管理者ガイド

Sun™ ONE Web Server

Version 6.1

817-4162-10
2003 年 8 月

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

Copyright 2003 Sun Microsystems, Inc. All rights reserved.

Sun、Sun Microsystems、Sun のロゴマーク、Java、Solaris、Sun ONE、iPlanet, およびすべての Sun ONE ベースのロゴマークは、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。

Adobe® GoLive® は、米国およびその他の国における Adobe Systems Incorporated 社の登録商標です。

Macromedia® DreamWeaver® 米国およびその他の国における Macromedia, Inc. 社の登録商標です。

Netscape は、米国およびその他の国における Netscape Communications Corporation 社の登録商標です。

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

本書で説明されている製品は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。Sun および Sun のライセンサーの書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

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

目 次

このマニュアルについて **21**
内容の紹介 21
マニュアルの構成 22
第 1 部 : サーバーの基本 22
第 2 部 : 管理サーバーの使用 22
第 3 部 : 設定と監視 23
第 4 部 : 仮想サーバーとサービスの管理 24
第 5 部 : 付録 24
Sun ONE Web Server のマニュアルの使用 25
表記上の規則 27
製品サポート 28

第 1 部 サーバーの基本 **29**

第 1 章 Sun ONE Web Server の概要 **31**
Sun ONE Web Server 31
Sun ONE Web Server 6.1 の新機能 32
Java Servlet 2.3 および JSP (JavaServer Pages) 1.2 のサポート 32
JDK 1.4.1_03 のサポート 32
WebDAV のサポート 32
NSAPI フィルタのサポート 33
HTTP 圧縮のサポート 33
新しい検索エンジンのサポート 33
強化されたセキュリティ 34
JNDI のサポート 34

JDBC のサポート 34
Sun ONE Studio 5 のサポート 34
NSS 3.3.5 および NSPR 4.1.5 のサポート 35
PHP との互換性 35
強化されたハードウェアアクセラレータによる暗号化のサポート 35
起動時開始オプション 35
その他の機能 36
Sun ONE Web Server の管理と運用 36
Sun ONE Web Server の設定 36
管理サーバー 37
サーバーマネージャ 38
クラスマネージャ 39
仮想サーバーマネージャ 40
リソースピッカーの使用 41
リソースピッカーで使用するワイルドカード 41

第 2 章 Sun ONE Web Server の管理 45
管理サーバーの起動 45
UNIX/Linux プラットフォーム 45
Windows プラットフォーム 46
複数のサーバーの稼動 47
仮想サーバー 47
1 つのサーバーへの複数のインスタンスのインストール 47
サーバーの削除 48
以前のバージョンからのサーバーの移行 49

第 2 部 管理サーバーの使用 51

第 3 章 ユーザーとグループの管理 53
ユーザーとグループに関する情報へのアクセス 53
ディレクトリサービスについて 54
ディレクトリサービスの種類 54
ディレクトリサービスの設定 55
識別名 (DN) の理解 56
LDIF の使用 57
ユーザーの作成 57
LDAP ベース認証データベースの新規ユーザーの作成 58
LDAP ベースユーザーエントリ作成のガイドライン 58
新規ユーザーエントリの作成方法 59
Directory Server のユーザーエントリ 59

ファイルベース認証データベースの新規ユーザーの作成 61
新規ユーザーエントリの作成 61
ダイジェストベース認証データベースの新規ユーザーの作成 62
ユーザーの管理 62
ユーザー情報の検索 63
カスタム検索クエリの構築 64
ユーザー情報の編集 66
ユーザーのパスワードの管理 66
ユーザーライセンスの管理 67
ユーザー名の変更 67
ユーザーの削除 68
グループの作成 69
スタティックグループ 70
スタティックグループ作成のガイドライン 70
スタティックグループを作成するには 70
ダイナミックグループ 71
Sun ONE Web Server がダイナミックグループを実装するしくみ 71
スタティックでダイナミックなグループ 72
ダイナミックグループがサーバーパフォーマンスに与える影響 72
ダイナミックグループ作成のガイドライン 72
ダイナミックグループを作成するには 74
グループの管理 74
グループエントリの検索 75
Find all groups whose フィールド 75
グループ属性の編集 76
グループメンバーの追加 76
グループメンバーリストへのグループの追加 77
グループメンバーリストからのエントリの削除 78
所有者の管理 78
See Also の管理 78
グループの削除 79
グループの名前の変更 79
組織単位の作成 80
組織単位の管理 80
組織単位の検索 81
Find all units whose フィールド 81
組織単位の属性の編集 82
組織単位名の変更 82
組織単位の削除 82

第 4 章 Web コンテナと Web アプリケーションの J2EE ベースのセキュリティ 85

Sun ONE Web Server のセキュリティについて 86

ACL ベースのアクセス制御の概要 87
J2EE/ サーブレットベースのアクセス制御の概要 88
レルムベースのセキュリティ 89
レルムベースのユーザー認証 89
LDAP レルム 90
file レルム 90
solaris レルム 91
証明書レルム 91
カスタムレルム 91
native レルム 91
ロールベースの認証 92
ロールと制限対象領域のマッピング 92
ロールによるアクセス制御の定義 92
レルムの設定方法 94
管理インタフェースの使用 94
server.xml ファイルの編集 94
native レルムの設定 95
デフォルトレルムの指定 97
プログラムによるセキュリティの使用 97
どのような場合に J2EE/ サーブレット認証モデルを使用するか 98

第 5 章 管理サーバーの設定 99
管理サーバーのシャットダウン 99
待機ソケット設定の編集 100
ユーザーアカウントの変更 (UNIX/Linux) 100
スーパーユーザー設定の変更 101
複数の管理者の許可 103
ログファイルオプションの指定 104
ログファイルの表示 105
アクセスログファイル 105
エラーログファイル 105
ログファイルの保管 105
schedulerd 制御ベースのログローテーション の使用 (UNIX/Linux) 106
ディレクトリサービスの設定 107
サーバーへのアクセスの制限 107

第 6 章 証明書と鍵の使用 109
証明書ベースの認証 110
証明書を使用した認証 110
サーバー認証 110
クライアント認証 110
仮想サーバー証明書 110

信頼データベースの作成 . . . 111
信頼データベースの作成 . . . 111
password.conf の使用 . . . 112
SSL 有効サーバーの自動起動 . . . 112
VeriSign 証明書の要求およびインストール . . . 113
VeriSign 証明書の要求 . . . 113
VeriSign 証明書のインストール . . . 114
他のサーバー証明書の要求およびインストール . . . 115
必要な CA 情報 . . . 115
他のサーバー証明書の要求 . . . 116
他のサーバー証明書のインストール . . . 118
証明書のインストール . . . 119
アップグレード時の証明書の移行 . . . 120
組み込みルート証明書モジュールの使用 . . . 121
証明書の管理 . . . 122
CRL と CKL のインストールと管理 . . . 124
CRL または CKL のインストール . . . 124
CRL と CKL の管理 . . . 125
セキュリティに関する詳細設定 . . . 126
SSL と TLS プロトコル . . . 127
SSL を使用した LDAP との通信 . . . 127
待機ソケットのセキュリティの有効化 . . . 128
セキュリティ機能をオンにする . . . 128
待機ソケットのサーバー証明書の選択 . . . 129
暗号化方式の選択 . . . 130
セキュリティのグローバルな設定 . . . 132
SSLSessionTimeout . . . 133
SSLCacheEntries . . . 133
SSL3SessionTimeout . . . 133
外部暗号化モジュールの使用 . . . 134
PKCS#11 モジュールのインストール . . . 134
modutil による PKCS#11 モジュールのインストール . . . 134
pk12util の使用 . . . 135
待機ソケットの証明書名の選択 . . . 137
FIPS-140 標準 . . . 138
クライアントセキュリティ要件の設定 . . . 140
クライアント認証の要求 . . . 140
クライアントの認証を要求するには . . . 141
LDAP へのクライアント証明書のマッピング . . . 142
certmap.conf ファイルの使用 . . . 143
カスタムプロパティの作成 . . . 146
マッピング例 . . . 146
Stronger Ciphers の設定 . . . 148

セキュリティに関するその他の問題 150
物理的アクセスの制限 150
管理アクセスの制限 151
確実なパスワードの選択 151
推測しにくいパスワードの作成 151
パスワードまたは PIN の変更 152
パスワードの変更 152
サーバー上での他のアプリケーションの制限 153
UNIX と Linux 153
Windows 153
クライアントによる SSL ファイルのキャッシングを防ぐ 154
ポートの制限 154
サーバーの限界を知る 154
サーバーを保護するためのその他の追加変更 154
仮想サーバークラスへの chroot の指定 155
仮想サーバーへの chroot の指定 156

第 7 章 サーバークラスタの管理 157
クラスタについて 157
サーバークラスタの使用に関するガイドライン 158
クラスタの設定 159
クラスタへのサーバーの追加 160
サーバー情報の変更 161
クラスタからのサーバーの削除 162
サーバークラスタの制御 162
変数の追加 163

第 3 部 設定と監視 165

第 8 章 サーバーの詳細設定 167
サーバーの起動と停止 167
終了タイムアウトの設定 168
サーバーの再起動 (UNIX/Linux) 169
SSL が有効なサーバーを自動的に起動 169
inittab を使用した再起動 (UNIX/Linux) 170
システムの rc (実行制御) スクリプトを使用した再起動 (UNIX/Linux) 170
サーバーの手動再起動 (UNIX/Linux) 170
サーバーの手動停止 (UNIX/Linux) 171
サーバーの再起動 (Windows) 171
自動再起動ユーティリティの使用 (Windows) 172

サーバーのパフォーマンスの調整 173
magnus.conf ファイルの編集 174
待機ソケットの追加と編集 174
MIME タイプの選択 175
アクセスの制限 176
設定の復元 177
ファイルキャッシュの設定 177
スレッドプールの追加と使用 178
ネイティブスレッドプールと汎用スレッドプール (Windows) 178
スレッドプール (UNIX/Linux) 178
スレッドプールの編集 179
スレッドプールの使用 179

第 9 章 サーバーへのアクセス制御 181

アクセス制御とは 182
ユーザー - グループのアクセス制御の設定 182
デフォルト認証 (Default) 183
基本認証 (Basic) 183
SSL 認証 (SSL) 184
ダイジェスト認証 (Digest) 186
その他の認証 (Other) 189
ホスト - IP のアクセス制御の設定 189
アクセス制御ファイルの使用 190
ACL ユーザーキャッシュの設定 190
アクセス制御のしくみ 191
アクセス制御の設定 193
グローバルなアクセス制御の設定 194
サーバーインスタンスに対するアクセス制御の設定 197
アクセス制御オプションの選択 203
アクションの設定 203
ユーザーとグループの指定 204
From Host の指定 206
プログラムへのアクセス制限 207
アクセス権の設定 208
カスタマイズされた式の作成 208
アクセス制御の解除 209
アクセスが拒否された場合の応答 210
サーバーの一部へのアクセス制御 210
サーバー全体に対するアクセス制限 211
ディレクトリ (パス) へのアクセス制限 211
URI (パス) へのアクセス制限 212
ファイルタイプに対するアクセス制限 213

時刻に基づくアクセス制限 214
セキュリティに基づくアクセス制限 215
分散管理によるアクセス制御のセキュリティ保護 216
リソースへのアクセスのセキュリティ保護 216
サーバーインスタンスへのアクセスのセキュリティ保護 216
IP ベースのアクセス制御の有効化 217
ダイナミックアクセス制御ファイルの使用 218
.htaccess ファイルの使用 218
ユーザーインタフェースからの .htaccess の有効化 219
magnus.conf からの .htaccess の有効化 219
既存の .nsconfig ファイルの .htaccess ファイルへの変換 221
htaccess-register の使用 222
.htaccess ファイルの例 222
サポートされる .htaccess 指令 223
.htaccess セキュリティに関する注意事項 227
仮想サーバーへのアクセス制御 227
仮想サーバーからデータベースへのアクセス 228
ユーザーインタフェースでの LDAP データベースの指定 228
仮想サーバーのアクセス制御リストの編集 229
ファイルベースの認証用 ACL の作成 230
ファイル認証に基づくディレクトリサービス用の ACL の作成 232
.htaccess 認証に基づくディレクトリサービス用の ACL の作成 233
ファイル認証データベースへの既存の .htaccess 情報の移行 234
ダイジェスト認証に基づくディレクトリサービス用の ACL の作成 235

第 10 章 ログファイルの使用 237

ログファイルについて 238
UNIX および Windows プラットフォームへのログオン 238
デフォルトのエラーログ 238
syslog を利用したログ 239
Windows の eventlog を利用したログ 240
ログレベル 241
仮想サーバーとログの記録について 241
アプリケーションとサーバーのログ出力のリダイレクト 242
ログファイルの保管 243
内部デーモンログローテーション 243
スケジューラベースのログローテーション 244
アクセスログの詳細設定 245
Cookie を使用した簡易ロギング 246
エラーロギングオプションの設定 246
管理サーバーインスタンスの設定 246
サーバーインスタンスの設定 246

LOG 要素の設定 247
アクセスログファイルの参照 248
エラーログファイルの参照 250
ログアナライザの実行 251
イベントの表示 (Windows) 254

第 11 章 サーバーの監視 255
統計情報によるサーバーの監視 256
　統計情報を使用可能にする 257
　統計情報の使用法 257
サービス品質の使用法 258
　サービス品質の例 258
　サービス品質の設定 259
　obj.conf で必要な変更 261
　サービス品質に関する既知の制限事項 261
SNMP の基本 263
Sun ONE Web Server の MIB 264
SNMP の設定 270
プロキシ SNMP エージェントの使用法 (UNIX または Linux) 272
　プロキシ SNMP エージェントのインストール 272
　プロキシ SNMP エージェントの起動 273
　ネイティブ SNMP デーモンの再起動 273
SNMP ネイティブエージェントの再設定 274
SNMP マスターエージェントのインストール 274
SNMP マスターエージェントを使用可能にして起動する 275
　マスターエージェントを別のポートで起動する 276
　SNMP マスターエージェントを手動で設定する 276
　マスターエージェントの CONFIG ファイルの編集 277
　sysContact 変数と sysLocation 変数の定義 277
　SNMP サブエージェントの設定 278
　SNMP マスターエージェントの起動 279
　　手動による SNMP マスターエージェントの起動 279
　　管理サーバーを使用して SNMP マスターエージェントを起動する 279
SNMP マスターエージェントの設定 281
　コミュニティ文字列の設定 281
　トラップ送信先の設定 281
サブエージェントを使用可能にする 282
SNMP メッセージについて 282

第 12 章 ネーミングとリソースの設定 285
Java の有効化と無効化 285
JVM の設定 287

一般設定 . . . 287
パスの設定 . . . 288
JVM オプションの設定 . . . 288
JVM プロファイラの設定 . . . 289
J2EE ネーミングサービスおよびリソースについて . . . 289
JDBC データソース . . . 290
JDBC 接続プール . . . 290
Java メールセッション . . . 291
カスタムリソース . . . 291
外部 JNDI リソース . . . 292
JNDI (Java Naming and Directory Interface) について . . . 292
J2EE ネーミングサービス . . . 292
ネーミング参照とバインド情報 . . . 294
J2EE 標準配備記述子内のネーミング参照 . . . 294
アプリケーション環境エントリ . . . 295
リソースへの参照 . . . 295
リソース環境参照 . . . 296
初期ネーミングコンテキスト . . . 297
JNDI 接続ファクトリ . . . 297
Java ベースのリソースの作成 . . . 298
JDBC 接続プールの新規作成 . . . 299
管理インタフェースを使用 . . . 299
コマンド行インタフェースを使用 . . . 302
JDBC リソースの作成 . . . 303
管理インタフェースを使用 . . . 303
コマンド行インタフェースを使用 . . . 303
カスタムリソースの作成 . . . 304
管理インタフェースを使用 . . . 304
コマンド行インタフェースを使用 . . . 304
外部 JNDI リソースの作成 . . . 304
管理インタフェースを使用 . . . 304
コマンド行インタフェースを使用 . . . 305
Java ベースのリソースの変更 . . . 305
JDBC 接続プールの変更 . . . 305
JDBC リソースの変更 . . . 306
カスタムリソースの変更 . . . 306
外部 JNDI リソースの変更 . . . 306
Java ベースのリソースの削除 . . . 307
JDBC 接続プールの削除 . . . 307
JDBC リソースの削除 . . . 307
カスタムリソースの削除 . . . 308
外部 JNDI リソースの削除 . . . 308

第 4 部 仮想サーバーとサービスの管理 311

第 13 章 仮想サーバーの使用 313
仮想サーバーの概要 313
複数のサーバーインスタンス 314
仮想サーバークラス 315
obj.conf ファイル 315
クラスに属する仮想サーバー 316
デフォルトのクラス 316
待機ソケット 316
仮想サーバー 317
仮想サーバーの種類 317
IP アドレスベースの仮想サーバー 317
URL ホストベースの仮想サーバー 318
デフォルトの仮想サーバー 318
要求を処理する仮想サーバーの選択 319
ドキュメントルート 319
ログファイル 320
前のリリースから仮想サーバーを移行する 320
仮想サーバーで Sun ONE Web Server の機能を使用する 321
仮想サーバーで SSL を使用する 321
仮想サーバーでアクセス制御を使用する 322
仮想サーバーで CGI を使用する 322
仮想サーバーで設定スタイルを使用する 322
仮想サーバーのユーザーインタフェースの使用法 323
クラスマネージャ 323
仮想サーバーマネージャ 324
変数の使用法 324
ダイナミック再設定 324
仮想サーバーの設定 325
待機ソケットの作成 326
仮想サーバークラスの作成 326
仮想サーバークラスの編集または削除 327
仮想サーバークラスと関連付けるサービスの指定 327
仮想サーバーの作成 327
仮想サーバーと関連付ける設定の指定 328
個々の仮想サーバーをユーザーが監視できるようにする 328
アクセス制御 331
ログファイル 331
仮想サーバーの配備 332
例 1 : デフォルトの構成 332
例 2 : セキュリティ保護されたサーバー 334

例 3 : イントラネットホスティング 335
例 4 : マスホスティング 337

第 14 章 仮想サーバーの作成と設定 339
仮想サーバーの作成 339
仮想サーバーの設定内容の変更 340
クラスマネージャを使用した変更 340
仮想サーバーの設定内容の変更 340
仮想サーバーの MIME の設定 341
仮想サーバーの ACL の設定 342
仮想サーバーのセキュリティの設定 342
仮想サーバーのサービス品質の設定 342
仮想サーバーのログの設定 343
仮想サーバーのログの有効化 344
仮想サーバーの Java Web アプリケーションの設定 345
仮想サーバーマネージャを使用した変更 345
仮想サーバーのレポートの生成 346
仮想サーバーのディレクトリサービスの選択 348
仮想サーバーの削除 348

第 15 章 プログラムによるサーバーの拡張 349
サーバーサイドプログラムの概要 349
サーバーで実行するサーバーサイドアプリケーションのタイプ 350
サーバーへのサーバーサイドアプリケーションのインストール方法 350
Java サーブレットと JavaServer Pages (JSP) 350
サーブレットと JavaServer Pages の概要 351
サーバーでサーブレットを実行するための要件 352
Web アプリケーションの配備 352
server.xml ファイルの使用 352
管理サーバーインタフェースを使用 353
コマンド行インタフェースを使用 354
Web アプリケーションに含まれていないサーブレットと JSP の配備 358
JVM の設定 358
バージョンファイルの削除 358
CGI プログラムのインストール 359
CGI の概要 360
CGI ディレクトリの指定 361
各仮想サーバーに固有の CGI 属性を設定する 362
ファイルタイプとして CGI を指定 363
実行可能ファイルのダウンロード 363
Windows CGI プログラムのインストール 364
Windows CGI プログラムの概要 364

Windows CGI ディレクトリの指定 365
ファイルタイプとして Windows CGI を指定 366
Windows でのシェル CGI プログラムのインストール 367
Windows 向けシェル CGI プログラムの概要 367
シェル CGI ディレクトリの指定 (Windows) 368
ファイルタイプとしてシェル CGI を指定 (Windows) 368
クエリハンドラの使用 370

第 16 章 コンテンツ管理 371
プライマリドキュメントディレクトリの設定 372
追加ドキュメントディレクトリの設定 373
ユーザー公開情報ディレクトリのカスタマイズ (UNIX/Linux) 374
コンテンツ発行の制限 375
起動時のパスワードファイル全体の読み込み 375
設定スタイルの使用 376
リモートファイル操作の有効化 376
ドキュメントの設定 377
ドキュメント設定の変更 377
インデックスファイル名の入力 377
ディレクトリのインデックス作成の選択 378
サーバーのホームページの指定 378
デフォルト MIME タイプの指定 378
URL 転送の設定 379
エラー応答のカスタマイズ 380
文字セットの変更 381
ドキュメントのフッターの変更 382
htaccess の使用 383
シンボリックリンクの制限 (UNIX/Linux) 383
サーバーが解析する HTML の設定 384
キャッシュ制御指令の設定 385
強固な暗号の使用 386
コンテンツを圧縮するためのサーバー設定 386
事前に圧縮したコンテンツを配信するようにサーバーを設定する 386
コンテンツをオンデマンドで圧縮するようにサーバーを設定する 388
圧縮に関連する obj.conf 内の変更 389

第 17 章 設定スタイルの適用 391
設定スタイルの作成 391
設定スタイルの割り当て 394
設定スタイルの割り当ての一覧表示 394
設定スタイルの編集 395
設定スタイルの削除 396

第 18 章 検索機能の使い方 397
検索について 398
仮想サーバーでの検索アプリケーションの有効化 399
仮想サーバーでの検索アプリケーションの無効化 400
検索コレクションについて 400
コレクションの作成 401
コレクションの設定 402
コレクションの更新 403
コレクションの削除 404
コレクションの保守 405
コレクションのインデックス再作成 405
コレクションの保守スケジュールの追加 406
コレクション保守スケジュールの編集 407
コレクションの保守スケジュールの削除 408
検索の実行 408
検索ページ 409
クエリの作成 410
詳細検索 410
検索結果の表示 412
検索ページのカスタマイズ 412
検索インタフェースのコンポーネント 413
ヘッダー 413
フッター 413
フォーム 413
結果 413
検索クエリページのカスタマイズ 413
水平バー形式 414
サイドバーブロック形式 414
検索結果ページのカスタマイズ 416
独立したフォームページと結果ページのカスタマイズ 420
タグの規則 421
タグの仕様 421

第 19 章 WebDAV による Web パブリッシング 423
WebDAV について 423
一般的な WebDAV 用語 424
WebDAV の使用 428
WebDAV の有効化 428
サーバーインスタンスレベルでの WebDAV の有効化 429
仮想サーバークラスレベルでの WebDAV の有効化 430
コレクションレベルでの WebDAV の有効化 431
WebDAV コレクションの作成 432

WebDAV コレクションの編集 433
WebDAV の設定 434
仮想サーバーレベルでの WebDAV の設定 435
URI レベルでの WebDAV の設定 436
WebDAV 対応サーバーでのソース URI と Translate:f ヘッダーの使用 437
リソースのロックとロック解除 438
排他的ロック 438
共有ロック 439
ロックの管理 439
ロックの最小タイムアウト 439
ロック要求の例 440
WebDAV のアクセス制御の有効化 441
WebDAV 有効リソースへのアクセスの制限 441
セキュリティに関する注意事項 442

第 5 部 付録 445

付録 A コマンド行ユーティリティ 447

HttpServerAdmin (仮想サーバーの管理) 447
HttpServerAdmin の構文 448
control コマンド 449
オプション 449
構文 449
パラメータ 449
例 450
create コマンド 450
オプション 450
仮想サーバークラスの作成 450
待機ソケットの作成 451
仮想サーバーの作成 452
JDBC 接続プールの作成 453
構文 454
オプション 454
例 455
JDBC リソースの作成 455
構文 455
オプション 455
例 455
カスタムリソースの作成 456
構文 456
オプション 456

例 . . . 456
外部 JNDI リソース . . . 457
構文 . . . 457
オプション . . . 457
例 . . . 457
メールリソースの作成 . . . 458
構文 . . . 458
オプション . . . 458
例 . . . 459
delete コマンド . . . 459
オプション . . . 459
クラスの削除 . . . 460
待機ソケットの削除 . . . 460
仮想サーバーの削除 . . . 461
JDBC 接続プールの削除 . . . 462
JNDI リソースの削除 . . . 462
list コマンド . . . 463
構文 . . . 463
オプション . . . 463
例 . . . 463

付録 B Hypertext Transfer Protocol . . . 465

HTTP (HyperText Transfer Protocol) について . . . 465
要求 . . . 466
要求メソッド . . . 466
要求ヘッダー . . . 466
要求データ . . . 467
応答 . . . 467
状態コード . . . 467
応答ヘッダー . . . 469
応答データ . . . 469

付録 C ACL ファイルの構文 . . . 471

ACL ファイルの構文 . . . 471
認証メソッド . . . 473
承認文 . . . 474
承認文の階層 . . . 474
属性式 . . . 475
式の演算子 . . . 476
デフォルト ACL ファイル . . . 477
汎用構文の項目 . . . 477
obj.conf での ACL ファイルの参照 . . . 478

付録 D 国際化とローカライズのサポート 479
マルチバイトデータの入力 479
ファイル名またはディレクトリ名 479
LDAP ユーザーとグループ 479
複数文字エンコーディングのサポート 480
WebDAV 480
検索 480
言語の設定 481
ローカライズされたコンテンツを配信するようにサーバーを設定する 481

用語集 483

索引 493

このマニュアルについて

このマニュアルでは、Sun™ Open Net Environment (Sun ONE) Web Server 6.1 の設定および管理の方法について説明します。このマニュアルは、クライアントサーバーアプリケーションを WWW (World Wide Web) を経由してより幅広いユーザーに拡張したいと考えている、企業の IT 管理者を対象にしています。

この章には、次の内容が記述されています。

- 内容の紹介
- マニュアルの構成
- Sun ONE Web Server のマニュアルの使用
- 表記上の規則
- 製品サポート

内容の紹介

このマニュアルは、Sun ONE Web Server の設定および管理の方法について説明します。サーバーの設定が終了したら、このマニュアルはサーバーの保守管理に使用します。

サーバーのインストール終了後、このマニュアルは、サーバーのルートディレクトリの `/manual/https/ag` に置かれ、HTML 形式で表示することができます。デフォルトでは、サーバーのルートディレクトリは `C:¥Sun¥WebServer6.1¥` または `/opt/SunWwbsvr` です。

マニュアルの構成

このマニュアルは、5 部からなる内容と用語集、および索引に分かれています。Sun ONE Web Server 6.1 を初めて使用する場合、第 1 部「サーバーの基本」から始めて製品の概要を理解してください。Sun ONE Web Server のこのバージョンをすでに使用したことがある場合には、第 1 部「サーバーの基本」の内容にざっと目を通すだけにして、第 2 部「管理サーバーの使用」に進んでください。

管理サーバーの基本的な使用方法を理解したら、第 3 部「設定と監視」を参照します。ここには、Sun ONE Web Server の設定や監視の例が示されています。第 4 部「仮想サーバーとサービスの管理」には、プログラムの使用と設定スタイルに関する情報が記述されています。

最後に、付録では、次の項目を含むさまざまな事項についての参照情報を提供します。HTTP (Hypertext Transfer Protocol)、サーバー設定ファイル、ACL ファイル、国際化に関する問題、サーバー拡張、Sun ONE Web Server ユーザーインタフェースのリファレンスなど、必要に応じて参照できます。ただし、ユーザーインタフェースに関する付録は、オンラインバージョンでのみ参照可能です。

第 1 部 : サーバーの基本

第 1 部では、Sun ONE Web Server の概要について説明します。次の章が記述されています。

- 第 1 章「Sun ONE Web Server の概要」では、Sun ONE Web Server の概要を説明します。
- 第 2 章「Sun ONE Web Server の管理」では、管理サーバーを使用して Sun ONE Web Server を管理する方法について説明します。

第 2 部 : 管理サーバーの使用

第 2 部では、Sun ONE Web Server を管理する管理サーバーの使用方法について、概念と手順を詳しく説明します。次の章が記述されています。

- 第 3 章「ユーザーとグループの管理」では、管理サーバーユーザーおよびグループフォームを使用して、Sun ONE Web Server を設定する方法について説明します。
- 第 4 章「Web コンテナと Web アプリケーションの J2EE ベースのセキュリティ」では、Sun ONE Web Server のセキュリティを設定する方法と、2 つのセキュリティ モジュール (ACL ベースのアクセス制御と J2EE™ (Java™ 2 Platform, Enterprise Edition)/ サーブレットベースの認証および承認) について説明します。

- 第 5 章「管理サーバーの設定」では、管理サーバーの設定とグローバル設定フォームを使用して、Sun ONE Web Server を設定する方法について説明します。
- 第 6 章「証明書と鍵の使用」では、証明書と公開鍵を使用してセキュリティを向上する方法について説明します。この章を読む前に、公開鍵暗号法と SSL (Secure Sockets layer) プロトコルの基本的な概念を理解しておく必要があります。この概念には、暗号化と復号化、鍵、デジタル証明書と電子署名、SSL 暗号化、暗号化方式、SSL 接続 (ハンドシェーク) の主な手順が含まれます。
- 第 7 章「サーバークラスタの管理」では、サーバーのクラスタ化の概念、およびそれを使用してサーバー間で設定を共有する方法について説明します。

第 3 部 : 設定と監視

第 3 部では、Server Manager を使用して Sun ONE Web Server を設定および監視する方法を、例を示して説明します。次の章が記述されています。

- 第 8 章「サーバーの詳細設定」では、Sun ONE Web Server のサーバー設定方法について説明します。
- 第 9 章「サーバーへのアクセス制御」では、サーバーの各部にアクセスできるユーザーを指定する方法について説明します。
- 第 10 章「ログファイルの使用」では、HTTP (Hypertext Transfer Protocol) プロトコルを使用して、ログファイルを記録および表示する、またはオペレーティングシステムのパフォーマンス監視ツールを使用することで Sun ONE Web Server を監視する方法について説明します。
- 第 11 章「サーバーの監視」では、SNMP (Simple Network Management Protocol) を使用して Sun ONE Web Server を監視する方法について説明します。
- 第 12 章「ネーミングとリソースの設定」では、JNDI (Java Naming and Description Interface) リソースを設定して、サーバーでのデータベース接続を可能にする方法について説明します。

第 4 部 : 仮想サーバーとサービスの管理

第 4 部では、プログラムや設定スタイルについてサーバーマネージャの使用に関する情報を提供します。次の章が記述されています。

- 第 13 章「仮想サーバーの使用」では、Sun ONE Web Server を使用して仮想サーバーを設定および管理する方法について説明します。
- 第 14 章「仮想サーバーの作成と設定」では、個々の仮想サーバーを作成して設定する方法について説明します。
- 第 15 章「プログラムによるサーバーの拡張」では、Java アプレット、CGI プログラム、JavaScript アプリケーション、およびその他のプラグインをサーバーにインストールする方法について説明します。
- 第 16 章「コンテンツ管理」では、サーバーの内容を設定および管理する方法について説明します。
- 第 17 章「設定スタイルの適用」では、Sun ONE Web Server で設定スタイルを使用する方法について説明します。
- 第 18 章「検索機能の使い方」では、サーバー上のドキュメントの内容および属性を検索する方法について説明します。さらにこの章では、ユーザーコミュニティに合わせたカスタムテキスト検索インタフェースの作成方法についても説明します。
- 第 19 章「WebDAV による Web パブリッシング」では、Web パブリッシングおよび相互に離れた複数のユーザー間での Web 上の共同オーサリングを可能にする WebDAV を使用できるように仮想サーバーを設定する方法について説明します。

第 5 部 : 付録

第 5 部には、役に立つと思われるリファレンス情報を記載したさまざまな付録が含まれています。次の付録が含まれています。

- 付録 A「コマンド行ユーティリティ」では、ユーザーインタフェース画面の代わりにコマンド行ユーティリティを使用する方法について説明します。
- 付録 B「Hypertext Transfer Protocol」では、HTTP に関するいくつかの基本概念について簡単に説明します。
- 付録 C「ACL ファイルの構文」では、アクセス制御リスト (ACL) ファイルとその構文について説明します。
- 付録 D「国際化とローカライズのサポート」では、Sun ONE Web Server の国際化されたバージョンについて説明します。

用語集では、頻繁に使用される用語で、Sun ONE Web Server 管理者にはあまりなじみがないものについて解説しています。

Sun ONE Web Server のマニュアルの使用

Sun ONE Web Server のマニュアルは、次の Web サイトで PDF 形式および HTML 形式のオンラインファイルとして提供されています。

http://docs.sun.com/db/prod/s1websrv?l=ja#hic

次の表は、Sun ONE Web Server のマニュアルで説明しているタスクと概念を示しています。

表 1 Sun ONE Web Server のマニュアルガイド

記載されている情報	参照するマニュアル
ソフトウェアとマニュアルの最新情報	『リリースノート』
サーバーの基本知識および機能を紹介する実践練習を含む、Sun ONE Web Server の初心者向け情報 (初めてのユーザーにお勧めします)	『Getting Started Guide』
インストールおよび移行タスクの実行 • Sun ONE Web Server、および各種コンポーネント、サポートされるプラットフォーム、環境のインストール • Sun ONE Web Server 4.1 または 6.0 から Sun ONE Web Server 6.1 への移行	『インストールおよび移行ガイド』

表 1　Sun ONE Web Server のマニュアルガイド (続き)

記載されている情報	参照するマニュアル
次の管理タスクの実行 • 管理インタフェースおよびコマンド行インタフェースの使用 • サーバーの詳細設定 • サーバーインスタンスの使用 • サーバーアクティビティの監視と記録 • ユーザー証明書と公開鍵暗号化によるサーバーのセキュリティ保護 • アクセス制御の設定によるサーバーのセキュリティ保護 • J2EE™ (Java™ 2 Platform, Enterprise Edition) セキュリティ機能の使用 • アプリケーションの配備 • 仮想サーバーの管理 • サーバーの作業負荷の定義と、目的パフォーマンスに合わせたシステムのサイズ設定 • Java™ アプレット、CGI プログラム、JavaScript アプリケーション、その他のプラグインのインストール • サーバードキュメントのコンテンツと属性の検索、およびテキスト検索インタフェースの作成 • コンテンツを圧縮するためのサーバー設定 • WebDAV による Web パブリッシングとコンテンツオーサリングのためのサーバー設定	『管理者ガイド』
次の処理を行うためのプログラミング技術と API の使用 • Sun ONE Web Server の拡張と修正 • クライアント要求に対応するコンテンツのダイナミックな生成 • サーバーコンテンツの修正	『Programmer's Guide』

表 1　Sun ONE Web Server のマニュアルガイド (続き)

記載されている情報	参照するマニュアル
カスタム NSAPI (Netscape Server Application Programmer's Interface) プラグインの作成	『NSAPI Programmer's Guide』
Sun ONE Web Server へのサーブレットおよび JSP™ (JavaServer Pages™) 技術の実装	『Programmer's Guide to Web Applications』
設定ファイルの編集	『Administrator's Configuration File Reference Guide』
Sun ONE Web Server のパフォーマンスを最適化するためのチューニング	『Performance Tuning, Sizing, and Scaling Guide』

表記上の規則

このマニュアル全体に適用される表記規則について説明します。

- **ファイルとディレクトリパス**は、UNIX 形式で表記されます (ディレクトリ名はスラッシュで区切られます)。Windows バージョンでは、ディレクトリパスは同じですが、ディレクトリの区切りには ¥ (円マーク) が使用されます。
- URL は、次の形式で表記されます。

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

  この URL で、server はアプリケーションを実行するサーバーの名前、domain は使用しているインターネットドメイン名、path はサーバーのディレクトリ構造、file は個々のファイル名を表わします。URL 内の斜体文字列は、可変数を表わします。
- **書体の表記規則**は次のとおりです。
  - モノスペース (monospace) フォントは、コード例とコードリスト、API と言語要素 (関数名やクラス名など)、ファイル名、パス名、ディレクトリ名、HTML タグの表記に使用されます。
  - 斜体 (*Italic*) は、コード変数の表記に使用されます。
  - 斜体 (*Italic*) は、マニュアルタイトル、強調、変数、専門用語の文字通りの意味での解釈を表記する場合にも使用されます。
  - 太字 (**Bold**) は、段落の導入部、および専門用語の文字通りの意味での解釈の表記に使用されます。
- このマニュアルでは、**インストールのルートディレクトリ**は *install_dir* と表記されます。

UNIX ベースのプラットフォームでは、*install_dir* のデフォルトの場所は次のディレクトリです。

/opt/SUNWwbsvr/

Windows 環境では、次のディレクトリです。

C:¥Sun¥WebServer6.1

製品サポート

システムの使用にあたって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 Web サイトからアクセスするオンラインサポート

  http://jp.sun.com/supportraining/

第 1 部

サーバーの基本

第 1 章「Sun ONE Web Server の概要」

第 2 章「Sun ONE Web Server の管理」

第 1 章

Sun ONE Web Server の概要

この章では、Sun ONE Web Server を紹介し、サーバーの基本的な概念についていくつかを説明します。この章を読み、Sun ONE Web Server の機能の概要を理解してください。

この章には、次の内容が記述されています。

- Sun ONE Web Server
- Sun ONE Web Server の設定
- 管理サーバー
- サーバーマネージャ
- クラスマネージャ
- 仮想サーバーマネージャ
- リソースピッカーの使用

Sun ONE Web Server

Sun ONE Web Server 6.1 は、マルチプロセス、マルチスレッドの Web サーバーで、オープン規格に基いて構築されています。この製品は、どのような規模の企業にも、高い性能、信頼性、スケーラビリティ、管理性を提供します。

この節では、Sun ONE Web Server の機能について説明し、基本管理タスクの一部を紹介します。この節は、次のトピックから構成されます。

- Sun ONE Web Server 6.1 の新機能
- Sun ONE Web Server の管理と運用

Sun ONE Web Server 6.1 の新機能

Sun ONE Web Server 6.1 には、次の新機能が用意されています。

Java Servlet 2.3 および JSP (JavaServer Pages) 1.2 のサポート

Sun ONE Web Server 6.1 には、Java™ Servlet 2.3 および JSP™ (JavaServer Pages™) 1.2 仕様の J2EE™ (Java™ 2 Platform, Enterprise Edition) に準拠した実装が含まれます。J2EE 準拠の Web コンテナは、Java™ 技術標準に準拠した Web アプリケーションの設計と配備に必要な、柔軟性と信頼性を提供します。Web アプリケーションは、仮想サーバー単位で配備することができます。

この技術の詳細については、次の情報を参照してください。

Java サーブレット
http://java.sun.com/products/servlet/index.html

Java Servlet 2.3 仕様
http://java.sun.com/products/servlet/download.html

JavaServer Pages
http://java.sun.com/products/jsp/index.html

Sun ONE Web Server での サーブレットと JSP の開発については、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』を参照してください。

JDK 1.4.1_03 のサポート

Sun ONE Web Server 6.1 は、JDK™ (Java Developer's Kit) 1.4.1_03 をサポートしています。この JDK は、Web Server にバンドルされ、Web Server のインストール時に同時にインストールされます (インストールを選択した場合)。Web Server のインストール後に、独自の JDK をインストールすることもできます。管理サーバー、および Java とサーブレットのサポートを利用するには、JDK をインストールしておく必要があります。

WebDAV のサポート

Sun ONE Web Server 6.1 は、WebDAV (Web-based Distributed Authoring and Versioning) プロトコルをサポートしています。このプロトコルは、次の機能によって共同 Web パブリッシングを可能にします。

RFC 2518 準拠、および RFC 2518 クライアントとの相互操作性

- Web パブリッシングのためのセキュリティとアクセス制御
- ファイルシステムベースの WebDAV コレクションおよびリソースに対する基本パブリッシング操作

WebDAV は、コンテンツメタデータ、ネームスペース管理、上書き保護を総合的にサポートします。これらの技術と WebDAV がサポートする多数のオーサリングツールを組み合わせることで、共同作業環境向けの理想的な開発プラットフォームが提供されます。

NSAPI フィルタのサポート

Sun ONE Web Server 6.1 では、NSAPI フィルタをサポートできるように NSAPI (Netscape Server Application Programmer's Interface) が拡張されています。

フィルタを使用することで、HTTP 要求および応答のストリームをカスタム処理できます。これにより、関数が、別の関数に提供されるコンテンツや別の関数により作成されたコンテンツを途中で受け取って変更することも可能になります。たとえば、プラグインに NSAPI フィルタをインストールして別のプラグインの SAF (Server Application Function) が生成する XML ページを受信し、その XML ページをクライアントに合わせて HTML、XHTML、または WAP ページに変換することもできます。あるいは、NSAPI フィルタを使用して、クライアントから受信したデータを解凍してから別のプラグインに渡すことができます。

詳細については、『Sun ONE Web Server 6.1 NSAPI Programmer's Guide』を参照してください。

HTTP 圧縮のサポート

Sun ONE Web Server 6.1 では、コンテンツの圧縮を行うことができます。そのため、クライアントへの配信速度を高め、ハードウェア側の負荷を増やすことなく、より大容量のコンテンツに対応できます。コンテンツを圧縮することで、コンテンツのダウンロード時間が短縮されるため、ダイヤルアップ接続や、高トラフィックの接続を利用するユーザーにとって大きな利点となります。

詳細については、『Sun ONE Web Server 6.1 管理者ガイド』を参照してください。

新しい検索エンジンのサポート

Sun ONE Web Server 6.1 は、全文検索のインデックス生成および検索結果の取得機能に対応した、新しい Java ベースの検索エンジンをサポートします。この検索機能を利用することで、ユーザーはサーバー上のドキュメントを検索し、結果を Web ページに表示できます。サーバー管理者はドキュメントのインデックスを作成し、ユーザーはこのインデックスに対して検索を行います。また、サーバー管理者は具体的な必要に合わせて検索インタフェースをカスタマイズすることができます。

詳細については、『Sun ONE Web Server 6.1 管理者ガイド』を参照してください。

強化されたセキュリティ

Sun ONE Web Server 6.1 の新機能を使用することで、フラットファイル認証を使ってアクセスを制限できます。これまでのバージョンの Web Server と異なり、Sun ONE Web Server 6.1 は Java Security Manager もサポートするようになりました。製品インストール時のデフォルト設定では、Java Security Manager は無効に設定されています。`server.xml` の詳細については、『Sun ONE Web Server 6.1 Administrator's Configuration File Reference Guide』を参照してください。

JNDI のサポート

Sun ONE Web Server 6.1 は、JNDI (Java Naming and Directory Interface™) をサポートします。JNDI により、複数種類が混在するエンタープライズネーミングサービスおよびディレクトリサービスへのシームレスな接続が可能になります。

JDBC のサポート

Sun ONE Web Server は、特別な設定なしで使用できるシームレスな JDBC™ (Java™ DataBase Connectivity) を提供し、業界標準の JDBC ドライバから、カスタマイズされたドライバまで広くサポートします。

Sun ONE Studio 5 のサポート

Sun ONE Web Server 6.1 は、Sun™ ONE Studio 5, Standard Edition をサポートしています。Sun ONE Studio 技術は、Java テクノロジ開発者向けの、強力で拡張可能な統合型開発環境 (IDE) です。Sun ONE Studio 5 は、NetBeans™ ソフトウェアを基本とし、Sun ONE プラットフォームに統合されています。(Sun ONE Web Server 6.1 は、NetBeans 3.5 および 3.5.1 もサポートしています。)

Sun ONE Studio は、Sun ONE Web Server 6.1 がサポートするすべてのプラットフォームでサポートされています。Web Server 用のプラグインは、次の方法で入手できます。

- Sun ONE Web Server 6.1 メディアキットに含まれる付属 CD
- Sun ONE Studio の自動更新機能
- 次の Web サイトにある Sun ONE Web Server 6.1 のダウンロードセンター
  http://www.sun.com/software/download/inter_ecom.html

Sun ONE Web Server 6.1 用の Sun ONE Studio 5 プラグインは、ローカル Web サーバーだけで機能することに注意してください (つまり、IDE と Web サーバーが同じマシンに存在する必要があります)。

Sun ONE Web Server 6.1 用の Sun ONE Studio 5 プラグインは、Sun™ ONE Application Server 7 用プラグインと同じように動作します。Sun ONE Studio 5 の Web アプリケーション機能の使用については、次の Web サイトにあるチュートリアルを参照してください。
http://developers.sun.com/tools/javatools/documentation/s1s5/cdshop.pdf

Sun ONE Web Server 6.1 インスタンスをデフォルトとして設定し、チュートリアルの説明どおりに操作します。

また、次の NetBeans のチュートリアルも参照してください。
http://usersguide.netbeans.org/tutorials/webapps/index.html

Sun ONE Studio 5 の詳細については、次の Web サイトを参照してください。
http://www.sun.com/software/sundev/jde/

NSS 3.3.5 および NSPR 4.1.5 のサポート

Sun ONE Web Server 6.1 は、NSS (Network Security Services) 3.3.5 および NSPR (Netscape Portable Runtime) 4.1.5 をサポートします。

PHP との互換性

Sun ONE Web Server 6.1 は、広く使用されている多目的のオープンソース Web スクリプト言語である PHP と互換性があります。PHP (PHP Hypertext Preprocessor) は、すべての主要オペレーティングシステムで動作します。

Sun ONE Web Server 6.1 で使用する場合は、PHP バージョン 4.3.2 をお勧めします。PHP 関連のインストールと設定に関する Sun ONE Web Server に固有の情報については、次の Web サイトを参照してください。
http://www.php.net/manual/en/install.netscape-enterprise.php

強化されたハードウェアアクセラレータによる暗号化のサポート

Sun ONE Web Server 6.1 は、Web サーバーでの SSL のパフォーマンスを向上させる暗号化アクセラレータボード Sun™ Crypto Accelerator 1000 用に、ハードウェアアクセラレータをサポートします。

起動時開始オプション

UNIX プラットフォームの Sun ONE Web Server 6.1 では、起動時開始オプションを利用できます。このオプションを使用すると、システムの起動時に Web サーバーが自動的に起動されるように設定できます。詳細については、『Sun ONE Web Server 6.1 インストールおよび移行ガイド』を参照してください。

その他の機能

複数のプロセス、プロセスモニター、フェイルオーバー、自動回復、ダイナミックログローテーションなどをサポートします。

Sun ONE Web Server の管理と運用

Sun ONE Web Server は、次のユーザーインタフェースを使用して管理できます。

- Sun ONE Web Server 管理サーバー
- サーバーマネージャ
- クラスマネージャ
- 仮想サーバーマネージャ

以前のリリースでは、Web サーバーやその他の Netscape サーバーは、管理サーバーという単一のサーバーで管理されていました。リリース 4.x では、「管理サーバー」は、Sun ONE Web Server の単なる追加のインスタンスの 1 つとなり、Sun ONE Web Server 管理サーバーまたは管理サーバーと呼ばれます。Sun ONE Web Server のすべてのインスタンスの管理に管理サーバーを使用します。詳細は、37 ページの「管理サーバー」を参照してください。

注	設定ファイルを編集したり、コマンド行ユーティリティを使用したりすることで、管理作業を手動で行うこともできます。

Sun ONE Web Server の各インスタンスの管理には、サーバーマネージャを使用します。詳細は、38 ページの「サーバーマネージャ」を参照してください。

仮想サーバーを管理するにはクラスマネージャを使用します。詳細は、39 ページの「クラスマネージャ」を参照してください。

Sun ONE Web Server の設定

Sun ONE Web Server の設定を変更することで、各種機能を有効または無効にしたり、各クライアントの要求に対する応答方法を指定したり、サーバーで稼動しサーバーとの対話を行うプログラムを記述することが可能となります。これらのオプションを識別する命令 (指令と呼ばれます) は、設定ファイルに格納されます。Sun ONE Web Server は、起動時およびクライアントからの要求時に設定ファイルを読み取り、選択内容と適切なサーバーの動作をマッピングします。

これらのファイルの詳細については、『Sun ONE Web Server 6.1 Administrator's Configuration File Reference』を参照してください。

管理サーバー

管理サーバーは、Web ベースのサーバーで、Java フォームが含まれており、これを使用して Sun ONE Web Server のすべてを設定できます。

Sun ONE Web Server をインストールしたら、ブラウザを使用して、管理サーバーのページを開き、フォームに必要事項を入力して Sun ONE Web Server を設定します。フォームを送信すると、Sun ONE Web Server が管理するサーバーの設定を変更します。

管理サーバーのページを開くのに使用する URL は、Sun ONE Web Server のインストール時に指定した、管理サーバーのコンピュータホスト名やポート番号により異なります。たとえば、管理サーバーをポート 1234 にインストールした場合の URL は次のとおりです。

```
http://myserver.sun.com:1234/
```

管理サーバーでは、フォームのページを開く前に、本人の認証を求めるプロンプトが表示されます。ユーザー名とパスワードを入力する必要があります。コンピュータに Sun ONE Web Server をインストールするときに、「スーパーユーザー」のユーザー名とパスワードを設定します。

インストール終了後は、分散管理を使用して、複数のユーザーが管理サーバーの各種フォームにアクセスできるようにすることができます。分散管理については、第 5 章「管理サーバーの設定」の 103 ページの「複数の管理者の許可」を参照してください。

管理サーバーの設定は、複数のタブから構成された右側のウィンドウ枠に表示されます。

管理サーバーにアクセスしたときに表示される最初のページは「Servers」と呼ばれます。このページにある各ボタンを使用して、Sun ONE Web Server の管理、追加、削除、移行を実行します。管理サーバーには管理者レベルのタスク用に次のタブが用意されています。

- Servers
- Preferences
- Global Settings
- Users and Groups
- Security

- Cluster Mgmt (Cluster Management)

注	サーバーの設定に必要な CGI プログラムを起動できるように、ブラウザの cookie を有効にする必要があります。

管理者レベルのタスクについての情報を含む、管理サーバーの使用方法については、45 ページの「Sun ONE Web Server の管理」を参照してください。

サーバーマネージャ

サーバーマネージャは、Web ベースのインタフェースで、Java フォームが含まれており、これを使用して、Sun ONE Web Server の各インスタンスを設定します。

Sun ONE Web Server のサーバーマネージャにアクセスするには、次の手順を実行します。

1. Sun ONE Web Server をインストールし、起動します。

   管理サーバーに「Servers」ページが表示されます。

2. 「Manage Servers」領域から、目的のサーバーを選択し、「Manage」をクリックします。

   Sun ONE Web Server に、サーバーマネージャの「Preferences」ページが表示されます。

注	サーバーの設定に必要な CGI プログラムを実行できるように、ブラウザの cookie を有効にする必要があることに留意してください。

「Preferences」ページのリンクを使用して、スレッドプール設定などのオプションを指定したり、Web サーバーのオン、オフを実行したりします。

さらに、サーバーマネージャには次のタブが用意してあり、Sun ONE Web Server 管理タスクを実行できます。

- Security
- Logs
- Monitor
- Virtual Server Class
- Java

詳細は、オンラインヘルプの「Server Manager」を参照してください。

クラスマネージャ

クラスマネージャは、Web ベースのインタフェースで、Java フォームが含まれており、これを使用して、仮想 Sun ONE Web Server を設定します。仮想サーバーのユーザーインタフェースには、サーバーマネージャとクラスマネージャの 2 つの部分があります。クラスマネージャでは、単一クラスや単一仮想サーバーに影響を与える設定を行います。クラスマネージャでクラスにサービスを設定したり、仮想サーバー (クラスのメンバ) を追加したり、個々の仮想サーバーを設定することができます。

Sun ONE Web Server のクラスマネージャにアクセスするには、次の手順を実行します。

1. サーバーマネージャから、「Virtual Server Class」タブをクリックします。

   サーバーマネージャに「Manage a Class of Virtual Server」ページが表示されます。

2. ドロップダウンリストから、仮想サーバークラスを選択し、「Manage」をクリックします。

   Sun ONE Web Server に、クラスマネージャの「Select a Virtual Server」ページが表示されます。

   クラスマネージャには、画面の右上隅にあるクラスマネージャのリンクを単にクリックするだけでも、アクセスすることができます。

クラスマネージャには、Sun ONE Web Server 仮想サーバーを管理するための次のタブが用意してあります。

- Virtual Servers
- Programs
- Content Management
- Styles

詳細は、オンラインヘルプの「Class Manager」を参照してください。

仮想サーバーマネージャ

仮想サーバーマネージャにアクセスするにはクラスマネージャの「Virtual Servers」タブを表示し、「Manage Virtual Servers」ページのリストから仮想サーバーを選択して、「Manage」をクリックします。またはツリービューで仮想サーバーへのリンクをクリックします。

仮想サーバーマネージャのページから、ステータスや設定を確認したり、Java Web アプリケーションの状態をオンにしたり、選択した仮想サーバーについてのレポートを生成したりすることができます。

仮想サーバーマネージャには、Sun ONE Web Server 仮想サーバーを管理するための次のタブが用意してあります。

- Preferences
- Logs
- Web Applications
- WebDAV
- Search

リソースピッカーの使用

サーバーマネージャとクラスマネージャのページの多くは、Sun ONE Web Server 全体やクラス全体を設定するのに使用します。ただし、サーバー (またはクラス) 全体、またはサーバー (またはクラス) が管理するファイルとディレクトリ、のどちらかを設定できるページもあります。これらのページには、最上部にリソースピッカーがあります。

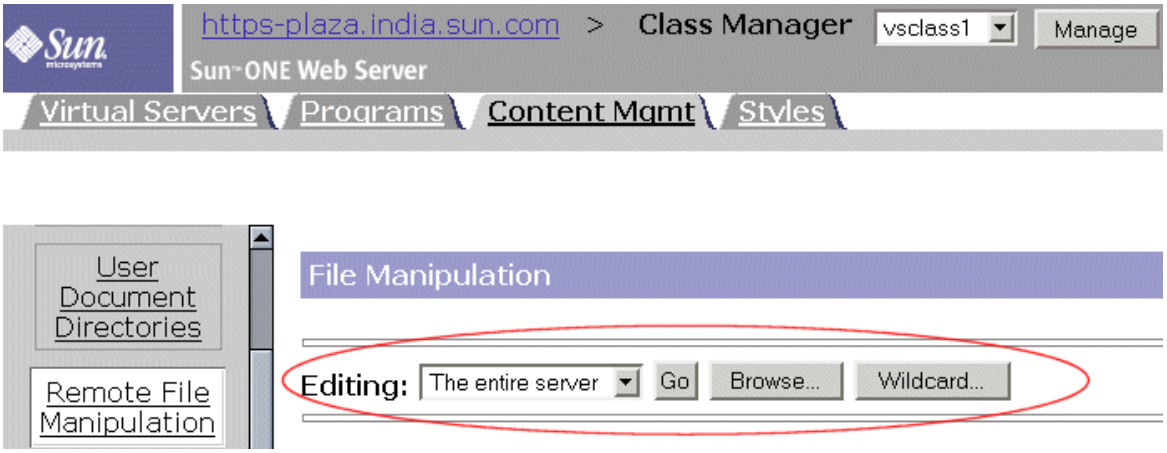


図 1-1　リソース ピッカー

リソースピッカーが表示されているページは数多くあり、サーバーマネージャの「Log Preferences」ページやクラスマネージャの「Content Management」タブからアクセスできる画面の多くが含まれます。

リソースピッカーを使用するには、設定のドロップダウンリストからリソースを選択します。「Browse」をクリックしてプライマリドキュメントを直接参照し、「Wildcard」をクリックして特定の拡張子を持つファイルを設定します。

リソースピッカーで使用するワイルドカード

サーバー設定の大部分で、ワイルドカードパターンを指定して、設定する 1 つ、または、複数の項目を示すことができます。ただし、アクセス制御に用いるワイルドカードと、この節で述べるワイルドカードとは、一部、異なる場合があるため注意が必要です。

ワイルドカードパターンには、特殊文字を使用します。これらの特殊文字を特殊な意味を持たせずに使用したい場合は、1 つの ¥ (円マーク) を該当する特殊文字の前に付けます。

ワイルドカードパターンは、ファイル名だけでなく、ディレクトリパスにも適用されます。このため、ワイルドカードパターンは、特定のディレクトリ内のファイルだけに適用されます。たとえば、/tmp ディレクトリにファイルを追加するには、tmp/*.html というワイルドカードパターンを指定します。すべてのサブディレクトリからの index.html を追加する場合は、パターンは */index.html となります。

表 1 リソースピッカーのワイルドカードパターン

パターン	使用法
*	0 個以上の文字に相当します。
?	任意の文字 1 個に相当します。
|	OR 式を構成します。この演算子とともに使用する部分文字列では、* または $ のような他の特殊文字を含むことができます。部分文字列は (a|b|c) などのようにカッコで囲む必要があります。しかし、( ) を入れ子にして使用することはできません。
$	文字列の末尾に相当します。これを OR 式で使用すると便利です。
[abc]	a、b、または c という文字の 1 回の出現に相当します。この式では、特殊文字として扱われる文字は] だけです。その他の文字は特殊文字として扱われません。
[a-z]	a から z までの文字の 1 回の出現に相当します。
[^az]	a でも z でもない、任意の 1 個の文字に相当します。
*~	この式は、後ろに別の式が続き、2 番目の式と一致するパターンをすべて削除します。

表 2 リソースピッカーのワイルドカードの例

パターン	使用法
*.sun.com	.sun.com という綴りで終わる任意の文字列に相当します。
(products|docs).sun.com	products.sun.com または docs.sun.com のいずれかに相当します。
198.93.9[23].???	198.93.92 または 198.93.93 のいずれかで始まり、任意の 3 文字で終わる数値文字列に相当します。
.	ピリオドが含まれる任意の文字列に相当します。
~sun-	文字列の最初の部分が sun- という綴りではない、任意の文字列に相当します。

表 2　リソースピッカーのワイルドカードの例 (続き)

パターン	使用法
*.sun.com~docs.sun.com	sun.com というドメインに含まれているすべてのホスト (ただし docs.sun.com を除く) に相当します。
*.sun.com~(products|docs|software).sun.com	ホスト products.sun.com、docs.sun.com、および software.sun.com を除き、ドメイン sun.com に含まれるすべてのホストに相当します。
.com~.sun.com	サブドメイン sun.com のホストを除き、ドメイン com に含まれるすべてのホストに相当します。

第 2 章

Sun ONE Web Server の管理

この章では、Sun ONE Web Server 管理サーバーを使用して Sun ONE Web Server 6.1 を管理する方法について説明します。管理サーバーを使用して、サーバーの管理、サーバーの追加や削除、以前のリリースからのサーバーの移行を行うことができます。

この章には、次の内容が記述されています。

- 管理サーバーの起動
- 複数のサーバーの稼動
- 1 つのサーバーへの複数のインスタンスのインストール
- サーバーの削除
- 以前のバージョンからのサーバーの移行

管理サーバーの起動

この節では、UNIX/Linux や Windows プラットフォームで管理サーバーにアクセスする方法を説明します。

UNIX/Linux プラットフォーム

UNIX または Linux プラットフォームで管理サーバーにアクセスするには、次の手順を実行します。

1. *server_root*/https-admserv/ ディレクトリ (たとえば /usr/s1ws61/servers/https-admserv/) に移動します。
2. ./start と入力します。

   このコマンドで、管理サーバーは、インストール時に指定したポート番号を使用して起動します。

Windows プラットフォーム

Windows プラットフォームでは、Sun ONE Web Server インストールプログラムは、1 つのプログラムグループと数個のアイコンを作成します。このプログラムグループには、次のアイコンが含まれます。

- Release Note
- Start Web Server Administration Server
- Uninstall Web Server
- Administer Web Server

管理サーバーは、サービスアプレットとして動作するため、コントロールパネルの画面から、直接、このサービスを起動することもできます。

Windows プラットフォーム で 管理サーバーにアクセスするには、次の手順を実行します。

1. 「Start Web Server Administration Server」アイコンをダブルクリックするか、または、管理サーバーを起動するための次の URL をブラウザに入力します。

   `http://`*hostname.domain-name:administration_port*

   Sun ONE Web Server が起動し、ユーザー名とパスワードを求める画面が表示されます。

2. インストール時に指定した管理者ユーザー名とパスワードを入力します。

   Sun ONE Web Server に、「Administration Server」ページが表示されます。

詳細は、オンラインヘルプの「Administration Server ページ」を参照してください。

注	サーバーの設定に必要な CGI プログラムを起動できるように、ブラウザの cookie を有効にする必要があります。

管理サーバーへは、Netscape Navigator のようなクライアントソフトウェアにアクセスできれば、離れた場所からでもアクセスできます。管理サーバーはブラウザ経由でアクセスできるため、ネットワークを介してサーバーに接続できるマシンならばどのマシンからでも管理サーバーにアクセスできます。

複数のサーバーの稼動

使用しているシステムで Web サーバーを稼動させるには、2 つの方法があります。

- 仮想サーバーを使用する
- 1 つのサーバーに複数のインスタンスをインストールする

仮想サーバー

仮想サーバーを使用すると、インストールされた 1 つのサーバーで、複数の会社または個人に対して、ドメイン名、IP アドレス、およびサーバー監視機能を提供できます。ユーザーにとってはまるで自分の Web サーバーを手に入れたようになりますが、実際に、ハードウェアや Web サーバーの基本的なメンテナンスを提供するのはユーザーではありません。

仮想サーバーの設定は、*server_root/server_id*/config ディレクトリの server.xml ファイルに保存されます。仮想サーバーを使用するのにこのファイルを編集する必要はありませんが、このファイルについてさらに詳細を知りたい場合は、『Sun ONE Web Server 6.1 Administrator's Configuration File Reference』を参照してください。

仮想サーバーについては、第 13 章「仮想サーバーの使用」を参照してください。

1 つのサーバーへの複数のインスタンスのインストール

Sun ONE Web Server の以前のリリースでは、仮想サーバーごとの固有の設定情報がありませんでした。サーバーに別個の設定情報を持たせる唯一の方法は、新規にサーバーインスタンスを生成することでした。しかし、Sun ONE Web Server 6.1 では、仮想サーバーは別個の設定情報を持つことができるため、複数のサーバーインスタンスは必要なくなりました。従来の機能もサポートされていますが、複数のサーバーを持つには仮想サーバーを使用することが推奨されます。

Web サーバーの複数のインスタンスのインストールを選択する場合、管理サーバーを使用して、次のどちらかを実行します。

- Windows 上のサーバーの複数のコピーを、別々のインスタンスとして、それぞれに異なる IP アドレスをつけて、インストールします。
- すべて同じ IP アドレス (ポート番号は異なる) を使用する一連のサーバーを設定します。

システムを複数の IP アドレスで待機するように設定している場合は、インストールするサーバーごとに、システムに割り当てられている IP アドレスの 1 つを入力します。

システムを複数の IP アドレスが割り当てられるように設定する前に、すでにサーバーをインストールしていた場合は、システムの設定を別の複数の IP アドレスに対応できるように変更します。このあとで、ハードウェア仮想サーバーをインストールするか、または、サーバーマネージャを使用してサーバーのバインドアドレスを変更して、IP アドレスごとにサーバーの別個のインスタンスをインストールします。

別のサーバーインスタンスを追加するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Servers」タブを選択します。
2. 「Add Server」リンクをクリックします。
3. 指定されたフィールドに必要な情報を入力します。

   サーバー識別子は、数字で始めることはできません。また、インスタンス名には Latin-1 文字のみを使用する必要があります。
4. 「OK」をクリックします。

詳細は、オンラインヘルプの「Add Server ページ」を参照してください。

サーバーの削除

管理サーバーを使用して、システムからサーバーを削除できます。このプロセスは元に戻すことはできないため、削除する前に、そのサーバーを今後使用することがないかどうか確認してください。

注	Windows サーバーには、アンインストールプログラムがついているものがあり、これを使用してサーバーや関連の管理サーバーを削除することができます。詳細は、製品に付属しているマニュアルを確認してください。

使用しているマシンからサーバーを削除するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Servers」タブを選択します。
2. 「Remove Server」をクリックします。
3. 削除するサーバーを選択し、「Yes」をクリックします。
4. 「OK」をクリックします。

管理サーバーは、続いて、サーバーの設定ファイル、サーバーマネージャフォーム、さらに、次のディレクトリ (およびサブディレクトリ) を削除します。

server_root/https-*server-id*

詳細は、オンラインヘルプの「Remove Server ページ」を参照してください。

以前のバージョンからのサーバーの移行

Sun ONE Web Server は、バージョン 4.1 または 6.0 から 6.1 に移行することができます。それまでの 4.1 または 6.0 サーバーは保持され、新たに、6.1 サーバーが同じ設定を使用して作成されます。

設定の移行の前に、4.1 または 6.0 サーバーの稼動を停止する必要があります。設定の移行の前に、コンピュータにインストールされている Web ブラウザのバージョンと互換性があるかどうか確認します。

以前のバージョンから Sun ONE Web Server 6.1 へのサーバーの移行方法についての詳細は、『インストールおよび移行ガイド』を参照してください。

詳細は、オンラインヘルプの「Migrate Server ページ」を参照してください。

第 2 部

管理サーバーの使用

第 3 章「ユーザーとグループの管理」

第 4 章「Web コンテナと Web アプリケーションの J2EE ベースのセキュリティ」

第 5 章「管理サーバーの設定」

第 6 章「証明書と鍵の使用」

第 7 章「サーバークラスタの管理」

第 3 章

ユーザーとグループの管理

この章では、Sun ONE Web Server にアクセスするユーザーとグループの追加、削除、編集について説明します。

この章には、次の内容が記述されています。

- ユーザーとグループに関する情報へのアクセス
- ディレクトリサービスについて
- ディレクトリサービスの設定
- ユーザーの作成
- ユーザーの管理
- グループの作成
- グループの管理
- 組織単位の作成
- 組織単位の管理

ユーザーとグループに関する情報へのアクセス

管理サーバーを使用して、ユーザーアカウント、グループリスト、アクセス特権、組織単位、その他のユーザーやグループに固有の情報に関するアプリケーションデータにアクセスできます。

ユーザーとグループの情報は、テキスト形式のフラットファイル、または LDAP (Lightweight Directory Access Protocol) をサポートする Sun ONE Directory Server などのディレクトリサーバーに格納されます。LDAP は、オープンディレクトリアクセスプロトコルで、TCP/IP 上で動作し、グローバルサイズに、また百万単位のエントリにまで拡張可能です。

Sun ONE Web Server はローカル LDAP をサポートしていないため、ユーザーやグループを追加する場合、事前にディレクトリサーバーをインストールしておく必要があります。

ディレクトリサービスについて

Sun ONE Directory Server などのディレクトリサーバーを使用することで、単一ソースからのすべてのユーザー情報を管理できます。ディレクトリサーバーを設定すると、ネットワークに容易にアクセスできる複数の場所から、ユーザーがディレクトリ情報を取得できるようになります。

Sun ONE Web Server 6.1 では、3 種類のディレクトリサービスを設定して、ユーザーおよびグループの認証と承認を行えます。ディレクトリサービスが設定されていない場合、作成される新しいディレクトリサービスには、種類に関係なく `default` という値が設定されます。

ディレクトリサービスを作成すると、ディレクトリサービスの詳細によって *server-root*`/userdb/dbswitch.conf` ファイルが更新されます。

ディレクトリサービスの種類

Sun ONE Web Server 6.1 がサポートするディレクトリサービスには、次の種類があります。

- LDAP: ユーザーおよびグループ情報を LDAP ベースのディレクトリサーバーに格納します。

  LDAP サービスがデフォルトサービスの場合、`dbswitch.conf` ファイルが次の例のように更新されます。

  ```
  directory default
  ldap://draco.india.sun.com:589/dc%3Dindia%2Cdc%3Dsun%2Cdc%3Dcom

  default:binddn cn=Directory Manager

  default:encoded bindpw YWRtaW5hZG1pbg==
  ```

  LDAP サービスがデフォルト以外のサービスの場合、`dbswitch.conf` ファイルが次の例のように更新されます。

  ```
  directory ldap
  ldap://draco.india.sun.com:589/dc%3Dindia%2Cdc%3Dsun%2Cdc%3Dcom

  ldap:binddn cn=Directory Manager

  ldap:encoded bindpw YWRtaW5hZG1pbg==
  ```

- **鍵ファイル**：鍵ファイルは、ハッシュ形式のユーザーパスワード、およびそのユーザーが所属するグループのリストが含まれているテキストファイルです。鍵ファイルに格納されたユーザーとグループは、認証と承認のために `file` レルムだけで使用され、システムのユーザーおよびグループとは関連しません。`file` レルムの詳細については、90 ページの「file レルム」を参照してください。

  鍵ファイルベースのデータベースを作成すると、`dbswitch.conf` ファイルが次の例のように更新されます。

  ```
  directory keyfile file

  keyfile:syntax keyfile

  keyfile:keyfile D:¥draco¥keyfile¥keyfiledb
  ```

- **ダイジェストファイル**：ユーザーとグループの情報を、暗号化されたユーザー名とパスワードに基づいて格納します。

  鍵ファイルベースのデータベースを作成すると、`dbswitch.conf` ファイルが次の例のように更新されます。

  ```
  directory digest file

  digest:syntax digest

  digest:digestfile D:¥draco¥digest¥digestdb
  ```

注 分散管理を設定するときは、LDAP ベースのディレクトリサービスをデフォルトディレクトリとする必要があります。

ディレクトリサービスの設定

ディレクトリサービスの詳細を設定するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Global Settings」タブをクリックします。
2. 「Configure Directory Service」リンクをクリックします。
3. 「Create New Service of Type」ドロップダウンリストから、作成するディレクトリサービスの種類を選択します。
4. 「New」をクリックします。

   選択したディレクトリサービスの種類に対応するページで、ディレクトリサービスの情報を設定できるようになります。

注 ディレクトリサービスが設定されていない場合、作成される新しいディレクトリサービスには、種類に関係なく `default` という値が設定されます。

5. 「OK」をクリックして変更を保存します。

ディレクトリサービスを作成および設定すると、仮想サーバーごとにディレクトリサービスを割り当てられるようになります。ディレクトリサービスに関連する権限とアクセス権は、アクセス制御規則を評価および適用するときに、サーバーによって使用されます。詳細は、348 ページの「仮想サーバーのディレクトリサービスの選択」を参照してください。

識別名 (DN) の理解

管理サーバーの「Users and Groups」タブを使用して、ユーザー、グループ、および組織単位を作成したり、変更したりします。ユーザーとは、企業の社員などのように、LDAP データベース内の個人を意味します。グループとは、同じ属性を共有する複数のユーザーを意味します。組織単位は、`organizationalUnit` オブジェクトクラスを使用する企業内の区分を意味します。ユーザー、グループ、および組織単位については、この章の最後に詳細を説明します。

企業内のユーザーやグループは、それぞれ、識別名 (DN) 属性で表されます。DN 属性は、関連するユーザー、グループ、またはオブジェクトを識別する情報が記述されたテキスト文字列です。ユーザーやグループのディレクトリエントリを変更する場合は、必ず DN を使用します。たとえば、ディレクトリエントリを作成したり変更したり、アクセス制御を設定したり、メールまたはパブリッシングなどのアプリケーション用のユーザーアカウントを設定したりする場合はその都度、DN 情報を指定する必要があります。DN を作成または変更するときは、Sun ONE Web Server 管理コンソールでユーザーとグループのインタフェースを使用すると便利です。

次の例は、Sun Microsystems の社員の一般的な DN を表しています。

```
uid=doe,e=doe@sun.com,cn=John Doe,o=Sun Microsystems Inc.,c=US
```

この例では、各等号の前の略語は、それぞれ次の意味を示します。

- `uid`: ユーザー ID
- e: 電子メールアドレス
- cn: ユーザーの共通名
- o: 組織名
- c: 国名

DN には、さまざまな名前 - 値の組み合わせを含めることができます。DN は、証明書の項目、および LDAP をサポートするディレクトリ内のエントリの両方を識別するために使用されます。

LDIF の使用

この時点でまだディレクトリがない場合、または、既存のディレクトリに新規のサブツリーを追加したい場合、Directory Server の管理サーバー LDIF インポート機能を使用できます。この機能を使用すれば、LDIF を含むファイルを取り扱うことができ、ディレクトリを構築したり、LDIF エントリから新規のサブツリーを構築することが可能です。また、Directory Server の LDIF エクスポート機能を使用して、現在のディレクトリを LDIF へエクスポートすることもできます。この機能は、該当するディレクトリを表す LDIF 書式のファイルを作成します。エントリは、`ldapmodify` コマンドを適切な LDIF 更新文とともに使用して追加、編集します。

LDIF を使用してデータベースにエントリを追加するには、まず、`LDIF` ファイル内のエントリを定義し、次に、Directory Server から `LDIF` ファイルをインポートします。

ユーザーの作成

管理サーバーの「Users and Groups」タブを使用して、ユーザーエントリを作成したり、変更したりします。ユーザーエントリには、データベース内の個人やオブジェクトに関する情報があります。

ユーザーを作成するときは、そのユーザーがリソースに不正にアクセスできないように設定することで、サーバーのセキュリティを保護する必要があります。Sun ONE Web Server 6.1 には、セキュリティ強化のためのオプションが数多く用意されています。

- J2EE およびサーブレットベースのレルムの認証を使用してユーザーを認証および承認する方法については、89 ページの「レルムベースのセキュリティ」を参照してください。
- アクセス制御リスト (ACL) ベースの認証および承認の使用方法については、191 ページの「アクセス制御のしくみ」を参照してください。
- Java ベースのセキュリティモデルと ACL ベースのセキュリティモデルを結ぶ native レルム機能の使用については、95 ページの「native レルムの設定」を参照してください。

この節では、次の内容について説明します。

- LDAP ベース認証データベースの新規ユーザーの作成
- ファイルベース認証データベースの新規ユーザーの作成
- ダイジェストベース認証データベースの新規ユーザーの作成

LDAP ベース認証データベースの新規ユーザーの作成

LDAP ベースのディレクトリサービスにユーザーエントリを追加すると、ユーザーの認証と承認に、基本となる LDAP ベースディレクトリサーバーのサービスが使用されます。この項では、LDAP ベースの認証データベースを使用する場合に注意すべきガイドラインを示し、管理サーバーを通じてユーザーを追加する方法について説明します。

- LDAP ベースユーザーエントリ作成のガイドライン
- 新規ユーザーエントリの作成方法
- Directory Server のユーザーエントリ

LDAP ベースユーザーエントリ作成のガイドライン

管理者フォームを使用して LDAP ベースのディレクトリサービスに新しいユーザーエントリを作成するときは、次のガイドラインを考慮してください。

- 名 (ファーストネーム) および姓を入力すると、フォームにはユーザーのフルネームとユーザー ID が自動的に入力されます。ユーザー ID は、ユーザーのファーストネームの最初の 1 文字の後にユーザーのラストネームを組み合わせて生成されます。たとえば、ユーザーの名前が Billie Holiday の場合、ユーザー ID は、自動的に bholiday となります。このユーザー ID は、必要に応じて、独自に作成する ID と置き換えることができます。
- ユーザー ID は一意である必要があります。管理サーバーは、検索ベース (ベース DN) の下のディレクトリ全体を検索し、同じユーザー ID が使われていないかを調べて、ユーザー ID が一意であることを確認します。ただし、Directory Server の `ldapmodify` コマンド行ユーティリティを使用して (使用可能ならば)、ユーザーを作成する場合は、ユーザー ID が一意であるかどうかは確認されないため注意が必要です。ディレクトリに重複したユーザー ID が存在していた場合、該当するユーザーは、そのディレクトリでは認証されなくなります。
- ベース DN は、識別名を指定します。それはディレクトリの検索がデフォルトで実行され、Sun ONE Web 管理サーバーのエントリがすべてディレクトリツリーに配置される場所です。「DN」は、ディレクトリサーバーのエントリ名を表す文字列です。
- 新規にユーザーエントリを作成する場合、少なくとも次のユーザー情報を指定してください。
  - 姓 (ラストネーム)
  - フルネーム
  - ユーザー ID

- 組織単位がディレクトリに定義されている場合、「Add New User To」リストを使用して、新規のユーザーを配置したい場所を指定できます。デフォルトの場所は、ディレクトリのベース DN (またはルートポイント) になります。

注　国際情報についてのユーザーが編集するテキストフィールドは、管理サーバーと Sun ONE Web Server 管理コンソールでは異なります。Sun ONE Web Server 管理コンソールには、タグのない cn フィールドのほかに、言語を選択する cn フィールドがありますが、管理サーバーにはこのフィールドはありません。

新規ユーザーエントリの作成方法

ユーザーエントリを作成するには、58 ページの「LDAP ベースユーザーエントリ作成のガイドライン」に記載のガイドラインを読み、その後で次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「New User」リンクをクリックします。
3. 「Select Directory service」ドロップダウンリストから「LDAP Directory Service」を選択し、「Select」をクリックします。
4. 表示されるページに必要な情報を入力します。

   詳細は、「Directory Server のユーザーエントリ」を参照してください。
5. 「OK」をクリックします。

詳細は、オンラインヘルプの「New User ページ」を参照してください。

Directory Server のユーザーエントリ

次のユーザーエントリについての注意は、主にディレクトリ管理者を対象としています。

- ユーザーエントリは、`inetOrgPerson`、`organizationalPerson`、および `person` オブジェクトクラスを使用します。
- デフォルトでは、ユーザーの識別名のフォームは次のとおりです。

  ```
  cn=full name, ou=organization, ...,o=base organization, c=country
  ```

  たとえば、Billie Holiday のユーザーエントリを Marketing という組織単位内に作成し、ディレクトリのベース DN が o=Ace Industry、c=US の場合、このユーザーの DN は次のようになります。

  ```
  cn=Billie Holiday, ou=Marketing, o=Ace Industry, c=US
  ```

  ただし、この書式は、uid ベースの識別名に変更することができます。

- ユーザーフォームフィールドの値は、次の LDAP 属性として格納されます (「ユーザー」と「グループ」以外の情報を格納する場合はすべて、Directory Server のフルライセンスが必要です)。

表 3-1　LDAP 属性

ユーザーフィールド	対応する LDAP 属性
Given Name (名、ファーストネーム)	`givenName`
Surname (姓、ラストネーム)	`sn`
Full Name (フルネーム)	`cn`
User ID (ユーザー ID)	`uid`
Password (パスワード)	`userPassword`
Email Address (電子メールアドレス)	`mail`

ユーザーエントリを編集する際、次のフィールドもまた使用可能です。

表 3-2　ユーザーエントリ LDAP 属性

ユーザーフィールド	対応する LDAP 属性
Title (役職名)	`title`
Telephone (電話)	`telephoneNumber`

- ユーザーの名前は、デフォルト言語以外の言語の文字で表現する方がより正確に表現できる場合があります。デフォルト言語が英語の場合でも、ユーザーが使用する preferred language (言語) を選択して、ユーザー名をその言語の文字で表示することができます。ユーザーの preferred language の設定については、オンラインヘルプの「Manage Users ページ」を参照してください。

ファイルベース認証データベースの新規ユーザーの作成

Sun ONE Web Server 6.1 では、ユーザー情報をテキスト形式のフラットファイルに格納するネイティブ認証データベースをサポートするようになりました。ファイルベースの認証データベースは、次の種類のファイルと互換性があります。

- 鍵ファイルスタイルのファイル
- ダイジェストスタイルのファイル
- htaccess スタイルのファイル

新規ユーザーエントリの作成

ファイルベースの認証データベースにユーザーエントリを作成するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「New User」リンクをクリックします。
3. 「Select Directory service」ドロップダウンリストからファイルベースのディレクトリサービスを選択し、「Select」をクリックします。
4. 次の情報を入力します。
    - User ID: (必須) 一意のユーザー名を指定します。
    - Password: ユーザーのパスワードを指定します。
    - Password (again): 「Password」フィールドで入力したパスワードを確認します。
    - Groups: コンマで区切って指定されているグループのリストからユーザーがメンバーとなっているグループを指定します。
5. 「Create User」をクリックします。

ダイジェストベース認証データベースの新規ユーザーの作成

ユーザーとグループの情報を暗号化された形式で格納するダイジェストベースの認証データベースにユーザーエントリを作成するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「New User」リンクをクリックします。
3. 「Select Directory Service」ドロップダウンリストからダイジェストベースのディレクトリサービスを選択し、「Select」をクリックします。
4. 次の情報を入力します。
   - User ID: **(必須)** 一意のユーザー名を指定します。
   - Realm: このユーザーを認証するレルムを指定します。
   - Password: ユーザーのパスワードを指定します。
   - Password (again): 「Password」フィールドで入力したパスワードを確認します。
   - Groups: コンマで区切って指定されているグループのリストからユーザーがメンバーとなっているグループを指定します。
5. 「OK」をクリックします。

ユーザーの管理

ユーザーの属性は、管理サーバーの「Manage Users」フォームから編集できます。このフォームを使用して、ユーザーエントリの検索、変更、名前の変更、削除を行なったり、ユーザーライセンスを管理したりすることができます。また、製品固有の情報を変更できる場合もあります。

Sun ONE サーバーの中には、この領域に製品固有の情報を管理するためのフォームを追加しているものもあります。たとえば、メッセージングサーバーが管理サーバー配下にインストールされている場合、メッセージングサーバー固有の情報を編集できるようにフォームが追加されています。このような追加の管理機能については、サーバーのマニュアルを参照してください。

この節では、次の内容について説明します。

- ユーザー情報の検索
- ユーザー情報の編集
- ユーザーのパスワードの管理
- ユーザーライセンスの管理

- ユーザー名の変更
- ユーザーの削除

ユーザー情報の検索

ユーザーエントリを編集する際は、事前に関連情報を表示する必要があります。特定のユーザー情報を検索するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「Manage Users」リンクをクリックします。
3. 「Find User」フィールドに、編集したいエントリに関連する文字 (値) を入力します。検索フィールドに入力できる値は、次のとおりです。
   - 名前。フルネーム、または名前の一部を入力します。検索文字列と完全に一致するすべてのエントリが返されます。これに該当するエントリがない場合は、検索文字列を含むすべてのエントリが検索されます。これにも該当しない場合は、検索文字列と類似したエントリが検索されます。
   - ユーザー ID。
   - 電話番号。電話番号の一部だけを入力すると、最後の部分が検索番号に一致する電話番号を含むエントリがすべて返されます。
   - 電子メールアドレス。アットマーク (@) 記号を含む検索文字列は、すべて、電子メールアドレスとして認識されます。完全に一致するエントリがない場合は、検索文字列で始まる電子メールアドレスがすべて検索されます。
   - アスタリスク (*) を入力すると、現在ディレクトリにあるエントリがすべて表示されます。検索フィールドに何も入力しないで検索しても、同じ結果が得られます。
   - 任意の LDAP 検索フィルタ。等号 (=) を含む文字列はすべて、検索フィルタとして認識されます。

   他の方法としては、「Find all users whose」フィールドのドロップダウンメニューを使用して、検索結果を絞り込む方法もあります。
4. 「Look within」フィールドで、エントリの検索を行う組織単位を選択します。

   デフォルトは、ディレクトリのルートポイント (つまり最上位のエントリ) です。
5. 「Format」フィールドで、「On-Screen」または「Printer」を選択します。
6. 「Find」をクリックします。

   選択された組織単位に含まれるユーザーがすべて表示されます。
7. 検索結果テーブルで、編集したいエントリの名前をクリックします。

   ユーザー編集フォームが表示されます。

8. 必要に応じてフィールドの変更を行い、「Save Changes」をクリックします。

   変更は、すぐに有効となります。

カスタム検索クエリの構築

「Find all users whose」フィールドを使用して、カスタム検索フィルタを構築できます。このフィールドを使用して、「Find user」検索で返される検索結果を絞り込みます。

「Find all users whose」フィールドでは、次のような検索条件を使用します。

- 一番左のドロップダウンリストでは、検索の基準となる属性を指定できます。

次の表は、使用可能な検索属性のオプションを示します。

表 3-3　検索属性オプション

オプション名	説明
full name (フルネーム)	各エントリのフルネームで一致しているものを検索します。
last name (ラストネーム)	各エントリのラストネーム (姓) で一致しているものを検索します。
user id (ユーザー ID)	各エントリのユーザー ID で一致しているものを検索します。
phone number (電話番号)	各エントリの電話番号で一致しているものを検索します。
email address (電子メールアドレス)	各エントリの電子メールアドレスで一致しているものを検索します。
unit name (組織単位名)	各エントリの組織単位名で一致しているものを検索します。
description	各エントリの記述で一致しているものを検索します。

- 中央のドロップダウンリストでは、実行する検索のタイプを選択します。

次の表は、使用可能な検索タイプのオプションを示します。

表 3-4　検索タイプのオプション

オプション名	説明
contains	部分文字列検索を実行します。指定した検索文字列を含む属性値のエントリを返します。たとえば、ユーザー名に「Dylan」が含まれているとわかっている場合、このオプションを使用して、検索文字列に「Dylan」と入力しユーザーのエントリを検索します。
is	正確に一致するものを検索します。すなわち、このオプションは完全一致検索を実行します。正確なユーザーの属性値がわかっているときには、このオプションを使用します。たとえば、ユーザー名の正確なスペルがわかっている場合は、このオプションを使用します。
isn't	属性値が検索文字列と完全一致ではないエントリをすべて返します。たとえば、ユーザー名が「John Smith」でない、ディレクトリ内のすべてのユーザーを検索したい場合、このオプションを使用します。ただし、このオプションを使用すると返されるエントリ数が膨大になるため、注意が必要です。
sounds like	近似検索、または表記の似た検索を実行します。属性のおよその値はわかっているが、スペルが正確にはわからない場合に、このオプションを使用します。たとえば、ユーザー名のスペルが、「Sarret」、「Sarette」、または「Sarett」か、不確かな場合には、このオプションを使用します。
starts with	部分文字列検索を実行します。属性値が指定した検索文字列で始まるエントリをすべて返します。たとえば、ユーザー名が「Miles」で始まるのはわかっているが、名前の残りの部分がわからない場合に、このオプションを使用します。
ends with	部分文字列検索を実行します。属性値が指定した検索文字列で終わるエントリをすべて返します。たとえば、ユーザー名が「Dimaggio」で終わるのはわかっているけれども、名前の残りの部分がわからない場合には、このオプションを使用します。

- 一番右側のテキストフィールドに、検索文字列を入力します。

Look Within ディレクトリ内のユーザーエントリをすべて表示するには、テキストフィールドに、アスタリスク (*) を入力するか、または、何も入力せずに検索します。

ユーザー情報の編集

ユーザーエントリを変更するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 63 ページの「ユーザー情報の検索」の説明に従って、ユーザーエントリを表示します。
3. 変更したい属性に対応するフィールドを編集します。

詳細は、オンラインヘルプの「Edit Users ページ」を参照してください。

注	変更したい属性値が「edit user」フォームに表示されていない場合でも、変更は可能です。この場合、Directory Server の `ldapmodify` コマンド行ユーティリティが使用できる場合は、これを使用します。

また、このフォームからユーザーのファーストネーム、ラストネーム、およびフルネームのフィールドを変更することができますが、エントリ (エントリの識別名を含む) の名前を完全に変更するには、「Rename User」フォームを使用する必要があることに注意してください。エントリの名前の変更については、67 ページの「ユーザー名の変更」を参照してください。

ユーザーのパスワードの管理

ユーザーエントリに設定するパスワードは、ユーザー認証のためにさまざまなサーバーに使用されます。

ユーザーのパスワードを変更または作成するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 63 ページの「ユーザー情報の検索」の説明に従って、ユーザーエントリを表示します。
3. 変更を行い、「OK」をクリックします。

詳細は、オンラインヘルプの「Manage Users ページ」を参照してください。

注	管理サーバーのユーザーを、root からオペレーティングシステム上の別のユーザーに変更できます。これにより、同じグループに属する複数のユーザーが設定ファイルを編集、管理できるようになります。ただし、UNIX/Linux プラットフォームの場合は、インストーラが任意のグループに設定ファイルの「rw」(読み書き) を許可しますが、Windows プラットフォームの場合は、ユーザーは「Administrators」グループに属している必要があります

ユーザーのパスワードは、「Disable Password」ボタンをクリックして無効にすることができます。こうすることで、そのユーザーのディレクトリエントリを削除せずに、そのユーザーがサーバーへログインできなくすることができます。このユーザーに再度アクセスを許可するには、「Password Management」フォームを使用して、新しいパスワードを入力します。

ユーザーライセンスの管理

管理サーバーを使用して、ユーザーが使用許可のライセンスを持っている Sun ONE サーバー製品を調べることができます。

ユーザーが使用可能なライセンスを管理するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 63 ページの「ユーザー情報の検索」の説明に従って、ユーザーエントリを表示します。
3. 「User Edit」フォームの上部にある「Licenses」リンクをクリックします。
4. 変更を行い、「OK」をクリックします。

詳細は、オンラインヘルプの「Manage Users ページ」を参照してください。

ユーザー名の変更

名前の変更機能は、ユーザーの名前だけ変更します。他のフィールドは変更されません。また、ユーザーの古い名前は残されたままなので、古い名前で検索しても新しいエントリが表示されます。

ユーザーエントリ名を変更する場合、変更できるのはユーザー名だけです。名前の変更機能を使って、エントリを 1 つの組織単位から別の組織単位へ移動することはできません。たとえば、Marketing と Accounting という組織単位があり、「Billie Holiday」というエントリが Marketing 組織単位に属していると仮定します。エントリの名前は、Billie Holiday から Doc Holiday に変更できますが、Marketing 組織単位所属の Billie Holiday を Accounting 組織単位所属の Billie Holiday にするようエントリの名前を変更することはできません。

ユーザーエントリの名前を変更するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 63 ページの「ユーザー情報の検索」の説明に従って、ユーザーエントリを表示します。

