIN

注	管理インタフェースでセキュリティ保護された待機ソケットを作成すると、 <code>init.conf</code> のグローバルなセキュリティが自動的に有効になります。 <code>server.xml</code> でセキュリティ保護された待機ソケットを手動で作成する場合は、 <code>init.conf</code> を編集してセキュリティをオンにします。
---	--

仮想サーバーの SSL を有効にする方法については、[第 2 章「サーバー設定ファイル」](#)を参照してください。

構文

`Security [on|off]`

デフォルト値

`off`

例

`Security off`

SSLCacheEntries

キャッシュできる SSL のセッション数を指定します。上限はありません。

構文

`SSLCacheEntries number`

number が 0 の場合は、デフォルト値の 10000 が使用されます。

SSLClientAuthDataLimit

クライアント証明書のハンドシェークフェーズ時にバッファに入れるアプリケーションデータの最大数をバイト単位で指定します。

デフォルト値

デフォルト値は 1048576 (1M バイト) です。

SSLClientAuthTimeout

クライアント証明書のハンドシェークフェーズのタイムアウト時間を秒単位で指定します。

デフォルト値

60

SSLSessionTimeout

`SSLSessionTimeout` 指令は、SSL2 セッションのキャッシュ化を制御します。

構文

`SSLSessionTimeout seconds`

seconds 値は、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です。
`SSLSessionTimeout` 指令が指定されている場合、この秒数の値は暗黙的に 5 ～ 100 秒に制限されます。

デフォルト値

デフォルト値は 100 です。

SSL3SessionTimeout

SSL3SessionTimeout 指令は、SSL3 セッションのキャッシュ化を制御します。

構文

SSL3SessionTimeout *seconds*

seconds 値は、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です。デフォルト値は 86400 (24 時間) です。SSL3SessionTimeout 指令が指定されている場合、この秒数の値は暗黙的に 5 ～ 86400 秒に制限されます。

チャンクされたエンコード

この節では、チャンクされたエンコードを制御する指令を示します。

- [UseOutputStreamSize](#)
- [ChunkedRequestBufferSize](#)
- [ChunkedRequestTimeout](#)

これらの指令は `obj.conf` の Service SAF パラメータと同等です。obj.conf パラメータはこれらの指令を無視します。詳細については、『Sun ONE Application Server NSAPI Developer's Guide』を参照してください。

UseOutputStreamSize

UseOutputStreamSize 指令では、`net_read` および `netbuf_grab` NSAPI 関数の出力ストリームバッファサイズのデフォルト値を指定します。これらの関数の詳細については、『Sun ONE Application Server NSAPI Developer's Guide』を参照してください。

注	obj.conf ファイルの UseOutputStreamSize パラメータをゼロに設定すると、出力ストリームバッファを無効にできます。init.conf ファイルの UseOutputStreamSize をゼロに設定しても無効になりません。
---	--

構文

UseOutputStreamSize *size*

size の値はバイト数です。

デフォルト値

デフォルト値は 8192 (8K バイト) です。

ChunkedRequestBufferSize

ChunkedRequestBufferSize 指令では、チャンクしない要求データのデフォルトバッファサイズを指定します。

構文

ChunkedRequestBufferSize *size*

size の値はバイト数です。

デフォルト値

デフォルト値は 8192 です。

ChunkedRequestTimeout

ChunkedRequestTimeout 指令では、チャンクしない要求データのデフォルトタイムアウトを指定します。

構文

ChunkedRequestTimeout *seconds*

seconds の値は秒数です。

デフォルト値

デフォルト値は 60 (1 分) です。

その他

この節では、`init.conf` のその他の指令を示します。

- [ChildRestartCallback](#)
- [HTTPVersion](#)
- [MaxRqHeaders](#)
- [ReentrantTimeFunctions](#) (Solaris のみ)
- [Umask](#) (UNIX のみ)

ChildRestartCallback

この指令は、サーバーの再起動時やシャットダウン時に `daemon_atrestart` NSAPI 関数を使って登録された NSAPI 関数をコールバックさせます。値は `on`、`off`、`yes`、`no`、`true`、または `false` です。`daemon_atrestart` の詳細については、『Sun ONE Application Server NSAPI Developer's Guide』を参照してください。

デフォルト値

No

HTTPVersion

現在サーバーが使用している HTTP バージョンは、`m.n` の形式で表します。`m` はバージョンのメジャー番号で、`n` はマイナー番号です。

デフォルト値

デフォルト値は 1.1

MaxRqHeaders

要求のヘッダー行の最大行数を指定する。0 ～ 512 の値を取ります。

デフォルト値

64

ReentrantTimeFunctions (Solaris のみ)

ReentrantTimeFunctions 指令では、サーバーがサーバーの再入可能な時間フォーマット実装をオペレーティングシステムの実装の代わりに使用するかどうかを指定します。サーバーの再入可能な実装を使うと、多数の CPU が搭載されているコンピュータのパフォーマンスがあまり向上しないことがあります。

デフォルト値は `off` で、サーバーはオペレーティングシステムの実装を使用します。

デフォルト値

`off`

Umask (UNIX のみ)

この指令では、NSAPI 関数 `system_fopenWA` および `system_fopenRW` で使用する `umask` 値を指定して、異なるモードでファイルを開きます。この指令で有効な値は標準の UNIX `umask` 値です。

これらの関数の詳細については、『Sun ONE Application Server NSAPI Developer's Guide』の `system_fopenWA` および `system_fopenRW` を参照してください。

MIME タイプ

この章では、MIME タイプファイルについて説明します。次の節があります。

- [はじめに](#)
- [MIME タイプの決定](#)
- [Type の応答への影響](#)
- [クライアントの行う MIME タイプの処理](#)
- [MIME タイプファイルの構文](#)
- [MIME タイプファイルの例](#)

`mime.types` ファイルは `obj.conf` ファイルと相互に作用します。`obj.conf` の詳細については、『Sun ONE Application Server Developer's Guide to NSAPI』を参照してください。

注	<code>mime.types</code> インタフェースは不確定です。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。
---	---