   共通名ベースの DN を使用している場合は、ユーザーのフルネームを指定するようにしてください。UID ベースの識別名を使用している場合は、エントリに使用したい新規の UID 値を入力します。
3. 「Rename User」ボタンをクリックします。
4. エントリの新しい識別名に合わせて、「Given Name」、「Surname」、「Full Name」または「UID」フィールドを変更します。
5. エントリ名を変更するときに keepOldValueWhenRenaming パラメータを「false」に設定すれば、古いフルネームや古い UID 値を今後保持しないよう管理サーバーに指示できます。このパラメータは、次のファイルにあります。

   server_root`/admin-serv/config/dsgw-orgperson.conf`

詳細は、オンラインヘルプの「Manage Users ページ」を参照してください。

ユーザーの削除

ユーザーエントリを削除するには、次の手順を実行します。

1. 管理者サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 63 ページの「ユーザー情報の検索」の説明に従って、ユーザーエントリを表示します。
3. 「Delete User」をクリックします。

詳細は、オンラインヘルプの「Manage Users ページ」を参照してください。

グループの作成

グループは、LDAP データベースにおいてオブジェクトのセットを表現するオブジェクトです。Sun ONE Web Server グループは、共通する属性を共有する複数のユーザーで構成されています。たとえば、会社のマーケティング部門で働く多数の従業員がオブジェクトのセットになります。この従業員たちは、「Marketing」というグループに属します。

グループのメンバーシップを定義するには、スタティックな方法とダイナミックな方法の 2 つがあります。スタティックグループは、メンバーオブジェクトを明示的に列挙します。スタティックグループは CN であり、`uniqueMembers`、`memberURLs`、`memberCertDescriptions` のいずれかまたはそのすべてが含まれます。スタティックグループでは、メンバーは、`CN=<Groupname>` 属性以外の共通の属性は共有しません。

ダイナミックグループでは、LDAP URL を使用してグループメンバーだけと一致させるための規則セットを定義できます。ダイナミックグループでは、メンバーは共通属性、または `memberURL` フィルタに定義される属性セットを共有します。たとえば、Sales 部門のすべての従業員を含むグループが必要で、全員がすでに「`ou=Sales,o=Airius.com`」の下の LDAP に含まれている場合は、次の `memberurl` を使用してダイナミックグループを定義します。

```
ldap:///ou=Sales,o=sun??sub?(uid=*)
```

このグループは、「`ou=Sales,o=sun`」ポイントの下のツリーで、`uid` 属性を持つすべてのオブジェクト、つまりすべての Sales メンバーを持つことになります。

スタティックおよびダイナミックグループで、`memberCertDescription` を使用している場合は、メンバーは証明書から共通の属性を共有できます。ただし、これは、ACL が SSL メソッドを使用している場合だけ有効となります。

新規グループを作成したら、このグループにユーザーやメンバーを追加することができます。

この節では、次の内容について説明します。

- スタティックグループ
- ダイナミックグループ

スタティックグループ

管理サーバーを使って、複数ユーザーの DN の中に、同じグループ属性を指定して、スタティックグループを作成できます。スタティックグループは、ユーザーの追加や削除を実行しないかぎり、変更されることはありません。

スタティックグループ作成のガイドライン

新規のスタティックグループを作成するために管理サーバーのフォームを使用するときには、次のガイドラインを考慮してください。

- スタティックグループには、その他のスタティックグループまたはダイナミックグループを含めることができます。
- また、任意で、新規グループに説明を追加できます。
- 組織単位がディレクトリにすでに定義されている場合、「Add New Group To」リストを使用して、新規のグループを配置する場所を指定できます。デフォルトの場所は、ディレクトリのルートポイント (つまり最上位のエントリ) です。
- 必要な情報の入力が終わったら、「Create Group」をクリックして新規グループを追加すると、ただちに「New Group」フォームの画面に戻ります。別の方法として、「Create and Edit Group」をクリックしグループを追加して、追加したグループの「Edit Group」フォームに進む方法もあります。グループの編集については、76 ページの「グループ属性の編集」を参照してください。

スタティックグループを作成するには

スタティックグループエントリを作成するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「New Group」リンクをクリックします。
3. 必要な情報を入力し、「OK」をクリックします。

詳細は、オンラインヘルプの「New Group ページ」を参照してください。

ダイナミックグループ

1 つのダイナミックグループは、groupOfURLs に対応する 1 つの objectclass、0 個以上の memberURL 属性、および、それぞれのオブジェクトのセットを説明する LDAP URL を持ちます。

Sun ONE Web Server では、任意の属性に基づいてユーザーを自動的にグループ化する場合、または一致する DN を含む特定のグループに ACL を適用する場合に、ダイナミックグループを作成できます。たとえば、department=marketing という属性を持つ任意の DN を自動的に含めるグループを作成することができます。department=marketing に検索フィルタを適用すると、検索結果は、department=marketing 属性を含むすべての DN からなるグループを返します。次に、このフィルタに基づいて検索結果からダイナミックグループを定義できます。さらに、結果として生成されるダイナミックグループの ACL を定義できます。

この項では、次の内容について説明します。

- Sun ONE Web Server がダイナミックグループを実装するしくみ
- スタティックでダイナミックなグループ
- ダイナミックグループがサーバーパフォーマンスに与える影響
- ダイナミックグループ作成のガイドライン
- ダイナミックグループを作成するには

Sun ONE Web Server がダイナミックグループを実装するしくみ

Sun ONE Web Server は、ダイナミックグループを objectclass = groupOfURLs として LDAP サーバースキーマ内に実装します。groupOfURLS クラスは、複数の memberURL 属性を持つことができます。この属性は、それぞれがディレクトリ内のオブジェクトセットを列挙する LDAP URL から構成されます。グループのメンバーは、これらのセットの組み合わせです。たとえば、次のグループには 1 つのメンバー URL だけが含まれます。

```
ldap:///o=mcom.com??sub?(department=marketing)
```

この例は、部署名が「marketing」である「o=mcom.com」の下のすべてのオブジェクトから構成されるセットを示しています。LDAP URL は、検索ベース DN、スコープ、フィルタを含むことができますが、ホスト名とポートを含むことはできません。つまり、同じ LDAP サーバー上のオブジェクトだけを参照できます。すべてのスコープがサポートされます。

グループに DN を個別に追加しなくても、すべての DN が自動的に含まれます。Sun ONE Web Server は ACL 検証でグループルックアップが必要になるたびに LDAP サーバー検索を行うため、グループはダイナミックに変化します。ACL ファイルで使用されるユーザー名とグループ名は、LDAP データベース内のオブジェクトの cn 属性に対応します。

注	Sun ONE Web Server は cn （commonName）属性を ACL のグループ名として使用します。

ACL から LDAP データベースへのマッピングは、dbswitch.conf 設定ファイル (ACL データベース名と実際の LDAP データベース URL を関連付けます) と ACL ファイル (どの ACL でどのデータベースが使用されるかを定義します) の両方に定義されます。たとえば、「staff」というグループのメンバーシップに基本アクセス権を設定する場合、ACL コードは groupOf<anything> というオブジェクト クラスを持ち、CN が「staff」に設定されているオブジェクトを検索します。オブジェクトは、メンバー DN を明示的に列挙するか (スタティックグループの groupOfUniqueNames と同様)、または LDAP URL を指定することによって (たとえば groupOfURLs)、グループのメンバーを定義します。

スタティックでダイナミックなグループ

グループは、objectclass = groupOfUniqueMembers と objectclass = groupOfURLs の両方の属性を持つことにより、「uniqueMember」属性と「memberURL」属性の両方を有効にできます。グループのメンバーシップには、スタティックなメンバーとダイナミックなメンバーが混在します。

ダイナミックグループがサーバーパフォーマンスに与える影響

ダイナミックグループを使用した場合、サーバーのパフォーマンスに影響が生じます。グループメンバーシップをテストするときに、DN がスタティックグループのメンバーではない場合、Sun ONE Web Server はデータベースのベース DN に含まれるすべてのダイナミックグループをチェックします。Sun ONE Web Server は、ベース DN とスコープをユーザーの DN と比較して各 memberURL が一致するかどうかを調べ、次にユーザー DN をベース DN とし、memberURL のフィルタを使用してベース検索を実行することで、このチェックを行います。この処理では、膨大な数の検索が行われることがあります。

ダイナミックグループ作成のガイドライン

新規のダイナミックグループを作成するために管理サーバーのフォームを使用するときには、次のガイドラインを考慮します。

- ダイナミックグループに他のグループを含めることはできません。
- グループの LDAP URL は、次の形式で入力します (ホストとポートの情報は無視されるので、これらのパラメータは指定しません)

  ldap:///<basedn>?<attributes>?<scope>?<(filter)>

  次の表で、必須パラメータについて説明します。

表 3-5　ダイナミックグループの必須パラメータ

パラメータ名	説明
<base_dn>	検索ベースの識別名 (DN)。すべての検索は、LDAP ディレクトリ内のこのポイントより下で実行されます。多くの場合、このパラメータは、「o=mcom.com」のようにディレクトリのサフィックスまたはルートに設定されます。
<attributes>	検索によって返される属性のリスト。複数の属性を指定するときは、属性をコンマで区切ります (たとえば「cn,mail,telephoneNumber」)。属性を指定しない場合、すべての属性が返されます。このパラメータは、グループメンバーシップのチェックでは無視されることに注意してください。
<scope>	検索のスコープ。次のいずれかの値を指定します。 • **base:** URL に指定された識別名 (<base_dn>) に関する情報だけを取得します。 • **one:** URL に指定された識別名 (<base_dn>) の 1 レベル下のエントリに関する情報を取得します。このスコープにはベースエントリは含まれません。 • **sub:** URL に指定された識別名 (<base_dn>) の下のすべてのレベルのエントリに関する情報を取得します。このスコープにはベースエントリが含まれます。 これは必須パラメータです。
<(filter)>	検索の指定スコープ内のエントリに適用される検索フィルタ。管理サーバーのフォームを使用する場合は、この属性を指定する必要があります。( ) で囲む必要があるので注意してください。 これは必須パラメータです。

<attributes>、<scope>、および <(filter)> パラメータは、URL 内の位置で識別されます。どの属性も指定しない場合でも、そのフィールドに疑問符を含める必要があります。

- また、任意で、新規グループに説明を追加できます。
- 組織単位がディレクトリにすでに定義されている場合、「Add New Group To」リストを使用して、新規のグループを配置する場所を指定できます。デフォルトの場所は、ディレクトリのルートポイント (つまり最上位のエントリ) です。
- 必要な情報の入力が終わったら、「Create Group」をクリックして新規グループを追加すると、ただちに「New Group」フォームの画面に戻ります。別の方法として、「Create and Edit Group」をクリックしグループを追加して、追加したグループの「Edit Group」フォームに進む方法もあります。グループの編集については、76 ページの「グループ属性の編集」を参照してください。

ダイナミックグループを作成するには

ディレクトリ内にダイナミックグループエントリを作成するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「New Group」リンクをクリックします。
3. 「Type of Group」ドロップダウンリストから「Dynamic Group」を選択します。
4. 必要な情報を入力し、「OK」をクリックします。

詳細は、オンラインヘルプの「New Group ページ」を参照してください。

グループの管理

管理サーバーを使用して、「Manage Groups」フォームからグループを編集したり、グループのメンバーシップを管理したりできます。この節では、次の内容について説明します。

- グループエントリの検索
- グループ属性の編集
- グループメンバーの追加
- グループメンバーリストへのグループの追加
- グループメンバーリストからのエントリの削除
- 所有者の管理
- See Also の管理
- グループの削除
- グループの名前の変更

グループエントリの検索

グループエントリを編集する前に、エントリを検索して表示する必要があります。

グループエントリを検索するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「Manage Groups」リンクをクリックします。
3. 検索するグループ名を「Find Group」フィールドに入力します。

   検索フィールドに入力できる値は、次のとおりです。

   - 名前。フルネーム、または名前の一部を入力します。検索文字列と完全に一致するすべてのエントリが返されます。これに該当するエントリがない場合は、検索文字列を含むすべてのエントリが検索されます。これにも該当しない場合は、検索文字列と類似したエントリが検索されます。
   - アスタリスク (*) を入力すると、ディレクトリにある現在の組織単位がすべて表示されます。検索フィールドに何も入力しないで検索しても、同じ結果が得られます。
   - 任意の LDAP 検索フィルタ。等号 (=) を含む文字列はすべて、検索フィルタとして認識されます。

   他の方法としては、「Find all groups whose」フィールドのドロップダウンメニューを使用して、検索結果を絞り込む方法もあります。
4. 「Look within」フィールドで、エントリの検索を行う組織単位を選択します。

   デフォルトは、ディレクトリのルートポイント (つまり最上位のエントリ) です。
5. 「Format」フィールドで、「On-Screen」または「Printer」を選択します。
6. 「Find」をクリックします。

   検索条件に一致するグループがすべて表示されます。
7. 検索結果テーブルで、編集したいエントリの名前をクリックします。

Find all groups whose フィールド

「Find all groups whose」フィールドを使用して、カスタム検索フィルタを構築できます。このフィールドを使用して、「Find groups」で返される検索結果を絞り込みます。

Look Within ディレクトリ内のグループエントリをすべて表示するには、テキストフィールドに、アスタリスク (*) を入力するか、または、何も入力せずに検索します。

カスタム検索フィルタを構築する方法については、64 ページの「カスタム検索クエリの構築」を参照してください。

グループ属性の編集

グループエントリを編集するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「Manage Groups」リンクをクリックします。
3. 編集したいグループを特定し、必要な箇所を変更します。

   特定のエントリの検索については、75 ページの「グループエントリの検索」で説明されている内容を参照してください。

注	管理サーバーのユーザーを、root からオペレーティングシステム上の別のユーザーに変更できます。これにより、同じグループに属する複数のユーザーが設定ファイルを編集、管理できるようになります。ただし、UNIX/Linux プラットフォームの場合は、インストーラが任意のグループに設定ファイルの「rw」(読み書き) を許可しますが、Windows プラットフォームの場合は、ユーザーは「Administrators」グループに属している必要があります

詳細は、オンラインヘルプの「Manage Groups ページ」を参照してください。

注	変更したい属性値が「group edit」フォームに表示されていない場合でも、変更は可能です。この場合、Directory Server の `ldapmodify` コマンド行ユーティリティが使用できる場合は、これを使用します。

グループメンバーの追加

グループにメンバーを追加するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「Manage Groups」リンクをクリックします。
3. 75 ページの「グループエントリの検索」で説明されているように、管理したいグループを特定し、「Group Members」の下の「Edit」ボタンをクリックします。

   Sun ONE Web Server に、エントリ検索のための新しいフォームが表示されます。リストにユーザーエントリを追加する場合、「Users 」が「Find」のドロップダウンメニューに表示されていることを確認します。グループにグループエントリを追加する場合、必ず「Group」が表示されていることを確認します。
4. 一番右側のテキストフィールドに、検索文字列を入力します。次のオプションのうち、いずれかを入力します。

- 名前。フルネーム、または名前の一部を入力します。検索文字列と名前が一致するすべてのエントリが返されます。これに該当するエントリがない場合は、検索文字列を含むすべてのエントリが検索されます。これにも該当しない場合は、検索文字列と類似したエントリが検索されます。
- ユーザーエントリを検索する場合は、ユーザー ID。
- 電話番号。電話番号の一部だけを入力すると、最後の部分が検索番号に一致する電話番号を含むエントリがすべて返されます。
- 電子メールアドレス。アットマーク (@) 記号を含む検索文字列は、すべて、電子メールアドレスとして認識されます。完全に一致するエントリがない場合は、検索文字列で始まる電子メールアドレスがすべて検索されます。
- 現在ディレクトリ内にあるエントリまたはグループをすべて表示するには、テキストフィールドに、アスタリスク (*) を入力するか、または、何も入力せずに検索します。
- 任意の LDAP 検索フィルタ。等号 (=) を含む文字列はすべて、検索フィルタとして認識されます。

5. 「Find and Add」をクリックして、一致するすべてのエントリを検索し、グループにこのエントリを追加します。

   グループに追加する必要のないエントリが返された場合は、「Remove from list?」列内のボックスをクリックします。また、削除したいエントリに一致する検索フィルタを作成して、「Find and Remove」をクリックすることもできます。

6. グループメンバーのリストが完成したら、「Save Changes」をクリックします。

   現在表示されているエントリがグループのメンバーとなります。

グループメンバーの追加の詳細は、オンラインヘルプの「Edit Members ページ」を参照してください。

グループメンバーリストへのグループの追加

グループのメンバーリストには、個々のメンバーではなく、グループを追加することができます。グループを追加すると、追加されたグループに属するユーザーは、追加先のグループのメンバーになります。たとえば、Neil Armstrong が「Engineering Managers」グループのメンバーであり、この「Engineering Managers」グループを「Engineering Personnel」グループのメンバーにする場合、Neil Armstrong は、「Engineering Personnel」グループのメンバーにもなります。

グループを別のグループのメンバーリストへ追加するには、ユーザーエントリと同様に、グループを追加します。詳細は、76 ページの「グループメンバーの追加」を参照してください。

グループメンバーリストからのエントリの削除

グループメンバーリストからエントリを削除するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「Manage Groups」リンクをクリックして、75 ページの「グループエントリの検索」で説明されているように、管理したいグループを特定し、「Group Members」の下の「Edit」ボタンをクリックします。
3. リストから削除したい各メンバーについて、「Remove from list?」列の下の、対応するボックスをクリックします。

   ほかに、削除したいエントリを検索するフィルタを作成して、「Find and Remove」をクリックする方法もあります。検索フィルタの作成の詳細は、76 ページの「グループメンバーの追加」を参照してください。
4. 「Save Changes」をクリックします。エントリが、グループメンバーリストから削除されます。

所有者の管理

グループの所有者リストは、グループメンバーリストと同様の方法で管理します。詳細についての参照先は、次の表に示します。

表 3-6 追加情報

タスク	参照先
グループに所有者を追加する	76 ページの「グループメンバーの追加」
所有者リストにグループを追加する	77 ページの「グループメンバーリストへのグループの追加」
所有者リストからエントリを削除する	78 ページの「グループメンバーリストからのエントリの削除」

See Also の管理

「See Also」(関連項目) は、現在のグループに関連のある、他のディレクトリのエントリへの参照です。「See Also」を使用して、現在のグループと関連のあるユーザーや他のグループのエントリを簡単に見つけることができます。

「See Also」は、グループメンバーリストと同様の方法で管理します。詳細についての参照先は、次の表に示します。

表 3-7　追加情報

タスク	参照先
「See Also」にユーザーを追加する	76 ページの「グループメンバーの追加」
「See Also」にグループを追加する	77 ページの「グループメンバーリストへのグループの追加」
「See Also」からエントリを削除する	78 ページの「グループメンバーリストからのエントリの削除」

グループの削除

グループを削除するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「Manage Groups」リンクをクリックして、75 ページの「グループエントリの検索」で説明されているように、管理したいグループを特定し、「Delete Group」をクリックします。

注　管理サーバーは、グループエントリだけを削除します。削除したグループに属する個々のメンバーは削除されません。

グループの名前の変更

グループの名前を変更するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「Manage Groups」リンクをクリックして、75 ページの「グループエントリの検索」で説明されているように、管理したいグループを特定します。
3. 「Rename Group」ボタンをクリックして、表示されたダイアログボックスに新しいグループの名前を入力します。

グループエントリ名を変更する場合、変更できるのはグループの名前だけです。グループ名の変更機能 (Rename Group) を使って、エントリを 1 つの組織単位から別の組織単位へ移動することはできません。たとえば、ある企業には次のような組織があるとします。

- Marketing および Product Management という組織単位
- Marketing という組織単位の下に Online Sales というグループ

この例では、Online Sales というグループ名を Internet Investments に変更することはできますが、Marketing という組織単位の下の Online Sales を、Product Management という組織単位の下の Online Sales にするようエントリの名前を変えることはできません。

組織単位の作成

組織単位には、複数のグループを含めることができ、それらは通常、部、課などの業務グループを表します。DN は、複数の組織単位に存在させることができます。

組織単位を作成するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「New Organizational Unit」リンクをクリックし、必要な情報を入力します。

詳細は、オンラインヘルプの「New Organizational Unit page ページ」を参照してください。

次の項目は、主にディレクトリ管理者を対象としています。

- 新規の組織単位は、`organizationalUnit` オブジェクトクラスを使用して作成します。
- 新規の組織単位の識別名のフォームは次のとおりです。

```
ou=new organization, ou=parent organization, ...,o=base
organization, c=country
```

たとえば、Accounting という新規の組織を、組織単位 West Coast 内に作成する場合、ベース DN が o=Ace Industry、c=US とすると、新規の組織単位の DN は、次のようになります。

```
ou=Accounting, ou=West Coast, o=Ace Industry, c=US
```

組織単位の管理

組織単位の編集や管理には、「Organizational Unit Edit」フォームを使用します。この節では、次のタスクについて説明します。

- 組織単位の検索
- 組織単位の属性の編集
- 組織単位名の変更
- 組織単位の削除

組織単位の検索

組織単位を検索するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Users & Groups」タブを選択します。
2. 「Manage Organizational Units」リンクをクリックします。
3. 検索する組織単位の名前を「Find Organizational Units」フィールドに入力します。検索フィールドに入力できる値は、次のとおりです。
   - 名前。フルネーム、または名前の一部を入力します。検索文字列と完全に一致するすべてのエントリが返されます。これに該当するエントリがない場合は、検索文字列を含むすべてのエントリが検索されます。これにも該当しない場合は、検索文字列と類似したエントリが検索されます。
   - アスタリスク (*) を入力すると、ディレクトリにある現在の組織単位がすべて表示されます。検索フィールドに何も入力しないで検索しても、同じ結果が得られます。
   - 任意の LDAP 検索フィルタ。等号 (=) を含む文字列はすべて、検索フィルタとして認識されます。

   他の方法としては、「Find all units whose」フィールドのドロップダウンメニューを使用して、検索結果を絞り込む方法もあります。
4. 「Look within」フィールドで、エントリの検索を行う組織単位を選択します。

   デフォルトでは、ディレクトリのルートポイントとなります。
5. 「Format」フィールドで、「On-Screen」または「Printer」を選択します。
6. 「Find」をクリックします。

   検索条件に一致する組織単位がすべて表示されます。
7. 検索結果テーブルで、編集したい組織単位の名前をクリックします。

Find all units whose フィールド

「Find all units whose」フィールドを使用して、カスタム検索フィルタを構築できます。このフィールドを使用して、「Find organizational unit」で返される検索結果を絞り込みます。

Look Within ディレクトリ内の組織単位エントリをすべて表示するには、テキストフィールドに、アスタリスク (*) を入力するか、または、何も入力せずに検索します。

カスタム検索フィルタを構築する方法については、64 ページの「カスタム検索クエリの構築」を参照してください。

組織単位の属性の編集

組織単位のエントリを変更するには、管理サーバーにアクセスし、次の手順を実行します。

1. 81 ページの「組織単位の検索」で説明されているように、編集したい組織単位を特定します。

   組織単位編集フォームが表示されます。

2. 必要に応じてフィールドの変更を行い、「Save Changes」をクリックします。

   変更は、すぐに有効となります。

注	変更したい属性値が「organizational unit edit」フォームに表示されていない場合でも、変更は可能です。この場合、Directory Server の `ldapmodify` コマンド行ユーティリティが使用できる場合は、これを使用します。

組織単位名の変更

組織単位エントリの名前を変更するには、管理サーバーにアクセスし、次の手順を実行します。

1. 名前を変更したい組織単位の下のディレクトリには、他のエントリが何も入っていないことを確認します。
2. 81 ページの「組織単位の検索」で説明されているように、編集したい組織単位を特定します。
3. 「Rename」ボタンをクリックします。
4. 表示されたダイアログボックスに新しい組織単位名を入力します。

注	組織単位エントリの名前を変更する場合、変更できるのは組織単位の名前だけです。名前の変更機能を使って、エントリを 1 つの組織単位から別の組織単位へ移動することはできません。詳細は、82 ページの「組織単位名の変更」を参照してください。

組織単位の削除

組織単位エントリを削除するには、管理サーバーにアクセスし、次の手順を実行します。

1. 削除したい組織単位の下のディレクトリには、他のエントリが何も入っていないことを確認します。
2. 81 ページの「組織単位の検索」で説明されているように、編集したい組織単位を特定します。
3. 「Delete」ボタンをクリックします。
4. 表示される確認ボックスで、「OK」をクリックします。

   組織単位がすぐに削除されます。

第 4 章

Web コンテナと Web アプリケーションの J2EE ベースのセキュリティ

この章では、Sun ONE Web Server 6.1 Web コンテナおよび Web アプリケーションの J2EE ベースセキュリティの基本機能について説明します。まず、Web サーバーでサポートされる 2 つの主要認証、承認モデルである、ACL (アクセス制御リスト) ベースのセキュリティモデルと、J2EE/ サーブレットベースのセキュリティモデルについて説明します。また、両方のセキュリティシステムを活用した Java Web アプリケーションを配備できる Sun ONE Web Server 6.1 の新機能についても説明します。

この章の残りのページでは、J2EE/ サーブレットの設定に関する問題について説明します。関連するセキュリティの問題については、次の各章で説明します。

- 証明書と公開鍵暗号化 : 第 6 章「証明書と鍵の使用」
- ACL ベースのセキュリティ : 第 9 章「サーバーへのアクセス制御」

この章では、次の項目について説明します。

- Sun ONE Web Server のセキュリティについて
- ACL ベースのアクセス制御の概要
- J2EE/ サーブレットベースのアクセス制御の概要
- レルムベースのセキュリティ
- レルムの設定方法
- デフォルトレルムの指定
- プログラムによるセキュリティの使用
- どのような場合に J2EE/ サーブレット認証モデルを使用するか

Sun ONE Web Server のセキュリティについて

Web サーバー上のリソースを、認証、承認、アクセス制御などのセキュリティサービスおよびメカニズムで保護することができます。

認証とは、同一性 (ID) を確認するためのプロセスのことです。承認とは、アクセスが制限されたリソースへのアクセス権を ID に与え、アクセス制御メカニズムはその制限を強化します。認証と承認は、多数のセキュリティモデルおよびサービスによって強化することができます。

Sun ONE Web Server 6.1 は、HTTP エンジンによって提供される ACL ベースのセキュリティモデルと、Web コンテナによって提供される J2EE Servlet バージョン 2.3 仕様に基づくセキュリティモデルの 2 モデルをサポートしています。

Sun ONE Web Server 6.1 プロセスのライフタイムでは、両方のモデルが共存します。それぞれのモデルが、クライアント認証と承認の両方のセキュリティ サービスをサポートします。

Sun ONE Web Server 6.1 の Web コンテナは、JAAS (Java Authentication and Authorization Service) ベースのレルムメカニズムによりクライアント認証を提供し、J2EE ロールベースのメカニズムにより承認を提供します。Sun ONE Web Server 6.1 が提供するレルムの一つに native レルムがあります。このレルムは、2 つのセキュリティモデルを結びつけています。

Sun ONE Web Server 6.1 は、宣言によるセキュリティとプログラムによるセキュリティの両方をサポートします。

Sun ONE Web Server 6.1 は、J2EE プラットフォームの機能を利用して、アプリケーションコンポーネントの開発およびアセンブル実行者と、操作環境でのアプリケーション設定者の間の宣言的契約を定義します。アプリケーションセキュリティの面では、アプリケーションの設定時にセキュリティ要件を満たせるように、アプリケーションプロバイダはアプリケーションのセキュリティ要件を宣言する必要があります。アプリケーションで使用される宣言によるセキュリティのメカニズムは、配備記述子というドキュメントに宣言構文として記述されます。アプリケーションの配備担当者はコンテナ固有のツールを使用して、配備記述子に定義されているアプリケーション要件を、J2EE コンテナによって実装されるセキュリティメカニズムにマッピングします。Sun ONE Web Server 6.1 の Web アプリケーション用配備記述子ファイルは、`web.xml` および `sun-web.xml` です。

プログラムによるセキュリティは、セキュリティ認識アプリケーションによるセキュリティに関する決定を参照します。プログラムによるセキュリティは、アプリケーションのセキュリティモデルを設計する上で、宣言によるセキュリティだけでは不十分な場合に便利です。たとえば、時刻、呼び出しのパラメータ、および Web コンポーネントの内部状態に基づいて認証を決定するアプリケーションがあります。また、データベースに格納されているユーザー情報に基づいてアクセスを制限するアプリケーションもあります。

この章の残りのページでは、Sun ONE Web Server 6.1 がサポートする次の認証および承認の主要概念について説明します。

- ACL ベースのアクセス制御については、「ACL ベースのアクセス制御の概要」で説明します。
- J2EE ベースのアクセス制御については、「J2EE/ サーブレットベースのアクセス制御の概要」で説明します。
- native レルムのサポートについては、「native レルム」で説明します。
- プログラムによるセキュリティについては、「プログラムによるセキュリティの使用」で説明します。

ACL ベースのアクセス制御の概要

ACL ベースのアクセス制御については、第 9 章「サーバーへのアクセス制御」で詳しく説明します。この節では、主要概念についてその概要を簡単に説明します。

Sun ONE Web Server 6.1 では、ローカルに保存されている ACL (アクセス制御リスト) により認証と承認を行います。ACL では、リソースに対してユーザーがどのようなアクセス権を持つかを規定しています。たとえば、ACL 内のエントリは、John というユーザーに、特定のフォルダ `misc` に対する読み取りアクセス権を与えることができます。

```
acl "path=/export/user/990628.1/docs/misc/";

  authenticate (user,group) {

      database = "default";

      method = "basic";

  };

  deny (all) (user="anyone");

  allow (read) (user = "John");
```

Sun ONE Web Server 6.1 内のコア ACL は、基本、SSL、ダイジェスト、という 3 種類の認証をサポートします。

基本認証は、クリアテキストとして渡される、ユーザー名とパスワードのリストを使用します。SSL 認証では、ブラウザがユーザー証明書を持つ必要があります。この証明書には、ユーザーの公開鍵と、名前や電子メールアドレスなど、その他のユーザー情報が含まれます。ダイジェスト認証は、暗号化方式を使用して、ユーザーの証明情報を暗号化します。

ACL ベースのアクセス制御モデルの主な機能は次のとおりです。

- ACL ベースの認証と承認は、次の設定ファイルを使用します。
  - *server-install*/httpacl/*.acl files
  - *server-install*/userdb/dbswitch.conf
  - *server-install*/*server-instance*/config/server.xml
- 認証データベースは、dbswitch.conf ファイルに設定されている auth-db モジュールによって提供されます。
- ACL が設定されている場合、*server-install*/httpacl/*.acl ファイル内のアクセス制御規則のセットによって認証と承認が行われます。適用される認証規則は、要求を処理する仮想サーバーに対応する ACL ファイルに定義されています (server.xml 内の適切な VS エントリとして設定されています)。『Sun ONE Web Server 6.1 Administrator's Configuration File Reference』で ACLFILE 要素と、VS 要素の aclids プロパティを参照してください。通常、これらのファイルは /httpacl/ ディレクトリに配置されていますが、server.xml の設定を変更した場合はその限りではありません。

さらに、Sun ONE Web Server 6.1 の SSL エンジンは、SSL 処理の負荷を低減できるように外部の暗号化ハードウェアをサポートし、改ざん性に優れた鍵のストレージを提供します。

アクセス制御、および外部暗号化ハードウェアの使用の詳細は、第 9 章「サーバーへのアクセス制御」を参照してください。

J2EE/ サーブレットベースのアクセス制御の概要

J2EE/ サーブレットベースのアクセス制御については、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』で詳しく説明しています。次の項では、主要概念についてその概念を簡単に説明します。

Sun ONE Web Server 6.1 は、ACL ベースの認証のほかに、J2EE 1.3 Specification で定義されているセキュリティモデルを利用し、安全な Java ベースの Web アプリケーションの開発および使用に役立ついくつかの機能を備えています。

J2EE ベースの一般的な Web アプリケーションは、次の項目から構成されます。一部またはすべての項目へのアクセスは制限することが可能です。

- サーブレット
- JSP (JavaServer Pages) コンポーネント
- HTML ドキュメント
- イメージファイルや圧縮アーカイブなどの、その他のリソース

J2EE/ サーブレットベースのアクセス制御インフラストラクチャは、セキュリティレルムを使用します。ユーザーが Web ブラウザ経由でアプリケーションのアクセス保護対象セクションにアクセスしようとすると、Web コンテナはユーザーに証明情報の入力を要求し、そのアプリケーション用のセキュリティサービスで現在有効になっているレルムにその情報を渡して検証します。

J2EE/ サーブレットベースのアクセス制御モデルの主な機能は次のとおりです。

- J2EE/ サーブレットベースの認証は、次の設定ファイルを使用します。
  - Web アプリケーション配備記述子ファイル `web.xml` および `sun-web.xml`
  - *server-install*`/`*server-instance*`/config/server.xml`
- 認証は、`server.xml` ファイル内の `AUTHREALM` エントリに設定される Java セキュリティレルムによって実行されます。
- アクセス制御規則が設定されている場合、承認は配備記述子ファイル `web.xml` 内のアクセス制御規則によって実行されます。

次の節では、セキュリティレルムの概念について簡単に説明します。J2EE セキュリティモデルとレルムベースの認証の詳細な解説については、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』を参照してください。

レルムベースのセキュリティ

J2EE ベースのセキュリティモデルは、ユーザーの識別および認証を行うセキュリティレルムを提供します。ユーザー情報は、基礎となるセキュリティレルムから取得されます。レルムベースのセキュリティは、次の 2 つの要素から構成されます。

- レルムベースのユーザー認証 : 基本となるレルムを使用してユーザーを検証する
- ロールベースの認証 : リソースへのアクセスが許可されているか、あるいはアクセスが制限されているロールをユーザーに割り当てる

レルムベースのユーザー認証

認証プロセスは、セキュリティドメインとも呼ばれる、基本となるレルムを使用してユーザーを検証します。レルムは、ユーザーのセット、オプションのグループマッピング、および認証要求を評価する認証ロジックから構成されます。設定されているレルムによって認証要求が検証され、セキュリティコンテキストが確立されると、`run-as` 条件によって否定されない限り、以後のすべての認証決定にこの ID が適用されます。

サーバーインスタンスに設定できるレルムの数に制限はありません。設定情報は、`server.xml` ファイルの `AUTHREALM` 要素に記録されます。

Sun ONE Web Server では、認証サービスは、プラグイン可能なセキュリティドメインを提供する JAAS を使用して行われます。Sun ONE Web Server 6.1 の Java 認証レルムは、Sun ONE Application Server 7.0 のレルムと互換性があります。

Sun ONE Web Server 6.1 には、次のレルムが用意されています。

- LDAP レルム
- file レルム
- solaris レルム
- 証明書レルム
- カスタムレルム
- native レルム

LDAP レルム

`ldap` レルムを使用することで、LDAP データベースからユーザーのセキュリティ情報を取得できます。LDAP ディレクトリサービスは、属性と、それぞれの一意の ID の集合です。`ldap` レルムは、運用システムへの配備に適しています。

`ldap` レルムに対してユーザーを認証するには、LDAP ディレクトリ内に必要なユーザーを作成しておく必要があります。ユーザーの作成は管理サーバーの「Users & Groups」タブ、または LDAP ディレクトリ製品のユーザー管理コンソールから行えます。詳細は、58 ページの「LDAP ベース認証データベースの新規ユーザーの作成」を参照してください。

file レルム

`file` レルムは、Sun ONE Web Server のインストール時のデフォルトレルムです。これは、設定が簡単で、開発者には便利なレルムです。

`file` レルムは、テキストファイルに保存されているユーザーデータに対してユーザーを認証します。file レルムがサポートする認証データベースは、次のとおりです。

- 鍵ファイルスタイルのデータベース
- htaccess スタイルのデータベース
- ダイジェストスタイルのデータベース

ファイルベースの各種認証データベースについて詳細は、<add> を参照してください。

file レルムが使用するユーザー情報ファイルは、最初は空なので、file レルムを使用し始める前にユーザーを追加する必要があります。この方法については、61 ページの「ファイルベース認証データベースの新規ユーザーの作成」を参照してください。

solaris レルム

solaris レルムでは、認証に Solaris のユーザー名とパスワードのデータを使用できます。このレルムをサポートしているのは、Solaris 9 のみです。このレルムは Solaris 9 オペレーティング環境のデータベースを使用するため、データベースを別に設定する手順は必要ありません。

証明書レルム

証明書レルムは、SSL 認証をサポートします。証明書レルムは、Sun ONE Web Server のセキュリティコンテキストにユーザー ID を設定し、その ID にクライアント証明書からのユーザーデータを移行します。次に、J2EE コンテナが、証明書に含まれる各ユーザーの DN に基づいて認証処理を行います。このレルムは、X.509 証明書による SSL または TLS クライアント認証を使用してユーザーを認証します。

サーバーとクライアントの証明書を設定する方法については、第 6 章「証明書と鍵の使用」を参照してください。

カスタムレルム

プラグイン可能な JAAS ログインモジュールと、レルムの実装を使用して、特定の用途に合わせて Oracle などのその他のデータベース用にレルムを作成することができます。クライアント側の JAAS ログインモジュールは、Sun ONE Web Server での使用に適していないことに注意してください。

Sun ONE Web Server 6.1 のサンプルレルムをテンプレートとして使用できます。

native レルム

native レルムは、ACL ベースのコア認証モデルと、J2EE/ サーブレット認証モデルの間を結びつける特殊なレルムです。Java Web アプリケーションで native レルムを使用することで、Java Web コンテナを使用する代わりに ACL サブシステムを使用して認証を行い、この ID を Java Web アプリケーションで使用できるようにすることができます。

認証処理が開始されると、native レルムはこの認証をコア認証サブシステムに委託します。ユーザー側では、これはたとえば、設定されている LDAP サーバーに LDAP レルムが認証を委託するのと同じように見えます。native レルムで処理されるグループメンバーシップクエリも、コア認証サブシステムに委託されます。Java Web モジュールおよび開発者側から見ると、native レルムは Web モジュールで利用できるその他の Java レルムと変わりません。

native レルムは認証をコアに委託するため、追加設定が必要になります。詳細は、「native レルムの設定」を参照してください。

『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』には、J2EE セキュリティレルムと、セキュリティレルムの設定パラメータに関する詳細な情報が記載されています。

ロールベースの認証

Java Servlet 2.3 Specification には、アクセス制御規則を作成して、さまざまな J2EE アプリケーションリソースへのアクセスを制御する方法が定義されています。

ロールと制限対象領域のマッピング

J2EE アクセス制御は、ロールに基づいて行われます。特定の HTML ページ、サーブレット、JSP などへのアクセスを制限するには、次の項目を定義する必要があります。

- 制限対象領域 : Web モジュール記述子 (`web.xml`) に記述
- ロール : `web.xml` 内でどのロールに各制限対象領域へのアクセス権を与えるかを記述
- ユーザーおよびグループとロールのマッピング : `sun-web.xml` 内でどのユーザーが、どの制限対象領域へのアクセスを承認されるかを決定

ユーザーには複数のロールを割り当てることができます。検証時に、少なくとも 1 つのロールが割り当てられていれば、そのロールに対応する領域へのアクセスが許可されます。

`webapps/security` ディレクトリに保存されている、Sun ONE Web Server 6.1 での各種アクセス制限のサンプルをテンプレートとして使用できます。サーブレットロールベースのセキュリティの詳細は、Servlet 2.3 仕様を参照してください。

ロールによるアクセス制御の定義

J2EE アプリケーションロールは抽象的で、特定のアプリケーションに適用されます。実際の環境で、アクセスが認証ユーザーに限定されたアプリケーションを実行するには、`sun-web.xml` 記述子でユーザー名とロールをマッピングする必要があります。これは、次のいずれか、または両方の方法で行います。

主体マッピング - `sun-web.xml` 内で、1 つまたは複数のユーザー名をロールに直接マッピングします。この方法はテストには便利ですが、各ロールにマッピングできるユーザー数が限定されています。

グループマッピング - `sun-web.xml` 内で、1 つまたは複数のグループを指定して、1 つまたは複数のユーザー名を間接的にロールにマッピングします。(たとえば、engineers、managers、staff などのグループ名を指定します。) 指定したグループに所属する認証ユーザーに、アプリケーションロールが割り当てられます。指定したグループにどのユーザーが所属しているかは、有効なレルム実装 (または参照されるデータベース) によって決定されることに注意してください。

主体 (ユーザー) が、たとえば サーブレットや JSP などの特定の Web リソースを要求すると、Web コンテナがセキュリティ制約、または配備記述子ファイル内のリソースに関連付けられているアクセス権を確認し、その主体がリソースへのアクセスを承認されているかどうかを決定します。

ロールマッピングのエントリは、モジュール記述子内でロールとユーザーまたはグループをマッピングします。その例を次に示します。

```
<sun-web-app>
    <security-role-mapping>
    <role-name>manager</role-name>
    <principal-name>jsmith</principal-name>
    <group-name>divmanagers</group-name>
</sun-web-app>
```

配備記述子ファイルの詳細は、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』を参照してください。

レルムの設定方法

レルムの設定は、次のいずれかの方法で行います。

- 管理インタフェースの使用
- server.xml ファイルの編集

管理インタフェースの使用

管理インタフェースを使用してレルムを設定するには、次の手順を実行します。

1. 管理サーバーインタフェースから対象サーバーインスタンスにアクセスし、「Java」タブをクリックします。
2. 「Security Realms」リンクをクリックします。

   デフォルトでは、次のレルムが用意されています。

   - file
   - native
   - ldap
3. レルムを追加するときは、「New」ボタンをクリックします。レルムを削除するときは、レルム名の隣のチェックボックスにチェックマークを付け、「OK」をクリックします。レルムを編集するときは、レルム名をクリックします。
4. レルムを追加または編集するときは、レルム名、クラス名、プロパティ、ユーザー (`file` レルムのみ) を入力し、「OK」ボタンをクリックします。
5. 「OK」をクリックします。

server.xml ファイルの編集

レルムは、実際には `server.xml` ファイルの `SECURITY` 要素に設定されます。`SECURITY` 要素は、次のように設定されます。

```
<SECURITY defaultrealm="file" anonymousrole="ANYONE"
      audit="false">
   <AUTHREALM name="file"
         classname="com.iplanet.ias.security.auth.realm.file.FileRe
alm">
      <property name="file" value="instance_dir/config/keyfile"/>
```

```
      <property name="jaas-context" value="fileRealm"/>
   </AUTHREALM>
   ...
</SECURITY>
```

defaultrealm 属性は、サーバーがデフォルトで使用するレルムを指定します。デフォルトレルムは、それぞれの web.xml ファイルに有効なレルムが指定されていないすべての Web アプリケーションで使用されます。デフォルトレルムには、設定されているいずれかの AUTHREALM 名を指定する必要があります。デフォルトは file レルムです。

audit フラグは、監査情報をログに記録するかどうかを決定します。true に設定すると、サーバーは認証および承認イベントのすべての監査メッセージをログに記録します。

レルムの設定を変更した場合、その変更を適用するにはサーバーを再起動する必要があります。

server.xml ファイルの詳細は、『Sun ONE Web Server 6.1 Administrator's Configuration File Reference』を参照してください。

native レルムの設定

すべてのレルムと同様に、native レルムも設定できます。設定には、server.xml ファイルの SECURITY 要素に含まれる AUTHREALM 要素を使用します。その例を次に示します。

```
<AUTHREALM name="native"
classname="com.sun.enterprise.security.auth.realm.webcore.NativeRea
lm">

   <PROPERTY name="auth-db" value="mykeyfile">

   <PROPERTY name="jaas-context" value="nativeRealm"/>

</AUTHREALM>
```

auth-db プロパティは、この native レルムがすべての認証要求の処理を委託するコア認証データベースを指定します。この例では、認証データベースの名前は「mykeyfile」です。このプロパティはオプション (省略可能) です。指定しない場合、コア認証エンジンはデフォルトの auth-db を使用して、native レルムからのすべての要求を処理します。ほとんどのレルムでは、jaas-context プロパティは JAAS ログインコンテキスト (login.conf に定義されています) を指定します。

native レルムでは、その他の設定は必要ありません。ただし、要求がコア認証データベースに委託されるため、その認証データベースを適切に設定しておく必要があります。この節の残りのページでは、コア認証データベースの設定例を紹介します。

コア (native) 認証データベースを設定するには、auth-db 名とデータベース名をマッピングする USERDB 要素が、server.xml ファイルの VS 要素に含まれている必要があります。その例を次に示します。

```
<VS id="https-plaza.com"  ....

....

   <USERDB id="mykeyfile" database="myalt"/>

....

</VS>
```

auth-db プロパティを指定しない場合 (この場合は「default」が使用されます)、USERDB エントリの一部のデータベース名には「id="default"」がマッピングされます。マッピングが指定されていない場合は、default にマッピングされます。

次に、*install-root*/userdb/dbswitch.conf ファイルには、myalt データベースが設定されている必要があります。次の例は、ファイルベースの認証データベースとして myalt を定義しています。

```
directory myalt file

myalt:syntax keyfile

myalt:keyfile /local/ws61/https-plaza.com/config/keyfile
```

これは、native レルムに固有の設定ではありません。native レルムでは、処理の委託先認証データベースとして、任意の認証ディレクトリの有効な設定を使用できます。つまり、ネイティブ LDAP 認証データベースや、カスタムネイティブ認証データベースにさえも処理を委託するように native レルムを設定することができます。

注

Sun ONE Web Server 6.1 の Web アプリケーションには、認証エンジンとして LDAP を使用する次の 2 つの異なるメカニズムがあります。

- Java LDAP レルムを使用する
- ネイティブ LDAP 認証データベースに処理を委託するように設定された Java Native レルムを使用する

デフォルトレルムの指定

デフォルトレルムは、それぞれの web.xml 配備記述子ファイルに有効なレルムが指定されていない、すべての Web アプリケーションの認証イベントの処理に使用されます。有効な認証レルムをサーバーインスタンスに指定するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「Java Security」リンクをクリックします。
3. 次の情報を設定します。
   - Default Realm: このサーバーインスタンスの、有効な認証レルム (name 属性 AUTHREALM) を指定する
   - Anonymous Role: (省略可) デフォルトの名前、つまり匿名ロールとして使用
   - Audit Enabled: (省略可) 「true」の場合、アクセスの追加情報がログに記録され、監査情報が提供される。監査情報には、次の情報が含まれる
     - 認証の成功と失敗のイベント
     - サーブレットアクセスの認可と拒否
   - Log Level: (省略可) エラーログに記録されるメッセージの種類を制御
4. 「OK」をクリックします。

プログラムによるセキュリティの使用

Sun ONE Web Server 6.1 は、レルムが提供するコンテナ管理による認証のほかに、プログラム化されたログインインタフェースによりアクセスを可能にする、管理型認証をサポートしています。このインタフェースは、レルムのインフラストラクチャには適さないカスタム認証モデルをサポートしています。プログラムによるログインは、それ自体の認証コンテキストを直接確立するために J2EE アプリケーションでも使用されます。ただし、アプリケーションの移植性は低下し、保守も困難になるため、このような手法はお勧めできません。

アプリケーションでプログラムによるログインメカニズムを呼び出すには、ProgrammaticLoginPermission アクセス権が必要です。これは J2EE の標準メカニズムではないため、このアクセス権は配備されるアプリケーションにデフォルトでは与えられません。

Sun ONE Web Server 6.1 は、Security Manager をサポートしています。製品インストール時のデフォルト設定では、Security Manager は無効に設定されています。サーバーインスタンスで Java Security Manager を有効にしたときは、プログラムによるログインを使用するすべての Web アプリケーションにこのアクセス権を与える必要があります。

必要なアクセス権をアプリケーションに与えるには、`server.policy` ファイルを編集する必要があります。

標準の Java ポリシーエントリを `server.xml` ファイルに指定することで、ポリシーのサポートを有効にすることができます。

```
<JVMOPTIONS>-Djava.security.manager</JVMOPTIONS>
```

```
<JVMOPTIONS>-Djava.security.policy=install-root/https-servername/config/server.policy</JVMOPTIONS>
```

`server.policy` ファイルの詳細は、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』を参照してください。

どのような場合に J2EE/ サーブレット認証モデルを使用するか

この節では、どのような場合に J2EE/ サーブレットベースの認証モデルを使用するかについて説明します。

J2EE/ サーブレット認証モデルは、次のような場合に使用します。

- 一般的に、J2EE/ サーブレットベースのほとんどの新規 Web アプリケーション
- 変更したくない既存の `.war` ファイル
- 現在または将来、J2EE/ サーブレットとの完全な互換性が重要になる Web アプリケーションを作成する場合
- フォームベースの認証を使用する場合 (ACL がフォームベースの認証をサポートしないため)

ACL ベースのインフラストラクチャを使用する場合でも、Java レルムである native レルムを使用して、ユーザー ID をサーブレットで利用できるようにすることができます。

第 5 章

管理サーバーの設定

管理サーバーは、「Preferences」タブと「Global Settings」タブのページを使用して設定できます。サーバーの設定に必要な CGI プログラムを実行できるように、ブラウザの cookie を有効にする必要があることに留意してください。

この章には、次の節が記述されています。

- 管理サーバーのシャットダウン
- 待機ソケット設定の編集
- ユーザーアカウントの変更 (UNIX/Linux)
- スーパーユーザー設定の変更
- 複数の管理者の許可
- ログファイルオプションの指定
- ディレクトリサービスの設定
- サーバーへのアクセスの制限

管理サーバーのシャットダウン

サーバーがインストールされると、サーバーは常時稼働して HTTP 要求を待機し、受け取ります。しかし、サーバーを停止し再起動する必要のある場合もあります。たとえば、JDK (Java Development Kit) や Directory Server をインストールした直後、または待機ソケットの設定を変更した場合などです。

次の方法のいずれかを使ってサーバーを停止できます。

- 管理サーバーにアクセスし、「Preferences」タブを選び、「Shut Down」リンクを選択し、「Shut down the administration server!」ボタンをクリックします。

  詳細は、オンラインヘルプの「Shut Down ページ」を参照してください。

- 「コントロールパネル」の「サービス」ウィンドウを使用します (Windows の場合)。
- `stop` を使用して、サーバーを完全にシャットダウンします。サービスは、サーバーが再起動するまで中断されます。

サーバーをシャットダウンしたあと、シャットダウンプロセスが完了し、ステータスが「Off」に変更されるまでに数秒かかる場合があります。

待機ソケット設定の編集

サーバーで要求を処理するには、サーバーは待機ソケットから要求を受け取り、適切な仮想サーバーにその要求を送信する必要があります。Sun ONE Web Server をインストールすると、ls1 という待機ソケットが自動的に作成されます。この待機ソケットは、0.0.0.0 の IP アドレス と、インストール時に HTTP サーバーのポート番号として指定したポート番号 (デフォルトでは 8888) を使用します。デフォルトの待機ソケットは削除できません。

サーバーの待機ソケットの設定は、管理サーバーの「Listen Sockets Table」を使用して編集できます。この表にアクセスするには、以下の手順を実行します。

1. 管理サーバーにアクセスして、「Preferences」タブをクリックします。
2. 「Edit Listen Sockets」リンクをクリックします。
3. 変更を行い、「OK」をクリックします。

詳細については、第 13 章「仮想サーバーの使用」およびオンラインヘルプの「Edit Listen Sockets ページ」を参照してください。

ユーザーアカウントの変更 (UNIX/Linux)

「Server Settings」ページでは、UNIX や Linux マシン上の Web サーバーのユーザーアカウントを変更できます。サーバーの処理はすべて、このユーザーアカウントとして実行されます。

1024 を超えるポート番号が指定されており、`root` ユーザー以外のユーザーアカウントで稼動している場合は、サーバーユーザーを指定する必要はありません (この場合、サーバーを起動するために `root` でログオンする必要はありません)。ここでユーザーアカウントを指定しない場合、サーバーは、起動時のユーザーアカウントで稼動します。サーバーを起動するときは、必ず適正なユーザーアカウントを使用する必要があります。

注 システムに新規ユーザーを作成する方法が不明の場合、システム管理者に連絡するか、または、使用しているシステムのマニュアルを参照してください。

サーバーを root で起動した場合でも、常時 root でサーバーを稼動すべきではありません。サーバーに、システムリソースへのアクセスを一部制限させたり、非特権ユーザーとして稼動させたりしたい場合もあります。サーバーユーザーとして入力したユーザー名は、すでに、通常の UNIX/Linux ユーザーアカウントとして存在しているはずです。サーバーの起動後は、このユーザー名で稼動します。

ユーザーアカウントを新規に作成したくない場合は、ユーザーとして nobody を選択するか、または、同じホストで稼動している、別の HTTP サーバーで使用されるアカウントを選択することができます。ただしシステムによっては、ユーザー nobody は、ファイルを所有することはできますが、プログラムは実行できない場合があります。

「Server Settings」ページにアクセスするには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Preferences」タブをクリックします。
2. 「Server Settings」リンクをクリックします。
3. 変更を行い、「OK」をクリックします。

スーパーユーザー設定の変更

管理サーバーのスーパーユーザーのアクセスを設定できます。この設定は、スーパーユーザーアカウントにのみ影響します。つまり、管理サーバーが分散管理方式を採用している場合には、許可する管理者に対しては、別のアクセス制御を設定する必要があります。

警告 ユーザーやグループを管理するのに Sun ONE Directory Server を使用する場合、スーパーユーザー名やパスワードを変更する前に、ディレクトリ内のスーパーユーザーエントリを更新する必要があります。先にディレクトリを更新しないと、管理サーバーの「Users & Groups」フォームにアクセスできません。これに対処するには、このディレクトリにアクセスできる管理者アカウントを使用して管理サーバーにアクセスするか、または Sun ONE Directory サーバーコンソールや設定ファイルを使用してディレクトリを更新する必要があります。

管理サーバーのスーパーユーザー設定を変更するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Preferences」タブをクリックします。

2. 「Superuser Access Control」リンクをクリックします。
3. 変更を行い、「OK」をクリックします。

注 管理サーバーのユーザーを、root からオペレーティングシステム上の別のユーザーに変更できます。これにより、同じグループに属する複数のユーザーが設定ファイルを編集、管理できるようになります。ただし、UNIX/Linux プラットフォームの場合は、インストーラが任意のグループに設定ファイルの「rw」(読み書き) を許可しますが、Windows プラットフォームの場合は、ユーザーは「Administrators」グループに属している必要があります

スーパーユーザーのユーザー名とパスワードは、*server_root*/https-admserv/config/admpw ファイルに格納されています。ユーザー名を忘れた場合は、このファイルを開いてユーザー名を確認できます。ただし、パスワードは、暗号化されているため読み取ることはできません。このファイルは、username:password の書式で記述されています。パスワードを忘れた場合は、admpw ファイルを開き、暗号化されているそのパスワードをまず削除します。次に「Server Manager」フォームへ移動し、新規のパスワードを指定します。

警告 admpw ファイルは編集可能なため、サーバーコンピュータを安全な場所に保管し、ファイルシステムへのアクセスを厳しく制限することは非常に重要です。

- UNIX/Linux システムでは、ファイルの書き込みは root のみに限定するようにファイルの所有権を変更することを検討してください。さもないと、どのようなシステムユーザーでも管理サーバーデーモンを実行できるようになってしまいます。
- Windows システムでは、ファイル所有権は、管理サーバーが使用するユーザーアカウントに限定します。

複数の管理者の許可

分散管理により、複数の管理者がサーバーの特定の部分を変更することができます。

注	分散管理を機能させるには、デフォルトの Directory Server は、LDAP ベースのディレクトリサーバーである必要があります。

分散管理では、ユーザーは 2 つのレベルに分類されます。

- **スーパーユーザー**とは、`server_root/https-admserv/config/admpw` ファイルに記載されているユーザーです。これは、インストール時に指定したユーザー名 (およびパスワード) になります。このユーザーは、管理サーバーのすべてのフォーム (ただし、「Users & Groups」フォームは除く) へのすべてのアクセス権を持っています。「Users & Groups」フォームへは、Sun ONE Directory Server のような LDAP サーバーで有効なアカウントを持つスーパーユーザーがアクセスできます。
- **管理者**は、管理サーバーを含む、特定のサーバーの「Server Manager」フォームへ直接、アクセスできます。フォームの内容は、設定されているアクセス制御規則 (通常はスーパーユーザーにより設定される) により変わります。管理者は、限定された管理業務を実行でき、また、ユーザーの追加、アクセス制御の変更などその他のユーザーに影響する項目を変更できます。

アクセス制御の詳細については、182 ページの第 9 章「サーバーへのアクセス制御」に記載されている「アクセス制御とは」を参照してください。

注	分散管理を有効にする前に、Directory Server をインストールする必要があります。詳細は、Sun ONE Web Server の『インストールおよび移行ガイド』および『Sun ONE Directory Server 管理者ガイド』を参照してください。

分散管理を有効にするには、次の手順を実行します。

1. Directory Server がインストールされていることを確認します。
2. 管理サーバーへアクセスします。
3. Directory Server をインストールしたら、管理グループをまだ作成していない場合には管理グループを作成する必要があります。

   管理グループを作成するには、次の手順を実行します。

   a. 「Users & Groups」タブを選択します。

   b. 「New Group」リンクをクリックします。

c. LDAP ディレクトリに「administrators」グループを作成し、管理サーバー (または、サーバー root にインストールされたその他のサーバー) を設定することを許可したいユーザーの名前を追加します。「administrators」グループ内のすべてのユーザーは、管理サーバーへのすべてのアクセス権を保持していますが、アクセス制御を使用して、それらのユーザーが設定できるサーバーやフォームを制限することもできます。

警告 アクセス制御リストを作成すると、このリストに分散管理グループが追加されます。「administrators」というグループ名を変更する場合は、参照先のグループを変更するため、アクセス制御リストを手動で編集する必要があります。

4. 「Preferences」タブを選択します。
5. 「Distributed Admin」リンクをクリックします。
6. 変更を行い、「OK」をクリックします。

詳細は、オンラインヘルプの「Distributed Administration ページ」を参照してください。

ログファイルオプションの指定

管理サーバーのログファイルには、サーバーに関するデータが記録されます。これには、検出したエラーのタイプやサーバーアクセスに関する情報が記述されます。ログを表示することで、検出したエラーのタイプや特定のファイルがアクセスされた時間などのデータが得られ、サーバーのアクティビティを監視したり、障害追跡に役立てたりすることができます。

「Log Preferences」ページを使用して、管理サーバーログに記録されるデータのタイプや書式を指定できます。たとえば、管理サーバーにアクセスするすべてのクライアントについてログデータを記録したり、特定のクライアントをログから省いたりすることができます。また、サーバーについて決まった量の情報を提供する共通ログファイル形式を選択したり、必要に応じてカスタムログファイル書式を作成したりすることもできます。

「Preferences」タブを選択し、「Logging Options」リンクをクリックして、管理サーバーの「Log Preferences」ページにアクセスします。

詳細は、オンラインヘルプの「Logging Options ページ」、および第 10 章「ログファイルの使用」を参照してください。

ログファイルの表示

管理サーバーのログファイルは、サーバーのルートディレクトリの admin/logs に格納されます。たとえば、Windows の場合、ログファイルへのパスは、c:¥Sun¥server6¥https-admserv¥logs などのようになります。エラーログとアクセスログについては、両方とも、Sun ONE Web Server コンソールから表示したり、テキストエディタを使用して表示することができます。

アクセスログファイル

アクセスログには、サーバーへの要求やサーバーからの応答に関する情報が記録されます。

アクセスログファイルを表示するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Preferences」タブをクリックします。
2. 「View Access Log」リンクをクリックして、「OK」をクリックします。

詳細は、オンラインヘルプの「View Error Log ページ」、および第 10 章「ログファイルの使用」を参照してください。

エラーログファイル

エラー ログには、ログファイル作成以降にサーバーが検出したエラーすべてが列記されます。このファイルには、サーバーの起動時刻や、サーバーへのログインに失敗したユーザー名などのサーバーに関する情報メッセージも記述されます。

エラーログファイルを表示するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Preferences」タブをクリックします。
2. 「View Error Log」リンクをクリックして、「OK」をクリックします。

詳細は、オンラインヘルプの「View Access Log ページ」、および第 10 章「ログファイルの使用」を参照してください。

ログファイルの保管

ログファイルは、自動的に保管されるように設定できます。特定の時刻、または特定間隔の経過後に、Sun ONE Web Server はアクセスログをローテーションします。Sun ONE Web Server は、古いログファイルを保存し、そのファイルに保存日時を含む名前を付けます。

たとえば、ファイルを 1 時間ごとにローテーションするように設定すると、Sun ONE Web Server は「access.199907152400」という名前を付けてファイルを保存します。この名前では、「ファイル名 | 年 | 月 | 日 | 時刻 (24 時間表示)」が 1 つの文字列で表わされます。アクセスログアーカイブファイルの書式は、厳密には、設定するログローテーションのタイプにより異なります。

アクセスログロテーションは、サーバーの起動時に初期化されます。ローテーションを有効にすると、Sun ONE Web Server はタイムスタンプの付いたアクセスログファイルを作成し、ローテーションがサーバーの起動時に開始されます。

ローテーションが開始されると、Sun ONE Web Server は、アクセスログファイルに記録する必要のある要求があったときに、タイムスタンプの付いたアクセスログファイルを新規作成します。これは事前にスケジュールされた「次のローテーション時刻」の後で実行されます。

schedulerd 制御ベースのログローテーション の使用 (UNIX/Linux)

Sun ONE Web Server のいくつかの機能を設定して、自動的に操作したり、特定の時刻に起動するように設定したりできます。schedulerd 制御デーモンがコンピュータの時刻を確認し、一定の時刻に処理を開始します。(これらの設定は、schedulerd ファイルに格納されます。

この schedulerd 制御デーモンは、Sun ONE Web Server の cron タスクを制御し、管理サーバーから起動したり、停止したりできます。cron プロセスが実行するタスクは、サーバーの種類に依存します。(Windows プラットフォームでは、スケジューリングは、個々のサーバー内で行われることに注意してください。

schedulerd 制御デーモンが制御できるタスクのなかには、コレクションの保守のスケジューリングやログファイルの保管などがあります。スケジュールされたタスクの設定を変更した場合は、その都度 schedulerd 制御デーモンを再起動する必要があります。

schedulerd 制御デーモンを再起動、起動、または停止するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Global Settings」タブをクリックします。
2. 「Cron Control」リンクをクリックします。
3. 「Restart」、「Start」、「Stop」をクリックし schedulerd 制御を変更します。

schedulerd にタスクを追加した場合は、その都度デーモンを再起動する必要があることに注意してください。

ディレクトリサービスの設定

ユーザーの名前やパスワードなどの情報は、LDAP (Lightweight Directory Access Protocol) というオープンシステムのサーバープロトコルを使用して、1 つの Directory Server で保管し、管理することができます。また、サーバーを設定して、簡単にアクセスできる複数のネットワークロケーションからユーザーがディレクトリ情報を引き出せるようにすることもできます。

ディレクトリサービスの詳細を設定するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Global Settings」タブをクリックします。
2. 「Configure Directory Service」リンクをクリックします。
3. 変更を行い、「OK」をクリックします。

詳細は、オンラインヘルプの「Configure Directory Service ページ」を参照してください。

サーバーへのアクセスの制限

サーバーへのアクセスの制御は、サーバー全体に対して、またはサーバーの一部 (ディレクトリ、ファイル、ファイルタイプなど) に対して行うことができます。サーバーが受信した要求を評価する場合、アクセス制御エントリ (ACE) と呼ばれる規則の階層に基づいてアクセス権を決定し、一致するエントリを使用して、要求を承認するか、拒否するかを決定します。各 ACE は、サーバーが階層内の次の ACE に進むべきかどうかを指定します。ACE の集合は、アクセス制御リスト (ACL) と呼ばれます。要求がサーバーに着信すると、サーバーは、*vsclass*.obj.conf (*vsclass* は仮想サーバーのクラス名) で ACL を検索して参照し、アクセスの可否を決定します。デフォルトでは、サーバーには、複数の ACL が含まれる 1 つの ACL ファイルがあります。

アクセス制御は、管理サーバーを使用してすべてのサーバーに対して全体的に設定したり、サーバーマネージャを使用して特定のサーバーインスタンス内のリソースに対して設定したりすることができます。リソースに対するアクセス制御の設定の詳細は、第 9 章「サーバーへのアクセス制御」に記載されている「アクセス制御の設定」(193 ページ) を参照してください。

注	サーバーへのアクセスを制限する前に、分散管理を有効にする必要があります。

Sun ONE Web Server へのアクセスを制限するには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Global Settings」タブをクリックします。

2. 「Restrict Access」リンクをクリックします。
3. 目的のサーバーを選択し、「Create ACL」をクリックします。

   管理サーバーには、指定したサーバーのアクセス制御規則が表示されます。
4. アクセス制御の変更を行い、「OK」をクリックします。詳細は、オンラインヘルプの「Restrict Access ページ」を参照してください。

第 6 章

証明書と鍵の使用

この章では、証明書と鍵の認証を使用した、Sun ONE Web Server 6.1 のセキュリティ保護について説明します。データを保護し、侵入者からのアクセスを拒否して、適切なユーザーからのアクセスを許可するための、各種セキュリティ機能を有効化する方法について説明します。Sun ONE Web Server 6.1 には、Sun ONE サーバーのすべてのセキュリティアーキテクチャが統合されています。Sun ONE Web Server 6.1 のセキュリティアーキテクチャは、相互運用性と整合性を最大限確保するため、業界標準および標準プロトコルに基づいて構築されています。

この章を読む前に、公開鍵の暗号法に関する基本概念をよく知っておくことをお勧めします。知っておくべき概念には、暗号化と復号化、公開鍵と非公開鍵、電子証明書および暗号化プロトコルなどがあります。詳細は、「Introduction to SSL」を参照してください。

Web サーバーをセキュリティ保護する手順の詳細は、次の各節で説明します。

- 証明書ベースの認証
- 信頼データベースの作成
- VeriSign 証明書の要求およびインストール
- 他のサーバー証明書の要求およびインストール
- アップグレード時の証明書の移行
- 証明書の管理
- CRL と CKL のインストールと管理
- セキュリティに関する詳細設定
- 外部暗号化モジュールの使用
- クライアントセキュリティ要件の設定
- Stronger Ciphers の設定
- セキュリティに関するその他の問題

証明書ベースの認証

認証とは、同一性 (ID) を確認するためのプロセスのことです。ネットワークを利用した対話の中で、一方のグループは、認証によって他方のグループとの同一性を識別します。証明書は、認証をサポートする方法の 1 つです。

証明書を使用した認証

証明書は、個人、企業、またはその他のエンティティの名前を指定するデジタルデータで構成され、その公開鍵が、証明書に含まれていれば、そのエンティティに属しているという証明となります。クライアントとサーバーの両方に証明書を持たせることができます。

証明書は、認証局 (CA) によって発行され、デジタル署名がなされます。CA は、インターネットを通じて証明書を販売する企業の場合も、企業のイントラネットやエクストラネットの証明書の発行を担当する、企業内の部門の場合もあります。ユーザーの同一性 (ID) の検証手段として、どの CA を信頼するかはユーザーが決定します。

証明書には、公開鍵および証明書によって識別されるエンティティの名前のほかに、有効期限、証明書を発行した CA の名称および証明書を発行する CA の「デジタル署名」も含まれます。証明書の内容および書式については、「Introduction to SSL」を参照してください。

注	暗号化機能を有効にするには、事前にサーバー証明書をインストールしておく必要があります。

サーバー認証

サーバー認証とは、クライアントによる、サーバーの確実な ID、すなわち特定のネットワークアドレスにあるサーバーに対して責任を持つとされている組織の ID です。

クライアント認証

クライアント認証とは、サーバーによる、クライアントの確実な ID、すなわちクライアントソフトウェアを使用していると見なされる人の ID です。クライアントは、複数の証明書を所有できます。これは、1 人の人が数種類の ID を所有しているのと同じことです。

仮想サーバー証明書

仮想サーバーごとに異なる証明書データベースを持つことができます。各仮想サーバーのデータベースには、複数の証明書を含めることができます。仮想サーバーも同様に各インスタンス内に複数の異なる証明書を所有できます。

信頼データベースの作成

サーバー証明書を要求する前に、信頼データベースを作成しておく必要があります。Sun ONE Web Server では、管理サーバーと各サーバーのインスタンスが、それぞれ専用の信頼データベースを所有できます。信頼データベースは、ローカルマシン上にだけ作成できます。

信頼データベースを作成するときには、鍵ペアファイルに使用されるパスワードを指定します。このパスワードは、暗号化された通信を使用してサーバーを起動させるときにも必要です。パスワードを変更するときに考慮するガイドラインのリストは、152 ページの「パスワードまたは PIN の変更」を参照してください。

信頼データベースでは、鍵ペアファイルと呼ばれる公開鍵と非公開鍵を作成し、保存します。鍵ペアファイルは、SSL 暗号化に使用されます。サーバー証明書を要求し、インストールするときには、鍵ペアファイルを使用します。証明書は、インストールしたあとに信頼データベースに格納されます。鍵ペアファイルは、次のディレクトリ内に暗号化されて保存されます。

server_root/alias/<serverid-hostname>-key3.db.

管理サーバーは、信頼データベースを 1 つだけ所有できます。また、サーバーに含まれる各インスタンスは、それぞれ専用の信頼データベースを所有できます。仮想サーバーは、そのサーバーインスタンス用に作成された信頼データベースによってカバーされます。

信頼データベースの作成

信頼データベースを作成するには、次の手順を実行します。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Security」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

2. 「Create Database」リンクをクリックします。
3. データベースのパスワードを入力します。
4. 操作を繰り返します。
5. 「OK」をクリックします。
6. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

password.conf の使用

デフォルトでは、管理者は、Web サーバーを起動するときに、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。Web 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。このファイルと鍵のデータベースが危険にさらされないようにするために、これを行うのはシステムが充分にセキュリティ保護されている場合だけにします。

通常、サーバーは起動する前にパスワードを要求するため、/etc/rc.local ファイルまたは /etc/inittab ファイルで、UNIX の SSL 有効サーバーを起動することはできません。ファイル内にプレーンテキストでパスワードを保存しておくと SSL 有効サーバーを自動的に起動することができますが、これは推奨される方法ではありません。サーバーの password.conf ファイルは、root またはサーバーをインストールしたユーザーだけが所有し、その所有者のみがこのファイルの読み書きアクセス権を持つべきです。

UNIX で、password.conf ファイル内に SSL が有効なサーバーのパスワードを保存しておくことは、セキュリティ上のリスクが大きくなります。ファイルにアクセスできるユーザーなら誰でも、SSL が有効なサーバーのパスワードにアクセスできます。したがって、SSL が有効なサーバーのパスワードを password.conf ファイルに保存する前に、セキュリティ上のリスクについて検討しておく必要があります。

Windows では、NTFS ファイルシステムを使用する場合は、password.conf ファイルを使用しなくても、アクセス制限によってこのファイルの保存されているディレクトリのセキュリティを保護してください。ただしこのディレクトリには、管理サーバーのユーザーと Web サーバーのユーザーに対して読み取り / 書き込み許可を持たせる必要があります。ディレクトリのセキュリティを保護しておくと、他者が偽の password.conf ファイルを作成することを防げます。FAT ファイルシステム上では、ディレクトリやファイルへのアクセスを制限しても、ディレクトリやファイルのセキュリティを保護することはできません。

SSL 有効サーバーの自動起動

セキュリティ上のリスクが問題とならない場合は、以下の手順を実行して SSL が有効なサーバーを自動的に起動します。

1. SSL が有効になっていることを確認します。
2. サーバーインスタンスの config サブディレクトリ内に、新規の password.conf ファイルを作成します。
   - 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:*your_password*

- それ以外の PKCS#11 モジュール (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) を使用している場合は、PKCS#11 モジュールの名前を指定し、その後ろにパスワードを入力します。その例を次に示します。

  `nFast:`*your_password*

3. 新しい設定が有効になるように、サーバーを停止させてからもう一度起動させます。

`password.conf` ファイルを作成した後でも、Web サーバーを起動させるときには、毎回パスワードを入力するよう求めるプロンプトが表示されます。

VeriSign 証明書の要求およびインストール

VeriSign は、Sun ONE Web Server の推奨する認証局です。VeriSign の VICE プロトコルは、証明書要求プロセスをシンプルにします。VeriSign は、直接サーバーに対して証明書を返せるという利点があります。

サーバーに証明書信頼データベースを作成後、証明書を要求し、認証局 (CA) にこれを提出できます。会社に独自の内部 CA がある場合には、そこから発行される証明書を要求します。商用 CA からの証明書購入を予定している場合には、CA を選定し、CA が必要とする情報の特定の書式を入手してください。Web サイトのリンク先を含む、利用可能な認証局のリストは、「Request a Certificate」ページにあります。CA が必要とする情報については、「Request a Certificate」ページの下の「Server Administrator」ページおよび「Server Manager Security」ページの「List of Certificate Authorities」(認証局リスト) に記載されています。

管理サーバーは、サーバー証明書を 1 つしか所有できません。各サーバーのインスタンスは、専用のサーバー証明書を所有できます。各仮想サーバーについては、サーバーインスタンスの証明書を選択することができます。

VeriSign 証明書の要求

VeriSign 証明書を要求するには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Security」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

2. 「Request VeriSign Certificate」リンクをクリックします。
3. 必要な手順を確認します。

4. 「OK」をクリックします。
5. VeriSign の手順に従います。

VeriSign 証明書のインストール

VeriSign 証明書を要求し、承認が得られたら、1 ～ 3 日ほどで「Install Verisign Certificate」ページのドロップダウンリストに証明書が表示されます。VeriSign 証明書をインストールするには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Security」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。
2. 「Install VeriSign Certificate」リンクをクリックします。
3. 外部の暗号化モジュールを使用する場合以外は、暗号化モジュールのドロップダウンリストから「internal (software) 」を選択します。
4. 鍵ペアファイルのパスワードまたは PIN を入力します。
5. ドロップダウンリストから「Transaction ID to Retrieve」を選択します。

   通常は、一番下の選択肢に該当します。
6. 「OK」をクリックします。
7. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

他のサーバー証明書の要求およびインストール

VeriSign のほかに、他の認証局からの証明書を要求し、インストールすることができます。CA のリストは、「Request a Certificate」ページの下の「Server Administrator」ページと「Server Manager Security」ページで入手できます。会社または組織が独自の内部証明書を提供している場合もあります。この節では、このような他の種類のサーバー証明書を要求しインストールする方法について説明します。

必要な CA 情報

要求処理を始める前に、CA が必要とする情報を確認しておく必要があります。商用の CA が発行するサーバー証明書を要求する場合でも、内部 CA に要求する場合でも、次の情報を提供する必要があります。

- **共通名** (Common Name) は、DNS 検索で使用される絶対パスによるホスト名である必要があります (たとえば、*www.sun.com*)。これは、ブラウザがサイトに接続するのに使用する URL 内のホスト名です。これら 2 つの名前が一致しない場合、証明書名とサイトの名前が一致していないため証明書の認証性に疑いがあることが、クライアントに対して通知されます。CA によっては異なる情報を必要としていることもあるため、これらについて確認することが重要です。

  内部 CA から証明書を要求する場合は、このフィールドにワイルドカードおよび正規表現で入力できます。ほとんどのベンダーでは、共通名の入力にワイルドカードや正規表現を使用した証明書の要求を承認しません。

- **電子メールアドレス** (Email Address) は、ユーザーがビジネスで使用する電子メールアドレスです。これは、ユーザーと CA との間の連絡に使用されます。

- **組織** (Organization) は、ユーザーの会社、教育機関、提携先などの公式かつ法律上の名前です。ほとんどの CA が、この情報を法的文書 (ビジネスライセンスのコピーなど) で証明するように要求します。

- **組織単位** (Organizational Unit) は、会社内組織について記述する、オプションの (省略可能な) フィールドです。このフィールドには、たとえば *Inc.* や *Corp.* などを付けないなど、正式ではない会社名を記述しておくのに使用することもできます。

- **市区町村名** (Locality) は、通常は、組織が所在する都市、郡または地方名を記述する、オプションのフィールドです。

- **都道府県名** (State or Province) は、通常必須ですが、いくつかの CA では省略可能である場合があります。ほとんどの CA では都道府県名の省略した表記は認められませんが、念のため確認してください。

- **国名**は必須です。国名を 2 文字の省略形 (ISO 書式) で入力します。米国の国コードは US です。

これらの情報の全体は、識別名 (DN) と呼ばれ、属性と属性値のペアの系列のように結合されており、証明書のサブジェクトを一意に識別することができます。

商用の CA から証明書を購入する場合は、証明書が発行される前に、上記のほかにどんな情報が必要とされているのか知るために、事前に CA に確認しておく必要があります。ほとんどの CA では、身分証明書を要求してきます。たとえば、会社名や、会社によってサーバー管理者権限を与えられている人の名前を確認します。そして、場合によっては、提供した情報を使用する法的権利をユーザーが持っているかどうかを尋ねられることもあります。

一部の商用 CA では、さらに徹底した識別情報を提供した組織や個人に対して、さらに詳細で正確性の高い証明書を発行します。たとえば、個人が www.sun.com というサイトが動作しているコンピュータの正当な管理者であるということを確認したことに加えて、企業が過去 3 年間に渡って運営されており、現在顧客と係争中の訴訟が無いことを CA が確認したことを記述した証明書を購入することもできます。

他のサーバー証明書の要求

証明書を要求するには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Security」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

2. 「Request a Certificate」リンクをクリックします。
3. 新しい証明書か証明書の更新かを選択します。

   多くの証明書は、6 か月や 1 年などの一定期間が経過すると、有効期限が終了になります。自動的に更新した証明書を送信してくる CA もあります。

4. 証明書の要求を送信する方法を指定するには、次の手順に従います。

   - CA が電子メールのメッセージで要求を受け付けている場合は、「CA Email」にチェックマークを付け、CA の電子メールアドレスを入力します。CA のリストが必要な場合には、「List of available Certificate Authorities」をクリックします。
   - Netscape Certificate Server を使用している内部 CA が発行する証明書を要求する場合は、「CA URL」をクリックし、Certificate Server の URL を入力します。この URL には、証明書の要求を扱う証明書サーバーのプログラムを指定する必要があります。URL の例は、次のとおりです。`https://CA.mozilla.com:444/cms`

5. ドロップダウンリストから、証明書を要求するときに使用する鍵ペアファイルの暗号化モジュールを選択します。

6. 鍵ペアファイルのパスワードを入力します。

   このパスワードは、内部モジュール以外の暗号化モジュールを選択していないかぎり、信頼データベースを作成したときに指定したパスワードと同一です。サーバーは、このパスワードを使用して、ユーザーの非公開鍵を取得したり、CA に対するメッセージを暗号化します。そして、ユーザーの公開鍵と暗号化されたメッセージの両方を CA に送信します。CA は、公開鍵を使用してメッセージを復号化します。

7. ユーザーの ID 情報を入力します。

   この情報の書式は、CA によって異なります。これらのフィールドの一般的な説明は、「Request a Certificate」ページの下の「Server Administrator」ページおよび「Server Manager Security」ページの「List of Certificate Authorities」(認証局) に記載されています。これらの情報のほとんどは、証明書の更新の場合には通常必要ありません。

8. 正確に行うため、入力内容を見直します。

   情報が正確であれば、証明書の承認も早まります。要求を証明書サーバーに送るとき、送信する前に、フォーム情報を確認するよう求めるプロンプトが表示されます。

9. 「OK」をクリックします。

10. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

サーバーは、入力した情報を含む証明書要求を作成します。要求には、ユーザーの非公開鍵を使用して作成されたデジタル署名が含まれます。CA は、デジタル署名を使用して、サーバーマシンから CA に送付されている間、その要求が不正に変更されていないことを確認します。まれに要求が不正に変更されたような場合には、通常 CA から電話などでの連絡があります。

要求を電子メールで送信する場合には、サーバーがその要求を含んだ電子メールメッセージを作成して、CA に送信します。通常、電子メールにより証明書が返されます。証明書サーバーに URL を指定した場合は、サーバーがその URL を使用して Certificate Server にその要求を送信します。電子メールで返信を受けるか、その他の手段になるかは、CA によって異なります。

CA は、証明書を発行することに同意するかどうかを通知します。ほとんどの場合、CA は、電子メールで証明書を送信します。所属している組織が証明書サーバーを使用している場合には、証明書サーバーのフォームを使用して証明書を検索できます。

注	商用 CA に証明書を要求しても、必ず証明書が発行されるとはかぎりません。多くの CA で、証明書の発行前に、ユーザーが自らの ID を示すことが要求されます。証明書発行の承認には 1 日から 2 か月かかることがあります。つまり、必要な情報をすべて迅速に CA に提供することが重要です。

証明書を受け取ったら、それをインストールできます。それまでの間は、SSL を使用せずにサーバーを運用することになります。

他のサーバー証明書のインストール

CA から証明書を受け取るときには、ユーザーだけがこれを復号化できるように、公開鍵で暗号化されています。信頼データベースの正しいパスワードを入力しないと、証明書を復号化しインストールすることはできません。

証明書には、次の 3 種類があります。

- クライアントに提示するための、ユーザーのサーバーの証明書
- 証明書チェーンで使用される、CA の独自の証明書
- 信頼できる CA の証明書

証明書チェーンは、連続した認証局によって署名された、一連の階層的証明書です。CA 証明書は、認証局 (CA) の ID を示すもので、その機関によって発行される証明書に署名するのに使用されます。CA 証明書は、次に親 CA の CA 証明書によって署名されるというように、順にルート CA まで署名されることができます。

注	CA が CA の証明書を自動的にユーザーに送信しない場合には、ユーザーから証明書を要求する必要があります。多くの CA は、ユーザーの証明書を電子メールで送信する際に CA 証明書も同時に送信してくるため、ユーザーのサーバーは、両方の証明書を同時にインストールします。

CA から証明書を受け取るときには、ユーザーだけがこれを復号化できるように、公開鍵で暗号化されています。サーバーは、証明書をインストールする際、その証明書を復号するのに指定した鍵ペアファイルのパスワードを使用します。サーバーがアクセス可能な場所にその電子メールを保存するか、または次に説明する「Install Certificate」フォームにペーストするためにその電子メールのテキストをコピーしておきます。

証明書のインストール

証明書をインストールするには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Security」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

2. 「Install Certificate」リンクをクリックします。
3. インストールする証明書の種類を確認します。
   - 「This Server」は、使用しているサーバーだけに関係する 1 つの証明書に使用します。
   - 「Server Certificate Chain」は、証明書チェーンに組み込む CA の証明書に使用します。
   - 「Trusted Certificate Authority (CA)」は、クライアントの認証のための信頼できる CA として受け入れたい CA の証明書に使用します。
4. ドロップダウンリストから「Cryptographic Module」を選択します。
5. 鍵ペアファイルのパスワードを入力します。
6. 次の場合を除いて、証明書がこのサーバーインスタンスにだけに使用される場合は、証明書フィールドの名前を空白のままにします。
   - 複数の証明書を仮想サーバーに使用する場合。

     サーバーインスタンス内で一意の証明書名を入力します。
   - 内部モジュール以外の暗号化モジュールを使用する場合。

     1 つの暗号化モジュール内のすべてのサーバーインスタンスで一意の証明書名を入力します。

   名前は入力されると、「Manage Certificates」リストに表示されるため、内容を識別しやすい名前にしてください。たとえば、「United States Postal Service CA」は CA の名前で、「VeriSign Class 2 Primary CA」は CA と証明書の種類の両方を表しています。証明書名を入力しない場合は、デフォルトの値が使用されます。
7. 次のいずれかを選択します。
   - 電子メールが保存されているファイルへの、絶対パスを入力する
   - 「Message text (with headers)」というフィールドに、電子メールのメッセージテキストをペーストする

     テキストをコピーしてペーストする場合には、必ず、「Begin Certificate」と「End Certificate」の 2 つのヘッダーを入れます。メッセージの最初と最後にあるハイフンも含める必要があります。
8. 「OK」をクリックします。

9. 次のいずれかを選択します。
   - 新しい証明書をインストールする場合は、「Add Certificate」
   - 更新された証明書をインストールする場合は、「Replace Certificate」
10. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

証明書は、サーバーの証明書データベース内に格納されます。ファイル名は、`<alias>-cert8.db` となります。その例を次に示します。

`https-`*`serverid-hostname`*`-cert8.db`

アップグレード時の証明書の移行

iPlanet Web Server 4.1 または 6.0 から移行する場合には、ユーザーのファイル (信頼データベースと証明書データベースを含む) は自動的に更新されます。

しかし、Enterprise Server 3.x からアップグレードする場合には、ユーザーの信頼データベースと証明書データベースを移行する必要があります。Sun ONE Web Server 6.1 管理サーバーのユーザーが古い 3.x データベースのファイルに対して読み書き許可を持っていることを確認してください。ファイル名は `<alias>-cert.db` および `<alias>-key.db` で、`<3.`*`x`*`_server_root>/alias` ディレクトリ内にあります。

サーバーでセキュリティが有効になっている場合だけ、鍵ペアファイルと証明書が移行されます。「Administration Server」ページと「Server Manager」ページの「Security」タブを使用して、鍵と証明書を移行させることもできます。

旧バージョンでは、証明書と鍵ペアファイルは、複数のサーバーインスタンスによって使用できるエイリアスによって参照されていました。管理サーバーは、すべてのエイリアスとそれらの構成要素である証明書を管理していました。Sun ONE Web Server 6.1 では、管理サーバーと各サーバーインスタンスに独自の証明書と鍵ペアファイルがあり、エイリアスではなく信頼データベースとして参照されます。

サーバー証明書とすべての含まれている認証局を含む、信頼データベースとその構成要素である証明書を管理するには、それら自体の管理については管理サーバーを、サーバーインスタンスについてはサーバーマネージャを使用します。証明書および鍵ペアデータベースファイルは、それらを使用するサーバーインスタンス名をとって、名付けられます。以前のバージョンで複数のサーバーインスタンスが同じエイリアスを共用していた場合は、移行されると、証明書と鍵ペアファイルは新しいサーバーインスタンスの名前をとって名前変更されます。

サーバーインスタンスに関連のある信頼データベース全体が移行されます。以前のデータベースにリストされている認証局はすべて、Sun ONE Web Server 6.1 データベースに移行されます。CA が重複している場合には、有効期限が切れるまで以前の CA を使用します。重複している CA は削除しないでください。

組み込みルート証明書モジュールの使用

ダイナミックに読み込み可能なルート証明書モジュールが、Sun ONE Web Server 6.1 に付属しており、VeriSign を含む多数の CA のルート証明書が格納されています。ルート証明書モジュールを使用すると、旧バージョンと比べて、より簡単な方法でルート証明書を新しいバージョンにアップグレードできます。旧バージョンでは、古いルート証明書を 1 つずつ削除し、その後新しいルート証明書を 1 つずつインストールする必要がありました。よく知られている CA 証明書をインストールすると、ルート証明書モジュールファイルを、将来 Sun ONE Web Server や Service Pack の新しいバージョンへ更新するだけですみます。

ルート証明書は PKCS#11 暗号化モジュールとして実装されているため、モジュールに含まれているルート証明書を削除することはできません。削除のオプションは、ルート証明書を管理するときには提供されません。サーバーインスタンスからルート証明書を削除する場合は、サーバーの `alias` ファイル内で次の情報を削除すれば、ルート証明書モジュールを無効にできます。

- `libnssckbi.so` (ほとんどの UNIX プラットフォームの場合)
- `libnssckbi.sl` (HP-UX の場合)
- `nssckbi.dll` (Windows の場合)

ルート証明書モジュールをあとで復元する場合は、`bin/https/lib` (UNIX および HP) または `bin¥https¥bin` (Windows) から、該当する拡張子を持つファイルを `alias` サブディレクトリにコピーして、後から元の場所に戻すことができます。

ルート証明書の信頼情報は変更できます。信頼情報は、編集されるサーバーインスタンスの証明書データベースに書き込まれ、ルート証明書モジュールそのものには戻されません。

証明書の管理

ユーザーのサーバーにインストールされたさまざまな証明書の信頼の設定値を表示、削除または編集できます。これには、ユーザー自身の証明書や CA から取得した証明書も含まれます。

証明書リストを管理するには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Security」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

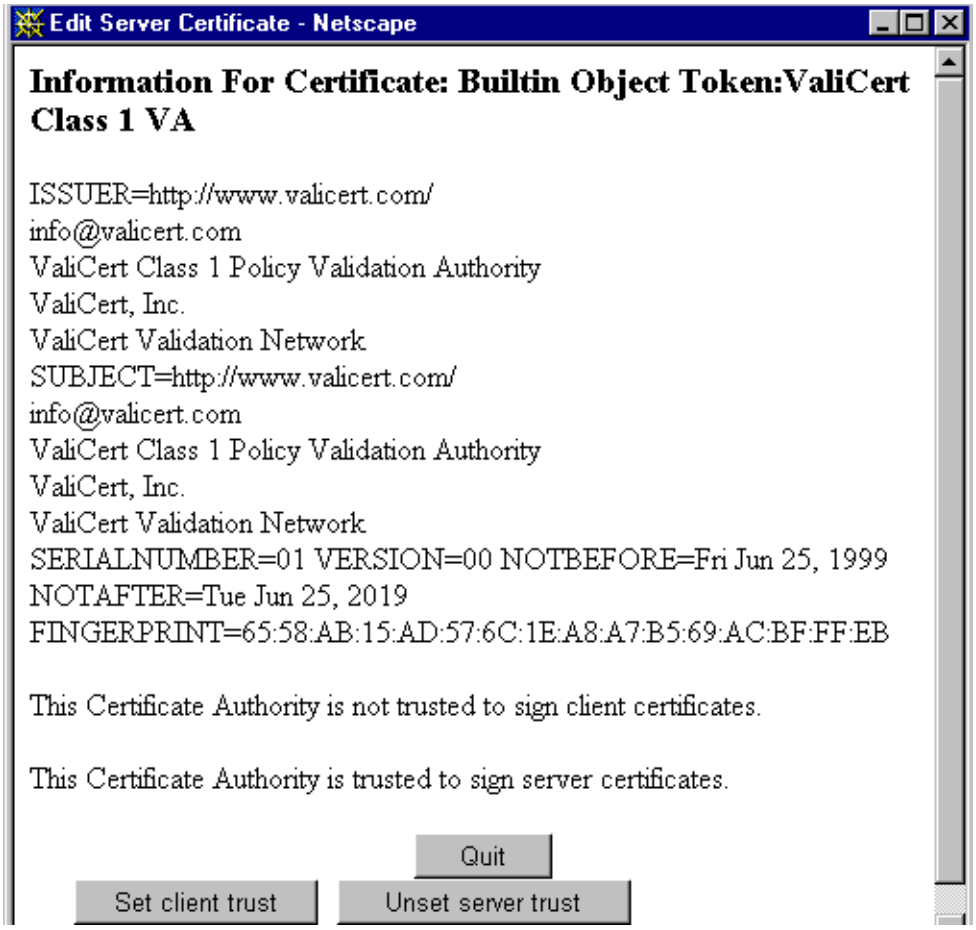
2. 「Manage Certificates」リンクをクリックします。

   - 内部暗号化モジュールを使用して、デフォルト設定の証明書を管理する場合には、インストールされているすべての証明書のリストがその種別および有効期限とともに表示されます。証明書はすべて、ディレクトリ *server_root*/`alias` に格納されています。
   - ハードウェアアクセラレータなどの外部の暗号化モジュールを使用している場合には、各モジュールのパスワードを最初に入力し、「OK」をクリックします。モジュール内に証明書が組み込まれ、証明書リストが更新されます。

3. 管理する「Certificate Name」をクリックします。

   その種類の証明書に関する管理オプションのある「Edit Server Certificate」ページが表示されます。クライアントの信頼情報を設定したり設定解除できるのは、CA 証明書だけです。外部の暗号化モジュールのなかには、証明書を削除できないものもあります。

   「Edit Server Certificate」ページ



4. 「Edit Server Certificate」ウィンドウには、次の選択肢があります。
   - 内部的に取得した証明書については、「Delete Certificate」または「Quit」
   - CA から発行された証明書については、「Set client trust」、「Unset server trust」、または「Quit」
5. 「OK」をクリックします。

6. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

証明書情報には、所有者と発行者が含まれます。

信頼の設定では、クライアントの信頼情報を設定したり、サーバーの信頼情報の設定を解除したりできます。LDAP サーバー証明書の場合は、サーバーが信頼されている必要があります。

CRL と CKL のインストールと管理

証明書の失効リスト (CRL) および危殆化鍵リスト (CKL) は、クライアントまたはサーバーのユーザーが信頼すべきでない証明書および鍵を知らせます。証明書のデータが変わった場合、たとえば、証明書の有効期限が切れる前にユーザーが事務所を変更したり、その組織を離れるような場合には、その証明書は無効になり、そのデータが CRL に表示されます。鍵が不正に変更されたり、その他不正に使用された場合には、その鍵とそのデータが CKL に表示されます。CRL と CKL は、両方とも CA によって作成され、定期的に更新されます。

CRL または CKL のインストール

CA から CRL または CKL を取得するには、次の手順を行います。

1. CRL または CKL をダウンロードするための、CA の URL を確認します。
2. ブラウザに URL を入力して、CA のサイトにアクセスします。
3. CA の指示に従って CRL または CKL をローカルディレクトリにダウンロードします。
4. 管理サーバーまたはサーバーマネージャにアクセスし、「Security」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。
5. 「Install CRL/CKL」リンクをクリックします。
6. 次のいずれかを選択します。
   - 「Certificate Revocation List」
   - 「Compromised Key List」
7. インストールするファイルへの絶対パス名を入力します。
8. 「OK」をクリックします。

- 「Certificate Revocation List」を選択した場合には、CRL 情報をリストした「Add Certificate Revocation List」ページが表示されます。
- 「Certificate Revocation Key List」を選択した場合には、CKL 情報をリストした「Add Compromised Key List」ページが表示されます。

注 データベースに CRL または CKL リストがすでにある場合には、「Replace Certificate Revocation List」ページまたは「Replace Compromised Key List」ページが表示されます。

9. 「Add」をクリックします。
10. 「OK」をクリックします。
11. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

CRL と CKL の管理

CRL と CKL を管理するには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Security」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

2. 「Manage CRL/CKL」リンクをクリックします。

   「Manage Certificate Revocation Lists /Compromised Key Lists」ページが表示されます。すべてのインストールされている Server CRL と Server CKL が、有効期限とともに一覧されます。

3. 「Server CRL」または「Server CKL」リストのどちらかから「Certificate Name」を選択します。
4. 次のいずれかを選択します。
   - 「Delete CRL」
   - 「Delete CKL」
5. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

セキュリティに関する詳細設定

証明書を取得すると、サーバーのセキュリティ保護を開始できます。Sun ONE Web Server は複数のセキュリティ要素を提供しています。

暗号化とは、情報を対象とした受信者以外の人が読めないような内容にするための、変換プロセスのことです。復号化とは、暗号化された情報を判読可能な状態に戻すための、変換プロセスのことです。Sun ONE Web Server は、SSL と TLS 暗号化プロトコルをサポートしています。

暗号化方式とは、暗号化または復号化に使用する暗号アルゴリズム (数学関数) です。SSL と TLS プロトコルには、多数の暗号化方式のセットが含まれています。暗号化方式には、他に比べて強力でよりセキュリティ性の高いものもあります。一般的に、暗号化方式で使用するビット数が多いほど、データの復号化は難しくなります。

双方向の暗号化プロセスでは、必ず、送信側と受信側の両方が同じ暗号化方式を使用する必要があります。多数の暗号化方式があるため、最も一般的に使用されている方式に対してサーバーを有効にしておく必要があります。

セキュリティ保護された接続時には、クライアントとサーバーは、通信に、その双方が持てる最も強力な暗号化方式を使用します。SSL2、SSL3 および TLS プロトコルから暗号化方式を選択できます。

注 SSL バージョン 2.0 より後のバージョンでは、安全性と性能が向上しています。このため、システムに SSL3 を使用できないクライアントが存在する場合を除き、SSL2 を使用すべきではありません。クライアント証明書は、SSL2 暗号化方式での動作が保証されていません。

暗号化プロセスだけでは、サーバーの機密情報のセキュリティ保護には十分ではありません。実際に暗号化結果を生成したり、すでに暗号化された情報を復号化するためには、暗号化方式と一緒に鍵を使用する必要があります。暗号化プロセスでは、この結果を出すために 2 つの鍵を使用します。1 つは公開鍵でもう 1 つが非公開鍵です。公開鍵を使用して暗号化された情報は、対応する非公開鍵を使用した場合にのみ復号化できます。公開鍵は、証明書の一部として発行され、対応する非公開鍵だけがセキュリティ保護されます。

各種暗号化方式のセットについての説明と、鍵および証明書については、「Introduction to SSL」を参照してください。

サーバーが使用できる暗号化方式を指定するには、リスト内で暗号化方式にチェックマークを付けます。特定の暗号化方式を使用してはならない理由がある場合を除き、すべてにチェックマークを付けるようにします。ただし、最適ではないと思われる暗号化方式を有効にする必要はありません。

警告 「No Encryption, only MD5 message authentication」は選択しないでください。クライアントサイドでその他の暗号化方式を利用できない場合には、サーバーがデフォルトによりこの設定を使用し、暗号化は行われません。

SSL と TLS プロトコル

Sun ONE Web Server 6.1 は、暗号化通信に SSL (Secure Sockets Layer) プロトコルと TLS (Transport Layer Security) プロトコルをサポートしています。SSL と TLS は、アプリケーションには依存せず、より高レベルのプロトコルをこの上に透過的に階層化することができます。

SSL と TLS の両プロトコルは、サーバーとクライアントを相互に認証するのに使用されるさまざまな暗号化方式をサポートし、証明書を送信してセッション鍵を確定します。クライアントとサーバーは、サポートしているプロトコルや、暗号化の強度についての会社の方針および暗号化されたソフトウェアの輸出に対する行政上の制約条件などの要因に基づいて、別の暗号化方式セットをサポートすることができます。他の機能の中でも特に、SSL と TLS ハンドシェイクプロトコルは、どの暗号化方式のセットを通信に使用するかをサーバーとクライアントが交渉する方法を決定します。

SSL を使用した LDAP との通信

管理サーバーは SSL を使用して LDAP と通信するようにする必要があります。管理サーバーで SSL を有効にするには、次の手順を行います。

1. 管理サーバーにアクセスして、「Global Settings」タブをクリックします。
2. 「Configure Directory Service」リンクをクリックします。
3. 「Yes」を選択して、接続に SSL (Secure Sockets Layer) を使用します。
4. 「Save Changes」をクリックします。
5. 「OK」をクリックして、SSL を介した LDAP の標準ポートにポートを変更します。

待機ソケットのセキュリティの有効化

次の方法で、サーバーの待機ソケットをセキュリティ保護できます。

- セキュリティ機能をオンにします。
- 待機ソケットのサーバー証明書を選択します。
- 暗号化方式を選択します。

セキュリティ機能をオンにする

待機ソケット用に他のセキュリティ設定を行うには、セキュリティ機能をオンにしておく必要があります。新しい待機ソケットを作成したり、既存の待機ソケットを編集するときに、セキュリティ機能をオンにできます。

待機ソケットの作成時にセキュリティ機能をオンにする

新しい待機ソケットを作成するときにセキュリティをオンにするには、次の手順を行います。

1. サーバーマネージャにアクセスし、ドロップダウンリストから待機ソケットが作成されるサーバーインスタンスを選択します。
2. まだ表示されていない場合には「Preferences」タブを選択します。
3. 「Edit Listen Sockets」リンクを選択します。

   「Edit Listen Sockets」ページが表示されます。
4. 「New」ボタンをクリックします。

   「Add Listen Socket」ページが表示されます。
5. 必要な情報を入力してから、デフォルトの仮想サーバーを選択します。
6. セキュリティをオンにするには、「Security」ドロップダウンリストから「Enabled」を選択します。
7. 「OK」をクリックします。
8. 「Apply」をクリックしてから、変更内容を有効にするため「Restart」をクリックします。

注	待機ソケットを作成したあとでセキュリティの設定を行うには、「Edit Listen Sockets」リンクを使用する必要があります。

待機ソケットの編集時にセキュリティ機能をオンにする

管理サーバーまたはサーバーマネージャのどちらかから待機ソケットを編集するときにも、セキュリティ機能をオンにできます。待機ソケットの編集時にセキュリティ機能をオンにするには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Security」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

2. まだ表示されていない場合には「Preferences」タブを選択します。
3. 「Edit Listen Sockets」リンクを選択します。

   「Edit Listen Sockets」ページが表示されます。

4. 待機ソケットを編集するには、編集する待機ソケットの「Listen Socket ID」をクリックします。

   「Edit Listen Socket」ページが表示されます。

5. 待機ソケットのセキュリティをオンにするには、「Security」ドロップダウンリストから「Enabled」を選択します。
6. 「OK」をクリックします。
7. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

待機ソケットのサーバー証明書の選択

管理サーバーまたはサーバーマネージャのどちらかで、ユーザーが要求しインストールしたサーバー証明書を使用するように待機ソケットを設定できます。

注	少なくとも 1 つは証明書をインストールしておく必要があります。

待機ソケットが使用するサーバー証明書を選択するには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Preferences」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

2. 「Edit Listen Sockets」リンクを選択します。

   「Edit Listen Sockets」ページが表示されます。

3. 待機ソケットを編集するには、編集する待機ソケットの「Listen Socket ID」をクリックします。

   「Edit Listen Socket」ページが表示されます。

4. 待機ソケットのセキュリティをオンにするには、「Security」ドロップダウンリストから「Enabled」を選択します。

   .