はじめに

`instance_dir/config` ディレクトリにある MIME タイプファイルには、MIME (Multipurpose Internet Mail Extensions) タイプとファイル拡張子との間のマッピングが含まれています。たとえば、MIME タイプファイルは次のように拡張子 `.html` や `.htm` を `text/html` タイプにマッピングします。

```
type=text/html exts=htm,html
```

Sun ONE Application Server はクライアントから Web のリソース要求を受け取ると、MIME タイプマッピングを使って要求されているリソースの種類を判断します。

MIME タイプには、`lang` (言語を表す)、`enc` (エンコーディングを表す)、それに `type` (コンテンツのタイプを表す) の 3 つの属性が定義されています。各タイプにこれらの属性の 1 つ以上が存在する必要があります。最もよく使われる属性は、`type` です。サーバーは、通常、クライアントへの応答の生成方法を決定する際 `type` を使います。`enc` および `lang` 属性を使うことはほとんどありません。

デフォルトの MIME タイプファイルは `mime.types` と呼ばれます。

MIME タイプの決定

要求処理プロセスの `ObjectType` ステップで、サーバーはクライアントが要求するリソースの MIME タイプ属性を判断します。MIME タイプの決定にはいくつかの異なる関数が使われますが、通常使われるのは `type-by-extension` です。この関数は、MIME タイプテーブルにある要求されたリソースのファイル拡張子に従って MIME タイプを検索するようサーバーに伝えます。

次に拡張子に従って MIME タイプを検索するようサーバーに伝える `obj.conf` の指令を示します。

```
ObjectType fn=type-by-extension
```

サーバーが `force-type` のような別の関数を使って `type` を決定する場合、特定の要求のために MIME タイプテーブルが使われることはありません。

Type の応答への影響

サーバーは、`obj.conf` でクライアントへの応答の生成に使う `Service` 指令を決定するときに、`type` 属性の値を考慮します。

デフォルトでは、`type` が `magnus-internal/` で始まっていない場合、サーバーは要求されたファイルをクライアントに送信するだけです。この命令を含む `obj.conf` の指令は、次のとおりです。

```
Service method=(GET|HEAD|POST) type=~magnus-internal/* fn=send-file
```

慣例では、要求されたリソースをクライアントに送信すること以外の作業をサーバーに要求する `type` のすべての値は、`magnus-internal/` で始まります。

たとえば、要求されたリソースのファイル拡張子が `.map` の場合、タイプは `magnus-internal/imagemap` にマッピングされます。拡張子が `.cgi`、`.exe` または `.bat` の場合、タイプは `magnus-internal/cgi` にマッピングされます。

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

`type` が `magnus-internal/` で始まっている場合、サーバーは `obj.conf` にある `Service` 指令のうち、指定されたタイプと一致するすべてを実行します。たとえば、タイプが `magnus-internal/imagemap` の場合、サーバーは次の指令の指示に従い `imagemap` 関数を使ってクライアントへの応答を生成します。

```
Service method=(GET|HEAD) type=magnus-internal/imagemap fn=imagemap
```

クライアントの行う MIME タイプの処理

Service 関数はデータを生成し、要求元のクライアントに送信します。サーバーはクライアントに、データとヘッダーを送信します。これらのヘッダーには、既知のすべての MIME タイプ属性 (通常は type) が含まれます。

クライアントはデータを受信すると、MIME タイプを使ってデータの処理内容を決定します。ブラウザクライアントでは、データは通常ブラウザウィンドウに表示されます。

要求されたリソースをブラウザに表示できず、ほかのアプリケーションで処理する必要がある場合、その type は application/ から始まります。たとえば、.bin ファイル拡張子では application/octet-stream、.fm ファイル拡張子では application/x-maker のようになります。クライアントにはユーザーが編集可能なマッピングの独自のセットがあり、どのアプリケーションを使ってどのデータを処理するのかを伝えます。

たとえば、type が application/x-maker の場合、クライアントは通常 Adobe FrameMaker を開いてファイルを表示し、処理します。

MIME タイプファイルの構文

MIME タイプファイルの最初の行では、次のようにしてファイル形式を明示します。

```
#--Sun Microsystems MIME Information
```

コメント以外の行は次の形式にします。

```
type=type/subtype exts=[file extensions]
```

- type/subtype はタイプとサブタイプです。
- exts は、このタイプに関連づけられたファイル拡張子です。

MIME タイプファイルの例

MIME タイプファイルの例を次に示します。

```

#--Sun Microsystems MIME Information
# 上の行は削除しないでください。ファイルタイプの識別に使われます。
type=application/octet-stream      exts=bin,exe
type=application/oda                exts=oda
type=application/pdf                exts=pdf
type=application/postscript         exts=ai,eps,ps
type=application/rtf                exts=rtf
type=application/x-mif              exts=mif,fm
type=application/x-gtar             exts=gtar
type=application/x-shar             exts=shar
type=application/x-tar              exts=tar
type=application/mac-binhex40       exts=hqx

type=audio/basic                    exts=au,snd
type=audio/x-aiff                   exts=aif,aiff,aifc
type=audio/x-wav                    exts=wav

type=image/gif                      exts=gif
type=image/ief                      exts=ief
type=image/jpeg                     exts=jpeg,jpg,jpe
type=image/tiff                     exts=tiff,tif
type=image/x-rgb                     exts=rgb
type=image/x-xbitmap                 exts=xbm
type=image/x-xpixmap                 exts=xpm
type=image/x-xwindowdump             exts=xwd

type=text/html                      exts=htm,html
type=text/plain                     exts=txt
type=text/richtext                   exts=rtx
type=text/tab-separated-values       exts=tsv
type=text/x-setext                   exts=etx

type=video/mpeg                     exts=mpeg,mpg,mpe
type=video/quicktime                 exts=qt,mov
type=video/x-msvideo                 exts=avi

enc=x-gzip                           exts=gz
enc=x-compress                       exts=z
enc=x-uencode                        exts=uu,uue

type=magnus-internal/imagemap        exts=map
type=magnus-internal/parsed-html     exts=shtml
type=magnus-internal/cgi              exts=cgi,exe,bat
type=magnus-internal/jsp              exts=jsp

```

MIME タイプファイルの例

その他の設定ファイル

本章では、他の章では触れていない重要な設定ファイルを対象に、その**目的**と**場所**、**内容**や**構文**を要約し、各ファイルで利用できるすべての指令とパラメータを含めて、表形式で簡潔に説明しています。他の章またはマニュアルに指令やパラメータの詳細な説明がある場合、**関連項目**の見出しの後に相互参照を示します。変更してはならない設定ファイルはこの章のリストには記載されていません。

次の設定ファイルをアルファベット順に示します。

- [dbswitch.conf](#)
- [配備記述子](#)
- [generated.instance.acl](#)
- [loadbalancer.xml](#)
- [nsfc.conf](#)
- [password.conf](#)
- [server.policy](#)

dbswitch.conf

目的
Sun ONE Application Server が使用する LDAP ディレクトリを指定します。

注
dbswitch.conf インタフェースは、不確定です。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。

場所
instance_dir/config

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 を介して匿名バインドを実行するには、ファイルを次のように編集します。
directory default ldaps://directory.sun.com:636:/dc%3Dcom
SSL を介さないで匿名バインドを実行するには、ファイルを次のように編集します。
directory default ldap://directory.sun.com:389:/dc%3Dcom

関連項目
[98 ページの「ユーザーデータベースの選択」](#)

次の表に、dbswitch.conf ファイルのプロパティを示します。

表 5-1 dbswitch.conf のプロパティ

プロパティ	許容値	デフォルト値	説明
nsessions	正の整数	8	データベースへの LDAP 接続の数
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ほかのグループを含めることができる

表 5-1 dbswitch.conf のプロパティ (続き)

プロパティ	許容値	デフォルト値	説明
binddn	有効な DN		データベースへの接続に使用する DN。binddn と bindpw の両方がない場合、バインドは匿名
bindpw			データベースへの接続に使用するパスワード。binddn と bindpw の両方がない場合、バインドは匿名
dcsuffix	有効な DN (LDAP URL と関連して)	なし	指定した場合、要求の仮想サーバーの基本 DN のデフォルト値は、仮想サーバーの host 属性のうち、クライアントの提供する Host ヘッダと一致する属性の dcsuffix DN で始まる DC ツリー検索で決定される。host 属性が 1 つも一致しない場合、親 http-listener の server-name 属性が使われる 指定しない場合、基本 DN のデフォルト値は LDAP URL での base DN 値 server.xml ファイルの auth-db 要素の basedn 属性は、この値に優先する
digestauth	off、on	off	データベースがダイジェスト認証できる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ープラグインが必要。このプラグインのインストール方法の詳細については、『Sun ONE Application Server 管理者ガイド』を参照

LDAP データベースがスキーマに準拠している場合、クライアントの提供する Host ヘッダーと一致する virtual-server 要素の host 属性の DC ツリー検索を使って、アクセスの基本 DN が計算されます。host 属性が 1 つも一致しない場合、親 http-listener の server-name 属性が使われます。検索結果には、基本 DN を含む inetDomainBaseDN 属性が含まれている必要があります。この基本 DN は、そのまま取得され、いずれの基本 DN 値とも関連がありません。

配備記述子

目的
配備されたモジュールおよびアプリケーションに対応した Sun ONE Application Server の機能を設定します。

場所
モジュールまたはアプリケーションの META-INF または WEB-INF ディレクトリ

関連項目
次の表は、Sun ONE Application Server 配備記述子についての詳細情報の参照先を示しています。

表 5-2 Sun ONE Application Server 記述子

配備記述子	詳細情報の参照先
sun-application.xml	『Sun ONE Application Server Developer's Guide』
sun-web.xml	『Sun ONE Application Server Developer's Guide to Web Applications』
sun-ejb-jar.xml および sun-cmp-mapping.xml	『Sun ONE Application Server Developer's Guide to Enterprise JavaBeans Technology』
sun-application-client.xml および sun-acc.xml	『Sun ONE Application Server Developer's Guide to Clients』
sun-ra.xml	『Sun ONE J2EE CA Service Provider Implementation Administrator's Guide』

generated.instance.acl

目的

サーバーインスタンスへのアクセス権を設定します。これはデフォルトの ACL ファイルですが、ユーザーが作成したりほかのファイルを使うこともできます。

注

ACL ファイルインタフェースは、不確定です。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。

場所

`instance_dir/config`

関連項目

『Sun ONE Application Server セキュリティ管理者ガイド』

loadbalancer.xml

目的

クラスタとロードバランスの設定を行います。

注

loadbalancer.xml インタフェースは不確定です。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。

場所

loadbalancer.xml ファイルは、フロントエンドの Web サーバーにあります。

構文

ほとんどのファイルは、要素が入れ子になった、次のような基本的な XML 構文で記述されています。

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

「loadbalancer.xml の要素と属性」の表では、属性と区別しやすいように要素を太字で表記しています。

関連項目
『Sun ONE Application Server 管理者ガイド』

Sun ONE Web Server マニュアルセット

次の表では、loadbalancer.xml ファイルの要素、属性、およびプロパティについて説明します。

表 5-3 loadbalancer.xml の要素と属性

要素 / 属性	サブ要素またはデフォルト値	説明
loadbalancer	cluster 、 property	ロードバランサを定義する
cluster	instance 、 web-module 、 health-checker	アプリケーションサーバーインスタンスのクラスタを定義する
name	なし	クラスタの名前を指定する。クラスタ名はロードバランサ内で一意でなければならない
instance	なし	アプリケーションサーバーインスタンスを定義する
name	なし	サーバーインスタンスの名前を指定する。インスタンス名はクラスタ内で一意でなければならない
enabled	true	(省略可能) インスタンスがロードバランサを適用される要求に対してアクティブ (有効) かどうかを指定する
disable-timeout-in-minutes	31	(省略可能) 休止状態のタイムアウト間隔を指定する。この間隔で、ロードバランサはインスタンスを無効にし、アクティブな接続をすべて終了する
listeners	なし	空白文字で区切ってインスタンスのリスナーの URL を指定する。リスナーの URL はクラスタ内で一意でなければならない
web-module	なし	Web モジュールを定義する
context-root	なし	Web モジュールのコンテキストルート指定する。コンテキストルートはクラスタ内で一意でなければならない
enabled	true	(省略可能) Web モジュールがロードバランサを適用される要求に対してアクティブ (有効) かどうかを指定する