注	外部モジュールがインストールされている場合には、処理を続行する前に、外部モジュールのパスワードを入力するよう求める「Manage Server Certificates」ページが表示されます。

5. 「Server Certificate Name」ドロップタウンリストから待機ソケットのサーバー証明書を選択します。

   このリストには、インストールされているすべての内部および外部の証明書が記載されています。

注	サーバー証明書がインストールされていない場合は、それを警告するメッセージが「Server Certificate Name」ドロップダウンリストの場所に表示されます。

6. 「OK」をクリックします。
7. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

暗号化方式の選択

Web サーバーのセキュリティを保護するためには、SSL を有効にすることをお勧めします。SSL2.0、SSL3.0 および TLS 暗号化プロトコルを有効にして、各種の暗号化方式セットを選択することができます。管理サーバーの待機ソケットで、SSL および TLS を有効にできます。サーバーマネージャの待機ソケットで SSL と TLS を有効にすると、その待機ソケットに関連するすべての仮想サーバーに対して、これらのセキュリティの指定が設定されます。

仮想サーバーをセキュリティ保護されていない状態で使用するには、それらをすべてセキュリティ機能をオフにした同じ待機ソケットに設定する必要があります。

デフォルトの設定では、最も一般的に使用されている暗号化方式が使用できます。特定の暗号化方式セットを使用したくない特別な理由がある場合を除き、それらをすべて選択すべきです。特定の暗号化方式については、「Introduction to SSL」を参照してください。

注	少なくとも 1 つは証明書をインストールしておく必要があります。

tlsrollback パラメータの推奨される設定 (デフォルト設定) は true です。true に設定すると、人が介在するバージョンロールバック攻撃を検出するようにサーバーが設定されます。TLS 仕様を正しく実装できない一部のクライアントとの相互運用を確保するために、この値を false に設定しなければならない場合があります。

tlsrollback を false に設定すると、通信がバージョンロールバック攻撃を受けやすくなるので注意してください。バージョンロールバック攻撃は、第三者がクライアントおよびサーバーで SSLv2 などの古くてセキュリティ保護機能が低いプロトコルを使用して通信を行うようにするメカニズムです。SSLv2 プロトコルには既知の脆弱性があるため、バージョンロールバック攻撃を検出できない場合、第三者による暗号化された通信の傍受と復号化が容易になってしまいます。

SSL と TLS を有効にするには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Preferences」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

2. 「Edit Listen Sockets」リンクをクリックします。

   「Edit Listen Sockets」ページが表示されます。待機ソケットがセキュリティ保護されている場合は、「Edit Listen Sockets」ページに使用可能な暗号化方式セットが表示されます。

注 待機ソケットのセキュリティ機能が有効になっていない場合は、SSL と TLS の情報は表示されません。暗号化方式を使用するには、選択している待機ソケットでセキュリティ機能が有効になっている必要があります。詳細は、「待機ソケットのセキュリティの有効化」を参照してください。

3. 必要な暗号化方式セットに対応するチェックボックスにチェックマークを付けます。

注 Netscape Navigator 6.0 では、TLS と SSL3 の両方にチェックマークを付けます。「TLS Rollback」の場合にも、TLS にチェックマークを付け、SSL3 と SSL2 の両方が無効になっていることを必ず確認してください。

4. 「OK」をクリックします。

5. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

注	待機ソケットのセキュリティ機能をオンにしたあとで変更を適用するときには、セキュリティがオンであることを示すよう magnus.conf ファイルが自動的に変更され、その待機ソケットに関連するすべての仮想サーバーに自動的にデフォルトのセキュリティパラメータが割り当てられます。

サーバーで SSL が有効になったら、その URL には http の代わりに https が使用されます。SSL 有効サーバー上のドキュメントを示す URL の書式は次のとおりです。

https://*servername*.[*domain*.[*dom*]]:[*port#*]

例 ：https://admin.sun.com:443

デフォルトのセキュリティ保護された http ポート番号 (443) を使用する場合には、URL にポート番号を入力する必要はありません。

セキュリティのグローバルな設定

SSL 有効サーバーをインストールすると、グローバルセキュリティパラメータの指令エントリが、magnus.conf ファイル (サーバーのメイン設定ファイル) 内に作成されます。仮想サーバーのセキュリティ設定が機能するよう、セキュリティは「On」に設定しておく必要があります。仮想サーバーの SSL のプロパティは、server.xml ファイルの SSLPARAMS 要素内にサーバーごとに記述されています。

SSL 設定ファイル指令の値を設定するには、次の手順を行います。

1. サーバーマネージャにアクセスし、仮想サーバーのサーバーインスタンスをドロップダウンリストから選択します。
2. 設定する待機ソケットでセキュリティが有効になっていることを確認してください。これを行うには、次の手順を実行します。
   a. 「Edit Listen Sockets」リンクをクリックします。
   b. セキュリティを有効にする待機ソケットに対応する「Listen Socket ID」をクリックします。

      「Edit Listen Socket」ページが表示されます。
   c. 「Security」ドロップダウンリストから「Enabled」を選択します。
   d. 「OK」をクリックします。
3. 「Magnus Editor」リンクをクリックします。

4. ドロップダウンリストから「SSL Settings」を選択し、「Manage」をクリックします。
5. 次の項目の値を入力します。
   - SSLSessionTimeout
   - SSLCacheEntries
   - SSL3SessionTimeout
6. 「OK」をクリックします。
7. 「Apply」をクリックしてから、変更内容を有効にするため「Restart」をクリックします。

これらの SSL 設定ファイル指令について、次に説明します。

SSLSessionTimeout

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

構文

`SSLSessionTimeout` seconds

`seconds` は、キャッシュされた SSL 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 100 です。`SSLSessionTimeout` 指令が指定された場合には、秒数値は暗黙のうちに 5 〜 100 秒の間であると想定されます。

SSLCacheEntries

キャッシュできる SSL のセッション数を指定します。

SSL3SessionTimeout

SSL3SessionTimeout 指令は、SSL3 および TLS セッションのキャッシュを制御します。

構文

`SSL3SessionTimeout` seconds

`seconds` 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 秒間に制限されます。

外部暗号化モジュールの使用

Sun ONE Web Server 6.1 は、スマートカードやトークンリングなどの外部の暗号化モジュールとして、次の方法をサポートしています。

- PKCS#11
- FIPS-140

FIPS-140 暗号化標準を有効化する前に、PKCS#11 モジュールを追加しておく必要があります。

PKCS#11 モジュールのインストール

Sun ONE Web Server は、PKCS (Public Key Cryptography Standard) #11 をサポートします。この標準は、SSL と PKCS#11 モジュール間の通信に使用されるインタフェースを定義します。PKCS#11 モジュールは、SSL ハードウェアアクセラレータへの標準ベースの接続に使用されます。外部のハードウェアアクセラレータにインポートされた証明書と鍵は、`secmod.db` ファイルに格納されます。このファイルは、PKCS#11 モジュールをインストールしたときに生成されます。

modutil による PKCS#11 モジュールのインストール

PKCS#11 モジュールを、`modutil` ツールを使用して `.jar` ファイルまたはオブジェクトファイルの形式でインストールできます。

`modutil` を使用して PKCS#11 モジュールをインストールするには、次の手順を行います。

1. 管理サーバーを含むすべてのサーバーが停止していることを確認します。
2. データベースが置かれている `server_root/alias` ディレクトリに移動します。
3. PATH に `server_root/bin/https/admin/bin` を追加します。
4. `server_root/bin/https/admin/bin` で `modutil` を特定します。
5. 環境を設定します。その例を次に示します。
   - UNIX では、`setenv`

     ```
     LD_LIBRARY_PATH server_root/bin/https/lib:${LD_LIBRARY_PATH}
     ```
   - IBM-AIX では、`LIBPATH`
   - HP-UX では、`SHLIB_PATH`
   - Windows では、`PATH` に次を追加します。

LD_LIBRARY_PATH server_root/bin/https/bin

使用しているマシンの PATH は、以下で参照できます。
server_root/https-admin/start.

6. 次のコマンドを入力します : modutil

   オプションの一覧が表示されます。

7. 必要な操作を行います。

   たとえば、UNIX に PCKS#11 モジュールを追加する場合には、次のように入力します。

   modutil -add (PCKS#11 ファイルの名前) -libfile (PCKS#11 用のユーザーの libfile) -nocertdb -dbdir (db ディレクトリ)

pk12util の使用

pk12util 使用して、内部データベースから証明書と鍵をエクスポートしたり、内部または外部の PKCS#11 モジュールにこれらをインポートすることができます。証明書と鍵は内部データベースにいつでもエクスポートできますが、ほとんどの外部トークンでは証明書と鍵のエクスポートは許可されません。デフォルトでは、pk12util は、cert8.db と key3.db という名前の証明書と鍵データベースを使用します。

pk12util によるエクスポート

内部データベースから証明書と鍵をエクスポートするには、次の手順を行います。

1. データベースが置かれている server_root/alias ディレクトリに移動します。
2. PATH に server_root/bin/https/admin/bin を追加します。
3. server_root/bin/https/admin/bin で pk12util を特定します。
4. 環境を設定します。その例を次に示します。
   - UNIX では、
     setenv LD_LIBRARY_PATH server_root/bin/https/lib:${LD_LIBRARY_PATH}
   - IBM-AIX では、LIBPATH
   - HP-UX では、SHLIB_PATH
   - Windows では、PATH に次を追加します。

     LD_LIBRARY_PATH server_root/bin/https/bin

     使用しているマシンの PATH は、以下で参照できます。
     server_root/https-admin/start.
5. 次のコマンドを入力します : pk12util

   オプションの一覧が表示されます。

6. 必要な操作を行います。

   たとえば、UNIX では次のように入力します。

   ```
   pk12util -o certpk12 -n Server-Cert [-d /server/alias] [-P
   https-test-host]
   ```

7. データベースパスワードを入力します。
8. pkcs12 パスワードを入力します。

pk12util によるインポート

内部または外部の PKCS#11 モジュールに証明書と鍵をインポートするには、次の手順を行います。

1. データベースが置かれている server_root/alias ディレクトリに移動します。
2. PATH に server_root/bin/https/admin/bin を追加します。
3. server_root/bin/https/admin/bin で pk12util を特定します。
4. 環境を設定します。その例を次に示します。
   - UNIX では、
     setenv LD_LIBRARY_PATH server_root/bin/https/lib:${LD_LIBRARY
     _PATH}
   - IBM-AIX では、LIBPATH
   - HP-UX では、SHLIB_PATH
   - Windows では、PATH に次を追加します。

     LD_LIBRARY_PATH server_root/bin/https/bin

     使用しているマシンの PATH は、以下で参照できます。
     server_root/https-admin/start
5. 次のコマンドを入力します : pk12util

   オプションの一覧が表示されます。
6. 必要な操作を行います。

   たとえば、UNIX では次のように入力します。

   ```
   pk12util -i pk12_sunspot [-d certdir][-h "nCipher"][-P
   https-jones.redplanet.com-jones-]
   ```

   -P は -h のあとに、最後の引数として使用します。

   引用符記号の中の大文字と空白文字を含む、正確なトークン名を入力します。
7. データベースパスワードを入力します。
8. pkcs12 パスワードを入力します。

外部証明書を使用してサーバーを起動するには : 外部 PKCS#11 モジュール (たとえば、ハードウェアアクセラレータなど) にサーバーの証明書をインストールする場合には、`server.xml` を編集するか、または次に述べるように、証明書名を指定するまで、サーバーはその証明書の使用を開始できません。

サーバーは常に、「Server-Cert」という名前の証明書を使用して起動しようとします。しかし、外部 PKCS#11 モジュール内の証明書には、識別子内にモジュールのトークン名のうちの 1 つが含まれています。たとえば、「smartcard0」と呼ばれる外部スマートカードリーダー上にインストールされているサーバー証明書の名前が「smartcard0:Server-Cert」となるなどです。

外部モジュールにインストールされている証明書を使用してサーバーを起動するには、稼動する待機ソケットの証明書名を指定する必要があります。

待機ソケットの証明書名の選択

待機ソケットの証明書名を選択するには、次の手順を行います。

注 待機ソケットのセキュリティ機能が有効になっていない場合は、証明書情報は表示されません。待機ソケットの証明書名を選択するには、最初にセキュリティ機能が有効になっている必要があります。詳細は、「待機ソケットのセキュリティの有効化」を参照してください。

---

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Preferences」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

2. まだ選択されていない場合には「Preferences」タブを選択します。
3. 「Edit Listen Sockets」リンクをクリックします。

   「Edit Listen Sockets」ページが表示されます。

4. 証明書と関連付ける待機ソケットに対応する「Listen Socket Id」リンクをクリックします。

   「Edit Listen Socket」ページが表示されます。

5. 「Server Certificate Name」ドロップタウンリストから待機ソケットのサーバー証明書を選択します。

   このリストには、インストールされているすべての内部および外部の証明書が記載されています。

注	サーバー証明書がインストールされていない場合は、それを警告するメッセージが「Server Certificate Name」ドロップダウンリストの場所に表示されます。

6. 「OK」をクリックします。
7. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

手動で `server.xml` ファイルを編集することにより、代わりにそのサーバー証明書を使用して起動することをサーバーに指示することもできます。SSLPARAMS の `servercertnickname` 属性を次のように変更します。

```
$TOKENNAME:Server-Cert
```

`$TOKENNAME` に使用する値を知るには、サーバーの「Security」タブに移動して、「Manage Certificates」リンクを選択します。Server-Cert の格納されている外部モジュールにログインすると、「`$TOKENNAME:$NICKNAME`」フォームのリスト内にその証明書が表示されます。

注	信頼データベースを作成していない場合には、外部 PKCS#11 モジュールの証明書を要求するかまたはインストールすると信頼データベースが 1 つ作成されます。作成されるデフォルトのデータベースには、パスワードがないためアクセスできません。外部モジュールは動作しますが、サーバー証明書を要求してインストールすることはできません。パスワードのないデフォルトのデータベースが作成された場合には、「Security」タブの「Create Database」ページを使用してパスワードを設定してください。

FIPS-140 標準

PKCS#11 API を使用すれば、暗号化操作を実行するソフトウェアまたはハードウェアモジュールとの通信が可能です。PKCS#11 をサーバー上にインストールすると、Federal Information Processing Standards (FIPS) - 140 に準拠するよう Sun ONE Web Server を設定できます。これらのライブラリは、SSL バージョン 3.0 にのみ含まれています。

FIPS-140 を有効にするには、次の手順を行います。

1. FIPS-140 の指示に従ってプラグインをインストールします。
2. 管理サーバーまたはサーバーマネージャにアクセスし、「Preferences」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。
3. 「Edit Listen Sockets」リンクをクリックします。

   「Edit Listen Sockets」ページが表示されます。待機ソケットがセキュリティ保護されている場合は、「Edit Listen Sockets」ページに使用可能なセキュリティ設定が表示されます。

注 FIPS-140 を使用するには、選択している待機ソケットでセキュリティが有効になっている必要があります。詳細は、「待機ソケットのセキュリティの有効化」を参照してください。

---

4. 「Enabled」が選択されていない場合は、「SSL Version 3」ドロップダウンリストから選択します。
5. 次のうち、適切な FIPS-140 暗号化方式のセットを選択します。
   - (FIPS) DES with 56 bit encryption and SHA message authentication
   - (FIPS) Triple DES with 168 bit encryption and SHA message authentication
6. 「OK」をクリックします。
7. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

クライアントセキュリティ要件の設定

サーバーをセキュリティ保護するためのすべての手順が終了したあと、クライアントに関するその他のセキュリティ要件を設定できます。

クライアント認証の要求

管理サーバーと各サーバーインスタンスの待機ソケットが、クライアント認証を要求できるようになります。クライアント認証を有効にすると、クエリに対してサーバーが応答を送信する前に、クライアントの証明書が必要となります。

Sun ONE Web Server は、クライアント認証に含まれる CA と、信頼できる署名済みクライアント証明書を比較することによってクライアント証明書の認証をサポートします。管理サーバーの「Security」の下にある「Manage Certificates」ページで、クライアント証明書に署名済みの信頼できる CA のリストを参照できます。CA には、次の 4 種類があります。

- Untrusted CA (一致しない)
- Trusted Server CA (一致しない)
- Trusted Server CA (一致する)
- Trusted Client/Server CA (一致する)

信頼できる CA からのクライアント証明書を持っていないクライアントを拒絶するよう Web サーバーを設定できます。信頼できる CA を受け入れるまたは拒絶するには、その CA についてクライアント信頼情報を設定しておく必要があります。詳細は、122 ページの「証明書の管理」を参照してください。

Sun ONE Web Server は、エラーの記録、証明書の拒絶、また証明書が期限切れの場合にはクライアントに対してメッセージの返送を行います。また、管理サーバーの「Manage Certificates」ページで、有効期限切れの証明書を参照できます。

クライアントの証明書から情報を収集し、これを LDAP ディレクトリ内のユーザーエントリと照合するようにサーバーを設定できます。このようにすると確実に、LDAP ディレクトリ内に有効な証明書とエントリをクライアントが持つことを確認できます。また、クライアント証明書が LDAP ディレクトリ内の証明書と確実に一致することを確認できます。これを実行する方法については、142 ページの「LDAP へのクライアント証明書のマッピング」を参照してください。

証明書のあるユーザーは、信頼できる CA だけでなく、アクセス制御の規則 (ACL) とも一致しなければならないように、クライアント証明書をアクセス制御と組み合わせることができます。詳細は、190 ページの「アクセス制御ファイルの使用」を参照してください。

クライアントの証明書からの情報も処理することができます。詳細については、『Sun ONE Web Server 6.1 NSAPI Programmer's Guide』を参照してください。

クライアントの認証を要求するには

クライアントの認証を要求するには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Preferences」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。

2. 「Edit Listen Sockets」リンクをクリックします。

   「Edit Listen Sockets」ページが表示されます。

3. クライアント認証を要求する待機ソケットに対応する「Listen Socket Id」リンクをクリックします。

   「Edit Listen Socket」ページが表示されます。

4. 待機ソケットのクライアント認証を要求するには、「Client Authentication」ドロップダウンリストから「Required」を選択します。

5. 「OK」をクリックします。

6. サーバーマネージャを使用する場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

注 現在、Web サーバーインスタンスごとに 1 つの証明書信頼データベースが存在します。そのサーバーインスタンスのもとで稼動しているセキュリティ保護された仮想サーバーはすべて、信頼できるクライアント CA の同じリストを共有します。2 つの仮想サーバーが別の信頼できる CA を要求する場合には、これらの仮想サーバーは、別個の信頼データベースを使用して、異なるサーバーインスタンスで稼動する必要があります。

LDAP へのクライアント証明書のマッピング

この節では、Sun ONE Web Server が LDAP ディレクトリ内のエントリにクライアント証明書をマップするために使用するプロセスを説明します。

サーバーがクライアントから要求を取得すると、処理を進める前にクライアントの証明書を求めます。一部のクライアントは、要求と一緒にクライアント証明書をサーバーに送信します。

注	LDAP にクライアント証明書をマップする前に、必要な ACL も設定しておく必要があります。詳細は、第 9 章「サーバーへのアクセス制御」を参照してください。

サーバーは、管理サーバーの信頼できる CA リストとその証明書の発行元である CA を照合します。一致しなかった場合には、Sun ONE Web Server はその接続を終了します。一致した場合、サーバーは要求の処理を続行します。

証明書が信頼できる CA からのものであることを確認したあと、サーバーは、次の方法で LDAP エントリにその証明書をマップします。

- クライアント証明書の発行者と対象 DN を LDAP ディレクトリ内の分岐点にマップします。
- クライアント証明書の対象 (エンドユーザー) に関する情報と一致するエントリがないか LDAP ディレクトリを検索します。
- (省略可能) DN に対応する LDAP エントリ内のクライアント証明書とそのクライアント証明書を比較検証します。

サーバーは、`certmap.conf` と呼ばれる証明書マッピングファイルを使用して LDAP 検索を実行する方法を決定します。このマッピングファイルは、クライアント証明書から入手すべき値 (エンドユーザー名、電子メールアドレスなど) をサーバーに通知します。サーバーは、これらの値を使用して LDAP ディレクトリ内にユーザーエントリがないか検索しますが、はじめに、LDAP ディレクトリ内のどこから検索を開始すべきかを決定する必要があります。このような開始すべき場所も、証明書マッピングファイルがサーバーに通知します。

サーバーが、検索を開始する場所および検索すべき内容を確認すると (手順 1)、LDAP ディレクトリ内で検索を実行します (手順 2)。一致するエントリがなかったり、一致するエントリがあってもマッピングが証明書を検証するように設定されていない場合には、検索は失敗します。検索結果についての予期される動作のリストは、次の表 6-1 を参照してください。予期される動作を ACL で指定できることに注意してください。たとえば、証明書照合が失敗した場合は Sun ONE Web Server がユーザー自身だけを受け入れるよう指定することができます。ACL の詳細設定については、190 ページの「アクセス制御ファイルの使用」を参照してください。

表 6-1　LDAP 検索結果

LDAP 検索結果	証明書の比較検証が有効 (ON)	証明書の比較検証が無効 (OFF)
検出されたエントリなし	認証失敗	認証失敗
検出されたエントリが 1 つのみ	認証失敗	認証成功
検出されたエントリが複数	認証失敗	認証失敗

サーバーが LDAP ディレクトリ内で一致するエントリと証明書を検出したあと、その情報を使用してトランザクションを処理できます。たとえば、一部のサーバーでは、サーバーへのアクセスを判断するのに証明書 - LDAP 間マップを使用します。

certmap.conf ファイルの使用

証明書のマッピングは、LDAP ディレクトリ内のユーザーエントリをサーバーがどのように検索するか決定します。`certmap.conf` を使用して、名前で指定された証明書を LDAP エントリにマップする方法を設定できます。このファイルを編集し、エントリを追加して、ユーザーの LDAP ディレクトリの組織に一致するようにし、ユーザーが持っているべき証明書をリストにするようにします。ユーザーは、`subjectDN` 内で使用されているユーザー ID、電子メールアドレス、またはその他の値に基づいて認証されることができます。特に、マッピングファイルは、次の情報を定義します。

- LDAP ツリー内でサーバーが検索を開始する場所
- LDAP ディレクトリ内のエントリを検索するときにサーバーが検索条件として使用すべき証明書の属性
- サーバーが追加の検証プロセスを実施するか、または実施しないか

証明書マッピングファイルは、次の場所に格納されています。

server_root`/userdb/certmap.conf`

このファイルには、それぞれが異なる CA に適用される、1 つまたは複数の名前付きのマッピングが格納されています。マッピングの構文は、次のとおりです。

```
certmap <name> <issuerDN>

<name>:<property> [<value>]
```

最初の行にはエントリの名前と、CA 証明書内に記載されている識別名を設定する属性を指定します。名前は任意です。好きな名前に定義できます。ただし、issuerDN は、そのクライアント証明書を発行した CA の発行者 DN と正確に一致している必要があります。たとえば、次の 2 つの issuerDN 行は、属性間に空白文字があるかどうかという点が異なるだけですが、サーバーは、これら 2 つのエントリを別のものとして取り扱います。

```
certmap sun1 ou=Sun Certificate Authority,o=Sun,c=US
certmap sun2 ou=Sun Certificate Authority,o=Sun, c=US
```

ヒント Sun ONE Directory Server を使用しているときに issuerDN 照合に問題があった場合は、Directory Server のエラーログを調べて有用な情報を探します。

名前付きマッピングの 2 行目以降の行は、プロパティが値と照合されます。certmap.conf ファイルには、次に示す 6 つのデフォルトのプロパティがあります (証明書 API を使用すると、ユーザー独自のプロパティをカスタマイズできます)。

- DNComps はコンマで区切った属性のリストで、ユーザーの情報 (すなわちクライアント証明書の所有者) と一致するエントリの検索を、サーバーが LDAP ディレクトリ内のどこから開始すべきかを判断するのに使用されます。サーバーは、クライアント証明書からこれらの属性の値を収集し、LDAP DN を設定するためにその値を使用します。これが、LDAP ディレクトリ内でサーバーが検索を開始する場所を決定します。たとえば、DN の o 属性と c 属性を使用するよう DNComps を設定した場合、サーバーは、LDAP ディレクトリ内の o=<org>, c=<country> エントリから検索を開始します。この場合、<org> と <country> は、証明書内の DN に記述されている値と置き換えられます。

  次のような場合には注意が必要です。

  - マッピング内に DNComps エントリがない場合、サーバーは CmapLdapAttr の設定、またはクライアント証明書内の対象 DN 全体 (すなわちエンドユーザー情報) のいずれかを使用します。
  - DNComps エントリはあるが値がないという場合、サーバーは LDAP ツリー全体でフィルタに一致するエントリを検索します。

- FilterComps は、コンマで区切った属性のリストで、クライアント証明書内のユーザーの DN から情報を収集してフィルタを作成するのに使用されます。サーバーは、これらの属性の値を使用して、LDAP ディレクトリ内でエントリを照合するのに使用する検索条件を作成します。サーバーが LDAP ディレクトリ内で、証明書から収集したユーザー情報に一致する 1 つまたは複数のエントリを検出した場合、検索は成功し、オプションでサーバーが検証を行います。

たとえば、電子メール属性と ユーザー ID 属性を使用するよう FilterComps を設定すると (FilterComps=e,uid)、電子メールとユーザー ID の値がクライアント証明書から収集したエンドユーザーの情報と一致するエントリを、サーバーがディレクトリ内で検索します。電子メールアドレスと ユーザー ID は、通常はディレクトリ内で一意のエントリであるため、フィルタとして適切なものです。フィルタは、LDAP データベース内で 1 つだけのエントリと一致するような特有のものである必要があります。

x509v3 証明書属性のリストについては、次の表を参照してください。

表 6-2　x509v3 証明書の属性

属性	説明
c	国
o	組織
cn	共通名
l	場所
st	州
ou	組織単位
uid	UNIX/Linux ユーザー ID
email	電子メールアドレス

フィルタのための属性名は、LDAP ディレクトリではなく、証明書から取得した属性名にする必要があります。たとえば、一部の証明書ではユーザーの電子メールアドレスの e 属性がありますが、LDAP は、この属性を mail と呼んでいることもあります。

- verifycert は、LDAP 内でにある証明書とクライアントの証明書を比較すべきかどうかをサーバーに指示します。これは、2 つの値のいずれかをとります。すなわちオン、またはオフです。ただし、このプロパティは、LDAP ディレクトリに証明書があるときだけ使用してください。この機能は、有効であり、取り消されていない証明書を確実にエンドユーザーが所有できるようにするのに便利です。
- CmapLdapAttr は、LDAP ディレクトリ内の属性の名前で、そのユーザーに属しているすべての証明書に記載されている対象 DN が格納されています。このプロパティのデフォルトは、certSubjectDN です。この属性は標準の LDAP 属性ではないため、このプロパティを使用するときには、LDAP スキーマを拡張する必要があります。詳細は、「Introduction to SSL」を参照してください。

このプロパティが certmap.conf ファイル内に存在する場合は、対象の完全な DN (証明書から取得) に 属性 (このプロパティの名前の付いた) が一致しているエントリを、サーバーが LDAP ディレクトリ全体で検索します。エントリが検出されなかった場合には、サーバーは DNComps マッピングと FilterComps マッピングを使用して、再度検索します。

LDAP エントリと証明書を照合するためのこの方法は、DNComps と FilterComps を使用してエントリを照合することが難しい場合に便利です。

- Library は、値が共用ライブラリまたは DLL へのパス名であるプロパティです。証明書 API を使用して、独自のプロパティを作成する場合には、このプロパティを使用するだけですみます。詳細は、『NSAPI Programmer's Guide』を参照してください。
- InitFn は、値がカスタムライブラリの init 関数の名前であるプロパティです。証明書 API を使用して、独自のプロパティを作成する場合には、このプロパティを使用するだけですみます。

これらプロパティについては、146 ページの「マッピング例」に記述されている例を参照してください。

カスタムプロパティの作成

クライアント証明書 API を使用すると、独自のプロパティを作成できます。クライアント証明書 API のプログラミング法と使用法については、『NSAPI Programmer's Guide』を参照してください。

カスタムマッピングが行われると、次のようにマッピングを参照します。

```
<name>:library <path_to_shared_library>
<name>:InitFn <name_of_init_function>
```

その例を次に示します。

```
certmap default1 o=Sun Microsystems, c=US
default1:library /usr/sun/userdb/plugin.so
default1:InitFn plugin_init_fn
default1:DNComps  ou o c
default1:FilterComps l
default1:verifycert on
```

マッピング例

certmap.conf ファイルには、少なくとも 1 つのエントリが必要です。次の例では、certmap.conf ファイルを使用できる別の方法を示しています。

例 1

この例は、デフォルトのマッピングが 1 つだけある certmap.conf ファイルを表わしています。

```
certmap default default
default:DNComps ou, o, c
default:FilterComps e, uid
default:verifycert on
```

この例を使用すると、ou=<orgunit>, o=<org>, c=<country> エントリを格納している LDAP 分岐点からサーバーは検索を開始します。< > 内のテキストは、クライアント証明書内の対象 DN に記載されている値と置き換えられます。

次に、サーバーが証明書に記載されている電子メールアドレスとユーザー ID の値を使用して、LDAP ディレクトリ内で一致するエントリを検索します。エントリを検出すると、サーバーは、ディレクトリ内に格納されているエントリとクライアントが送信したエントリを比較して、証明書を検証します。

例 2

次のファイル例には、2 つのマッピングがあります。1 つはデフォルト用で、もう 1 つは US Postal Service 用です。

```
certmap default default
default:DNComps
default:FilterComps e, uid

certmap usps ou=United States Postal Service, o=usps, c=US
usps:DNComps ou,o,c
usps:FilterComps e
usps:verifycert on
```

サーバーが US Postal Service 以外から証明書を取得している場合、サーバーはデフォルトのマッピングを使用します。これは、LDAP ツリーの一番上から、クライアントの電子メールとユーザー ID に一致するエントリの検索を開始します。その証明書が US Postal Service からのものである場合、サーバーは、その組織単位を格納している LDAP 分岐から、一致する電子メールアドレスの検索を開始します。また、その証明書が USPS (US Postal Service) からのものである場合には、サーバーは証明書の検証を行います。それ以外の証明書は検証されません。

警告 証明書内の発行者 DN (すなわち CA の情報) は、マッピングの最初の行に記述されている発行者 DN と同じでなくてはなりません。前述の例では、o=United States Postal Service,c=US という発行者 DN からの証明書は、o 属性と c 属性の間に空白文字がないため一致しません。

例 3

次の例では、CmapLdapAttr プロパティを使用して、クライアント証明書から取得された対象 DN 全体と同一の値をもつ certSubjectDN という属性を、LDAP データベース内で検索します。

```
certmap myco ou=My Company Inc, o=myco, c=US
myco:CmapLdapAttr certSubjectDN
myco:DNComps  o, c
myco:FilterComps mail, uid
myco:verifycert on
```

クライアント証明書の対象が次の場合には、

```
uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

サーバーは、はじめに次の情報を格納しているエントリを検索します。

```
certSubjectDN=uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

1 つまたは複数の一致したエントリが検出された場合、サーバーはそのエントリの検証処理を進めます。一致するエントリが検出されなかった場合には、サーバーは、DNComps と FilterComps を使用して、一致するエントリを検索します。この例では、サーバーは、o=LeavesOfGrass Inc, c=US の下にあるすべてのエントリで uid=Walt Whitman を検索します。

注 この例では、LDAP ディレクトリに certSubjectDN 属性のあるエントリが格納されていると想定しています。

Stronger Ciphers の設定

「Stronger Ciphers」オプションでは、アクセスするための非公開鍵のサイズに 168、128 または 56 ビットのいずれか、または制限なしの選択肢があります。制限に適合しない場合に使用されるファイルを指定することができます。ファイルが指定されていない場合は、Sun ONE Web Server が、「Forbidden」ステータスを返します。

アクセスのための鍵サイズとして、「Security Preferences」の下にある現在の暗号化方式の設定と整合しないサイズを選択すると、Sun ONE Web Server が、暗号化方式でより大きいサイズの非公開鍵を利用可能にする必要があると知らせるポップアップダイアログを表示します。

現在、鍵サイズ制限の実装は、Service fn=key-toosmall ではなく、obj.conf にある NSAPI PathCheck 指令に基づいています。この指令は次のとおりです。

```
PathCheck fn="ssl-check" [secret-keysize=<nbits>]
[bong-file=<filename>]
```

ここで、<nbits> は、非公開鍵で必要とされる最小ビット数で、<filename> は、制限に適合しない場合に使用されるファイル (URI ではなく) の名前です。

SSL が有効ではない場合、または secret-keysize パラメータが指定されていない場合には、PathCheck は REQ_NOACTION を返します。現在のセッションの非公開鍵サイズが指定された secret-keysize より小さいときは、関数は、bong-file が指定されていない場合には PROTOCOL_FORBIDDEN のスターテスと一緒に REQ_ABORTED を返し、それ以外の場合には REQ_PROCEED を返して、「path」変数が bong-file <filename> に設定されます。また、鍵のサイズ制限に適合しない場合は、現在のセッションの SSL セッションキャッシュエントリが無効化されるため、次回、同じクライアントがサーバーに接続するとき完全な SSL ハンドシェイクが起こります。

注 「Stronger Ciphers」フォームは、PathCheck fn=ssl-check を追加するときにオブジェクト内で検出する Service fn=key-toosmall 指令を削除します。

「Stronger Ciphers」を設定するには、次の手順を行います。

1. サーバーマネージャにアクセスし、サーバーのサーバーインスタンスをドロップダウンリストから選択します。
2. 「Virtual Server Class」タブをクリックします。
3. クラスをドロップダウンリストから選択し、「Manage」をクリックします。

   「Class Manager」ページが表示されます。
4. 「Content Mgmt」タブを選択します。
5. 「Stronger Ciphers」を選択します。
6. 編集項目を選択します。
   - ドロップダウンリストから
   - 「Browse」をクリックして
   - 「Wildcard」をクリックして
7. 非公開鍵サイズの制限を選択します。
   - 168 bit or larger (168 ビットまたはそれ以上)
   - 128 bit or larger (128 ビットまたはそれ以上)
   - 56 bit or larger (56 ビットまたはそれ以上)
   - No restrictions (制限なし)
8. アクセスを拒絶するメッセージのファイルの場所を入力します。
9. 「OK」をクリックします。
10. 「Apply」をクリックします。

11.「hard start/restart」または「dynamically apply」を選択します。

詳細は、「Introduction to SSL」を参照してください。

セキュリティに関するその他の問題

他人が暗号を解読しようとすること以外にも、セキュリティに関するリスクがあります。ネットワークは常に、内部と外部の両側から、ハッカーのリスクにさらされています。ハッカーはさまざまな作戦を使って、サーバー本体やサーバーに格納されている情報にアクセスしようとしています。

したがって、サーバーで暗号化を有効にするだけでなく、さらに別のセキュリティの対策を立てる必要があります。たとえば、セキュリティ保護された部屋にサーバーマシンを設置し、信頼できない個人にサーバーへのプログラムのアップロードを許可しないようにするなどです。

次の各節では、サーバーをさらに安全に保護するのに必要な、最も重要な事項について説明します。

- 物理的アクセスを制限する
- 管理アクセスを制限する
- 確実なパスワードを選択する
- パスワードまたは PIN を変更する
- サーバー上で他のアプリケーションを制限する
- クライアントによる SSL ファイルのキャッシュを防ぐ
- ポートを制限する
- サーバーの限界を知る
- サーバーを保護するためその他の追加変更を行う

物理的アクセスの制限

このシンプルなセキュリティ手段が、意外と見逃されがちです。この方法では、承認された人だけが入室できる鍵の掛かった部屋にサーバーマシンを設置します。このようにすると、サーバーマシンへの物理的なハッキングを防げます。

また、マシンの管理 (root) パスワードを所有している場合には、パスワードを保護しておく必要があります。

管理アクセスの制限

リモート構成を使用している場合、必ず、数人のユーザーと数台のコンピュータだけが管理作業を実行できるように、アクセス制御を設定します。管理サーバーからエンドユーザーの権限で LDAP サーバーやローカルディレクトリにアクセスさせる場合、2 台の管理サーバーでクラスタ設定を整え、1 台では SSL を有効にしてマスターとなる管理サーバーを設定し、もう 1 台のサーバーでエンドユーザーからのアクセスを許可することを、検討してください。

クラスタについては、157 ページの「クラスタについて」を参照してください。

管理サーバーの暗号化機能もオンにする必要があります。管理に SSL 接続を使用しない場合、リモートサーバーの管理にセキュリティ保護されていないネットワークを使用することになるという点に注意してください。つまり、SSL を使用しない場合には、通信の途中で管理パスワードを盗まれて、サーバーが不正に設定されてしまう可能性があるということです。

確実なパスワードの選択

サーバーでは多数のパスワードが使用されています。管理パスワード、非公開鍵パスワード、データベースパスワードなどです。この中でもっとも重要なパスワードは管理パスワードです。このパスワードを使えば、誰もがコンピュータ上のどのサーバーの設定でも行えるからです。次に重要なのは、非公開鍵パスワードです。非公開鍵と非公開鍵パスワードを第三者が入手した場合、その第三者が偽のサーバーを作成したり、ユーザー自身のものであるように見せかけたり、サーバーとの間の通信を傍受したり、改ざんしたりする可能性があります。

優れたパスワードは、ユーザー自身がすぐに思い出せて、第三者には推測できないようなパスワードです。たとえば、「My Child is 12 months old!」からは *MCi12!mo* を思い出すことができます。悪いパスワードは、たとえば子供の名前や誕生日などです。

推測しにくいパスワードの作成

安全性の高いパスワードを作成するのに役立つ、いくつかの簡単な注意事項を示します。

1 つのパスワードに次の規則のすべてを取り込む必要はありませんが、使用する規則が多ければ多いほど、パスワードは推測されにくくなります。

- パスワードの長さは 6 ～ 14 文字にする (Mac のパスワードは 8 文字まで)
- 正規以外の文字は使用しない (*、"、空白文字)
- 辞書に載っている語を使用しない (どの言語でも)
- E を 3 にする、L を 1 にする、などの推測しやすい文字の置き換えは行わない

- 以下の種類の文字を、できるだけ多く混合させる
  - 大文字
  - 小文字
  - 数字
  - 記号

パスワードまたは PIN の変更

信頼データベース / 鍵ペアファイルのパスワードまたは PIN を定期的に変更することを習慣付けてください。管理サーバーで SSL を有効にしている場合、サーバーを起動するときにこのパスワードが必要です。パスワードを定期的に変更すると、サーバーのセキュリティ保護が強化されます。

このパスワードは、ローカルマシンにおいてのみ変更すべきです。パスワードを変更するときに考慮すべき注意事項のリストは、151 ページの「推測しにくいパスワードの作成」を参照してください。

パスワードの変更

管理サーバーまたはサーバーインスタンスの信頼データベース / 鍵ペアファイルのパスワードを変更するには、次の手順を行います。

1. 管理サーバーまたはサーバーマネージャにアクセスし、「Security」タブを選択します。

   サーバーマネージャを使用する場合には、はじめにドロップダウンリストからサーバーインスタンスを選択する必要があります。
2. 「Change Password」リンクを選択します。
3. パスワードを変更したいセキュリティトークンをドロップダウンリストから選択します。

   デフォルトでは、内部鍵データベース用の「internal」になっています。PKCS#11 モジュールがインストールされている場合は、すべてのトークンが一覧で表示されます。「Change Password」リンクをクリックしてください。
4. 現在のパスワードを入力します。
5. 新しいパスワードを入力します。
6. 新しいパスワードをもう一度入力します。
7. 「OK」をクリックします。
8. サーバーマネージャの場合には、「Apply」をクリックし、変更内容を有効にするため「Restart」をクリックします。

鍵ペアファイルは必ずセキュリティ保護されるようにします。管理サーバーは、*server_root*/alias ディレクトリ内に鍵ペアファイルを格納します。コンピュータ上にインストールされている Sun ONE Web Server だけがファイルとディレクトリを読めるようにすることを検討してください。

ファイルがバックアップテープ上に格納されるかどうか、またはそのファイルが第三者が傍受できるような状態かどうかを知っておくことも大切です。そのような場合には、バックアップをサーバーと同等に、完全にセキュリティ保護する必要があります。

サーバー上での他のアプリケーションの制限

サーバーと同じマシンで稼動するすべてのアプリケーションを十分に検討します。サーバー上で稼動する他のアプリケーションのセキュリティホールを使って、サーバーのセキュリティが保護されなくなる可能性があります。不必要なプログラムやサービスはすべて無効にしてください。たとえば、UNIX sendmail デーモンは、安全に設定することが難しく、他のプログラムがサーバーマシン上で稼動するようにプログラムされてしまう可能性もあります。

UNIX と Linux

inittab スクリプトと rc スクリプトから開始するプロセスを注意して選択します。telnet または rlogin をサーバーマシン上で起動させないでください。また、サーバーマシン上に rdist を格納すべきではありません。格納すると、ファイルを分散することができる反面、サーバーマシン上のファイルの更新に使用されてしまう可能性もあります。

Windows

他のマシンと共有するドライブやディレクトリについて、十分に検討してください。また、どのユーザーがアカウントやゲストの特権を所有しているかについても検討してください。

同様に、管理者がサーバー上に置いているプログラムや、サーバー上で他のユーザーにインストールを許可するプログラムについても注意を払ってください。他のユーザーのプログラムには、セキュリティホールがあるかもしれません。最悪の場合には、セキュリティを侵害するために設計された悪意のあるプログラムをアップロードする人がいるかもしれません。したがって、サーバー上にプログラムを置くことを許可する前に、そのプログラムを良く調べてください。

クライアントによる SSL ファイルのキャッシングを防ぐ

HTML 形式のファイルの <HEAD> タグの部分に次の行を追加することで、暗号化されているファイルをクライアントがキャッシュに書き込むのを前もって防止することができます。

```
<meta http-equiv="pragma" content="no-cache">
```

ポートの制限

マシン上で使用していないポートは、すべて無効にします。ルータやファイアウォールの設定を使用して、最少数のポートセット以外には着信接続ができないように設定します。このように設定すると、マシン上でシェルを取得する方法が、サーバーマシンを物理的に使用することだけになり、制限された領域内にしか接続できないことになります。

サーバーの限界を知る

サーバーは、サーバーとクライアントの間にセキュリティ保護された接続を提供します。サーバーは、情報を一度クライアントに取得されると、情報のセキュリティを制御することはできず、サーバーマシン自体およびディレクトリやファイルに対するアクセスも制御できません。

このような限界を認識しておくことは、避けるべき状況を理解するのに役立ちます。たとえば、SSL 接続を通じてクレジットカードの番号を取得するとして、そのような番号をサーバーマシン上のセキュリティ保護されたファイルに保存できるでしょうか。SSL 接続が終了したあとでこれらの番号に何が起こるのでしょうか。SSL を通じてクライアントが送信してきた情報に対しては、セキュリティ保護する責任があります。

サーバーを保護するためのその他の追加変更

保護されているサーバーと保護されていないサーバーの両方が必要な場合には、保護されているサーバーとは異なるマシンで保護されていないサーバーを運用する必要があります。リソースが限られており、保護されているサーバーと同じマシン上で保護されていないサーバーを稼動しなくてならない場合には、次のようにしてください。

- 適切なポート番号を割り当てます。保護されているサーバーと保護されていないサーバーに、必ず異なるポート番号が割り当てられる必要があります。登録されているデフォルトのポート番号は、次のとおりです。
  - 443 (保護されているサーバー用)

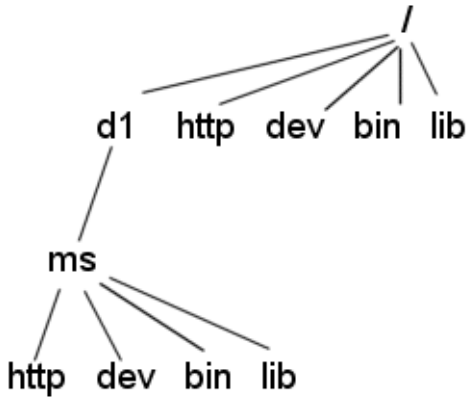
- 80 (保護されていないサーバー用)

- UNIX または Linux の場合には、ドキュメントルートディレクトリに対して chroot 機能を有効にします。保護されていないサーバーは、chroot を使用してリダイレクトされたドキュメントルートに対する参照権限を持つ必要があります。

chroot を使用して、サーバーを特定のディレクトリに限定するよう第 2 のルートディレクトリを作成できます。保護されていないサーバーの保護対策としてこの機能を使用します。たとえば、ルートディレクトリを /d1/ms とすることもできます。そのとき、Web サーバーが、ルートディレクトリにアクセスしようとする場合は常に、実際には /d1/ms に行き着きます。また、/dev にアクセスしようとした場合は、/d1/ms/dev に行き着きます。したがって、実際のルートディレクトリの下にあるファイルにはいっさいアクセスさせずに、UNIX/Linux システムで Web サーバーを稼動することができます。

ただし、chroot を使用する場合には、次の図に示すように、Sun ONE Web Server によって要求されるディレクトリ構造全体を代替ルートディレクトリの下に設定する必要があります。

chroot ディレクトリ構造の例



仮想サーバークラスへの chroot の指定

次の手順で、仮想サーバークラスに chroot ディレクトリを指定できます。

1. サーバーマネージャにアクセスし、サーバーのサーバーインスタンスをドロップダウンリストから選択します。
2. 「Virtual Server Class」タブを選択します。
3. 「Edit Classes」リンクをクリックします。
4. chroot を指定したいクラスについて、「Option」が「Edit」に設定されていることを確認します。

5. そのクラスの「Advanced」ボタンをクリックします。

   「Virtual Servers CGI Settings」ページが表示されます。

6. 「Chroot」フィールドに絶対パス名を入力します。
7. 「OK」をクリックします。
8. 「Apply」をクリックします。
9. 「Load Configuration Files」を選択して、ダイナミックに適用します。

仮想サーバーへの chroot の指定

次の手順で、特定の仮想サーバーに `chroot` ディレクトリを指定できます。

1. サーバーマネージャにアクセスし、サーバーのサーバーインスタンスをドロップダウンリストから選択します。
2. 「Virtual Server Class」タブを選択します。
3. 「Tree View of the Server」から `chroot` ディレクトリを指定したい仮想サーバーへのリンクをクリックします。
4. 「Settings」タブを選択します。

   「Settings」ページが表示されます。

5. 「Chroot Directory」の隣にある「Set to」フィールドに、絶対パス名を入力します。
6. 「OK」をクリックします。
7. 「Apply」をクリックします。
8. 「Load Configuration Files」を選択して、ダイナミックに適用します。

「Class Manager Virtual Servers」タブと「CGI Settings」リンクを使用して、仮想サーバーに `chroot` ディレクトリを指定することもできます。

仮想サーバーに `chroot` ディレクトリを指定する方法の詳細は、『Sun ONE web Server 6.1 Programmer's Guide』を参照してください。

第 7 章

サーバークラスタの管理

この章では、Sun ONE Web Server のクラスタ化の概念、およびそれを使用してサーバー間で設定を共有する方法について説明します。

この章には、次の節が記述されています。

- クラスタについて
- サーバークラスタの使用に関するガイドライン
- クラスタの設定
- クラスタへのサーバーの追加
- サーバー情報の変更
- クラスタからのサーバーの削除
- サーバークラスタの制御
- 変数の追加

クラスタについて

クラスタとは、1 つの管理サーバーから管理することができる、複数の Sun ONE Web Server で構成されたグループのことです。各クラスタには、管理サーバーとして指定された 1 つのサーバーを組み込む必要があります。複数のクラスタを使用している場合、1 つの「マスター」管理サーバーから、すべてのクラスタを管理できます。このマスター管理サーバーは、クラスタに関するすべての情報を取得して、クラスタを構成する Sun ONE Web Server を管理するためのインタフェースを提供します。

サーバーをクラスタ構成にすると、次のようなタスクを実行できるようになります。

- すべての Sun ONE Web Server を集中して管理する
- 1 つ、または複数の設定ファイルをサーバー間で共有する

- 1 つの「マスター」管理サーバーから、すべてのサーバーの起動または停止を行う
- 指定したサーバーの、アクセスログやエラーログを表示する

Sun ONE Web Server をクラスタ化することで、マスター管理サーバーを指定して、すべてのクラスタを管理することができます。

注 個々のサーバーは、ネットワーク内の任意のコンピュータにインストールできますが、「マスター」として指定する管理サーバーは、クラスタを構成するすべてのサーバーの情報を持っており、クラスタ内のすべての管理サーバーにアクセスできるように設定する必要があります。

サーバークラスタの使用に関するガイドライン

クラスタを設定する際、すべてのクラスタの情報を持つマスター管理サーバーは、クラスタの各管理サーバーと通信します。クラスタの各管理サーバーには、マスター管理サーバーと同じ管理ユーザー名とパスワードを登録しておく必要があります。

クラスタを設定する場合は、事前にそのクラスタに含めるすべてのサーバーへのインストールを完了しておく必要があります。たとえば、1 クラスタあたり 5 つの Sun ONE Web Server を組み込んだ、合計 3 つのクラスタを設置する場合、次の手順を行います。

1. コンピュータにすべてのサーバーをインストールし、各サーバーが、マスター管理サーバーと同じ管理ユーザー名とパスワードを使用して実行できるように設定します。
2. 各クラスタに、1 つの Sun ONE Web Server を管理サーバーとして設定します。
3. クラスタに含まれる管理サーバーのうち 1 つを選択し、すべてのクラスタに対するマスター管理サーバーとして設定します。マスター管理サーバーとして選択するのは、どの管理サーバーでもかまいません。

警告 クラスタは、同じプラットフォームのものでなければなりません。クラスタ内のすべてのサーバーは、UNIX または Windows のいずれかで統一されていなければなりません。同一のクラスタの中に UNIX と Windows のサーバーが混在していると、ハングアップやクラッシュを引き起こす可能性があります。

次に、サーバーのグループで複数のクラスタを構成する際のガイドラインを示します。

- クラスタの作成に先立って、クラスタに組み込むサーバーをすべてインストールしておきます。

- クラスタ内のサーバーがすべて、Sun ONE Web Server 6.1 であることを確認します。
- すべてのクラスタに固有の管理サーバーが、マスター管理サーバーと同じユーザ ID とパスワードを持っていることを確認します。分散管理の機能を使用して、各管理サーバーに複数の管理者を設定することもできます。
- ネットワーク内のすべてのコンピュータに、サーバーをインストールします。ただし、1 つのクラスタ内のすべてのコンピュータは、Windows または UNIX のいずれかで統一する必要があります。
- クラスタ固有の任意の管理サーバーを、マスター管理サーバーに指定することができます。
- マスター管理サーバーがクラスタ固有の各管理サーバーにアクセスできることを確認します。マスター管理サーバーは、すべてのインストールされている Sun ONE Web Server に関する情報を取得します。
- 管理サーバーはすべて、Sun ONE Web Server 6.0 または 6.1 であること、また、同じプロトコル (HTTP または HTTPS) を使用していることを確認します。クラスタへの追加をサポートしているのは Sun ONE Web Server 6.0 または 6.1 だけです。
- 1 つのクラスタ内の 1 つの管理サーバーのプロトコルを変更する場合は、すべての残りの管理サーバーのプロトコルも同様に変更する必要があります。その場合、「Modify Server」のインタフェースを使用して、クラスタの個々のサーバーの設定を変更できます。

クラスタの設定

Sun ONE Web Server のクラスタを設定するには、次の手順を実行します。

1. Sun ONE Web Server を、クラスタに含めるすべてのコンピュータにインストールします。

   各クラスタの管理サーバーが、マスター管理サーバーが認証に使用できるユーザ名とパスワードを持っていることを確認します。これを行うには、デフォルトのユーザ名とパスワードを使用するか、または分散管理を設定します。
2. マスター管理サーバーに使用するサーバーをインストールします。ユーザ名とパスワードが、手順 1 で設定したものと一致していることを確認します。
3. サーバーをクラスタリストに追加します。
4. リモートサーバーの管理は、クラスタフォームから「Server Manager」フォームにアクセスするか、または、同じクラスタ内のサーバーの設定ファイルを別のサーバーにコピーして行います。

注	リモートサーバーの設定を変更したら、リモートサーバーを再起動します。

クラスタへのサーバーの追加

クラスタにサーバーを追加する際は、そのクラスタを管理している管理サーバーとポート番号を指定します。追加する管理サーバーが複数のサーバー情報を持っている場合、すべてのサーバーが、そのクラスタに追加されます。個々のサーバーは、後から削除することができます。

注	リモート管理サーバーがクラスタの情報を持っている場合、このリモートクラスタの中のサーバーは追加されません。マスター管理サーバーに追加するサーバーは、リモートコンピュータに物理的にインストールされているサーバーだけです。

クラスタにリモートサーバーを追加するには、次の手順を実行します。

1. マスター管理サーバーが起動していることを確認します。
2. マスター管理サーバーにアクセスして、「Cluster Mgmt」タブを選択します。
3. 「Add Server」リンクをクリックします。
4. リモート管理サーバーが使用するプロトコルを選択します。
   - `http` (通常の管理サーバーの場合)
   - `https` (セキュリティ保護された管理サーバーの場合)
5. 「Admin Server Hostname」フィールドに、リモートサーバーの `magnus.conf` ファイルに表示されているように、絶対パスによるドメイン名を入力します。

   例 : `plaza.sun.com`
6. リモート管理サーバーのポート番号を入力します。
7. 「OK」をクリックします。

   これで、マスター管理サーバーは、リモートサーバーへの通信を試みます。この処理には、2、3 分かかります。その後、サーバーがクラスタに追加されたという確認メッセージが表示されます。

8. 「OK」をクリックします。

注	異なるコンピュータに組み込んでいる複数のサーバーで、同じ識別子を使用している場合は、各コンピュータのサーバー識別子とホスト名が表示されます。サーバー識別子とホスト名が両方とも同じサーバーが存在する場合には、ポート番号も表示されます。

注	クラスタ制御を有効にすると、クラスタのマスターがクラスタ内のスレーブごとに、`https-`*`server-instance`*`/config/cluster/`*`server-name`*`/https-`*`server-name`*`/` ディレクトリ内に多数のファイルを作成します。これらのファイルが作成される場所は変更できません。

サーバー情報の変更

「Modify Server」オプションは、スレーブサーバー上で、スレーブ管理ポート情報が変更されたあと、その情報を更新するときだけに使用します。クラスタ内のリモート管理サーバーのポート番号を変更したときは、そのクラスタに格納されている管理サーバーの情報も変更する必要があります。スレーブ管理サーバーに対するその他の変更の場合は、一旦そのサーバーを削除し、変更が終わったら、元のようにクラスタに追加する必要があります。

リモート管理サーバーは、マスタークラスタデータベースが変更されても、関連のファイルが Cluster Control を経由して転送されていない限り、影響を受けません。

クラスタ内のサーバーに関する情報を変更するには、次の手順を実行します。

1. マスター管理サーバーにアクセスして、「Cluster Mgmt」タブを選択します。
2. 「Modify Server」リンクをクリックします。

   一意のサーバー識別子で識別された、すべてのサーバーが表示されます。
3. 次のどちらかで、変更するサーバーを 1 つまたは複数選択します。
   - 特定のサーバーにチェックマークを付ける
   - 「Select All」を選択する

   すべての選択を元に戻すときは「Reset」をクリックします。
4. 新しいポート番号を入力します。
5. 「OK」をクリックします。

クラスタからのサーバーの削除

クラスタからサーバーを削除するには、次の手順を実行します。

1. マスター管理サーバーにアクセスして、「Cluster Mgmt」タブを選択します。
2. 「Remove Server」リンクをクリックします。
3. 次のどちらかで、削除するリモートサーバーを 1 つまたは複数選択します。
   - 特定のサーバーにチェックマークを付ける
   - 「Select All」を選択する

   選択を元に戻すときは「Reset Selection」をクリックします。
4. 「OK」をクリックします。

サーバーがクラスタから削除されることを確認するメッセージが表示されます。削除したサーバーには、そのクラスタからはもうアクセスできません。アクセスできるのは、そのサーバーの管理サーバーからのみとなります。

サーバークラスタの制御

Sun ONE Web Server 6.1 を使用すると、クラスタ内のリモートサーバーを、次のように制御することができます。

- リモートサーバーを起動または停止する
- アクセスログやエラーログを表示する
- 設定ファイルをサーバーに転送する

警告 クラスタは、同じプラットフォームのものでなければなりません。クラスタ内のすべてのサーバーは、UNIX または Windows のいずれかで統一されていなければなりません。設定ファイルを異なるプラットフォームから転送すると、サーバーがハングアップしたり、クラッシュしたりする可能性があります。

クラスタ内のサーバーを制御するには、次の手順を実行します。

1. マスター管理サーバーのサーバーマネージャにアクセスして、「Cluster Mgmt」タブを選択します。
2. 「Cluster Control」リンクをクリックします。
3. 次のいずれかにより、制御するサーバーを、1 つまたは複数、選択します。
   - 特定のサーバーにチェックマークを付ける

- 「Select All」を選択し、そのクラスタ内のサーバーをすべて選択する

選択を元に戻すときは「Reset Selection」をクリックします。

4. ドロップダウンメニューから、「Start remote servers」または「Stop remote servers」を選択します。
5. ドロップダウンメニューから、「View Access log records」または「View Error log records」を選択し、表示したい行の番号を入力します。
6. 設定ファイルを転送するには、次のいずれかを行います。
   a. ドロップダウンメニューから、転送する設定ファイルを選択します。
   b. ドロップダウンメニューから、設定ファイルの転送元であるサーバーを選択します。
   c. 「Transfer」をクリックします。

変数の追加

変数は、クラスタ内のサーバーに複数の異なる値を設定する必要がある場合に使用されます。この値は、異なるポート番号を使用してスレーブを定義するためのマクロであったり、異なる `shlib` パスを定義するためのプラグインであったりします。

変数の追加は、マスタークラスタデータベースにのみ影響します。関連ファイルが Cluster Control 経由で転送されている場合を除き、リモート管理サーバーは影響を受けません。変数が定義されると、管理サーバーは独立して稼動することができなくなります。

クラスタ内のリモートサーバーに変数を追加するには、次の手順を実行します。

1. マスター管理サーバーから、「Cluster Mgmt」タブを選択します。
2. 「Add Variables」リンクをクリックします。
3. 変数を追加したい特定のサーバーにチェックマークを付けます。
4. 「Name」フィールドに、追加する変数のタイプを入力します。

   たとえば、「Port」と入力します。
5. 「Value」フィールドに、追加する値を入力します。

   たとえば、「Name」フィールドに「Port」と入力した場合、この値はポート番号となります。
6. 「OK」をクリックします。

   サーバー変数が追加されたという確認メッセージが表示されます。
7. 「OK」をクリックします。

この変数は、スレーブに転送するサーバーの設定ファイルにも追加する必要があります。たとえば、変数 port を転送する場合、サーバー設定ファイル (たとえば server.xml) 内にこの変数を次のように宣言する必要があります。

```
<SERVER legacyls="ls1" qosactive="no" qosmetricsinterval="30"
qosrecomputeinterval="100">

...

<LS id="ls1" ip="0.0.0.0" port="$port" security="off"
acceptorthreads="1" blocking="no">

...

</SERVER>
```

設定ファイルで、各スレーブに対して異なる値を持つ複数の変数を設定することができます。一旦追加した変数は、「Add Variables」ページでドロップダウンの「Option」リストを使用して、編集したり削除したりできます。

第 3 部

設定と監視

第 8 章「サーバーの詳細設定」

第 9 章「サーバーへのアクセス制御」

第 10 章「ログファイルの使用」

第 11 章「サーバーの監視」

第 12 章「ネーミングとリソースの設定」

第 8 章

サーバーの詳細設定

この章では、Sun ONE Web Server のサーバー設定方法について説明します。

この章では、次の項目について説明します。

- サーバーの起動と停止
- サーバーのパフォーマンスの調整
- magnus.conf ファイルの編集
- 待機ソケットの追加と編集
- MIME タイプの選択
- アクセスの制限
- 設定の復元
- ファイルキャッシュの設定
- スレッドプールの追加と使用

サーバーの起動と停止

サーバーがインストールされると、サーバーは常時稼働して HTTP 要求を待機し、受け取ります。

サーバーのステータスが「Server On/Off」ページに表示されます。以下のいずれかの方法で、サーバーの起動と停止を実行できます。

- 「Server On/Off」ページの「Server On」または「Server Off」をクリックします。
- Windows: 「コントロールパネル」の「サービス」ウィンドウを使用します。

- UNIX/Linux：start を使用します。このスクリプトを init と一緒に使用する場合、/etc/inittab に起動コマンド http:2:respawn:*server_root*/*type-identifier*/start -start -i を記述する必要があります。
- UNIX/Linux：stop を使用して、サーバーを完全にシャットダウンします。サービスは、サーバーが再起動するまで中断されます。etc/inittab ファイルで自動的に再起動するように設定している場合は (「respawn」を使用)、サーバーをシャットダウンする前に Web サーバーに関する行を etc/inittab から削除する必要があります。この処理を行わない場合、サーバーは自動的に再起動されます。

サーバーをシャットダウンしたあと、シャットダウンプロセスが完了し、ステータスが「Off」に変更されるまでに数秒かかる場合があります。

マシンに障害が発生した場合やオフラインになっている場合、サーバーは停止し、処理中の要求が失われる可能性があります。

注 サーバーにセキュリティモジュールがインストールされている場合、サーバーを起動したり、停止したりする前に、適切なパスワードを入力するように要求されます。

注 UNIX では、Sun ONE Web Server のインストールに、オペレーティングシステムでデフォルトで使用可能になっているよりも多くのメモリーとファイル記述子のいずれか、またはその両方が必要になる場合があります。サーバーを起動できない場合は、ulimit コマンドを使用して、オペレーティングシステムのリソースの制限値を確認します。詳細は、オペレーティングシステムの ulimit のマニュアルページを参照してください。

終了タイムアウトの設定

サーバーをオフにすると、新しい接続の受け入れは停止します。その後、サーバーはすべての未処理の接続処理が完了するまで待ちます。タイムアウトになるまでサーバーが待機する時間は、magnus.conf ファイルで設定できます。このファイルは *server_root*/https-*server_name*/config/ にあります。デフォルトでは、30 秒に設定されています。この値を変更するには、次の行を magnus.conf に追加します。

```
TerminateTimeout seconds
```

seconds は、タイムアウトになるまでサーバーが待機する秒数を表します。

この値を変更することによる利点は、接続の処理が完了するまでサーバーが待機する時間が、より長くなることです。ただし、サーバーは応答していないクライアントに接続されていることがあるため、終了タイムアウト値を大きくすると、サーバーのシャットダウンにかかる時間が長くなる可能性があります。

サーバーの再起動 (UNIX/Linux)

以下のいずれかの方法で、サーバーを再起動できます。

- `inittab` ファイルから自動的に再起動します。

  System V から派生したものではないバージョン (SunOS 4.1.3 など) の UNIX/Linux を使用している場合は、`inittab` ファイルを使用できないことに注意してください。

- マシンの再起動時に、`/etc/rc2.d` 内のデーモンで自動的に再起動します。
- 手動で再起動します。

インストールスクリプトでは `/etc/rc.local` ファイルや `/etc/inittab` ファイルを編集できないため、テキストエディタでそれらのファイルを編集する必要があります。これらのファイルの編集方法がわからない場合は、システム管理者に問い合わせるか、ご使用のシステムのマニュアルを参照してください。

通常、SSL が有効なサーバーは、起動する前にパスワードを要求するため、これらのファイルのいずれかで起動することはできません。パスワードをプレーンテキストでファイルに保存していると、SSL が有効なサーバーを自動的に起動できますが、この方法は推奨されません。

警告 SSL が有効なサーバーの起動スクリプトにプレーンテキストでパスワードを保存しておくことは、セキュリティ上、非常に危険です。ファイルにアクセスできるユーザーなら誰でも、SSL が有効なサーバーのパスワードにアクセスできるからです。SSL が有効なサーバーのパスワードをプレーンテキストで保存する前に、セキュリティ上の危険性を考慮してください。

サーバーの起動スクリプト、鍵ペアファイル、および鍵パスワードは、ルートが所有しており (または、ルートではないユーザーがサーバーをインストールした場合は、そのユーザーのアカウントが所有しています)、その所有者のみがそれらへ対する読み取りおよび書き込みアクセス権を持ちます。

SSL が有効なサーバーを自動的に起動

セキュリティ上のリスクが問題とならない場合は、以下の手順を実行して SSL が有効なサーバーを自動的に起動します。

1. テキストエディタを使用して起動ファイルを開きます。起動ファイルは *server_root*/https-*server_id* にあります。
2. スクリプト内の -start 行を検索し、以下のテキストを挿入します。

   echo "*password*"|

   password は、選択した SSL パスワードです。

   たとえば、SSL パスワードが netscape の場合、編集後の行は以下のようになります。

   ```
   -start)
   echo "netscape"|./$PRODUCT_BIN -d $PRODUCT_SUBDIR/config $@
   ```

inittab を使用した再起動 (UNIX/Linux)

inittab を使用してサーバーを再起動するには、/etc/inittab ファイル内に以下のテキストを 1 行で挿入します。

```
http:23:respawn:server_root/type-identifier/start -start -i
```

server_root はサーバーをインストールしたディレクトリ、*type-identifier* はサーバーのディレクトリです。

-i オプションは、サーバーがバックグラウンド処理に切り替わることを防止します。

この行は、サーバーを停止する前に削除する必要があります。

システムの rc (実行制御) スクリプトを使用した再起動 (UNIX/Linux)

/etc/rc.local、または使用しているシステムのそれに相当するスクリプトを使用する場合は、/etc/rc.local 内に以下の行を追加します。

```
server_root/type-identifier/start
```

server_root を、サーバーがインストールされているディレクトリに変更します。

サーバーの手動再起動 (UNIX/Linux)

コマンド行からサーバーを再起動するには、1024 より小さい番号のポートでサーバーを実行している場合は、ルートとしてログインします。1024 またはそれより大きな番号の場合は、ルートとして、またはそのサーバーのユーザーアカウントを使用してログインします。コマンド行プロンプトで、以下の行を入力し、Enter キーを押します。

```
server_root/type-identifier/start
```

server_root はサーバーをインストールしたディレクトリです。

行の末尾で、省略可能なパラメータ -i を使用できます。-i オプションを使用すると、inittab モードでサーバーが実行されます。このモードでは、サーバーのプロセスが強制終了されたかクラッシュした場合に、inittab がサーバーを自動的に再起動します。また、このオプションは、サーバーがバックグラウンド処理に切り替わることを防止します。

注	サーバーがすでに稼働している場合、start コマンドは失敗します。まず、サーバーを停止してから start コマンドを使用してください。また、サーバーの起動に失敗した場合は、再起動を試行する前にプロセスを強制終了する必要があります。

サーバーの手動停止 (UNIX/Linux)

etc/inittab ファイルを使用してサーバーを再起動した場合は、サーバーの停止を試行する前に、/etc/inittab からサーバーを起動するための行を削除し、kill -1 1 を入力する必要があります。そうしないと、サーバーは停止した後で自動的に再起動してしまいます。

サーバーを手動で停止するには、root として、またはサーバーのユーザーアカウントを使用して (そのアカウントを使用してサーバーを起動した場合) ログインし、コマンド行で以下を入力します。

server_root/*type-identifier*/stop

サーバーの再起動 (Windows)

以下の方法でサーバーを再起動できます。

- 「サービス」コントロールパネルを使用してサーバーを再起動します。
- 「サービス」コントロールパネルを使用して、マシンが再起動されるたびにサーバーまたは管理サーバーを再起動するようにオペレーティングシステムを設定します。

Windows の場合は、以下の手順を実行します。

1. コントロールパネルの「サービス」アイコンをダブルクリックします。
2. サービスのリストをスクロールし、サーバー用のサービスを選択します。
3. 「自動」にチェックマークをつけ、コンピュータが起動や再起動するたびに、コンピュータがサーバーを起動するようにします。
4. 「OK」をクリックします。

注	「サービス」ダイアログボックスを使用して、サーバーが使用するアカウントを変更することもできます。サーバーが使用するアカウントの変更について詳細は、100 ページの「ユーザーアカウントの変更 (UNIX/Linux)」を参照してください。

デフォルトでは、管理者は、Web サーバーを起動するときに、鍵データベースのパスワードを入力する必要があります。Web 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。このファイルと鍵データベースが危険にさらされないようにするために、これを行うのはシステムが充分にセキュリティ保護されている場合だけにします。

自動再起動ユーティリティの使用 (Windows)

サーバーに障害が発生した場合、サーバー監視ユーティリティによって、サーバーは自動的に再起動されます。デバッグ用のツールがインストールされているシステムでは、サーバーに障害が発生した場合、デバッグ情報とともにダイアログボックスが表示されます。サーバープラグインの API プログラム (たとえば、NSAPI プログラム) のデバッグを支援するために、タイムアウトに非常に大きな値を設定して、自動的に起動する機能を無効にすることができます。また、レジストリエディタを使用して、デバッグダイアログボックスを無効にすることもできます。

時間間隔の変更 (Windows)

Windows の起動後、サーバーが自動的に再起動されるまでに経過する時間間隔を変更するには、以下の手順を実行します。

1. レジストリエディタを起動します。
2. サーバーのキーを選択します (「レジストリエディタ」ウィンドウの左側で、`HKEY_LOCAL_MACHINE¥SOFTWARE¥Netscape¥Enterprise¥6.0` を開きます)。
3. 「編集」メニューから「Add Value」を選択します。「Add Key」ダイアログボックスが表示されます。
4. 「Value Name」に `MortalityTimeSecs` と入力します。
5. 「Data Type」ドロップダウンリストから「REG_DWORD」を選択します。
6. 「OK」をクリックします。「DWORD Editor」ダイアログボックスが表示されます。
7. OS の起動からサーバーの自動再起動までの時間間隔を秒単位で入力します。

   間隔は、2 進数、10 進数、または 16 進数の形式にすることができます。
8. 前の手順で入力した値の数値形式 (2 進数、10 進数、または 16 進数) をクリックします。

9. 「OK」をクリックします。

   「Registry Editor」ウィンドウの右側に、16 進数形式で MortalityTimeSecs 値が表示されます。

デバッグダイアログボックスの無効化 (Windows)

システムのデバッグ設定を変更したアプリケーション (コンパイラなど) がインストールされている場合、サーバーに障害が発生すると、システムによって生成されたアプリケーションエラーダイアログボックスが表示されます。「OK」をクリックするまで、サーバーは再起動されません。

サーバーに障害が発生した場合にデバッグダイアログボックスが表示されないようにするには、以下の手順を実行します。

1. 「Registry Editor」を起動します。
2. 「Registry Editor」ウィンドウの左側で、HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows NT¥CurrentVersion に表示される「AeDebug」キーを選択します。
3. ウィンドウの右側に表示される「Auto」値をダブルクリックします。

   「String Editor」ダイアログボックスが表示されます。
4. 文字列の値を「1」に変更します。

サーバーのパフォーマンスの調整

スレッドの制限値を調整するには、magnus.conf ファイルを編集する方法と、サーバーマネージャから調整する方法の 2 つの方法があります。

magnus.conf ファイルを編集する場合、RqThrottleMinPerSocket が最小値で、RqThrottle が最大値です。

最小制限値は、サーバーが WaitingThreads 状態で保持しようとするスレッド数の目標値です。この数値はあくまでも目標です。この状態での実際のスレッド数は、この値より若干大きくても、小さくてもかまいません。デフォルト値は 48 です。最大スレッド数は、同時に実行できるアクティブなスレッドの最大数の厳密な制限値を表します。これは、パフォーマンスのボトルネックとなる可能性があります。デフォルト値は 128 です。

サーバーマネージャを使用する場合、以下の手順を実行します。

1. 「Preferences」タブを選択します。
2. 「Performance Tuning」リンクをクリックします。
3. 「Maximum simultaneous requests」フィールドに値を入力します。

詳細は、オンラインヘルプの「Performance Tuning ページ」を参照してください。

magnus.conf ファイルの編集

起動された Sun ONE Web Server は、サーバーの動作と設定に影響するグローバル変数セットの設定を確定するために、*server_root*/*server_id*/config ディレクトリから magnus.conf というファイルを検索します。Sun ONE Web Server は、magnus.conf に定義されているすべての指令を実行します。サーバーマネージャの「Magus Editor」を使用して、magnus.conf ファイル内の特定の設定を編集することができます。

magnus.conf ファイルの完全な説明と、テキストエディタによるファイルの編集については、『Sun ONE Web Server 6.1 Administrator's Configuration File Reference』および『Sun ONE Web Server 6.1 NSAPI Programmer's Guide』を参照してください。

「Magnus Editor」にアクセスするには、以下の手順を実行します。

1. サーバーマネージャにアクセスし、「Preferences」タブを選択します。
2. 「Magnus Editor」リンクをクリックします。
3. ドロップダウンリストから編集する設定を選択し、「Manage」をクリックします。

   選択した設定を編集するためのエディタが表示されます。
4. 必要に応じて設定を変更し、「OK」をクリックします。

各「Settings」ページの詳細は、オンラインヘルプの「Magnus Editor ページ」を参照してください。

待機ソケットの追加と編集

サーバーで要求を処理するには、待機ソケットから要求を受け取り、適切な仮想サーバーにその要求を送信する必要があります。Sun ONE Web Server をインストールすると、ls1 という待機ソケットが自動的に作成されます。この待機ソケットには、0.0.0.0 の IP アドレス と、インストール時に HTTP サーバーのポート番号として指定したポート番号 (デフォルトでは 80) が割り当てられます。デフォルトの待機ソケットは削除できません。

サーバーの待機ソケットの設定は、サーバーマネージャの「Listen Sockets Table」を使用して編集できます。この表にアクセスするには、以下の手順を実行します。

1. サーバーマネージャにアクセスし、「Preferences」タブをクリックします。
2. 「Edit Listen Sockets」リンクをクリックします。

3. 変更を行い、「OK」をクリックします。

MIME タイプの選択

「MIME Types」ページでは、サーバーの MIME ファイルを編集できます。

Multi-purpose Internet Mail Extension (MIME) タイプは、メールシステムでサポートするマルチメディアファイルのタイプを制御します。また、MIME タイプはどのファイル拡張子が特定のサーバーのファイルタイプに属しているかを示します。たとえば、どのファイルが CGI プログラムかを示します。

仮想サーバーごとに別個の MIME タイプのファイルを作成する必要はありません。必要な数の MIME タイプファイルを作成し、それらを仮想サーバーに関連付けます。サーバーには `mime.types` という MIME タイプファイルがデフォルトで存在し、これを削除することはできません。このファイルは絶対パスにできます。

「MIME Types」ページにアクセスするには、以下の手順を実行します。

1. サーバーマネージャにアクセスし、「Preferences」タブをクリックします。
2. 「MIME Types」リンクをクリックします。
3. 変更を行い、「OK」をクリックします。

詳細は、オンラインヘルプの「Mime Settings ページ」、および第 13 章「仮想サーバーの使用」を参照してください。

アクセスの制限

サーバーマネージャの「Restrict Access」ページを使用して、サーバー全体またはサーバーの一部 (つまり、ディレクトリ、ファイル、ファイルタイプ) へのアクセスを制御できます。サーバーが受信した要求を評価する場合、アクセス制御エントリ (ACE) と呼ばれる規則の階層に基づいてアクセス権を決定し、一致するエントリを使用して、要求を承認するか、拒否するかを決定します。各 ACE は、サーバーが階層内の次の ACE に進むべきかどうかを指定します。ACE の集合は、アクセス制御リスト (ACL) と呼ばれます。要求がサーバーに着信すると、サーバーは *vsclass*.obj.conf (*vsclass* は仮想サーバーのクラス名) で ACL を検索して参照し、アクセスの可否を決定します。デフォルトでは、サーバーには、複数の ACL が含まれる 1 つの ACL ファイルがあります。

アクセス制御は、管理サーバーを使用してすべてのサーバーに対して全体的に設定したり、サーバーマネージャを使用して特定のサーバーインスタンス内のリソースに対して設定したりすることができます。リソースに対するアクセス制御の設定の詳細は、第 9 章「サーバーへのアクセス制御」の 193 ページの「アクセス制御の設定」を参照してください。

注	サーバーへのアクセスを制限する前に、分散管理を有効にする必要があります。

Sun ONE Web Server へのアクセスを制限するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Preferences」タブを選択します。
2. 「Restrict Access」リンクをクリックします。

詳細は、第 9 章「サーバーへのアクセス制御」およびオンラインヘルプの「Restrict Access ページ」を参照してください。

設定の復元

「Restore Configuration」ページでは、設定ファイルのバックアップコピーを参照し、特定の日に保存された設定データに戻すことができます。

注	Windows では、設定ファイルに対して自分が行なった変更を元に戻す場合にだけこのページを使用します。インストール時に作成したバックアップバージョンには戻さないでください。このバージョンは完全ではない可能性があります。

詳細は、オンラインヘルプの「Restore Configuration ページ」を参照してください。

ファイルキャッシュの設定

Sun ONE Web Server では、ファイルキャッシュを使用して、スタティックな情報をより早く提供します。以前のバージョンのサーバーでは、要求をファイルキャッシュに転送するアクセラレータキャッシュもありましたが、アクセラレータキャッシュは使用されなくなりました。ファイルキャッシュには、ファイルに関する情報とスタティックなファイルの内容が保存されています。また、ファイルキャッシュは、サーバーで解析される HTML の処理速度を向上させるために使用される情報もキャッシュします。

デフォルトでは、ファイルキャッシュが有効になっています。ファイルキャッシュの設定は、`nsfc.conf` という名前のファイルに保存されています。ファイルキャッシュの設定は、サーバーマネージャを使用して変更することができます。