表 5-3 loadbalancer.xml の要素と属性 (続き)

要素 / 属性	サブ要素またはデフォルト値	説明
disable-timeout -in-minutes	31	(省略可能) 休止状態のタイムアウト間隔を指定する。この間隔で、ロードバランサは Web モジュールを無効にし、アクティブな接続をすべて終了する
health-checker	なし	クラスタのヘルスチェッカーを設定する
url	/	(省略可能) リスナーの状態を判断するため ping を実行する URL を指定する
interval-in-seconds	30	(省略可能) インスタンスのヘルスチェックを行う間隔を指定する
timeout-in-seconds	10	(省略可能) タイムアウト間隔を指定する。この間に応答が返されれば、リスナーは正常であると判断される
property	description	プロパティを定義する
name	なし	プロパティの名前を指定する
value	なし	プロパティの値を指定する
description		プロパティを説明するテキストを定義する

表 5-4 loadbalancer.xml のプロパティ

プロパティ名	デフォルト値	説明
response-timeout -in-seconds	60	タイムアウト間隔を指定する。この間にロードバランス要求の応答が返されなければ、インスタンスの状態は悪いと判断される
reload-poll- interval-in- seconds	0	ロードバランサが loadbalancer.xml のタイムスタンプが変更されたかどうかをチェックする間隔を指定する。変更された場合、ロードバランサはファイルを再度読み込む。0 の場合、ポーリングは無効
https-routing	false	ロードバランサが着信 HTTPS 要求を HTTPS 要求としてインスタンスに配信するかどうかを指定する
require-monitor-data	false	ロードバランサの監視を有効にするかどうかを指定する

nsfc.conf

目的
ファイルキャッシュのパラメータを設定します。このファイルは、ファイルのキャッシュパラメータがデフォルトから変更されている場合にのみ存在します。

注 nsfc.conf インタフェースは、不確定です。不確定なインタフェースは試験的または一時的なインタフェースであるため、次のリリースで互換性がなくなったり、削除されたり、または安定したインタフェースに置き換えられたりする場合があります。

場所
instance_dir/config

構文
parameter=value

関連項目
『Sun ONE Application Server Performance Tuning Guide』

次の表に、nsfc.conf ファイルのパラメータを示します。

表 5-5 nsfc.conf のパラメータ

パラメータ	許容値	デフォルト値	説明
FileCacheEnable	on、off	on	ファイルキャッシュを有効にする
CacheFileContent	on、off	on	MediumFileSizeLimit より小さい (TransmitFiles が on になっている場合は、SmallFileSizeLimit より小さい) ファイルについて、ファイル情報だけでなくファイルの内容のキャッシュも有効にする
MaxAge	秒数	30	有効なキャッシュエントリの最大生存期間。この設定は、ファイルがキャッシュされた後の、キャッシュされた情報の使用期間を制御する。MaxAge より古いエントリは、同じファイルの新しいエントリに置き換えられる
MediumFileSizeLimit	利用可能なメモリーサイズ	537600 (525K)	(UNIX のみ) TransmitFiles が off になっている場合に、メモリーマップファイルとしてキャッシュできる 1 つのファイルの最大サイズ

表 5-5 nsfc.conf のパラメータ (続き)

パラメータ	許容値	デフォルト値	説明
MediumFileSpace	利用可能な メモリーサ イズ	10485760 (10M)	TransmitFiles が off になっている場合 に、メモリーマップファイルとしてキャッ シュされるファイルの合計サイズ
SmallFileSizeLimit	利用可能な メモリーサ イズ	2048 (2K)	(UNIX のみ) メモリーに読み込めるファイ ルの最大サイズ
SmallFileSpace	利用可能な メモリーサ イズ	1048576 (UNIX、1M)	メモリーに読み込まれるすべてのファイル の合計サイズ
TransmitFiles	on、off	off (UNIX)	TransmitFile システム呼び出しの使用を 有効にする。IRIX、Compaq、Linux では サポートされない
MaxFiles		1024	ファイルキャッシュ内の最大ファイル数
HashInitSize	利用可能な メモリーサ イズ	0	ハッシュバケットの初期番号。0 の場合、 ハッシュバケットの番号は $2 * \text{MaxFiles} + 1$ として動的に決定される
TempDir	パス名	<i>system_temp/in stance</i>	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

password.conf

目的

デフォルトでは、アプリケーションサーバーの起動時に、管理者が SSL キーデータベースのパスワードを入力する必要があります。アプリケーションサーバーを人の介入なしで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する必要があります。システムを十分にセキュリティ保護して、このファイルとキーデータベースが危険にさらされないようにする必要があります。

場所

instance_dir/config

デフォルトでは、このファイルは存在しません。必要な場合はファイルを作成する必要があります。

構文

PKCS#11_module_name:password

サーバーに付属している内部 PKCS#11 ソフトウェア暗号化モジュールを使用している場合には、次の情報を入力します。

internal:password

それ以外の PKCS#11 モジュールを (たとえばハードウェアの暗号化またはハードウェアアクセラータ用に) 使用している場合は、PKCS#11 モジュールの名前を指定し、続けて次のようにパスワードを入力する必要があります。

internal:password

関連項目

『Sun ONE Application Server 管理者ガイド』

server.policy

目的

アプリケーションがアクセスするリソースを制御します。これは標準の J2SE ポリシーファイルです。

場所

instance_dir/config

構文

```
grant [codeBase "path"] {  
    permission permission_class "package", "permission_type";  
    ...  
};
```

関連項目

『Sun ONE Application Server Developer's Guide』

<http://java.sun.com/docs/books/tutorial/security1.2/tour2/index.html>

server.policy

日時の形式

ここでは、サーバーログの中で、日付と時刻の書式を定める文字列について説明します。`init.conf` の `ErrorLogDateFormat` 指令では、これらの形式を使います。