詳細は、`http://docs.sun.com` で、オンラインの『Performance Tuning and Sizing Guide』を参照してください。

スレッドプールの追加と使用

スレッドプールを使用すると、特定のサービスに対して特定数のスレッドを割り当てることができます。

スレッドプールのもう 1 つの用途は、スレッドに対して安全ではないプラグインの実行用です。プールの最大スレッド数を「1」に定義すると、指定されたサービス機能で許容される要求が 1 つだけになります。

スレッドプールを追加するときに指定する情報は、スレッドの最小数と最大数、スタックのサイズ、キューのサイズなどです。

詳細は、`http://docs.sun.com` で、オンラインの『Performance Tuning and Sizing Guide』を参照してください。

ネイティブスレッドプールと汎用スレッドプール (Windows)

Windows では、ネイティブスレッドプール (`NativePool`) と追加の汎用スレッドプールの 2 つのタイプのスレッドプールを使用できます。

ネイティブスレッドプールを編集するには、サーバーマネージャで「Native Thread Pool」ページにアクセスします。

目的に応じて、必要な数だけ汎用スレッドプールを作成できます。汎用スレッドプールを作成するには、サーバーマネージャで「Generic Thread Pools」ページにアクセスします。

スレッドプール (UNIX/Linux)

UNIX/Linux でのスレッドは必ず (ユーザーによるスケジューリングではなく) OS でスケジューリングされるため、UNIX/Linux ユーザーは `NativePool` を使用する必要がなく、この設定を編集するためのサーバーマネージャのページもありません。ただし、UNIX/Linux ユーザーがスレッドプールを作成することはできます。汎用スレッドプールを作成するには、サーバーマネージャで「Thread Pools」ページにアクセスします。

スレッドプールの編集

スレッドプールを追加すると、サーバーマネージャでスレッドプールの設定された値 (最小スレッド数、最大スレッド数など) を変更できます。

スレッドプールの設定は、*vsclass*.obj.conf でも編集できます。この *vsclass* は、仮想サーバーのクラス名です。

vsclass.obj.conf には、スレッドプールは次のように表示されます。

```
Init fn="thread-pool-init" name=name_of_the_pool MaxThreads=n
MinThreads=n QueueSize=n StackSize=n
```

パラメータ MinThreads、MaxThreads、QueueSize、および StackSize を使用して、プールを変更します。

Windows ユーザーはサーバーマネージャを使用して、いつでもネイティブプールの設定を編集できます。

スレッドプールの使用

スレッドプールを設定したあと、それを特定のサービス用のスレッドプールとして指定すると、使用できるようになります。

スレッドプールを設定するには、サーバーマネージャの「Preferences」タブをクリックし、「Thread Pool」を選択します。スレッドプールが設定されると、「Thread Pool」リストに、指定した特定のサービスに使用できるスレッドプールが表示されます。

また、*vsclass*.obj.conf の (この *vsclass* は仮想サーバーのクラス名です) load-modules 関数に含まれる pool パラメータを使用して、スレッドプールを指定することもできます。

```
pool="name_of_pool"
```

さらに、NSAPI 関数で pool パラメータを使用し、指定したプールでその NSAPI 関数だけが実行されるようにすることもできます。

第 9 章

サーバーへのアクセス制御

この章では、管理サーバーや、Web サイト上に配置されたファイルやディレクトリへのアクセスを制御するためのさまざまな方法について説明します。たとえば、管理サーバーについては、マシンにインストールされているすべてのサーバーを完全に制御できるユーザーや、1 台以上のサーバーを部分的に管理できるユーザーを指定できます。管理サーバーでアクセス制御を使用するには、分散管理を有効にして、LDAP データベースに管理グループを設定する必要があります。この章では、すでに分散管理を設定していて、LDAP データベースにユーザーとグループを定義していることを前提としています。

第 4 章「Web コンテナと Web アプリケーションの J2EE ベースのセキュリティ」および第 6 章「証明書と鍵の使用」で説明されている Web サーバーのセキュリティについても、確認する必要があります。

この章では、次の項目について説明します。

- アクセス制御とは
- アクセス制御のしくみ
- ファイルベースの認証用 ACL の作成
- アクセス制御の設定
- アクセス制御オプションの選択
- サーバーの一部へのアクセス制御
- ダイナミックアクセス制御ファイルの使用
- 仮想サーバーへのアクセス制御

アクセス制御とは

アクセス制御を使用すると、次の内容を指定できます。

- Sun ONE Web 管理サーバーにアクセスできるユーザー
- ユーザーがアクセスできるプログラム
- Web サイト上のファイルやディレクトリにアクセスできるユーザー

サーバー全体やサーバーの一部、または Web サイトのファイルやディレクトリへのアクセスを制御できます。アクセス制御エントリ (ACE) と呼ばれる規則の階層を作成します。これによって、アクセスを許可したり拒否したりすることができるようになります。各 ACE は、サーバーが階層内の次の ACE を確認する必要があるかどうかを指定します。作成する ACE の集合は、アクセス制御リスト (ACL) と呼ばれます。

デフォルトでは、サーバーには、複数の ACL が含まれる 1 つの ACL ファイルがあります。Sun ONE Web Server は、受信する要求に使用する仮想サーバーを指定したあと、その仮想サーバーに対して ACL が設定されているかどうかを確認します。現在の要求に適用される ACL が見つかった場合、Sun ONE Web Server は ACL に含まれる ACE を評価し、アクセスを許可するか拒否するかを決定します。

次の内容を基準にして、アクセスを許可または拒否します。

- 要求を送信したユーザー (ユーザー - グループ)
- 要求の送信元 (ホスト - IP)
- 要求が発生した日時 (たとえば、時刻)
- 使用されている接続のタイプ (SSL)

ユーザー - グループのアクセス制御の設定

特定のユーザーまたはグループに対して、Web サーバーへのアクセスを制限することができます。ユーザー - グループのアクセス制御を設定すると、ユーザーはユーザー名とパスワードを入力しないと、サーバーへのアクセス権を取得できなくなります。サーバーは、クライアント証明書の情報またはクライアント証明書自体を、ディレクトリサーバーのエントリと比較します。

管理サーバーでは、基本認証だけを使用します。管理サーバーでクライアント認証を要求する場合、`obj.conf` の ACL ファイルを手動で編集し、認証メソッドを SSL に変更する必要があります。

なお、( ) 内は画面上の表示を示しています。

- デフォルト
- 基本 (Basic)

- SSL
- ダイジェスト (Digest)
- その他 (Other)

これらすべてのメソッドで、ディレクトリサーバーが必要です。

ユーザー - グループ認証の場合、ユーザーは自分自身の認証を行わないと、管理サーバー、および Web サイト上のファイルやディレクトリにアクセスできません。クライアント証明書、またはダイジェスト認証プラグインを使用して認証を行う場合、ユーザーはユーザー名とパスワードを入力することによって識別情報を証明します。クライアント証明書を使用する場合は、暗号化が必要です。暗号化とクライアント証明書については、第 4 章「Web コンテナと Web アプリケーションの J2EE ベースのセキュリティ」を参照してください。

デフォルト認証 (Default)

デフォルト認証は、優先されるメソッドです。デフォルト設定では、`obj.conf` ファイルで指定したデフォルトメソッドを使用します。`obj.conf` でメソッドが設定されていない場合は、「基本」メソッドを使用します。「Default」チェックボックスにチェックマークを付けた場合、ACL 規則によって ACL ファイル内のメソッドが指定されることはありません。「Default」を選択すると、`obj.conf` ファイル内の 1 行を編集することによって、すべての ACL のメソッドを簡単に変更できます。

基本認証 (Basic)

基本認証では、Web サーバーまたは Web サイトにアクセスするユーザーに対して、ユーザー名とパスワードを要求します。これがデフォルトの設定です。Sun ONE Directory Server などの LDAP データベースにユーザーとグループのリストを作成して格納する必要があります。ここでは Web サーバーではなく、別のサーバールートにインストールされているディレクトリサーバー、またはリモートマシンにインストールされているディレクトリサーバーを使用する必要があります。

管理サーバー内、または Web サイト上のユーザー - グループ認証が設定されているリソースにユーザーがアクセスした場合、Web ブラウザにユーザー名とパスワードの入力を求めるダイアログボックスが表示されます。サーバーに対する暗号化が設定されているかどうかに応じて、サーバーはこの情報を暗号化された状態、または暗号化されていない状態で受信します。

注	SSL 暗号化なしで基本認証を使用する場合、暗号化されていないユーザー名とパスワードがネットワークを経由して送信されます。ネットワークパケットは不正に読み取られる可能性があり、ユーザー名とパスワードが不正に知られてしまう可能性があります。基本認証は、SSL 暗号化とホスト - IP 認証のどちらかまたはその両方と組み合わせた場合にもっとも効果的です。ダイジェスト認証を使用しても、この問題を回避できます。

ユーザーがサーバーに対して自分自身の認証を行う場合、次のダイアログが表示されます。

ユーザー名とパスワードのプロンプトの例

「OK」をクリックすると、次の内容が表示されます。

- Sun ONE Web 管理サーバーへのアクセスに対して認証された場合は、「Server Administration」ページ
- Web サイトにログインしている場合は、要求されたファイルまたはディレクトリのリスト
- ユーザー名またはパスワードが無効な場合は、アクセスが拒否されたことを示すメッセージ

認証を受けていないユーザーが「Access Denied Response」ページで受信するアクセス拒否メッセージは、カスタマイズすることができます。

SSL 認証 (SSL)

サーバーは、次の 2 つの方法で、セキュリティ証明書付きのユーザーの識別情報を確認できます。

- クライアント証明書の情報を識別情報として使用する
- LDAP ディレクトリで発行されたクライアント証明書を確認する (追加)

クライアントの認証で証明書の情報を使用するようにサーバーを設定した場合、サーバーは次の処理を実行します。

- まず、証明書が信頼できる CA から発行されたものであるかどうかを確認します。そうでない場合、認証は失敗し、トランザクションが終了します。クライアント証明書を有効にする方法については、140 ページの「クライアント認証の要求」を参照してください。

- 証明書が信頼できる認証局 (CA) から発行されている場合は、certmap.conf ファイルを使用して、証明書をユーザーのエントリにマッピングします。証明書マッピングファイルの設定方法については、143 ページの「certmap.conf ファイルの使用」を参照してください。
- 証明書が正しくマッピングされている場合は、そのユーザーに対して指定されている ACL 規則を確認します。証明書が正しくマッピングされている場合でも、ACL 規則によってユーザーのアクセスが拒否される可能性もあります。

特定のリソースへのアクセスを制御するためにクライアント認証を要求することは、サーバーへの接続のすべてに対してクライアント認証を要求することとは異なります。すべての接続に対してクライアント認証を要求するようにサーバーを設定した場合、クライアントは信頼できる CA によって発行された有効な証明書を提示するだけで済みます。ユーザーとグループの認証に SSL メソッドを使用するようにサーバーのアクセス制御を設定した場合、クライアントでは次の内容をすべて満たすことが必要です。

- 信頼できる CA によって発行された有効な証明書を提示する
- 証明書は LDAP 内の有効なユーザーにマッピングされている
- アクセス制御リストで、適切に評価する

アクセス制御を使ってクライアント認証を要求する場合、Web サーバーでは SSL 暗号化方式を有効にする必要があります。SSL を有効にする方法については、第 6 章「証明書と鍵の使用」を参照してください。

SSL で認証されるリソースにアクセスするには、Web サーバーで信頼されている CA から、クライアント証明書が発行されている必要があります。ブラウザのクライアント証明書とディレクトリサーバーのクライアント証明書を比較するように Web サーバーの certmap.conf ファイルが設定されている場合、クライアント証明書はディレクトリサーバー内で発行されている必要があります。ただし、証明書から選択した情報とディレクトリサーバーのエントリだけを比較するように、certmap.conf ファイルを設定することもできます。たとえば、ブラウザ証明書のユーザー ID および電子メールアドレスとディレクトリサーバーのエントリだけを比較するように、certmap.conf ファイルを設定することができます。certmap.conf と証明書のマッピングの詳細は、第 6 章「証明書と鍵の使用」を参照してください。

注 LDAP ディレクトリに対する証明書が確認されるため、SSL 認証メソッドだけは certmap.conf ファイルの修正を必要とします。サーバーへの接続のすべてに対してクライアント認証を要求するメソッドでは、この必要はありません。使用するクライアント証明書を選択する場合は、magnus.conf の AcceptTimeout 指令の値を大きくする必要があります。

ダイジェスト認証 (Digest)

ダイジェスト認証では、ユーザーがユーザー名とパスワードをプレーンテキスト形式 (暗号化されていない形式) で送信することなく、ユーザー名とパスワードに基づいた認証を行うことができます。ブラウザは、ユーザーのパスワードと Web サーバーによって提供される情報の一部を使用し、MD5 アルゴリズムによりダイジェスト値を作成します。このダイジェスト値は、サーバー側でもダイジェスト認証プラグインを使用して計算し、クライアントによって提示されたダイジェスト値と比較します。ダイジェスト値が一致する場合、ユーザーは認証を受けたものと見なします。

これが機能するには、ディレクトリサーバーがプレーンテキスト形式のユーザーのパスワードにアクセスできる必要があります。Sun ONE Directory Server にはリバーシブルパスワードプラグインが用意されています。このプラグインは、対称暗号化アルゴリズムを使用して格納時にデータを暗号化し、後にそれを元の形式に復号化することができます。データへの鍵を持っているのは Directory Server だけです。

ダイジェスト認証の場合は、リバーシブルパスワードプラグインと、Sun ONE Web Server 6.1 に組み込まれているダイジェスト認証に固有のプラグインを有効にする必要があります。ダイジェスト認証を処理するように Web サーバーを設定するには、`dbswitch.conf` でデータベース定義の `digestauth` プロパティを設定します。

サーバーは、指定されている ACL メソッドに基づいて、LDAP データベースに対して認証を試みます。表 9-1 を参照してください。ACL メソッドを指定しない場合、認証が必要であればダイジェスト認証または基本認証が使用され、認証が必要ない場合は基本認証が使用されます。基本認証が使用されるのは、これが優先されるメソッドのためです。

表 9-1 ダイジェスト認証要求の生成

ACL メソッド	認証データベースによってサポートされるダイジェスト認証	認証データベースによってサポートされないダイジェスト認証
デフォルト	ダイジェストと基本	基本
指定なし		
基本	基本	基本
ダイジェスト	ダイジェスト	エラー

`method = digest` を指定して ACL を処理する場合、サーバーは次の内容を実行して認証を試みます。

- 認証要求のヘッダーを確認します。見つからない場合は、ダイジェスト要求に対して 401 の応答が生成され、プロセスが停止します。

- 認証のタイプを確認します。認証のタイプがダイジェストの場合、サーバーは次の内容を実行します。
  - nonce を確認します。このサーバーによって生成された、有効で新しい nonce がない場合は、401 の応答が生成されてプロセスが停止します。無効な場合は、`stale=true` として 401 の応答が生成されてプロセスが停止します。

    server_root/`https-`*server_name*/`config/` ディレクトリ内の `magnus.conf` ファイルに指定されている `DigestStaleTimeout` パラメータの値を変更することで、nonce の有効期限を設定できます。この値を設定するには、次の行を `magnus.conf` に追加します。

    `DigestStaleTimeout` *seconds*

    この *seconds* は、nonce の有効期限です (秒単位)。指定されている秒数が経過すると、nonce の有効期限は切れ、ユーザーからの新しい認証が必要になります。
  - レルムを確認します。レルムが一致しない場合は、401 の応答が生成され、プロセスが停止します。
  - LDAP ディレクトリにユーザーが存在しているかどうかを確認します。見つからない場合は、401 の応答が生成されてプロセスが停止します。
  - ディレクトリサーバーから要求 - ダイジェスト値を取得し、クライアントの要求 - ダイジェスト値と一致することを確認します。一致しない場合は、401 の応答が生成され、プロセスが停止します。
  - 認証情報ヘッダーを構築し、サーバーヘッダーに挿入します。

UNIX でのダイジェスト認証プラグインのインストール

ダイジェスト認証プラグインは、次の両方に存在する共用ライブラリで構成されています。

- `libdigest-plugin.lib`
- `libdigest-plugin.ldif`

UNIX でダイジェスト認証プラグインをインストールするには、次の手順を実行します。

1. この共用ライブラリが、Sun ONE Web Server がインストールされているのと同じサーバーマシンにあることを確認します。
2. Directory Manager のパスワードを確認します。
3. `/path/to` へのすべての参照を、ダイジェストプラグインの共用ライブラリのインストール先に参照先が変更されるように、`libdigest-plugin.ldif` ファイルを修正します。
4. プラグインをインストールするには、次のコマンドを入力します。

```
% ldapmodify -D "cn=Directory Manager" -w password -a <
libdigest-plugin.ldif
```

Windows でのダイジェスト認証プラグインのインストール

ダイジェストプラグインとともに Sun ONE Directory Server を正常に起動できるようにするには、Sun ONE Web Server がインストールされている場所にある .dll ファイルを Sun ONE Directory Server のサーバーマシンにいくつかコピーする必要があります。

Windows でダイジェスト認証プラグインをインストールするには、次の手順を実行します。

1. 次の場所にインストールされている Sun ONE Web Server の共用ライブラリにアクセスします。

   [*server_root*]¥bin¥https¥bin

2. 次のファイルをコピーします。
   - nsldap32v50.dll
   - libspnr4.dll
   - libplds4.dll
3. これらのファイルを次の両方の場所にペーストします。
   - ¥Winnt¥system32
   - Sun ONE Directory Server のインストールディレクトリ [*server_root*]¥bin¥sldap¥server

DES アルゴリズムを使用するための Sun ONE Directory Server の設定

DES アルゴリズムは、ダイジェストパスワードが格納されている属性を暗号化するために必要です。

DES アルゴリズムを使用するように Sun ONE Directory Server を設定するには、次の手順を実行します。

1. Sun ONE Directory サーバーコンソールを起動します。
2. iDS 5.0 のインスタンスを開きます。
3. 「Configuration」タブを選択します。
4. プラグインの隣にある + 記号をクリックします。
5. DES プラグインを選択します。
6. 「Add」を選択して新しい属性を追加します。
7. 「iplanetReversiblePassword」と入力します。

8. 「Save」をクリックします。
9. Sun ONE Directory Server のインスタンスを再起動します。

注	iplanetReversiblePassword 属性でユーザーのダイジェスト認証のパスワードを設定するには、エントリに iplanetReversiblePasswordobject オブジェクトが含まれている必要があります。

その他の認証 (Other)

アクセス制御 API を使用すると、カスタム認証メソッドを作成できます。

ホスト - IP のアクセス制御の設定

管理サーバー、または Web サイトのファイルやディレクトリに対して、特定のコンピュータで動作しているクライアントだけが利用できるように、アクセスを制限できます。そのためには、アクセスの許可または拒否を行うコンピュータをホスト名または IP アドレスで指定します。ワイルドカードパターンを使用して、複数のコンピュータまたはネットワーク全体を指定することができます。ホスト - IP 認証を使用したファイルまたはディレクトリへのアクセスは、ユーザーが意識することなく行われます。このため、各ユーザーは、ユーザー名やパスワードを入力することなく、すぐにファイルやディレクトリにアクセスできます。

特定のコンピュータであっても複数のユーザーが使用しているため、ホスト - IP 認証は、ユーザー - グループ認証と組み合わせるとより効果的です。両方の認証メソッドが使用される場合は、アクセスするときにユーザー名とパスワードが要求されます。

ホスト - IP 認証では、サーバーが動作しているマシンで DNS を設定する必要はありません。ただし、ホスト - IP 認証を使用する場合は、ネットワーク上で DNS が稼働していて、この認証方式を使用するようにサーバーが設定されている必要があります。サーバーに対して DNS を有効にするには、サーバーマネージャの「Preferences」タブにある「Performance Tuning」ページを使用します。

DNS を有効にすると、サーバーで DNS ルックアップを実行する必要があるため、Sun ONE Web Server のパフォーマンスが低下します。サーバーパフォーマンスに対する DNS ルックアップの影響を小さくするには、すべての要求に対して IP アドレスを解決する代わりに、アクセス制御と CGI に対してだけ IP アドレスを解決します。このためには、`obj.conf` ファイルの `AddLog fn="flex-log" name="access"` を `iponly=1` にします。

```
AddLog fn="flex-log" name="access" iponly=1
```

アクセス制御ファイルの使用

管理サーバーまたは Web サイト上のファイルやディレクトリに対してアクセス制御を使用する場合、拡張子が .acl のファイルに設定が格納されます。アクセス制御ファイルは *install_dir*/httpacl ディレクトリに格納されます。*install_dir* はサーバーがインストールされている場所です。たとえば、/usr/Sun/Servers にサーバーをインストールした場合、管理サーバーとサーバーに設定されている各サーバーインスタンスの両方の ACL ファイルが、/usr/Sun/Servers/httpacl/ に格納されます。

主要な ACL ファイルの名前は generated-https-*server-id*.acl で、一時的な作業ファイルは genwork-https-*server-id*.acl という名前です。Sun ONE 管理サーバーを使用してアクセスを設定する場合は、これらの 2 つのファイルが作成されます。ただし、複雑な制約が必要な場合は、複数のファイルを作成し、server.xml ファイルから参照することができます。また、時刻や曜日を基準にしたサーバーへのアクセス制限など、ファイルを編集することによって利用可能になる機能もいくつかあります。

また、.acl ファイルを手動で作成して編集し、API を使用してアクセス制御をカスタマイズすることもできます。アクセス制御 API の使用の詳細は、『Programmer's Guide』を参照してください。

アクセス制御ファイルとその構文については、付録 C「ACL ファイルの構文」を参照してください。

ACL ユーザーキャッシュの設定

デフォルトでは、Sun ONE Web Server によるユーザーとグループの認証の結果が、ACL ユーザーキャッシュに保存されます。magnus.conf ファイルの ACLCacheLifetime 指令を使用して、ACL ユーザーキャッシュを有効にする期間を制御することができます。キャッシュのエントリが参照されるたびにその経過時間が計算され、ACLCacheLifetime と照合されます。経過時間が ACLCacheLifetime と同じか、それよりも長い場合、このエントリは使用されません。デフォルト値は 120 秒です。値を 0 (ゼロ) に設定すると、キャッシュが無効になります。この値に大きな値を使用する場合は、LDAP エントリを変更するたびに Sun ONE Web Server の起動が必要となる可能性があります。たとえば、この値を 120 秒に設定した場合は、Sun ONE Web Server と LDAP の同期が 2 分間に渡ってとられない可能性があります。LDAP ディレクトリが頻繁に変更される可能性が低い場合にだけ、大きな値を設定します。

magnus.conf の ACLUserCacheSize パラメータを使用すると、キャッシュ内に保存できるエントリの最大数を設定できます。このパラメータのデフォルト値は 200 です。新しいエントリがリストの先頭に追加され、キャッシュが最大サイズに達すると、新しいエントリを作成するために、このリストの最後のエントリが再利用されます。

また、magnus.conf に含まれるパラメータである ACLGroupCacheSize を使用して、ユーザーエントリごとにキャッシュできるグループメンバーの最大数を設定することができます。このパラメータのデフォルト値は 4 です。ただし、グループのメンバーではないユーザーはキャッシュされず、要求ごとに何回か LDAP ディレクトリにアクセスすることになります。

ACL ファイルの指令の詳細は、『NSAPI Programmer's Guide』を参照してください。

アクセス制御のしくみ

サーバーはページへの要求を受け取ると、ACL ファイルの規則を使用して、アクセスを許可するか拒否するか判断します。規則は、要求を送信しているコンピュータのホスト名や IP アドレスを参照できます。また、規則が LDAP ディレクトリに格納されているユーザーやグループを参照するように設定することもできます。

たとえば、次の ACL ファイルには、管理サーバー (admin-serv) に対する 2 つのデフォルトエントリと、「admin-reduced」グループ内のユーザーが管理サーバーの「Preferences」タブにアクセスできるようにするための追加エントリがあります。

```
version 3.0;
# The following "es-internal" rules protect files such
# as icons and images related to Sun ONE Web Server.
# These "es-internal" rules should not be modified.
  acl "es-internal";
  allow (read, list, execute,info) user = "anyone";
  deny (write, delete) user = "anyone";
# The following "default" rules apply to the entire document
# directory of Sun ONE Web Server. In this example, the rules
# are set up so that "all" users in the directory server are
# allowed to read, execute, list, and get information.
# The "all" users are not allowed to write to or delete any files.
# All clients that accesses the document directory of the web
# server will be required to submit a username and password
# since this example is using the "basic" method of
# authentication. A client must be in the directory server
# to gain access to this default directory since "anyone"
# not in the directory server is denied, and "all" in the
# directory server are allowed.
  acl "default";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");
  allow (read,execute,list,info)
  (user = "all");
# The following rules deny access to the directory "web"
```

```
# to everyone not in the directory server and deny everyone
# in the directory server who is not in GroupB.
# Only the users in GroupB are allowed read, execute, list,
# and info permissions. GroupA can not gain access to the
# directory "web" even though (in the ACL rule below) they
# can access the directory "my_stuff". Furthermore, members
# of GroupB can not write or delete files.
  acl "path=/export/user/990628.1/docs/my_stuff/web/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");

  allow (read,execute,list,info)
  (group = "GroupB");

# The following rule denies everyone not in the directory
# server and denies everyone in the directory server except
# user with the ID of "SpecificMemberOfGroupB". The ACL rule
# in this setting also has a requirement that the user
# connect from a specific IP address. The IP address setting
# in the rule is optional; it has been added to for extra
# security. Also, this ACl rule has a Customized prompt
# of "Presentation Owner". This Customized prompt appears
# in the username and password dialog box in the client's
# browser.

  acl
"path=/export/user/990628.1/docs/my_stuff/web/presentation.html"
;
  authenticate (user,group) {
     database = "default";
     method = "basic";
     prompt = "Presentation Owner";
  };
  deny (all)
  (user = "anyone" or group = "my_group");
  allow (all)
  (user = "SpecificMemberOfGroupB") and
  (ip = "208.12.54.76");

# The following ACL rule denies everyone not in the directory
# server and everyone in the directory server except for
# GroupA and GroupB access to the directory "my_stuff"
  acl "path=/export/user/990628.1/docs/my_stuff/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny (all)
  (user = "anyone");
  allow (read,execute,list,info)
  (group = "GroupA,GroupB");
```

たとえば、ユーザーが http://*server_name*/my_stuff/web/presentation.html という URL を要求したと仮定します。

Sun ONE Web Server はまず、サーバー全体へのアクセス制御を確認します。サーバー全体への ACL が認証の処理を続行するように設定される場合、サーバーは my_stuff ディレクトリへの ACL を確認します。ACL が存在する場合、サーバーは ACL 内の ACE を確認し、次のディレクトリに移動します。このプロセスは、アクセスを拒否する ACL が見つかるまで、または要求された URL の最後の ACL (この場合は、ファイル presentation.html) に達するまで続行します。

サーバーマネージャを使用してこの例のアクセス制御を設定するには、このファイルだけを対象とした ACL の作成のほか、ファイルへ誘導する各リソースの ACL を作成することができます。つまり、1 つはサーバー全体用、1 つは my_stuff ディレクトリ用、1 つは my_stuff/web ディレクトリ用、1 つはファイル用です。

注 一致する ACL が複数ある場合は、一致する最後の ACL 文を使用します。一致する最後の ACL 文は uri なので、default ACL はバイパスされます。

アクセス制御の設定

この節では、Web サイト上のファイルまたはディレクトリに対して、アクセスを制限するための手順を説明しています。すべてのサーバーに対してグローバルアクセス制御規則を設定することも、特定のサーバーに対して個別に設定することもできます。たとえば、人材管理部門を対象に、認証を受けているすべてのユーザーに各自の給与計算データを参照することは許可して、データを更新できるのは人材管理部門の給与計算担当者だけに制限する ACL を作成することができます。

管理サーバーによって、すべてのサーバーに対してグローバルにアクセス制御を設定することができます。次の節「アクセス制御オプションの選択」では、各オプションについて詳しく説明します。

注 グローバルアクセス制御を作成するには、分散管理を設定して有効にしておく必要があります。

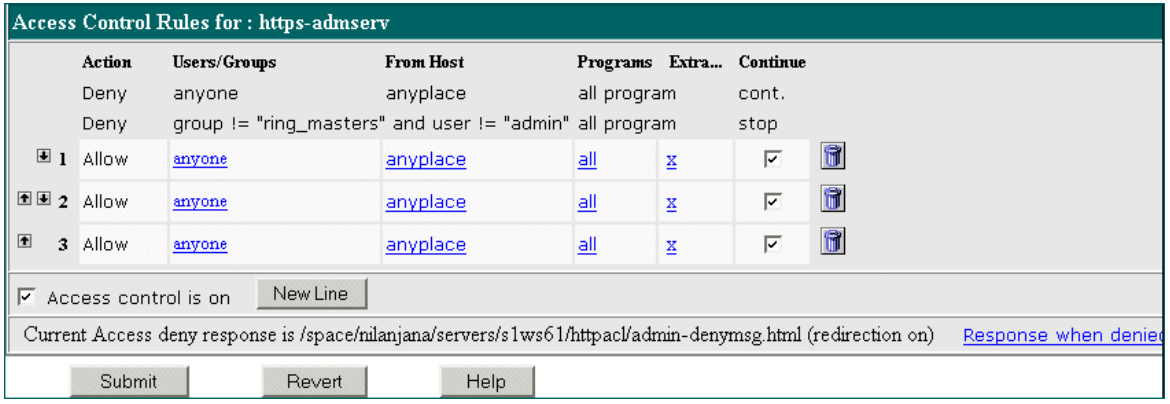
グローバルなアクセス制御の設定

すべてのサーバーに対してグローバルにアクセス制御を作成したり編集したりするには、次の手順を実行します。

1. 管理サーバーにアクセスして、「Global Settings」タブをクリックします。
2. 「Restrict Access」リンクをクリックします。
3. ドロップダウンリストから管理サーバー (https-admserv) を選択します。
4. 「Create ACL」ボタン、および「Go」ボタンをクリックします。

   「Access Control Rules for uri=/https-admserv/」ページが表示されます。

「Access Control Rules」ページ



管理サーバーには、編集できないデフォルトアクセス制御規則が 2 行あります。

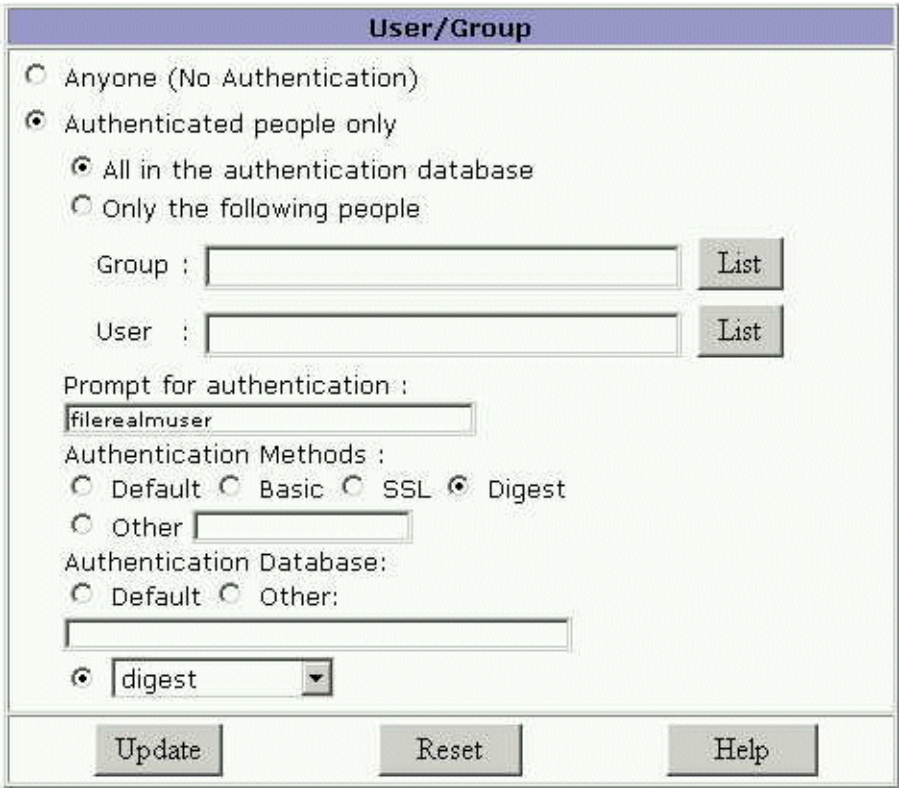
5. チェックマークが付いていない場合は、「Access control is on」チェックボックスにチェックマークを付けます。
6. テーブルの最下行にデフォルトの ACL 規則を追加するには、「New Line」ボタンをクリックします。

   アクセス制御制限を、その前のアクセス制御制限と置き換えるときは、上向き矢印をクリックします。

   アクセス制御制限を、その後のアクセス制御制限と置き換えるときは、下向き矢印をクリックします。
7. 「Users/Groups」列の「anyone」をクリックします。

   下のフレームに「User/Group」ページが表示されます。

「User/Group」ページ

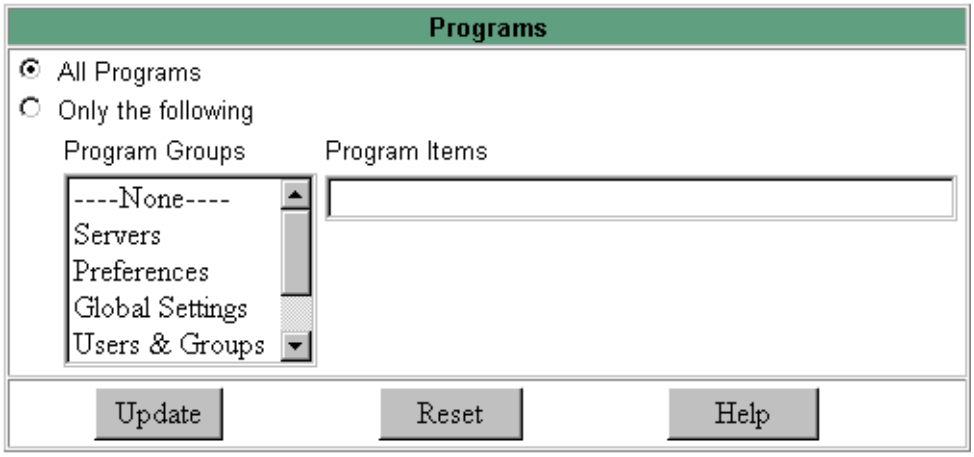


8. アクセスを許可するユーザーやグループを選択し、「Update」をクリックします。

   「Group」と「User」の「List」をクリックすると、選択肢のリストが表示されます。

9. 「From Host」列の「anyplace」をクリックします。
10. アクセスを許可するホスト名と IP アドレスを入力し、「Update」をクリックします。
11. 「Programs」列で「All Programs」をクリックします。

「Programs」



12. 「Program Groups」を選択するか、または「Program Items」フィールドにアクセスを許可する特定のファイル名を入力し、「Update」をクリックします。
13. (省略可能) 「Extra」列の「x」をクリックして、カスタマイズした ACL 式を追加します。
14. デフォルトとして選択されていない場合は、「Continue」列にチェックマークを付けます。

    サーバーは次の行を評価してから、ユーザーがアクセスを許可されているかどうかを判断します。複数の行を作成する場合は、より一般的な制限からより特殊な制限へと処理を行います。
15. (省略可能) 別の URL または URI の内容をユーザーに対して表示することを拒否する場合、「Response」をクリックします。
16. 絶対 URL または相対 URI へのパスを入力し、「update」をクリックします。
17. 「Submit」をクリックして、新しいアクセス制御規則を ACL ファイルに保存します。

注 「Revert」をクリックすると、作成したすべての設定が画面を表示した時点の内容に戻ります。

サーバーインスタンスに対するアクセス制御の設定

サーバーマネージャを使用すると、特定のサーバーインスタンスに対するアクセス制御の作成、編集、または削除を実行できます。

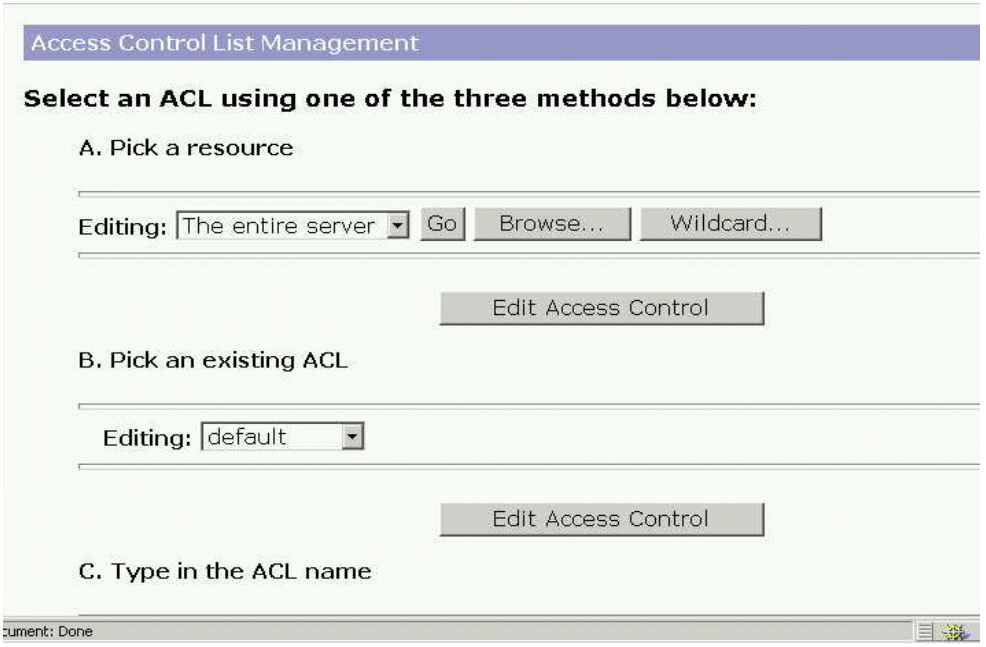
注	削除する場合、ACL ファイルからすべての ACL 規則を削除しないでください。サーバーを起動するには、最小限の ACL 規則が含まれる ACL ファイルが少なくとも 1 つ必要です。すべての ACL 規則を削除してサーバーを再起動すると、構文エラーが発生します。

サーバーインスタンスに対してアクセス制御を作成するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、ACL を作成または編集するサーバーインスタンスを選択します。
2. サーバーマネージャの「Preferences」タブを選択します。
3. 「Restrict Access」リンクをクリックします。
4. 「Option」列で、次のいずれかを選択します。
   - 「Add」を選択して、ACL ファイルの場所を入力します。
   - 「Edit」を選択し、ドロップダウンメニューから ACL ファイルを選択します。
   - ドロップダウンメニューから「Delete」を選択し、ACL ファイルを選択します。

   「Access Control List Management」ページに、次の 3 つのオプションが表示されます。

「Access Control List Management」ページ



5. 次のいずれかを選択します。

   - リソースを選択し、ファイルまたはディレクトリのワイルドカードパターン (`*.html` など) を指定し、制限するディレクトリまたはファイル名を選択するか、またはファイルやディレクトリを参照します。
   - 有効にしたすべての ACL のリストから選択する既存の ACL を選びます。有効にしていない既存の ACL は、このリストに表示されません。
   - ACL 名を入力すると、名前付きの ACL を作成できます。このオプションは、ACL ファイルについての知識が豊富な場合にだけ使用します。名前付きの ACL をリソースに適用する場合、手動で `obj.conf` を編集する必要があります。

   表 9-2 は、使用できるリソースワイルドカードと、その説明を示しています。

表 9-2 サーバーリソースのワイルドカード

リソースのワイルドカード	意味
デフォルト	インストール時に作成される名前付き ACL。LDAP ディレクトリ内のユーザーだけがドキュメントを発行できるように、書き込みアクセスを制限する

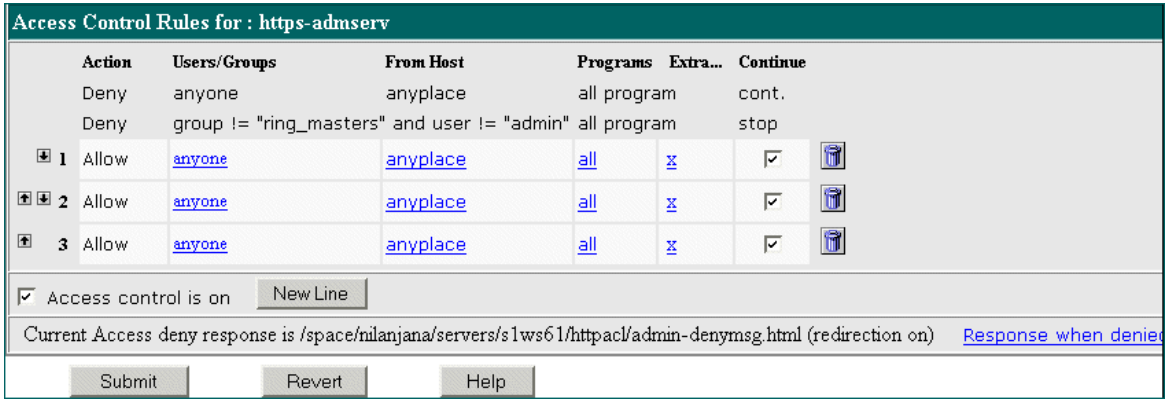
表 9-2　サーバーリソースのワイルドカード

リソースのワイルドカード	意味
サーバー全体	1 組の規則によって、稼動中の仮想サーバーを含めた Web サイト全体へのアクセスが定義される。仮想サーバーへのアクセスを制限するには、そのドキュメントルートのパスを指定すること
`/usr/sun/server4/docs/cgi-bin/*`	`cgi-bin` ディレクトリ内のすべてのファイルとディレクトリへのアクセスを制御する。絶対パスを指定する必要がある。Windows では、パスにドライブ文字を含める必要がある
`uri="/sales"`	ドキュメントルートの `sales` ディレクトリへのアクセスを制御する。URI を指定するには、名前付き ACL を作成すること

6. 「Edit Access Control」をクリックします。

   「Access Control Rules for: (サーバーインスタンス)」が表示されます。

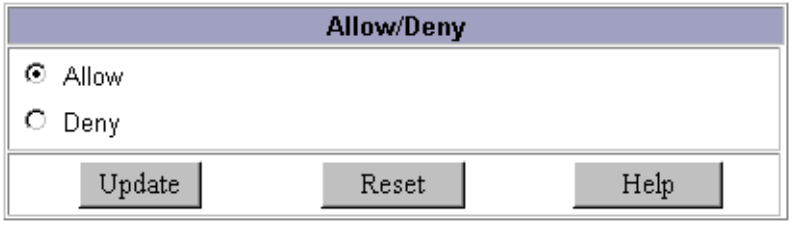
「Access Control Rules」ページ



7. チェックマークが付いていない場合は、「Access control is on」チェックボックスにチェックマークを付けます。
8. このサーバーインスタンス用の ACL を作成したり編集したりするには、「Action」列で「Deny」をクリックします。

   下のフレームに「Allow/Deny」ページが表示されます。

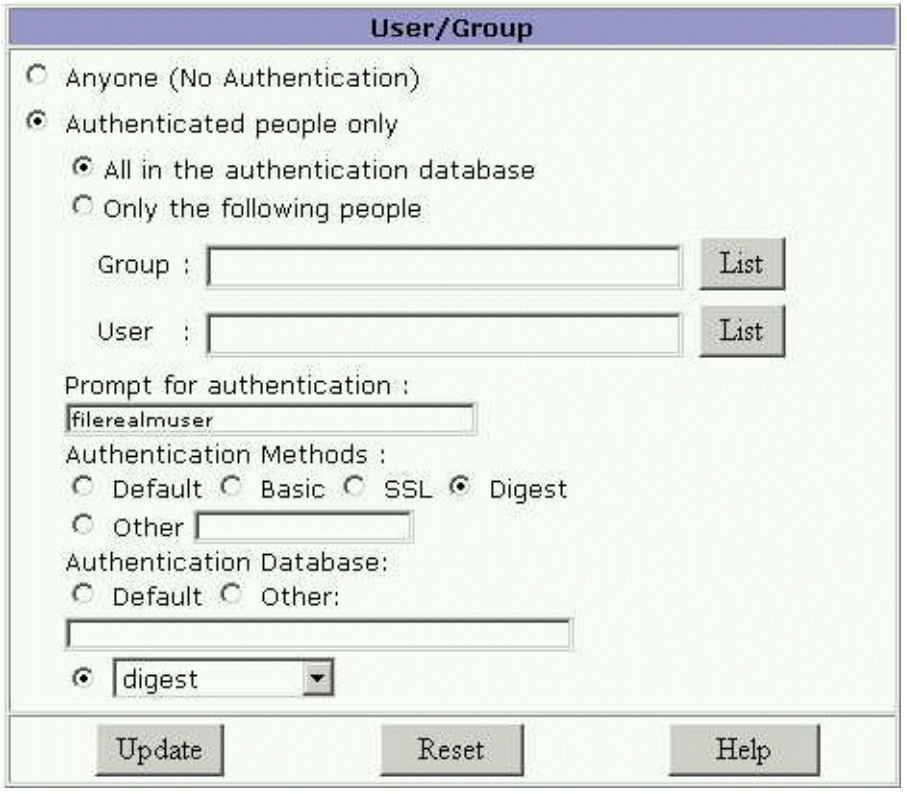
「Allow/Deny」ページ



9. デフォルトで選択されていない場合は「Allow」を選択し、「Update」をクリックします。
10. 「Users/Groups」列の「anyone」をクリックします。

    下のフレームに「User/Group」ページが表示されます。

「User/Group」ページ

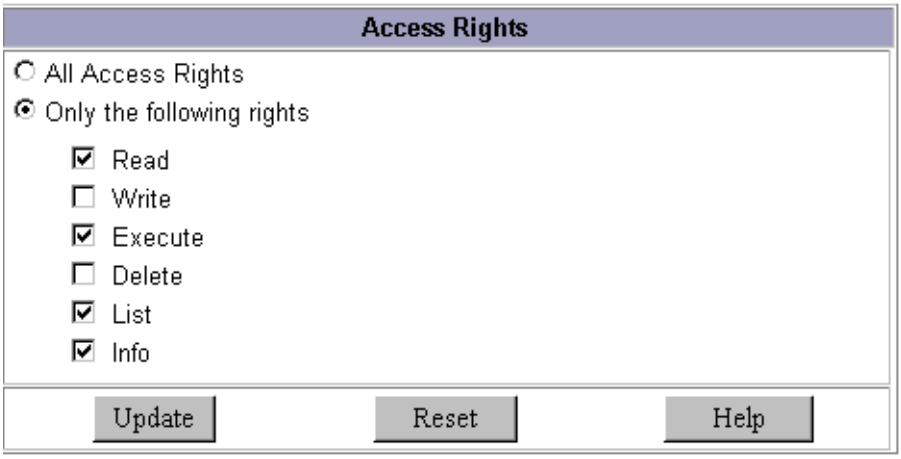


11. アクセスを許可するユーザーやグループを選択し、「Update」をクリックします。

    「Group」と「User」の「List」をクリックすると、選択肢のリストが表示されます。

12. 「From Host」列の「anyplace」をクリックします。
13. アクセスを許可するホスト名と IP アドレスを入力し、「Update」をクリックします。
14. 「Rights」列で「all」をクリックします。

「Access Rights」ページ



15. 次のどちらかを選択し、「Update」をクリックします。
    - All Access Rights (すべてのアクセス権)
    - 「Only the following rights」をクリックし、このユーザーに適したすべての権利にチェックマークを付けます。
16. (省略可能) 「Extra」列の「x」をクリックして、カスタマイズした ACL 式を追加します。
17. デフォルトとして選択されていない場合は、「Continue」列にチェックマークを付けます。

    サーバーは次の行を評価してから、ユーザーがアクセスを許可されているかどうかを判断します。複数の行を作成する場合は、より一般的な制限からより特殊な制限へと処理を行います。
18. (省略可能) 別の URL または URI の内容をユーザーに対して表示することを拒否する場合、「Response」をクリックします。
19. 絶対 URL または相対 URI へのパスを入力し、「Update」をクリックします。
20. 「Submit」をクリックして、新しいアクセス制御規則を ACL ファイルに保存します。

注 「Revert」をクリックすると、作成したすべての設定が画面を表示した時点の内容に戻ります。

21. 目的の各サーバーインスタンスに対して上記のすべての手順を繰り返し、アクセス制御を確立します。
22. 完了したら、「Apply」をクリックします。
23. 「hard start /restart」または「dynamically apply」を選択します。

また、仮想サーバーごとに ACL 設定を有効にすることもできます。この実行方法については、229 ページの「仮想サーバーのアクセス制御リストの編集」を参照してください。

アクセス制御オプションの選択

次の節では、アクセス制御を設定するときに選択できる個々のオプションについて説明します。管理サーバーの場合は、最初の 2 行はデフォルトとして設定されていて、編集できません。

アクションの設定

要求がアクション制御規則と一致する場合にサーバーが実行するアクションを指定できます。

- 「Allow」は、ユーザーまたはシステムが、要求されたリソースにアクセスできることを意味します
- 「Deny」は、ユーザーまたはシステムが、要求されたリソースにアクセスできないことを意味します

サーバーはアクセス制御式 (access control expressions、ACE) のリストを参照して、アクセス権を指定します。たとえば、最初の ACE は通常、すべてのユーザーを拒否します。最初の ACE に「continue」が設定されている場合、サーバーはリストの 2 番めの ACE を確認し、一致している場合は、次の ACE を確認します。「continue」チェックボックスにチェックマークが付いていない場合は、すべてのユーザーがリソースへのアクセスを拒否されます。サーバーは、一致しない ACE か、一致していても「continue」チェックボックスにチェックマークが付いていない ACE のどちらかに達するまでリストを参照します。一致する最後の ACE によって、アクセスが許可されるか拒否されるかが決まります。

ユーザーとグループの指定

ユーザーとグループの認証が行われる場合、ユーザーがアクセス制御規則で指定されているリソースにアクセスするには、ユーザー名とパスワードを入力する必要があります。

Sun ONE Web Server は、Sun ONE Directory Server などの LDAP サーバーに格納されているユーザーとグループのリストを確認します。

データベース内のすべてのユーザーに対してアクセスを許可または拒否することも、ワイルドカードパターンを使用して特定のユーザーに対してアクセスを許可または拒否することも、アクセスを許可または拒否する対象をユーザーとグループのリストから選択することもできます。

- 「Anyone (No Authentication)」はデフォルト値で、すべてのユーザーがユーザー名とパスワードを入力することなく、リソースにアクセスできることを意味します。ただし、ホスト名や IP アドレスなど、その他の設定に基づいてユーザーのアクセスが拒否される場合もあります。管理サーバーの場合、このオプションは、分散管理によって指定した管理者グループ内のすべてのユーザーがページにアクセスできることを意味します。
- 「Authenticated people only」
  - 「All in the authentication database」は、データベースにエントリがあるユーザーに一致します。
  - 「Only the following people」では、一致するユーザーとグループを指定できます。エントリをコンマ (,) で区切るか、またはワイルドカードパターンを使用すると、ユーザーとユーザーグループの任意のリストを作成することができます。また、データベースに格納されているユーザーやグループのリストから選択することもできます。「Group」は、指定したグループ内のすべてのユーザーに一致します。「User」は、指定した個々のユーザーに一致します。管理サーバーでは、ユーザーを分散管理のために指定した管理者グループのメンバーにする必要があります。
- 「Prompt for authentication」では、「authentication」ダイアログボックスに表示されるメッセージテキストを入力できます。このテキストを使用して、ユーザーが入力する必要のある項目について説明することができます。オペレーティングシステムによっては、最初の 40 文字程度しか表示されない場合があります。Netscape Navigator と Netscape Communicator では、ユーザー名とパスワードがキャッシュに保存され、プロンプトのテキストと関連付けられます。同じプロンプトがあるファイルやディレクトリにユーザーがアクセスする場合は、ユーザー名とパスワードをもう一度入力する必要はありません。特定のファイルやディレクトリに対してユーザーが再び認証を受けるようにしたい場合、必要な操作はそのリソースの ACL に対するプロンプトを変更するだけです。
- 「Authentication Methods」では、クライアントから認証情報を取得するためにサーバーで使用するメソッドを指定します。管理サーバーで使用できるのは、認証の基本メソッドだけです。

- 「Default」では、obj.conf ファイルで指定したデフォルトメソッドを使用します。obj.conf でメソッドが設定されていない場合は、「Basic」メソッドを使用します。「Default」チェックボックスにチェックマークを付けた場合、ACL 規則によって ACL ファイル内のメソッドが指定されることはありません。「Default」を選択すると、obj.conf ファイル内の 1 行を編集することによって、すべての ACL のメソッドを簡単に変更できます。
- 「Basic」では、HTTP メソッドを使用して、クライアントから認証情報を取得します。ユーザー名とパスワードが暗号化されるのは、サーバー側で暗号化するように設定されている場合だけです。
- 「SSL」では、クライアント証明書を使用してユーザーの認証を行います。このメソッドを使用するには、サーバー側で SSL を有効にする必要があります。暗号化するように設定されている場合は、基本メソッドと SSL メソッドを組み合わせて使用することができます。
- 「Digest」では、ユーザー名とパスワードをプレーンテキストとして送信することなく、ブラウザでユーザー名とパスワードに基づいて認証を行えるようにする認証機構を使用します。ブラウザは MD5 アルゴリズムを使用して、ユーザーのパスワードと Web サーバーによって提供される情報の一部を使用するダイジェスト値を作成します。このダイジェスト値は、サーバー側でダイジェスト認証プラグインを使用して計算され、クライアントによって提示されたダイジェスト値と比較されます。
- 「Other」では、アクセス制御 API を使用して作成するカスタム認証メソッドを使用します。

- 「Authentication Database」では、サーバーでユーザーの認証に使用するデータベースを選択します。このオプションを使用できるのは、サーバーマネージャを使用する場合だけです。「Default」を選択した場合、サーバーはデフォルトとして設定されているディレクトリサービス内のユーザーとグループを検索します。複数のデータベースを使用するように個々の ACL を設定する場合、「Other」を選択し、ドロップダウンリストでデータベースを選択します。デフォルト以外のデータベースと LDAP ディレクトリは、*server_root*/userdb/dbswitch.conf ファイルで指定されている必要があります。Oracle や Informix などのカスタムデータベースにアクセス制御 API を使用する場合は、「Other」を選択し、データベース名を入力します。

From Host の指定

どのコンピュータから要求されたかに基づいて、管理サーバーや Web サイトへのアクセスを制限できます。

- 「Anyplace」では、すべてのユーザーとシステムに対してアクセスを許可します。
- 「Only from」では、特定のホスト名または IP アドレスへのアクセスを制限できます。

「Only from」オプションを選択する場合は、「Host Names」フィールドまたは「IP アドレス」フィールドに、ワイルドカードパターンまたはコンマで区切ったリストを入力します。ホスト名に基づく制限は、IP アドレスに基づく制限よりも柔軟性があります。ユーザーの IP アドレスが変更された場合でも、このリストを更新する必要はありません。ただし、IP アドレスによる制限には、より高い信頼性があります。これは、接続されているクライアントを対象とした DNS 検索に失敗した場合、ホスト名によるアクセス制限が機能しなくなるためです。

コンピュータのホスト名または IP アドレスと一致するワイルドカードパターンとして使用できるのは、* というワイルドカード表記だけです。たとえば、指定ドメインのすべてのコンピュータに対してアクセスを許可するか、または拒否する場合、*.sun.com のように特定ドメイン内のすべてのホストと一致するワイルドカードパターンを指定します。管理サーバーにアクセスしているスーパーユーザーに対しては、その他のユーザーとは異なるホスト名と IP アドレスを設定することもできます。

ホスト名については、* が名前のコンポーネント全体を表現していなくてはなりません。つまり、*.sun.com は許容されますが、*users.sun.com は許容されません。また、ホスト名で * を使用する場合、この記号を文字列の一番左に使用する必要があります。たとえば、*.sun.com は許容されますが、users.*.com は許容されません。

IP アドレスについては、* がアドレスのバイト全体を表現していなくてはなりません。たとえば、198.95.251.* は許容されますが、198.95.251.3* は許容されません。IP アドレスで * を使用する場合、この記号を文字列の一番右に使用する必要があります。たとえば、198.* は許容されますが、198.*.251.30 は許容されません。

プログラムへのアクセス制限

プログラムへのアクセスを制限できるのは、管理サーバーだけです。プログラムへのアクセス制限を適用すると、特定のユーザーだけがサーバーマネージャのページを参照し、そのサーバーを設定できるように制限できます。たとえば、一部の管理者に対して管理サーバーの「Users & Groups」セクションを設定することを許可するものの、「Global Settings」へのアクセスは拒否するように制限することができます。

異なるユーザーが異なる機能ドメインにアクセスするように設定することもできます。選択したいくつかの機能ドメインへのアクセス権をユーザーに設定すると、そのユーザーのログイン後、そのユーザーにアクセスを許可した機能ドメインだけから管理サーバーページを表示できます。

- 「All Programs」では、すべてのプログラムへのアクセスを許可または拒否できます。デフォルトでは、管理者はサーバーのすべてのプログラムにアクセスできます。
- 「Only the following Program Groups」では、ユーザーのアクセスを許可するプログラムを指定できます。ドロップダウンリストからプログラムを選択します。Control キーを押しながらグループをクリックすると、複数のプログラムグループを選択できます。アクセスを制限できるプログラムグループは、次のとおりです。
  - None (デフォルト)
  - Servers
  - Preferences
  - Global Settings
  - Users & Groups
  - Security
  - Cluster Mgmt

  リストに表示されたプログラムグループは、管理サーバーのタブに反映されます。たとえば、「Preferences」や「Global Settings」などのタブに反映され、各ページへのアクセスを表します。管理者が管理サーバーにアクセスする場合、サーバーは管理者のユーザー名、ホスト、および IP を使用して、参照できるページを特定します。
- 「Program Items」では、「Program Items」フィールドにページ名を入力して、プログラムの特定のページへのアクセスを制御することができます。

アクセス権の設定

サーバーインスタンスに対するアクセス権を設定できるのは、サーバーマネージャを使用した場合だけです。アクセス権は、Web サイトのファイルやディレクトリへのアクセスを制限します。すべてのアクセス権の許可または拒否に加えて、一部のアクセス権の許可または拒否を行うための規則を指定することもできます。たとえば、ユーザーに対してファイルへの読み取り専用アクセスを許可することができます。この設定では、ユーザーは情報を参照することはできますが、ファイルを変更することはできません。

- 「All Access Rights」はデフォルトで、すべてのアクセス権を承認または拒否します。
- 「Only the following rights」では、許可、または拒否するアクセス権の組み合わせを選択できます。
  - 「Read」は、ユーザーにファイルの参照を許可します。これには、GET、HEAD、POST、および INDEX の HTTP メソッドが含まれます。
  - 「Write」は、ユーザーにファイルの変更や削除を許可します。これには PUT、DELETE、MKDIR、RMDIR、および MOVE の HTTP メソッドが含まれます。ファイルを削除するには、書き込み権と削除権の両方が必要です。
  - 「Execute」は、ユーザーに CGI プログラム、Java アプレット、エージェントなどのサーバー側アプリケーションの実行を許可します。
  - 「Delete」は、書き込み権を持つユーザーにファイルやディレクトリの削除を行う権限を与えます。
  - 「List」は、ユーザーに `index.html` ファイルが存在しないディレクトリ内のファイルリストへのアクセスを許可します。
  - 「Info」は、ユーザーに URI、たとえば `http_head` についての情報の取得を許可します。

カスタマイズされた式の作成

ACL には、カスタマイズした式を入力できます。このオプションは、ACL ファイルの構文や構造をよく理解している場合にだけ選択してください。ACL ファイルを編集するか、カスタマイズした式を作成する場合にだけ使用できる機能がいくつかあります。たとえば、時刻、曜日、またはその両方を基準として、サーバーへのアクセスを制限することができます。

次のカスタマイズされた式で、時刻や曜日によってアクセスを制限する方法を示します。この例では、LDAP ディレクトリに 2 つのグループがあることを前提としています。「regular」グループは、月曜日から金曜日までの午前 8 時から午後 5 時までアクセスできます。「critical」グループは、いつでもアクセスできます。

```
allow (read)
{
    (group=regular and dayofweek="mon,tue,wed,thu,fri");
    (group=regular and (timeofday>=0800 and timeofday<=1700));
    (group=critical)
}
```

有効な構文と ACL ファイルについては、付録 C「ACL ファイルの構文」および 478 ページの「obj.conf での ACL ファイルの参照」を参照してください。

アクセス制御の解除

「Access control is on」チェックボックスのチェックマークを外した場合、ACL 内のレコードを消去するかどうかを確認するプロンプトが表示されます。「OK」をクリックすると、サーバーは ACL ファイルから該当するリソースの ACL エントリを削除します。

ACL を無効にしたい場合、`generated-https-server-id.acl` ファイルの各行の先頭に # 記号を挿入することによって、ACL が記述された行をコメントにすることができます。

管理サーバーからアクセス制御を作成し、特定のサーバーインスタンスに対して有効に設定し、ほかのサーバーに対しては無効 (デフォルト) のままにしておくことができます。たとえば、管理サーバーからサーバーマネージャページへのアクセスをすべて拒否することができます。デフォルトでは、ほかのサーバーに対して、分散管理は有効に、アクセス制御は無効に設定されます。管理者は管理サーバーを設定することはできませんが、ほかのサーバーにアクセスして設定することはできます。

注	このアクセス制御は、管理者グループのユーザーに対して、分散管理のために設定されます。管理サーバーはまず、ユーザー (スーパーユーザーを除く) が管理者グループのメンバーであることを確認してから、アクセス制御規則を評価します。

アクセスが拒否された場合の応答

アクセスが拒否された場合、Sun ONE Web Server は「FORBIDDEN. Your client is not allowed access to the restricted object.」というデフォルトメッセージを出力します。別の応答メッセージを選択することもできます。また、アクセス制御オブジェクトごとに異なるメッセージを作成することもできます。

特定の ACL に送信されるメッセージを変更するには、次の手順を実行します。

1. 「ACL」ページの「Response when denied」リンクをクリックします。
2. 下のフレームの「Respond with the following」チェックボックスにチェックマークを付けます。
3. 絶対 URL または相対 URI へのパスを入力し、「Update」をクリックします。

   ユーザーにリダイレクト先の URL または URI へのアクセス権があることを確認します。
4. 「Update」をクリックします。
5. 上部フレームの「Submit」をクリックし、アクセス制御規則を送信します。

サーバーの一部へのアクセス制御

この節では、Web サーバーとその内容に対して一般的に使用されているアクセス制限について説明します。各制限の手順では、実行する必要のある操作を詳細に説明しています。ただし、197 ページの「サーバーインスタンスに対するアクセス制御の設定」で説明している手順も、すべて実行する必要があります。

この節で説明する制限を次に示します。

- サーバー全体に対するアクセス制限
- ディレクトリ (パス) へのアクセス制限
- URI (パス) へのアクセス制限
- ファイルタイプに対するアクセス制限
- 時刻に基づくアクセス制限
- セキュリティに基づくアクセス制限

サーバー全体に対するアクセス制限

呼び出されたグループ内のユーザーに対して、サブドメイン内のコンピュータからサーバーへのアクセスを許可したい場合があります。たとえば、ある会社のある部署のサーバーで、ネットワークの特定のサブドメインにあるコンピュータからのアクセスだけをユーザーに対して許可したい場合です。

サーバーインスタンスに対するアクセス制御の設定の節で説明した手順を使用して、次の操作を実行します。

1. サーバーマネージャを使用して、サーバーインスタンスを選択します。
2. 「Preferences」タブを選択します。
3. 「Restrict Access」リンクをクリックします。
4. 編集する ACL ファイルを選択します。
5. サーバーリソース全体を選択し、「Edit Access Control」をクリックします。
6. すべてのアクセスを拒否する、新しい規則を追加します。
7. 特定のグループへのアクセスを許可する、別の新しい規則を追加します。
8. アクセスを許可するコンピュータのホスト名として、ワイルドカードパターンを入力します。

   たとえば、「`*.employee.sun.com`」と入力します。
9. 「Continue」チェックボックスのチェックマークを外します。
10. 変更を送信して、適用します。

ディレクトリ (パス) へのアクセス制限

グループの所有者によって制御されているディレクトリおよびサブディレクトリにある、ファイルやアプリケーションの読み取りまたは実行をグループ内のユーザーに許可することができます。たとえば、プロジェクトマネージャは、参照するプロジェクトチームのステータス情報を更新できます。

サーバーのディレクトリへのアクセスを制限するには、サーバーインスタンスに対するアクセス制御の設定に関する節で説明した手順を使用して、次の操作を実行します。

1. サーバーマネージャを使用して、サーバーインスタンスを選択します。
2. 「Preferences」タブを選択します。
3. 「Restrict Access」リンクをクリックします。
4. 編集する ACL ファイルを選択します。

5. 「Pick a Resource」セクションを参照して、アクセスを制限するディレクトリを選択します。

   サーバーのドキュメントルート内のディレクトリが表示されます。選択すると、「Editing」ドロップダウンリストにディレクトリへの絶対パスが表示されます。

注	サーバールート内のすべてのファイルが表示されるようにする場合は、「Options」をクリックし、ディレクトリと「List files」チェックボックスにチェックマークを付けます。

6. 「Edit Access Control」をクリックします。
7. 新しい規則を作成し、デフォルト設定をそのまま使用して、すべての場所からのすべてのアクセスを拒否します。
8. 特定のグループのユーザーに対して、読み取り権と実行権だけを許可する別の新しい規則を作成します。
9. 3 行目を作成し、特定のユーザーに対してすべてのアクセス権を許可します。
10. 2 行目と 3 行目の「Continue」チェックボックスのチェックマークを外して、「Update」をクリックします。
11. 変更を送信して、適用します。

ファイルまたはディレクトリへの絶対パスが docroot ディレクトリに作成されます。ACL ファイルのエントリは、次のようになります。

```
acl "path=d:¥sun¥suitespot¥docroot1¥sales/";
```

URI (パス) へのアクセス制限

URI を使用して、Web サーバー上のシングルユーザーのコンテンツへのアクセスを制御することができます。URI は、サーバーのドキュメントルートディレクトリを始点とした、相対的なパスとファイル名です。サーバーのコンテンツのすべて、または一部 の名前 (たとえば、ディスクのボリューム名) を頻繁に変更したり削除したりする場合、URI を使用すると簡単です。また、ほかにドキュメントルートがある場合にも、URI を使用するとアクセス制御を簡単に行えます。

URI へのアクセスを制限するには、サーバーインスタンスに対するアクセス制御の設定に関する節で説明した手順を使用して、次の操作を実行します。

1. サーバーマネージャを使用して、サーバーインスタンスを選択します。
2. 「Preferences」タブを選択します。
3. 「Restrict Access」リンクをクリックします。

4. 「ACL name」セクションの「Type」に、アクセスを制限する URI を入力します。

   例 : `uri=/my_directory`

5. 「Edit Access Control」をクリックします。
6. すべてのユーザーに対して読み取りアクセスを許可する、新しい規則を作成します。
7. ディレクトリの所有者に対してアクセスを許可する、別の規則を作成します。
8. 1 番目の規則と 2 番目の規則の両方の「Continue」チェックボックスのチェックマークを外します。
9. 「Submit and Apply your changes」をクリックします。

ドキュメントルートに対して相対的な URI へのパスが作成されます。ACL ファイルのエントリは、次のようになります。

```
acl "uri=/my_directory";
```

ファイルタイプに対するアクセス制限

ファイルのタイプに基づいて、サーバーまたは Web サイトへのアクセスを制限することができます。たとえば、特定のユーザーだけに、サーバーで実行されるプログラムの作成を許可することができます。すべてのユーザーがプログラムを実行できますが、作成や削除を実行できるのはグループ内の指定されたユーザーだけです。

アクセスできるファイルタイプを制限するには、サーバーインスタンスに対するアクセス制御の設定に関する節で説明した手順を使用して、次の操作を実行します。

1. サーバーマネージャを使用して、サーバーインスタンスを選択します。
2. 「Preferences」タブを選択します。
3. 「Restrict Access」リンクをクリックします。
4. 「Pick a resource」セクションで「Wildcard」をクリックし、ワイルドカードパターンを入力します。

   例 : `*.cgi`

5. 「Edit Access Control」をクリックします。
6. すべてのユーザーに対して読み取りアクセスを許可する、新しい規則を作成します。
7. 指定されたグループだけに読み取りアクセスと削除アクセスを許可する、別の規則を作成します。
8. 変更を送信して、適用します。

ファイルタイプの制限については、両方の「Continue」チェックボックスのチェックマークを付けたままにします。ファイルが要求されると、サーバーはまず、ACL のファイルタイプを確認します。

Pathcheck 関数は obj.conf 内に作成されます。この関数には、ファイルまたはディレクトリのワイルドカードパターンが含まれる場合あります。ACL ファイルのエントリは、次のようになります。

```
acl "*.cgi";
```

時刻に基づくアクセス制限

指定した日の指定した時間に、サーバーに対する読み取りアクセスと削除アクセスを制限することができます。この制限を使用して、ファイルがアクセスされている可能性がある勤務時間中には、ユーザーがファイルを変更したり削除したりできないようにすることができます。

時刻を基にしてアクセスを制限するには、サーバーインスタンスに対するアクセス制御の設定に関する節で説明した手順を使用して、次の操作を実行します。

1. サーバーマネージャを使用して、サーバーインスタンスを選択します。
2. 「Preferences」タブを選択します。
3. 「Restrict Access」リンクをクリックします。
4. 「Pick a Resource」ドロップダウンリストから「entire server」を選択して、「Edit Access Control」をクリックします。
5. すべてのユーザーに対して読み取り権と実行権を許可する、新しい規則を作成します。

   これは、ユーザーがファイルやディレクトリの追加、更新、または削除を行う場合にはこの規則が適用されず、サーバーは該当する別の規則を検索することを意味します。
6. すべてのユーザーに対して書き込み権と削除権を拒否する、別の規則を作成します。
7. 「X」リンクをクリックして、カスタマイズされた式を作成します。
8. アクセスを許可する曜日と時刻を入力します。

   その例を次に示します。

```
user = "anyone" and
dayofweek = "sat,sun" or
(timeofday >= 1800 and
timeofday <= 600)
```

カスタム式を作成すると、「Unrecognized expressions」メッセージが「Users/Groups」フィールドと「From Host」フィールドに表示されます。

9. 変更を送信して、適用します。

カスタマイズした式にエラーがあると、エラーメッセージが生成されます。修正してから、もう一度送信してください。

セキュリティに基づくアクセス制限

Sun ONE Web Server 6.1 では、1 つのサーバーインスタンスに SSL を使用する待機ソケットと SSL を使用しない待機ソケットを設定することができます。セキュリティに基づく制限を使用すると、セキュリティ保護されたチャネルを経由して送信する必要のあるリソースを保護できます。

セキュリティに基づいてアクセスを制限するには、サーバーインスタンスに対するアクセス制御の設定に関する節で説明した手順を使用して、次の操作を実行します。

1. サーバーマネージャを使用して、サーバーインスタンスを選択します。
2. 「Preferences」タブを選択します。
3. 「Restrict Access」リンクをクリックします。
4. 「Pick a Resource」ドロップダウンリストから「entire server」を選択して、「Edit Access Control」をクリックします。
5. すべてのユーザーに対して読み取り権と実行権を許可する、新しい規則を作成します。

   これは、ユーザーがファイルやディレクトリの追加、更新、または削除を行う場合にはこの規則が適用されず、サーバーは該当する別の規則を検索することを意味します。
6. すべてのユーザーに対して書き込み権と削除権を拒否する、別の規則を作成します。
7. 「X」リンクをクリックして、カスタマイズされた式を作成します。
8. 「`ssl="on"`」と入力します。

   その例を次に示します。

```
user = "anyone" and ssl="on"
```

9. 変更を送信して、適用します。

カスタマイズした式にエラーがあると、エラーメッセージが生成されます。修正してから、もう一度送信してください。

分散管理によるアクセス制御のセキュリティ保護

ここでは、分散管理を有効にした後で、Sun ONE Web Server 6.1 でアクセス制御でセキュリティを保護するために必要な追加タスクについて説明します。

- リソースへのアクセスのセキュリティ保護
- サーバーインスタンスへのアクセスのセキュリティ保護
- IP ベースのアクセス制御の有効化

リソースへのアクセスのセキュリティ保護

`generated.https-`*server-id*`.acl` ファイルの `https-`*server-id* オブジェクトタグに指定されている PathCheck 指令の実行順序によって、リソースに対して不適切なアクセス権が与えられてしまうことがあります。これを防止するには、`<`*server-root*`>/generated.https-`*server-id*`.acl` をアクセス制御が必要なプログラムグループをコンマで区切って次のように指定します。

```
allow (all)
user=<username> and program=<program group, program group...>;
```

上の行の下に、次の行を追加します。

```
deny absolute (all)
user=<username> and program!=<program group, program group...>;
```

サーバーインスタンスへのアクセスのセキュリティ保護

サーバーインスタンスへのアクセスを制御するように Sun ONE Web Server 6.1 を設定するには、`<`*server-root*`>/httpacl/*.https-admserv.acl` ファイルを編集して、アクセス制御権限を与えるユーザーを指定します。その例を次に示します。

```
acl "https-<instance>";
authenticate (user,group) {
database = "default";
method = "basic";
};
deny absolute (all) user != "UserA";
```

IP ベースのアクセス制御の有効化

アクセス制御エントリが、管理サーバーに関連する ACL ファイル (`gen*.https-admserv.acl`) の `ip` 属性を参照する場合、次の手順 1、2 を実行してください。

アクセス制御エントリが、サーバーインスタンスに関連する ACL ファイルの `ip` 属性を参照する場合は、その ACL について次の手順 1 だけを実行してください。

1. *<server-root>*`/httpacl/gen*.https-admserv.acl` ファイルを次のように編集して、`user` と `group` に加えて、`ip` を認証リストに追加します。

   ```
   acl "https-admserv";

   authenticate (user,group,ip) {

   database = "default";

   method = "basic";

   };
   ```

2. 次のアクセス制御エントリを追加します。

   ```
   deny absolute (all) ip !="ip_for_which_access_is_allowed";
   ```

   その例を次に示します。

   ```
   acl "https-admserv";

   authenticate (user,group,ip) {

   database = "default";

   method = "basic";

   };

   deny absolute (all) ip !="205.217.243.119";
   ```

ダイナミックアクセス制御ファイルの使用

サーバーのすべてのコンテンツが 1 人のユーザーによって管理されることはほとんどありません。このため、Sun ONE Web Server へのアクセス権を与えることなく、一般ユーザーが必要な設定を行えるように、設定オプションのサブセットへのアクセスを許可することが必要な場合があります。設定オプションのサブセットは、ダイナミック設定ファイルに保存されます。

この節で説明する内容を次に示します。

- .htaccess ファイルの使用

- サポートされる .htaccess 指令
- .htaccess セキュリティに関する注意事項

.htaccess ファイルの使用

Sun ONE Web Server は、ダイナミック設定ファイルである .htaccess をサポートします。ユーザーインタフェースから、または設定ファイルを手動で変更することによって、.htaccess ファイルを有効にすることができます。.htaccess をサポートするファイルは、*server_root*/plugins/htaccess ディレクトリにあります。これらのファイルには、.htaccess ファイルと、.nsconfig ファイルを .htaccess ファイルに変換するためのスクリプトを使用できるようにするプラグインが含まれています。

.htaccess ファイルは、サーバーの標準アクセス制御と組み合わせて使用することができます。標準アクセス制御は、PathCheck 指令の順序に関係なく、必ず .htaccess アクセス制御の前に適用されます。ユーザー - グループ認証が「基本」の場合、ユーザー認証には標準アクセス制御と .htaccess アクセス制御の両方は必要ありません。標準サーバーアクセス制御を経由して SSL クライアント認証を使用でき、.htaccess ファイルを経由して HTTP「基本」認証を要求することもできます。

この節では、次の内容について説明します。

- ユーザーインタフェースからの .htaccess の有効化
- magnus.conf からの .htaccess の有効化
- 既存の .nsconfig ファイルの .htaccess ファイルへの変換
- htaccess-register の使用
- .htaccess ファイルの例

ユーザーインタフェースからの .htaccess の有効化

.htaccess を使用するように Sun ONE Web Server を設定するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、.htaccess を有効にするサーバーインスタンスを選択します。
2. 画面の一番上にある「Class Manager」リンクをクリックします。
3. 「Content Mgmt」タブを選択します。
4. .htaccess Configuration」リンクをクリックします。
5. 次の方法で、編集するサーバーを選択します。
    - ドロップダウンリストからサーバー全体または特定のサーバーを選択します。

- 「Browse」をクリックして、編集するディレクトリやファイルを選択します。
- 「Wildcard」をクリックして、編集するワイルドカードパターンを選択します。

6. 「Yes」を選択して、.htaccess を有効にします。
7. htaccess の設定を追加するファイルの名前を入力します。
8. 「OK」をクリックします。
9. 完了したら、「Apply」をクリックします。
10. 「hard start /restart」または「dynamically apply」を選択します。

magnus.conf からの .htaccess の有効化

.htaccess を使用するサーバーを手動で有効にするにはまず、プラグインを読み込んで初期化してから起動するように、サーバーの magnus.conf ファイルを修正する必要があります。

1. *server_root*/https-*server_name*/config ファイルの magnus.conf を開きます。
2. ほかの Init 指令のあとに、必要な行を追加します。
   - UNIX/Linux の場合は、次の行を追加します。

     ```
     Init fn="load-modules" funcs="htaccess-init,htaccess-find"
     shlib="server_root/plugins/htaccess/htaccess.so"
     NativeThread="no"
     Init fn="htaccess-init"
     ```

   - Windows の場合は、次の行を追加します。

     ```
     Init fn="load-modules"
     funcs="htaccess-init,htaccess-find,htaccess-register"
     shlib="server_root/plugins/htaccess/htaccess.dll"
     NativeThread="no"
     Init fn="htaccess-init"
     ```

   - HP の場合は、次の行を追加します。

     ```
     Initfn="load-modules"
     funcs="htaccess-init,htaccess-find,htaccess-register"
     shlib="<server_root>/plugins/htaccess/htaccess.sl"
     NativeThread="no"

     Init fn="htaccess-init"
     ```

3. (省略可能) 最後の行を次のように編集します。

   ```
   Init fn="htaccess-init"[groups-with-users=yes]
   ```

4. 「File」の「Save」をクリックします。
5. obj.conf を開きます。

6. オブジェクトの最後の指令として、PathCheck 指令を追加します。

   a. 仮想サーバーによって管理されるすべてのディレクトリに対して .htaccess ファイルの処理を有効にするには、object.conf ファイルのデフォルトオブジェクトに PathCheck 指令を追加します。

   ```
   <Object name="default">
   ...
   PathCheck fn="htaccess-find"
   </Object>
   ```

   .htaccess の処理をオブジェクトの最後の PathCheck 指令にする必要があります。

   b. サーバー上の特定のディレクトリを対象とした .htaccess ファイルの処理を有効にするには、magnus.conf 内の対応する定義に PathCheck 指令を配置します。

7. .htaccess ファイルに .htaccess 以外の名前を付けるには、次の形式を使用して PathCheck 指令でファイル名を指定する必要があります。

   ```
   PathCheck fn="htaccess-find" filename="filename"
   ```

注	次に管理サーバーを使用するとき、手動で修正が行われたことを知らせる警告メッセージが表示されます。「Apply」をクリックすると、変更が有効になります。

それ以降のサーバーへのアクセスは、指定されたディレクトリでの .htaccess によるアクセス制御の対象となります。たとえば、.htaccess ファイルへの書き込みアクセスを制限するには、対象ファイルの設定スタイルを作成し、その設定スタイルに対してアクセス制御を適用します。詳細は、第 17 章「設定スタイルの適用」を参照してください。

既存の .nsconfig ファイルの .htaccess ファイルへの変換

Sun ONE Web Server 6.1 には、旧バージョンで使用していた既存の .nsconfig ファイルを .htaccess ファイルに変換するための htconvert プラグインが組み込まれています。Sun ONE Web Server 6.1 では、.nsconfig ファイルはサポートされなくなりました。このため、これまで .nsconfig ファイルを使用していた場合は、.htaccess ファイルに変換する必要があります。

htconvert が有効になっている場合、pfx2dir 指令と document-root 指令に対して、指定された server.xml ファイルが検索されます。検出された各 .nsconfig ファイルは、.htaccess ファイルに変換されます。設定によっては、複数の obj.conf ファイルを変換できます。

注 既存の .htaccess ファイルがある場合、htconvert によって htaccess.new ファイルが生成され、警告メッセージが表示されます。.htaccess と .htaccess.new がすでに存在している場合、新しいファイルは .htaccess.new.new という名前になります。つまり .new は繰り返し追加されます。

現在、htconvert プラグインは RestrictAccess 指令と RequireAuth 指令、および <Files> ラッパーだけをサポートしています。<Files*> 以外の <Files> がある場合は、警告メッセージが表示され、スクリプトはディレクトリ内のすべてのファイルへのアクセスが制御されるように動作します。

ファイルを変換するには、コマンドプロンプトで、使用しているシステムの Perl へのパス、プラグインスクリプトへのパス、server.xml ファイルへのパスを入力します。その例を次に示します。

```
server_root¥install¥perl server_root/plugins/htaccess/htconvert
server_root/https-server_name/config/server.xml
```

すべての .nsconfig ファイルが .htaccess ファイルに変換されますが、変換元のファイルが削除されることはありません。

groups-with-users オプションによって、多数のユーザーをグループとして処理できます。多数のユーザーがグループのメンバーとなっている場合は、次の手順を実行します。

1. ユーザーファイルの書式を修正して、ユーザーがメンバーとなっているすべてのグループのリストを表示します。

   ```
   username:password:group1,group2,group3,...groupn
   ```

2. AuthGroupFile 指令を修正して、AuthUserFile と同じファイルを指定します。

また、次のように実行することもできます。

1. AuthGroupFile 指令全体を削除します。
2. magnus.conf ファイルの Init fn=htaccess-init 行に、次の内容を追加します。

   ```
   groups-with-users="yes"
   ```

htaccess-register の使用

htaccess-register は、認証メソッドを独自に作成するための新しい関数です。Apache と同様に、外部認証モジュールを作成し、htaccess-register を使用して .htaccess モジュールに組み込むことができます。*server_root*/plugins/nsapi/htaccess に 2 つのサンプルモジュールがあります。

外部モジュールを使用すると、1 つまたは複数の新しい指令を作成できます。たとえば、認証のためのユーザーデータベースを指定できます。ただし、指令を <Limit> タグまたは <LimitExcept> タグで囲むことはできません。

.htaccess ファイルの例

次に、.htaccess ファイルの例を示します。

```
<Limit GET POST>
order deny,allow
deny from all
allow from all
</Limit>
<Limit PUT DELETE>
order deny,allow
deny from all
</Limit>
AuthName mxyzptlk.kawaii.com
AuthUserFile /server_root/mxyz-docs/service.pwd
AuthGroupFile /server_root/mxyz-docs/service.grp
```

サポートされる .htaccess 指令

このバージョンでは、次の .htaccess 指令がサポートされます。

allow

構文

Allows from host: host は次のいずれかです。

- すべてのクライアントホストからのアクセスを許可する場合、host は all
- DNS ホスト名のすべてまたは最後の部分
- IP アドレス全体またはその一部

<Limit> または <LimitExcept> で囲む必要はありませんが、通常は囲まれています。

効果
指定したホストに対してアクセスを許可します。通常、<Limit> タグで囲まれています。

deny

構文
Deny from host: host は次のいずれかです。

- すべてのクライアントホストからのアクセスを拒否する場合、host は all
- DNS ホスト名のすべてまたは最後の部分
- IP アドレス全体またはその一部

<Limit> タグまたは <LimitExcept> タグで囲む必要はありませんが、通常は囲まれています。

効果
指定したホストに対してアクセスを拒否します。通常、<Limit> タグで囲まれています。

AuthGroupFile

構文
AuthGroupFile filename: filename は、groupname: user user という形式でグループ定義が含まれるファイルの名前です。

<Limit> タグや <LimitExcept> タグで囲むことはできません。

効果
指定されたグループファイルが、require group 指令で参照されるグループ定義で使用されることを示します。AuthGroupFile 指令で指定されたファイル名が AuthUserFile 指令で指定されたファイル名と同じである場合、このファイルには次の形式でユーザーとグループが含まれると想定されることに注意してください。

```
username:DES-encrypted-password:comma-separated-list-of-groups
```

AuthUserFile

構文
AuthUserFile filename: 次のように指定します。

- filename は、username:password という形式でユーザー定義が含まれるファイルの名前
- username はユーザーのログイン名で、password は DES で暗号化されたパスワード

<Limit> タグや <LimitExcept> タグで囲むことはできません。

効果
指定されたユーザーファイルが、require user 指令または require valid-user 指令で参照されるユーザー名で使用されることを示します。

obj.conf の Init fn=htaccess-init 指令で groups-with-users=yes を使用したり、同じファイル名で AuthGroupFile 指令を指定したりすると、そのファイルが次の形式であると想定されることに注意してください。

```
username:DES-encrypted-password:comma-separated-list-of-groups
```

AuthName

構文
AuthName authentication realm: authentication realm は、ユーザー認証の要求に関連付けられる認証領域を指定する文字列です。

<Limit> タグや <LimitExcept> タグで囲むことはできません。

効果
authentication realm 文字列は、通常、クライアント側のユーザー名とパスワードのプロンプトに表示されます。クライアントのユーザー名とのパスワードのキャッシュへの保存に影響を与える場合があります。

AuthType

構文
AuthType Basic: <Limit> タグや <LimitExcept> タグで囲むことはできません。

効果
ユーザー認証メソッドとして、現在サポートされている唯一のメソッドである HTTP 基本認証を指定します。

<Limit>

構文

```
<Limit method method ...>
allow, deny, order, or require directives
</Limit>
```

method は GET、POST、PUT などの HTTP メソッドです。ここでは、Web サーバーに対して使用できるすべてのメソッドを使用できます。

効果

指定された HTTP メソッドを使用する要求に対してだけ、タグで囲まれている指令が適用されます。

<LimitExcept>

構文

```
<LimitExcept method method ...>
allow, deny, order, or require directives
</LimitExcept>
```

method は GET、POST、PUT などの HTTP メソッドです。ここでは、Web サーバーに対して使用できるすべてのメソッドを使用できます。

効果

指定された HTTP メソッドと一致しないタイプの要求に対してだけ、タグで囲まれている指令が適用されます。

order

構文

Order ordering。ordering は次のいずれかです。

- allow、deny
- deny、allow
- mutual-failure

`<Limit>` または `<LimitExcept>` で囲む必要はありませんが、通常は囲まれています。

効果

- allows、denies、evaluates は指令を許可し、そのあと、指令を拒否する
- denies、allows、evaluates は指令を拒否し、そのあと、指令を許可する
- mutual-failure は順序に関係なく、allow 指令と deny 指令の両方でリストに表示されているホストに対するアクセスを拒否する

require

構文

- require group groupname groupname
- require user username username
- require valid-user

`<Limit>` または `<LimitExcept>` で囲む必要はありませんが、通常は囲まれています。

効果

- require group は、認証を受けるユーザーが、指定したグループのいずれかのメンバーであることを要求する
- require user は、認証を受けるユーザーが、指定したユーザーのいずれかであることを要求する
- require valid-user は、認証されたユーザーを要求する

.htaccess セキュリティに関する注意事項

デフォルトでは、サーバーによる HTTP PUT のサポートが無効になっています。クラスマネージャの「Content Mgmt」の「Remote File Manipulation」ページを使用して、HTTP PUT を有効にすることができます。`.htaccess` ファイルが保存されているディレクトリへの PUT アクセスを許可する場合、このファイル自体の置換も許可することになるため、細心の注意が必要です。アクセスを制限することによって、ディレクトリ内のすべてのファイルに対する PUT アクセスを防止することができます。211 ページの「ディレクトリ (パス) へのアクセス制限」を参照してください。

仮想サーバーへのアクセス制御

Sun ONE Web Server 6.1 のアクセス制御についての情報は、各仮想サーバーの ACL ファイルと、ドキュメントディレクトリの .htaccess ファイルから入手できます。.htaccess システムは iPlanet Web Server 4.x から変更されていません。

server.xml ファイルには、特定の標準 Sun ONE Web Server 6.x ACL ファイルに関連付けられた ID を定義する、1 つまたは複数の ACLFILE タグが含まれています。その例を次に示します。

```
<ACLFILE id="standard" file="standard.acl">
```

アクセス制御を使用する仮想サーバーの場合は、「aclids」プロパティに 1 つまたは複数の ACL ファイル ID への参照を作成する必要があります。その例を次に示します。

```
<VS aclids="standard">
```

この設定では、複数の仮想サーバーで同じ ACL ファイルを共有することができます。仮想サーバーに対するユーザー - グループ認証を要求する場合は、その定義に 1 つまたは複数の USERDB タグを追加する必要があります。このような USERDB タグは、ACL ファイル内のデータベース名と dbswitch.conf 内の実際のデータベースを関連付けます。

次の例では、「database」属性のない ACL を dbswitch.conf の「default」データベースにマッピングします。

```
<VS>
    <USERDB id="default" database="default"/>
</VS>
```