次の表に、日付と時刻の形式を示します。

表 A-1 日時の形式	
記号	意味
%a	曜日の名前 (3 文字の短縮形)
%d	月内の日付 (01 から 31 までの 10 進数)
%S	秒 (00 から 59 までの 10 進数)
%M	分 (00 から 59 までの 10 進数)
%H	時 (00 から 23 までの 24 時間形式)
%Y	年 (2099 までの 10 進数)
%b	月の名前 (3 文字の短縮形)
%h	月の名前 (3 文字の短縮形)
%T	"HH:MM:SS" 形式の時刻
%X	"HH:MM:SS" 形式の時刻
%A	曜日の名前 (完全形)
%B	月の名前 (完全形)
%C	"%a %b %e %H:%M:%S %Y"
%c	"%m/%d/%y %H:%M:%S" 形式の日付と時刻
%D	"%m/%d/%y" 形式の日付
%e	月内の日付 (先頭にゼロを付加しない、1 から 31 までの 10 進数)
%I	時 (01 から 12 までの 12 時間形式)

表 A-1 日時の形式 (続き)

記号	意味
%j	年内の日付 (001 から 366 までの 10 進数)
%k	時 (先頭にゼロを付加しない、0 から 23 までの 24 時間形式)
%l	時 (先頭にゼロを付加しない、1 から 12 までの 12 時間形式)
%m	月 (01 から 12 までの 10 進数)
%n	改行文字
%p	A.M./P.M. 表示 (12 時間形式の場合)
%R	"%H:%M" 形式の時刻
%r	"%I:%M:%S %p" 形式の時刻
%t	タブ文字
%U	年内の週 (00 から 51 までの 10 進数。日曜日を週の開始日とする)
%w	曜日 (0 から 6 までの 10 進数。日曜日を 0 とする)
%W	年内の週 (00 から 51 までの 10 進数。月曜日を週の開始日とする)
%x	"%m/%d/%y" 形式の日付
%y	年 (下 2 桁だけの、00 から 99 までの 10 進数)
%%	パーセント記号

サーバー設定要素のアルファベット順リスト

A

- `acl` [37](#)
- `admin-service` [30](#)
- `applications` [92](#)
- `auth-db` [44](#)
- `auth-realm` [71](#)
- `availability-service` [56](#)

C

- `connector-module` [97](#)
- `custom-resource` [81](#)

D

- `description` [30](#)

E

- `ejb-container` [50](#)
- `ejb-module` [95](#)
- `external-jndi-resource` [82](#)

H

H

[http-listener](#) 32

[http-qos](#) 43

[http-service](#) 31

I

[iiop-listener](#) 47

[iiop-service](#) 45

J

[j2ee-application](#) 94

[java-config](#) 75

[jdbc-connection-pool](#) 88

[jdbc-resource](#) 83

[jms-resource](#) 85

[jms-service](#) 66

[jvm-options](#) 79

L

[lifecycle-module](#) 93

[log-service](#) 68

M

[mail-resource](#) 84

[manager-properties](#) 60

[mdb-container](#) 53

mime [37](#)

O

orb [46](#)

P

persistence-manager-factory-resource [87](#)

persistence-store [58](#)

profiler [78](#)

property [29](#)

R

resources [80](#)

S

security-service [70](#)

server [27](#)

server-instance [30](#)

session-config [58](#)

session-manager [59](#)

session-properties [64](#)

ssl [35](#)

ssl-client-config [47](#)

store-properties [62](#)

T

T

transaction-service [73](#)

V

virtual-server [39](#)

virtual-server-class [38](#)

W

web-container [49](#)

web-module [96](#)

init.conf の指令のアルファベット順リスト

A

ACLCacheLifetime [124](#)
ACLGroupCacheSize [124](#)
ACLUserCacheSize [124](#)
AsyncDNS [113](#)

C

CGIExpirationTimeout [121](#)
CGIStubIdleTimeout [121](#)
ChildRestartCallback [129](#)
ChunkedRequestBufferSize [128](#)
ChunkedRequestTimeout [128](#)
ConnQueueSize [115](#)

D

DNS [113](#)

E

ErrorLogDateFormat [122](#)

H

H

HeaderBufferSize [115](#)

HTTPVersion [129](#)

I

IOTimeout [115](#)

K

KeepAliveThreads [116](#)

KeepAliveTimeout [116](#)

L

ListenQ [116](#)

LogFlushInterval [122](#)

M

MaxCGIStubs [121](#)

MaxKeepAliveConnections [116](#)

MaxRqHeaders [129](#)

MinCGIStubs [122](#)

N

NativePoolMaxThreads [120](#)

NativePoolMinThreads [120](#)

NativePoolQueueSize [120](#)

NativePoolStackSize [119](#)

NetSiteRoot [111](#)

P

PidLog [122](#)

PostThreadsEarly [117](#)

R

RcvBufSize [117](#)

ReentrantTimeFunctions (Solaris のみ) [130](#)

RqThrottle [117](#)

RqThrottleMin [117](#)

S

Security [125](#)

SndBufSize [118](#)

SSL3SessionTimeout [127](#)

SSLCacheEntries [126](#)

SSLClientAuthDataLimit [126](#)

SSLClientAuthTimeout [126](#)

SSLSessionTimeout [126](#)

StackSize [118](#)

StrictHttpHeaders [118](#)

T

TempDir [111](#)

U

TempDirSecurity [112](#)

TerminateTimeout [118](#)

ThreadIncrement [118](#)

U

Umask (UNIX のみ) [130](#)

UseOutputStreamSize [127](#)

User (UNIX のみ) [112](#)