仮想サーバーからデータベースへのアクセス

dbswitch.conf ファイルで、ユーザー認証データベースをグローバルに定義できます。このファイルは、サーバーの起動時に読み込まれます。

dbswitch.conf 内の LDAP URL の baseDN は、データベースへのすべてのアクセスのグローバルルートを定義します。これによって、下位互換が維持されます。最新のインストールでは、baseDN には何も指定されません。

dcsuffix は dbswitch.conf 内の LDAP データベースの新しい属性で、Sun ONE LDAP スキーマに従って DC ツリーのルートを定義します。これは、LDAP URL の baseDN に対する相対位置になります。dcsuffix 属性が存在する場合、LDAP データベースは Sun ONE LDAP スキーマに準拠し、動作の一部が変更されます。Sun ONE LDAP スキーマの詳細と例については、『Sun ONE Web Server 6.1 Administrator's Configuration Reference』の第 2 章に記載されている「The Sun ONE LDAP Schema」を参照してください。

仮想サーバーごとに、ディレクトリの 1 つを指定する 1 つまたは複数の USERDB ブロックを定義できます。また、追加情報を定義することもできます。USERDB ブロック ID は、ACL のデータベースパラメータから参照することができます。仮想サーバーに USERDB ブロックがない場合、ユーザーまたはグループを基準にした ACL は失敗します。

USERDB タグは、ACL のデータベース属性と dbswitch.conf の間の間接参照の追加層を定義します。間接参照のこの層では、仮想サーバーの管理者がアクセスするデータベースをサーバー管理者が完全に制御するために必要な保護を追加します。

USERDB の詳細については、『Sun ONE Web Server 6.1 Administrator's Configuration Reference』の第 2 章に記載されている「User Database Selection」を参照してください。

ユーザーインタフェースでの LDAP データベースの指定

dbswitch.conf で 1 つまたは複数のユーザー認証データベースを定義すると、クラスマネージャを使用して、各仮想サーバーが認証のためにどのデータベースを使用するかを設定できるようになります。また、クラスマネージャを使用して、仮想サーバーでの認証のために dbswitch.conf での設定に対して新しく作成したデータベース定義を追加することができます。

LDAP データベースまたは仮想サーバーで使用するデータベースを指定するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Virtual Server Class」タブを選択します。
2. 「Tree View of the Server」のリストから、LDAP データベースを指定したい仮想サーバークラスのリンクをクリックします。
3. 表示されていない場合は、「Virtual Servers」タブを選択します。
4. 「ACL Settings」リンクをクリックします。

   「ACL Settings for Virtual Servers」ページが表示されます。
5. 表示されていない場合は、「Option」列のドロップダウンリストから「Edit」を選択します。
6. 編集している仮想サーバーの「Database」列でドロップダウンリストからデータベース設定を選択します。

7. 「OK」をクリックします。
8. 「Edit ACL Files」ウィンドウを閉じます。
9. 「Apply」をクリックします。
10. 「dynamically apply」を選択します。

仮想サーバーのアクセス制御リストの編集

仮想サーバーの ACL は、仮想サーバーがあるサーバーインスタンスに対して作成されます。仮想サーバーの ACL 設定は、サーバーインスタンスに対して作成される ACL 設定のデフォルトとなります。ただし、各仮想サーバーのアクセス制御はクラスマネージャから編集できます。また、このメソッドを使用して、新しく作成された ACL ファイルを仮想サーバーに追加します。

仮想サーバーの ACL 設定を編集するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Virtual Server Class」タブを選択します。
2. 「Tree View of the Server」のリストから、LDAP データベースを指定したい仮想サーバークラスのリンクをクリックします。
3. 表示されていない場合は、「Virtual Servers」タブを選択します。
4. 「ACL Settings」リンクをクリックします。
5. 変更する仮想サーバーの「Option」フィールドのドロップダウンリストから「Edit」または「Delete」を選択します。
6. 「ACL File」フィールドの「Edit」リンクをクリックすると、使用できる ACL ファイルが表示されます。
7. 仮想サーバーに追加または削除する 1 つまたは複数の ACL ファイルを選択します。

   仮想サーバーには複数のドキュメントルートが存在する場合があるため、複数の ACL ファイルが存在する可能性があります。
8. ドロップダウンリストから、ACL リストに関連付けるデータベースを選択します。
9. (必要に応じて) BaseDN を入力します。
10. 変更が終わったら、「OK」をクリックします。
11. 「Apply」をクリックします。
12. 「dynamically apply」を選択します。

ファイルベースの認証用 ACL の作成

Sun ONE Web Server 6.1 は、ファイルベース認証データベースの使用をサポートしています。このデータベースには、ユーザーとグループに関する情報がテキスト形式のフラットファイルとして格納されます。ACL のフレームワークは、ファイル認証データベースを利用できるように設計されています。

注 Sun ONE Web Server は、ダイナミックフラットファイルをサポートしていません。フラットファイルデータベースは、サーバーの起動時に読み込まれます。ファイルに加えた変更は、サーバーを再起動した場合にだけ適用されます。

ACL エントリは、database キーワードを使用してユーザーデータベースを参照できます。その例を次に示します。

```
acl "default";
    authenticate (user) {
...
    database="myfile";
...
};
```

myfile データベースは、server-root/userdb/dbswitch.conf ファイル内の対応する定義とリンクしている server.xml 内の VS の USERDB 要素中で参照するようにできます。その例を次に示します。

```
<VS>
...
    <USERDB id="myfile" database="myfiledb">
...
</VS>
```

server-root/userdb/dbswitch.conf ファイルには、ファイル認証データベースとその設定を定義するエントリが含まれます。その例を次に示します。

```
directory myfiledb file
myfiledb:syntax keyfile
myfiledb:keyfile /path/to/config/keyfile
```

次の表を参照してください。

表 9-3　ファイル認証データベースがサポートするパラメータ

syntax	[省略可能] 値は keyfile、digest、または htaccess。省略した場合のデフォルト値は keyfile
keyfile	[syntax=keyfile の場合は必須] ユーザーデータを含むファイルへのパス
digestfile	[syntax=digest の場合は必須] ダイジェスト認証用のユーザーデータを含むファイルへのパス
groupfile	[syntax=htaccess の場合は必須] AuthGroupFile へのパス
userfile	[syntax=htaccess の場合は必須] AuthUserFile へのパス

警告　ファイル認証データベースファイル (htaccess、digestfile、または keyfile) で許可されている一行の長さの上限は 255 バイトです。

この制限を超える行がある場合、サーバーは起動に失敗し、ログファイルにエラーが記録されます。

注　ACL がファイルベース認証データベースを使用するように設定する前に、次の前提条件が満たされることを確認してください。

- ファイルベースの認証ディレクトリサービスがすでに設定されている。方法については、55 ページの「ディレクトリサービスの設定」を参照
- ACL が設定される仮想サーバーが、必要となるファイルベース認証データベース (keyfile、htaccess、または digestauth) のファイルタイプを使用する。このように設定しない場合、デフォルトとして設定されているディレクトリサービスに対して ACL 制限が設定されます。

ファイル認証に基づくディレクトリサービス用の ACL の作成

ファイル認証に基づいてディレクトリサービス用の ACL エントリを作成するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、ACL を作成または編集するサーバーインスタンスを選択します。
2. サーバーマネージャの「Preferences」タブを選択します。

3. 「Restrict Access」リンクをクリックします。
4. 「Option」列の下で、ドロップダウンリストから ACL ファイルを選択し、「Edit ACL」をクリックします。
5. 「Access Control Rules」ページの上部フレームで、編集する ACL の「Users/Groups」リンクをクリックします。
6. 「User/Group」ページの下部フレームで、「Authentication database」ドロップダウンリストから「keyfile」を選択します。
7. 「Update」をクリックします。

   keyfile ベースのファイル認証データベースに対して ACL を設定すると、次のエントリ例のように、`dbswitch.conf` ファイルが ACL エントリで更新されます。

```
version 3.0;
  acl "default";
  authenticate (user) {
  prompt = "Sun One Web Server 6.1";
  database = "mykeyfile";
  method = "basic";
  };
deny (all) user = "anyone";
allow (all) user = "all";
```

.htaccess 認証に基づくディレクトリサービス用の ACL の作成

Sun ONE Web Server は、.htaccess ベースのフラットファイル認証をサポートしています。これまで .htaccess 認証を使用していた場合は、既存のデータファイルを変更せずに、ファイル認証データベースに移行できます。「.htaccess ファイルの使用」でも説明したように、.htaccess のユーザーとグループのデータは、1 つのファイルに格納することも、2 つのファイル (ユーザーデータ用とグループデータ用) に分割することもできます。ファイル認証データベースでは、両方の形式がサポートされます。

htaccess 認証に基づいてディレクトリサービス用の ACL エントリを作成するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、ACL を作成または編集するサーバーインスタンスを選択します。
2. サーバーマネージャの「Preferences」タブを選択します。

3. 「Restrict Access」リンクをクリックします。
4. 「Option」列の下で、ドロップダウンリストから ACL ファイルを選択し、「Edit ACL」をクリックします。
5. 「Access Control Rules」ページの上部フレームで、編集する ACL の「Users/Groups」リンクをクリックします。
6. 「User/Group」ページの下部フレームで、「Authentication database」ドロップダウンリストから「htaccess」を選択します。
7. 「Update」をクリックします。

   htaccess ベースのファイル認証データベースに対して ACL を設定すると、次のエントリ例のように、`dbswitch.conf` ファイルが ACL エントリで更新されます。

```
version 3.0;
acl "default";
  authenticate (user) {
  prompt = "Sun One Web Server 6.1";
  database = "myhtaccessfile";
  method = "basic";
  };
deny (all) user = "anyone";
allow (all) user = "all";
```

ファイル認証データベースへの既存の .htaccess 情報の移行

既存の .htaccess 情報を Sun ONE Web Server 6.1 のファイル認証データベースに移行するには、次の手順を実行します。

- .htaccess ユーザーファイルデータベースを *server-root*`/`*server-instance*`/config/userfile` にコピーします。
- .htaccess グループファイルデータベースを *server-root*`/`*server-instance*`/config/groupfile` にコピーします。

ユーザーファイルの形式は次のとおりです。

```
#user:password
```

グループファイルの形式は次のとおりです。

```
#group1:user1 user2
#group2:user3 user4
```

注　　メンバー名は、空白文字で区切ります。

ユーザーファイルとグループファイルの名前が同じ場合は、次に示す行の構文で両方のファイルを 1 つにまとめることができます。

```
#user:password:group1,group2
```

注　　各列は、コロンで区切ります。

htaccess データベースの例

例 1

```
#sample userfile  (user/password "j2ee/j2eepwd"  user/password
"user1/user1pwd" )

j2ee:9hmjfRwNxvJLU

user1:wvQirF86BsjSk
```

例 2

```
#sample group file

staff:j2ee  user1

eng:j2ee
```

例 3

```
#sample user/group file (username "j2ee", user password "j2eepwd")

j2ee:9hmjfRwNxvJLU:staff,eng
```

ダイジェスト認証に基づくディレクトリサービス用の ACL の作成

ファイル認証データベースは、RFC 2617 に規定されているダイジェスト認証での使用に適したファイル形式もサポートしています。ハッシュはパスワードに基づき、レルムは格納されます。プレーンテキストパスワードは維持されません。

digestauth ベースの認証に基づいてディレクトリサービス用の ACL エントリを作成するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、ACL を作成または編集するサーバーインスタンスを選択します。
2. サーバーマネージャの「Preferences」タブを選択します。
3. 「Restrict Access」リンクをクリックします。

4. 「Option」列の下で、ドロップダウンリストから ACL ファイルを選択し、「Edit ACL」をクリックします。
5. 「Access Control Rules」ページの上部フレームで、編集する ACL の「Users/Groups」リンクをクリックします。
6. 「User/Group」ページの下部フレームで、「Authentication database」ドロップダウンリストから「digest」を選択します。
7. 「Update」をクリックします。

   digestauth ベースのファイル認証データベースに対して ACL を設定すると、次のエントリ例のように、`dbswitch.conf` ファイルが ACL エントリで更新されます。

```
version 3.0;

  acl "default";

  authenticate (user) {

  prompt = "filerealm";

  database = "mydigestfile";

  method = "digest";

  };

deny (all) user = "anyone";

allow (all) user = "all";
```

第 10 章

ログファイルの使用

複数の方法で、サーバーのアクティビティを監視することができます。この章では、ログファイルを記録して参照することによって、サーバーを監視する方法について説明します。組み込み型のパフォーマンス監視サービス、サービス品質機能、SNMP の使用の詳細は、「サーバーの監視」を参照してください。

この章では、次の項目について説明します。

- ログファイルについて
- UNIX および Windows プラットフォームへのログオン
- ログレベル
- 仮想サーバーとログの記録について
- アプリケーションとサーバーのログ出力のリダイレクト
- ログファイルの保管
- アクセスログの詳細設定
- エラーロギングオプションの設定
- LOG 要素の設定
- アクセスログファイルの参照
- エラーログファイルの参照
- ログアナライザの実行
- イベントの表示 (Windows)

ログファイルについて

サーバーのログファイルには、サーバーのアクティビティが記録されます。このようなログを使用してサーバーを監視すると、障害追跡時に役立ちます。サーバーのルートディレクトリの `https-`*server_name*`/logs/errors` に保存されるエラーログファイルには、サーバーで検出されたすべてのエラーのリストがあります。サーバーのルートディレクトリの `https-`*server_name*`/logs/access` に保存されるアクセスログには、サーバーに対する要求とサーバーの応答に関する情報が記録されます。Sun ONE Web Server の `access` ログファイルに記録される情報を指定することができます。サーバーの統計情報を生成するには、ログアナライザを使用します。サーバーのエラーログファイルとアクセスログファイルをアーカイブし、バックアップをとっておくことができます。

注	オペレーションシステムでの制限によって、Linux 上で稼動している Sun ONE Web Server では 2G バイトを超えるログファイルを処理できません。最大サイズに達すると、ロギングが終了します。

UNIX および Windows プラットフォームへのログオン

この節では、ログファイルがどのように作成されるのかについて説明します。さらに、この節では、次の項目についても説明します。

- デフォルトのエラーログ
- syslog を利用したログ
- Windows の eventlog を利用したログ

デフォルトのエラーログ

UNIX と Windows のどちらのプラットフォームでも、管理サーバーからのログは、管理サーバーの `https-admserve/logs/` ディレクトリに収集されます。サーバーインスタンスからのログは、`https-`*server_name*`/logs/` ディレクトリで収集されます。

サーバー全体のデフォルトのログレベルを設定することができます。stdout と stderr をサーバーのイベントログにリダイレクトし、ログをオペレーティングシステムのシステムログに出力できます。さらに、stdout と stderr の内容をサーバーのイベントログに出力することもできます。デフォルトでは、ログメッセージは stderr と、指定のサーバーログファイルに送信されます。

また、ログメッセージと共に仮想サーバー ID を記録する機能も用意されています。これは、複数の仮想サーバーがメッセージの記録に同じログファイルを使用している場合に便利です。ログメッセージをシステムログに書き込むこともできます。この場合は、ログの記録はエラーログファイルでは行われません。その代わりに、ログの生成と管理は UNIX のシステムロギングサービス、または Windows プラットフォームのシステムロギングサービスによって行われます。

`server.xml` 属性を使用して、このファイルの内容を制御することもできます。`server.xml` ファイルの詳細については、『Sun ONE Web Server 6.1 Administrator's Configuration File Reference』を参照してください。

syslog を利用したログ

一元的なログ記録が必要となる安定した操作環境では、syslog が適しています。診断とデバッグのためにログ出力が頻繁に必要となる環境では、各サーバーインスタンスまたは仮想サーバーのログのほうが管理が容易です。

注

- サーバーインスタンスおよび管理サーバーのすべてのログデータが 1 つのファイルに記録される場合、内容を解釈したり、デバッグに利用することが難しくなります。syslog マスターログファイルは、円滑に稼動している配備済みアプリケーションだけで使用することをお勧めします。
- ログに記録されるメッセージには、Solaris デーモンアプリケーションからのその他のログも含まれています。

syslog ログファイルを syslogd およびシステムログデーモンと組み合わせて使用することで、`syslog.conf` ファイルを設定して次の処理を行うことができます。

- 適切なシステムログにメッセージを記録する
- システムコンソールにメッセージを出力する
- ログに記録されたメッセージをユーザーリストに転送する、または、ログに記録されたメッセージをネットワーク経由で別のホストの syslogd に転送する

syslog へのログ記録では、Sun ONE Web Server およびその他のデーモンアプリケーションからのログが同じファイルに収集されるため、Sun ONE Web Server に固有のメッセージと、特定のサーバーまたは仮想サーバーインスタンスのメッセージを区別するために、ログメッセージに次の情報が追加されます。

- 一意のメッセージ ID
- タイムスタンプ

- インスタンス名
- プログラム名 (webservd または webserv-wdog)
- プロセス ID (webserv プロセスの PID)
- スレッド ID (オプション)
- サーバー ID

`server.xml` ファイルでは、管理サーバーとサーバーインスタンスの両方に合わせて LOG 要素を設定することができます。

UNIX オペレーティング環境で使用される syslog のロギングメカニズムの詳細を参照するには、端末のプロンプトで次の man コマンドを使用します。

```
man syslog
man syslogd
man syslog.conf
```

Windows の eventlog を利用したログ

Windows オペレーティング環境で使用されるイベントログメカニズムの詳細を参照するには、Windows のヘルプシステムの索引から「イベントログ」を参照してください。

ログレベル

次の表は、ログレベルと Sun ONE Web Server のメッセージを重要度の順に示しています。

表 10-1　ログレベル

ログレベル	説明
finest	詳細なデバッグメッセージ。finest が最も詳細な情報を示す
finer	
fine	
info	通常はサーバーの設定または状態に関するものなどの単なる情報を示す。すぐに対応が必要なエラーメッセージではない
warning	警告を示す。このメッセージには、例外も含まれる
failure	アプリケーションの通常の実行の妨げとなる、深刻な障害を示す
config	特定の設定に関連する問題のデバッグに役立つ設定の各種統計情報を示す
security	セキュリティに関する問題を示す
catastrophe	致命的なエラーを示す

仮想サーバーとログの記録について

Sun ONE Web Server では、仮想サーバーインスタンスを利用できます。Sun ONE Web Server インスタンス内の各仮想サーバーは独自の ID を持ち、それぞれに専用のログファイルを持つこともあります。各仮想サーバーで独立したログファイルを使用することで、特定のトランザクションやリソースに関するサーバーのアクティビティを容易に追跡できます。

また、ログに記録されたメッセージを、複数の仮想サーバーから 1 つのサーバーログファイルに出力することもできます。この場合、server.xml ファイルの LOG 要素に含まれる logvsid を有効にすることができます。これにより、ログメッセージがどの仮想サーバーから出力されたものであるかを識別できます。

```
<SERVER>
  ...
```

```
  <LOG file="/export//https-iws-files2.red.iplanet.com/logs/errors"
loglevel="finest" logtoconsole="true" usesyslog="false"
createconsole="false" logstderr="true" logstdout="true"
logvsid="true"/>

</SERVER>
```

この例では、<LOG logvsid="true"> という設定によって、すべてのログメッセージに仮想サーバー ID が記録されます。この記録された ID で、メッセージを出力した仮想サーバーを識別できます。VS 要素に errorlog 属性を指定しない場合、すべての仮想サーバーのメッセージが 1 つのファイルに記録されます。

アプリケーションとサーバーのログ出力のリダイレクト

開発者にとっては、Web アプリケーションコンポーネントと J2EE アプリケーションの単体テストの段階では、アプリケーションログとサーバーログをすぐに参照できることが重要です。Windows プラットフォームでは、デスクトップのコマンドウィンドウでログメッセージを確認する方法を開発者は主に使用します。UNIX プラットフォームでは、サーバーインスタンスを開始した端末ウィンドウで stderr にログメッセージを出力するか、ログファイルに書き込まれたメッセージを tail -f コマンドを使用して確認する方法を多くの開発者が使用しています。

server.xml ファイルには、ログに記録されたメッセージをログファイルまたは端末ウィンドウに出力するように stdout および stderr を設定するための属性が含まれています。stdout と stderr の使用方法の詳細については、『Sun ONE Web Server 6.1 Administrator's Configuration File Reference』を参照してください。

ログファイルの保管

アクセスログファイルとエラーログファイルが自動的にアーカイブされるように設定することができます。指定の時刻、または指定の間隔で、ログがローテーションされます。Sun ONE Web Server は、古いログファイルを保存し、そのファイルに保存日時を含む名前を付けます。

たとえば、ファイルを毎時間ローテーションするように設定すると、Sun ONE Web Server は「access.200307152400」という名前を付けてファイルを保存します。この名前は、ログファイル名、年、月、日、24 時間形式の時刻が 1 つの文字列で表わされます。ログアーカイブファイルの形式は、設定したログローテーションのタイプによって異なります。

Sun ONE Web Server では、内部デーモンログローテーションと Cron ベースのログローテーションの 2 つのタイプのアーカイブファイルのログローテーションを使用できます。

内部デーモンログローテーション

このタイプのログローテーションは HTTP デーモン内で行われ、起動時にだけ設定を変更できます。内部デーモンログローテーションでは、サーバーの再起動を必要とせず、サーバーで内部的にログをローテーションできます。この方法でローテーションされるログは、次の形式で保存されます。

```
access.<YYYY><MM><DD><HHMM>
```

```
error. <YYYY><MM><DD><HHMM>
```

ログファイルをローテーションし、新しいログファイルでの記録を開始する間隔として使用される時間を指定できます。たとえば、ローテーションの開始時刻が午前 0 時であれば、ローテーション間隔は 1440 分 (1 日) となり、変更を保存して適用すると、現在の時刻に関係なく新しいログファイルが直ちに作成されます。ログファイルは毎日午前 0 時にローテーションされ、アクセスログのタイムスタンプは午前 0 時になり、access.200307152400 という名前で保存されます。同様に、間隔を 240 分 (4 時間) に設定した場合、午前 0 時から 4 時間おきにログがローテーションされます。アクセスログファイルには、午前 0 時から午前 4 時まで、午前 4 時から午前 8 時まで、それ以降同様に 4 時間で収集された情報が保存されます。

ログローテーションが有効になっている場合、サーバーの起動時にログファイルのローテーションが開始されます。ローテーションされる最初のログファイルでは、現在時刻から次のローテーションまでの間の情報が収集されます。前の例を使用して、開始時刻を午前 0 時に設定し、ローテーションの間隔を 240 分に設定した場合、現在時刻が午前 6 時とすると、ローテーションされる最初のログファイルには午前 6 時から午前 8 時の間に収集された情報が保存され、次のログファイルには午前 8 時から午後 12 時 (正午) までの間に収集された情報が保存されます。

スケジューラベースのログローテーション

このタイプのログローテーションは、*server_root*/https-admserv/config/ ディレクトリの scheduler.conf ファイルに記録される時間を基準にします。この方法では、すぐにログファイルをアーカイブすることも、サーバーで特定の日の特定の時間にログファイルをアーカイブするように設定することもできます。サーバーのスケジューラ設定オプションは、*server_root*/https-admserv/config/ ディレクトリの schedulerd.conf に保存されます。スケジューラベースの方法でローテーションされるログは、次の形式で保存されます。

<*original_filename*>.<YYYY><MM><DD><HHMM>

たとえば、午後 4 時 30 分にローテーションされる access は access.200307151630 という名前になります。

ログローテーションは、サーバーの起動時に初期化されます。ローテーションを有効にすると、Sun ONE Web Server はタイムスタンプの付いたアクセスログファイルを作成し、ローテーションがサーバーの起動時に開始されます。

ローテーションが開始されると、アクセスログファイルまたはエラーログファイルに記録する必要のある要求やエラーがあり、事前にスケジュールされている「次のローテーション時」の後にその要求やエラーが発生した場合、Sun ONE Web Server で新しいタイムスタンプが付いたログファイルが作成されます。

注	ログアナライザを実行する前に、サーバーログをアーカイブする必要があります。

ログファイルをアーカイブし、内部デーモンの方法とスケジュールベースの方法のどちらを使用するかを指定するには、サーバーマネージャで「Archive Log Files」ページを使用します。

アクセスログの詳細設定

インストール中、サーバーに access という名前のアクセスログファイルが作成されます。アクセスをログに記録するか否か、ログの記録に使用する形式、クライアントがリソースにアクセスした場合にサーバーでそのクライアントのドメイン名を検索する必要があるか否かを指定することによって、リソースへのアクセスログをカスタマイズできます。

ログファイルの書式文字列に %vsid% を追加するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Logs」タブを選択します。
2. 「Access Log Preferences」リンクをクリックします。
3. 「Log File:」テキストボックスに、新しいログファイルの保存場所とファイル名を入力します。
4. 「Only Log:」ラジオボタンをクリックします。
5. 「Virtual Server Id」チェックボックスにチェックマークを付けます。または、「Custom Format:」ラジオボタンをクリックし、文字列「%vsid%」を追加します。

注 カスタム形式の文字列「%vsid%」を追加する場合、新しいアクセスログファイルを使用する必要があります。

既存のログファイルの形式を変更する場合は、最初に既存のログファイルを削除するか、名前を変更します。あるいは、別のファイル名を使用します。

サーバーアクセスログは、共通ログファイル形式、フレキシブルログ形式、または独自のカスタマイズ可能な形式にすることができます。共通ログファイル形式は一般的にサポートされている形式で、サーバーに関する一定量の情報が提供されます。フレキシブルログ形式では、(Sun ONE Web Server から) ログに記録する内容を選択できます。カスタマイズ可能な形式では、パラメータブロックを指定してログの内容を制御します。カスタマイズ可能な形式のパラメータのリストについては、『NSAPI Programmer's Guide』を参照してください。

リソースのアクセスログが作成されると、そのログをアーカイブする場合や、同じリソースに対して新しいアクセスログファイルを作成する場合を除いて、アクセスログの形式を変更することはできません。

ログの詳細設定を指定するには、サーバーマネージャの「Access Log Preferences」ページを使用するか、または obj.conf ファイルで次の指令を手動で変更します。magnus.conf では、サーバーは関数 flex-init を呼び出してフレキシブルロギングシステムを初期化し、obj.conf で関数 flex-log を呼び出して要求された特定の

データをフレキシブルログ形式で記録します。共通ログファイル形式を使用して要求をログに記録するには、サーバーは `init-clf` を呼び出して `obj.conf` で使用される共通ログのサブシステムを初期化し、`common-log` を呼び出して要求された特定のデータを (ほとんどの HTTP サーバーで使用される) 共通ログ形式で記録します。

NSAPI ロギング関数と有効な指令とパラメータの詳細は、『NSAPI Programmer's Guide』を参照してください。

Cookie を使用した簡易ロギング

Sun ONE Web Server には、flexlog 機能を使用して簡単に特定の cookie のログをとる方法もあります。`obj.conf` 設定ファイル内の `flex-log` サブシステムを初期化する行に「`Req->headers.cookie.`*cookie_name*」を追加します。これによって、要求のヘッダーに cookie 変数がある場合は cookie 変数 *cookie_name* の値がログに記録され、ない場合は「-」が記録されます。

エラーロギングオプションの設定

Sun ONE Web Server 6.1 では、サーバーのエラーログに記録される情報を設定することができます。

管理サーバーインスタンスの設定

1. 管理サーバーへアクセスします。
2. 「Preferences」タブを選択します。
3. 「Access Logging Options」リンクをクリックします。
4. 必要な情報を入力します。
5. 「OK」をクリックし、「Apply」をクリックして変更を適用します。

サーバーインスタンスの設定

1. サーバーインスタンスへアクセスします。
2. 「Logs」タブを選択します。
3. 「Error Log Preferences」リンクをクリックします。
4. 必要な情報を入力します。
5. 「OK」をクリックし、「Apply」をクリックして変更を適用します。

LOG 要素の設定

次の表は、server.xml ファイル内で設定できる LOG 要素の属性を示しています。

表 2　LOG 属性

属性	デフォルト	説明
file	errors	デフォルト仮想サーバーからのメッセージを格納するファイルを指定します。VS 要素に明示的に errorlog 属性が指定されていない限り、設定されているその他の仮想サーバーからのメッセージもここに格納されます。
loglevel	info	その他の要素がエラーログに記録するメッセージのデフォルトタイプを制御します。指定できる値は、詳細度の高いものから低いものへ、順に次のとおりです。 finest、fine、fine、info、warning、failure、config, security、および catastrophe。
logvsid	false	(省略可能) true に設定した場合、仮想サーバーログに仮想サーバー ID が表示されます。これは、複数の VS 要素が同じログファイルを共有する場合に便利です。 Sun ONE Web Server 6.1 では、magnus.conf ファイルに logvsid 要素を設定できないことに注意してください。
logstdout	true	(省略可能) true に設定した場合、stdout の出力がエラーログにリダイレクトされます。有効な値は、on、off、yes、no、1、0、true、false です。
logstderr	true	(省略可能) true に設定した場合、stderr の出力がエラーログにリダイレクトされます。有効な値は、on、off、yes、no、1、0、true、false です。
logtoconsole	true	(省略可能、UNIX のみ) true に設定した場合、ログメッセージがコンソールにリダイレクトされます。
createconsole	false	(省略可能、Windows のみ) true に設定した場合、stderr 出力用の Windows コンソールが作成されます。有効な値は、on、off、yes、no、1、0、true、false です。

表 2　LOG 属性 (続き)

属性	デフォルト	説明
usesyslog	false	(省略可能) true に設定した場合、ログの生成と管理に UNIX の syslog サービス、または Windows のイベントログが使用されます。有効な値は、on、off、yes、no、1、0、true、false です。

アクセスログファイルの参照

サーバーで使用中のアクセスログファイル、およびアーカイブされたアクセスログファイルを参照できます。

管理サーバーのアクセスログを管理サーバーから参照するには、「Preferences」タブを選択し、「View Access Log」ページを選択します。

サーバーマネージャから サーバーインスタンスのアクセスログを参照するには、「Logs」タブを選択し、「View Access Log」ページを選択します。

クラスマネージャから個々の仮想サーバーのアクセスログを参照するには、強調表示されている「Manage Virtual Servers」ページから管理する仮想サーバーを選択し、仮想サーバーマネージャのページで「Access Log」という見出しの横のリンクをクリックします。参照するエントリ数、または参照したい条件修飾子付きのエントリを指定できます。

次の内容は、共通ログファイル形式のアクセスログの例です。ログファイルの形式は、「Log Preferences」ウィンドウで指定します。詳細は、245 ページの「アクセスログの詳細設定」を参照してください。

```
wiley.a.com - - [16/Feb/2001:21:18:26 -0800] "GET / HTTP/1.0" 200 751
wiley.a.com - - [17/Feb/2001:1:04:38 -0800] "GET /docs/grafx/icon.gif
HTTP/1.0" 204 342
wiley.a.com - - [20/Feb/2001:4:36:53 -0800] "GET /help HTTP/1.0" 401 571
arrow.a.com - john [29/Mar/2001:4:36:53 -0800] "GET /help HTTP/1.0" 401 571
```

表 10-3 では、このサンプルアクセスログの最後の行について説明します。

表 10-3　サンプルアクセスログファイルの最後の行のフィールド

アクセスログフィールド	例
Hostname or IP address of client	arrow.a.com。(この場合、Web サーバーの DNS 検索の設定が有効になっているため、ホスト名が表示される。DNS 検索が無効になっている場合は、クライアントの IP アドレスが表示される
RFC 931 information	- (RFC 931 の識別情報は表示されない)
Username	john (認証のためにクライアントによって入力されたユーザー名)
Date/time of request	29/Mar/1999:4:36:53 -0800
Request	GET /help
Protocol	HTTP/1.0
Status code	401
Bytes transferred	571

次の内容は、フレキシブルロギング形式を使用したアクセスログの例です。ログファイルの形式は、「Log References」ページで指定します。詳細は、245 ページの「アクセスログの詳細設定」を参照してください。

```
wiley.a.com - - [25/Mar/2001:12:55:26 -0800] "GET /index.htm HTTP/1.0" "GET"
"/?-" "HTTP/   1.0" 304 0 - Mozilla/2.0 (WinNT; I)
wiley.a.com - - [25/Mar/2001:12:55:26 -0800] "GET / HTTP/1.0" "GET" "/?-"
"HTTP/1.0" 304 0    - Mozilla/2.0 (WinNT; I)
wiley.a.com - - [25/Mar/2001:12:55:26 -0800] "GET / HTTP/1.0" "GET" "/?-"
"HTTP/1.0" 304 0    - Mozilla/2.0 (X11; I; IRIX 5.3 IP22)
```

エラーログファイルの参照

エラーログファイルには、ログファイルが作成されてからサーバーで検出されたエラーが記録されます。また、このログファイルにはサーバーの起動時などのサーバーに関する情報メッセージも記録されます。エラーログには、失敗したユーザー認証も記録されます。エラーログを使用して、誤った URL パスや不足しているファイルを見つけることもできます。

管理サーバーから管理サーバーのエラーログファイルを参照するには、「Preferences」タブを選択し、「View Error Log」ページを選択します。

サーバーマネージャからサーバーインスタンスのエラーログファイルを参照するには、「Logs」タブを選択し、「View Error Log」ページを選択します。

クラスマネージャから個々の仮想サーバーのエラーログを参照するには、強調表示されている「Manage Virtual Servers」ページから管理する仮想サーバーを選択し、仮想サーバーマネージャのページで「Error Log」という見出しの横のリンクをクリックします。参照するエントリ数、または参照したい条件修飾子付きのエントリを指定できます。

次の内容には、エラーログ内のエントリの 2 つの例が含まれています。最初の例は、サーバーの起動に成功したことを示す情報メッセージです。2 番目の例は、クライアントの `wiley.a.com` が `report.html` ファイルを要求したが、ファイルがサーバーのプライマリドキュメントディレクトリに存在しなかったことを示します。

```
[[22/Jan/2001:14:31:41] info (39700): successful server startup
[22/Jan/2001:14:31:41] info (39700): SunONE-WebServer/6.1 BB1-01/22/2001 01:45
[22/Jan/2001:14:31:42] warning (13751): for host wiley.a.com trying to GET
/report.html, send-file reports: can't find
/usr1/irenem/ES60-0424/docs/report.html (File not found)
```

ログアナライザの実行

server-root/extras/log_anlyディレクトリには、サーバーマネージャのユーザーインタフェースから実行するログ分析ツールがあります。このログアナライザは、共通ログ形式のファイルだけを分析します。log_anly ディレクトリにある HTML ドキュメントに、このツールのパラメータが説明されています。
server-root/extras/flex_anlgディレクトリには、フレキシブルログファイル形式用のコマンド行ログアナライザがあります。ただし、サーバーマネージャのデフォルト設定では、共通ログファイル形式とフレキシブルログファイル形式のどちらを選択したかに関係なく、フレキシブルログファイルレポートツールを使用するように設定されています。

ログアナライザを使用して、アクティビティの要約、もっとも頻繁にアクセスされるURL、サーバーがもっとも頻繁にアクセスされる時間など、デフォルトサーバーの統計情報を生成します。ログアナライザは Sun ONE Web Server から実行することも、コマンド行から実行することもできます。ログアナライザでは、デフォルトサーバー以外の仮想サーバーの統計情報を生成することはできません。ただし、248 ページの「アクセスログファイルの参照」で説明されているように、各仮想サーバーの統計情報を参照することはできます。

flexanlg コマンド行ユーティリティを実行する前に、ライブラリパスを設定する必要があります。各種プラットフォームでの設定は、次のとおりです。

Solaris および Linux:

```
LD_LIBRARY_PATH=server_root/bin/https/lib:$LD_LIBRARY_PATH
```

AIX:

```
LIBPATH=server_root/bin/https/lib:$LIBPATH
```

HP-UX:

```
SHLIB_PATH=server_root/bin/https/lib:$SHLIB_PATH
```

Windows:

```
path=server_root¥bin¥https¥bin;%path%
```

注 ログアナライザを実行する前に、サーバーログをアーカイブする必要があります。サーバーログのアーカイブの詳細は、243 ページの「ログファイルの保管」を参照してください。

サーバーマネージャからログアナライザを実行するには、次の手順を実行します。

1. サーバーマネージャの「Logs」タブを選択します。
2. 「Genarate Report」をクリックします。

3. 各フィールドに必要事項を入力します。
4. 「OK」をクリックします。

   新しいウィンドウにレポートが表示されます。

詳細は、オンラインヘルプの「Generate Report ページ」を参照してください。

コマンド行からアクセスログファイルを分析するには、`flexanlg` ツールを実行します。このツールは *server-install*`/extras/flex_anlg` ディレクトリにあります。

`flexanlg` を実行するには、コマンドプロンプトに以下のコマンドとオプションを入力します。

```
flexanlg [ -P ] [-n name] [-x] [-r] [-p order] [-i file]* [ -m
metafile ]* [ o file][ c opts] [-t opts] [-l opts]
```

次に構文を説明します。

```
flexanlg -h.):
-P: proxy log format                                    Default:no
-n servername: The name of the server
-x : Output in HTML                                     Default:no
-r : Resolve IP addresses to hostnames                  Default:no
-p [c,t,l]: Output order (counts, time stats, lists)  Default: ctl
-i filename: Input log file(s)                          Default:none
-o filename: Output log file                            Default:stdout
-m filename: Meta file(s)                               Default:none
-c [h,n,r,f,e,u,o,k,c,z]: Count these item(s) -       Default: hnreuokc
    h: total hits
    n: 304 Not Modified status codes (Use Local Copy)
    r: 302 Found status codes (Redirects)
    f: 404 Not Found status codes (Document Not Found)
    e: 500 Server Error status codes (Misconfiguration)
    u: total unique URL's
    o: total unique hosts
    k: total kilobytes transferred
    c: total kilobytes saved by caches
    z: Do not count any items.
-t [sx,mx,hx, xx,z]: Find general stats - Default:s5m5h24x10
    s(number): Find top (number) seconds of log
    m(number): Find top (number) minutes of log
    h(number): Find top (number) hours of log
    u(number): Find top (number) users of log
    a(number): Find top (number) user agents of log
    r(number): Find top (number) referers of log
    x(number): Find top (number) for miscellaneous keywords
    z: Do not find any general stats.
-l [cx,hx]: Make a list of - Default: c+3h5
    c(x,+x): Most commonly accessed URLs
             (x: Only list x entries)
             (+x: Only list if accessed more than x times)
    h(x,+x): Hosts (or IP addresses) most often accessing your server
             (x: Only list x entries)
             (+x: Only list if accessed more than x times)
    z: Do not make any lists
```

イベントの表示 (Windows)

Sun ONE Web Server は、サーバーエラーログにエラーを記録します (250 ページの「エラーログファイルの参照」を参照) が、深刻なシステムエラーのログはイベントビューアにも記録します。イベントビューアでは、システム上のイベントを監視できます。イベントビューアを使用して、基本設定での問題によって発生したエラーを参照します。この問題は、エラーログが開けるようになる前に発生する可能性があります。

イベントビューアを使用するには、次の手順を実行します。

1. 「スタート」メニューから、「プログラム」、「管理ツール」を順に選択します。「管理ツール」プログラムグループで「イベントビューア」を選択します。
2. 「ログ」メニューの「アプリケーション」を選択します。

   「イベントビューア」に「アプリケーションログ」が表示されます。Sun ONE Web Server でのエラーには、`https-`*`serverid`* または `WebServer6.1` というソースラベルが付いています。
3. 「表示」メニューの「検索」を選択すると、ログ内でこのようなラベルを検索できます。「表示」メニューの「最新の情報に更新」を選択すると、更新されたログエントリを表示できます。

   イベントビューアの詳細は、使用しているシステムのマニュアルを参照してください。

第 11 章

サーバーの監視

この章では、組み込み型の監視ツール、サービス品質の機能、Simple Network Management Protocol (SNMP) など、サーバーの監視手法について説明します。

Sun ONE の Management Information Base (MIB) や、HP OpenView のようなネットワーク管理ソフトウェアとともに SNMP を使用して、ネットワーク内の他のデバイスを監視するのと同じように、リアルタイムでサーバーを監視できます。

注 Windows では、Sun ONE Web Server 6.1 をインストールする前に、Windows SNMP コンポーネントがすでにインストールされていることを確認してください。

統計機能または SNMP を使用することによって、サーバーの状態をリアルタイムで表示できます。UNIX または Linux を使用している場合に、SNMP の使用を計画するときは、Sun ONE サーバーを SNMP 用に設定する必要があります。この章では、UNIX または Linux 上で、Sun ONE サーバーとともに SNMP を使用する際に必要な情報を提供します。

この章では、次の内容について説明します。

- 統計情報によるサーバーの監視
- サービス品質の使用法
- SNMP の基本
- Sun ONE Web Server の MIB
- SNMP の設定
- プロキシ SNMP エージェントの使用法 (UNIX または Linux)
- SNMP ネイティブエージェントの再設定
- SNMP マスターエージェントのインストール

- SNMP マスターエージェントを使用可能にして起動する
- SNMP マスターエージェントの設定
- サブエージェントを使用可能にする
- SNMP メッセージについて

統計情報によるサーバーの監視

統計機能を使用して、サーバーの現在の稼動状況を監視できます。統計情報は、サーバーが処理している要求数と、それらの要求の処理状況を示します。個々の仮想サーバーに関する統計情報や、サーバーインスタンス全体に関する統計情報を表示できます。対話型サーバーモニターを通してサーバーが多数の要求を処理していることがわかる場合、要求数に合わせてサーバー設定またはシステムのネットワークカーネルを調整する必要があることもあります。詳細は、`http://docs.sun.com/` で、オンラインマニュアル『Performance Tuning and Sizing Guide』を参照してください。

統計情報を使用可能にすると、次の分野の統計情報を表示できます。

- 接続
- DNS
- KeepAlive
- キャッシュ
- 仮想サーバー

対話型サーバーモニターで総計をレポートするサーバーの各種統計情報については、オンラインヘルプの「Monitor Current Activity」ページを参照してください。

警告 統計情報のプロファイリングを使用可能にすると、そのサーバーのすべてのユーザーが統計情報を使用できるようになります。詳細については、『NSAPI Programmer's Guide』に記載されている stats-xml の説明を参照してください。

統計情報を使用可能にする

統計情報を使用可能にするには、次の手順に従います。

1. サーバーマネージャから「Monitor」タブをクリックします。
2. 「Monitor Current Activity」をクリックします。
3. 「Yes」をクリックして統計情報を使用可能にします。
4. 「OK」をクリックします。
5. 「Apply」をクリックして変更を適用します。サーバーを再起動する必要はありません。

統計情報を使用可能にする方法の詳細については、オンラインヘルプを参照してください。

統計情報の使用法

統計情報を使用可能にすると、サーバーインスタンスや仮想サーバーの稼動状況に関するさまざまな情報を得ることができます。統計情報は、機能別に分類されます。

統計情報にアクセスするには、次の手順に従います。

1. サーバーマネージャから「Monitor」タブをクリックします。
2. 「Minitor Current Activity」をクリックします。
3. ドロップダウンリストからポーリング間隔を選択します。

   ポーリング間隔は、統計情報の表示について更新間隔を示す秒数です。
4. ドロップダウンリストから、表示する統計情報の種類を選択します。
5. 「Submit」をクリックします。

   サーバーインスタンスが稼動中で、統計情報のプロファイリングを使用可能にしている場合、選択した統計情報を示すページが表示されます。このページは、ポーリング間隔として選択した値に応じて、5 〜 15 秒ごとに更新されます。

統計情報に表示されるデータを使用してサーバーを調整できます。詳細は、`http://docs.sun.com/` で、オンラインマニュアル『Performance Tuning and Sizing Guide』を参照してください。

サービス品質の使用法

サービス品質は、サーバーインスタンスの仮想サーバークラスまたは仮想サーバーに対して設定するパフォーマンス制限です。たとえば、ISP の場合、許可する帯域幅に応じて、各仮想サーバーにそれぞれ異なる課金をしたい場合があります。その場合、2 つの領域を制限できます。1 つは帯域幅の量、もう 1 つは接続数です。

サーバーマネージャの「Monitor」タブで、サーバー全体または仮想サーバークラスに関するこれらの設定を有効にできます。ただし、個々の仮想サーバーについて、サーバー全体またはクラスレベルの設定を無効にできます。個々のサーバーに関するサービス品質の制限設定の詳細については、342 ページの「仮想サーバーのサービス品質の設定」を参照してください。

再計算間隔と測定時間の 2 つの設定値により、帯域幅の再計算頻度とトラフィックの計算方法を制御します。再計算間隔は、帯域幅を計算する頻度 (ミリ秒単位) です。測定時間は、トラフィック計算でデータを使用する時間です。

この節では、次の内容について説明します。

- サービス品質の例
- サービス品質の設定
- obj.conf で必要な変更
- サービス品質に関する既知の制限事項

サービス品質の例

次の例では、サービス品質の情報に関して、収集と計算の方法を示しています。

サーバーに設定されている測定時間は、30 秒です。

このサーバーは、0 秒で起動します。

起動から 1 秒が経過した時点で、HTTP 接続によって、サーバーとの間に 5000 バイトのトラフィックが生成されました。

以降、接続は発生しませんでした。30 秒経過の時点では、最後の 30 秒間の合計トラフィックは、5000 バイトです。

ただし、32 秒が経過すると、1 秒経過の時点でのトラフィックは破棄されます。これは、測定時間数に設定された 30 秒より、古いトラフィックであるからです。その結果、この時点での最後の 30 秒間の合計トラフィックは、0 になります。

再計算間隔も同様です。このサーバーの再計算間隔は 100 ミリ秒です。

前述の例を使用し、帯域幅は 100 ミリ秒ごとに再計算されます。この計算は、トラフィック量と測定時間に基づいて行われます。

0 秒の時点で、帯域幅の最初の計算が行われます。この時点での合計トラフィックは 0 で、測定時間の 30 秒で割ると、帯域幅は 0 になります。

1 秒の時点では、帯域幅の 10 回目 (1000 ミリ秒／ 100 ミリ秒) の計算が行われます。この時点での合計トラフィックは 5000 バイトで、これを 30 秒で割ります。したがって、帯域幅は 5000/30 = 166 バイト／秒になります。

30 秒の時点では、帯域幅の 300 回目の計算が行われます。この時点での合計トラフィックは 5000 バイトで、これを 30 秒で割ります。したがって、帯域幅は 5000/30 = 166 バイト／秒になります。

32 秒の時点では、帯域幅の 320 回目の計算が行われます。この時点でのトラフィックは 0 バイトになり (トラフィックを生成する 1 つの接続が古くなってカウントされなくなるため)、これを 30 秒で割ると、0 バイト／秒になります。

サービス品質の設定

サーバーインスタンスまたは仮想サーバークラスのサービスの品質を設定するには、ユーザーインタフェースを使用して設定する必要があります。サービス品質の設定を実際に有効にするためには、`obj.conf` ファイルの Server Application Function (SAF) も設定する必要があります。

サービス品質を設定するには、次の手順に従います。

1. サーバーマネージャから「Monitor」タブをクリックします。
2. 「Quality of Service」をクリックします。

   サービス品質の一般的な設定値を示すページが表示されます。続いて、サーバーインスタンス全体および仮想サーバーの各クラスのリストも表示されます。
3. サーバーインスタンス全体でサービス品質を使用可能にするには、「Enable」をクリックします。

   デフォルトでは、サービス品質は有効になっています。サービス品質を有効にすると、サーバーのオーバーヘッドがわずかに増えます。
4. 「Recompute Interval」を選択します。

   再計算間隔は、すべてのサーバー、クラス、および仮想サーバーに関する帯域幅の計算間隔を示すミリ秒数です。デフォルトは 100 ミリ秒です。

5. 「Metric Interval」を選択します。

   測定時間は、トラフィックを測定する間隔を示す秒数です。デフォルト値は 30 秒です。この時間に測定されたすべての帯域幅を平均して、秒当たりのバイト数が得られます。

   大規模なファイルの転送を多数扱うサイトの場合、このフィールドには、大きい値 (数分またはそれ以上) を使用します。大規模なファイルの転送では、測定時間が短いと、許容帯域幅のすべてが占有されることがあり、最大帯域幅の設定を有効にしている場合には接続が拒否されることがあります。帯域幅は測定時間で平均されるため、この間隔を長くすると、大規模ファイルによる帯域幅の急上昇が均等化されます。

   帯域幅制限値が使用可能帯域幅よりもはるかに小さい場合 (たとえば、帯域幅の制限値が 1M バイト／秒で、バックボーンとの接続が 1G バイト／秒の場合など) は、測定時間を短くする必要があります。

   大規模なスタティックファイルの転送を扱う場合で、帯域幅制限値が使用可能帯域幅よりもはるかに小さいときは、それぞれの問題が相反する解決法を必要とするので、どちらの状況を調整するかを決める必要があります。

6. サーバーインスタンスまたは仮想サーバークラス、あるいはその両方のサービス品質を使用可能にします。

   画面の下の部分に、サーバーインスタンスとサーバークラスが一覧表示されています。サービス品質を使用可能にする項目の隣にある「Enable」を選択します。

7. 最大帯域幅をバイト／秒単位で設定します。

8. 最大帯域幅の設定を適用するかどうかを選択します。

   最大帯域幅を適用する場合、サーバーがその帯域幅の制限値に達すると、それ以上の接続は拒否されます。

   最大帯域幅を適用しない場合は、最大帯域幅を超えると、サーバーのエラーログにメッセージが記録されます。

9. 最大接続許可数を選択します。

   この数は、同時に処理する要求の数です。

10. 最大接続数の設定を適用するかどうかを選択します。

    最大接続数を適用する場合、サーバーがその最大接続数に達すると、それ以上の接続は拒否されます。

11. 最大接続数を適用しない場合は、最大接続数を超えると、サーバーのエラーログにメッセージが記録されます。

12. 「OK」をクリックします。

obj.conf で必要な変更

サービス品質を使用可能にするには、AuthTrans `qos-handler` および Error `qos-error` という 2 つの Server Application Function (SAF) を呼び出す指令を `obj.conf` に追加する必要があります。

`qos-handler` AuthTrans 指令を正しく動作させるためには、この指令を、デフォルトのオブジェクトに最初の `AuthTrans` として設定する必要があります。サービス品質ハンドラの役割は、仮想サーバー、仮想サーバークラス、およびグローバルサーバーの現在の統計情報を調べ、エラーを返して制限値を適用することです。

Sun ONE Web Server には、`qos-handler` という組み込みのサービス品質ハンドラ SAF サンプルが含まれています。この SAF は、制限値に達した時にログを記録し、サーバーに 503 「Server busy」を返して、NSAPI で処理されるようにします。

Sun ONE Web Server には、`qos-error` という組み込みのエラー SAF サンプルも含まれており、これは、503 エラーの原因となった制限値およびその制限の原因となった統計値を示すエラーページを返します。サンプルコードを修正して、別のエラー情報を提供することもできます。

これらのサンプルは、*server_root*`/plugins/nsapi/examples/qos.c` に用意されています。これらのサンプルを使用することも、独自の SAF を記述することもできます。

これらの SAF およびその使用方法については、『NSAPI Programmer's Guide』を参照してください。

サービス品質に関する既知の制限事項

サービス品質の機能を使用する時は、以下の制限事項に留意してください。

- パフォーマンスを低下させないために、接続または帯域幅の統計情報は、サーバープロセス間で共有されません。つまり、`MaxProc` の設定は統計情報に反映されません。そのため、すべての制限値はサーバープロセスに個別に適用され、全プロセスの総計に対しては適用されません。`MaxProcs` および複数プロセスの詳細については、`http://docs.sun.com/` から、オンライン形式の『Performance Tuning and Sizing Guide』を参照してください。
- サービス品質の機能では、アプリケーションレベルの HTTP 帯域幅のみを測定します。HTTP 帯域幅は、次のようなさまざまな理由により、実際の TCP ネットワーク帯域幅とは異なる場合があります。
  - SSL が有効になっている場合、ハンドシェークおよびクライアント証明書交換がトラフィックに追加されますが、量は測定されません。
  - チャンクエンコーディングがどちらか一方の方向または両方向で有効になっている場合、チャンク層によってチャンクヘッダーが削除されて、トラフィックに算入されません。その他のヘッダーまたはプロトコル項目は算入されます。

- サービス品質の機能では、PR_TransmitFile コールからのトラフィックを正確に測定できません。PR_Send()/net_write や PR_Recv()/net_read などの基本 I/O オペレーションでは、1 回のシステムコールで転送されるバイト数は通常はバッファのサイズであり、I/O コールから即時に返されるため、転送されたデータは帯域幅マネージャによって即時に計算されます。このため、ダイナミックなコンテンツアプリケーションの瞬間的な帯域幅を正確に測定できます。ただし、PR_TransmitFile から転送されるデータの量は、転送の終了時点までわからないため、転送が完了するまでは測定できません。

  PR_TransmitFile が短時間の場合は、サービス品質機能は適切に動作します。ただし、ダイアルアップユーザーが大きいファイルをダウンロードする場合など、PR_TransmitFile が長時間に及ぶ場合は、転送の完了時に、転送された全体のデータ量が算入されます。次の再計算間隔が始まってから帯域幅マネージャが帯域幅を再計算すると、その大規模な PR_TransmitFile が原因で、帯域幅が非常に大きくなります。この場合、サーバーは、次の測定時間まですべての要求を拒否することがあります。そして、帯域幅マネージャがファイル転送オペレーションを「除外」したときには、ファイル転送オペレーションが終了し、帯域幅の値はふたたび小さくなります。かなり長時間に及ぶスタティックファイルのダウンロードを扱うサイトでは、測定時間をデフォルトの 30 秒よりも長くする必要があります。

- 計算される帯域幅は、瞬時に測定されるのではなく、一定の間隔で一定期間にわたって再計算されるので、常に近似値となります。たとえば、測定時間がデフォルトの 30 秒で、サーバーが 29 秒間アイドル状態の場合、次の 1 秒間で、クライアントがその帯域幅制限値の 30 倍を使用することもあります。

- サービス品質の帯域幅統計情報は、サーバーがダイナミックに再設定されると失われます。さらに、サービス品質の制限事項は、古い、アクティブでない設定で接続したスレッドには適用されません。これは、帯域幅マネージャのスレッドでは、アクティブな設定の帯域幅統計値のみを計算するためです。長時間ソケットを閉じずにアクティブになっているためにサーバーがタイムアウトにしないクライアントでは、サーバーのダイナミックな再設定後、サービス品質の制限事項の影響を受けない場合もあります。

- 同時に複数の接続が発生する場合には、仮想サーバーの統計情報は仮想サーバークラスおよびグローバルサーバーインスタンスとは異なる単位で計算されます。個々の仮想サーバーの接続カウンタは、要求が解析されて仮想サーバーに配信された直後に、ひとつひとつ増分されます。また、その要求に対する応答処理が終了した時点で、カウンタはひとつひとつ減らされます。このため、仮想サーバーの接続統計情報は、どの時点でも常に正確なものになります。

  ただし、仮想サーバークラスおよびグローバルサーバーインスタンスの接続統計情報は、瞬時には更新されません。これらの統計情報は、再計算間隔ごとに帯域幅マネージャのスレッドによって更新されます。仮想サーバークラスの接続数は、そのクラスのすべての仮想サーバー上の接続の合計数であり、グローバルサーバーインスタンスの接続数は、すべての仮想サーバークラス上の接続の合計数です。

それぞれの値の計算方法により、仮想サーバーの接続数は常に正確 (接続数の制限値を設定している場合は、その制限値を超えることができない) ですが、仮想サーバークラスおよびサーバーインスタンスの接続数は、一定の間隔ごとに計算されるので、十分に正確なものではありません。

SNMP の基本

SNMP は、ネットワークアクティビティに関するデータをやり取りするために使用されるプロトコルです。SNMP では、管理対象デバイスとネットワークマネージメントステーション (NMS) の間をデータが移動します。管理対象デバイスは、SNMP を使用するすべてのデバイス、つまり、ネットワーク上のホスト、ルーター、Web サーバー、その他のサーバーなどです。NMS は、そのネットワークをリモートで管理するためのマシンです。一般に、NMS ソフトウェアでは、収集されたデータをグラフに表示したり、そのデータを使用してサーバーが特定の許容範囲内で動作していることを確認します。

NMS は通常、1 つ以上のネットワーク管理アプリケーションがインストールされた強力なワークステーションです。HP OpenView のようなネットワーク管理アプリケーションでは、Web サーバーなどの管理対象デバイスに関する情報がグラフィカルに表示されます。たとえば、社内のどのサーバーが稼動またはダウンしているかを表示したり、受け取ったエラーメッセージの数と種類を表示したりできます。Sun ONE サーバーで SNMP を使用する場合、この情報は、サブエージェントとマスターエージェントという 2 種類のエージェントを使用して、NMS とサーバーの間で転送されます。

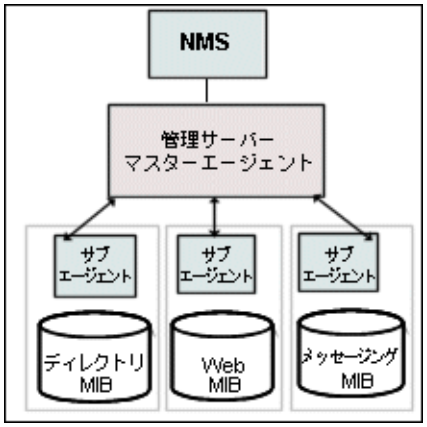
サブエージェントは、サーバーに関する情報を収集し、その情報をサーバーのマスターエージェントに渡します。管理サーバー以外のすべての Sun ONE サーバーには、サブエージェントがあります。

注 SNMP の設定を変更したあとは、「Apply」ボタンをクリックし、SNMP サブエージェントを再起動する必要があります。

マスターエージェントは、NMS と通信します。マスターエージェントは、管理サーバーとともにインストールされます。

1 つのホストコンピュータに複数のサブエージェントをインストールできますが、マスターエージェントは 1 つしかインストールできません。たとえば、Directory Server、Sun ONE Web Server、および Messaging Server を同じホストにインストールしている場合、次のように、各サーバーのサブエージェントは、同じマスターエージェントと通信します。

ネットワーク管理ステーションと SNMP エージェント

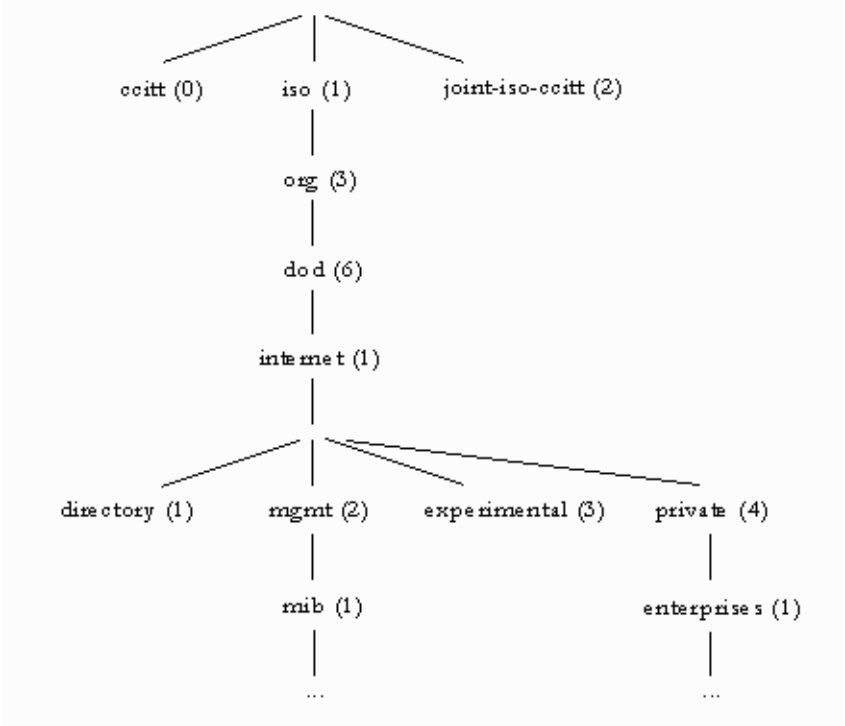


Sun ONE Web Server の MIB

Sun ONE Web Server には、ネットワーク管理に関する変数が保存されます。マスターエージェントがアクセスできる変数は、管理対象オブジェクトと呼ばれます。これらのオブジェクトは、MIB (Management Information Base) と呼ばれるツリー構造で定義されます。MIB により、Web サーバーのネットワーク設定、状態、および統計情報にアクセスすることができます。SNMP を使用すると、この情報をネットワークマネージメントステーション (NMS) から見ることができます。

サーバーの MIB には、そのサーバーのネットワーク管理に関する変数定義が含まれます。MIB ツリーのトップレベルは、次の図のとおりです。

MIB ツリーのトップレベル



MIB ツリーのトップレベルを見ると、インターネットオブジェクト識別子には directory (1)、mgmt (2)、experimental (3)、および private (4) という 4 つのサブツリーがあることがわかります。private (4) サブツリーには、enterprises (1) ノードが含まれています。enterprises (1) ノードの各サブツリーは、個別の企業に割り当てられます。この企業は、独自の MIB 拡張機能を登録している組織です。企業は、自社のサブツリーの下に製品別のサブツリーを作成できます。企業が作成した MIB は、enterprises (1) ノードの下に置かれます。Sun ONE の MIB は、enterprises (1) ノードの下に置かれます。

各 Sun ONE サーバーのサブエージェントには、SNMP 通信で使用する MIB が用意されています。Sun ONE サーバーは、これらの変数が含まれたメッセージまたはトラップを送信することによって、重大なイベントをネットワークマネージメントステーション (NMS) に報告します。NMS では、サーバーの MIB にデータを照会したり、MIB の変数をリモートで変更することもできます。

各 Sun ONE サーバーには、専用の MIB があります。Sun ONE の MIB はすべて、次の場所にあります。

server_root/plugins/snmp

Sun ONE Web Server の MIB は、webserv61.mib という名前のファイルです。この MIB には、Sun ONE Web Server のネットワーク管理に関する各種変数の定義が格納されています。

Sun ONE Web Server 6.1 の MIB は、http 60 (iws60 OBJECT IDENTIFIER ::= {http 60 }) というオブジェクト識別子を持ち、*server_root*/plugins/snmp ディレクトリにあります。

Sun ONE Web Server の MIB を使用すると、リアルタイムで Web サーバーに関する管理情報を確認し、そのサーバーを監視できます。表 11-1 に、webserv61.mib に格納されている管理対象オブジェクトとその説明を示します。

表 11-1 webserv61.mib の管理対象オブジェクトと説明

管理対象オブジェクト	説明
iwsInstanceTable	Sun ONE Web Server インスタンス
iwsInstanceEntry	Sun ONE Web Server インスタンス
iwsInstanceIndex	サーバーインスタンスのインデックス
iwsInstanceId	サーバーインスタンスの識別子
iwsInstanceVersion	バージョンを示す文字列。 例：SunONE-WebServer/6.1 BB1-01/24/2001 17:15 (SunOS DOMESTIC)
iwsInstanceDescription	サーバーインスタンスの説明
iwsInstanceOrganization	サーバーインスタンスを担当する組織
iwsInstanceContact	サーバーインスタンス担当者の連絡先情報
iwsInstanceLocation	サーバーがある場所
iwsInstanceStatus	サーバーインスタンスの状態
iwsInstanceUptime	サーバーが稼動している時間
iwsInstanceDeathCount	サーバーインスタンスのプロセスが機能停止した回数
iwsInstanceRequests	サーバーインスタンスが処理した要求数
iwsInstanceInOctets	サーバーインスタンスが受信したオクテット数。情報を利用できない場合は 0 を示す
iwsInstanceOutOctets	サーバーインスタンスが送信したオクテット数。情報を利用できない場合は 0 を示す
iwsInstanceCount2*xx*	サーバーインスタンスが発行した 200 番レベル (Successful) の応答数

表 11-1 webserv61.mib の管理対象オブジェクトと説明 (続き)

管理対象オブジェクト	説明
iwsInstanceCount3*xx*	サーバーインスタンスが発行した 300 番レベル (Redirection) の応答数
iwsInstanceCount4*xx*	サーバーインスタンスが発行した 400 番レベル (Client Error) の応答数
iwsInstanceCount5*xx*	サーバーインスタンスが発行した 500 番レベル (Server Error) の応答数
iwsInstanceCountOther	サーバーインスタンスが発行したその他の (2xx、3xx、4xx、5xx のいずれでもない) 応答数
iwsInstanceCount200	サーバーインスタンスが発行した 200 (Request Fulfilled) の応答数
iwsInstanceCount302	サーバーインスタンスが発行した 302 (Moved Temporarily) の応答数
iwsInstanceCount304	サーバーインスタンスが発行した 304 (Not Modified) の応答数
iwsInstanceCount400	サーバーインスタンスが発行した 400 (Bad Request) の応答数
iwsInstanceCount401	サーバーインスタンスが発行した 401 (Unauthorized) の応答数
iwsInstanceCount403	サーバーインスタンスが発行した 403 (Forbidden) の応答数
iwsInstanceCount404	サーバーインスタンスが発行した 404 (Not Found) の応答数
iwsInstanceCount503	発行された 503 (Unavailable) の応答数
iwsVsTable	Sun ONE Web Server 仮想サーバー
iwsVsEntry	Sun ONE Web Server 仮想サーバー
iwsVsIndex	仮想サーバーのインデックス
iwsVsId	仮想サーバーの識別子
iwsVsRequests	仮想サーバーが処理した要求数
iwsVsInOctets	仮想サーバーが受信したオクテット数
iwsVsOutOctets	仮想サーバーが送信したオクテット数
iwsVsCount2xx	仮想サーバーが発行した 200 番レベル (Successful) の応答数

表 11-1　webserv61.mib の管理対象オブジェクトと説明 (続き)

管理対象オブジェクト	説明
iwsVsCount3xx	仮想サーバーが発行した 300 番レベル (Redirection) の応答数
iwsVsCount4xx	仮想サーバーが発行した 400 番レベル (Client Error) の応答数
iwsVsCount5xx	仮想サーバーが発行した 500 番レベル (Server Error) の応答数
iwsVsCountOther	仮想サーバーが発行したその他の (2xx、3xx、4xx、5xx のいずれでもない) 応答数
iwsVsCount200	仮想サーバーが発行した 200 (Request Fulfilled) の応答数
iwsVsCount302	仮想サーバーが発行した 302 (Moved Temporarily) の応答数
iwsVsCount304	仮想サーバーが発行した 304 (Not Modified) の応答数
iwsVsCount400	仮想サーバーが発行した 400 (Bad Request) の応答数
iwsVsCount401	仮想サーバーが発行した 401 (Unauthorized) の応答数
iwsVsCount403	仮想サーバーが発行した 403 (Forbidden) の応答数
iwsVsCount404	仮想サーバーが発行した 404 (Not Found) の応答数
iwsVsCount503	発行された 503 (Unavailable) の応答数
iwsProcessTable	Sun ONE Web Server プロセス
iwsProcessEntry	Sun ONE Web Server プロセス
iwsProcessIndex	プロセスのインデックス
iwsProcessId	起動中のシステムプロセスの識別子
iwsProcessThreadCount	要求処理スレッド数
iwsProcessThreadIdle	現在アイドル状態の要求処理スレッド数
iwsProcessConnectionQueueCount	接続キューに入っている現在の接続数
iwsProcessConnectionQueuePeak	これまでに同時にキューに入れた接続の最大数

表 11-1 webserv61.mib の管理対象オブジェクトと説明 (続き)

管理対象オブジェクト	説明
iwsProcessConnectionQueueMax	接続キューに入れることができる最大接続数
iwsProcessConnectionQueueTotal	これまでに受け入れた接続数
iwsProcessConnectionQueueOverflows	接続キューのオーバーフローにより拒否した接続数
iwsProcessKeepaliveCount	キープアライブキューに入っている現在の接続数
iwsProcessKeepaliveMax	キープアライブキューに入れることができる最大接続数
iwsProcessSizeResident	プロセスの常駐サイズ (k バイト単位)
iwsProcessSizeVirtual	プロセスのサイズ (k バイト単位)
iwsProcessFractionSystemMemoryUsage	システムメモリのうち、プロセスメモリが占める割合
iwsListenTable	Sun ONE Web Server 待機ソケット
iwsListenEntry	Sun ONE Web Server 待機ソケット
iwsListenIndex	待機ソケットのインデックス
iwsListenId	待機ソケットの識別子
iwsListenAddress	ソケットが待機するアドレス
iwsListenPort	ソケットが待機するポート
iwsListenSecurity	暗号化のサポート
iwsThreadPoolTable	Sun ONE Web Server スレッドプール
iwsThreadPoolEntry	Sun ONE Web Server スレッドプール
iwsThreadPoolIndex	スレッドプールのインデックス
iwsThreadPoolID	スレッドプールの識別子
iwsThreadPoolCount	キューに入っている要求数
iwsThreadPoolPeak	これまでに同時にキューに入れた要求の最大数
iwsThreadPoolMax	キューに入れることができる最大要求数
iwsInstanceStatusChange	iwsInstanceStatus が変更されたことを示す iwsInstanceStatusChange トラップ

表 11-1　webserv61.mib の管理対象オブジェクトと説明 (続き)

管理対象オブジェクト	説明
iwsInstanceLoad1MinuteAverage	1 分間のシステムロードの平均
iwsInstanceLoad5MinuteAverage	5 分間のシステムロードの平均
iwsInstanceLoad15MinuteAverage	15 分間のシステムロードの平均
iwsInstanceNetworkInOctets	1 秒間にネットワーク上で転送されたオクテットの数
iwsInstanceNetworkOutOctets	1 秒間にネットワーク上で受信したオクテットの数
iwsCpuIndex	CPU インデックス
iwsCpuId	CPU ID
iwsCpuIdleTime	CPU アイドル時間
iwsCpuUserTime	CPU ユーザー時間
iwsCpuKernelTime	CPU カーネル時間

SNMP の設定

通常、SNMP を使用する場合は、システムにマスターエージェントと 1 つ以上のサブエージェントがインストールされ、実行されている必要があります。サブエージェントを使用可能にするためには、マスターエージェントをインストールする必要があります。

SNMP の設定手順は、システムによって異なります。表 11-2 は、さまざまな条件下での設定手順の概要を示しています。実際の手順は、この章の後半で詳細に説明します。

設定を開始する前に、次の 2 つの点を確認する必要があります。

- SNMP エージェント (使用するオペレーティングシステムのネイティブエージェント) がシステムですでに稼動していること
- 稼動している場合、ネイティブ SNMP エージェントが SMUX 通信をサポートしていること (AIX プラットフォームを使用している場合は、SMUX がサポートされる)

この情報を確認する方法については、使用しているシステムのマニュアルを参照してください。

注　管理サーバーの SNMP の設定を変更したあと、新しいサーバーをインストールしたあと、または既存のサーバーを削除したあとは、次の手順を実行する必要があります。

- Windows の場合は、Windows SNMP サービスを再起動するか、マシンを再起動します。
- UNIX の場合は、管理サーバーを使用して SNMP マスターエージェントを再起動します。

表 11-2　SNMP のマスターエージェントおよびサブエージェントを使用可能にする手順の概要

サーバーが満たしている条件	実行手順 (次の節で詳細に説明)
• ネイティブエージェントが現在実行されていない	1. マスターエージェントを起動します。 2. システムにインストールされている各サーバーのサブエージェントを使用可能にします。
• ネイティブエージェントが現在実行されている • SMUX をサポートしていない • ネイティブエージェントの使用を継続する必要がない	1. 管理サーバーのマスターエージェントをインストールする場合はネイティブエージェントを停止します。 2. マスターエージェントを起動します。 3. システムにインストールされている各サーバーのサブエージェントを使用可能にします。
• ネイティブエージェントが現在実行されている • SMUX をサポートしていない • ネイティブエージェントの使用を継続する必要がある	1. プロキシ SNMP エージェントをインストールします。 2. マスターエージェントを起動します。 3. プロキシ SNMP エージェントを起動します。 4. マスターエージェントのポート番号以外のポート番号を使用してネイティブエージェントを再起動します。 5. システムにインストールされている各サーバーのサブエージェントを使用可能にします。

表 11-2　SNMP のマスターエージェントおよびサブエージェントを使用可能にする手順の概要 (続き)

サーバーが満たしている条件	実行手順 (次の節で詳細に説明)
• ネイティブエージェントが現在実行されている • SMUX をサポートしている	1. SNMP ネイティブエージェントを再設定します。 2. システムにインストールされている各サーバーのサブエージェントを使用可能にします。

プロキシ SNMP エージェントの使用法 (UNIX または Linux)

ネイティブエージェントがすでに実行されていて、今後も Sun ONE Web Server マスターエージェントと同時に使用し続けたい場合には、プロキシ SNMP エージェントを使用する必要があります。ここでの手順を始める前に、ネイティブのマスターエージェントを停止してください。詳細については、使用しているシステムのマニュアルを参照してください。

注　プロキシエージェントを使用するには、このエージェントをインストールして起動する必要があります。さらに、Sun ONE Web Server のマスターエージェントが実行されているポート番号以外のポート番号を使用してネイティブ SNMP エージェントを再起動する必要があります。

この節では、次の内容について説明します。

- プロキシ SNMP エージェントのインストール
- プロキシ SNMP エージェントの起動
- ネイティブ SNMP デーモンの再起動

プロキシ SNMP エージェントのインストール

SNMP がシステムで稼動中で、ネイティブ SNMP デーモンの使用を継続する必要がある場合は、次の手順に従います。

1. SNMP マスターエージェントをインストールします。274 ページの「SNMP マスターエージェントのインストール」を参照してください。

2. プロキシ SNMP エージェントをインストールして起動し、ネイティブ SNMP デーモンを再起動します。272 ページの「プロキシ SNMP エージェントの使用法 (UNIX または Linux)」を参照してください。
3. SNMP マスターエージェントを起動します。275 ページの「SNMP マスターエージェントを使用可能にして起動する」を参照してください。
4. サブエージェントを使用可能にします。282 ページの「サブエージェントを使用可能にする」を参照してください。

SNMP プロキシエージェントをインストールするには、CONFIG ファイル (このファイルに別の名前を付けることも可能) を編集して、SNMP デーモンが待機するポートを指定します。このファイルは、サーバーのルートディレクトリの plugins/snmp/sagt にあります。また、プロキシ SNMP エージェントが転送する MIB ツリーおよびトラップも指定する必要があります。

CONFIG ファイルの例を次に示します。

```
AGENT AT PORT 1161 WITH COMMUNITY public
SUBTREES 1.3.6.1.2.1.1,
       1.3.6.1.2.1.2,
       1.3.6.1.2.1.3,
       1.3.6.1.2.1.4,
       1.3.6.1.2.1.5,
       1.3.6.1.2.1.6,
       1.3.6.1.2.1.7,
       1.3.6.1.2.1.8
FORWARD ALL TRAPS;
```

プロキシ SNMP エージェントの起動

プロキシ SNMP エージェントを起動するには、コマンドプロンプトで次のように入力します。

```
# sagt -c CONFIG&
```

ネイティブ SNMP デーモンの再起動

プロキシ SNMP エージェントを起動したあと、CONFIG ファイルで指定したポートでネイティブ SNMP デーモンを再起動する必要があります。ネイティブ SNMP デーモンを再起動するには、コマンドプロンプトで次のように入力します。

```
# snmpd -P port_number
```

port_number は、CONFIG ファイルで指定したポート番号を示します。たとえば、Solaris プラットフォームで、前に示した CONFIG ファイル例のポート番号を使用する場合は、次のように入力します。

```
# snmpd -P 1161
```

SNMP ネイティブエージェントの再設定

SNMP デーモンが AIX で稼動している場合は、SMUX がサポートされています。このため、マスターエージェントをインストールする必要はありません。ただし、AIX の SNMP デーモンの設定を変更する必要があります。

AIX では、いくつかの設定ファイルを使用して通信内容を制限しています。設定ファイルの 1 つである snmpd.conf を変更して、SNMP デーモンが SMUX サブエージェントからの着信メッセージを受け入れるようにする必要があります。詳細については、オンラインマニュアルの snmpd.conf のページを参照してください。各サブエージェントを定義する行を追加する必要があります。

たとえば、snmpd.conf に次の行を追加します。

```
smux 1.3.6.1.4.1.1.1450.1 "" IP_address net_mask
```

IP_address は、そのサブエージェントを実行するホストの IP アドレス、*net_mask* は、そのホストのネットワークマスクを示します。

注	ループバックアドレスの 127.0.0.1 は使用できません。実 IP アドレスを使用してください。

SNMP マスターエージェントのインストール

SNMP マスターエージェントを設定するには、root ユーザーとして管理サーバーインスタンスをインストールする必要があります。ただし、root 以外のユーザーでも、マスターエージェントとやり取りを行う SNMP サブエージェントを設定することで、MIB のブラウズなどの基本的な SNMP タスクを Web サーバーインスタンスから実行できます。

サーバーマネージャを使用してマスター SNMP エージェントをインストールするには、次の手順を実行します。

1. スーパーユーザー (root) としてログインします。

2. SNMP デーモン (snmpd) がポート 161 で実行されているかどうか確認します。

   SNMP デーモンが実行されていない場合は、手順 4 に進みます。

   SNMP デーモンが実行されている場合は、その再起動方法と、どの MIB ツリーをサポートしているかを確認します。

3. SNMP デーモンが実行されている場合は、そのプロセスを終了します。
4. サーバーマネージャで、「Global Settings」タブから「SNMP Master Agent Trap」ページを選択します。「Manager Entries」ページが表示されます。
5. ネットワーク管理ソフトウェアを実行するシステムの名前を入力します。
6. ネットワーク管理システムがトラップを待機するポート番号を入力します。一般的なポートは 162 です。トラップの詳細については、281 ページの「トラップ送信先の設定」を参照してください。
7. トラップで使用するコミュニティ文字列を入力します。コミュニティ文字列の詳細については、281 ページの「コミュニティ文字列の設定」を参照してください。
8. 「OK」をクリックします。
9. サーバーマネージャで、「Global Settings」タブから「SNMP Master Agent Community」ページを選択します。「Community Strings」ページが表示されます。
10. マスターエージェントのコミュニティ文字列を入力します。
11. コミュニティの動作を選択します。
12. 「OK」をクリックします。

SNMP マスターエージェントを使用可能にして起動する

マスターエージェントの動作は、`CONFIG` という名前のエージェント設定ファイルに定義されています。サーバーマネージャを使用して `CONFIG` ファイルを編集できます。また、手動でこのファイルを編集することもできます。SNMP サブエージェントを使用可能にするためには、マスター SNMP エージェントをインストールする必要があります。

マスターエージェントを再起動しようとしたときに、「System Error: Could not bind to port」のようなバインドエラーが発生する場合は、`ps -ef | grep snmp` を使用して、`magt` が実行されているかどうかを確認します。実行されている場合は、`kill -9` *pid* コマンドを使用して、そのプロセスを終了します。SNMP 用の CGI がふたたび機能し始めます。

この節では、次の内容について説明します。

- マスターエージェントを別のポートで起動する
- SNMP マスターエージェントを手動で設定する
- マスターエージェントの CONFIG ファイルの編集
- sysContact 変数と sysLocation 変数の定義
- SNMP サブエージェントの設定
- SNMP マスターエージェントの起動

マスターエージェントを別のポートで起動する

管理インタフェースでは、161 以外の ポートで SNMP マスターエージェントを起動できません。ただし、手動操作により、他のポートで SNMP マスターエージェントを起動できます。その手順を次に示します。

1. /*server_root*/plugins/snmp/magt/CONFIG を編集して、目的のポートを指定します (*server_root* はサーバールートを示します)。
2. 次のように起動スクリプトを実行します。

   ```
   cd /server_root/https-admserv

   ./start -shell /server_root/plugins/snmp/magt/magt

   /server_root/plugins/snmp/magt/CONFIG

   /server_root/plugins/snmp/magt/INIT
   ```

これで、マスターエージェントが目的のポートで起動します。マスターエージェントが動作していることは、ユーザーインタフェースから検出できます。

SNMP マスターエージェントを手動で設定する

マスター SNMP エージェントを手動で設定するには、次の手順を実行します。

1. スーパーユーザーとしてログインします。
2. SNMP デーモン (snmpd) がポート 161 で実行されているかどうか確認します。

   SNMP デーモンが実行されている場合は、その再起動方法と、どの MIB ツリーをサポートしているかを確認します。次に、そのプロセスを終了します。
3. サーバーのルートディレクトリの plugins/snmp/magt にある CONFIG ファイルを編集します。

4. (省略可能) CONFIG ファイルに sysContact 変数と sysLocation 変数を定義します。

マスターエージェントの CONFIG ファイルの編集

CONFIG ファイルには、マスターエージェントが処理するコミュニティおよびマネージャが定義されています。マネージャの値は、有効なシステム名または IP アドレスである必要があります。

基本的な CONFIG ファイルの例を次に示します。

```
COMMUNITY           public
                    ALLOW ALL OPERATIONS

MANAGER             manager_station_name
                    SEND ALL TRAPS TO PORT 162
                     WITH COMMUNITY public
```

sysContact 変数と sysLocation 変数の定義

CONFIG ファイルを編集して、MIB-II 変数の sysContact と sysLocation を指定する sysContact と sysLocation の初期値を追加できます。この例では、sysContact および sysLocation に指定する文字列が引用符で囲まれています。空白文字、改行、タブなどを含む文字列は、引用符で囲む必要があります。また、16 進法表記で値を指定することもできます。

sysContact 変数および sysLocation 変数が定義された CONFIG ファイルの例を次に示します。

```
COMMUNITY           public
                    ALLOW ALL OPERATIONS

MANAGER             nms2
                    SEND ALL TRAPS TO PORT 162
                    WITH COMMUNITY public

INITIAL             sysLocation "Server room
501 East Middlefield Road
Mountain View, CA 94043
USA"

INITIAL             sysContact "John Doe
email: jdoe@netscape.com"
```

SNMP サブエージェントの設定

SNMP サブエージェントを設定してサーバーを監視することができます。

SNMP サブエージェントを設定するには、次の手順を実行します。

1. 管理サーバーからサーバーインスタンスを選択し、「Manage」をクリックします。
2. 「Monitor」タブを選択します。
3. 「SNMP Subagent Configuration」を選択します。
4. (UNIX のみ) 「Master Host」フィールドにサーバーの名前とドメインを入力します。
5. 「Description」にサーバーの説明 (オペレーティングシステム情報を含む) を指定します。
6. 「Organization」にサーバーを管理する組織を指定します。
7. 「Location」フィールドにサーバーの絶対パスを指定します。
8. 「Contact」フィールドにサーバーの担当者名と、担当者の連絡先情報を指定します。
9. 「Enable the SNMP Statistics Collection」の「On」を選択します。
10. 「OK」をクリックします。
11. 「Apply」をクリックします。
12. 「Apply Changes」を選択して、変更をサーバーに適用します。

SNMP マスターエージェントの起動

SNMP マスターエージェントはインストールしたあと、手動で、または管理サーバーを使用して起動できます。

手動による SNMP マスターエージェントの起動

マスターエージェントを手動で起動するには、コマンドプロンプトで次のように入力します。

```
# magt CONFIG INIT&
```

INIT ファイルは、システムの場所や連絡先情報など、MIB-II システムグループからの情報が格納された不揮発性ファイルです。INIT ファイルが既存しない場合は、マスターエージェントを最初に起動した時に作成されます。無効なマネージャ名が CONFIG ファイルに指定されていると、マスターエージェントの起動が失敗する原因になります。

マスターエージェントを標準以外のポートで起動するには、次の 2 つの方法のどちらかを使用します。

方法 1 :CONFIG ファイルに、マスターエージェントがマネージャからの SNMP 要求を待機する各インタフェースのトランスポートマッピングを指定します。トランスポートマッピングを指定することで、マスターエージェントは標準ポートと標準以外のポートで接続を受け入れることができます。また、マスターエージェントは、標準以外のポートで SNMP トラフィックを受け入れることもできます。同時に使用可能な SNMP の最大数は、プロセス当たりのオープンソケット数またはファイル記述子数に関するシステムの制限値によって制限されます。トランスポートマッピングのエントリ例を次に示します。

```
TRANSPORT           extraordinary   SNMP
                    OVER UDP SOCKET
                     AT PORT 11161
```

CONFIG ファイルを手動で編集したあと、コマンドプロンプトで次のように入力して、手動でマスターエージェントを起動する必要があります。

```
# magt CONFIG INIT&
```

方法 2 :/etc/services ファイルを編集して、マスターエージェントが標準ポートと標準以外のポートで接続を受け入れられるようにします。

管理サーバーを使用して SNMP マスターエージェントを起動する

管理サーバーを使用して SNMP マスターエージェントを起動するには、次の手順を実行します。

1. 管理サーバーにログインします。

2. サーバーマネージャで、「Global Settings」タブから「SNMP Master Agent Control」ページを選択します。「SNMP Master Agent Control」ページが表示されます。
3. 「Start」をクリックします。

   「SNMP Master Agent Control」ページから、SNMP エージェントの停止および再起動も実行できます。

SNMP マスターエージェントの設定

マスターエージェントを使用可能にし、ホストコンピュータのサブエージェントを使用可能にしたあと、ホストの管理サーバーを設定する必要があります。そのためには、コミュニティ文字列とトラップ送信先を指定する必要があります。

コミュニティ文字列の設定

コミュニティ文字列は、SNMP エージェントが認証に使用するテキスト文字列です。ネットワークマネージメントステーションは、エージェントに送信する各メッセージと一緒にコミュニティ文字列を送信します。この結果、エージェントは、そのネットワークマネージメントステーションが情報の取得を承認されているかどうかを確認できます。コミュニティ文字列は、SNMP パケットでの送信時に秘匿されることはなく、ASCII テキストで送信されます。

SNMP マスターエージェントのコミュニティ文字列は、サーバーマネージャの「Community Strings」ページから設定できます。また、特定のコミュニティで実行できる SNMP 関連オペレーションを定義することもできます。サーバーマネージャから、設定済みのコミュニティの表示、編集、および削除を行うこともできます。

トラップ送信先の設定

SNMP トラップとは、SNMP エージェントがネットワークマネージメントステーションに送信するメッセージのことです。たとえば、SNMP エージェントは、インタフェースの状態が稼動から停止に変わった時にトラップを送信します。SNMP エージェントにトラップの送信先がわかるように、ネットワークマネージメントステーションのアドレスを設定する必要があります。SNMP マスターエージェントのトラップ送信先は、Sun ONE Web Server から設定できます。また、設定済みのトラップ送信先の表示、編集、および削除を行うこともできます。Sun ONE Web Server を使用してトラップ送信先を設定する場合、実際には、`CONFIG` ファイルを編集することになります。

サブエージェントを使用可能にする

管理サーバーに付属するマスターエージェントをインストールしたあと、そのマスターエージェントを起動する前に、サーバーインスタンスのサブエージェントを使用可能にする必要があります。マスターエージェントのインストールについては、274 ページの「SNMP マスターエージェントのインストール」を参照してください。サーバーマネージャを使用してサブエージェントを使用可能にできます。

UNIX プラットフォームまたは Linux プラットフォームで SNMP 機能を停止する場合は、サブエージェントを先に停止し、その後でマスターエージェントを停止する必要があります。マスターエージェントを先に停止すると、サブエージェントを停止できなくなることがあります。そうなった場合は、マスターエージェントを再起動し、サブエージェントを停止し、次にマスターエージェントを停止します。

SNMP サブエージェントを使用可能にするには、サーバーマネージャの「SNMP Subagent Configuration」ページを使用し、「SNMP Subagent Control」ページからサブエージェントを起動します。詳細については、オンラインヘルプの対応する項目を参照してください。

サブエージェントを使用可能にすると、「SNMP Subagent Control」ページ、または Windows のコントロールパネルの「サービス」からそのサブエージェントを起動、停止、または再起動できます。

注	SNMP の設定を変更したあとは、「Apply」ボタンをクリックし、SNMP サブエージェントを再起動する必要があります。

SNMP メッセージについて

GET および SET は、SNMP で定義されている 2 種類のメッセージです。GET メッセージと SET メッセージは、ネットワークマネージメントステーション (NMS) によってマスターエージェントに送信されます。管理サーバーで、これらのメッセージのどちらか一方または両方を使用できます。

SNMP は、プロトコルデータユニット (PDU) の形式でネットワーク情報をやり取りします。このユニットには、Web サーバーなどの管理対象デバイスに保存された変数に関する情報が格納されます。これらの変数は、管理対象オブジェクトとも呼ばれ、必要に応じて NMS に報告される値とタイトルを含んでいます。サーバーから NMS に送信されるプロトコルデータユニットは「トラップ」と呼ばれます。次の例では、GET、SET、およびトラップの各メッセージの使用法を示します。

NMS 主導の通信 : NMS は、サーバーからの情報を要求するか、サーバーの MIB 内に格納されている変数の値を変更します。その例を次に示します。

1. NMS は、管理サーバーのマスターエージェントにメッセージを送信します。このメッセージは、データの要求 (GET メッセージ) の場合と、MIB の変数を設定する命令 (SET メッセージ) の場合があります。
2. マスターエージェントは、そのメッセージを適切なサブエージェントに転送します。
3. サブエージェントは、データを取り出すか、または MIB 内の変数を変更します。
4. サブエージェントは、マスターエージェントにデータまたは状態を報告します。次に、マスターエージェントはそのメッセージ (GET メッセージ) を NMS に返送します。
5. NMS は、ネットワーク管理アプリケーションを通して、そのデータを文字またはグラフィックで表示します。

サーバー主導の通信：サーバーのサブエージェントは、重大なイベントが発生した時に、メッセージまたはトラップを NMS を送信します。その例を次に示します。

1. サブエージェントは、サーバーが停止したことをマスターエージェントに通知します。
2. マスターエージェントは、イベントを報告するメッセージまたはトラップを NMS に送信します。
3. NMS は、ネットワーク管理アプリケーションを通して、その情報を文字またはグラフィックで表示します。

第 12 章

ネーミングとリソースの設定

コンポーネントベースの J2EE™ (Java™ 2 Platform, Enterprise Edition) テクノロジには、エンタープライズレベルの開発と配備を簡略化するために、Web サービス用のインフラストラクチャが用意されています。

この章では、Sun ONE Web Server が提供する J2EE リソースを紹介し、これらのリソースを作成、管理する方法について説明します。

Java のセキュリティとレルムベースの認証については、第 4 章「Web コンテナと Web アプリケーションの J2EE ベースのセキュリティ」を参照してください。

この章には、次の内容が記述されています。

- Java の有効化と無効化
- JVM の設定
- J2EE ネーミングサービスおよびリソースについて
- JNDI (Java Naming and Directory Interface) について
- Java ベースのリソースの作成
- Java ベースのリソースの変更
- Java ベースのリソースの削除

Java の有効化と無効化

Java の有効と無効の設定は、Sun ONE Web Server のインスタンス単位でグローバルで行うか、特定の仮想サーバークラス単位で行うことができます。デフォルトでは、Sun ONE Web Server では Java が有効になっていて、`magnus.conf` ファイルには次の行が追加されています。

```
Init fn="load-modules"
shlib="<server-root>/bin/https/lib/libj2eeplugin.so"
```

特定の仮想サーバーで Java を有効にすることもできます。この場合、サーバーはその仮想サーバーの `obj.conf` ファイルを、必要な J2EE 指令で更新します。

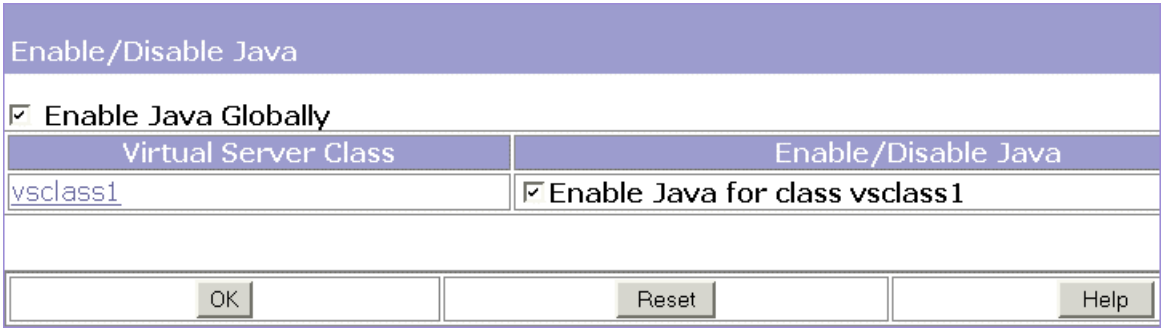
`obj.conf` ファイルと `magnus.conf` ファイルの詳細については、『Sun ONE Web Server 6.1 Administrator's Configuration File Reference』および『Sun ONE Web Server 6.1 NSAPI Programmer's Guide』を参照してください。

サーバー全体、または特定の仮想サーバークラスがスタティックコンテンツだけを提供する場合など、Java をグローバルに、または特定の仮想サーバークラスで無効にすることが必要な場合もあります。

Java を有効または無効にするには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「Enable/Disable Servlets/JSP」をクリックします。

「Enable/Disable Servlets/JSP」インタフェース



3. Java をグローバルに有効または無効にするときは、「Enable/Disable Java Globally」を選択するか、選択解除します。

   または特定の仮想サーバークラスで Java を有効または無効にするときは、対応する仮想サーバークラスの「Enable/Disable Java」を選択するか、選択解除します。
4. 「OK」をクリックします。

JVM の設定

従来のバージョンとは異なり、Sun ONE Web Server 6.1 はスタンドアロン Java 実行時環境 (Java Runtime Environment、JRE) をサポートしなくなりました。その代わりに、サーバーでは JDK 1.4.1 以降が必須となります。サーバーをインストールするときに、デフォルトの JDK オプションを選択した場合は、*server-root*/bin/https/jdk ディレクトリに JDK (Java Development Kit) 1.4.1_03 がインストールされます。

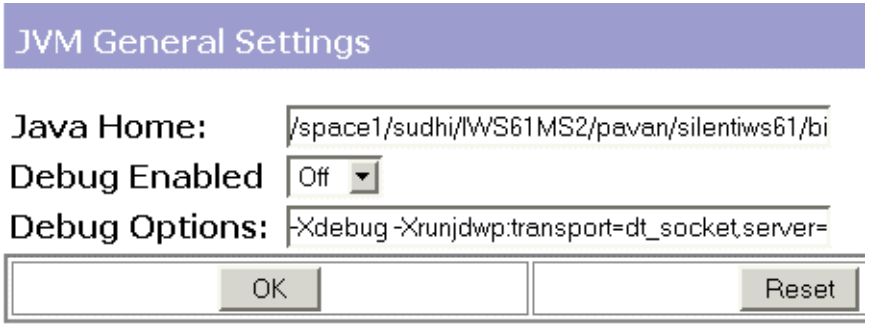
サーバーインスタンスに合わせて、JVM (Java Virtual Machine) を設定することができます。この設定には、Java ホームの場所、コンパイラオプション、デバッグオプション、プロファイラ情報などが含まれます。これらの設定を行う理由の一つは、パフォーマンスの向上です。パフォーマンスの詳細については、『Sun ONE Web Server 6.1 Performance Tuning, Sizing, and Scaling Guide』を参照してください。

一般設定

JDK の場所を編集し、デバッグオプションを指定するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「JVM General」をクリックします。

「JVM General」インタフェース



3. 「Java Home」を設定します。

   Java Home は、JDK (Java Developer's Kit) がインストールされているディレクトリへのパスです。Sun ONE Web Server は、Sun JDK 1.4.1_03 をサポートしています。

4. デバッグを有効にするかどうかを選択し、デバッグオプションを設定します。

   デバッグオプションのリストは、次の場所で参照できます。
   `http://java.sun.com/products/jpda/doc/conninv.html#Invocation`

5. 「OK」をクリックします。

パスの設定

JVM パスの設定が必要な場合もあります。たとえば、XML Parser クラスなどのシステムクラスの設定を置き換えたるためにシステムクラスパスのサフィックスを選択したり、環境変数により運用環境に悪影響を与えないために環境クラスパスを無視するように設定することが必要になる場合があります。

管理インタフェースで JVM のパスを設定するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「JVM Path Settings」をクリックします。
3. システムのクラスパスのサフィックスを選択します。
4. 環境変数のクラスパスを無視するかどうかを指定します。

   クラスパスを無視しない場合、`CLASSPATH` 環境変数が読み込まれ、Sun ONE Web Server のクラスパスに追加されます。`CLASSPATH` 環境変数は、`classpathsuffix` の一番最後の部分に追加されます。

   開発環境では、クラスパスを使用してください。運用環境では、このクラスパスを無視して環境変数に影響を与えないようにする必要があります。

5. ネイティブライブラリパスのプレフィックスとサフィックスを設定します。

   ネイティブライブラリパスは、Web Server のインストール相対パスに、ネイティブ共有ライブラリ、標準 JRE ネイティブライブラリパス、シェル環境設定 (UNIX の `LD_LIBRARY_PATH`)、さらに `profiler` 要素で指定されるすべてのパスを指定することで、自動的にこれらを連結して作成されます。これは合成されたものなので、明示的にサーバー設定としては表示されません。

6. 「OK」をクリックします。

JVM オプションの設定

管理インタフェースで JVM コマンド行オプションを設定するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。

2. 「JVM Options」をクリックし、必要な変更を加えます。

   各 JVM オプションについては、次の URL を参照してください。

   `http://java.sun.com/docs/hotspot/VMOptions.html`

3. 「OK」をクリックします。

JVM プロファイラの設定

サーバー側のパフォーマンスのボトルネックを見つけるために、プロファイラを使って、Sun ONE Web Server 上でリモートプロファイリングを行うことができます。

管理インタフェースで JVM プロファイラを設定するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「JVM Profiler」をクリックします。
3. クラスパスとネイティブライブラリパスを指定し、プロファイラを有効にするかどうかを指定します。
4. プロファイラの JVM オプションを追加、削除、編集し、「OK」をクリックします。

プロファイラの詳細については、『Sun ONE Web Server 6.1 Programmer's Guide』を参照してください。

J2EE ネーミングサービスおよびリソースについて

Web アプリケーションは、リソースマネージャ、データソース (たとえば SQL データソースなど)、メールセッション、URL 接続ファクトリなど、さまざまなリソースにアクセスします。J2EE プラットフォームは、JNDI (Java Naming and Directory Interface) を経由して、これらのリソースをアプリケーションに公開します。

Sun ONE Web Server では、次の J2EE リソースを作成、管理できます。

- JDBC データソース
- JDBC 接続プール
- Java メールセッション
- カスタムリソース
- 外部 JNDI リソース

JDBC データソース

JDBC データソースは、Sun ONE Web Server で作成、管理できる J2EE リソースです。

JDBC API は、リレーショナルデータベースシステムとの接続用 API です。JDBC API は、次の 2 つの部分に分かれています。

- データベースにアクセスするためにアプリケーションコンポーネントが使用する、アプリケーションレベルのインタフェース
- JDBC ドライバを J2EE プラットフォームに接続するための、サービスプロバイダインタフェース

JDBC データソースオブジェクトは、Java プログラミング言語へのデータソースの実装です。データソースは、基本的にはデータを格納するための機能です。これは、大企業向けの複雑なデータベースである場合もあり、行と列から構成される簡単なファイルである場合もあります。JDBC データソースは、Sun ONE Web Server 経由で作成、管理できる J2EE リソースです。

JDBC API は、さまざまなリレーショナルデータベースに安定してアクセスできるように、標準 SQL データベースアクセスインタフェース用のクラスセットを Java に提供します。

JDBC を使用することで、仮想的にあらゆるデータベース管理システム (DBMS) に SQL 文を送信できます。これは、リレーショナル DBMS とオブジェクト DBMS の両方のインタフェースとして使用されます。

カスタムリソースの作成については、「JDBC リソースの作成」を参照してください。

JDBC 接続プール

JDBC 接続プールは、データベースへの JDBC 接続のグループで、名前が付けられています。これらの接続は、Sun ONE Web Server の起動時に、プールで最初の接続要求が行われるときに作成されます。

JDBC 接続プールは、接続プールの作成に使用されるプロパティを定義します。個々の接続プールは、サーバーの起動時に、JDBC ドライバを使用して実際のデータベースとの接続を確立します。

JDBC ベースのアプリケーションまたはリソースは、プールとの接続を確立してそれを使用し、接続が不要になると、接続を閉じて接続プールに戻します。複数の JDBC リソースが同じプール定義を指定する場合、実行時に同じ接続プールが使用されます。

新しい JDBC 接続プールの作成方法については、「JDBC 接続プールの新規作成」を参照してください。

Java メールセッション

JMS 送信先は、Sun ONE Web Server 経由で作成、管理できる J2EE リソースです。

多くのインターネットアプリケーションでは、電子メールによる通知を送信する機能を必要とするため、J2EE プラットフォームには、アプリケーションコンポーネントがインターネットメールを送信できるように、JavaMail API と JavaMail サービスプロバイダが用意されています。JavaMail API は、次の 2 つの部分に分かれています。

- メールを送信するためにアプリケーションコンポーネントが使用する、アプリケーションレベルのインタフェース
- J2EE API レベルで使用されるサービスプロバイダインタフェース

JMS メールセッションは、Sun ONE Web Server 経由で作成、管理できる J2EE リソースです。

注 Sun ONE Web Server には、Java メールセッションを作成するための管理サーバーインタフェースが用意されていません。この操作は、コマンド行インタフェースを使って実行できます。コマンド行ユーティリティを使ってメールリソースを作成する方法については、「メールリソースの作成」を参照してください。

カスタムリソース

カスタムリソースは、ローカル JNDI リポジトリにアクセスします。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には、`server.xml` に定義されている `customresource` 要素を使用します。このような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ます。この要素は、サーバー全体のネームスペースで使用される JNDI 名 (その他の Sun ONE Web Server リソースと同様に、`jndiname` サブ要素を使って指定されます) と、そのタイプ、リソースファクトリクラスの名前、および同じインスタンスの作成に使用される標準プロパティを関連付けます。

リソース参照の環境参照が、`server.xml` の `customresource` タグおよび `externaljndiresource` タグによって定義される、設定済みのサーバー全体のリソースにリンクされていることを確認する必要があります。アプリケーションコンポーネントのダイナミックな再配備は、JNDI ネーミング環境で問題となります。Sun ONE Web Server は、アプリケーションに固有のすべての参照を解放し、新たにインストールされたアプリケーションのネーミングコンテキスト内で、新しい参照をすべて作成し直します。

カスタムリソースの作成については、「カスタムリソースの作成」を参照してください。

外部 JNDI リソース

Sun ONE Web Server で稼動するアプリケーションの多くは、外部 JNDI リポジトリに格納されているリソースにアクセスします。たとえば、汎用 Java オブジェクトは、Java スキーマとして LDAP サーバーに格納することができます。カスタムリソースではローカル JNDI リポジトリにアクセスできましたが、外部 JNDI リポジトリにアクセスするには、外部 JNDI リソースを使用しなければなりません。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。

外部 JNDI リソースの作成については、「外部 JNDI リソースの作成」を参照してください。

JNDI (Java Naming and Directory Interface) について

ここでは、JNDI (Java Naming and Directory Interface) について説明します。JNDI は、さまざまなネーミングサービスやディレクトリサービスにアクセスするためのアプリケーションプログラミングインタフェース (API) です。J2EE コンポーネントは、JNDI 検索メソッドを呼び出してオブジェクトを特定します。

この節では、次の内容について説明します。

- J2EE ネーミングサービス
- ネーミング参照とバインド情報
- J2EE 標準配備記述子内のネーミング参照
- JNDI 接続ファクトリ

J2EE ネーミングサービス

JNDI 名は、ユーザーにとって理解しやすいオブジェクト名です。これらの名前は、J2EE サーバーが提供するネーミングとディレクトリのサービスによって、実際のオブジェクトにバインドされます。J2EE コンポーネントは、JNDI API 経由でこのサービスにアクセスするため、ユーザーが理解しやすいオブジェクト名を JNDI 名と呼びます。たとえば、Oracle データベースの JNDI 名としては、`jdbc/Oracle` が考えられます。Sun ONE Web Server を起動すると、設定ファイルの情報が読み取られ、JNDI データベース名が自動的にネームスペースに追加されます。

アプリケーションコンポーネントのネーミング環境は、運用時またはアセンブル時にアプリケーションコンポーネントのビジネスロジックをカスタマイズできるメカニズムです。アプリケーションコンポーネントの環境を使用することで、アプリケーションコンポーネントのソースコードを変更せずに、アプリケーションコンポーネントをカスタマイズすることができます。

J2EE コンテナには Web アプリケーションコンポーネントの環境が実装されて、アプリケーションコンポーネントインスタンスに、JNDI ネーミングコンテキストとして提供されます。アプリケーションコンポーネントの環境は、次のように使用されます。

- Web アプリケーションコンポーネントのビジネスメソッドは、JNDI インタフェースを使用して環境にアクセスします。アプリケーションコンポーネントプロバイダは、実行時にアプリケーションコンポーネントがその環境で必要とする可能性のあるすべての環境エントリを、配備記述子内で宣言します。
- コンテナは、アプリケーションコンポーネント環境を格納する JNDI ネーミングコンテキストの実装を提供します。コンテナは、デプロイヤが各アプリケーションコンポーネントの環境の作成と管理に使用するツールも提供します。
- デプロイヤは、コンテナが提供するツールを使用して、アプリケーションコンポーネントの配備記述子で宣言された環境エントリを初期化します。デプロイヤは、環境エントリの値を設定および変更することができます。
- コンテナは、実行時にアプリケーションコンポーネントインスタンスが、環境ネーミングコンテキストを使用できるようにします。アプリケーションコンポーネントのインスタンスは、JNDI インタフェースを使用して、環境エントリの値を取得します。

各アプリケーションコンポーネントは、独自の環境エントリセットを定義します。同じコンテナに含まれるアプリケーションコンポーネントのすべてのインスタンスは、同じ環境エントリを共有します。アプリケーションコンポーネントインスタンスは、実行時に環境を変更することができません。

ネーミング参照とバインド情報

リソース参照は、配備記述子内の要素で、コンポーネントのコード化されたリソース名を識別します。より具体的には、コード化された名前は、そのリソースの接続ファクトリを参照します。次の項で紹介する例では、リソース参照名は `jdbc/SavingsAccountDB` です。

リソースの JNDI 名と、リソース参照の名前は異なります。この命名方法では、配備の前に 2 つの名前をマッピングする必要がありますが、リソースからコンポーネントが切り離されることにもなります。この切り離しにより、後にコンポーネントが別のリソースにアクセスしなければならなくなったときに、コード内で名前を変更する必要がなくなります。また、この柔軟性により、既存のコンポーネントから J2EE アプリケーションを構築することが容易になります。

次の表は、推奨される JNDI 検索と、Sun ONE Web Server で使用される、それぞれに関連付けられている J2EE リソースの参照を示しています。

表 12-1 JNDI 検索と関連参照

JNDI 検索名	関連付けられている参照
`java:comp/env`	アプリケーション環境エントリ
`java:comp/env/jdbc`	JDBC DataSource リソース
`java:comp/env/mail`	JavaMail セッション接続ファクトリ
`java:comp/env/url`	URL 接続ファクトリ

J2EE 標準配備記述子内のネーミング参照

ネーミング参照は、アプリケーションが指定のネーミングコンテキストからオブジェクトを検索するために使用する文字列です。各 Web アプリケーションにはネーミングコンテキストがあり、参照は標準のコンポーネント配備記述子に設定されています。ここでは、Sun ONE Web Server で使用される標準の配備記述子の機能について説明します。この節では、次の内容について説明します。

- アプリケーション環境エントリ
- リソースへの参照
- リソース環境参照

アプリケーション環境エントリ

<env-entry> によって定義される環境エントリは、配備時のパラメータを J2EE Web アプリケーションに指定するための方法です。<context-param> を使用してサーブレットコンテキスト初期化パラメータを定義することもできますが、アプリケーションデプロイヤは、名前、タイプ、値を明示的に指定してこのようなアプリケーションパラメータを設定するため、<env-entry> のほうが好まれます。

次の例は、J2EE 標準配備記述子に指定される <env-entry> の構文を示しています。

```
<env-entry>
<description> Send pincode by mail </description>
<env-entry-name> mailPincode </env-entry-name>
<env-entry-value> false </env-entry-value>
<env-entry-type> java.lang.Boolean </env-entry-type>
</env-entry>
```

<env-entry-type> タグは、エントリの完全修飾クラス名を指定します。次に、JNDI を使用してサーブレットまたは JSP から <env-entry> を検索するコード例を示します。

```
Context initContext = new InitialContext();
Boolean mailPincode = (Boolean)
initContext.lookup("java:comp/env/mailPincode");
// one could use relative names into the sub-context
Context envContext = initContext.lookup("java:comp/env");
Boolean mailPincode = (Boolean)
envContext.lookup("mailPincode");
```

リソースへの参照

ファクトリは、その他のオブジェクトを必要に応じて作成するオブジェクトです。リソースファクトリは、データベース接続やメッセージサービス接続などのリソースオブジェクトを作成します。これは、<resource-ref> 要素を使用して、標準配備記述子に設定されます。

次に、ファクトリの使用例を示します。

例

javax.sql.DataSource というタイプのオブジェクトを返す JDBC 接続ファクトリへの参照の宣言

```
<resource-ref>
```

```
<description> Primary database </description>
<res-ref-name> jdbc/primaryDB </res-ref-name>
<res-type> javax.sql.DataSource </res-type>
<res-auth> Container </res-auth>
</resource-ref>
```

<res-type> は、リソースファクトリの完全修飾クラス名です。<res-auth> 変数には、値として Container または Application を割り当てることができます。

Container を指定した場合、Web コンテナは、リソースファクトリを JNDI 検索レジストリにバインドする前に認証を処理します。Application を指定した場合、サーブレットがプログラム上で認証を処理する必要があります。リソースのタイプを指定する異なるサブコンテキストによって、異なるリソースファクトリが検索されます。

- jdbc/ の場合は JDBC javax.sql.DataSource ファクトリ
- mail/ の場合は javax.mail.Session ファクトリ
- url/ の場合は java.net.URL ファクトリ

次に、アプリケーションコンポーネントから JDBC 接続を取得し、コンテナが認証を処理するコード例を示します。

```
InitialContext initContext = new InitialContext();
DataSource source =
(DataSource) initContext.lookup("java:comp/env/jdbc/primaryDB");
Connection conn = source.getConnection();
```

これらのリソース参照が確実に機能するには、実行時に res-ref-name が有効なリソースファクトリにマッピングされる必要があることに注意してください。

リソース環境参照

リソース環境参照は、JNDI 検索によって、リソースに関連付けられている管理対象オブジェクトにアクセスするための方法です。アプリケーションは、標準配備記述子に定義される <resource-env-ref> 要素を使用してリソース要件を宣言します。

<resource-env-ref> 要素と <resource-ref> 要素の最大の違いは、特定のリソース認証要件が指定されていないことです。これらの要素は、どちらもリソースファクトリ記述子によってバックアップされる必要があります。

例

```
<resource-env-ref>
    <description> My Topic </description>
```

```
    <res-env-ref-name> jdbc/MyTopic </res-ref-name>
    <res-env-ref-type> javax.jdbc.Topic </res-type>
</resource-env-ref>
```

次に、JMS Topic オブジェクトにアクセスするためのコード例を示します。

```
InitialContext initContext = new InitialContext();
javax.jms.Topic myTopic = (javax.jdbc.Topic)
initContext.lookup("java:comp/env/jdbc/MyTopic");
```

初期ネーミングコンテキスト

Sun ONE Web Server によるネーミングのサポートは主に J2EE 1.3 に基づき、それにいくつかの拡張が追加されています。アプリケーションコンポーネントが `InitialContext()` を通じて初期コンテキストを作成すると、Sun ONE Web Server は、その Web アプリケーションのネーミング環境へのハンドルとして機能するオブジェクトを返します。このオブジェクトは、`java:comp/env` ネームスペースのサブコンテキストを提供します。各 Web アプリケーションは固有のネームスペースを取得します。つまり、`java:comp/env` ネームスペースは Web アプリケーションごとに異なり、ある Web アプリケーションでネームスペースにバインドされるオブジェクトは、別の Web アプリケーションでバインドされるオブジェクトと競合しません。

JNDI 接続ファクトリ

J2EE Web アプリケーションでは、`web.xml` ファイル内の配備記述子は、アプリケーション環境エントリ、またはリソースマネージャ (SQL データソースなど) 接続ファクトリへの参照を定義するためのプレイスホルダです。アプリケーションは、J2EE コンテナが提供する `JNDI InitialNamingContext` を使用してこの参照を検索します。これにより、アプリケーションのソースコードにアクセスしたり、それを変更したりすることなく、配備記述子に変更を加えるだけで、別の Web Server 環境でアプリケーションを利用できます。

接続ファクトリは、J2EE コンポーネントがリソースにアクセスするための接続オブジェクトを生成するオブジェクトです。データベースの接続ファクトリは `javax.sql.DataSource` オブジェクトで、これは `java.sql.Connection` オブジェクトを作成します。

Sun ONE Web Server では、次のリソースおよびリソースファクトリにアクセスする方法を設定できます。

- JDBC 接続ファクトリ
- JavaMail セッション接続ファクトリ

- ユーザーが記述する汎用のカスタムリソースオブジェクトファクトリ
- LDAP などの外部リソースリポジトリのサポート

Sun ONE Web Server のすべてのリソースファクトリは、`server.xml` ファイルの `<resources> </resources>` タグ内に指定され、`jndiname` 属性によって指定される JNDI 名を持ちます (例外として、`jdbconnectionpool` は JNDI 名を持ちません)。この属性は、サーバー全体のネームスペースへのファクトリの登録に使用されます。デプロイヤは、`sun-web.xml` ファイルの `resource-ref` 要素を使用して、ユーザーが指定したアプリケーション固有のリソース参照名 (`resource-ref` 要素または `resource-env-ref` 要素内で宣言されます) を、このサーバー全体のリソースファクトリにマッピングできます。これにより、特定のアプリケーションでどの JDBC リソース (およびその他のリソースファクトリ) を使用するかを、配備時に決定できます。

カスタムリソースはローカル JNDI リポジトリにアクセスし、外部リソースは外部 JNDI リポジトリにアクセスします。どちらのリソースも、ユーザー指定のファクトリクラス要素、JNDI 名属性などを必要とします。

ここでは、各種 J2EE リソースを作成する方法、およびこれらのリソースにアクセスする方法について説明します。

- Java ベースのリソースの作成
- Java ベースのリソースの変更

Java ベースのリソースの作成

ここでは、管理インタフェースを使用して各種 J2EE ベースリソースを作成する方法について説明します。

- JDBC 接続プールの新規作成
- JDBC リソースの作成
- カスタムリソースの作成
- 外部 JNDI リソースの作成

JDBC 接続プールの新規作成

新しい JDBC 接続プールは、次の方法で作成できます。

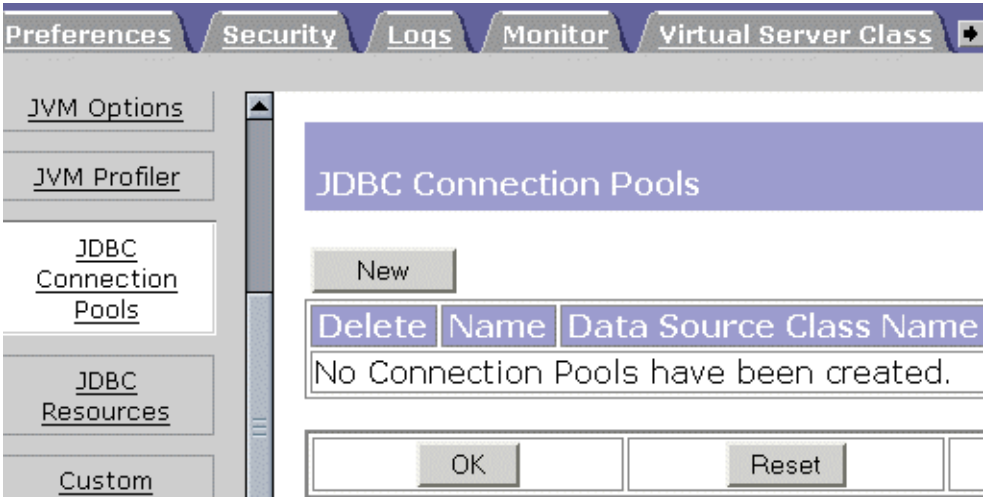
- 管理インタフェースを使用
- コマンド行インタフェースを使用

管理インタフェースを使用

管理インタフェースを使用して新しい JDBC 接続プールを作成するには、次の手順を実行します。

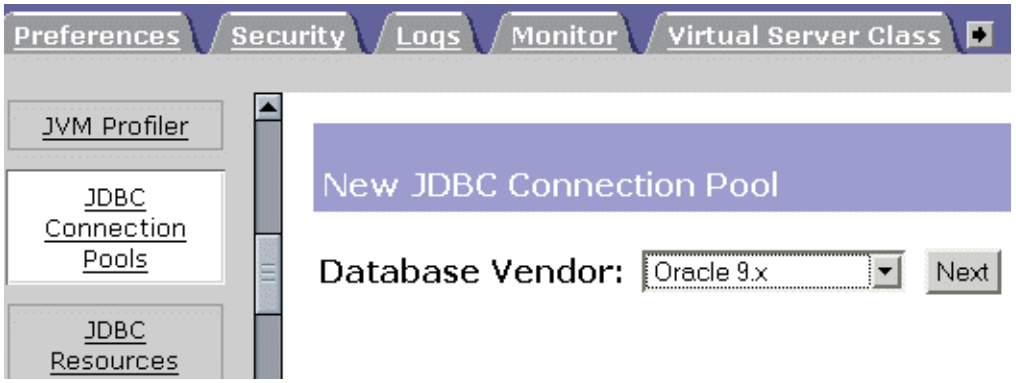
1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「JDBC Connection Pools」をクリックします。
3. 「New」をクリックします。

「JDBC Connection Pools」インタフェース



4. 「Database Vendor」ドロップダウンメニューから、接続するデータベースのタイプを選択します。適切な DBMS が表示されないときは、「Other」を選択します。

「New JDBC Connection Pool」インタフェース



5. 「Next」をクリックします。

   「Add New JDBC Connection Pool」ページが表示されます。

6. 新しい接続プールのプロパティを指定し、「OK」をクリックします。

   次に、指定が必要な接続プールのプロパティを示します。

   General

   - Pool Name : 新しい接続プールの名前を入力します。
   - DataSource Classname : データソースを実装するベンダー固有のクラス名。「New JDBC Connection Pool」ページの「Database Vendor」リストで「Other」を選択したときは、使用するデータソースのベンダーに固有のクラス名を入力する必要があります。このクラスは、`javax.sql.DataSource` を実装しなければならないことに注意してください。

   Properties

   標準の JDBC 接続プールプロパティや独自の JDBC 接続プールプロパティを指定します。これらのプロパティの多くはオプションです。デフォルトでは、すべての標準的なプロパティの名前が指定されます。データベースベンダーのドキュメントを調べ、標準プロパティとベンダー独自のプロパティのどれが必要かを決定する必要があります。

   Pool Settings

   - Steady Pool Size : プール内で必要な最小接続数を指定します。要求スレッドに対して接続が行われると、その接続はプールから削除され、現在のプールサイズが小さくなります。さらに、通常プールサイズは、サーバー起動時にプールに追加される接続の数によっても変わります。
   - Max Pool Size : プール内で常時可能な最大接続数を指定します。

- Pool Resize Quantity : プールが通常プールサイズまで縮小する場合、バッチ単位でサイズ変更されます。この値で、バッチの大きさが決まります。この値が大き過ぎると接続リサイクルが遅れ、小さ過ぎると効率が低下します。プールには、一度に 1 つの接続しか追加されないので、このフィールドに指定した値がプールの大きさに影響を与えることはありません。
- Idle Timeout (secs) : 接続がプール内でアイドル状態でいられる最大時間 (秒)。この時間が経過すると、プールの実装はこの接続を閉じることができます。
- Max Wait Time (milli secs) : 接続がタイムアウトになるまで、呼び出し側が待機する時間。デフォルトの待ち時間は `long` で、呼び出し側が長い時間待機できることを意味します。この値を `0` に設定すると、呼び出し側は接続を利用できるようになるまでブロックされます。

Connection Validation

- Connection Validation Required : このフィールドを「On」にすると、接続がアプリケーションに渡される前に検証されます。これにより、ネットワーク障害やデータベースサーバーのクラッシュのためにデータベースが利用不可能になる場合でも、Web サーバーは自動的にデータベース接続を再確立することができます。接続の検証を行うと、オーバーヘッドが増大し、パフォーマンスがわずかに低下します。
- Validation Method : Web サーバーがデータベース接続の検証に使用する方法を指定します。次の値から選択します。
  - auto-commit : このモードでは、クエリ文が個々のトランザクションとして実行およびコミットされます。`auto-commit` を無効にすると、コミットまたはロールバックメカニズムによって終了される可能性のあるトランザクションに、クエリ文がグループ化されます。
  - meta-data : このモードでは、接続データベースがテーブル、ストアドプロシージャなどを記述したメタ情報を提供することができます。メタデータオブジェクトの各インスタンスは、それに関連付けられている特定のクエリを持ちます。メタデータオブジェクトは、そのクエリを実行し、結果をキャッシュします。
  - table : この方法では、Web サーバーがユーザー指定のテーブルに対してクエリを実行する必要があります。
- Table Name : 「Validation Method」ドロップダウンリストから `table` 検証オプションを選択したときは、ここにテーブル名を指定します。
- Fail All Connections : プールの単一の接続が不良であることが確認される場合、そのプールのすべての接続を無効にして再確立するかどうかを指定します。チェックマークを付けない場合、接続は使用時にだけ個別に再確立されます。

Transaction Isolation

トランザクションが使用する遮断レベルで、アプリケーションが他のユーザーのトランザクションが行う変更からどれくらい影響を受けやすいかが決まり、その結果、これらの変更の影響を受けないようにロックする必要のある時間が決まります。

- Transaction Isolation : この接続のトランザクション遮断レベルを選択できます。次の値から選択します。
  - read-uncommitted : この遮断レベルは「ダーティリード」とも呼ばれ、データがコミットされているかどうかに関係なく、トランザクションがデータページ上の現在のデータを読み取ります。
  - read-committed : 別のトランザクションによって変更されているが、まだコミットされていないデータが読み込まれないように、データに共有ロックをかけます。コミットされていないデータは読み込まれないので、`read-committed` 遮断を実行しているトランザクションで、データを再度照会することで、そのデータが変更されるか、または元々のクエリの条件に合うようなデータが追加表示されます。
  - repeatable-read : クエリで使用されているすべてのデータにロックをかけます。ユーザーがトランザクションをコミットしていない場合、またはトランザクションをロールバックしていない場合は、現在トランザクションがアクセスしているデータを他のユーザーが変更することはできません。
  - serializable : クエリが再発行された場合に、最初と 2 番目のクエリの間で、どのデータも変更されず、データ行も追加されないように、データ範囲をロックします。
- Guarantee Isolation Level : このプールからのすべての接続の遮断レベルを同じにします。たとえば、接続の最後の使用時にプログラムによって遮断レベルが変更された場合 (たとえば、`con.setTransactionIsolation`)、この機能は接続を指定の遮断レベルに戻します。

コマンド行インタフェースを使用

コマンド行インタフェースを使用して新しい JDBC 接続プールを作成する方法については、付録 A「コマンド行ユーティリティ」の「JDBC 接続プールの作成」を参照してください。

JDBC リソースの作成

JDBC リソースはデータソースとも呼ばれ、`getConnection()` を使用してデータベースへの接続を提供します。JDBC リソースは、次の方法で作成できます。

- 管理インタフェースを使用
- コマンド行インタフェースを使用

管理インタフェースを使用

管理インタフェースを使用して JDBC リソースを作成するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「JDBC Resources」をクリックします。
3. 「New」ボタンをクリックします。
4. 次の情報を入力します。
    - JNDI Name (必須) : JDBC リソースへのアクセスにアプリケーションコンポーネントが使用しなければならない JNDI 名を入力します。
    - Pool Name (必須) : この JDBC リソースで使用される接続プールの名前 (または ID) をリストから選択します。詳細は、「JDBC 接続プールの新規作成」を参照してください。
5. JDBC リソースを有効にするには、「Data Source Enabled」ドロップダウンリストから「`On`」を選択します。

   JDBC リソースが無効な場合、どのアプリケーションコンポーネントもこのリソースに接続できませんが、設定はサーバーインスタンスで維持されます。
6. 「OK」をクリックします。
7. 「Apply Changes」をクリックします。

コマンド行インタフェースを使用

コマンド行インタフェースを使用して新しい JDBC リソースを作成する方法については、付録 A「コマンド行ユーティリティ」の「JDBC リソースの作成」を参照してください。

カスタムリソースの作成

カスタムリソースは、次の方法で作成できます。

- 管理インタフェースを使用
- コマンド行インタフェースを使用

管理インタフェースを使用

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「Custom Resources」をクリックします。
3. 「New」ボタンをクリックします。
4. 次の情報を入力します。
    - JNDI Name (必須) : カスタムリソースへのアクセスにアプリケーションコンポーネントが使用しなければならない JNDI 名を入力します。
    - Resource Type (必須) : カスタムリソースの完全修飾タイプを入力します。
    - Factory Class (必須) : ユーザーが記述するファクトリクラスの完全修飾名を入力します。このクラスは `javax.naming.spi.ObjectFactory` を実装します。
    - Custom Resource Enabled (省略可能) :「On」を選択すると、実行時にカスタムリソースが有効になります。
5. 「OK」をクリックします。
6. 「Apply Changes」をクリックします。

コマンド行インタフェースを使用

コマンド行インタフェースを使用して新しいカスタムリソースを作成する方法については、付録 A「コマンド行ユーティリティ」の「カスタムリソースの作成」を参照してください。

外部 JNDI リソースの作成

外部リソースは、次の方法で作成できます。

- 管理インタフェースを使用
- コマンド行インタフェースを使用

管理インタフェースを使用

1. サーバーマネージャにアクセスし、「Java」タブを選択します。

2. 「External JNDI Resources」をクリックします。
3. 「New」ボタンをクリックします。
4. 次の情報を入力します。
    - JNDI Name (必須) : カスタムリソースへのアクセスにアプリケーションコンポーネントが使用しなければならない JNDI 名を入力します。
    - Resource Type (必須) : カスタムリソースの完全修飾タイプを入力します。
    - Factory Class (必須) : ユーザーが記述するファクトリクラスの完全修飾名を入力します。このクラスは `javax.naming.spi.ObjectFactory` を実装します。
    - JNDI Lookup (必須) : 外部リポジトリを検索するための JNDI 値を入力します。たとえば、メールクラスをテストするために外部リポジトリに接続する外部リソースを作成する場合、JNDI 検索は `cn=testmail` を読み込みます。
    - External Resource Enabled (省略可能) : 「`On`」を選択すると、実行時に外部リソースが有効になります。
5. 「OK」をクリックします。
6. 「Apply Changes」をクリックします。

コマンド行インタフェースを使用

コマンド行インタフェースを使用して新しいカスタムリソースを作成する方法については、付録 A「コマンド行ユーティリティ」の「外部 JNDI リソース」を参照してください。

Java ベースのリソースの変更

ここでは、作成した Java ベースのリソースのプロパティを、管理インタフェースを使用して変更する方法について説明します。

- JDBC 接続プールの変更
- JDBC リソースの変更
- カスタムリソースの変更
- 外部 JNDI リソースの変更

JDBC 接続プールの変更

JDBC 接続プールのプロパティを変更するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。

2. 「JDBC Connection Pools」リンクをクリックします。
3. 編集する JDBC 接続プールのリンクをクリックします。
4. 必要に応じて設定を変更します。
5. 「OK」をクリックします。

JDBC リソースの変更

JDBC リソースのプロパティを変更するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「JDBC Resources」リンクをクリックします。
3. 編集する JDBC リソースのリンクをクリックします。
4. 必要に応じて設定を変更します。
5. 「OK」をクリックします。

カスタムリソースの変更

カスタムリソースのプロパティを変更するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「Custom Resources」リンクをクリックします。
3. 編集するカスタムリソースのリンクをクリックします。
4. 必要に応じて設定を変更します。
5. 「OK」をクリックします。

外部 JNDI リソースの変更

外部 JNDI リソースのプロパティを変更するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「External JNDI Resources」リンクをクリックします。
3. 編集する外部 JNDI リソースのリンクをクリックします。
4. 必要に応じて設定を変更します。
5. 「OK」をクリックします。

Java ベースのリソースの削除

ここでは、管理インタフェースを使用して各種 Java ベースリソースを削除する方法について説明します。

- JDBC 接続プールの削除
- JDBC リソースの削除
- カスタムリソースの削除
- 外部 JNDI リソースの削除

JDBC 接続プールの削除

JDBC リソースは、次のいずれかの方法で削除できます。

- 管理サーバーを使用
- コマンド行ユーティリティを使用

管理サーバーを使用

管理サーバーを使用して JDBC 接続プールを削除するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「JDBC Connection Pools」リンクをクリックします。
3. 削除する JDBC 接続プールのチェックボックスを選択します。
4. 「OK」をクリックします。

コマンド行ユーティリティを使用

使用するコマンド行オプションの構文については、「コマンド行ユーティリティ」を参照してください。

JDBC リソースの削除

JDBC リソースは、次のいずれかの方法で削除できます。

- 管理サーバーを使用
- コマンド行ユーティリティを使用

管理サーバーを使用

管理サーバーを使用して JDBC リソースを削除するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「JDBC Resources」リンクをクリックします。
3. 削除する JDBC リソースのチェックボックスを選択します。
4. 「OK」をクリックします。

コマンド行ユーティリティを使用

使用するコマンド行オプションの構文については、「コマンド行ユーティリティ」を参照してください。

カスタムリソースの削除

カスタムリソースは、次のいずれかの方法で削除できます。

- 管理サーバーを使用
- コマンド行ユーティリティを使用

管理サーバーを使用

管理サーバーを使用してカスタムリソースを削除するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。
2. 「Custom Resources」リンクをクリックします。
3. 削除するカスタムリソースのチェックボックスを選択します。
4. 「OK」をクリックします。

コマンド行ユーティリティを使用

使用するコマンド行オプションの構文については、「コマンド行ユーティリティ」を参照してください。

外部 JNDI リソースの削除

外部 JNDI リソースは、次のいずれかの方法で削除できます。

- 管理サーバーを使用
- コマンド行ユーティリティを使用

管理サーバーを使用

管理サーバーを使用して外部 JNDI リソースを削除するには、次の手順を実行します。

1. サーバーマネージャにアクセスし、「Java」タブを選択します。

2. 「External JNDI Resources」リンクをクリックします。
3. 削除する外部 JNDI リソースのチェックボックスを選択します。
4. 「OK」をクリックします。

コマンド行ユーティリティを使用

使用するコマンド行オプションの構文については、「コマンド行ユーティリティ」を参照してください。

第 4 部

仮想サーバーとサービスの管理

第 13 章「仮想サーバーの使用」

第 14 章「仮想サーバーの作成と設定」

第 15 章「プログラムによるサーバーの拡張」

第 16 章「コンテンツ管理」

第 17 章「設定スタイルの適用」

第 18 章「検索機能の使い方」

第 19 章「WebDAV による Web パブリッシング」

第 13 章

仮想サーバーの使用

この章では、Sun ONE Web Server を使用して仮想サーバーの設定および管理を行う方法を説明します。

この章では、次の項目について説明します。

- 仮想サーバーの概要
- 仮想サーバーで Sun ONE Web Server の機能を使用する
- 仮想サーバーのユーザーインタフェースの使用法
- 仮想サーバーの設定
- 個々の仮想サーバーをユーザーが監視できるようにする
- 仮想サーバーの配備

仮想サーバーの概要

仮想サーバーを使用すると、インストールされた 1 つのサーバーで、複数の会社または個人に対して、ドメイン名、IP アドレス、およびサーバー監視機能を提供できます。仮想サーバーを使用してハードウェアと基本的な Web サーバーの維持管理を提供しますが、ユーザーからは、それぞれが専用の Web サーバーを所有しているように見えます。

注	仮想サーバーを使用しない場合でも、Web サーバーインスタンスのコンテンツ、プログラム、およびその他の機能を設定するときはクラスマネージャの項目を使用します。Web サーバーをインストールすると、そのインスタンスのデフォルトの仮想サーバーが作成されます。仮想サーバーのユーザーインタフェースを使用して、デフォルトの仮想サーバーのコンテンツおよびサービスを管理できます。

仮想サーバーを設定するには、次の項目を設定する必要があります。

- 仮想サーバークラス
- 待機ソケット
- 仮想サーバー

仮想サーバーの設定は、`server.xml` ファイルに保存されます。このファイルは *server_root*/*server_ID*/`config` ディレクトリにあります。仮想サーバーを使用するのにこのファイルを編集する必要はありませんが、編集も可能です。このファイルと編集方法についてさらに詳細を知りたい場合は、『Sun ONE Web Server 6.1 Administrator's Configuration File Reference』を参照してください。

この節では、次の内容について説明します。

- 複数のサーバーインスタンス
- 仮想サーバークラス
- 待機ソケット
- 仮想サーバー
- 要求を処理する仮想サーバーの選択
- ドキュメントルート
- ログファイル
- 前のリリースから仮想サーバーを移行する

複数のサーバーインスタンス

これまでのリリースの Sun ONE Web Server では、仮想サーバーごとに固有の設定情報を柔軟に設定することができませんでした。サーバーごとに個別の設定情報を簡単に設定する方法として、ほとんどの場合、ユーザーは個別のサーバーインスタンスを作成していました。Sun ONE Web Server のバージョン 6.0 から、仮想サーバークラスごとに個別に情報を設定できるようになりました。複数のサーバーインスタンスを使用することは現在も可能ですが、個別の設定情報を持つサーバーを多数使用する場合は、仮想サーバーの使用をお勧めします。

仮想サーバークラス

仮想サーバーはクラスにグループ分けされます。クラスを使用すると、類似するサーバーを一度に設定できるので、各サーバーを個別に設定する必要がありません。同じクラスに含まれるすべての仮想サーバーは同じ基本設定情報を共有しますが、仮想サーバーごとに変数を設定して設定を変更することもできます。仮想サーバー間で設定情報を共有させたくない場合は、各仮想サーバークラスに 1 つずつ仮想サーバーを作成します。一方、複数の仮想サーバーが類似するプロパティを持つ場合は、1 つのクラスにグループ化して一緒に設定することができます。

たとえば、インターネットサービスプロバイダ (Internet Service Provider、ISP) で、さまざまなレベルのホスティングサービスを各顧客に異なる料金で提供する場合は、顧客に対して複数の仮想サーバークラスを設定できます。あるクラスの仮想サーバーでは Java サーブレットおよび JSP を使用可能にし、料金の低い別のクラスの仮想サーバーでは Java サーブレットおよび JSP を使用不可にすることができます。

仮想サーバーのクラスを作成するには、そのクラスに名前を付けて、ドキュメントルートを設定します。デフォルトでは、そのドキュメントルートがそのクラスに属するすべての仮想サーバーのドキュメントルートになります。`$id` 変数を使用して、クラス内の各仮想サーバーがそのクラスのドキュメントルート内に個別のドキュメントルートを持つように設定できます。詳細は、319 ページの「ドキュメントルート」を参照してください。

仮想サーバークラスを作成したあと、そのクラスにサービスを関連付けます。仮想サーバークラスでは、次の種類のサービスを有効にするか設定することができます。

- プログラム : 「プログラムによるサーバーの拡張」を参照してください。
- コンテンツ管理 : 「コンテンツ管理」を参照してください。
- 設定スタイル : 「設定スタイルの適用」を参照してください。

obj.conf ファイル

同じクラス内のすべての仮想サーバーは、1 つの `obj.conf` ファイルを共有します。このファイルには、その仮想サーバークラスに関する情報が格納されます。情報の一部は変数として格納され、各仮想サーバーの稼動時に、それぞれのサーバーに固有の変数値が代入されます。

`obj.conf` ファイルと変数についての詳細は、『NSAPI Programmer's Guide』を参照してください。ユーザーインタフェースでの変数の使用法については、324 ページの「変数の使用法」を参照してください。

クラスに属する仮想サーバー

同一のクラスに属する仮想サーバーを、そのクラスのメンバーと呼びます。仮想サーバーの設定項目には、クラス内のすべての仮想サーバーに対して設定する項目と、個別に設定する項目があります。仮想サーバーの設定項目は、クラスマネージャの「Virtual Servers」タブで設定します。詳細は、第 14 章「仮想サーバーの作成と設定」を参照してください。

デフォルトのクラス

Sun ONE Web Server のインストール時に、`defaultclass` というクラスが自動的に作成されます。このクラスには、デフォルトでそのサーバーインスタンス用の仮想サーバーメンバーが 1 つ作成されます。デフォルトのクラスにさらに仮想サーバーを追加できますが、デフォルトの仮想サーバーをクラスから削除することはできません。また、デフォルトのクラスも削除できません。

待機ソケット

サーバーとクライアントの間の接続は待機ソケットを通して確立されます。作成する各待機ソケットには、IP アドレス、ポート番号、サーバー名、デフォルト仮想サーバーが設定されます。1 台のマシンの特定のポートで設定済み IP アドレスのすべてを待機する待機ソケットを設定する場合は、IP アドレスとして 0.0.0.0、any、ANY、または INADDR_ANY を使用します。

Sun ONE Web Server をインストールすると、ls1 という待機ソケットが自動的に作成されます。この待機ソケットには、0.0.0.0 の IP アドレス と、インストール時に HTTP サーバーのポート番号として指定したポート番号 (デフォルトでは 80) が割り当てられます。デフォルトの待機ソケットは削除できません。仮想サーバーを使用しない場合は、この待機ソケットだけで十分です。仮想サーバーを使用する場合は、各仮想サーバー用に複数の待機ソケットを作成する必要がある場合があります。

待機ソケットは IP アドレスとポート番号の組み合わせであるため、複数の待機ソケットを作成する場合、IP アドレスが同じでもポート番号が異なっていればよく、また、IP アドレスが異なっていればポート番号が同じでもかまいません。たとえば、1.1.1.1:81 と 1.1.1.1:82 の待機ソケットを作成できます (1.1.1.1 はアドレス、81 と 82 はポート番号を示す)。また、1.1.1.1:81 と 1.2.3.4:81 のような待機ソケットも作成できます。

さらに、待機ソケットの受け入れスレッドの数を指定します。受け入れスレッドは、接続を待機するスレッドです。このスレッドは、接続を受け入れて、キューに入れます。キューの接続はこのあと、ワーカースレッドに引き継がれます。新しい要求が来たときに常に使用可能な受け入れスレッドが存在するように十分な受け入れスレッド

数を設定するのが理想的ですが、システムに負担がかかりすぎない程度の数に抑える必要があります。デフォルトは 1 です。システムの CPU 1 つ当たり 1 つの受け入れスレッドを設定することをお勧めします。パフォーマンスに問題がある場合は、この値を調節できます。

仮想サーバー

仮想サーバーを作成するには、まず、その仮想サーバーをどのクラスに入れるかを決める必要があります。次に、作成する仮想サーバーの種類を決める必要があります。仮想サーバーを作成するには、仮想サーバー ID、および 1 つ以上の URL ホストを指定する必要があります。

この節では、次の内容について説明します。

- 仮想サーバーの種類
- IP アドレスベースの仮想サーバー
- URL ホストベースの仮想サーバー
- デフォルトの仮想サーバー

仮想サーバーの種類

Sun ONE Web Server のリリース 6.0 の前までは、ハードウェアとソフトウェアの 2 種類の仮想サーバーがありました。ハードウェア仮想サーバーには、固有の IP アドレスが割り当てられていました。ソフトウェア仮想サーバーには、固有の IP アドレスはなく、代わりに固有の URL ホストが割り当てられていました。

Sun ONE Web Server 6.0 および Sun ONE Web Server 6.1 では、この概念は正確でなくなりました。すべての仮想サーバーに URL ホストが割り当てられます。ただし、待機ソケット基づいて、仮想サーバーに IP アドレスが割り当てられる場合もあります。

新しい要求を受け取ると、サーバーは IP アドレスまたは Host ヘッダーの値に基づいて、受け取った要求をどの仮想サーバーに送るかを決定します。サーバーは、最初に IP アドレスを評価します。詳細は 319 ページの「要求を処理する仮想サーバーの選択」を参照してください。

IP アドレスベースの仮想サーバー

1 つのコンピュータに複数の IP アドレスを設定するためには、オペレーティングシステムでマッピングするか、カードを追加する必要があります。オペレーティングシステムで複数の IP アドレスを設定するには、Windows の場合はコントロールパネルの「ネットワーク」、UNIX または Linux の場合は `ifconfig` ユーティリティを使用します。`ifconfig` を使用する方法は、プラットフォームによって異なります。詳細は、オペレーティングシステムのマニュアルを参照してください。

IP アドレスベースの仮想サーバーを作成するときは、通常は特定の IP アドレスで待機する待機ソケットを作成します。待機ソケットのデフォルト仮想サーバーは、IP アドレスベースの仮想サーバーです。仮想サーバーを配備する方法については、332 ページの「仮想サーバーの配備」を参照してください。

URL ホストベースの仮想サーバー

URL ホストベースの仮想サーバーを設定するには、各仮想サーバーに固有の URL ホストを割り当てます。サーバーは、Host 要求ヘッダーの内容によって、その要求を正しい仮想サーバーに振り向けます。

たとえば、aaa、bbb、および ccc という顧客の仮想サーバーを設定し、それぞれの顧客が個別のドメイン名を持てるようにするには、まず、各顧客の URL (`www.aaa.com`、`www.bbb.com`、`www.ccc.com`) を、使用する待機ソケットの IP アドレスへ名前解決して認識できるように DNS を設定します。次に、各仮想サーバーの URL ホストを正しく設定します (`www.aaa.com` など)。

URL ホストベースの仮想サーバーは、Host 要求ヘッダーを使用してユーザーに正しいページを表示するため、クライアントソフトウェアによっては、この処理ができない場合もあります。HTTP Host ヘッダーをサポートしていないクライアントソフトウェアでは、この処理ができません。そのようなクライアントは、待機ソケットのデフォルトの仮想サーバーを受け取ります。

デフォルトの仮想サーバー

URL ホストベースの仮想サーバーは、Host 要求ヘッダーを使用して選択されます。エンドユーザーのブラウザが Host ヘッダーを送信しない場合、または、指定された Host ヘッダーをサーバーが見つけられない場合は、デフォルトの仮想サーバーがその要求を処理します。

デフォルトの仮想サーバーは、待機ソケットによって設定されます。待機ソケットの作成時に、デフォルトの仮想サーバーを指定します。デフォルトの仮想サーバーはいつでも変更できます。

要求を処理する仮想サーバーの選択

サーバーで要求を処理するには、待機ソケットを介して要求を受け入れてから、適切な仮想サーバーにその要求を送信する必要があります。

その後、仮想サーバーは次のように選択されます。

- 待機ソケットがデフォルトの仮想サーバーのみに指定されている場合は、その仮想サーバーが選択されます。
- 待機ソケットが複数の仮想サーバーで構成されている場合は、要求の `Host` ヘッダーと一致する仮想サーバーの URL ホストが選択されます。`Host` ヘッダーが存在しない場合、または一致する URL ホストがない場合は、その接続グループのデフォルトの仮想サーバーが選択されます。

SSL 待機ソケットに設定されている仮想サーバーは、サーバーの起動時に、その URL ホストが証明書のサブジェクトパターンと照合され、一致しない場合は、警告が生成されてエラーログに書き込まれます。

仮想サーバーが決定されると、サーバーは、その仮想サーバーが属する仮想サーバークラスの `obj.conf` ファイルを実行します。サーバーが `obj.conf` で実行する指令を決定する方法については、『Sun ONE Web Server 6.1 NSAPI Programmer's Guide』を参照してください。

ドキュメントルート

プライマリドキュメントディレクトリまたはドキュメントルートは、仮想サーバーの全ファイルを格納してリモートクライアントに提供するための中央ディレクトリです。

ドキュメントルートディレクトリを利用すると、仮想サーバー上のファイルへのアクセスを簡単に制限できます。また、URL に指定されたパスはプライマリドキュメントディレクトリへの相対パスであるため、URL を変更せずに、ドキュメントを新しいディレクトリ (別のディスクの場合もある) に簡単に移動できます。

たとえば、ドキュメントディレクトリが `C:¥sun¥servers¥docs` の場合、`http://www.sun.com/products/info.html` などの要求を受け取ると、サーバーは `C:¥sun¥servers¥docs¥products¥info.html` でそのファイルを探します。ドキュメントルートを変更する (つまり、すべてのファイルおよびサブディレクトリを移動する) 場合は、仮想サーバーが使用するドキュメントルートを変更するだけなので、すべての URL を新しいディレクトリにマッピングしたり、クライアントに新しいディレクトリを探すように知らせる必要はありません。

Sun ONE Web Server のインストール時に、Web サーバーインスタンスのドキュメントルートを指定します。これが、デフォルトのクラスのドキュメントルートになります。クラスレベルで、ドキュメントルート用のディレクトリを変更できます。また、個々の仮想サーバーレベルでクラスレベルのディレクトリを無効にすることもできます。

クラスを追加する場合、ドキュメントディレクトリも指定する必要があります。このディレクトリは絶対パスで指定します。ただし、絶対パスをそのままで入力すると、そのクラスに属するすべての仮想サーバーのドキュメントルートが同じディレクトリにデフォルト設定されます。ドキュメントルートの絶対パスの最後に $id 変数を付けると、仮想サーバーごとに、*class_doc_root*/*virtual_server_ID* というドキュメントルートがデフォルト設定されます。たとえば、クラスのドキュメントディレクトリが `/sun/servers/docs/$id` の場合、そのクラスに属する仮想サーバー `vs1` のデフォルトのドキュメントディレクトリは、`/sun/servers/docs/vs1` になります。

変数の詳細については、324 ページの「変数の使用法」を参照してください。

クラスレベルのデフォルトのドキュメントディレクトリを、個別の仮想サーバーレベルでは無効にすることもできます。

ログファイル

新しい仮想サーバーを作成すると、デフォルトでは、ログファイルはサーバーインスタンスと同じログファイルが使用されます。ほとんどの場合、仮想サーバーごとに専用のログファイルを使用する必要があります。そのためには、各仮想サーバーのログパスを変更します。

詳細は、343 ページの「仮想サーバーのログの設定」を参照してください。

前のリリースから仮想サーバーを移行する

バージョン 4.1 の iPlanet Web Server を使用していた場合は、移行ツールを使用して現在のバージョンに移行できます。詳細は、『インストールおよび移行ガイド』を参照してください。

仮想サーバーで Sun ONE Web Server の機能を使用する

Sun ONE Web Server には、SSL やアクセス制御など、仮想サーバーで使用できる多くの機能があります。これらの機能の多くは、すべてのサーバー、1 つのサーバーインスタンス、仮想サーバークラス、または個別の仮想サーバーの設定に関係します。次の節では、これらの機能について説明し、詳細情報の参照先に関する情報を提供します。

この節では、次の内容について説明します。

- 仮想サーバーで SSL を使用する
- 仮想サーバーでアクセス制御を使用する
- 仮想サーバーで CGI を使用する
- 仮想サーバーで設定スタイルを使用する

仮想サーバーで SSL を使用する

仮想サーバーで SSL を使用するときは、ほとんどの場合、IP アドレスベースの仮想サーバーを使用します。ポートは通常、443 を使用します。Sun ONE Web Server は、要求を送信する URL ホストを決定する前に、その要求を読み取る必要があるため、URL ホストベースの仮想サーバーで SSL を使用するのは困難です。サーバーが要求を読み取った時点で、セキュリティ情報をやり取りする最初のハンドシェークが発生してしまうからです。

唯一の例外は、URL ホストベースの仮想サーバーのすべてが、「ワイルドカード証明書」を使用して、同一のサーバー証明書など、同じ SSL 設定を持つ場合です。詳細は、第 6 章「証明書と鍵の使用」を参照してください。

仮想サーバーで SSL を使用する方法の 1 つは、2 つの待機ソケットを設定して、一方は SSL を使用してポート 443 で待機し、もう一方は SSL を使用しないようにすることです。ユーザーは通常、SSL を使用しない待機ソケットから仮想サーバーにアクセスします。セキュリティ保護されたトランザクションの必要が生じた場合には、ユーザーは、Web ページ上のボタンをクリックして、セキュリティ保護されたトランザクションを起動します。この操作のあと、セキュリティ保護された待機ソケットが要求に使用されます。

SSL トランザクションは、SSL を使用しないトランザクションよりもかなり時間がかかるため、SSL トランザクションの使用は必要な場合のみに限定されます。必要な場合以外は、より高速な SSL を使用しない接続を使用します。

Sun ONE Web Server や仮想サーバーでセキュリティを設定する方法および使用する方法についての詳細は、第 6 章「証明書と鍵の使用」を参照してください。仮想サーバーでの SSL の設定例は、334 ページの「例 2 : セキュリティ保護されたサーバー」の図を参照してください。

仮想サーバーでアクセス制御を使用する

仮想サーバーでは、仮想サーバー単位でアクセス制御を設定できます。さらに、LDAP データベースを使用して、各仮想サーバーごとにユーザーおよびグループを認証できるように設定することもできます。詳細は、227 ページの「仮想サーバーへのアクセス制御」を参照してください。

仮想サーバーで CGI を使用する

仮想サーバーで CGI を使用できます。アクセスおよびセキュリティに関して設定できる項目が多数あります。

CGI の設定および使用法の詳細は、359 ページの「CGI プログラムのインストール」を参照してください。

仮想サーバーで設定スタイルを使用する

設定スタイルは、さまざまな仮想サーバーが維持管理する特定のファイルやディレクトリに対して、一連のオプションを簡単に適用する方法です。設定スタイルの詳細は、「設定スタイルの適用」を参照してください。

仮想サーバーのユーザーインタフェースの使用法

仮想サーバーの作成および編集には、ユーザーインタフェースまたはコマンド行ユーティリティを使用できます。

仮想サーバーを管理するためのユーザーインタフェースは、次の 3 つの部分で構成されます。

- サーバーマネージャでは、サーバー全体 (またはすべての仮想サーバー) に影響する項目を設定します。
- クラスマネージャでは、単一のクラスおよびクラス内の仮想サーバーに影響する項目を設定します。
- 仮想サーバーマネージャでは、個々の仮想サーバーに関する項目を設定します。

また、個々の仮想サーバーを所有するエンドユーザー用のユーザーインタフェースも使用できます。詳細は、328 ページの「個々の仮想サーバーをユーザーが監視できるようにする」を参照してください。

この節では、次の内容について説明します。

- クラスマネージャ
- 仮想サーバーマネージャ
- 変数の使用法
- ダイナミック再設定

クラスマネージャ

クラスマネージャにアクセスするには、次の手順に従います。

1. サーバーマネージャから、「Virtual Server Class」タブをクリックします。
2. 「Manage Classes」をクリックします。
3. クラスを選択して、「Manage」をクリックします。

サーバーのツリービューでクラス名をクリックする、またはサーバーマネージャの右上隅にある「Class Manager」ボタンをクリックする方法もあります。

仮想サーバーマネージャ

仮想サーバーマネージャにアクセスするには、次の手順に従います。

1. クラスマネージャから「Virtual Server」タブをクリックします。
2. 「Manage Virtual Servers」をクリックします。
3. 仮想サーバーを選択して、「Manage」をクリックします。

サーバーのツリービューで仮想サーバー名をクリックする方法もあります。

コマンド行ユーティリティの `HttpServerAdmin` を使用して、ユーザーインタフェースを使用する場合と同じ仮想サーバー関連操作を実行できます。コマンド行ユーティリティ `HttpServerAdmin` の詳細は、447 ページの「HttpServerAdmin (仮想サーバーの管理)」を参照してください。

変数の使用法

クラス内の仮想サーバーごとに共通の値を設定する変数を使用すると、それぞれの値を個別に指定する必要がありません。変数は、`obj.conf` ファイルに定義されます。独自の変数を定義できますが、ユーザーインタフェースでは独自に定義した変数は認識されません。ユーザーインタフェースでもっとも便利な変数は、仮想サーバーの ID を表す $id 変数です。この変数を入力すると、サーバーは、その値に各仮想サーバーの ID を代入します。

$accesslog (各仮想サーバーのアクセスログのパス) や $docroot (各仮想サーバーのドキュメントルートのパス) など、いくつかの変数がありますが、フィールドに入力する必要があるのは $id だけです。

変数の詳細については、『Sun ONE Web Server 6.1 NSAPI Programmer's Guide』を参照してください。

ダイナミック再設定

ダイナミック再設定を利用すると、稼動中の Web サーバーの設定を変更して、Web サーバーを停止したり再起動したりすることなく、変更を適用することができます。`server.xml` およびその関連ファイル内の設定に関する設定および属性のすべてを、サーバーを再起動することなくダイナミックに変更できます。仮想サーバーのユーザーインタフェースで加えたすべての変更が、サーバーを再起動することなく適用されます。変更後に、再設定スクリプトまたはユーザーインタフェースを使用して、サーバーをダイナミックに再設定することができます。

UNIX プラットフォームでは、ダイナミック再設定のスクリプトは、各インスタンスのディレクトリにある「reconfig」というシェルスクリプトです。このスクリプトにはコマンド行引数はありません。サーバーインスタンスのディレクトリで「reconfig」と入力するだけで、この再設定スクリプトを実行できます。

Windows では、ダイナミック再設定のスクリプトは、各インスタンスのディレクトリにある「reconfig.bat」というバッチファイルです。コマンド行引数はありません。サーバーインスタンスのディレクトリで「reconfig」または「reconfig.bat」と入力するだけで、この再設定スクリプトを実行できます。

このスクリプトを実行すると、ユーザーインタフェースと同様にサーバーのダイナミックな再設定が開始され、再設定に関連するサーバーメッセージが表示されます。

ダイナミック再設定の画面にアクセスするには、「Server Manager」、「Class Manager」、および「Virtual Server Manager」の各ページの右上隅にある「Apply」リンクをクリックし、次に、「Apply Changes」ページの「Load Configuration Files」ボタンをクリックします。新しい設定情報のインストールの際にエラーが発生した場合は、それまでの設定情報が復元されます。

仮想サーバーの設定

仮想サーバーを設定するには、次の手順に従います。

1. 待機ソケットを作成します。
2. 仮想サーバークラスを作成します。
3. クラスのサービスを設定します。
4. 仮想サーバークラス内の仮想サーバーを作成します。
5. 仮想サーバーを設定します。

待機ソケットを作成する場合、デフォルトの仮想サーバーのフィールドには、既存の仮想サーバーを入力する必要があります。サーバーのインストール時に作成された仮想サーバーをデフォルトの仮想サーバーとして使用し、追加の仮想サーバーを作成したあとで、必要に応じて、デフォルトの仮想サーバーを変更することができます。

待機ソケットの作成

待機ソケットを作成するには、次の手順に従います。

1. サーバーマネージャから、「Preferences」タブをクリックします。
2. 「Add Listen Socket」をクリックします。
3. 各フィールドに必要事項を入力します。

   待機ソケットのポート番号と IP アドレスは、他と重複しない組み合わせにする必要があります。IPV4 または IPV6 のアドレスを使用できます。IP アドレスベースの仮想サーバーの待機ソケットを作成する場合、IP アドレスは、0.0.0.0、ANY、any、または INADDR_ANY にする必要があります。これにより、この待機ソケットは、そのポートですべての IP アドレスを待機します。

   この待機ソケットでは、セキュリティ機能 (SSL) を使用可能にすることもできます。

   「Server Name」フィールドには、サーバーがクライアントに送る URL のホスト名を指定します。これは、サーバーが自動生成する URL には影響しますが、サーバーに格納されているディレクトリおよびファイルの URL には影響しません。サーバーがエイリアスを使用する場合は、この名前もエイリアス名にする必要があります。
4. 「OK」をクリックします。

仮想サーバークラスの作成

仮想サーバークラスを作成するには、次の手順に従います。

1. サーバーマネージャから、「Virtual Server Class」タブをクリックします。
2. 「Add Class」をクリックします。
3. クラスに名前を付けます。
4. そのクラスのドキュメントルートを入力します。

   これは、既存のディレクトリである必要があります。このクラスのすべての仮想サーバーでは、特に指定しない限り、この絶対パスのドキュメントルートが使用されます。パスの最後に `/$id` を付けると、そのクラスのドキュメントルートパス内に、その仮想サーバー ID の名前が付けられたドキュメントルートフォルダが自動的に作成されます。
5. 「OK」をクリックします。

仮想サーバーのクラスを作成したら、そのクラスに関連付けるサービスを選択します。詳細は、「コンテンツ管理」を参照してください。

仮想サーバークラスの編集または削除

仮想サーバークラスの設定を編集するには、次の手順に従います。

1. サーバーマネージャから、「Virtual Server Class」タブをクリックします。
2. 「Edit Classes」をクリックします。
3. 目的のクラスの横にあるプルダウンリストから、「Edit」または「Delete」を選択します。

   デフォルトのクラスは削除できません。
4. クラスのデフォルトのドキュメントルートの絶対パスを変更するには、「Document Root」フィールドを使用します。

   このクラスの仮想サーバーのドキュメントルートは、デフォルトでは、このディレクトリ内に作成されます。
5. 仮想サーバーのこのクラスが、Accept Language ヘッダーの解析を使用するように設定するときは、「Accept Language」フィールドに「On」を指定します。

   デフォルトは、「Off」です。
6. クラスに関連付けられている CGI のデフォルト設定を変更する場合は、「Advanced」をクリックします。

   CGI のデフォルト設定を示すウィンドウが表示されます。必要なフィールドを編集し、「OK」をクリックして「Edit Classes」ウィンドウに戻ります。「Reset」ボタンをクリックすると、変更が元に戻されます。
7. 「OK」をクリックします。これで、クラスが変更または削除されます。

仮想サーバークラスと関連付けるサービスの指定

仮想サーバークラス間の違いを示す特性に、それぞれの仮想サーバークラスで使用できるサービスの違いがあります。たとえば、CGI を使用できる仮想サーバークラスと、使用できない仮想サーバークラスを設定できます。サービスの設定方法の詳細は、「コンテンツ管理」を参照してください。

仮想サーバーの作成

仮想サーバークラスの設定が完了したら、仮想サーバーを作成できます。仮想サーバーは、仮想サーバークラスのメンバーであるため、クラスマネージャで作成します。

詳細は、339 ページの「仮想サーバーの作成」を参照してください。

仮想サーバーと関連付ける設定の指定

クラスの設定を仮想サーバーレベルでは無効にできます。また、設定を追加することもできます。これらの設定は、クラスマネージャで行います。

詳細は、「仮想サーバーの作成」を参照してください。

個々の仮想サーバーをユーザーが監視できるようにする

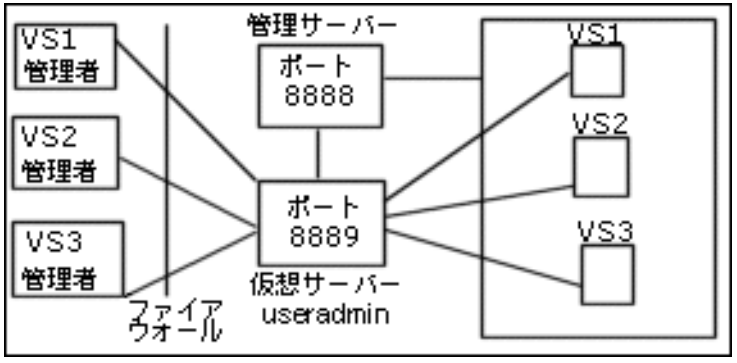
個々の仮想サーバーを管理するための特別なユーザーインタフェースがあります。これを利用すると、各仮想サーバーの管理者は、その仮想サーバーの設定を確認したり、アクセスログやエラーログを表示することができます。たとえば、3 つの部門用に 3 つの仮想サーバーを持つイントラネットの場合、それぞれの部門で、設定およびログファイルを個別に表示できます。

セキュリティ上の理由により、この管理ユーザーインタフェースは、管理サーバーポートまたは Web サーバーインスタンスポートとは別のポートにあります。

このユーザーインタフェースは、管理サーバー内の仮想サーバーで稼動します。この仮想サーバーは、デフォルトで設定され、useradmin という名前が付けられます。ユーザーが管理サーバーポートへのアクセス権を持たなくても仮想サーバーの管理ユーザーインタフェースにアクセスできるように、管理サーバーが稼動する待機ソケットとは別の待機ソケットを仮想サーバーに設定する必要があります。

次の図は、各仮想サーバーの管理者が、useradmin 仮想サーバーから各自の仮想サーバーの情報にアクセスする様子を示しています。

仮想サーバー管理者のユーザーインタフェースの設定



仮想サーバーを起動するときに、管理サーバーの `/config/server.xml` ファイルで特定の設定を編集した場合は、ユーザーは次の URL から管理することができます。

server_name:*port*/user-app/*server_instance*/*virtual_server_ID*

その例を次に示します。

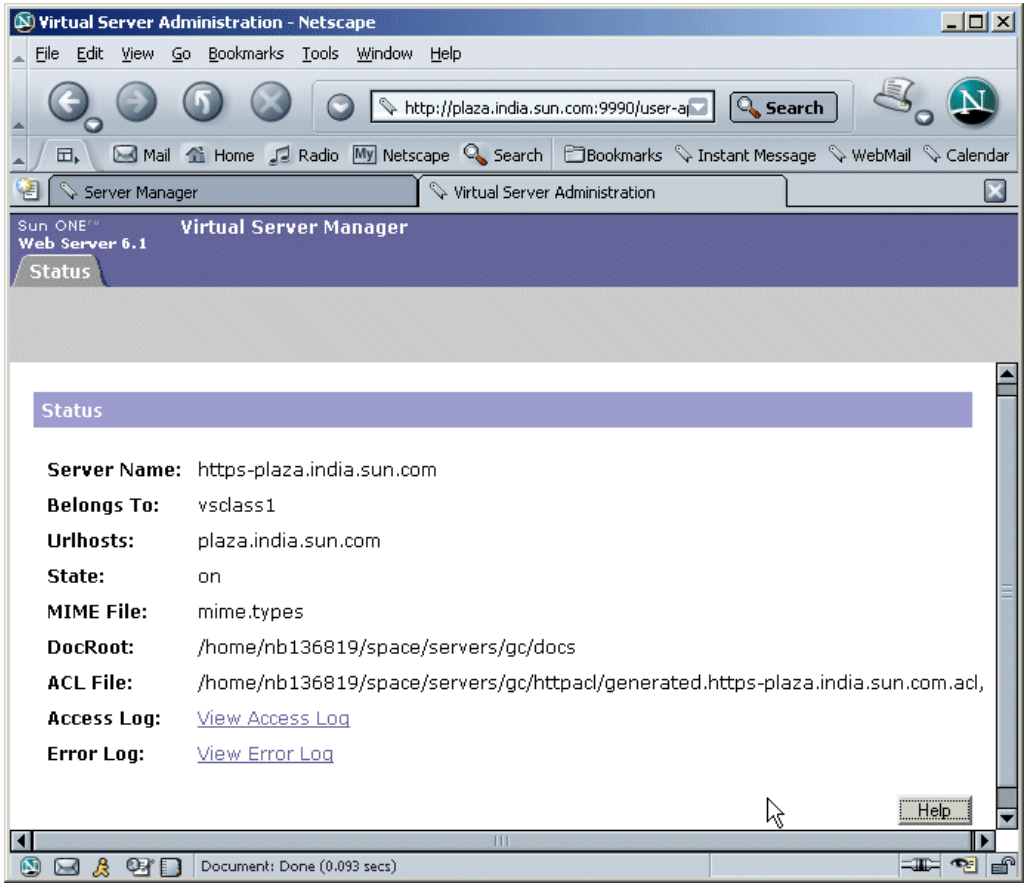
sun:9999/user-app/sun/vs2

サーバーインスタンスには、サーバーインスタンス名の「https」の部分は指定しません。

仮想サーバーの ID を特定するには、サーバーインスタンスの server.xml ファイルを調べます。

次の図は、エンドユーザーに表示されるユーザーインタフェースを示します。

仮想サーバーの管理ユーザーインタフェース



Sun ONE Web Server 6.1 をインストールすると、次の項目を作成するための特定の行が *server_root*/https-admserv/config/server.xml ファイル内でコメントになっています。

- useradmin という仮想サーバーのデフォルト待機ソケット
- 仮想サーバーの仮想サーバークラス

これらの項目のコメントアウトを解消するだけで、useradmin を設定できます。

この機能を使用できるようにサーバーを設定するには、次の手順に従います。

1. 管理サーバーが使用するポートとは別のポートを使用する新しい待機ソケットを作成します。

   たとえば、管理サーバーがポート 8888 で稼動する場合、新しく作成する待機ソケットには、別のポート番号を指定する必要があります。別の待機ソケットを使用することで、管理サーバーを保護できます。

   セキュリティ上の理由により、ユーザーインタフェースからこの待機ソケットを追加することはできません。その代わりに、管理サーバーの `server.xml` ファイルに待機ソケットを追加します。

2. 管理サーバーの `server.xml` ファイルを開きます。このファイルは、*server_root*`/https-admserv/config/server.xml` にあります。

3. `LS`、`VSCLASS`、`VS` 要素のデフォルト値を含むコメント行で、コメントアウトを解除します。その例を次に示します。

   ```
   <!--
   <LS id="ls2" port="9999" servername="plaza"
   defaultvs="useradmin"/>
   -->
   <!--
   <VSCLASS id="userclass" objectfile="userclass.obj.conf">
      <VS id="useradmin" connections="ls2" mime="mime1"
   aclids="acl1"   urlhosts="plaza">
         <PROPERTY name="docroot" value="/export1/wsinst/docs"/>
         <USERDB id="default"/>
         <WEBAPP uri="/user-app"
   path="/export1/wsinst/bin/https/webapps/user-app"/>
      </VS>
   </VSCLASS>
   -->
   ```

   これにより、セキュリティ上の理由から独立したポートに作成されている useradmin が有効になります。

4. 変更を `server.xml` に保存します。

5. 管理サーバーを再起動して変更を適用します。

6. これで、どのサーバーインスタンスのどの仮想サーバーでも、次の URL で管理ユーザーインタフェースにアクセスできます。

   server_name:port/user-app/*server_instance*/*virtual_server_ID*

   その例を次に示します。

   ```
   plaza:9999/user-app/plaza/https-plaza
   ```

アクセス制御

権限を持たないユーザーによる仮想サーバーの管理操作を禁止するために、ACL を設定できます。仮想サーバーはそれぞれ固有の URI を持つので、正当な管理者のみが仮想サーバーの設定にアクセスできるようにアクセス権を設定できます。

詳細は、第 9 章「サーバーへのアクセス制御」を参照してください。

ログファイル

仮想サーバーごとに専用のログファイルを設定できます。デフォルトでは、すべての仮想サーバーがサーバーインスタンスのログファイルを共有します。ユーザーが各自のログファイルを表示できるようにするには、ほとんどの場合、各仮想サーバーが専用のアクセスログおよびエラーログを使用するようにログファイルの設定を変更する必要があります。

詳細は、343 ページの「仮想サーバーのログの設定」を参照してください。

仮想サーバーの配備

Sun ONE Web Server の仮想サーバーアーキテクチャは、柔軟性に富んでいます。サーバーインスタンスには、セキュリティ保護されたものとそうでないものを含めて、任意の数の待機ソケットを作成できます。また、IP アドレスベースと URL ホストベースの両方の仮想サーバーを構成できます。

さらに、構成が類似する仮想サーバーを、任意の数の仮想サーバークラスにグループ分けすることもできます。仮想サーバークラスに属するすべての仮想サーバーは、`obj.conf` 内の同じ要求処理命令を共有します。

仮想サーバーごとに、専用の ACL、専用の `mime.types` ファイル、および専用の Java Web アプリケーションセットを設定することもできます (設定しなくてもかまいません)。

このように設計されているため、さまざまな用途に合わせてサーバーを柔軟に構成できます。次の例では、Sun ONE Web Server で利用可能な構成をいくつか説明します。

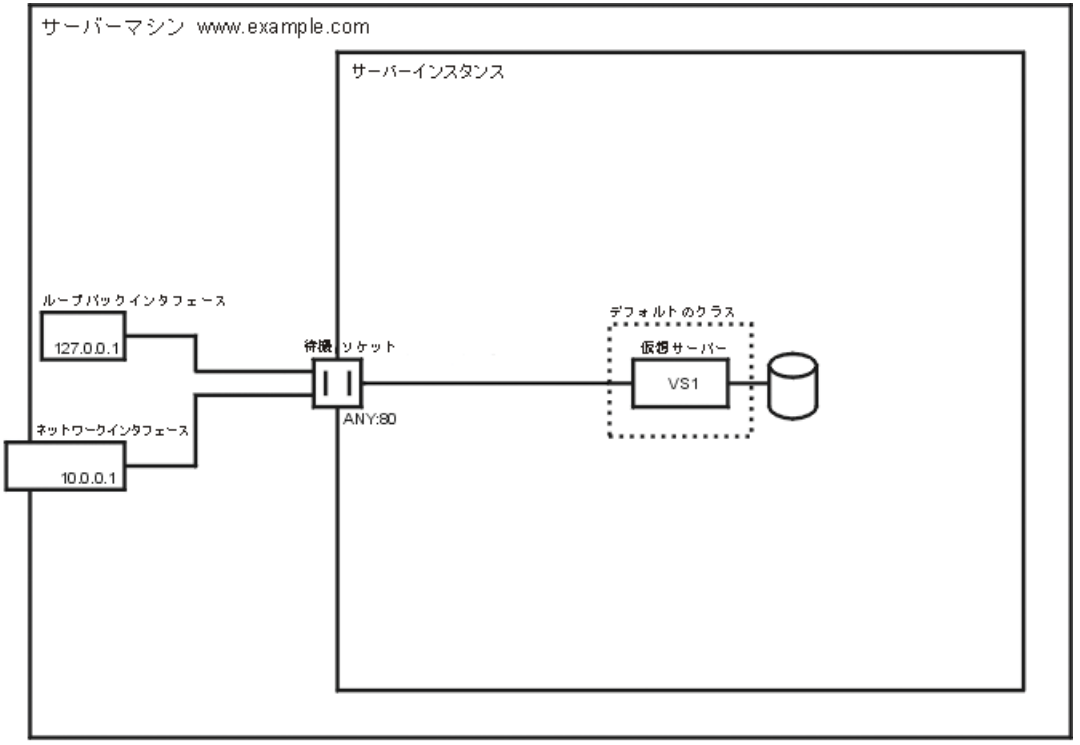
例 1 : デフォルトの構成

Sun ONE Web Server を新規にインストールすると、1 つのサーバーインスタンスができます。このサーバーインスタンスは、コンピュータに設定されているすべての IP アドレスのポート 80 (またはインストール時に選択したポート) で待機する待機ソケットを 1 つだけ持ちます。

ローカルネットワークのメカニズムによっては、コンピュータに設定されているアドレスごとに名前とアドレスのマッピングを確立する場合があります。次の例では、コンピュータに 2 つのネットワークインタフェースがあります。1 つはアドレス 127.0.0.1 のループバックインタフェース (ネットワークカードがなくても存在するインタフェース)、もう 1 つはアドレス 10.0.0.1 のイーサネットインタフェースです。

example.com という名前が、DNS により 10.0.0.1 にマッピングされます。待機ソケットは、そのマシンに設定されているすべてのアドレスのポート 80 で待機するように設定されます (「ANY:80」または「0.0.0.0:80」)。

デフォルトの構成



この構成では、次の場所への接続がサーバーに到達し、仮想サーバー VS1 によって処理されます。

- `http://127.0.0.1/` (`example.com` 上で開始される)
- `http://localhost/` (`example.com` 上で開始される)
- `http://example.com/`
- `http://10.0.0.1/`

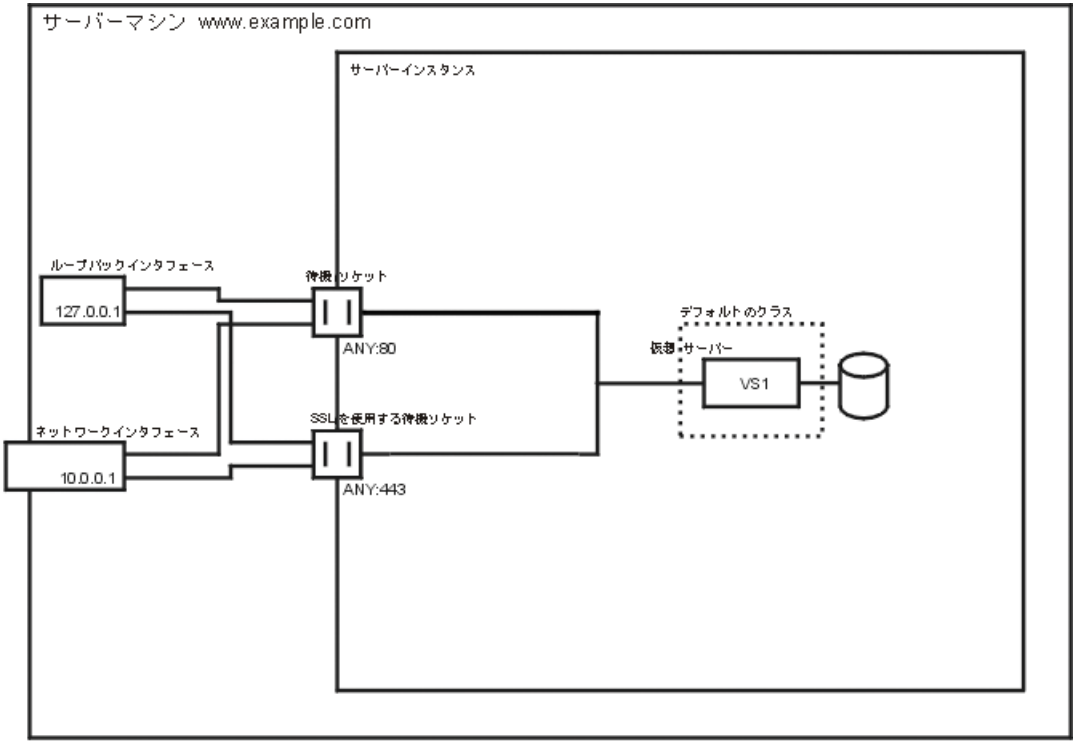
通常の Web サーバーの使用法では、この構成を使用します。仮想サーバーや待機ソケットをさらに追加する必要はありません。サーバーの設定を行うには、defaultclass (VS1 は defaultclass のメンバー) および VS1 そのもので設定を変更します。

例 2 : セキュリティ保護されたサーバー

デフォルトの構成で SSL を使用する場合は、単に待機ソケットをセキュリティモードに変更するだけです。これは、以前のバージョンの Sun ONE Web Server でセキュリティを設定する方法と同様です。

また、セキュリティ保護された待機ソケット (ANY:443 に設定) を新しく追加して、その新しい待機ソケットに VS1 を関連付けることもできます。この仮想サーバーは SSL を使用する待機ソケットと、SSL を使用しない待機ソケットを持ちます。これにより、サーバーは SSL を使用する場合と使用しない場合に同じコンテンツを提供します。つまり、`http://example.com/` と `https://example.com/` は同じコンテンツを提供します。

セキュリティ保護されたサーバー



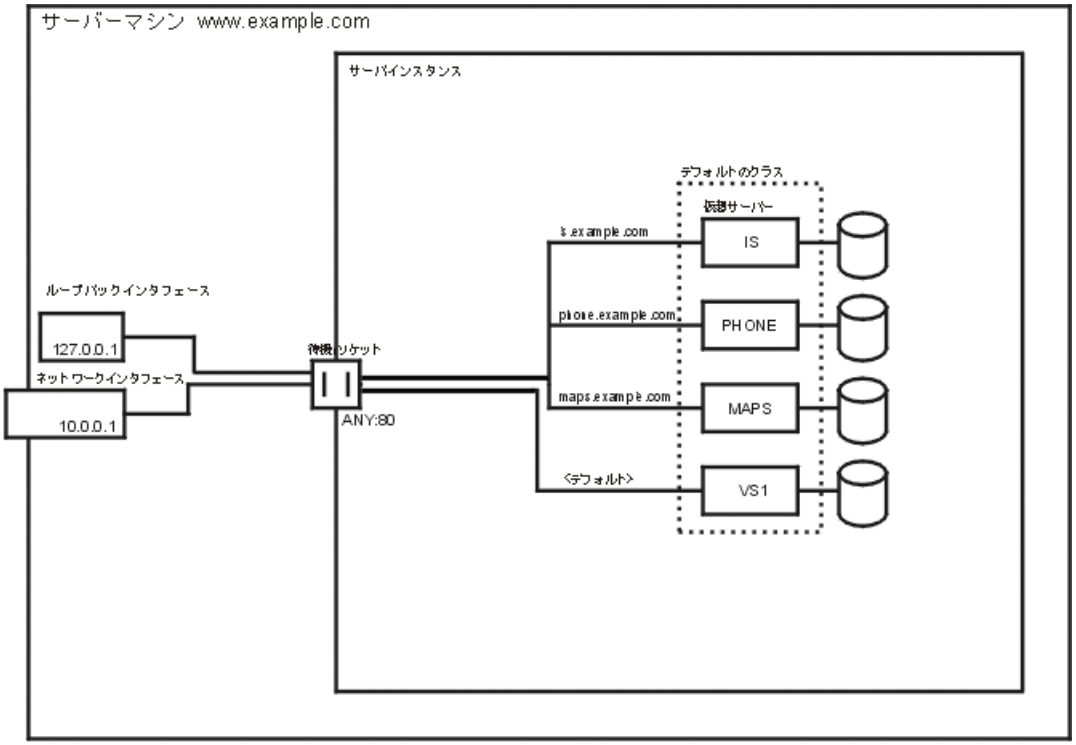
SSL パラメータは待機ソケットに対して設定します。つまり、特定の待機ソケットに設定されているすべての仮想サーバーに対して 1 つの SSL パラメータセットを設定します。

例 3 : イントラネットホスティング

さらに複雑な Sun ONE Web Server の構成として、イントラネットで、サーバーが複数の仮想サーバーをホスト処理する構成があります。たとえば、3 つの内部サイトがあり、従業員は、1 つ目のサイトで他のユーザーの電話番号を検索でき、2 つ目のサイトで構内の地図を参照でき、3 つ目のサイトでは情報サービス部門に出した要求の状態を追跡できるとします。この例では、以前は、これらのサイトが 3 つのコンピュータでホスト処理され、それぞれのコンピュータに `phone.example.com`、`maps.example.com`、および `is.example.com` という名前が割り当てられていました。

ハードウェアと管理のオーバーヘッドを最小化するために、3 つすべてのサイトを `example.com` のマシン上の 1 つの Web サーバーに統合します。これは、URL ホストベースの仮想サーバー、または独立した待機ソケットを使用する 2 つの方法で設定できます。両者には、それぞれに長所と短所があります。