索引

A

- accept-language 属性, 39, 42
- acceptor-threads 属性, 33
- accesslog 変数, 102
- ACLCacheLifetime init.conf 指令, 124
- ACLGroupCacheSize init.conf 指令, 124
- acls 属性, 42
- ACLUserCacheSize init.conf 指令, 124
- ACL ファイル, 37, 98
 - デフォルト, 141
 - の init.conf 指令, 124
- acl 要素, 37
- address 属性, 33, 48
- admin-password 属性, 66
- admin-service 要素, 30
- admin-user-name 属性, 66
- anonymous-role 属性, 70
- append-version プロパティ, 67
- application-root 属性, 28
- applications
 - location, 28
 - の設定, 94
- applications 要素, 92
- assign-groups プロパティ, 72
- AsyncDNS init.conf 指令, 113
- audit-enabled 属性, 71
- auth-db 要素, 44

- auth-realm 要素, 71
- automatic-recovery 属性, 74
- availability-enabled 属性, 57
- availability-service 要素, 56

B

- bandwidth-limit 属性, 44
- basedn 属性, 45
- base-dn プロパティ, 73
- binddn プロパティ, 139
- bindpw プロパティ, 139
- blocking-enabled 属性, 33
- bytecode-preprocessors 属性, 77

C

- CacheFileContent パラメータ, 144
- cache-idle-timeout-in-seconds 属性, 52
- cache-resize-quantity 属性, 51
- certmaps 属性, 45
- cert-nickname 属性, 35
- CGI
 - init.conf での設定, 121
 - の変数, 102
- CGIExpirationTimeout init.conf 指令, 121

D

CGIStubIdleTimeout init.conf 指令, 121
ChildRestartCallback init.conf 指令, 129
chroot 変数, 102
ChunkedRequestBufferSize init.conf 指令, 128
ChunkedRequestTimeout init.conf 指令, 128
class-name 属性, 93
classname 属性, 72
classpath-prefix 属性, 77
classpath-suffix 属性, 77
classpath 属性, 79, 94
client-auth-enabled 属性, 36
cluster 要素, 142
CMP, 87
cmt-max-runtime-exceptions プロパティ, 54
commit-option 属性, 53
config-file 属性, 39, 41
config ディレクトリ、位置, 17
connection-limit 属性, 44
connection-validation-method 属性, 90
connector-module 要素, 97
ConnQueueSize init.conf 指令, 115
context-root 属性, 96, 142
custom-resource 要素, 81

D

databaseName プロパティ, 91
database 属性, 45
datasource-classname 属性, 89
datasourceName プロパティ, 91
dbswitch.conf ファイル, 98, 99, 138
dcsuffix プロパティ, 139
debug-enabled 属性, 76
debug-options 属性, 76
default-object 属性, 39, 41
default-principal-password 属性, 70
default-principal 属性, 70

default-realm 属性, 70
default-virtual-server 属性, 34
default-web-module 属性, 41
description プロパティ, 91
description 要素, 30, 143
digestauth プロパティ, 139
directory プロパティ, 63, 72
dir 変数, 103
disable-distributed-transaction-logging プロパティ, 75
disable-timeout-in-minutes 属性, 142, 143
DNS init.conf 指令, 113
docroot 変数, 102
Domain Name Service、「DNS」を参照
dyngroups プロパティ, 138

E

echo-log-messages-to-stderr 属性, 69
ejb-container 要素, 50
ejb-module 要素, 95
enableCookies プロパティ, 65
enabled 属性, 142
enableURLRewriting プロパティ, 65
enc パラメータ, 132
enforce-bandwidth-limit 属性, 44
enforce-connection-limit 属性, 44
env-classpath-ignored 属性, 78
ErrorLogDateFormat init.conf 指令, 122
external-jndi-resource 要素, 82

F

factory-class 属性, 82, 83, 88
fail-all-connections 属性, 90
family 属性, 33

FileCacheEnable パラメータ, 144

file 属性, 69

acl 要素の, 38

mime 要素の, 37

file プロパティ, 72

G

group-base-dn プロパティ, 73

group-search-filter プロパティ, 73

group-target プロパティ, 73

group 変数, 102

H

HashInitSize パラメータ, 145

HeadBufferSize init.conf 指令, 115

health-checker 要素, 143

heuristic-decision 属性, 74

hosts 属性, 42

主題部のパターンとの比較, 40

http-listeners 属性, 41

http-listener 要素, 32

http-qos 要素, 43

http-service 要素, 31

https-routing プロパティ, 143

HTTPVersion init.conf 指令, 129

HTTP 待機ソケット, 32

HTTP 要求、処理, 18

HUP シグナル、PidLog および, 123

I

idLengthBytes プロパティ, 65

idle-thread-timeout-in-seconds 属性, 46

idle-timeout-in-seconds 属性, 54, 89

id 属性

acl 要素の, 38

auth-db 要素の, 45

http-listener 要素の, 33, 48

mime 要素の, 37

virtual-server-class 要素の, 38

virtual-server 要素の, 40

id 変数, 102

iiop-listener 要素, 47

sun-server_1_1.dtd ファイルの定義, 25

iiop-service 要素, 45

IIOP 待機ソケット, 47

imqBrokerHostName プロパティ, 86

imqBrokerHostPort プロパティ, 87

imqConfiguredClientID プロパティ, 87

imqDestinationName プロパティ, 86

init conf ファイル, 109

指令のアルファベット順リスト, 155

init.conf ファイル, 18

init-timeout-in-seconds 属性, 66

Init 関数, 111

instance-name-suffix プロパティ, 67

instance-name プロパティ, 67

instance 要素, 142

interval-in-seconds 属性, 143

IOTimeout init.conf 指令, 115

is-connection-validation-required 属性, 90

is-failure-fatal 属性, 94

is-isolation-level-guaranteed 属性, 90

J

j2ee-application 要素, 94

jaas-context プロパティ, 72

java-config 要素, 75

javac-options 属性, 76

Java Database Connectivity、「JDBC」を参照

java-home 属性, 76

JavaMail, [84](#)

Java Message Service、「JMS」を参照

Java Naming and Directory Interface、「JNDI」を参照

Java Platform Debugger Architecture、「JPDA」を参照

Java Transaction Service、「JTS」を参照

Java Virtual Machine、「JVM」を参照

jdbc-connection-pool 要素, [88](#)

jdbc-resource-jndi-name 属性, [88](#)

jdbc-resource 要素, [83](#)