URL ホストベースの仮想サーバーによるイントラネットホスティング

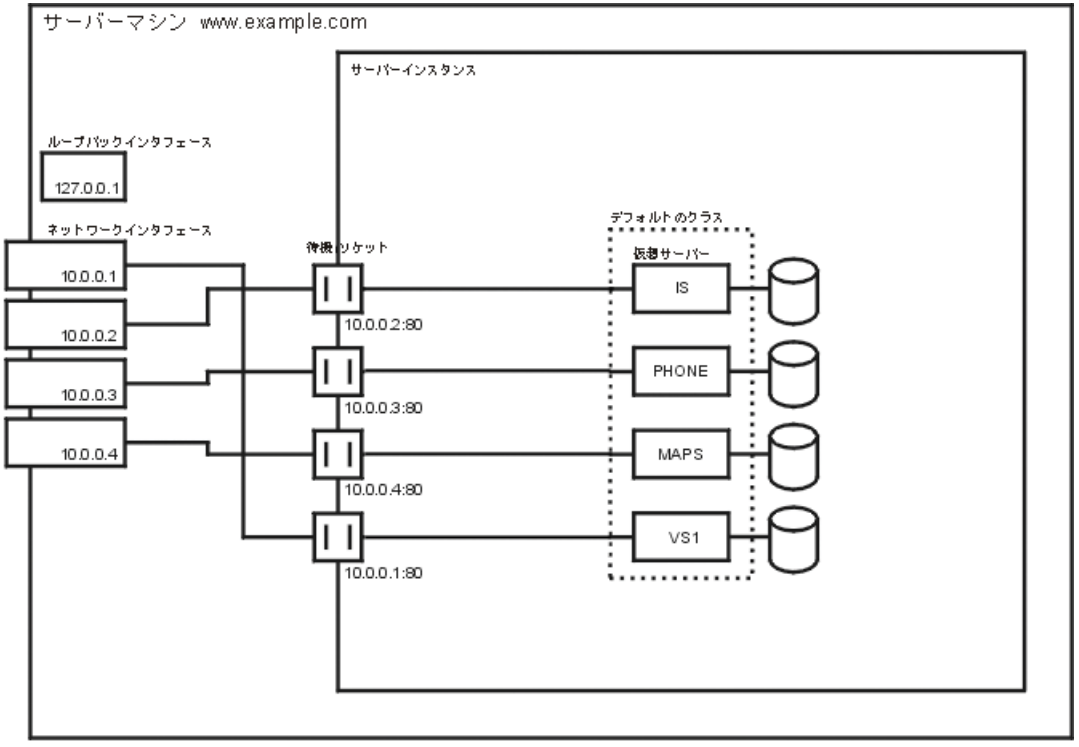


URL ホストベースの仮想サーバーは設定が簡単ですが、次のような制限があります。

- この構成で SSL を使用するには、ワイルドカード証明書による標準外の設定が必要です。詳細は、第 4 章「Web コンテナと Web アプリケーションの J2EE ベースのセキュリティ」を参照してください。
- URL ホストベースの仮想サーバーは、古いバージョンの HTTP クライアントでは動作しません。

アドレスごとに 1 つの待機ソケットを持つ IP アドレスベースの構成も可能です。

個別の待機ソケットを使用するイントラネットホスティング



IP アドレスベースの仮想サーバーによるイントラネットホスティングには、次のような長所があります。

- HTTP/1.1 Host ヘッダーをサポートしていない古いバージョンのクライアントでも動作します。
- SSL サポートを簡単に提供できます。

一方で、次のような短所があります。

- ホストコンピュータの設定 (実際のネットワークインタフェースまたは仮想ネットワークインタフェースの設定) を変更する必要があります。

- 何千もの仮想サーバーによる構成には対応できません。

どちらの構成でも、3 つの名前について名前とアドレスのマッピングを設定する必要があります。IP アドレスベースの構成では、それぞれの名前を別々のアドレスにマッピングします。同時に、それらのアドレスの接続をすべて受け入れるようにホストマシンを設定する必要があります。URL ホストベースの構成では、すべての名前を同じアドレス (もともとマシンに割り当てられているアドレス) にマッピングします。

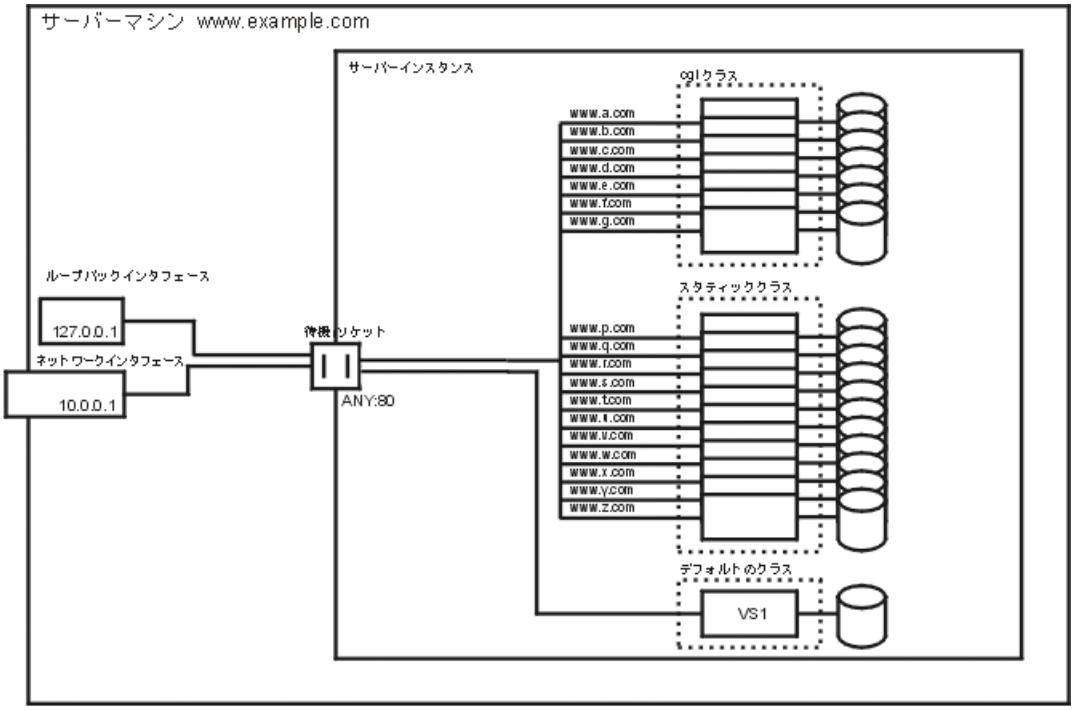
複数の待機ソケットを持つ構成では、要求の受け入れ先アドレスをサーバーが検索する必要がないため、負荷の上昇は最小限になります。ただし、複数の待機ソケットを使用すると、受け入れスレッドが増えるため、オーバーヘッド (メモリおよびスケジューリング) も増えます。

例 4 : マスホスティング

マスホスティングは、多数の低トラフィック仮想サーバーを使用可能にする構成です。たとえば、多数の低トラフィックの個人ホームページをホスト処理する ISP などは、このカテゴリに分類されます。

仮想サーバーは、通常は URL ホストベースであり、提供するサービスのレベルに応じて、複数の仮想サーバークラスの 1 つに属します。たとえば、スタティックコンテンツのみを使用できるクラスと、スタティックコンテンツと CGI を使用できるクラスを作成できます。

マスホスティング



この場合も、サーバーのインストール時にインストールされた仮想サーバー VS1 は、`defaultclass` に存在しています。

第 14 章

仮想サーバーの作成と設定

仮想サーバーのクラスには、仮想サーバー (クラスのメンバー) が関連付けられます。クラスレベルの設定の一部を、仮想サーバーレベルで無効にすることができます。この章では、個々の仮想サーバーを作成して設定する方法について説明します。仮想サーバークラスの設定については、「コンテンツ管理」を参照してください。仮想サーバーの概要については、「仮想サーバーの使用」を参照してください。

この章では、次の項目について説明します。

- 仮想サーバーの作成
- 仮想サーバーの設定内容の変更
- クラスマネージャを使用した変更
- 仮想サーバーマネージャを使用した変更
- 仮想サーバーの削除

仮想サーバーの作成

仮想サーバーを使用すると、インストールされた 1 つのサーバーで、複数の会社または個人に対して、ドメイン名、IP アドレス、およびサーバー管理機能を提供できます。仮想サーバーの紹介および Sun ONE Web Server での仮想サーバーの設定方法については、「仮想サーバーの使用」を参照してください。

仮想サーバーを作成するには、次の手順に従います。

1. クラスマネージャから、「Virtual Servers」タブを選択します。
2. 「Add Virtual Server」をクリックします。
3. 仮想サーバーの名前を選択します。
4. 仮想サーバーの URL ホストを選択します。

   複数の URL ホストを、空白文字で区切って入力できます。

5. 「OK」をクリックします。

仮想サーバーの作成に必要な設定項目はこれだけです。ただし、このタブのほかのページを使用して、さらに詳細な仮想サーバーの設定内容を設定できます。

仮想サーバーの設定内容の変更

仮想サーバーを設定したあと、設定内容を変更できます。仮想サーバーの設定を変更する方法は 2 通りあります。1 つはクラスマネージャを使用する方法で、もう 1 つは仮想サーバーマネージャを使用する方法です。

クラスマネージャでは、ページは変更する設定の種類別に編成されています。たとえば、クラスの 1 つまたは複数の仮想サーバーについてサービス品質の設定を変更する場合は、「Quality of Service」ページを使用します。

仮想サーバーマネージャでは、ページは 1 つの仮想サーバーだけに関係します。したがって、その仮想サーバーのすべての設定を表示したり変更したりできます。

クラスマネージャを使用した変更

次に示すクラスマネージャの各ページを使用して、仮想サーバーの設定を変更します。

仮想サーバーの設定内容の変更

仮想サーバーの一般設定を変更する場合は、「Edit Virtual Servers」ページを使用します。このページにアクセスするには、次の手順に従います。

1. クラスマネージャから、「Virtual Servers」タブをクリックします。
2. 「Edit Virtual Servers」をクリックします。
3. 仮想サーバーを編集するには、編集する仮想サーバーの隣のドロップダウンリストから「Edit」または「Delete」を選択します。

   デフォルトの仮想サーバーは、編集することはできますが、削除することはできません。

4. 「State」を、「On」、「Off」、または「Disabled」に設定します。

   管理者は仮想サーバーの状態を「Disabled」に設定しても再びオンに戻すことができますが、その仮想サーバーのエンドユーザーはオンにすることができません。

   これは仮想サーバーの状態であり、サーバーインスタンスのオンまたはオフとは関係ありません。このページで仮想サーバーの状態がオンと表示されている場合、サーバーインスタンスもオンのときだけ、その仮想サーバーは要求を受け付けることができます。

   デフォルトのサーバーインスタンスのデフォルトの仮想サーバーについても同じことがいえます。サーバーインスタンスをオフにすると、デフォルトの仮想サーバーはオンのままですが、接続を受け付けることはできません。

   サーバーインスタンスのデフォルトの仮想サーバーは、「Off」にしたり、「Disabled」にすることはできません。

5. 「Urlhosts」列に表示されているものと異なる URL ホストを使用する場合は、「URL Hosts」にそのホスト名を入力します。

   複数の URL ホストを、空白文字で区切って入力できます。

6. 仮想サーバーの変更作業が終了したら、「OK」をクリックします。

仮想サーバーの MIME の設定

個々の仮想サーバーに MIME タイプファイルを設定できます。MIME タイプファイルには、ファイル拡張子とファイルタイプのマッピング情報が格納されます。たとえば、MIME タイプファイルで、`.cgi` で終わるすべてのファイルを CGI ファイルとして扱うように指定できます。

仮想サーバーまたは仮想サーバークラスごとに個別の MIME タイプファイルを作成する必要はありません。必要な数の MIME タイプファイルを作成し、それらを仮想サーバーに関連付けます。サーバーには `mime.types` という MIME タイプファイルがデフォルトで 1 つ存在します。新しい MIME タイプファイルを作成するか、あるいは MIME タイプファイル内の定義を編集する場合は、175 ページの「MIME タイプの選択」を参照してください。

特定の仮想サーバーに MIME タイプファイルを設定するには、次の手順に従います。

1. クラスマネージャから、「Virtual Servers」タブをクリックします。
2. 「MIME Settings」をクリックします。
3. 仮想サーバーの横にあるドロップダウンリストから、MIME タイプファイルを選択します。
4. 「OK」をクリックします。

仮想サーバーの ACL の設定

ACL を使用して、仮想サーバーへのアクセスを制御できます。各仮想サーバーは、LDAP データベースで個別のベース DN を持つことができます。このため、各仮想サーバーは、Sun ONE Web Server で使用する 1 つの LDAP データベースに独自のエントリを持つことができます。

詳細は、227 ページの「仮想サーバーへのアクセス制御」を参照してください。

仮想サーバーのセキュリティの設定

仮想サーバーが、セキュリティ保護された待機ソケットにバインドされている場合、その仮想サーバーのセキュリティを設定できます。

セキュリティの詳細は、第 4 章「Web コンテナと Web アプリケーションの J2EE ベースのセキュリティ」を参照してください。

仮想サーバーのサービス品質の設定

サービス品質は、仮想サーバーに対して設定するパフォーマンス制限です。たとえば、ISP では、仮想サーバーで使用できる帯域幅の広さに応じて課金する必要がある場合があります。

サーバー全体または仮想サーバークラスに対してこの設定を有効にするには、サーバーマネージャの「Monitor」タブを使用します。ただし、個々の仮想サーバーについて、サーバー全体またはクラスレベルの設定を無効にできます。

仮想サーバーに対してサービス品質を有効にする前に、まず、サーバー全体に対してサービス品質を有効にし、基本的な値をいくつか設定する必要があります。258 ページの「サービス品質の使用法」を参照してください。

仮想サーバーのサービス品質を設定するには、次の手順に従います。

1. クラスマネージャから、「Virtual Servers」タブをクリックします。
2. 「Quality of Service」をクリックします。

   クラスのすべての仮想サーバーおよびそれらのサービス品質の設定項目のリストを示すページが表示されます。
3. 仮想サーバーのサービス品質を有効にするには、ドロップダウンリストから「Enable」を選択します。

   デフォルトでは、サービス品質は無効になっています。サービス品質を有効にすると、サーバーのオーバーヘッドがわずかに増えます。

4. その仮想サーバーの最大帯域幅をバイト／秒単位で設定します。
5. 最大帯域幅の設定を適用するかどうかを選択します。

   最大帯域幅を適用する場合、サーバーがその帯域幅の制限値に達すると、それ以上の接続は拒否されます。

   最大帯域幅を適用しない場合は、最大帯域幅を超えると、サーバーのエラーログにメッセージが記録されます。
6. その仮想サーバーに対して許可する最大接続数を選択します。

   この数は、同時に処理する要求の数です。
7. 最大接続数の設定を適用するかどうかを選択します。

   最大接続数を適用する場合、サーバーがその最大接続数に達すると、それ以上の接続は拒否されます。

   最大接続数を適用しない場合は、最大接続数を超えると、サーバーのエラーログにメッセージが記録されます。
8. 「OK」をクリックします。

サービス品質機能への制限に関する詳細は、258 ページの「サービス品質の使用法」を参照してください。

仮想サーバーのログの設定

仮想サーバーのアクセスログおよびエラーログの場所をデフォルトの場所から変更するには、次の手順に従います。

1. クラスマネージャから、「Virtual Servers」タブをクリックします。
2. 「Logging Settings」をクリックします。

   クラスのすべての仮想サーバーのリストおよびエラーログとアクセスログの場所を示すページが表示されます。
3. エラーログとアクセスログの絶対パスを入力します。既存のパスを入力する必要があります。

   デフォルトでは、すべての仮想サーバーに関するアクセスメッセージおよびエラーメッセージが、サーバーインスタンスのアクセスログおよびエラーログに記録されます。仮想サーバーごとの個別のログファイルを使用する場合は、ここでその設定を行います。
4. デフォルトのパスに戻す場合は、「Default」をクリックします。
5. 「OK」をクリックします。

特定の仮想サーバーのログを表示するには、次の手順に従います。

1. 仮想サーバーマネージャから、「Logs」タブを選択します。
2. 「View Access Log」または「View Error Log」をクリックします。
3. 表示するエントリ数および表示の条件を選択します。

   たとえば、すべての仮想サーバーに関するエントリが記録されているログでは、特定の仮想サーバーのエントリだけを表示できます。
4. 「OK」をクリックします。

仮想サーバーのログの有効化

仮想サーバーレベルのログを有効にするには、次の手順に従います。

1. サーバーマネージャでサーバーインスタンスの「Logs」タブを開き、「Log Preferences」を選択します。
2. 「Log File」フィールドにパスとファイル名を入力して、新しいアクセスログを作成します。

   `magnus.conf` ファイルを次のように変更して、新しいアクセスログを手動で作成することもできます。

   `Init fn=init access="$accesslog"` を `Init fn=init access="newaccesslog"` に変更します。
3. 「Format」の下の「Only Log」を選択し、「Virtual Server Id」にチェックマークを付けます。

   カスタムフォーマットでは、「Custom Format」を選択し、行の最後に `%vsid%` を追加します。

   `%vsid%` は、複数の仮想サーバーを使用する場合に便利です。このエントリは、アクセスログに vsid を記録します。

   `magnus.conf` ファイルの `Init fn` の最後に `%vsid%` を手動で追加することもできます。
4. 「OK」をクリックします。
5. 「Apply」をクリックします。
6. 「Apply Changes」をクリックして変更を適用します。

仮想サーバーの Java Web アプリケーションの設定

Web アプリケーションは、Java サーブレット、JSP、HTML ページ、クラス、その他のリソースの集合です。すべてのリソースは 1 つのディレクトリに格納され、そのディレクトリに対する要求はすべて、アプリケーションを実行させるものとなります。特定の仮想サーバーの Web アプリケーションを配備および編集するときは、仮想サーバーマネージャの「Web Applications」タブにあるページを使用します。

Web アプリケーションと Web アプリケーションの配備記述子ファイル `sun-web.xml` の詳細は、『Sun ONE Web Server 6.1 Administrator's Configuration File Reference』を参照してください。

仮想サーバーマネージャを使用した変更

仮想サーバーマネージャには「Preferences」、「Logs」、「Web Applications」、「WebDAV」の 4 つのタブがあります。

「Preferences」タブには、次のページがあります。

- 「Status」
- 「Settings」

「Status」ページには、いくつかの設定項目と、仮想サーバーのアクセスログとエラーログへのリンクが表示されます。

「Settings」ページには、仮想サーバーに関する次の設定項目が表示されます。

- 状態 (オンまたはオフ)
- ドキュメントルート
- アクセスログとエラーログのディレクトリ
- ディレクトリサービス
- ACL ファイル
- MIME タイプファイル
- CGI 設定

1 つの仮想サーバーの設定を変更する場合は、仮想サーバーマネージャを使用して 1 つのページですべての設定を変更する方法が便利です。

「Logs」タブには、選択した仮想サーバーのレポートを生成するためのページが 1 ページだけ含まれています。

Web アプリケーションファイルの配備と編集の詳細については、第 15 章「プログラムによるサーバーの拡張」を参照してください。

「WebDAV」タブでは、仮想サーバーの WebDAV コレクションの作成と編集を行うことができます。WebDAV は、WebDAV 操作が有効なリソースまたはリソースの集合です。WebDAV を使用することで、Web 上のさまざまな場所から共同でドキュメントを作成できます。WebDAV を使用して、WebDAV が有効なリソースに対してさまざまなレベルのロックを設定できます。このため、Web 上でのコンテンツを共同で作成する場合に、上書きの衝突を防止することができます。

「WebDAV」タブには、次のページがあります。

- 「Add Collection」ページ
- 「Edit DAV Collection」ページ
- 「Lock Management」ページ

「Add Collection」ページでは、WebDAV コレクションを作成できます。

「Edit DAV Collection」ページでは、WebDAV が有効なコレクションを設定できます。

「Lock Management」ページでは、サーバー上の WebDAV が有効なリソースについて、設定されているロックと、ロックに関連するその他の情報を参照できます。

詳細は、第 19 章「WebDAV による Web パブリッシング」を参照してください。

仮想サーバーのレポートの生成

仮想サーバーマネージャを使用して、1 つの仮想サーバーに関するレポートを生成できるようになりました。レポートを生成するには、以下に示すように仮想サーバーが使用する新規アクセスログを作成し、この新規アクセスログを仮想サーバーの設定に追加します。

仮想サーバーのレポートを生成するには、次の手順に従います。

1. サーバーマネージャでサーバーインスタンスの「Logs」タブを開き、「Log Preferences」を選択します。
2. 「Log File」フィールドにパスとファイル名を入力して、新しいアクセスログを作成します。

   `magnus.conf` ファイルを次のように変更して、新しいアクセスログを手動で作成することもできます。

   `Init fn=init access="$accesslog"` を `Init fn=init access="newaccesslog"` に変更します。

3. 「Format」の下の「Only Log」を選択し、「Virtual Server Id」にチェックマークを付けます。

   カスタムフォーマットでは、「Custom Format」を選択し、行の最後に `%vsid%` を追加します。

   `%vsid%` は、複数の仮想サーバーを使用する場合に便利です。このエントリは、アクセスログに vsid を記録します。

   `magnus.conf` ファイルの `Init fn` の最後に `%vsid%` を手動で追加することもできます。
4. 「OK」をクリックします。
5. 「Apply」をクリックします。
6. 「Apply Changes」をクリックして変更を適用します。
7. レポートを生成する仮想サーバーを選択して「Virtual Server Manager」>「Manage Classes」に移動し、ツリービューから「Virtual server」を選択します。
8. 「Preferences」タブを表示し、「Settings」を選択します。

   「Access Log」フィールドで、アクセスログを新しいアクセスログに変更します。
9. 「OK」をクリックします。
10. 「Apply」をクリックします。
11. 「Apply Changes」をクリックして変更を適用します。
12. 「Logs」タブを選択します。

    「Generate Reports」ページが表示されます。

    このページは、仮想サーバーが作成され、`LogVSid` が「On」に設定されていない場合は表示されません。仮想サーバー ID の有効化については、「仮想サーバーのログの有効化」を参照してください。
13. (省略可能) 必要に応じて設定を変更します。
14. 「OK」をクリックしてレポートを生成します。

仮想サーバーのディレクトリサービスの選択

特定の仮想サーバーに特定のディレクトリサービスを割り当てることができます。この場合、選択したディレクトリサービスは、`server.xml` ファイル内の対応する `VS` (仮想サーバー) 要素の `USERDB` 要素に記録されます。このディレクトリサービスに関連する権限とアクセス権は、アクセス制御規則を評価および適用するときに、サーバーによって使用されます。

仮想サーバーにディレクトリサービスを割り当てるには、次の手順に従います。

1. 仮想サーバーマネージャの「Settings」タブを選択します。

   仮想サーバーの設定がリスト表示されます。

2. 「Directory Service」の隣にある「Edit」リンクをクリックします。

   「Pick Directory Services for Virtual Server」ページが新しいウィンドウに表示されます。

3. ディレクトリサービスを選択し、「OK」をクリックします。
4. 変更を保存して適用します。

注 特定の仮想サーバー用に選択したディレクトリサービスは、他の仮想サーバーとは共有されません。一方、アクセス制御ファイルは仮想サーバー間で共有されます。

仮想サーバーの削除

仮想サーバーを削除するには、次の手順に従います。

1. クラスマネージャから、「Virtual Servers」タブをクリックします。
2. 「Edit Virtual Servers」をクリックします。
3. 目的の仮想サーバーの横にあるドロップダウンリストから、「Delete」を選択します。

   サーバーのインストール時に作成されたデフォルトの仮想サーバーは削除できません。

4. 「OK」をクリックします。

   仮想サーバーが削除されます。

第 15 章

プログラムによるサーバーの拡張

この章では、クライアントからの要求に応じて HTML ページをダイナミックに生成するプログラムを Sun ONE Web Server にインストールする方法について説明します。このようなプログラムはサーバーサイドアプリケーションと呼ばれます。(クライアントにダウンロードされるクライアントサイドアプリケーションは、クライアントマシン上で動作します。)

この章には、次の内容が記述されています。

- サーバーサイドプログラムの概要
- Java サーブレットと JavaServer Pages (JSP)
- CGI プログラムのインストール
- Windows CGI プログラムのインストール
- Windows でのシェル CGI プログラムのインストール
- クエリハンドラの使用

サーバーサイドプログラムの概要

Java サーブレットと CGI プログラムは、それぞれ長所や用途が異なります。次に、このようなサーバーサイドプログラムの相違点について説明します。

- Java サーブレットは Java で記述します。Java はネットワークアプリケーションを作成するための機能の豊富なプログラム言語です。
- Common Gateway Interface (CGI) プログラムは、C、Perl またはその他のプログラミング言語で記述できます。CGI プログラムはすべて、標準的な方法でクライアントとサーバーの間で情報をやりとりします。

サーバーで実行するサーバーサイドアプリケーションのタイプ

Sun ONE Web Server では、サーバーサイドアプリケーションの次のタイプを実行して、コンテンツをダイナミックに生成することができます。

- Java サーブレット
- CGI プログラム

また、Sun ONE Web Server では、サーバー自体の動作を拡張したり修正したりするプログラムも実行できます。プラグインと呼ばれるこれらのプログラムは、Netscape Server Application Programming Interface (NSAPI) を使用して記述します。プラグインプログラムの作成とインストールについての詳細は、『Sun ONE Web Server 6.1 NSAPI Programmer's Guide』を参照してください。

サーバーへのサーバーサイドアプリケーションのインストール方法

プログラムのタイプによって、サーバーへのインストール方法が異なります。それぞれのインストールの手順は次のとおりです。

- Java サーブレットの場合は、Web アプリケーションを作成して配備できます。詳細は、352 ページの「サーバーでサーブレットを実行するための要件」を参照してください。
- CGI プログラムの場合は、特定のファイル名拡張子が付いているすべてのファイル、または、指定されたディレクトリにあるすべてのファイル、あるいは、その両方を CGI プログラムとして認識するようにサーバーを設定することができます。詳細については、359 ページの「CGI プログラムのインストール」、364 ページの「Windows CGI プログラムのインストール」、および 367 ページの「Windows でのシェル CGI プログラムのインストール」を参照してください。

次の節では、これらのインストール方法を説明します。

Java サーブレットと JavaServer Pages (JSP)

この節では、Sun ONE Web Server での Java サーブレットおよび JavaServer Pages のインストール方法と使用方法を説明します。

説明する内容を次に示します。

- サーブレットと JavaServer Pages の概要

- サーバーでサーブレットを実行するための要件
- Web アプリケーションの配備
- Web アプリケーションに含まれていないサーブレットと JSP の配備
- JVM の設定
- バージョンファイルの削除

サーブレットと JavaServer Pages の概要

Sun ONE Web Server 6.1 はサーブレット 2.3 API 仕様をサポートします。この仕様では、Web アプリケーションにサーブレットと JSP を組み込むことができます。

Web アプリケーションは、サーブレット、JSP (JavaServer Pages)、HTML ドキュメント、およびその他の Web リソースの集合です。Web リソースには、イメージファイル、圧縮アーカイブ、その他のデータが含まれる場合があります。Web アプリケーションは、アーカイブ (WAR ファイル) にパッケージ化される場合と、オープンディレクトリ構造に置かれる場合があります。

注	Servlet API バージョン 2.3 にはバージョン 2.1 との完全な下位互換性があるため、既存のサーブレットは変更またはコンパイルし直さなくても引き続き機能します。

サーブレットの開発には、Sun Microsystems の Java Servlet API を使用します。Java Servlet API の使用の詳細は、以下の Web サイトにある Sun Microsystems のドキュメントを参照してください。

`http://java.sun.com/products/servlet/index.html`

JSP は、HTML ページのように、Web ブラウザで表示できます。ただし、HTML タグのほかに、JSP タグや、Java コードと組み合わせた指令を指定することができるため、Web ページを作成する際にダイナミックコンテンツをページに取り込むことができます。これらの追加機能によって、属性値の表示や単純な条件の使用などが可能になります。Sun ONE Web Server 6.1 は、JavaServer Pages (JSP) 1.2 API 仕様をサポートしています。

注	アプリケーション要求の URI の大文字と小文字は (たとえば、`¥foo.JSP`)、ファイルシステムパスの標準的な表記 (たとえば、`C:¥Program Files¥WebServer¥docs¥foo.jsp`) と一致させてください。これは、Sun ONE Web Server 6.1 Java Web コンテナが、現時点ではパターンマッチで大文字と小文字を区別しているためです。

JSP の作成の詳細は、以下の Sun Microsystems の JSP に関する Web サイトを参照してください。

`http://java.sun.com/products/jsp/index.html`

Sun ONE Web Server で使用する サーブレットと JSP の開発については、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』を参照してください。

サーバーでサーブレットを実行するための要件

Sun ONE Web Server には、Java Development Kit (JDK) 1.4.1_03 が含まれています。従来のバージョンの Web Server では Java はサーバー全体に設定されていましたが、バージョン 6.1 では、Web サーバーのインスタンスごとに Java を設定できます。

Sun ONE Web Server 6.1 にバンドルされている JDK、または任意の JDK を使用することができます。任意の JDK を使用する場合は、その JDK へのパスを指定する必要があります。この方法については、287 ページの「JVM の設定」を参照してください。

デフォルトでは、Sun ONE Web Server のインストール時に Java は無効に設定されます。サーブレットを有効にするには、最初に Java を有効にする必要があります。

Java を有効にする方法については、285 ページの「Java の有効化と無効化」を参照してください。

Web アプリケーションの配備

次に、`wdeploy` コマンド行ユーティリティを使用して手動で、またはユーザーインタフェースを使用して、Web アプリケーションを配備、編集、削除する方法について説明します。

server.xml ファイルの使用

配備した Web アプリケーションは、デフォルトでは有効になっています。配備した Web アプリケーションを手動で無効にするには、`server.xml` ファイルを次のように編集する必要があります。

```
<VS>
<WEBAPP uri="/mywebapp" path="/webappdir" enabled = "false" >
</WEBAPP>

...

</VS>
```

同じ配備記述子で複数の Web アプリケーションを配備または編集してしまった場合、いずれかのアプリケーションは無効になり、`enabled = "false"` は無視されて `enabled = "true"` というデフォルト設定が引き続き適用されます。

`server.xml` ファイルの詳細については、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』を参照してください。

Web アプリケーションは、次の 2 つの方法で配備および編集できます。

- 管理サーバーインタフェースを使用
- コマンド行インタフェースを使用

管理サーバーインタフェースを使用

Sun ONE Web Server の 6.0 Service Pack 1 からは、指定の仮想サーバーの Web アプリケーションを配備、編集、削除、有効化、無効化することができます。

Web アプリケーションの配備

「Deploy Web Applications」ページにアクセスするには、仮想サーバーマネージャの「Web Applications」タブにある「Deploy Web Applications」を選択します。

Web アプリケーションを配備するには、次の手順に従います。

1. 「WAR File On」ドロップダウンリストから「Local Machine」または「Server Machine」を選択します。

   ローカルマシンからサーバーに、WAR ファイルをアップロードする場合は、「Local Machine」を選択します。サーバーマシンに WAR ファイルがすでに存在している場合は、「Server Machine」を選択します。

2. 表示されるフィールドに、ローカルマシンまたはサーバーマシン上で Web アプリケーションが格納されている `WAR` ファイルへのパスを入力します。

   サーバーマシンの場合は、WAR ファイルへの絶対パスを入力します。

   ローカルマシンの場合は、パスをブラウズして選択します。「Browse」をクリックすると、「File Upload」ウィンドウが開き、サーバーにアップロードする WAR ファイルを選択することができます。

3. 表示されるフィールドに、Web アプリケーションの仮想サーバー上の URI を入力します。

4. 抽出した WAR ファイルコンテンツの格納先となる、サーバーマシン上のディレクトリへの絶対パスを指定します。ディレクトリが存在しない場合、ディレクトリが作成されます。

5. 「OK」をクリックします。

6. 「Apply」をクリックします。

7. 配備する Web アプリケーションの「Dynamic Reconfiguration」を選択します。

Web アプリケーションの編集

すでに配備されている Web アプリケーションを削除、無効化、有効化することができます。「Edit Web Applications」ページにアクセスするには、仮想サーバーマネージャの「Web Applications」タブにある「Edit Web Applications」を選択します。

すでに配備されている Web アプリケーションを編集、削除、無効化、有効化するには、次の手順に従います。

1. 編集する Web アプリケーションの隣にある「Action」列のドロップダウンリストから、実行する操作を選択します。
   - 「Edit」- Web アプリケーションにアクセスするための URI を変更します。
   - 「Delete」- Web アプリケーションファイルから Web アプリケーションエントリを削除し、またそのアプリケーションが配備されているディレクトリを削除します。
   - 「Disable」- URI からその Web アプリケーションにアクセスできないようにしますが、削除は行いません。
   - 「Enable」- 無効にした Web アプリケーションを再び有効にします。

警告 Web アプリケーションを削除すると、アプリケーションが配備されていたディレクトリも削除されます。

2. (省略可能) Web アプリケーションを編集する場合は、「URI」フィールドに新しい URI を入力します。
3. 「OK」をクリックします。
4. 「Apply」をクリックします。
5. 配備する Web アプリケーションの「Dynamic Reconfiguration」を選択します。

コマンド行インタフェースを使用

手動で Web アプリケーションを配備する前に、*server_root*`/bin/https/httpsadmin/bin` ディレクトリがパスにあり、`IWS_SERVER_HOME` 環境変数が *server_root* ディレクトリに設定されていることを確認してください。

仮想サーバー Web アプリケーションを配備するには

コマンド行で `wdeploy` ユーティリティを使用して、仮想サーバーの Web アプリケーション環境に WAR ファイルを配備できます。

```
wdeploy deploy -u <uri_path> -i <instance> -v <vs_id> [ [-V <verboseLevel>] |
[-q] ] [-n] [-d <directory>] <war_file>
```

仮想サーバー Web アプリケーションを削除するには

```
wdeploy delete -u <uri_path> -i <instance> -v <vs_id> [ [-V <verboseLevel>] |
[-q] ] [-n] hard|soft
```

仮想サーバーの Web アプリケーションの URI とディレクトリを一覧表示するには

```
wdeploy list -i <instance> -v <vs_id> [ [-V <verboseLevel>] | [ -q] ]
```

コマンドのパラメータには以下の意味があります。

uri_path	Web アプリケーションの URI プレフィックス
instance	サーバーインスタンス名
vs_id	仮想サーバーの ID
directory	省略可能。アプリケーションが配備、または削除されるディレクトリ。指定しない場合、アプリケーションはドキュメントのルートディレクトリに配備される
`hard` | `soft`	ディレクトリと `server.xml` エントリを削除 (`hard`) するか、`server.xml` エントリだけを削除 (`soft`) するかを指定する
war_file	WAR ファイルの名前
verboseLevel	コンソールに表示するログメッセージの詳細レベル。0 ～ 4 の範囲で指定する。デフォルト値は 1 Sun ONE Web Server 6.1 では、この要素の代わりに `server.xml` の `LOG` 要素にある `loglevel` 属性が使用される
`-q`	(quiet の略) ログメッセージの詳細レベルをゼロに設定する。`-V 0` という設定と同じ意味
`-n`	`wdeploy` が自動的に Web サーバーに再設定コマンドを送信することを防止する。詳細は、「wdeploy コマンドでの -n の使用」を参照してください。

警告 Web アプリケーションを配備する際に *directory* を指定しない場合、アプリケーションはドキュメントのルートディレクトリに配備されます。その場合、`hard` パラメータを使用してアプリケーションを削除すると、ドキュメントのルートディレクトリも削除されます。

`wdeploy deploy` コマンドを実行すると、以下の 3 つの事柄が生じます。

- 指定された *uri_path* と *directory* とともに Web アプリケーションが `server.xml` ファイルに追加される
- 指定された *directory* に WAR ファイルが抽出される
- サーバーがダイナミックに再設定されて、新しい Web アプリケーションがロードされる

その例を次に示します。

```
wdeploy deploy -u /hello -i server.sun.com -v acme.com
-d /s1ws61/https-server.sun.com/acme.com/web-apps/hello
/s1ws61/plugins/servlets/examples/web-apps/HelloWorld/HelloWorld.wa
r
```

このユーティリティを実行した結果として、`server.xml` には、以下のエントリが追加されています。

```
<VS>
    <WEBAPP uri="/hello"
    dir="/s1ws61/https-server.sun.com/acme.com/webapps/hello"/>
</VS>
```

以下に、`/s1ws61/https-server.sun.com/acme.com/web-apps/hello` ディレクトリの内容を示します。

```
colors
index.jsp
META-INF
WEB-INF/
    web.xml
    /classes/
        HelloWorldServlet.class
        HelloWorldServlet.java
        SnoopServlet.class
        SnoopServlet.java
```

wdeploy コマンドでの -n の使用

Sun ONE Web Server のバージョン 6.0 Service Pack 1 では、Web アプリケーションの配備または削除後に、`wdeploy` を使用してサーバーをダイナミックに再設定し、配備または削除された Web アプリケーションをサーバーにロードしたり、サーバーからアンロードしたりします。従来は変更を適用するために、次のいずれかの方法でサーバーを明示的に再設定する必要がありました。

- reconfig スクリプトを使用する
- サーバーを再起動する
- 管理ユーザーインタフェースで「Apply」リンクをクリックする

現在では、wdeploy コマンドを正常に実行することで、新しい Web アプリケーションへの要求を受け付けたり、削除した Web アプリケーションへの要求を受け付けないようにすることができます。

-n オプションは、wdeploy が自動的に Web サーバーに再設定コマンドを送信することを防止します。スクリプトなどで複数の Web アプリケーションを配備または配備解除する場合に、Web アプリケーションの最後の配備以降に 1 回だけサーバーを再設定したいときは、コマンドに -n オプションを指定します。

配備した Web アプリケーションへのアクセス

アプリケーションを配備すると、ブラウザに以下の URL を指定して、アプリケーションにアクセスできます。

`http://`*vs_urlhost*`[:`*vs_port*`]/`*uri_path*`/[`*index_page*`]`

URL の各部には以下の意味があります。

vs_urlhost	仮想サーバーの `urlhosts` 値のひとつ
vs_port	(省略可能) 仮想サーバーがデフォルトのポートを使用しない場合に限り必要
uri_path	アプリケーションの配備に使用したものと同じ。コンテキストパス
index_page	(省略可能) エンドユーザーが最初にアクセスするアプリケーションのページ

その例を次に示します。

```
http://acme.com:80/hello/index.jsp
```

または

```
http://acme.com/hello/
```

戻り値

wdeploy オプションを実行すると、次の値を返します。

- `0` : wdeploy オプションが正常に実行されたことを示す
- `1` : コマンド行引数が無効である、または設定ファイルの内容が無効であるために、wdeploy の実行時にエラーが発生したことを示す
- `2` : オペレーティングシステムの設定が原因でエラーが発生したことを示す。指定したディレクトリが存在しないか、ファイルへのアクセス権が設定されていない

Web アプリケーションに含まれていないサーブレットと JSP の配備

Web アプリケーションに含まれていない 4.*x* のサーブレットや JSP を配備できますが、配備先は、デフォルトの仮想サーバー内だけです。詳細については、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』を参照してください。

JVM の設定

サーバーマネージャの「Java」タブでは、Java 仮想マシン (Java Virtual Machine、JVM) の属性を設定できます。

これらのオプションの詳細については、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』を参照してください。

バージョンファイルの削除

サーバーマネージャの「Java」タブの「Delete Version Files」ページでは、JavaServer Pages クラスキャッシュとセッションデータキャッシュのバージョン番号が保存されているファイルを削除できます。このページには、次のフィールドがあります。

Clear Session Data

サーバーが MMapSessionManager セッションマネージャを使用する場合に持続セッション情報が保存される SessionData ディレクトリを削除します。

Delete JSP ClassCache Files

JavaServer Pages (JSP) のキャッシュ情報が保存される ClassCache ディレクトリを削除します。次に、このディレクトリのデフォルトの場所を示します。

```
server_root/https-server_id/ClassCache/virtual_server_id/webapp_uri/
```

サーバーは「JSP」ページを処理するときに、JSP に関連付けられた .java ファイルと .class ファイルを作成し、ClassCache ディレクトリの下の JSP クラスキャッシュに保存します。

サーバーは JavaServer Pages (JSP) とサーブレットの情報をキャッシュに書き込むために、2 つのディレクトリを使用します。

- `ClassCache`

  JSP (JavaServer Pages) の情報をキャッシュするために、サーバーでは以下のディレクトリが使用されます。

server_root/https-*server_id*/ClassCache/*virtual_server_id*/*webapp_uri*/
(*virtual_server_id* は仮想サーバー ID、*webapp_uri* は Web アプリケーションの URI を示します)

サーバーは JSP ページを処理すると、JSP に関連付けられた .java および .class ファイルを作成し、ClassCache ディレクトリの下位の JSP クラスキャッシュに格納します。

- SessionData

  MMapSessionManager セッションマネージャを使用する場合にサーバーは、SessionData ディレクトリに持続セッション情報を保存します。

各キャッシュには version ファイルがあります。このファイルには、キャッシュ内のファイルやディレクトリの構造を決めるためにサーバーが使用するバージョン番号が保存されています。バージョンファイルを削除するだけで、キャッシュをクリーンアップすることができます。

起動時にバージョンファイルが見つからない場合にサーバーは、対応するキャッシュのディレクトリ構造を削除して、バージョンファイルを作成し直します。次回サーバーが JSP ページを処理するときに、JSP クラスキャッシュを作成し直します。また、次に MMapSessionManager セッションマネージャを使用して JSP またはサーブレットを処理するときにサーバーは、セッションデータのキャッシュを作成し直します。

サーバーの将来のアップグレードでキャッシュの異なる形式を使用する場合、サーバーはバージョンファイル内の番号を確認して、バージョン番号が正しくない場合はキャッシュを消去します。

CGI プログラムのインストール

この節では、CGI プログラムのインストール方法について説明します。説明する内容を次に示します。

- CGI の概要
- CGI ディレクトリの指定
- ファイルタイプとして CGI を指定
- 実行可能ファイルのダウンロード

さらに、次の節では Windows に固有の CGI プログラムのインストール方法を説明します。

- Windows CGI プログラムのインストール
- Windows でのシェル CGI プログラムのインストール

CGI の概要

Common Gateway Interface (CGI) プログラムは、多くのプログラミング言語で定義できます。UNIX/Linux マシンでは、Bourne シェルや Perl スクリプトで記述された CGI プログラムが一般的です。

注	UNIX/Linux の場合は、CGI の実行を補助するためにサーバーで使用する `CGIStub` プロセスが追加されます。このようなプロセスは、CGI に最初にアクセスしたときにだけ作成されます。その番号は、CGI によるサーバーへの負荷によって異なります。CGIStub プロセスを終了しないでください。サーバーを停止すると、これらのプロセスは消滅します。

Windows コンピュータでは、C++ で作成した CGI プログラムまたはバッチファイルが使用される場合があります。Windows の場合、Visual Basic などの Windows ベースのプログラム言語で作成された CGI プログラムが、異なるメカニズムを使用してサーバーと連動して動作します。このようなプログラムは、Windows CGI プログラムと呼ばれます。Windows CGI についての詳細は、364 ページの「Windows CGI プログラムのインストール」を参照してください。

注	コマンド行ユーティリティを実行するには、手動で `Path` 変数を設定して、*server_root*`/bin/https/bin` を組み込む必要があります。

プログラミング言語に関係なく、すべての CGI プログラムが同じ方法でデータの受け渡しを行います。CGI プログラム作成についての詳細は、次の情報リソースを参照してください。

- 『Sun ONE Web Server 6.1 Programmer's Guide』
- 次のサイトの「The Common Gateway Interface」

  `http://hoohoo.ncsa.uiuc.edu/cgi/overview.html`
- オンラインドキュメントの Web サイトで利用できる「CGI」の項。URL は次のとおり

  `http://docs.sun.com`

サーバーマシンに CGI プログラムを格納するには、次の 2 つの方法があります。

- CGI プログラムだけを格納するディレクトリを指定します。ファイル拡張子に関係なく、すべてのファイルがプログラムとして実行されます。

- すべての CGI プログラムを特定のファイルタイプとして指定します。つまり、すべてのファイルで .cgi、.exe、または .bat などの拡張子を使用します。プログラムは任意のディレクトリ内、またはドキュメントルートディレクトリの下に配置できます。

必要があれば、同時に両方のオプションを有効にすることができます。

どちらの方法にも利点があります。特定のユーザーだけが CGI プログラムを追加できるようにする場合、特定のディレクトリに CGI プログラムを格納し、そのディレクトリへのアクセスを制限します。HTML ファイルを追加できる任意のユーザーが CGI プログラムを追加できるようにする場合は、ファイルタイプを指定する方法を使用します。ユーザーは HTML ファイルと同じディレクトリに CGI ファイルを格納できます。

ディレクトリの方法を選択した場合、サーバーはそのディレクトリ内のすべてファイルを CGI プログラムとして解釈しようとします。同様に、ファイルタイプの方法を選択した場合、サーバーはファイル拡張子として .cgi、.exe、または .bat が付いたすべてのファイルを CGI プログラムとして処理しようとします。ファイルにこれらの拡張子が 1 つ付いていて CGI プログラムではない場合、ユーザーがアクセスしようとするとエラーが発生します。

注	デフォルトでは、CGI プログラムのファイル拡張子は .cgi、.exe、および .bat です。ただし、MIME タイプのファイルを修正して、CGI プログラムを示す拡張子を変更できます。MIME タイプのファイルを変更するには、サーバーマネージャの「Preferences」タブを選択し、「MIME Types」リンクをクリックします。

CGI ディレクトリの指定

仮想サーバーのクラスに対して CGI 専用ディレクトリを指定するには、次の手順を実行します。

1. クラスマネージャの「Programs」タブを選択します。

   「CGI Directory」ウィンドウが表示されます。

2. 「URL prefix」フィールドで、このディレクトリに使用する URL プレフィックスを入力します。つまり、入力したテキストは、CGI プログラムのディレクトリとして URL に表示されます。

   たとえば、URL プレフィックスとして「cgi-bin」と入力する場合、このような CGI プログラムへのすべての URL が次の構造になります。

   `http://`*`yourserver.domain.com`*`/cgi-bin/`*`program-name`*

注	指定する URL プレフィックスは、次の手順 3 で指定する実際の CGI ディレクトリと同じである必要はありません。

3. 「CGI Directory」テキストフィールドに、ディレクトリの場所を絶対パスで入力します。このディレクトリは、必ずしもドキュメントルートの下である必要はありません。このため、前の手順では URL プレフィックスを指定することが必要です。
4. 「OK」をクリックします。
5. 変更を保存して適用します。

既存の CGI ディレクトリを削除するには、「CGI Directory」フォームで CGI ディレクトリの「Remove」ボタンをクリックします。既存のディレクトリの URL プレフィックスまたは CGI ディレクトリを変更するには、ディレクトリの「Edit」ボタンをクリックします。

指定したディレクトリに CGI プログラムをコピーします。これらのディレクトリ内のファイルはすべて CGI ファイルとして処理されるため、CGI ディレクトリには HTML ファイルを置かないでください。

各仮想サーバーに固有の CGI 属性を設定する

単一の仮想サーバーに CGI 属性を指定するには、次の手順を実行します。

1. クラスサーバーで仮想サーバーを選択し、「Manage」ボタンをクリックします。
2. 仮想サーバーマネージャの「Settings」リンクを選択します。
3. 「CGI User」テキストフィールドに、CGI プログラムを実行するユーザーの名前を入力します。
4. 「CGI Group」テキストフィールドに、CGI プログラムを実行するグループの名前を入力します。
5. 「CGI Directory」テキストフィールドに、実行の開始前に `chroot` 後、`chdir` するディレクトリを入力します。

6. (UNIX のみ) 「CGI Nice」テキストフィールドに、CGI プログラムのサーバーに対する優先度を指定する増分を入力します。通常、サーバーは nice 値 0 で動作し、nice 値は、0 (CGI プログラムがサーバーと同じ優先度で動作する) から 19 (CGI プログラムがサーバーよりも低い優先度で動作する) までの間になります。nice 値の増分として -1 を指定し、CGI プログラムをサーバーよりも優先することはできますが、これは推奨されません。
7. 「Chroot Directory」テキストフィールドに、実行の開始前に chroot するディレクトリを入力します。
8. 「OK」をクリックします。
9. 変更を保存して適用します。

ファイルタイプとして CGI を指定

CGI プログラムをファイルタイプとして指定するには、次の手順を実行します。

1. クラスマネージャの「Programs」タブを選択します。
2. 「CGI File Type」ページをクリックします。

   「CGI as a File Type」ウィンドウが表示されます。
3. 「Editing」ピッカーから、この変更を適用するリソースを選択します。
4. 「Activate CGI as a file type ?」の下の「Yes」ラジオボタンをクリックします。
5. 「OK」をクリックします。
6. 変更を保存して適用します。

CGI ファイルには、ファイル拡張子 `.bat`、`.exe`、または `.cgi` を付ける必要があります。これらの拡張子が付いている CGI ファイル以外のファイルは、サーバーによって CGI ファイルとして処理され、エラーが発生します。

実行可能ファイルのダウンロード

CGI ファイルタイプとして `.exe` を使用している場合、`.exe` ファイルを実行可能ファイルとしてダウンロードできません。

この問題に対する解決方法の 1 つに、ユーザーにダウンロードしてもらう実行可能ファイルを圧縮し、拡張子を `.exe` 以外にする方法があります。この解決方法には、さらにダウンロード時間が短くなるという利点もあります。

別の解決方法として、magnus-internal/cgi タイプからファイル拡張子としての .exe を削除し、代わりに application/octet-stream タイプ (通常のダウンロード可能なファイルの MIME タイプ) に追加することもできます。サーバーマネージャでこれを実行するには、「Preferences」タブを選択し、「MIME Types」リンクをクリックします。ただし、この方法には、変更を行なったあと、.exe ファイルを CGI プログラムとして使用できなくなるという欠点があります。

さらに、サーバーの obj.conf ファイルを編集してダウンロードディレクトリを設定する方法もあります。このディレクトリ内のファイルはすべて自動的にダウンロードされます。サーバーの他の部分は影響を受けません。このディレクトリの設定方法については、次の場所にある技術情報を参照してください。

http://help.netscape.com/default.jsp

Windows CGI プログラムのインストール

この節では、Windows CGI プログラムのインストール方法について説明します。この節の内容は次のとおりです。

- Windows CGI プログラムの概要
- Windows CGI ディレクトリの指定
- ファイルタイプとして Windows CGI を指定

Windows CGI プログラムの概要

Windows CGI プログラムは、ほかの CGI プログラムとほぼ同様に処理されます。Windows CGI プログラムだけを含むディレクトリを指定するか、またはすべての Windows CGI プログラムに同じファイル拡張子を付けるように指定します。ほかの CGI プログラムと同様に、必要な場合は、同時に両方の方法を使用できます。たとえば、すべての Windows CGI プログラムを格納するディレクトリを作成し、Windows CGI ファイルの拡張子を 1 つ指定します。

Windows CGI プログラムは通常の CGI プログラムと同様に動作しますが、サーバーでの実際のプログラムの処理方法がわずかに異なります。このため、Windows CGI プログラムには別のディレクトリを指定する必要があります。Windows CGI ファイルタイプを有効にする場合、ファイル拡張子 .wcg を使用します。

Sun ONE Web Server は次の点を除いて、Windows CGI 1.3a の非公式仕様をサポートします。

- セキュリティメソッドをサポートするために、次のキーワードが [CGI] セクションに追加されています。

  - HTTPS: トランザクションが SSL を介して実行されるかどうかに応じて、値はオンまたはオフ
  - HTTPS Keysize: HTTPS がオンの場合、この値は暗号化に使用されるセッションキーのビット数を示す
  - HTTPS Secret Keysize: HTTPS がオンの場合、この値はサーバーの非公開鍵の生成に使用されるビット数を示す
- サーバーのドキュメントルートが単一ではないため、[CGI] セクションのキーワード Document Root が、予測されたドキュメントルートを参照しない場合があります。この変数で返されるディレクトリは、Windows CGI プログラムのルートディレクトリです。
- [CGI] セクションのキーワード Server Admin はサポートされません。
- [CGI] セクションのキーワード Authentication Realm はサポートされません。
- multi-part/form-data で符号化されて送信されるフォームはサポートされません。

Windows CGI ディレクトリの指定

Windows CGI ディレクトリを指定するには、次の手順を実行します。

1. クラスマネージャの「Programs」タブを選択します。
2. 「WinCGI Directory」リンクをクリックします。

   「WinCGI Directory」ウィンドウが表示されます。
3. 「URL Prefix」テキストフィールドで、このディレクトリに使用する URL プレフィックスを入力します。

   つまり、入力したテキストは、Windows CGI プログラムのディレクトリとして URL に表示されます。たとえば、URL プレフィックスとして「`wcgi-programs`」と入力すると、これらの Windows CGI プログラムへの URL は、次のようになります。

   http://*yourserver*.domain.com/wcgi-programs/program-name

注 指定した URL プレフィックスは、手順 5 で指定する実際の Windows CGI ディレクトリと同じである必要はありません。

4. スクリプトトレーシングを有効にするかどうかを選択します。

   「Enable Script Tracing?」の下の「Yes」または「No」のラジオボタンをクリックします。

   CGI パラメータは、ファイルを介して、サーバーから Windows CGI プログラムに渡されます。これらのファイルは、通常、Windows CGI プログラムの実行後にサーバーによって削除されます。スクリプトトレーシングを有効にする場合、これらのファイルは `/temp` ディレクトリ、または環境変数 TMP と TEMP の指定先に保存されます。また、スクリプトトレーシングが有効な場合には、Windows CGI プログラムによって立ち上げられたウィンドウが表示されます。

5. 「WinCGI Directory」テキストフィールドに、ディレクトリの場所を絶対パスで入力します。

   このディレクトリは、必ずしもドキュメントルートの下である必要はありません。このため、手順 3 で URL プレフィックスを指定する必要があります。

6. 「OK」をクリックします。

7. 変更を保存して適用します。

既存の Windows CGI ディレクトリを削除するには、「Windows CGI Directory」フォームで CGI ディレクトリの「Remove」ボタンをクリックします。既存のディレクトリの URL プレフィックスまたは Windows CGI ディレクトリを変更するには、ディレクトリの「Edit」ボタンをクリックします。

指定したディレクトリに Windows CGI プログラムをコピーします。このディレクトリ内のファイルはすべて Windows CGI ファイルとして処理されることに注意してください。

ファイルタイプとして Windows CGI を指定

Windows CGI ファイルのファイル拡張子を指定するには、次の手順を実行します。

1. サーバーマネージャの「Server Preferences」タブを選択します。

2. 「MIME Types」リンクをクリックします。

   「Global MIME Types」ウィンドウが表示されます。Global MIME Types についての詳細は、175 ページの「MIME タイプの選択」を参照してください。

3. 新しい MIME タイプを次の設定で追加します。

   - Type: `type`
   - Content type: `magnus-internal/wincgi`

- File Suffix: サーバーを関連付けたい Windows CGI のファイルサフィックスを入力します。CGI、WinCGI、およびシェル CGI ファイルタイプを有効にする場合、CGI のタイプごとに異なるサフィックスを指定する必要があります。たとえば、CGI プログラムとシェル CGI プログラムの両方にサフィックス .exe を使用することはできません。必要に応じて、サフィックスが一意になるように、同じページのほかの MIME タイプのフィールドを編集することができます。

4. 「New Type」ボタンをクリックします。
5. 変更を保存して適用します。

Windows でのシェル CGI プログラムのインストール

この節では、Windows でのシェル CGI プログラムのインストール方法について説明します。この節の内容は次のとおりです。

- Windows 向けシェル CGI プログラムの概要
- シェル CGI ディレクトリの指定 (Windows)
- ファイルタイプとしてシェル CGI を指定 (Windows)

Windows 向けシェル CGI プログラムの概要

シェル CGI は、Windows のファイル関連付けセットを使って CGI アプリケーションを実行するための、サーバー設定です。

たとえば、サーバーは `hello.pl` というシェル CGI ファイルに対する要求を受け取った場合、Windows のファイル関連付けを使用し、`.pl` という拡張子に関連付けられたプログラムを使用してファイルを実行します。`.pl` という拡張子がプログラム `C:¥bin¥perl.exe` と関連付けられている場合、サーバーは次のように `hello.pl` ファイルを実行します。

```
c:¥bin¥perl.exe hello.pl
```

もっとも簡単にシェル CGI を設定するには、サーバーのドキュメントルート内にシェル CGI ファイルだけが格納されるディレクトリを作成します。ただし、Sun ONE Web Server から MIME タイプを編集することによって、特定のファイル拡張子をシェル CGI に関連付けるようにサーバーを設定することもできます。

注	Windows でのファイル拡張子の設定方法については、Windows のマニュアルを参照してください。

シェル CGI ディレクトリの指定 (Windows)

シェル CGI ファイルのディレクトリを作成するには、次の手順を実行します。

1. コンピュータにシェルディレクトリを作成します。このディレクトリは、ドキュメントルートのサブディレクトリである必要はありません。
2. サーバーマネージャの「Class Manager」リンクを選択します。
3. 次にクラスマネージャを選択します。

   「shell CGI Directory」リンクが強調表示され、「CGI」ウィンドウが表示されます。
4. 「URL Prefix」フィールドで、シェル CGI ディレクトリに関連付ける URL プレフィックスを入力します。

   たとえば、すべてのシェル CGI ファイルを `C:¥docs¥programs¥cgi¥shell-cgi` ディレクトリに保存し、ユーザーに対してディレクトリを `http://www.`*`yourserver`*`.com/shell/` として表示させるとします。この場合、URL プレフィックスとして「`shell`」と入力します。
5. 「Shell CGI Directory」フィールドに、作成したディレクトリへの絶対パスを入力します。

警告 サーバーには、このディレクトリへの読み取りと実行の権限が必要です。Windows の場合は、サーバーを実行するユーザーアカウント (たとえば、`LocalSystem`) に、シェル CGI ディレクトリ内のプログラムに対する読み取りや実行の権限が必要です。

6. シェル CGI ディレクトリ内のファイルに対しても Windows のファイルとの関連付けが行われていることを確認します。ファイル拡張子の関連付けがないファイルを実行しようとすると、サーバーはエラーを返します。

ファイルタイプとしてシェル CGI を指定 (Windows)

Sun ONE Web Server の「MIME Types」ウィンドウを使用して、シェル CGI 機能にファイル拡張子を関連付けることができます。これは、Windows での関連付けの作成とは異なります。

サーバーのシェル CGI 機能とファイル拡張子を関連付ける場合、たとえば、.pl 拡張子の付いたファイルに対して関連付けを作成できます。サーバーは、この拡張子を持つファイルの要求を受けると、Windows でこのファイル拡張子に関連付けられた実行可能ファイルを起動して、そのファイルをシェル CGI ファイルとして処理します。

ファイル拡張子をシェル CGI ファイルとして関連付けるには、次の手順を実行します。

1. コンピュータにシェルディレクトリを作成します。このディレクトリは、ドキュメントルートのサブディレクトリである必要はありません。
2. サーバーマネージャの「Server Preferences」を選択します。
3. 「MIME Types」リンクをクリックします。

   「Global MIME Types」ウィンドウが表示されます。Global MIME Types についての詳細は、175 ページの「MIME タイプの選択」を参照してください。
4. 新しい MIME タイプを次の設定で追加します。
   - Type: `type`
   - Content type: `magnus-internal/shellcgi`
   - File Suffix: サーバーを関連付けたいシェル CGI のファイルサフィックスを入力します。CGI、WinCGI、およびシェル CGI ファイルタイプを有効にする場合、CGI のタイプごとに異なるサフィックスを指定する必要があります。たとえば、CGI プログラムとシェル CGI プログラムの両方にサフィックス .exe を使用することはできません。必要に応じて、サフィックスが一意になるように、同じページのほかの MIME タイプのフィールドを編集することができます。
5. 「New Type」ボタンをクリックします。
6. 変更を保存して適用します。

クエリハンドラの使用

注	クエリハンドラは現在、使用されなくなっています。Sun ONE Web Server と Netscape Navigator のクライアントでまだサポートされていますが、ほとんど使用されていません。HTML ページのフォームを使用してクエリを送信する方法がより一般的です。

デフォルトのクエリハンドラ CGI プログラムを指定できます。クエリハンドラは、HTML ファイル内の ISINDEX タグで送信されたテキストを処理します。

ISINDEX は、入力可能なテキストフィールドを HTML ページに作成する点で、フォームのテキストフィールドと似ています。ただし、フォームのテキストフィールドの情報とは異なり、「ISINDEX」ボックス内の情報は、ユーザーが Enter キーを押すとすぐに送信されます。デフォルトのクエリハンドラを指定する場合、入力された内容の送信先となるプログラムをサーバーに対して指定します。ISINDEX タグについての詳細は、HTML のリファレンスマニュアルを参照してください。

クエリハンドラを設定するには、次の手順を実行します。

1. クラスマネージャの「Programs」タブを選択します。
2. 「Query Handler」リンクをクリックします。

   「Query Handler」ウィンドウが表示されます。
3. 「Editing」ピッカーを使用して、デフォルトのクエリハンドラで設定したいリソースを選択します。

   ディレクトリを選択する場合、サーバーがそのディレクトリまたはディレクトリ内のファイルの URL を受信したときだけ指定したクエリハンドラが実行されます。
4. 「Default Query Handler」フィールドで、選択したリソースのデフォルトとして使用する CGI プログラムへの絶対パスを入力します。
5. 「OK」をクリックします。
6. 変更を保存して適用します。

第 16 章

コンテンツ管理

この章では、仮想サーバーのクラスと仮想サーバーについて、コンテンツを設定して管理する方法を説明します。

この章では、次の項目について説明します。

- プライマリドキュメントディレクトリの設定
- 追加ドキュメントディレクトリの設定
- ユーザー公開情報ディレクトリのカスタマイズ (UNIX/Linux)
- シンボリックリンクの制限 (UNIX/Linux)
- リモートファイル操作の有効化
- ドキュメントの設定
- URL 転送の設定
- エラー応答のカスタマイズ
- 文字セットの変更
- ドキュメントのフッターの変更
- htaccess の使用
- サーバーが解析する HTML の設定
- キャッシュ制御指令の設定
- 強固な暗号の使用
- コンテンツを圧縮するためのサーバー設定

プライマリドキュメントディレクトリの設定

プライマリドキュメントディレクトリ (ドキュメントルートとも呼ばれます) は、リモートクライアントで利用したいすべてのファイルを格納するための中央ディレクトリです。

クラスを追加する場合は、絶対パスでドキュメントディレクトリを指定します。この絶対パスの一部として変数を使用しない場合は、クラス内のすべての仮想サーバーに対するドキュメントルートがデフォルトで同じディレクトリになります。クラスマネージャでドキュメントルートを個別に変更することもできます。

別の方法として、クラスに対してパスを設定するとき、変数を使用することもできます。たとえば、`$id` 変数を使用して、クラス内のすべての仮想サーバーに対して、仮想サーバーの ID を名前に使用したディレクトリを作成することができます。クラスのドキュメントルートを *class_doc_root*/$id に設定することができます。このパスを使用すると、クラスのドキュメントディレクトリが `/sun/servers/docs/$id` の場合、そのクラスに属している仮想サーバー `vs1` のデフォルトのドキュメントディレクトリは `/sun/servers/docs/vs1` です。

ドキュメントディレクトリと、サーバーインスタンス、クラス、および仮想サーバーのレベルでのドキュメントディレクトリの使用方法の詳細については、319 ページの「ドキュメントルート」を参照してください。

プライマリドキュメントディレクトリを変更して別のパスや変数を使用するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「Primary Document Directory」をクリックします。
3. 仮想サーバーの横に、ディレクトリへの絶対パスや変数、または、パスと変数の組み合わせを入力します。

   ドキュメントルートの絶対パスの最後に $id 変数を付けると、デフォルトでは、仮想サーバーごとに、*class_doc_root*/*virtual_server_ID* というドキュメントルートが設定されます。たとえば、クラスのドキュメントディレクトリが `/sun/servers/docs/$id` の場合、そのクラスに属する仮想サーバー `vs1` のデフォルトのドキュメントディレクトリは、`/sun/servers/docs/vs1` になります。

   変数についての詳細は、324 ページの「変数の使用法」を参照してください。
4. 「OK」をクリックします。

詳細については、オンラインヘルプの「Primary Document Directory ページ」を参照してください。

注	通常、各仮想サーバーには固有のプライマリドキュメントディレクトリがあります。

追加ドキュメントディレクトリの設定

ほとんどの場合、仮想サーバー、またはサーバーインスタンスのドキュメントは、プライマリドキュメントディレクトリにあります。ただし、ドキュメントルート外のディレクトリからドキュメントを参照する場合もあります。追加ドキュメントディレクトリを設定すると、ドキュメントルート外のディレクトリからドキュメントを参照できます。ドキュメントルート外のドキュメントディレクトリを参照できるようにすることで、ほかのユーザーにプライマリドキュメントルートへアクセスすることなくドキュメントのグループを管理することを許可できます。

変数を使用しないで追加ドキュメントディレクトリを設定する場合は、そのディレクトリがクラスレベルで設定され、クラス内のすべての仮想サーバーによって使用されます。

クラス内の個々の仮想サーバーに対して追加ドキュメントディレクトリを設定する場合、URL プレフィックスがマッピングされるディレクトリは仮想サーバーごとに変わるため、変数を使用する必要があります。

追加ドキュメントディレクトリを追加するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「Additional Document Directories」をクリックします。
3. マッピングする URL プレフィックスを選択します。

   クライアントはドキュメントが必要なとき、この URL をサーバーに送信します。
4. URL をマッピングするディレクトリを指定します。
5. 必要に応じて、既存の設定スタイルを使用し、このディレクトリの設定方法を指定します。
6. 「OK」をクリックします。

詳細については、オンラインヘルプの「Additional Document Directories ページ」を参照してください。

デフォルトでは、サーバーインスタンスにいくつかの追加ドキュメントディレクトリがあります。そのディレクトリには、次のプレフィックスが付いています。

- `/manual`
- `/servlet`

これらのディレクトリへのアクセスを制限して、ユーザーが書き込めないようにする必要があります。ACL の例を次に示します。

```
deny (all) anyone;
allow (rxli) all;
allow (wd) privileged_user;
```

ユーザー公開情報ディレクトリのカスタマイズ (UNIX/Linux)

ユーザーが独自の Web ページを管理する場合もあります。サーバー上のすべてのユーザーが、自由にホームページやその他のドキュメントを作成できるように、公開情報ディレクトリを設定することができます。

この設定ができるのは、クラス全体を対象とする場合だけです。仮想サーバーごとにカスタマイズする方法はありません。

このシステムでは、クライアントは公開情報ディレクトリとしてサーバーに認識されている特定の URL を使用してサーバーにアクセスできます。たとえば、プレフィックス ~ とディレクトリ public_html を選択するとします。
http://www.sun.com/~jdoe/aboutjane.html が要求された場合、サーバーは ~jdoe がユーザーの公開情報ディレクトリを参照していると認識します。サーバーはシステムのユーザーデータベースの jdoe で Jane のホームディレクトリを検索します。次に、サーバーは ~/jdoe/public_html/aboutjane.html を検索します。

公開ディレクトリを使用するようにサーバーを設定するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「User Document Directories」をクリックします。
3. ユーザーの URL プレフィックスを選択します。

   チルド文字がユーザーのホームディレクトリにアクセスするための標準的な UNIX/Linux プレフィックスであるため、通常のプレフィックスは ~ です。
4. サーバーが HTML ファイルを検索する、ユーザーのホームディレクトリ内にあるサブディレクトリを選択します。

   通常のディレクトリは public_html です。
5. パスワードファイルを指定します。

   サーバーでは、システム上のユーザーのリストがあるファイルを検索する場所を認識している必要があります。サーバーはこのファイルを使用して、ユーザー名が有効であるかどうかを判断し、そのユーザーのホームディレクトリを検索します。この目的でシステムのパスワードファイルを使用する場合、サーバーは標準のライブラリコールを使用してユーザーを検索します。あるいは、別のユーザーファイルを作成して、ユーザーを検索することもできます。このユーザーファイルは絶対パスで指定することができます。

   ファイルの各行を次の構造にする必要があります (/etc/passwd ファイル内の必要のない要素は * で表示されています)。

   ```
   username:*:*:groupid:*:homedir:*
   ```

6. 起動時にパスワードデータベースを読み込むかどうかを選択します。

   詳細は、375 ページの「起動時のパスワードファイル全体の読み込み」を参照してください。

7. 設定スタイルを適用するかどうかを選択します。
8. 「OK」をクリックします。

詳細については、オンラインヘルプの「User Document Directories ページ」を参照してください。

ユーザーに個別のディレクトリを提供するもう 1 つの方法は、すべてのユーザーが修正できる中央ディレクトリへの URL マッピングを作成することです。

コンテンツ発行の制限

システム管理者が、ユーザードキュメントディレクトリからコンテンツを発行できるユーザーアカウントを制限したい場合もあります。ユーザーによる発行を制限するには、`/etc/passwd` ファイルのユーザーのホームディレクトリのパスの最後にスラッシュを追加します。

```
jdoe::1234:1234:John Doe:/home/jdoe:/bin/sh
```

を次のように修正します。

```
jdoe::1234:1234:John Doe:/home/jdoe/:/bin/sh
```

この修正を行うと、Sun ONE Web Server はこのユーザーのディレクトリのページを提供しなくなります。URI を要求するブラウザは「404 File Not Found」エラーを受信し、Web サーバーのアクセスログに 404 エラーが記録されます。エラーログにはエラーが記録されません。

この修正のあと、このユーザーにコンテンツの発行を許可する場合は、`/etc/passwd` エントリから最後のスラッシュを削除して、Web サーバーを再起動します。

起動時のパスワードファイル全体の読み込み

起動時にパスワードファイル全体を読み込むオプションもあります。このオプションを選択する場合、サーバーは起動時にパスワードファイルをメモリに読み込むため、ユーザーの検索がかなり速くなります。ただし、パスワードファイルが非常に大きい場合は、このオプションでメモリを使い過ぎる可能性があります。

設定スタイルの使用

サーバーに設定スタイルを適用して、公開情報ディレクトリからディレクトリへのアクセスを制御することができます。これによって、管理者が公開したくない情報にユーザーがシンボリックリンクを作成するのを防止できます。設定ファイルの詳細については、第 17 章「設定スタイルの適用」を参照してください。

リモートファイル操作の有効化

リモートファイル操作を有効にする場合、クライアントはファイルのアップロード、ファイルの削除、ディレクトリの作成、ディレクトリの削除、ディレクトリの中身のリスト表示、サーバー上のファイルの名前変更などを実行できます。ディレクトリ *server_root*/https-*serve-id*/config 内のファイル obj.conf には、リモートファイル操作を有効にした場合にアクティブになるコマンドが格納されています。これらのコマンドをアクティブにすると、リモートブラウザでサーバーのドキュメントを変更できるようになります。アクセス制御を使用して、これらのリソースへの書き込みを制限し、認証を受けていないユーザーによる変更を防止する必要があります。

リモートファイル操作を有効にしても、Microsoft Frontpage などのコンテンツ管理システムの使用に影響を及ぼすことはありません。

UNIX/Linux の場合 : ファイルへのアクセス権がないと、この機能は動作しません。つまり、ドキュメントルートユーザーをサーバーユーザーと同じにする必要があります。

リモートファイル操作を有効にするには、次の手順を実行します。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「Remote File Manipulation」をクリックします。
3. リモートファイル操作を有効にするオプションを選択します。
4. 「OK」をクリックします。

詳細については、オンラインヘルプの「Remote File Manipulation ページ」を参照してください。

ドキュメントの設定

「Document Preferences」ページを使用して、ドキュメント設定を行います。この節では、以下の項目について説明します。

- ドキュメント設定の変更
- インデックスファイル名の入力
- ディレクトリのインデックス作成の選択
- サーバーのホームページの指定
- デフォルト MIME タイプの指定

これらの設定はすべて、個々の仮想サーバーではなく、クラスに対して適用されます。

ドキュメント設定の変更

ドキュメント設定を変更するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「Document Preferences」をクリックします。
3. 次の節で説明するように、適切なフィールド値を選択します。
4. 「OK」をクリックします。

変更できる設定については、次の節で詳しく説明します。詳細は、オンラインヘルプの「Document Preferences ページ」を参照してください。

インデックスファイル名の入力

URL でドキュメント名が指定されていない場合は、自動的にインデックスファイルが表示されます。デフォルトのインデックスファイルは `index.html` と `home.html` です。複数のインデックスファイルが指定されている場合、どれか見つかるまでこのフィールドに表示される名前順に検索されます。たとえば、インデックスファイル名が `index.html` と `home.html` の場合、サーバーは `index.html` を検索し、見つからない場合は `home.html` を検索します。

ディレクトリのインデックス作成の選択

ほとんどの場合、ドキュメントディレクトリにはいくつかのサブディレクトリがあります。たとえば、products や people などの名前が付いたディレクトリがあるとします。多くの場合、クライアントがこのようなディレクトリの概要 (またはインデックス) にアクセスできると便利です。

サーバーは index.html または home.html という名前のインデックスファイルをディレクトリ内で検索して、ディレクトリのインデックスを作成します。index.html または home.html は、ディレクトリの中身の概要として作成し、管理するファイルです。詳細は、377 ページの「インデックスファイル名の入力」を参照してください。デフォルト名の 1 つを付けることによって、どのファイルでもディレクトリのインデックスファイルとして指定することができます。つまり、CGI が有効な場合には、CGI プログラムをインデックスとして使用することもできます。

インデックスファイルが見つからない場合、サーバーはドキュメントルート内のすべてのファイルをリスト表示するインデックスファイルを生成します。

警告 サーバーがファイアウォール外にある場合は、ディレクトリのインデックス作成を無効にして、ディレクトリ構造やファイル名にアクセスできないようにします。

サーバーのホームページの指定

エンドユーザーが最初にサーバーにアクセスしたときに表示されるファイルは、通常、ホームページと呼ばれます。通常、このファイルにはサーバーについての一般情報とほかのドキュメントへのリンクがあります。

デフォルトでは、サーバーは「Document Preferences」ページの「Index Filenames」フィールドで指定されているインデックスファイルを検索し、ホームページとして使用します。ただし、ホームページとして使用するファイルを指定することもできます。

デフォルト MIME タイプの指定

ドキュメントがクライアントに送信されるとき、クライアントがドキュメントを正しく表示できるように、ドキュメントのタイプを指定する部分を含めて送信されます。ただし、サーバーに対してドキュメントの拡張子が定義されていないために、サーバーがドキュメントのタイプを判断できない場合もあります。このような場合は、デフォルト値が送信されます。

デフォルトは通常、text/plain ですが、サーバーに格納されているもっとも一般的なタイプに設定する必要があります。次に一般的な MIME タイプの一部を示します。

- text/plain
- text/richtext
- image/jpeg
- application/x-tar
- application/x-gzip
- text/html
- image/tiff
- image/gif
- application/postscript
- audio/basic

URL 転送の設定

URL 転送を使用すると、ドキュメント要求を別のサーバーにリダイレクトできます。URL の転送またはリダイレクションは、サーバーがユーザーに URL を変更したこと (たとえば、ファイルを別のディレクトリまたはサーバーに移動した場合) を通知するための方法です。また、リダイレクションを使用して、あるサーバーのドキュメントをユーザーが要求した場合に、その要求をスムーズに別のサーバーのドキュメントに送信することができます。

たとえば、`http://www.sun.com/info/movies` をプレフィックス `film.sun.com` に転送する場合には、URL `http://www.sun.com/info/movies` が `http://film.sun.com/info/movies` にリダイレクトされます。

変数を使用して、ディレクトリを新しいディレクトリにマッピングすることができます。たとえば、`/new` を `/$docroot/new` にマッピングすることができます。マッピングによって、仮想サーバーのドキュメントルートに移動します。

変数についての詳細は、324 ページの「変数の使用法」を参照してください。

1 つのサブディレクトリ内のすべてのドキュメントに対する要求を特定の URL にリダイレクトする場合もあります。たとえば、あまりにも多くのトラフィックが生じるため、または、何らかの理由でドキュメントが公開されなくなったために、ディレクトリを移動する必要がある場合、ドキュメントに対する要求を、ドキュメントを利用できなくなった理由を説明するページに誘導することができます。たとえば、`/info/movies` のプレフィックスを `http://www.sun.com/explain.html` にリダイレクトするなどです。

URL 転送を設定するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「URL Forwarding」をクリックします。
3. リダイレクトする URL プレフィックスを入力し、リダイレクト先を別のプレフィックスにするか、またはスタティックな URL にするかを指定します。
4. 「OK」をクリックします。

詳細については、オンラインヘルプの「URL Forwarding ページ」を参照してください。

エラー応答のカスタマイズ

仮想サーバーでエラーが発生した場合にクライアントに詳細なメッセージを送信する、カスタムエラー応答を指定できます。送信するファイルまたは実行する CGI プログラムを指定できます。

たとえば、特定のディレクトリでエラーが発生した場合のサーバーの動作を変更することができます。クライアントがアクセス制御によって保護されているサーバーの一部に接続しようとする場合、アカウントの取得方法についての情報が記載されたエラーファイルを返すように設定できます。

カスタムエラー応答を有効にするには、エラー応答として送信する HTML ファイルまたは 実行する CGI プログラムを作成する必要があります。そのあとで、クラスマネージャで応答を有効にします。

カスタマイズされたエラー応答を有効にするには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「Error Responses」をクリックします。
3. リソースピッカーから「Entire Server」を選択し、クラス全体に対して変更を適用するか、特定の仮想サーバーに対するドキュメントルートまたは特定の仮想サーバー内の特定のディレクトリを指定します。
4. 変更するエラーコードごとに、エラー応答が含まれるファイルまたは CGI への絶対パスを指定します。
5. 「OK」をクリックします。

詳細については、オンラインヘルプの「Error Responses ページ」を参照してください。

文字セットの変更

ドキュメントの文字セットは、記述されている言語によってある程度決まります。1 つのドキュメント、ドキュメントのセット、またはディレクトリに対するクライアントのデフォルト文字セットの設定は、リソースを選択し、リソースに対する文字セットを入力することによって変更できます。

Netscape Navigator では、HTTP で MIME タイプの charset パラメータを使用して、文字セットを変更できます。サーバーが応答でこのパラメータを指定する場合、それに応じて Netscape Navigator の文字セットが変更されます。次に、その例を示します。

- `Content-Type:text/html;charset=iso-8859-1`
- `Content-Type:text/html;charset=iso-2022-jp`

Netscape Navigator で認識される次の charset 名は、RFC 1700 で指定されています (x- で始まる名前を除く)。

- us-ascii
- iso-2022-jp
- x-euc-jp
- iso-8859-1
- x-sjis
- x-mac-roman

さらに、us-ascii に対して次のエイリアスが認識されます。

- ansi_x3.4-1968
- ansi_x3.4-1986
- ascii
- us
- cp367
- iso-ir-6
- iso_646.irv:1991
- iso646-us
- ibm367

iso_8859-1 に対して、次のエイリアスが認識されます。

- latin1
- iso_8859-1:1987
- ibm819
- iso_8859-1
- iso-ir-100
- cp819

文字セットを変更するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「International Characters」をクリックします。
3. リソースピッカーから「Entire Server」を選択し、クラス全体に対して変更を適用するか、特定の仮想サーバーに対するドキュメントルートまたは特定の仮想サーバー内の特定のディレクトリを指定します。
4. サーバー全体またはその一部に対して文字セットを設定します。

   このフィールドを空白にしておくと、文字セットが「NONE」に設定されます。
5. 「OK」をクリックします。

詳細については、オンラインヘルプの「International Characters ページ」を参照してください。

ドキュメントのフッターの変更

サーバーの特定の部分にあるすべてのドキュメントに対して、フッターを指定できます。フッターには、最後に修正を行なった日時を含めることができます。このフッターは、CGI スクリプトの出力や解析される HTML (.shtml) ファイルを除くすべてのファイルに対して挿入できます。CGI スクリプトの出力または解析される HTML ファイルにドキュメントフッターを表示する必要がある場合は、別のファイルにフッターのテキストを入力し、1 行のコードまたは別のサーバーサイドインクルードを追加して、そのファイルをページの出力に追加します。

ドキュメントフッターを変更するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「Document Footer」をクリックします。
3. リソースピッカーから「Entire Server」を選択し、クラス全体に対して変更を適用するか、特定の仮想サーバーに対するドキュメントルートまたは特定の仮想サーバー内の特定のディレクトリを指定します。

   ディレクトリを選択する場合、ドキュメントフッターは、サーバーがそのディレクトリまたはディレクトリ内のファイルの URL を受信したときにだけ適用されます。
4. フッターを挿入するファイルのタイプを指定します。
5. データ書式を指定します。

6. フッターに表示するテキストを入力します。

   ドキュメントフッターに使用できる最大文字数は、765 文字です。ドキュメントの最終更新日を含みたい場合には、`:LASTMOD:` と入力します。

7. 「OK」をクリックします。

詳細は、オンラインヘルプの「Document Footer ページ」を参照してください。

htaccess の使用

htaccess の使用についての詳細は、218 ページの「.htaccess ファイルの使用」を参照してください。

シンボリックリンクの制限 (UNIX/Linux)

サーバーでのファイルシステムリンクの使用を制限することができます。ファイルシステムリンクは、ほかのディレクトリやファイルシステムに格納されているファイルへの参照です。参照によって、現在のディレクトリにあるかのようにリモートファイルにアクセスできるようになります。次の 2 つのタイプのファイルシステムリンクがあります。

- **ハードリンク** - ハードリンクは、同じデータブロックセットを参照する 2 つの実際のファイル名です。元のファイルとリンクは同一です。このため、別のファイルシステム上にハードリンクを作成することはできません。
- **シンボリック (ソフト) リンク** - シンボリックリンクは、データを含む元のファイルと、元のファイルを参照する別のファイルの 2 つのファイルで構成されます。シンボリックリンクは、ハードリンクより柔軟です。シンボリックリンクは、複数のファイルシステム間で使用でき、ディレクトリにリンクできます。

ハードリンクとシンボリックリンクについての詳細は、各 UNIX/Linux システムのマニュアルを参照してください。

ファイルシステムリンクは、プライマリドキュメントディレクトリ外にあるドキュメントへのポインタを簡単に作成するための方法で、誰でもリンクを作成できます。このため、ほかのユーザーが重要なファイル (たとえば、機密ドキュメントやシステムのパスワードファイル) へのポインタを作成する可能性が心配される場合もあるでしょう。

シンボリックリンクを制限するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「Symbolic Links」をクリックします。

3. リソースピッカーから「Entire Server」を選択し、クラス全体に対して変更を適用するか、特定の仮想サーバーに対するドキュメントルートまたは特定の仮想サーバー内の特定のディレクトリを指定します。
4. ソフトリンクまたはハードリンクのどちらか、あるいは、両方を有効にするかを選択し、開始ディレクトリを選択します。
5. 「OK」をクリックします。

詳細については、オンラインヘルプの「Symbolic Link ページ」を参照してください。

サーバーが解析する HTML の設定

HTML は通常、ディスク上に実際に存在している通りの状態でクライアントに送信されます。ただし、サーバーはドキュメントを送信する前に、HTML ファイル内にある特別なコマンドを検索できます (つまり、HTML を解析できます)。サーバーでこのようなファイルを解析し、要求に固有の情報またはファイルをドキュメントに挿入したい場合、HTML の解析を事前に有効にしておく必要があります。

HTML を解析するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「Parse HTML」をクリックします。
3. サーバーが HTML を解析するリソースを選択します。

   リソースピッカーから「Entire Server」を選択し、クラス全体に対して変更を適用するか、特定の仮想サーバーに対するドキュメントルートまたは特定の仮想サーバー内の特定のディレクトリを指定します。

   ディレクトリを選択する場合、サーバーはそのディレクトリまたはそのディレクトリ内のファイルの URL を受信したときにだけ HTML を解析します。
4. サーバーによる HTML の解析を有効にするかどうかを選択します。

   exec タグを有効にせずに HTML ファイルの解析を有効にすることも、exec タグも含めて HTML ファイルの解析を有効にすることもできます。exec タグを使用すると、HTML ファイルでサーバー上のほかのプログラムを実行できます。
5. 解析するファイルを選択します。

   .shtml という拡張子が付いているファイルだけを解析するか、すべての HTML ファイルを解析するかを選択できます。すべての HTML ファイルを解析する場合、パフォーマンスが低下します。UNIX/Linux を使用している場合、実行権限が有効な UNIX/Linux ファイルの解析を選択することもできます。ただし、この場合は信頼性が損なわれる可能性があります。
6. 「OK」をクリックします。

解析する HTML を受け入れるためのサーバーの設定についての詳細は、オンラインヘルプの「Parse HTML ページ」を参照してください。

サーバーによって解析された HTML の使用については、『Sun ONE Web Server 6.1 Programmer's Guide』を参照してください。

キャッシュ制御指令の設定

キャッシュ制御指令は、プロキシサーバーによってキャッシュに保存される情報を Sun ONE Web Server で制御するための手段です。キャッシュ制御指令を使用すると、プロキシのデフォルトで設定されているキャッシュに保存する機能が制限され、重要な情報がキャッシュに保存されてあとから取得される可能性がないように保護されます。このような指令を実行するには、プロキシサーバーが HTTP 1.1 に準拠している必要があります。

HTTP 1.1 の詳細については、Hypertext Transfer Protocol--HTTP/1.1 仕様 (RFC 2068) を参照してください。サイトは次のとおりです。

http://www.ietf.org/

キャッシュ制御指令を設定するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「Cache Control Directives」をクリックします。
3. 各フィールドに必要事項を入力します。応答指令として有効な値を次に示します。
   - Public: 応答は任意のキャッシュに保存されます。これがデフォルト設定です。
   - Private: 応答は公開されていない (共有ではない) キャッシュにだけ保存されます。
   - No Cache: 応答はどのキャッシュにも保存できません。
   - No Store: 不揮発性記憶装置にあるキャッシュに要求や応答を保存できません。
   - Must Revalidate: キャッシュのエントリは発信元サーバーから再検証される必要があります。
   - Maximum Age (sec): クライアントは、ここで設定された経過時間よりも長い経過時間がたった応答を受け入れません。
4. 「OK」をクリックします。

詳細については、オンラインヘルプの「Cache Control Directives ページ」を参照してください。

強固な暗号の使用

強固な暗号の設定についての詳細は、148 ページの「Stronger Ciphers の設定」を参照してください。

コンテンツを圧縮するためのサーバー設定

Sun ONE Web Server 6.1 は、HTTP コンテンツの圧縮をサポートしています。コンテンツを圧縮することで、ハードウェアに負担をかけることなくクライアントへの配信速度を向上し、コンテンツのボリュームを増やすことができます。コンテンツを圧縮すると、コンテンツのダウンロード時間が短縮されます。これは、ダイヤルアップ接続や、高トラフィックの接続を利用するユーザーにとって大きな利点となります。

コンテンツを圧縮した場合、Web サーバーは圧縮されたデータを送信し、そのデータを直ちに展開 (解凍) するようにブラウザに指示を出します。このため、送信するデータの容量が減り、ページの表示速度が速くなります。

圧縮されたデータの処理について、サーバーに次の 2 つの方法を設定できます。

- 事前に圧縮したコンテンツを配信するようにサーバーを設定する
- コンテンツをオンデマンドで圧縮するようにサーバーを設定する

サーバーの圧縮処理機能の拡張については、「圧縮に関連する obj.conf 内の変更」を参照してください。

事前に圧縮したコンテンツを配信するようにサーバーを設定する

ファイルの圧縮バージョンを事前に生成し、それを指定のディレクトリに格納するように Sun ONE Web Server を設定することができます。このように設定した場合、また、`Accept-encoding: gzip` ヘッダーを受信した場合に限り、圧縮済みコンテンツを保存するように設定されているディレクトリに格納されているファイルに対するすべての要求は、そのディレクトリに該当する圧縮済みファイルが実際に存在すれば、その圧縮済みファイルへの要求となります。たとえば、Web サーバーが `myfile.html` に対する要求を受信し、`myfile.html` と圧縮ファイル `myfile.html.gz` の両方が存在する場合、適切な `Accept-encoding` ヘッダーが含まれていればこれらの要求に対して、圧縮されたファイルが返されます。

事前に圧縮されたコンテンツを配信するようにサーバーを設定するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。

2. 「Serve Precompressed Content」をクリックします。
3. 次の情報を入力します。
   - Editing: 事前に圧縮されたコンテンツの配信元となるリソースをドロップダウンリストから選択します。ディレクトリを選択する場合、サーバーはそのディレクトリまたはそのディレクトリ内のファイルの URL を受信したときにだけ圧縮済みのコンテンツを提供します。

     「Browse」ボタンをクリックしてプライマリドキュメントディレクトリを参照するか、「Wildcard」ボタンをクリックしてワイルドカードパターンを指定します。ワイルドカードパターンの使用法については、「リソースピッカーで使用するワイルドカード」を参照してください。
   - Activate Serving Precompressed Content?: 選択したリソースの圧縮されたコンテンツを配信するようにサーバーを設定できます。
   - Check Age: 圧縮されたバージョンが圧縮されていないバージョンより古いかどうかの確認するかどうかを指定します。`yes` または `no` を指定できます。

     `yes` に設定すると、圧縮バージョンが圧縮されていないバージョンより古い場合に圧縮バージョンは選択されません。

     `no` に設定すると、圧縮バージョンが圧縮されていないバージョンより古い場合でも、常に圧縮バージョンが選択されます。

     デフォルトでは、`yes` に設定されています。
   - Vary Header: `Vary: Accept-encoding` ヘッダーを使用するかどうかを指定します。`yes` または `no` を選択します。