JDBC 接続プール, [88](#)

JDBC リソース (JDBC resource), [83](#)

JMS, [66](#), [85](#)

jms-resource 要素, [85](#)

jms-service 要素, [66](#)

JNDI, [82](#)

jndi-lookup-name 属性, [82](#)

jndi-name 属性

custom-resource 要素の, [81](#)

external-jndi-resource 要素の, [82](#)

jdbc-resource 要素の, [83](#)

jms-resource 要素の, [86](#)

mail-resource 要素の, [84](#)

persistence-manager-factory-resource 要素の, [88](#)

JPDA デバッグオプション, [76](#)

JTS, [73](#)

JVM, [75](#)

サーバーへのオプションの追加, [79](#)

jvm-options 要素, [79](#)

K

KeepAliveThreads init.conf 指令, [116](#)

KeepAliveTimeout init.conf 指令, [116](#)

keypoint-interval 属性, [74](#)

L

lang パラメータ, [132](#)

LDAP, [98](#)

Sun ONE スキーマ, [99](#)

マッピングに使う証明書を指定, [45](#)

level 属性, [69](#)

lifecycle-module 要素, [93](#)

listeners 属性, [142](#)

ListenQ init.conf 指令, [116](#)

loadbalancer.xml ファイル, [141](#)

loadbalancer 要素, [142](#)

load-order 属性, [94](#)

locale 属性, [28](#)

location 属性, [94](#), [95](#), [97](#)

log-file 属性, [42](#)

LogFlushInterval init.conf 指令, [122](#)

log-level 属性, [69](#)

log-root 属性, [28](#)

log-service 要素, [68](#)

log-stderr 属性, [69](#)

log-stdout 属性, [69](#)

log-virtual-server-id 属性, [69](#)

M

mail-resource 要素, [84](#)

manager-properties 要素, [60](#)

MaxAge パラメータ, [144](#)

max-cache-size 属性, [52](#)

MaxCGIStubs init.conf 指令, [121](#)

max-connections 属性, [46](#)

MaxFiles パラメータ, [145](#)

MaxKeepAliveConnections init.conf 指令, [116](#)

max-pool-size 属性, [51](#), [54](#), [89](#)

MaxRqHeaders init.conf 指令, [129](#)

maxSessions プロパティ, [61](#)

max-thread-pool-size 属性, [46](#)

max-wait-time-in-millis 属性, 89
 mdb-container 要素, 53
 MediumFileSizeLimit パラメータ, 144
 MediumFileSpace パラメータ, 145
 message-fragment-size 属性, 46
 mime.types ファイル
 server.xml 要素の参照, 37
 概要, 19
 構文, 134
 サンプル, 134
 デフォルト, 132
 mime 属性, 42
 MIME タイプ, 131
 mime 要素, 37
 MinCGIStubs init.conf 指令, 122
 monitoring-enabled 属性, 46, 49, 53, 54, 74

N

name 属性, 142, 143
 auth-realm 要素の, 72
 connector-module 要素の, 97
 ejb-module 要素の, 95
 j2ee-application 要素の, 94
 jdbc-connection-pool 要素の, 89
 lifecycle-module 要素の, 93
 profiler 要素の, 79
 property 要素, 29
 server 要素, 28
 web-module 要素の, 96
 native-library-path- prefix 属性, 77
 native-library-path-suffix 属性, 77
 native-library-path 属性, 79
 NativePoolMaxThreads init.conf 指令, 120
 NativePoolMinThreads init.conf 指令, 120
 NativePoolQueueSize init.conf 指令, 120
 NativePoolStackSize init.conf 指令, 119
 networkProtocol プロパティ, 91
 nice 変数, 103

nsessions プロパティ, 138
 nsfc.conf ファイル, 144

O

obj.conf ファイル, 18, 41
 仮想サーバー, 18
 oracle-xa-recovery-workaround プロパティ, 75
 orb 要素, 46

P

password.conf ファイル, 146
 password プロパティ, 91
 persistenceFrequency プロパティ, 62
 persistence-manager-factory-resource 要素, 87
 persistenceScope プロパティ, 63
 persistence-store 要素, 58
 persistence-type 属性, 59
 PidLog init.conf 指令, 122
 pool-idle-timeout-in-seconds 属性, 52
 pool-name 属性, 83
 pool-resize-quantity 属性, 51, 54, 89
 port 属性, 33, 48, 66
 port プロパティ, 91
 PostThreadsEarly init.conf 指令, 117
 profiler 要素, 78
 property 要素, 29, 143

R

RcvBufSize init.conf 指令, 117
 reapIntervalSeconds プロパティ, 61
 reconnect-delay-in-seconds プロパティ, 55
 reconnect-enabled プロパティ, 55

reconnect-max-retries プロパティ, 55
 ReentrantTimeFunctions init.conf 指令, 130
 reload-poll-interval-in-seconds プロパティ, 143
 removal-timeout-in-seconds 属性, 52
 require-monitor-data プロパティ, 143
 resources 要素, 80
 response-timeout-in-seconds プロパティ, 143
 res-type 属性, 81, 82, 86, 89
 rmic-options 属性, 76
 roleName プロパティ, 91
 RqThrottle init.conf 指令, 117
 RqThrottleMin init.conf 指令, 117

S

search-bind-dn プロパティ, 73
 search-bind-password プロパティ, 73
 search-filter プロパティ, 73
 Secure Socket Layer、「SSL」を参照
 security-enabled 属性, 34
 Security init.conf 指令, 125
 security-service 要素, 70
 send-cgi 関数、の変数, 102
 server-classpath 属性, 77
 server-instance 要素, 30
 server-name 属性, 34
 serverName プロパティ, 91
 server.policy ファイル, 147
 server.xml ファイル, 18
 location, 23
 アルファベット順要素リスト, 151
 要素, 26
 例, 104
 server 要素, 27
 session-config 要素, 58
 sessionFilename プロパティ, 61
 session-manager 要素, 59
 session-properties 要素, 64
 session-store 属性, 28
 session-timeout 要素, 65
 SmallFileSizeLimit パラメータ, 145
 SmallFileSpace パラメータ, 145
 SndBufSize init.conf 指令, 118
 SSL
 init.conf での設定, 125
 設定, 35
 ssl2-ciphers 属性, 35
 ssl2-enabled 属性, 35
 ssl3-enabled 属性, 36
 SSL3SessionTimeout init.conf 指令, 127
 ssl3-tls-ciphers 属性, 36
 SSL3 クライアント認証, 36
 SSLCacheEntries init.conf 指令, 126
 SSLClientAuthDataLimit init.conf 指令, 126
 SSLClientAuthTimeout init.conf 指令, 126
 ssl-client-config 要素, 47
 SSLSessionTimeout init.conf 指令, 126
 ssl 要素, 35
 sso-enabled プロパティ, 43
 sso-max-inactive-seconds プロパティ, 43
 sso-reap-inaterval-seconds プロパティ, 43
 StackSize init.conf 指令, 118
 start-args 属性, 66
 state 属性, 42
 stderr 出力, 69
 stdout 出力, 69
 steady-pool-size 属性, 50, 54, 89
 steady-thread-pool-size 属性, 46
 store-pool-jndi-name プロパティ, 58
 store-properties 要素, 62
 store-protocol-class 属性, 84
 store-protocol 属性, 84
 StrictHttpHeaders init.conf 指令, 118
 sun-acc.xml ファイル, 140
 sun-application-client.xml ファイル, 140
 sun-application.xml ファイル, 140

sun-cmp-mapping.xml ファイル, 140
 sun-ejb-jar.xml ファイル, 140
 Sun ONE Application Server、「サーバー」を参照
 Sun ONE Message Queue, 67
 sun-ra.xml ファイル, 140
 sun-server_1_1.dtd ファイル, 24
 sun-web.xml ファイル, 140
 sybase-xa-recovery-workaround プロパティ, 75

T

TempDir init.conf 指令, 111
 TempDirSecurity init.conf 指令, 112
 TempDir パラメータ, 145
 TerminateTimeout init.conf 指令, 118
 TERM シグナル, 123
 ThreadIncrement init.conf 指令, 118
 timeout-in-seconds 属性, 74, 143
 timeoutSeconds プロパティ, 65
 tls-enabled 属性, 36
 tls-rollback-enabled 属性, 36
 transaction-isolation-level 属性, 89
 transaction-service 要素, 73
 TransmitFiles パラメータ, 145
 transport-protocol-class 属性, 84
 transport-protocol 属性, 84
 tx-log-dir 属性, 74
 type-by-extension 関数, 132
 type パラメータ, 132

U

Umask init.conf 指令, 130
 UNIX ユーザーアカウント、指定, 112
 url 属性, 143
 url プロパティ, 91

UseOutputStreamSize init.conf 指令, 127
 User init.conf 指令, 112
 user プロパティ, 91
 user 変数, 102
 use-system-logging 属性, 69

V

validation-table-name 属性, 90
 value 属性, 29, 143
 victim-selection-policy 属性, 53
 virtual-server-class 要素, 38
 virtual-servers 属性, 95, 97
 virtual-server 要素, 39

W

web-container 要素, 49
 web-module 要素, 96, 142
 web.xml ファイル
 session-timeout 要素, 65
 Web モジュール、デフォルト, 41

X

xaresource-txn-timeout プロパティ, 75

あ

アクセスログファイル, 68
 アルファベット順参照
 init.conf 指令, 155
 server.xml 要素, 151

え

え

エンコード、チャンクされた, 127

か

カスタマサポート, 15

仮想サーバー

と obj.conf ファイル, 18

のログファイル, 68

く

クラスタ、設定の変更, 22

け

形式、時刻, 149

こ

コンテナ管理による持続性, 87

さ

サーバー

HUP シグナル, 123

init.conf の初期化指令, 109

TERM シグナル, 123

基本動作, 17

設定する init.conf 指令, 111

プロセスを終了, 123

変更, 17

サーバーログファイル, 68

location, 28, 69

サービス品質, 43

サブ要素, 25

し

時刻形式, 149

収束ツリー

AUXILIARY クラスの inetSubscriber, 100

LDAP スキーマの, 99

組織の, 100

ユーザーエントリは inetOrgPerson と呼ばれる, 100

収束ツリーの inetOrgPerson, 100

証明書, 72

init.conf での設定, 125

初期化の設定, 18, 109

シングルサインオン (single sign-on), 43

す

スレッド、init.conf での設定, 114

スレッドプール、init.conf での設定, 119

せ

セキュリティ, 70

init.conf での設定, 125

待機ソケットの, 32, 47

セッションマネージャ, 59

接続、init.conf での設定, 114

設定、動的な, 19

設定ファイル, 17

手動編集, 20

そ

属性, 26

た

待機ソケット, [32, 47](#)

ち

チャンクされたエンコード, [127](#)

て

データベースの選択, [44, 98](#)

デフォルトの仮想サーバー

 http-listener 要素の, [34](#)

 サーバー全体の, [39](#)

と

動的再設定, [19](#)

ドメインコンポーネントツリー (dc), [100](#)

トランザクションログファイル, [68, 74](#)

ね

ネイティブのスレッドプール、init.conf での設定, [119](#)

ネイティブライブラリパス、設定, [77](#)

は

配備記述子, [140](#)

ひ

必要指定数, [25](#)

ふ

ファイル、マッピングタイプ, [132](#)

ファイル名拡張子、MIME タイプ, [132](#)

プロセス、init.conf での設定, [114](#)

プロパティ, [29](#)

へ

変数, [101](#)

「obj.conf」で参照, [101](#)

インタフェースで使われる, [102](#)

置き換えを許可, [101](#)

形式, [101](#)

評価, [103](#)

め

メッセージ駆動型 Beans, [53](#)

ゆ

ユーザーデータベースの選択, [44, 98](#)

よ

要求応答プロセス, [18](#)

要求処理プロセス, [18](#)

り

リソースアダプタ, [97](#)

ろ

ろ

ロードバランス , [96](#)

ログ

init.conf での設定 , [122](#)

server.xml の設定 , [68](#)