     `yes` に設定すると、ファイルの圧縮バージョンが選択された場合は常に `Vary: Accept-encoding` ヘッダーが挿入されます。

     `no` に設定すると、`Vary: Accept-encoding` ヘッダーは挿入されません。

     デフォルトでは、`yes` に設定されています。
4. 「OK」をクリックします。

コンテンツをオンデマンドで圧縮するようにサーバーを設定する

転送データを直ちに圧縮するように Sun ONE Web Server 6.1 を設定することもできます。ダイナミックに生成される HTML ページは、ユーザーがそれを要求するまで存在しません。これは、電子商取引を行う Web アプリケーションや、データベースを多用するサイトで特に便利です。

コンテンツをオンデマンドで圧縮するようにサーバーを設定するには、次の手順に従います。

1. クラスマネージャの「Content Mgmt」タブをクリックします。
2. 「Compress Content On Demand」をクリックします。
3. 次の情報を入力します。
   - Editing: オンデマンドで圧縮されるコンテンツをダイナミックに配信するリソースをドロップダウンリストから選択します。ディレクトリを選択する場合、サーバーはそのディレクトリまたはそのディレクトリ内のファイルの URL を受信したときにだけ圧縮したコンテンツを提供します。

     「Browse」ボタンをクリックしてプライマリドキュメントディレクトリを参照するか、「Wildcard」ボタンをクリックしてワイルドカードパターンを指定します。ワイルドカードパターンの使用法については、「リソースピッカーで使用するワイルドカード」を参照してください。
   - Activate Compress Content on Demand?: サーバーが、選択したリソースの圧縮コンテンツを配信するかどうかを選択します。
   - Vary Header: `Vary: Accept-encoding` ヘッダーを挿入するかどうかを指定します。yes または no を選択します。

     `yes` に設定すると、ファイルの圧縮バージョンが選択された場合は常に `Vary: Accept-encoding` ヘッダーが挿入されます。

     `no` に設定すると、`Vary: Accept-encoding` ヘッダーは挿入されません。

     デフォルトでは、`yes` に設定されています。
   - Fragment Size: 圧縮ライブラリ (zlib) が一度に圧縮する量を制御するために使用するメモリフラグメントのサイズをバイト単位で指定します。デフォルト値は `8096` です。
   - Compression Level: 圧縮のレベルを指定します。1 ～ 9 の値を選択します。値 1 では速度が最高になり、値 9 では圧縮率が最高になります。デフォルト値は、速度と圧縮率の両方を考慮した 6 です。
4. 「OK」をクリックします。

圧縮に関連する obj.conf 内の変更

サーバーで圧縮を有効にすると、obj.conf ファイルにエントリが追加されます。次に、このエントリの例を示します。

```
Output fn="insert-filter" filter="http-compression" type="text/*"
```

圧縮を特定タイプのドキュメントに限定するか、あるいは圧縮されたコンテンツにうまく対応できないブラウザを除外するときは、obj.conf ファイルを編集する必要があります。この処理の詳細については、『Sun ONE Web Server 6.1 NSAPI Programmer's Guide』を参照してください。

第 17 章

設定スタイルの適用

設定スタイルは、さまざまな仮想サーバーが維持管理する特定のファイルやディレクトリに対して、一連のオプションを簡単に適用する方法です。たとえば、アクセスロギングを設定する設定スタイルを作成することができます。ログに記録したいファイルやディレクトリにその設定スタイルを適用すれば、仮想サーバー内のすべてのファイルやディレクトリに対して、個別にアクセスロギングを設定する必要はありません。

この章には、次の節が記述されています。

- 設定スタイルの作成
- 設定スタイルの割り当て
- 設定スタイルの割り当ての一覧表示
- 設定スタイルの編集
- 設定スタイルの削除

設定スタイルの作成

設定スタイルを作成するには、次の手順を実行します。

1. クラスマネージャにアクセスします。
2. 「Styles」タブを選択します。
3. 「New Style」リンクをクリックします。
4. 設定スタイルに付ける名前を入力します。「OK」をクリックします。

   「Edit a Style」ページが表示されます。
5. ドロップダウンリストから、編集する設定スタイルを選択し、「Edit this Style」をクリックします。

6. 使用できるリンクのリストから、スタイルを設定するカテゴリをクリックします。

   表 17-1 に示される情報を設定できます。

7. 表示されるフォームに必要事項を入力し、「OK」をクリックします。

8. 設定スタイルのほかの設定を変更するには、手順 6 と 7 を繰り返します。「OK」をクリックします。

編集するスタイルを選択すると、リソースピッカーに、ほかのリソースではなく設定スタイルが表示されます。スタイルの編集が終了したら、「OK」をクリックし、その後、「Apply」をクリックします。リソースピッカーのスタイルモードが終了します。また、リソースピッカーから「Exit styles mode」を選択して、スタイルモードの終了を選択することもできます。リソースピッカーの詳細については、第 1 章「Sun ONE Web Server の概要」の 41 ページの「リソースピッカーの使用」を参照してください。

表 17-1 設定スタイルのカテゴリ

カテゴリ	説明
CGI file type (CGI のファイルタイプ)	ファイルタイプとして CGI を有効にします。CGI の詳細については、第 15 章「プログラムによるサーバーの拡張」の 359 ページに記載されている「CGI プログラムのインストール」を参照してください。
Character Set (文字セット)	リソースの文字セットを変更できます。文字セットの詳細については、第 16 章「コンテンツ管理」の 381 ページに記載されている「文字セットの変更」を参照してください。
Default Query Handler (デフォルトのクエリハンドラ)	サーバーリソースに対してデフォルトのクエリハンドラを設定できます。クエリハンドラの詳細については、第 15 章「プログラムによるサーバーの拡張」の 370 ページに記載されている「クエリハンドラの使用」を参照してください。
Document Footer (ドキュメントのフッター)	サーバーリソースにドキュメントフッターを追加できるようにします。詳細は、第 16 章「コンテンツ管理」の 382 ページの「ドキュメントのフッターの変更」を参照してください。
.htaccess Configuration (.htaccess 設定)	サーバーマネージャへのアクセス権を与えることなく、ほかのユーザーに設定オプションのサブセットを与えることができます。アクセス制御の詳細については、第 9 章「サーバーへのアクセス制御」を参照してください。
Require Stronger Security (より強力なセキュリティを要求)	鍵サイズ制限を指定したり、特定のファイルへのアクセスを拒否できます。
Error Responses (エラー応答)	サーバーでエラーが発生した場合にクライアントに表示されるエラー応答を、カスタマイズできます。

表 17-1　設定スタイルのカテゴリ (続き)

カテゴリ	説明
Log preferences (ログ設定)	アクセスログの詳細を設定できます。ログ設定の詳細については、第 10 章「ログファイルの使用」の 245 ページに記載されている「アクセスログの詳細設定」を参照してください。
Remote File Manipulation (リモートファイル操作)	リモートブラウザからサーバーのドキュメントを変更できるようにするためのファイル操作コマンドを有効にできます。詳細は、第 16 章「コンテンツ管理」の 376 ページの「リモートファイル操作の有効化」を参照してください。
Server Parsed HTML (サーバーが解析する HTML)	ファイルがクライアントに送信される前に、サーバーが解析するかどうかを指定できます。詳細については、『Sun ONE Web Server 6.1 Programmer's Guide』を参照してください。
Serve Precompressed Content (圧縮済みコンテンツの提供)	サーバーが、ファイルの圧縮済みのバージョンを送信するかどうかを指定できます。詳細は、第 16 章「コンテンツ管理」の 386 ページの「事前に圧縮したコンテンツを配信するようにサーバーを設定する」を参照してください。
Compress Content on Demand (オンデマンドでのコンテンツの圧縮)	コンテンツを、サーバーがダイナミックに圧縮してからクライアントに送信するかどうかを指定できます。詳細は、第 16 章「コンテンツ管理」の 388 ページの「コンテンツをオンデマンドで圧縮するようにサーバーを設定する」を参照してください。
Symbolic links (UNIX/Linux)	サーバーでのファイルシステムリンクの使用を制限できます。詳細は、第 16 章「コンテンツ管理」の 383 ページの「シンボリックリンクの制限 (UNIX/Linux)」を参照してください。

詳細は、オンラインヘルプの「New Style ページ」を参照してください。

設定スタイルの割り当て

設定スタイルを作成すると、仮想サーバー内のファイルやディレクトリに割り当てることができます。個々のファイルやディレクトリを指定することも、ワイルドカードパターン (*.gif など) を指定することもできます。

設定スタイルを割り当てるには、次の手順を実行します。

1. クラスマネージャにアクセスします。
2. 「Styles」タブを選択します。
3. 「Assign Style」リンクをクリックします。
4. この設定スタイルを適用する URL のプレフィックスを入力します。

   ドキュメントルート内のディレクトリを選択する場合は、ドキュメントルートの後ろのパスだけを入力します。ディレクトリの後に /* を入力すると、設定スタイルがそのディレクトリ内のすべてのコンテンツに適用されます。
5. 適用する設定スタイルを選択します。

   以前にリソースに適用した設定スタイルを削除するには、「None」設定スタイルを適用します。「OK」をクリックします。

詳細は、オンラインヘルプの「Assign a Style ページ」を参照してください。

設定スタイルの割り当ての一覧表示

設定スタイルを作成し、ファイルやディレクトリに適用すると、設定スタイルとその適用先の一覧を表示できます。

設定スタイルの割り当てを一覧表示するには、次の手順を実行します。

1. クラスマネージャにアクセスします。
2. 「Styles」タブを選択します。
3. 「List Assignments」リンクをクリックします。

   サーバーリソースに適用された設定スタイルを示す「List Assignments」ページが表示されます。
4. 設定スタイルの割り当てを編集するには、設定スタイル名の隣にある「Edit」リンクをクリックします。

詳細は、オンラインヘルプの「List Assignments ページ」を参照してください。

設定スタイルの編集

設定スタイルを編集するには、次の手順を実行します。

1. クラスマネージャにアクセスします。
2. 「Styles」タブを選択します。
3. 「Edit Style」リンクをクリックします。
4. 編集する設定スタイルを選択し、「Edit this style」ボタンをクリックします。
5. 使用できるリンクのリストから、スタイルを設定するカテゴリをクリックします。

   カテゴリの詳細については、391 ページの「設定スタイルの作成」を参照してください。
6. 表示されるフォームに必要事項を入力し、「OK」をクリックします。
7. 設定スタイルのほかの変更を行うには、手順 5 と 6 を繰り返します。「OK」をクリックします。

編集するスタイルを選択すると、リソースピッカーに、ほかのリソースではなく設定スタイルが表示されます。スタイルの編集が終了したら、「OK」をクリックし、その後、「Apply」をクリックします。リソースピッカーのスタイルモードが終了します。また、リソースピッカーから「Exit styles mode」を選択して、スタイルモードの終了を選択することもできます。リソースピッカーの詳細については、第 1 章「Sun ONE Web Server の概要」の 41 ページの「リソースピッカーの使用」を参照してください。

詳細は、オンラインヘルプの「Edit Style ページ」を参照してください。

設定スタイルの削除

設定スタイルを削除する前に、設定スタイルが適用された割り当てを削除します。設定スタイルを削除する前にこの処理を行わない場合は、仮想サーバーのクラスの `obj.conf` ファイルを手動で編集し、ファイル内の設定スタイルを検索し、それを `None` に置換する必要があります。この検索と置換を行わないと、適用されていた設定スタイルが削除されたファイルやディレクトリにアクセスしたときに、サーバーの設定が間違っているというエラーメッセージが出力されます。

設定スタイルを削除するには、次の手順を実行します。

1. クラスマネージャにアクセスします。
2. 「Styles」タブを選択します。
3. 「List Assignments」リンクをクリックします。
4. 削除する「Edit Style Assignment」を選択します。
5. 「Remove this assignment」をクリックします。

詳細は、オンラインヘルプの「Remove Style ページ」を参照してください。

第 18 章

検索機能の使い方

Sun ONE Web Server 6.1 には、ユーザーがサーバー上のドキュメントを検索し、結果を Web ページに表示できる検索機能が用意されています。サーバー管理者はドキュメントのインデックスを作成し、ユーザーはこのインデックス (コレクションと呼びます) に対して検索を行います。また、サーバー管理者はユーザーのニーズに合わせて検索インタフェースをカスタマイズすることができます。

この章には、次の節が記述されています。

- 検索について
- 仮想サーバーでの検索アプリケーションの有効化
- 仮想サーバーでの検索アプリケーションの無効化
- 検索コレクションについて
- 検索の実行
- 検索ページ
- クエリの作成
- 詳細検索
- 検索結果の表示
- 検索ページのカスタマイズ

検索について

Sun ONE Web Server のインストール時に、他の Web コンポーネントとともに検索機能もインストールされます。Sun ONE Web Server 6.0 の場合と同様に、検索機能はサーバーインスタンスレベルではなく、仮想サーバーレベルで設定、管理されます。

各仮想サーバーでの検索機能の設定には、仮想サーバーマネージャの「Search」タブを使用します。このタブでは、次の処理を実行できます。

- 検索機能を有効化および無効化する
- 検索コレクションを作成、変更、削除し、インデックスを再作成する
- 検索コレクションの予定保守タスクを作成、変更、消去する

管理インタフェースで取得された情報は、VS 要素内でマッピングされている *<server-root>*`/config/server.xml` ファイルに格納されます。

サーバー管理者は、検索クエリページと検索結果ページをカスタマイズできます。これには、企業ロゴによるページの商標変更や、検索結果ページの表示方法の変更などが含まれます。従来のリリースでは、パターンファイルを使ってこれらの変更を行なっていました。Sun ONE Web Server 6.1 では、パターンファイルはサポートされていません。その代わりに、製品に含まれる JSP タグライブラリセットを使ってカスタマイズを行います。これらのライブラリは、パターンファイルと同様の機能を提供します。検索インタフェースのカスタマイズについては、「検索ページのカスタマイズ」を参照してください。

従来のリリースとは異なり、検索機能をグローバルに「オン」、「オフ」することはできません。その代わりにデフォルトの検索 Web アプリケーションが提供され、特定の仮想サーバーで検索機能を有効化または無効化します。この検索アプリケーションは、コレクションをクエリ送信し、結果を表示するための基本 Web ページを提供します。検索アプリケーションには、検索タグライブラリを使用して検索インタフェースをカスタマイズする方法を示すサンプル JSP が含まれます。

警告 Sun ONE Web Server 6.0 とは異なり、バージョン 6.1 は検索結果に対してアクセスチェックを行いません。セキュリティモデルおよびレルムの数には限りがないため、検索アプリケーション内でセキュリティチェックを行い、結果をフィルタリングすることは不可能です。コンテンツを保護するために適切なセキュリティメカニズムを導入することは、サーバー管理者の責任となります。

Sun ONE Web Server 6.1 は、複数ドキュメント検索をサポートしています。形式が異なるドキュメント (HTML、ASCII、PDF など) のインデックスを作成し、それに対して検索を行うことができます。

注 Sun ONE Web Server 6.1 は、Linux プラットフォームでは複数ドキュメント形式の検索をサポートしていません。

Sun ONE Web Server 6.1 では、従来のリリースで使用していた検索エンジンではなく、新しい検索エンジンが採用されました。このため、従来のリリースの Web Server から Sun ONE Web Server 6.1 に移行しても、既存の検索コレクションとインデックスは移行されません。

仮想サーバーでの検索アプリケーションの有効化

仮想サーバーで検索を有効化するには、Sun ONE Web Server に含まれる検索アプリケーションを有効にします。検索の有効化には、管理インタフェースを使用します。

注 検索を有効にするには、Java Web コンテナが有効になっている必要があります。

設定する仮想サーバーを含む仮想サーバークラスで Java が有効であることを確認したら、次の手順を実行して検索を有効にします。

1. 検索を有効にする仮想サーバーを選択し、「Manage」ボタンをクリックします。
2. 「Search」タブを選択し、「Search Configuration」リンクをクリックします。
3. 次の情報を入力します。
   - Max Hits: 検索クエリで検索される最大結果数を指定します。
   - URI: カスタム検索アプリケーションを使用する場合は、ここに URI を指定します。デフォルトの検索アプリケーションを使用する場合は、値を指定する必要はありません。
   - Path: カスタム検索アプリケーションを使用する場合は、ここにパスを指定します。デフォルトの検索アプリケーションを使用する場合は、値を指定する必要はありません。
   - Enabled: デフォルトの検索アプリケーションを有効にするには、このチェックボックスを選択します。
4. 「OK」をクリックします。

仮想サーバーでの検索アプリケーションの無効化

仮想サーバーで検索を無効にするには、Sun ONE Web Server に含まれる検索アプリケーションを無効にします。検索の無効化には、管理インタフェースを使用します。

仮想サーバーで検索を無効にするには、次の手順を実行します。

1. 検索を無効にする仮想サーバーを選択し、「Manage」ボタンをクリックします。
2. 「Search」タブを選択し、「Search Configuration」リンクをクリックします。
3. 「Enabled」チェックボックスの選択を解除します。
4. 「OK」をクリックします。

検索コレクションについて

検索を行うには、ユーザーが検索する検索可能なデータのデータベースが必要です。このデータベースはコレクションと呼ばれ、サーバー管理者によって作成されます。コレクションには、サーバー上のドキュメントのインデックスを作成し、ドキュメントに関する情報を格納します。サーバー管理者がサーバー上のすべてまたは一部のドキュメントのインデックスを作成すると、タイトル、作成日、作成者などの情報を検索で利用できるようになります。

コレクションについては、次の点に留意してください。

- コレクションは、管理する仮想サーバーに固有です。
- 仮想サーバーで表示可能なドキュメントだけが管理インタフェースに表示され、インデックス作成の対象となります。
- サーバーに作成できるコレクションの数には制限はありません。
- 1 つの検索コレクションに含めることができるファイルは、ファイルシステム上の 1 つの親ディレクトリの下に格納されているファイルだけです。
- 形式が異なるドキュメント (HTML、ASCII、PDF など) のインデックスを作成し、それに対して検索を行うことができます。
- 検索コレクションに含まれるドキュメントは、1 つの文字エンコーディングに限定されません。つまり、検索コレクションに複数のエンコーディングを関連付けることができます。
- コレクションに関する情報は、`server.xml` ファイルの VS 要素に格納されます。

この節では、次の内容について説明します。

- コレクションの作成
- コレクションの設定

- コレクションの更新
- コレクションの削除
- コレクションの保守
- コレクションのインデックス再作成
- コレクションの保守スケジュールの追加
- コレクション保守スケジュールの編集
- コレクションの保守スケジュールの削除

コレクションの作成

コレクションの作成と管理は、管理インタフェースから実行します。新しいコレクションを作成するときは、インデックスを作成するドキュメントを指定します。

新規コレクションを作成するには、次の手順を実行します。

1. コレクションを作成する仮想サーバーを選択し、「Manage」ボタンをクリックします。
2. 「Search」タブを選択し、「Create Collection」リンクをクリックします。
3. 次の情報を入力します。
    - Directory to Index: ドキュメントのインデックスをコレクションに追加するディレクトリをドロップダウンリストから選択します。この仮想サーバーで表示できるディレクトリだけがリスト表示されます。

      ディレクトリのコンテンツを表示するときは、「View」をクリックします。選択したディレクトリがサブディレクトリの場合、コンテンツは、「View *directory_name*」ページに表示されます。インデックスを作成するディレクトリを選択するには、「Index」をクリックします。ディレクトリを表示するには、フォルダをクリックします。

      インデックスを作成できるディレクトリのリストにディレクトリを追加するには、最初にドキュメントディレクトリを追加作成する必要があります。詳細は、「追加ドキュメントディレクトリの設定」を参照してください。
    - Collection Name: コレクション名を入力します。
    - Display Name: (省略可能) この名前は、検索クエリページにコレクション名として表示されます。表示名を指定しない場合は、コレクション名が表示されます。
    - Description: (省略可能) 新しいコレクションを説明するテキストを入力します。
    - Include Subdirectories?: 「No」を選択した場合、選択したディレクトリのサブディレクトリ内のドキュメントにインデックスは作成されません。デフォルトは「Yes」です。

- Pattern: ワイルドカードを指定して、インデックスを作成するファイルを選択します。ワイルドカードの詳細は、「リソースピッカーで使用するワイルドカード」を参照してください。

警告 特定のファイルだけにインデックスが作成されるようにワイルドカードパターンを慎重に使用してください。たとえば、*.* と指定すると、実行可能ファイルや Perl スクリプトのインデックスまで作成されてしまう場合があります。

- Default Encoding: インデックスを作成するドキュメントの文字エンコーディングを指定します。デフォルトは ISO-8859-1 です。インデックス作成エンジンは、埋め込まれているメタタグから HTML ドキュメントのエンコーディングを判断しようとします。エンコーディングが指定されていない場合、デフォルトのエンコーディングが使用されます。

  コレクション内のドキュメントの言語とエンコーディングは 1 つに制限されません。ドキュメントを追加するときに指定できるエンコーディングは 1 つだけですが、同じコレクションに次にドキュメントを追加するときは、異なるデフォルトエンコーディングを選択できます。

4. 「OK」をクリックします。

これにより、指定した名前の新しいコレクションが次の場所に作成されます。

<instance-root>/collections/*<vs-id>*/*<collection-name>*

また、適切な SEARCHCOLLECTION エントリが server.xml ファイルに作成されます。

コレクションの設定

コレクションを作成した後に、一部の設定を変更することができます。これらの設定は、server.xml ファイルに格納されます。コレクションを再設定すると、server.xml ファイルが更新され、変更内容が反映されます。

コレクションの設定に不必要な変更を加えることは避けてください。

既存のコレクションを再設定するには、次の手順を実行します。

1. 設定するコレクションを含む仮想サーバーを選択し、「Manage」ボタンをクリックします。
2. 「Search」タブを選択し、「Configure Collection」リンクをクリックします。
3. 設定するコレクションを「Collection」ドロップダウンリストから選択し、「Go」をクリックします。
4. 選択したコレクションについて、次の情報を編集できます。

- Display Name: (省略可能) この名前は、検索クエリページに新しいコレクション名として表示されます。
- Description: (省略可能) コレクションの説明テキストを編集します。
- Document URI: 検索コレクションのドキュメントルート URI を編集します。

注	「Additional Document Directories」ページでドキュメントルートの URI マッピングを変更していない限り、Document URI は変更しないでください。詳細は、「追加ドキュメントディレクトリの設定」を参照してください。

- Enabled: 有効化するときは「Yes」を選択します。「No」を選択すると、検索クエリページにコレクションが表示されなくなります。

5. 「OK」をクリックします。

これによりコレクションが再設定され、`server.xml` ファイルの適切な `SEARCHCOLLECTION` エントリが変更されます。

コレクションの更新

コレクションの作成後に、ファイルを追加または消去することができます。ドキュメントは、コレクションの作成時に指定したディレクトリの下からしか追加できません。ドキュメントを消去しても、そのファイルのエントリとそのメタデータだけがコレクションから削除されます。実際のファイル自体は、ファイルシステムから削除されません。

コレクションを更新するには、次の手順を実行します。

1. 更新するコレクションを含む仮想サーバーを選択し、「Manage」ボタンをクリックします。
2. 「Search」タブを選択し、「Update Collection」リンクをクリックします。
3. 更新するコレクションを「Collection」ドロップダウンリストから選択します。
4. 選択したコレクションについて、次の情報を更新できます。
   - Include subdirectories?: 「No」を選択した場合、選択したディレクトリのサブディレクトリ内のドキュメントにインデックスは作成されません。デフォルトは「Yes」です。

注	Include Subdirectories?: ドキュメントの追加だけに適用されます。

- Pattern: ワイルドカードを指定してインデックスを作成するか、コレクションから消去するファイルを選択します。ワイルドカードの詳細は、「リソースピッカーで使用するワイルドカード」を参照してください。

警告 ドキュメントを追加するときは、特定のファイルのインデックスだけが作成されるようにワイルドカードパターンを慎重に使用してください。たとえば、*.* と指定すると、実行可能ファイルや Perl スクリプトのインデックスまで作成されてしまう場合があります。

- Default Encoding: インデックスを作成するドキュメントの文字エンコーディングを指定します。デフォルトは ISO-8859-1 です。インデックス作成エンジンは、埋め込まれているメタタグから HTML ドキュメントのエンコーディングを判断しようとします。エンコーディングが指定されていない場合、デフォルトのエンコーディングが使用されます。

  コレクション内のドキュメントの言語とエンコーディングは 1 つに制限されません。ドキュメントを追加するときに指定できるエンコーディングは 1 つだけですが、同じコレクションに次にドキュメントを追加するときは、異なるデフォルトエンコーディングを選択できます。

5. インデックスを作成するドキュメントを追加するときは「Add Documents」をクリックします。適切なインデックスエントリを消去するときは、「Remove Documents」をクリックします。

注 追加できるドキュメントは、コレクションの作成時に指定したディレクトリに含まれるドキュメントだけです。

コレクションの削除

作成したコレクションを削除することができます。コレクションが削除されると、検索クエリページでユーザーが参照できなくなり、このコレクションに関連付けられたすべての設定ファイルとインデックスファイルが削除されます。コレクションを構成していた実際のドキュメントはファイルシステムから削除されず、コレクションに含まれるインデックスエントリだけが削除されます。

コレクションを削除するには、次の手順を実行します。

1. 削除するコレクションを含む仮想サーバーを選択し、「Manage」ボタンをクリックします。
2. 「Search」タブを選択し、「Maintain Collection」リンクをクリックします。
3. 削除するコレクションを「Collection」ドロップダウンリストから選択します。

4. 「Remove Collection」ボタンをクリックします。

注	コレクションを削除すると、コレクションの保守スケジュールも削除されます。保守スケジュールの詳細については、「コレクションの保守スケジュールの追加」を参照してください。

注	ローカルファイルマネージャを使用してコレクションを削除しないでください。この方法で削除した場合、対応する設定ファイルが更新されません。

コレクションの保守

定期的に、コレクションの保守が必要になる場合もあります。コレクションのインデックス作成や更新を頻繁に行わない限り、これらの保守タスクは必要ないこともあります。次のタスクを実行できます。

- コレクションのインデックス再作成
- コレクションの更新

コレクションのインデックス再作成

作成したコレクションのインデックスを再作成することができます。コレクションの作成後にドキュメントが変更されると、コレクションのインデックスは再作成されます。コレクションのインデックス再作成では、コレクション内の新しいコンテンツのインデックスは作成されず、コレクションの既存のコンテンツが更新されるだけです。サーバーのファイルシステムに存在しなくなったドキュメントのインデックスエントリが存在する場合は、そのエントリは消去されます。

コレクションのインデックスを再作成するには、次の手順を実行します。

1. インデックスを再作成するコレクションを含む仮想サーバーを選択し、「Manage」ボタンをクリックします。
2. 「Search」タブを選択し、「Maintain Collection」リンクをクリックします。
3. インデックスを再作成するコレクションを「Collection」ドロップダウンリストから選択します。
4. 「Reindex」ボタンをクリックします。

コレクションの保守スケジュールの追加

コレクションで定期的に実行される保守タスクのスケジュールを設定することができます。スケジュールを設定できるタスクは、インデックス再作成と更新です。特定のコレクションにタスクのスケジュールを設定するときは、管理インタフェースを使用します。次の情報を指定できます。

- 実行するタスク (インデックス再作成または更新)
- タスクを実行する時刻
- タスクを実行する曜日

コレクションの定期保守を追加するには、次の手順を実行します。

1. 保守のスケジュールを設定するコレクションを選択し、「Add Scheduled Maintenance」リンクをクリックします。
2. 次の情報を入力します。
    - Task: 自動化するタスクを選択します。インデックス再作成または更新を選択できます。

      「Update」を選択したときは、次の情報を指定する必要があります。

        - Recurse Subdirectories?: 「No」を選択した場合、選択したディレクトリのサブディレクトリ内のドキュメントにインデックスは作成されません。デフォルトは「Yes」です。
    - Pattern: ワイルドカードを指定して、インデックスを作成するファイルを選択します。ワイルドカードの詳細は、「リソースピッカーで使用するワイルドカード」を参照してください。

警告 特定のファイルだけにインデックスが作成されるようにワイルドカードパターンを慎重に使用してください。たとえば、*.* と指定すると、実行可能ファイルや Perl スクリプトのインデックスまで作成されてしまう場合があります。

    - Default Encoding: インデックスを作成するドキュメントの文字エンコーディングを指定します。デフォルトは ISO-8859-1 です。インデックス作成エンジンは、埋め込まれているメタタグから HTML ドキュメントのエンコーディングを判断しようとします。エンコーディングが指定されていない場合、デフォルトのエンコーディングが使用されます。

      コレクション内のドキュメントの言語とエンコーディングは 1 つに制限されません。ドキュメントを追加するときに指定できるエンコーディングは 1 つだけですが、同じコレクションに次にドキュメントを追加するときは、異なるデフォルトエンコーディングを選択できます。

- Scheduled Time: (必須) 予定保守タスクを実行する時刻を HH:MM 形式で指定します。たとえば、コレクションに含まれるドキュメントの変更が完了している可能性が高い、一日の終わりに保守の実行をスケジュールすることができます。
- Schedule day(s) of week: (必須) 1 つまたは複数のチェックボックスにチェックマークを付けて、予定保守タスクを実行する曜日を指定します。

3. 「OK」をクリックします。

注 UNIX/Linux ユーザーは、保守スケジュールを追加した後に、変更を適用するために cron 制御プロセスを再起動する必要があります。

コレクション保守スケジュールの編集

要件が変更された場合は、新しい要件に合わせてコレクションの保守スケジュールのプロパティを変更できます。たとえば、サイトが更新される可能性が高い時間帯を念頭に置いて保守スケジュールを変更してください。

コレクションの保守スケジュールを変更するには、次の手順を実行します。

1. 保守スケジュールを変更するコレクションを「Collection」ドロップダウンリストから選択します。
2. 再設定するタスクを選択し、必要な情報を入力します。詳細は、オンラインヘルプの「Edit Scheduled Collection ページ」を参照してください。
3. 「OK」をクリックします。

注 コレクションを削除すると、コレクションの保守スケジュールも削除されます。

注 UNIX/Linux ユーザーは、保守スケジュールを変更した後に、変更を適用するために cron 制御プロセスを再起動する必要があります。

コレクションの保守スケジュールの削除

コレクションの保守スケジュールが不要になった場合は、そのスケジュールを取消すことができます。

コレクションの保守スケジュールを取消すには、次の手順を実行します。

1. 保守スケジュールを削除するコレクションを「Collection」ドロップダウンリストから選択します。
2. 保守スケジュールを削除するタスク (「Reindex」または「Update」) を選択します。タスクがスケジュールされている場合は、その詳細が表示されます。
3. 更新タスクでは、削除するタスクの隣の「Delete」チェックボックスを選択します。
4. 「OK」をクリックします。

注	UNIX/Linux ユーザーは、保守スケジュールを削除した後に、変更を適用するために cron 制御プロセスを再起動する必要があります。

検索の実行

ユーザーが主に期待するのは、検索コレクション内でデータを照会し、ドキュメントのリストを取得することです。Sun ONE Web Server とともにインストールされる検索 Web アプリケーションには、デフォルトの検索クエリページと検索結果ページが用意されています。これらのページをそのまま使用することも、「検索ページのカスタマイズ」で説明する JSP タグセットを使用してカスタマイズすることもできます。

ユーザーは、サーバー管理者が作成したコレクションに対して検索を行います。次の検索を実行できます。

- キーワードのセットとオプションのクエリ演算子を入力して検索を実行する
- 仮想サーバーで表示できるコレクションだけを検索する
- 単一コレクション、または仮想サーバーで表示できるコレクションセットを対象に検索を実行する

サーバー管理者は、仮想サーバーの検索クエリページにアクセスするための URL をユーザーに提供する必要があります。

警告 Sun ONE Web Server 6.0 とは異なり、バージョン 6.1 は検索結果に対してアクセスチェックを行いません。セキュリティモデルおよびレルムの数には限りがないため、検索アプリケーション内でセキュリティチェックを行い、結果をフィルタリングすることは不可能です。コンテンツを保護するために適切なセキュリティメカニズムを導入することは、サーバー管理者の責任となります。

検索ページ

検索機能を使用するためにエンドユーザーがアクセスするデフォルトの URL は次のとおりです。

`http://<server-instance>:port number/search`

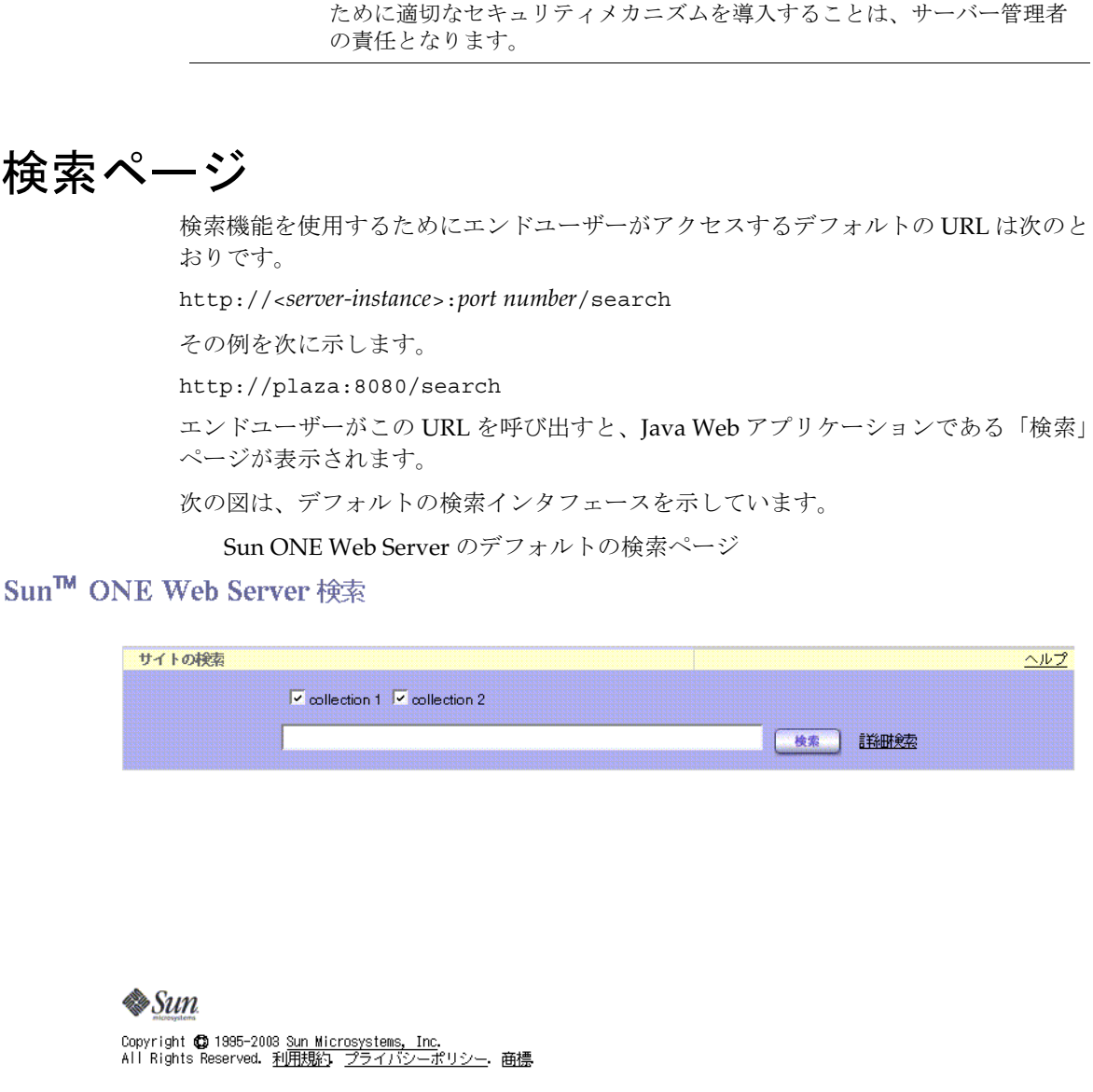
その例を次に示します。

`http://plaza:8080/search`

エンドユーザーがこの URL を呼び出すと、Java Web アプリケーションである「検索」ページが表示されます。

次の図は、デフォルトの検索インタフェースを示しています。

Sun ONE Web Server のデフォルトの検索ページ



「検索ページのカスタマイズ」で説明する JSP タグセットを使用して、このページをカスタマイズすることができます。

クエリの作成

検索クエリページは、コレクションに対する検索に使用されます。ユーザーは、キーワードのセットとオプションのクエリ演算子を入力し、ブラウザに表示される Web ページで結果を受信します。この結果ページには、検索条件を満たすサーバー上のドキュメントへのリンクが含まれます。

注	サーバー管理者は、「検索ページのカスタマイズ」で説明する方法で、検索クエリページをカスタマイズすることができます。

クエリを作成するには、次の手順を実行します。

1. ブラウザのアドレスバーに次の形式で URL を入力し、検索 Web アプリケーションにアクセスします。

   `http://<`*server-instance*`>:`*port number*`/search`

2. 表示される検索クエリページの検索対象で、検索するコレクションのチェックボックスを選択します。
3. クエリを説明する単語をいくつか入力し、Enter キーを押すと (または「検索」ボタンをクリックすると)、関連する Web ページがリスト表示されます。

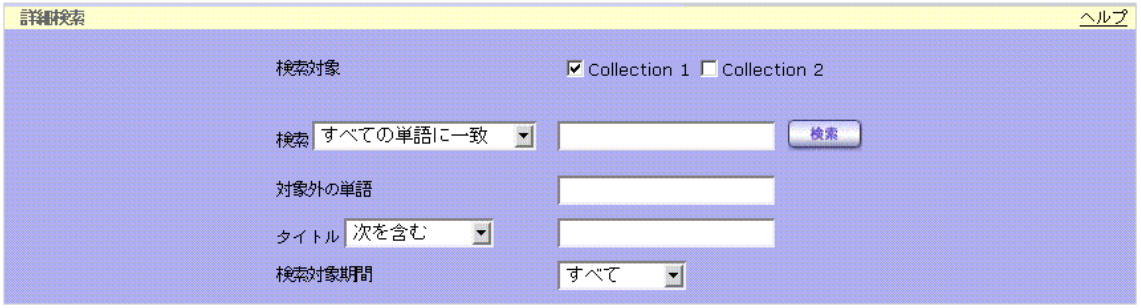
さらに詳細な検索を行うには、次の節で説明する「詳細検索」ページの検索パラメータを使用します。

詳細検索

ユーザーは、キーワードに演算子を追加して検索条件を限定することで、検索の精度を上げることができます。これらのオプションは、「詳細検索」ページで選択できます。

次の図は、詳細検索のページを示しています。

「詳細検索」ページ



詳細な検索クエリを作成するには、次の手順を実行します。

1. ブラウザのアドレスバーに次の形式で URL を入力し、検索 Web アプリケーションにアクセスします。

   `http://<server-instance>:port number/search`

2. 「詳細検索」リンクをクリックします。
3. 次のすべてまたは一部の情報を指定します。
   - **検索対象**：検索するコレクションを選択します。
   - **検索**：次の 4 つのオプションがサポートされています。
     - **すべての単語に一致**：「検索」フィールドに指定したすべての単語を含むページを検索します。
     - **いずれかの単語に一致**：「検索」フィールドに指定したいずれかの単語を含むページを検索します。
     - **完全に一致**：「検索」フィールドに入力したとおりの語句を含むページを検索します。
   - **対象外の単語**：指定した単語を含む Web ページが検索対象外となります。
   - **タイトル 次を含む / 含まない**：指定したキーワードをタイトルに含むページ、または含まないページに検索を限定します。
   - **検索対象期間**：指定した期間にインデックスが作成された Web ページに検索を限定します。

検索結果の表示

検索結果は、Web ページとしてユーザーのブラウザに表示されます。このページには、検索条件と一致するサーバー上のドキュメントへの HTML ハイパーリンクが含まれます。デフォルトでは、各ページに 10 レコード (件) が表示され、条件が一致する度合いの高いものから昇順でソートされます。各レコードには、ファイル名、サイズ、作成日などの情報が表示されます。また、一致した単語は強調表示されます。

注	サーバー管理者は、「検索ページのカスタマイズ」で説明する方法で検索結果ページをカスタマイズすることができます。

検索ページのカスタマイズ

Sun ONE Web Server には、基本的な検索クエリページと検索結果ページを提供するデフォルトの検索アプリケーションが含まれます。これらの Web ページは、そのまま使用することも、必要に合わせてカスタマイズすることもできます。このカスタマイズは、Web ページに別のロゴを表示する商標変更のような単純な場合もあれば、検索結果の表示順序を変更するような複雑な場合もあります。

Sun ONE Web Server 6.0 とは異なり、検索インタフェースのカスタマイズにはパターンファイルを使用しません。Sun ONE Web Server 6.1 では、その代わりに、製品に含まれる JSP タグライブラリセットを使ってカスタマイズを行います。デフォルト検索アプリケーションには、検索タグライブラリを使用して検索インタフェースをカスタマイズする方法を示すサンプル JSP が含まれます。カスタマイズ可能な検索タグの使用方法を示すサンプルアプリケーションとして、`/bin/https/webapps/search` に格納されているデフォルトの検索アプリケーションを参照してください。

デフォルトの検索インタフェースは、ヘッダー、フッター、クエリフォーム、結果の 4 つのコンポーネントから構成されます。

これらの基本要素は、タグの属性値を変更するだけで簡単にカスタマイズできます。タグライブラリを使用することで、より詳細なカスタマイズを行えます。

この節では、次の内容について説明します。

- 検索インタフェースのコンポーネント
- 検索クエリページのカスタマイズ
- 検索結果ページのカスタマイズ
- 独立したフォームページと結果ページのカスタマイズ
- タグの規則

- タグの仕様

検索インタフェースのコンポーネント

検索インタフェースは、次のコンポーネントから構成されます。

ヘッダー

ヘッダーには、ロゴ、タイトル、短い説明が含まれます。

フッター

フッターには、著作権情報が含まれます。

フォーム

クエリフォームには、検索対象コレクションに対応するチェックボックス、クエリ入力ボックス、送信ボタン、ヘルプボタンが含まれます。

結果

結果は、デフォルトでは 1 ページに 10 レコードずつリスト表示されます。各レコードには、タイトル、文章の一部、サイズ、作成日、URL などの情報が表示されます。文章の一部は検索されたページの一部で、一致した単語は強調表示されます。

検索クエリページのカスタマイズ

クエリフォームには、検索対象コレクションのチェックボックス、クエリ入力ボックス、送信ボタンが含まれます。このフォームは、`<s1ws:form>`、`<collElem>`、`<queryBox>`、`<submitButton>` のデフォルト値から作成されます。

```
<s1ws:form>
    <s1ws:collElem>
    <s1ws:queryBox> <s1ws:submitButton>
</s1ws:form>
```

クエリフォームは、ページの中央やサイドバー内など、ページのどの場所にも配置できます。また、コレクション選択ボックス、クエリ文字列入力ボックス、送信ボタンを水平に配置するクロスバー形式や、コレクションをチェックボックスとして表示し、その下にクエリ入力ボックスと送信ボタンを配置するブロック形式など、異なる形式で表示することができます。

次の例は、<searchForm> タグセットを使用して異なる形式のクエリフォームを作成する方法を示しています。

水平バー形式

次のコード例は、すべてのコレクションの選択ボックス、クエリ入力ボックス、送信ボタンを水平に配置した形式のフォームを作成します。

```
<s1ws:form>
   <table cellspacing="0" cellpadding="3" border="0">
   <tr class="navBar">
      <td class="navBar"><s1ws:collElem type=”select”></td>
      <td class="navBar">
         <s1ws:querybox size="30">
         <s1ws:submitButton class="navBar" style="padding: 0px;
margin: 0px; width: 50px">
      </td>
   </tr>
   </table>
</s1ws:form>
```

サイドバーブロック形式

フォーム要素をサイドバーに配置した「Search」というタイトルのフォームブロックを作成し、サイドバー内のその他の項目と同じ形式を適用することもできます。次の図は、このような配置の表示結果を示しています。

フォーム要素をサイドバーに表示するクエリページのカスタマイズ

次のコード例では、使用可能な検索対象コレクションが 3 つのチェックボックスとしてフォームの 1 つの列に配置されます。クエリ入力ボックスと送信ボタンは、すぐ下に配置されます。

```
<s1ws:searchForm>
    <table>
<!--... other sidebar items ... -->
    <tr class="Title"><td>Search</td></tr>
    <tr class="Body">
        <td>
        <table cellspacing="0" cellpadding="3" border="0">
```

```
       <tr class="formBlock">

          <td class="formBlock"> <s1ws:collElem type="checkbox"
cols="1" values="1,0,1,0" /> </td>

       </tr>

       <tr class="formBlock">

          <td class="formBlock"> <s1ws:querybox size="15"
maxlength="50"> </td>

       </tr>

       <tr class="formBlock">

          <td class="formBlock"> <s1ws:submitButton class="navBar"
style="padding: 0px; margin: 0px; width: 50px"> </td>

       </tr>

       </table>

       </td>

   </tr>

   </table>

</s1ws:searchForm>
```

検索結果ページのカスタマイズ

検索結果は、次のように生成されます。

- <formAction> タグは、すべてのフォーム要素からの値を受け取り、基本的な検証を行います。
- <search> タグ、<resultIteration> タグ、およびその他のタグは <formAction> タグ内で使用され、すべてのフォーム要素の値を使用できます。
- <search> タグは、<formAction> からのクエリ文字列とコレクションを使用して検索を実行し、検索結果を pageContext に保存します。
- 次に、<resultIteration> タグが結果を受け取り、結果セット全体でこれを繰り返します。

タグの属性値を変更するだけで検索結果ページをカスタマイズできます。

次のコード例は、タイトルバーから始まり、次に指定した数だけレコードを表示して、最後にナビゲーションバーを表示します。タイトルバーには、検索に使用されたクエリ文字列と、返される合計レコード数の範囲 (たとえば 1 - 10 など) が表示されます。各レコードのレコードセクションには、ファイルのタイトルとリンク、キーワードが強調表示された最大で 3 つの文章の一部、URL、作成日、ドキュメントサイズが表示されます。

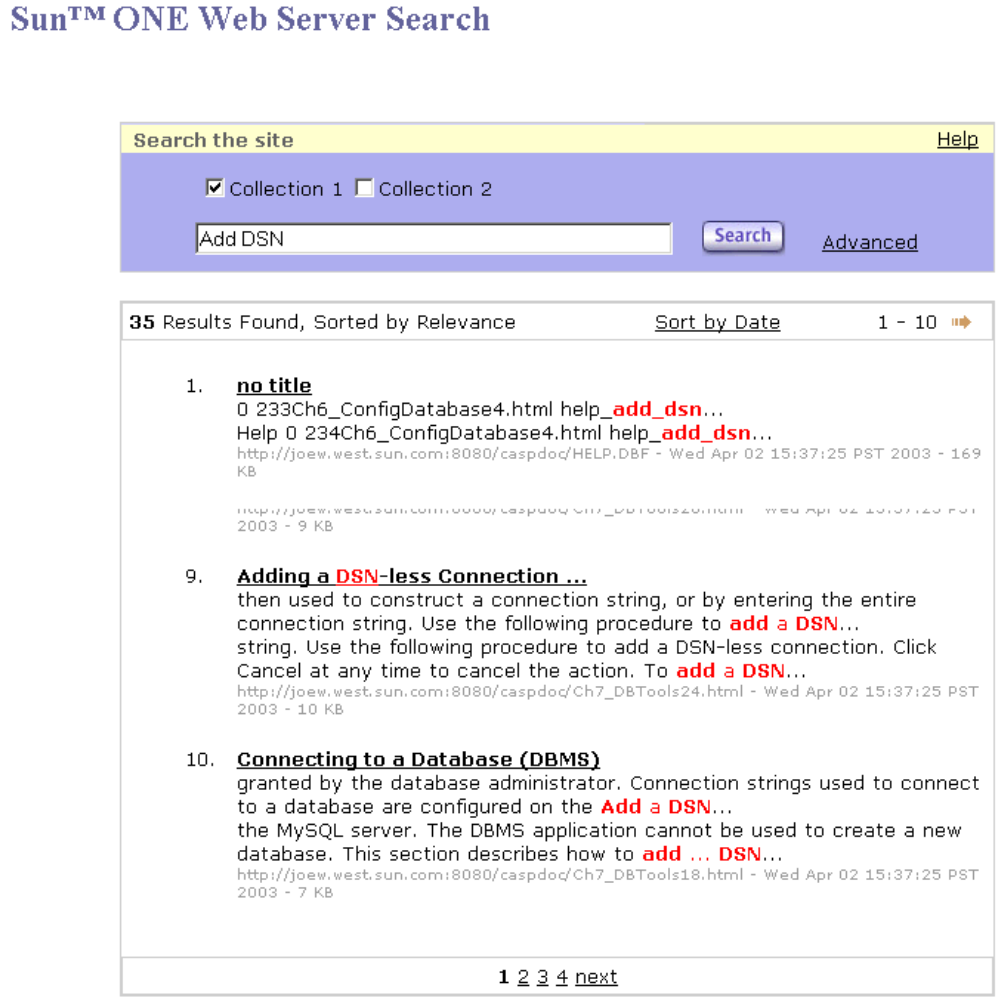
最後のセクションには、前後のページに移動するためのリンク、および現在のページの前後 8 つのページに直接移動するためのリンクを含むナビゲーションバーが表示されます。

```
<s1ws:formAction />

<s1ws:formSubmission success="true" >

   <s1ws:search scope="page" />

   <!--search results-->

   (...html omitted...)

      <s1ws:resultStat formId="test" type="total" /></b> Results
Found, Sorted by Relevance</span></td><td>

      <span class="body"><a href="/search/search.jsp?">Sort by
Date</a></span></td>

      <td align="right"><span class="body">

      <s1ws:resultNav formId="test" type="previous" caption="<img
border=0 src=¥"images/arrow-left.gif¥" alt=¥"Previous¥">" />

       <s1ws:resultStat formId="test" type="range" />

       <s1ws:resultNav formId="test" type="next"
caption="<img border=0 src=¥"images/arrow-right.gif¥" alt=¥"Next¥">"
/>

        <!img alt="Next" src="images/arrow-right.gif"
border="0" WIDTH="13" HEIGHT="9">

         (...html omitted...)

      <table border=0>

      <s1ws:resultIteration formId="test" start="1" results="15">

         <tr class=body>

            <td valign=top>

            <s1ws:item property='number' />.  

            </td>

            <td>
```

```
                    <b><a href="<s1ws:item property='url'
/>"><s1ws:item property='title' /></a></b>
                    <br>
                    <s1ws:item property='passages' />
                    <font color="#999999" size="-2">
                    <s1ws:item property='url' /> -
                    <s1ws:item property='date' /> -
                    <s1ws:item property='size' /> KB
                    </font><br><br>
                </td>
            </tr>
        </s1ws:resultIteration>
        </table>
        (...html omitted...)
        <s1ws:resultNav formId="test" type="previous" />
        <s1ws:resultNav formId="test" type="full" offset="8" />
        <s1ws:resultNav formId="test" type="next" />
    (...html omitted...)
</s1ws:formSubmission>
```

次の図は、カスタマイズされた検索結果ページを示しています。

カスタマイズされた検索結果ページ



基本的な検索結果インタフェースは、タグの操作と HTML の編集によって簡単にカスタマイズできます。たとえば、ナビゲーションバーをコピーして検索結果の上に配置することもできます。また、ユーザーは検索レコードの任意のプロパティを表示または非表示に設定できます。

フォームで使用する以外にも、<search> タグ、<resultIterate> タグ、および関連するタグを使用して、特定のトピックをリスト表示するこができます。次のコード例は、Java Web Service についてサイトで検索された上位 10 本の記事をリスト表示します。

```
<s1ws:search Collection="Articles" Query="Java Web Services" />

<table cellspacing="0" cellpadding="3" border="0">

  <tr class="Title"><td>Java Web Services</td></tr>

</table>

<table cellspacing="0" cellpadding="3" border="0">

<s1ws:resultIteration>

<tr>

<td><a href="<s1ws:item property='URL' />"> <s1ws:item
property='Title'/></a></td>

</tr>

</s1ws:resultIteration>

</table>
```

独立したフォームページと結果ページのカスタマイズ

フォームページと結果ページを分ける必要がある場合は、<form> タグセットを使用してフォームページを作成し、<formAction> タグセットを使用して結果ページを作成する必要があります。

ページの流れを円滑にするために、結果ページ内にフォームページへのリンクを追加する必要があります。

タグの規則

タグの規則については、次の点に留意してください。

- タグのクラスは、`com.sun.web.search.taglibs` パッケージに属します。
- すべての `pageContext` 属性には `com.sun.web` というプレフィックスが付けられます。検索結果の属性は、たとえば `com.sun.web.searchresults.`*form_id* のようになります。この *form_id* は、フォーム名です。
- タグライブラリは、`s1ws` というプレフィックスをつけて参照されます。タグの名前と属性は、たとえば `pageContext` のように、内部の各単語の先頭文字を大文字にした、大文字と小文字の混合で表記されます。

タグの仕様

Sun ONE Web Server には、検索インタフェースの検索クエリページと検索結果ページのカスタマイズに使用する JSP タグセットが含まれています。

検索ページのカスタマイズに使用できる JSP タグすべてを参照するには、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』を参照してください。

第 19 章

WebDAV による Web パブリッシング

Sun ONE Web Server 6.1 は、Web ベースの共同作業の標準として広く採用されている WebDAV (Web-based Distributed Authoring and Versioning) をサポートしています。WebDAV は HTTP/1.1 プロトコルの拡張で、クライアントがリモート Web コンテンツのオーサリング操作を行えます。

この章では、Sun ONE Web Server 6.1 で WebDAV を使用する方法について説明します。次の項目に分けて記述しています。

- WebDAV について
- WebDAV の有効化
- WebDAV コレクションの作成
- WebDAV コレクションの編集
- WebDAV の設定
- WebDAV 対応サーバーでのソース URI と Translate:f ヘッダーの使用
- リソースのロックとロック解除
- WebDAV のアクセス制御の有効化
- セキュリティに関する注意事項

WebDAV について

WebDAV は HTTP/1.1 プロトコルの拡張です。HTML や XML だけでなく、テキスト、グラフィックス、スプレッドシートなど、あらゆる形式の Web リソースに対応するための、新しい HTTP メソッドとヘッダーを追加します。

WebDAV を使用することで、次のようなタスクを実行できます。

- **プロパティ (メタデータ) の操作** : WebDAV メソッドの PROPFIND と PROPPATCH を使用して、作成者や作成日など、Web ページに関する情報を作成、削除、照会することができます。
- **コレクションとリソースの管理** : WebDAV メソッドの GET、PUT、DELETE、MKCOL を使用して、ドキュメントセットを作成し、階層構造のメンバーシップリスト (ファイルシステムのディレクトリリストに似ています) を取得できます。
- **ロック** : WebDAV を使用して、複数のユーザーが同じドキュメントを同時に操作できないようにすることができます。WebDAV メソッドの LOCK および UNLOCK を使用して相互に排他的なロック、または共有ロックを設定することで、更新が失われる問題 (変更の上書き) を回避できます。
- **ネームスペースの操作** : WebDAV のメソッド COPY および MOVE を使用して、サーバーに Web リソースのコピーと移動を指示することができます。

Sun ONE Web Server 6.1 では、WebDAV は次の機能を提供します。

- RFC2518 との互換性と RFC2581 クライアントとの相互動作性
- パブリッシングのためのセキュリティとアクセス制御
- ファイルシステムベースの WebDAV コレクションおよびリソースに対する効率的なパブリッシング操作

一般的な WebDAV 用語

ここでは、WebDAV の利用時に目にすることが多い一般的な用語について説明します。

URI : URI (Uniform Resource Identifier) は、短縮 URL によるセキュリティレイヤを追加したファイル識別子です。URL の先頭部分を URL マッピングに置き換えて、ファイルの完全な物理パス名をユーザーから隠します。

ソース URI : ソース URI は、アクセスするリソースのソースを意味します。ソース URI の概念を理解するために、次の例を考えてください。

foo.jsp という JSP ページが /docs/date.jsp という URI にあります。このページには、HTML マークアップと、実行時にクライアントのブラウザにその日の日付を出力する Java コードが含まれています。サーバーがクライアントから foo.jsp に対する GET 要求を受け取ると、サーバーはそのページを配信する前に Java コードを実行します。クライアントが受け取るのはサーバー上にある foo.jsp ではなく、現在の日付を含むダイナミックに生成されたページです。

たとえば、/publish/docs というソース URI を作成し、foo.jsp ページを含む /docs ディレクトリにそれをマッピングした場合、/publish/docs/foo.jsp への要求は、/docs/foo.jsp JSP ページのソースコードに対する要求となります。この場合、サーバーは Java コードを実行せずにページを配信します。クライアントは、ディスクに格納されている未処理のページを受け取ります。

このため、ソース URI に対する要求は、リソースのソースに対する要求となります。

コレクション: WebDAV は、WebDAV 操作が有効なリソースまたはリソースセットの集合です。コレクションには、WebDAV が有効なメンバーリソースを識別する URI (メンバー URL) のセットが含まれます。

メンバー URI: コレクションに含まれる URI セットのメンバーである URI です。

内部メンバー URI: コレクションに含まれる URI と直接的な関係を持つメンバー URI です。たとえば、http://info.sun.com/resources/info という URL のリソースが WebDAV に対応しており、http://info.sun.com/resources/ という URL のリソースも WebDAV に対応している場合、http://info.sun.com/resources/ という URL のリソースはコレクションであり、内部メンバーとして http://info.sun.com/resources/info を含みます。

プロパティ：リソースを説明する情報を含む、名前と値のペアです。プロパティは、リソースの効率的な検索と管理に使用されます。たとえば、「creationdate」プロパティを使用して、リソースの作成日ですべてのリソースにインデックスを作成したり、「author」プロパティを使用して、作成者のインデックスを作成することができます。

ライブプロパティ：サーバーによって実行されるプロパティです。たとえば、ライブプロパティ getcontentlength は、GET 要求の応答として返されるエンティティの長さを値として持ちます。この長さは、サーバーによって自動的に計算されます。ライブプロパティには次の種類があります。

- サーバーによって管理される、値が読み取り専用のプロパティ
- クライアントが値を管理するが、送信する値の構文チェックをサーバーが行うプロパティ

デッドプロパティ：サーバーによって実行されないプロパティです。サーバーはデッドプロパティの値を記録するだけなので、整合性の維持はクライアント側で行う必要があります。

Sun ONE Web Server 6.1 は、次のライブプロパティをサポートしています。

- creationdate
- displayname
- getcontentlanguage
- getcontentlength

- getcontenttype
- gettag
- getlastmodified
- lockdiscovery
- resourcetype
- supportedlock
- executable

注

Sun ONE Web Server は、ライブプロパティ executable をサポートしています。このプロパティを使用することで、リソースに関連するファイルのアクセス権をクライアントが変更することができます。

次に、executable ライブプロパティの PROPPATCH 要求の例を示します。

```
PROPPATCH /test/index.html HTTP/1.1
Host: sun
Content-type: text/xml
Content-length: XXXX
<?xml version="1.0"?>
<A:propertyupdate xmlns:A="DAV:"
xmlns:B="http://apache.org/dav/props/">
<A:set>
<A:prop>
<B:executable>T</B:executable>
</A:prop>
</A:set>
</A:propertyupdate>
```

ロック：リソースをロックすることで、他のユーザーがリソースを編集中に別のユーザーがそのリソースを変更することを防止するメカニズムが提供されます。ロックは上書きの競合を防止し、更新が失われる問題を解決します。

Sun ONE Web Server は、共有ロックと排他ロックの 2 種類のロックをサポートしています。

新しい HTTP ヘッダー: WebDAV は、HTTP/1.1 プロトコルの拡張として機能します。この拡張は、クライアントが WebDAV リソースの要求を通信するための新しい HTTP ヘッダーを定義します。次のヘッダーが含まれます。

- `Destination:`
- `Lock-Token:`
- `Timeout:`
- `DAV:`
- `If:`
- `Depth:`
- `Overwrite:`

新しい HTTP メソッド: WebDAV には、WebDAV が有効なサーバーに要求の処理方法を指示するための、いくつかの新しい HTTP メソッドがあります。これらのメソッドは、`GET`、`PUT`、`DELETE` などの既存の HTTP メソッドに加えて使用され、WebDAV トランザクションを処理します。次に、新しい HTTP メソッドについて簡単に説明します。

- `COPY`: リソースをコピーします。コレクションは `Depth:` ヘッダーを使用してコピーされ、ターゲットは `Destination:` ヘッダーによって指定されます。COPY メソッドは、必要に応じて `Overwrite:` ヘッダーも使用します。
- `MOVE`: リソースを移動します。コレクションは `Depth:` ヘッダーを使用して移動され、ターゲットは `Destination:` ヘッダーによって指定されます。MOVE メソッドは、必要に応じて `Overwrite:` ヘッダーも使用します。
- `MKCOL`: 新しいコレクションを作成します。このメソッドは、PUT メソッドのオーバーロードを防止するために使用されます。
- `PROPPATCH`: 1 つのリソースのプロパティを設定、変更、または削除します。
- `PROPFIND`: 1 つまたは複数のリソースに属する 1 つまたは複数のプロパティを取得します。クライアントがコレクションに対する PROPFIND 要求をサーバーに送信するときに、値として `0`、`1`、または `infinity` を持つ Depth: ヘッダーがその要求に含まれる場合があります。
  - `0`: 指定された URI のコレクションのプロパティを取得します。
  - `1`: 指定された URI のコレクションと、その直下にあるリソースのプロパティを取得します。
  - `infinity`: コレクションと、コレクションに含まれるすべてのメンバー URI のプロパティを取得します。`infinite` が指定された要求ではコレクション全体が検索されるため、サーバーに大きな負荷を生じる可能性があるので注意してください。
- `LOCK`: リソースにロックを追加します。`Lock-Token:` ヘッダーを使用します。

- UNLOCK: リソースからロックを消去します。Lock-Token: ヘッダーを使用します。

WebDAV の使用

完全な WebDAV トランザクションには、WebDAV リソースに対する要求を処理するサーバーとして、Sun ONE Web Server 6.1 などの WebDAV 対応サーバーと、Web パブリッシング要求をサポートする WebDAV 対応クライアント (Adobe の GoLive、Macromedia の DreamWeaver など) が必要です。

サーバー側では、WebDAV 要求を処理できるように Sun ONE Web Server 6.1 を有効にし、設定する必要があります。

WebDAV を使用できるように Sun ONE Web Server 6.1 を設定するには、次の手順を実行する必要があります。

- WebDAV の有効化
- WebDAV コレクションの作成
- WebDAV の設定
- WebDAV のアクセス制御の有効化

WebDAV の有効化

Sun ONE Web Server 6.1 のインストール時に、WebDAV はデフォルトでは無効に設定されています。

コレクションレベルで WebDAV を有効にする場合は、サーバーレベルと仮想サーバークラスレベルでも WebDAV を有効にする必要があります。

注	コレクションレベルで指定した属性は、仮想サーバーレベルで指定された属性に優先して適用されます。

次に、WebDAV を有効にする各レベルについて説明します。

- サーバーインスタンスレベルでの WebDAV の有効化
- 仮想サーバークラスレベルでの WebDAV の有効化
- コレクションレベルでの WebDAV の有効化

サーバーインスタンスレベルでの WebDAV の有効化

管理サーバーを使用して、サーバー全体で WebDAV を有効にすることができます。この場合は、WebDAV プラグインをロードする次の指令が `magnus.conf` ファイルに追加されます。

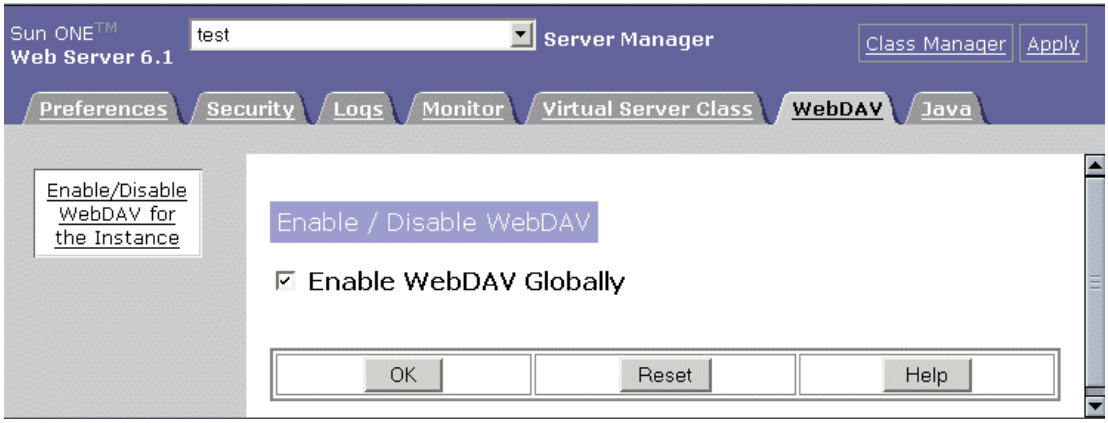
```
Init fn="load-modules" shlib="/s1ws6.1/lib/libdavplugin.so"
funcs="init-dav,ntrans-dav,pcheck-dav,service-dav"

shlib_flags="(global|now)"

Init fn="init-dav" LateInit=yes
```

Init 関数 `init-dav` は、WebDAV サブシステムを初期化して登録します。

WebDAV をグローバルに有効にするには、次の手順を実行します。

1. WebDAV を有効にするサーバーのサーバーマネージャを開きます。
2. 「Preferences」タブの「Enable/Disable WebDAV」リンクをクリックします。
3. 「Enable WebDAV Globally」チェックボックスを選択します。

   インスタンスレベルでの WebDAV の有効化



4. 「Apply」をクリックします。

5. 「Apply Changes」ボタンをクリックしてサーバーを再起動します。

   または

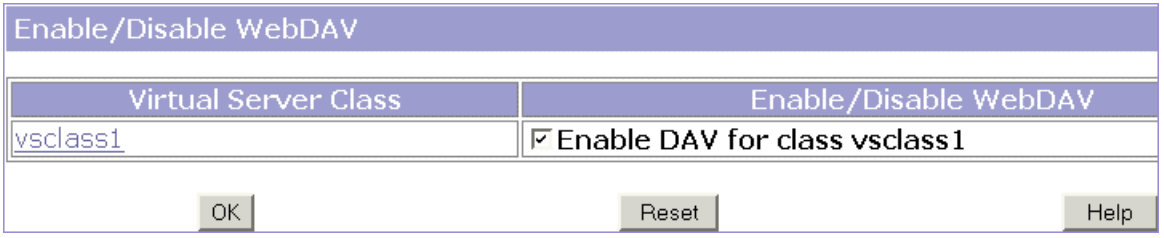
   「Load Configuration Files」をクリックして変更をダイナミックに適用します。

仮想サーバークラスレベルでの WebDAV の有効化

特定の仮想サーバークラスで WebDAV を有効にするには、次の手順を実行します。

1. 仮想サーバークラスを選択します。
2. 「Content Mgmt」タブをクリックします。
3. 「Enable/Disable WebDAV」リンクをクリックします。

   仮想サーバークラスレベルでの WebDAV の有効化



4. 「Enable DAV」チェックボックスを選択します。
5. 「OK」をクリックします。

仮想サーバークラスで WebDAV を有効にすると、関連する `obj.conf` ファイルが次のエントリで更新されます。

```
<Object name="default">
...
Service fn="service-dav"
method="(OPTIONS|PUT|DELETE|COPY|MOVE|PROPFIND|PROPPATCH|LOCK|UN
LOCK|MKCOL)"
Error fn="error-j2ee"
...
</Object>
...
<Object name="dav">
PathCheck fn="check-acl" acl="dav-src"
Service fn="service-dav"
method="(GET|HEAD|POST|PUT|DELETE|COPY|MOVE|PROPFIND|PROPPATCH|L
OCK|UNLOCK|MKCOL)"
</Object>
```

コレクションレベルでの WebDAV の有効化

1 つまたは複数の WebDAV コレクションを仮想サーバーに追加した場合は、各コレクションの WebDAV をいつでも無効または有効にすることができます。方法については、433 ページの「WebDAV コレクションの編集」を参照してください。

WebDAV コレクションの作成

WebDAV は、WebDAV 操作が有効なリソースまたはリソースセットの集合です。この操作には、Web パブリッシングと、共同作業によるオーサリング、ネームスペース管理、メタデータ管理が含まれます。

WebDAV コレクションを仮想サーバーに追加するには、次の手順を実行します。

1. サーバーインスタンスと仮想サーバークラスで WebDAV が有効であることを確認します。詳細については、429 ページの「サーバーインスタンスレベルでの WebDAV の有効化」および 430 ページの「仮想サーバークラスレベルでの WebDAV の有効化」を参照してください。
2. 管理する仮想サーバーにアクセスし、「WebDAV」タブをクリックします。
3. 「Add DAV Collection」ページに次の情報を入力します。
   - URI (必須) : アクセスするコンテンツの URI
   - Source URI (省略可能) : アクセスするソースの URI

注 CGI や SHTML などのダイナミックなコンテンツをパブリッシュする場合は、ソース URI を設定する必要があります。

ソース URI という用語の説明については、「一般的な WebDAV 用語」を参照してください。

- Lock Database (省略可能) : ロックデータベースを管理するディレクトリ。デフォルト値は *server-instance*/`lock-db/vs-id` です。
- Minimum Lock Timeout (省略可能) : ロックの最小存続時間 (秒単位)。デフォルト値は `0` です。詳細は、「ロックの最小タイムアウト」を参照してください。
- Limit XML Request Body (省略可能) : 要求のボディに含まれる XML コンテンツの最大サイズ。サービス妨害 (DOS) 攻撃を防止するには、このサイズを設定してください。
- Maximum Property Depth (省略可能) : `PROPFIND` 要求の適用レベルです。
  - `0`: 指定したリソースだけに適用します。
  - `1`: 指定したリソースと、そのリソースに含まれる次のレベルのリソースに適用します。
  - `infinity`: 指定したリソースと、そのリソースに含まれるすべてのリソースに適用します。

  デフォルト値は `0` です。
- Enabled (省略可能) : コレクションレベルで WebDAV を有効にします。

4. 「OK」をクリックします。

注

- 管理サーバーを使用してコレクションを追加しても、サーバーはファイルシステムにそのコレクションのディレクトリを作成しません。コレクションに対応するディレクトリをファイルシステムに作成することは、管理者の責任です。
- UNIX システムでは、`root` (スーパーユーザー) として Web Server をインストールし、別のユーザーとしてサーバーを実行している場合、作成した WebDAV コレクションに対応するディレクトリへの読み取りと書き込みアクセス権がこの実行ユーザーに設定されている必要があります。

WebDAV コレクションの編集

既存の DAV コレクションの属性を編集して、そのコレクションのアクセス制御などを設定することができます。

既存の WebDAV コレクションを編集するには、次の手順を実行します。

1. コレクションが存在する仮想サーバーにアクセスし、「WebDAV」タブをクリックします。
2. 「Edit DAV Collections」ページで次の情報を変更します。
   - Delete: コレクションを削除します。
   - URI: アクセスするコンテンツの URI を示します。
   - Enabled: WebDAV が有効であるか (`true`)、無効であるか (`false`) を示します。
   - Edit Collection: 次の情報を変更するときは、このボタンをクリックします。
     - URI (必須) : アクセスするコンテンツの URI
     - Source URI (省略可能) : アクセスするソースの URI
     - Lock Database (省略可能) : ロックデータベースを管理するディレクトリ
     - Minimum Lock Timeout (省略可能) : ロックの最小存続時間 (秒単位)。詳細は、「ロックの最小タイムアウト」を参照してください。

注

`minlocktimeout` の -1 という値は、ロックに時間制限がないことを意味します。

     - Limit XML Request Body (省略可能) : 要求のボディに含まれる XML コンテンツの最大サイズ

- Maximum Property Depth (省略可能) : PROPFIND 要求の適用レベルです。

  - 0: 指定したリソースにだけに適用します。

  - 1: 指定したリソースと、そのリソースに含まれる次のレベルのリソースに適用します。

  - infinity: 指定したリソースと、そのリソースに含まれるすべてのリソースに適用します。

  デフォルト値は 0 です。

- Enabled (省略可能) : コレクションレベルで WebDAV を有効にします。

- Edit ACL: このコレクションまたは URI のアクセス制御制限を設定するときは、これをクリックします。

WebDAV の設定

たとえば、サーバーのパフォーマンスを調整したり、セキュリティ上のリスクを解消したり、または競合のないリモートオーサリングを提供したりする場合など、WebDAV の設定にはさまざまな理由が考えられます。

要件に適した設定を行うには、サーバーが WebDAV リソースのロックを保持する最小時間、コレクションに対する PROPFIND 要求の適用レベル、要求のボディに含めることができる XML コンテンツの最大サイズなどを変更します。

デフォルトの WebDAV 属性は、仮想サーバーの下にあるすべてのコレクションについて、仮想サーバーレベルで変更できます。ここで設定する値は、server.xml ファイルの DAV 要素に対応します。

WebDAV 属性は、コレクションレベルで設定することもできます。この設定は、コレクションに設定されている仮想サーバーレベルの属性に優先して適用されます。コレクションレベルで設定した属性値は、server.xml ファイルの DAVCOLLECTION 要素に対応します。

- 仮想サーバーレベルでの WebDAV の設定
- URI レベルでの WebDAV の設定

仮想サーバーレベルでの WebDAV の設定

仮想サーバーレベルで WebDAV の機能を設定するには、DAV オブジェクトの属性を編集する必要があります。この処理は、管理サーバーを使用して行うか、server.xml ファイルを手動で編集して行います。

次の表は、編集可能な DAV オブジェクトの属性と、その説明を示しています。

表 19-1 DAV オブジェクトの属性

属性	説明
enabled	この仮想サーバーで WebDAV の機能が有効であるかどうかを指定します。 この属性は省略可能です。デフォルト値は true です。 指定できる値は true および false です。
lockdb	ロックデータベースを管理するディレクトリを指定します。 この属性は省略可能です。
minlocktimeout	ロックの最小存続時間 (秒単位) を指定します。この値は、ロックが自動的に解除されるまでに、要素がロックされている時間を示します。詳細は、「ロックの最小タイムアウト」を参照してください。 この属性は省略可能です。
maxxmlrequestbodysize	要求のボディに含まれる XML コンテンツの最大サイズを指定します。 この属性は省略可能です。デフォルト値は 8k バイトです。 サービス妨害 (DOS) 攻撃を防止するには、このサイズを設定してください。
maxpropdepth	PROPFIND 要求の適用レベルを指定します。 このパラメータは省略可能です。デフォルト値は 0 です。 メモリの過大消費を防止するには、このパラメータの値を設定します。

URI レベルでの WebDAV の設定

URI レベルで WebDAV の機能を設定するには、server.xml ファイルの DAVCOLLECTION オブジェクトの属性を編集する必要があります。

次の表は、編集可能な DAVCOLLECTION オブジェクトの属性と、その説明を示しています。

表 19-2 DAVCOLLECTION オブジェクトの属性

属性	説明
enabled	このコレクションで WebDAV の機能が有効であるかどうかを指定します。 この属性は省略可能です。 指定できる値は true および false です。デフォルト値は true です。
uri	アクセスするコンテンツの URI を指定します。 この属性の指定は必須です。
sourceuri	アクセスするソースの URI を指定します。詳細については、「一般的な WebDAV 用語」および「WebDAV 対応サーバーでのソース URI と Translate:f ヘッダーの使用」を参照してください。 この属性は省略可能です。 sourceuri を指定しない場合はデフォルトの動作が適用され、コレクション内のダイナミックなコンテンツのソースへのアクセスが拒否されます。 uri と sourceuri に同じ URI を指定することができます。この場合、サーバーは常にダイナミックなコンテンツのソースを返します。これは、パブリッシング用に独立した安全な仮想サーバーを利用している場合に便利です。
lockdb	ロックデータベースを管理するディレクトリを指定します。 この属性は省略可能です。
minlocktimeout	ロックの最小存続時間 (秒単位) を指定します。この値は、ロックが自動的に解除されるまでに、要素がロックされている時間を示します。詳細は、「ロックの最小タイムアウト」を参照してください。 この属性は省略可能です。

表 19-2　DAVCOLLECTION オブジェクトの属性 (続き)

属性	説明
maxxmlrequestbodysize	要求のボディに含まれる XML コンテンツの最大サイズを指定します。 この属性は省略可能です。 サービス妨害 (DOS) 攻撃を防止するには、このサイズを設定してください。
maxpropdepth	コレクションのメンバーリソースをリスト表示する PROPFIND 要求の適用レベルを指定します。 このパラメータは省略可能です。 メモリの過大消費を防止するには、このパラメータの値を設定します。

WebDAV 対応サーバーでのソース URI と Translate:f ヘッダーの使用

WebDAV メソッドは、リソースまたはコレクションのソースに対して実行されます。GET や PUT などの HTTP メソッドは WebDAV プロトコルによってオーバーロードするため、これらのメソッドを持つ要求は、リソースのソースに対する要求か、またはリソースのコンテンツ (出力) に対する要求となります。

Microsoft およびその他多数の WebDAV ベンダーは、要求に Translate:f ヘッダーを含め、その要求がソースに対するものであることをサーバーに指示することで、この問題を解決しています。広く利用されている WebDAV クライアントである Microsoft WebFolders との相互動作性を確保するために、Sun ONE Web Server 6.1 は、Translate:f ヘッダーによりリソースのソースに対する要求として認識します。Translate:f ヘッダーを送信しないクライアントに対応するために、Sun ONE Web Server 6.1 では ソース URI を定義します。ソース URI という用語の詳しい説明については、「一般的な WebDAV 用語」を参照してください。

WebDAV が有効なコレクションでは、URI に対する要求はリソースのコンテンツ (出力) を取得し、ソース URI に対する要求はリソースのソースを取得します。Translate:f ヘッダーを持つ URI に対する要求は、ソース URI に対する要求として処理されます。

リソースのソースに対するすべてのアクセスは、デフォルトでは dav-src ACL によって拒否されることに注意してください。サーバーインスタンスに固有の ACL ファイルには、次の宣言が含まれます。

```
PathCheck fn="check-acl" acl "dav-src";
```

```
deny (all) user = "anyone";
```

ユーザーは、ソース URI に対するアクセス権を追加することで、ソースに対するアクセスを有効にすることができます。URI 固有の ACL の詳細については、「WebDAV のアクセス制御の有効化」を参照してください。

リソースのロックとロック解除

Sun ONE Web Server では、サーバー管理者はリソースをロックし、そのリソースに対するアクセスを直列化することができます。ロックを使用することで、特定のリソースにアクセスするユーザーは、他のユーザーが同じリソースを変更しないようにできます。このようにすると、サーバー上の複数のユーザーがリソースを共有することによって生じる、更新が失われる問題が解決されます。クライアントが発行し、使用するロックトークンは、サーバーが管理するロックデータベースによって追跡されます。

Sun ONE Web Server 6.1 は、`opaquelocktoken` URI スキームをサポートしています。このスキームは、すべてのリソースで URI が常に一意であるように設計されています。これは、ISO-11578 で規定されている UUID (Universal Unique Identifier) メカニズムを使用します。

Sun ONE Web Server 6.1 は、次の 2 種類のロックメカニズムを認識します。

- 排他的ロック
- 共有ロック

排他的ロック

排他的ロックは、リソースに対するアクセス権を 1 人のユーザーだけに与えるロックです。別のユーザーは、リソースの排他的ロックが解除されるまでこのリソースにアクセスできません。

排他的ロックは、リソースをロックするメカニズムとしては厳密すぎたり、パフォーマンスへの影響が多きすぎる場合もあります。たとえば、プログラムがクラッシュしたり、ロックの所有者がリソースのロック解除を忘れた場合、排他的ロックを解除するにはロックタイムアウトに達するのを待つか、管理者が設定変更を行う必要があります。

共有ロック

共有ロックでは、複数のユーザーがリソースのロックを受け取ることができます。このため、適切なアクセス権があるユーザーであれば、ロックを取得できます。

共有ロックを使用するときは、ロックの所有者は何らかの通信手段を使用して作業を調整する必要があります。共有ロックの目的は、誰がリソースを操作できるかを共同作業者に知らせることです。

ロックの管理

Sun ONE Web Server 6.1 のロック管理機能を使用することで、設定されているすべてのロック、ロックの種類、対象リソース、ロックされている時間などを確認することができます。

ロック管理機能を使用するには、次の手順を実行します。

1. WebDAV 対応仮想サーバーにアクセスします。
2. 「WebDAV」タブをクリックします。
3. 「Lock Management」リンクをクリックします。
4. ロックデータベースを選択し、設定されているロックとその情報を表示する WebDAV 対応 URI を選択します。
5. 「List Lock Info」をクリックします。

ロックの最小タイムアウト

`server.xml` ファイルの `DAV` オブジェクトまたは `DAVCOLLECTION` オブジェクトで `minlocktimeout` 属性に値を設定することで、ロックを制御することができます。`minlocktimeout` 属性は、ロックの最小存続時間を秒単位で指定します。この値は、ロックが自動的に解除されるまでに、要素がロックされている時間を示します。

この属性は省略可能です。-1 という値が設定されている場合、そのロックには時間制限がありません。値を `0` に設定した場合、コレクションに含まれるすべてのリソースがロックされ、要求に Timeout ヘッダーが指定されます。

Timeout ヘッダーを指定しない場合、リソースのロックはタイムアウトしません。また、要求に含まれる Timeout ヘッダーの値が `Infinite` に設定されている場合も、リソースのロックはタイムアウトしません。

WebDAV リソースに対する要求に含まれる Timeout ヘッダーの値が `server.xml` ファイルの `minlocktimeout` の値と同じか、あるいはそれより大きい場合は、リソースのロック期間は要求に指定されている時間となります。

ただし、server.xml ファイルの minlocktimeout 値より小さい値が要求の Timeout ヘッダーに指定されている場合は、リソースのロック期間は server.xml ファイルに設定されている minlocktimeout の値となります。

次の表は、Sun ONE Web Server がロック要求をどのように処理するかを示しています。

表 19-3　Sun ONE Web Server によるロック要求の処理

要求に含まれる Timeout ヘッダーの値	リソースのロック期間
Infinite	タイムアウトが -1 (時間制限なし) の設定でロック
なし	タイムアウトが -1 (時間制限なし) の設定でロック
Second-*xxx*	• *xxx* が server.xml の minlocktimeout 値と同じまたは大きい場合は、*xxx* 秒間ロックされる または • *xxx* が server.xml の minlocktimeout 値より小さい場合は、server.xml の minlocktimeout に指定されている秒数だけロックされる

ロック要求の例

次の例は、タイムアウトを 500 秒に設定して /col1/myfile.html リソースへの書き込みを排他的にロックする要求を示しています。

```
LOCK /col1/myfile.html HTTP/1.1
Host: sun
Content-Type: text/xml; charset="utf-8"
Content-Length: 259
Timeout: Second-500
<?xml version=”1.0” encoding=”utf-8” ?>
<d:lockinfo xmlns:d=”DAV:”>
    <d:locktype><d:write/></d:locktype>
    <d:lockscope><d:exclusive/></d:lockscope>
    <d:owner>
    <d:href>http://info.sun.com/resources/info.html</d:href>
    </d:owner>
</d:lockinfo>
```

WebDAV のアクセス制御の有効化

WebDAV が有効なドキュメントとディレクトリに誰がアクセスできるか、また、どのユーザーまたはユーザーグループがファイルに対してどのような処理を実行できるかを制御することができます。また、ファイルやフォルダに対するアクセスを完全に禁止したり、認証された特定のユーザーだけにアクセスを制限することもできます。

サーバーに適用されるデフォルトのアクセス制御 (ACL) では制限や柔軟性が十分でない場合は、「Restrict Access」機能 (「Server Preferences」を選択し、「Restrict Access」リンクをクリック) を使用して、WebDAV が有効なリソースに対するアクセス制限に適した ACL を作成できます。

WebDAV 要求は、AuthTrans ステージと PathCheck NSAPI ステージでそれぞれ認証、承認されます。次の例は、「joe」というユーザー名を持たないすべてのユーザーにコレクション /catalog への書き込みアクセスおよび削除アクセスを拒否するアクセス制御規則を示しています。

```
acl "uri=/catalog/*";

deny(all)

user="anyone";

allow (read,list,execute,info)

user = "all";

allow(write,delete)

user="joe";
```

詳細は、「WebDAV コレクションの編集」を参照してください。

WebDAV 有効リソースへのアクセスの制限

WebDAV コレクションのアクセス制御は、ネイティブ ACL ファイルで指定します。すべての WebDAV メソッドは、WebDAV 有効リソースに対する適切なアクセス権を必要とします。たとえば、WebDAV 有効ファイルが同時に複数のユーザーによって共有される場合、同時アクセスを制御するためにリソースをロックまたはロック解除するには、リソースに対する書き込みアクセス権が必要です。

次の表は、各 WebDAV メソッドに必要なアクセス権を示しています。

表 19-4 WebDAV に必要なアクセス権

DAV メソッド	必要なアクセス権
DELETE	削除
PROPFIND	読み取り

表 19-4　WebDAV に必要なアクセス権 (続き)

DAV メソッド	必要なアクセス権
PROPPATCH	書き込み
LOCK/UNLOCK	書き込み
MKCOL	書き込み
COPY(*src,dst)*	*src* - 読み取り
	dst - 書き込み
MOVE(*src,dst*)	*src* - 削除
	dst - 書き込み
request-uri の GET	読み取り
request-uri の GET Translate:f	読み取り
request-uri の PUT	書き込み
request-uri の PUT Translate:f	書き込み

セキュリティに関する注意事項

WebDAV を使用するときは、セキュリティに関する次の事項に注意してください。

- WebDAV が有効なサーバーのプロセスは、制御対象のファイルシステムに対する読み取り、書き込み権限を持っていることを確認します。
- セキュリティ上の理由から、WebDAV が有効な仮想サーバーを、アクセスが制限され、SSL を使用してデータを暗号化して送信する別の待機ソケットに設定する必要がある場合があります。SSL の使用については、「証明書と鍵の使用」を参照してください。
- サービス妨害 (DOS) 攻撃を防止するために、要求のボディに含まれる XML コンテンツのサイズを制限してください。このサイズは、デフォルトでは 8k バイトに制限されています。
- 基本認証では認証詳細の転送にプレーンテキストが使用されるため、接続がセキュリティ保護されている場合を除き、WebDAV クライアントの認証には基本認証ではなく、ダイジェスト認証を使用してください。
- PROPFIND 要求を実行した場合、サーバーコンテンツに対して好ましくないアクセスが行われるリスクがあるので、アクセス制御を使用して WebDAV 対応リソースをセキュリティ保護してください。

- ソース URI 機能により、WebDAV にはスクリプトリソースなどの重要な情報を含む URI を開示してしまう可能性があります。スクリプトのリモート作成が可能になるというリスクを認識し、ソースリソースへの読み取りと書き込みのアクセスを、認証されたユーザーだけに限定してください。
- PROPFIND 要求の適用レベルを制限し、メモリの過大消費を回避してください。適用レベルは、デフォルトでは 0 に設定されています。

第 5 部

付録

付録 A 「コマンド行ユーティリティ」

付録 B 「Hypertext Transfer Protocol」

付録 C 「ACL ファイルの構文」

付録 D 「国際化とローカライズのサポート」

付録 A

コマンド行ユーティリティ

この付録では、HttpServerAdmin コマンド行ユーティリティの使用方法について説明します。

HttpServerAdmin (仮想サーバーの管理)

HttpServerAdmin は、サーバーマネージャとクラスマネージャでの仮想サーバーのユーザーインタフェースと同じ管理機能を実行するコマンド行ユーティリティです。コマンド行インタフェースを使用して仮想サーバーを設定する場合は、HttpServerAdmin を使用します。

注 HttpServerAdmin コマンド行ユーティリティを使用するには、システムのスーパーユーザー権限が必要です。

The HttpServerAdmin コマンド行ユーティリティは、*server_root*/bin/https/httpadmin/bin ディレクトリにあります。

HttpServerAdmin を実行するには、使用している環境でサーバーのルートディレクトリに環境変数 IWS_SERVER_HOME を設定する必要があります。

たとえば、UNIX/Linux システムでは、次のように設定します。

```
setenv IWS_SERVER_HOME /usr/sun/servers
```

Windows の場合は、次のようにして設定します。

1. 「コントロールパネル」で「システム」を選択します。
2. 「環境」タブをクリックします。
3. 「変数」フィールドに「IWS_SERVER_HOME」、「値」フィールドにサーバールートへのパスを入力します。

4. 「設定」をクリックします。
5. 「OK」をクリックします。

注	すべてのコマンドを実行するには、仮想サーバーの情報が格納されるファイル `server.xml` への書き込み権限が必要です。

HttpServerAdmin の構文

`HttpServerAdmin` の構文を以下に示します。

`HttpServerAdmin` *command_name command_options* `-d` *server_root* `-sinst` *http_instance*

次のコマンドを入力すると、コマンドパラメータのオンラインヘルプを参照できます。

```
./HttpServerAdmin -h
```

command_name パラメータとして使用可能な 4 つの値は、次のとおりです。

- `control`
- `create`
- `delete`
- `list`

各コマンドには、独自のコマンドオプションのセットがあります。詳細については、この章の各コマンドについて説明している節を参照してください。

コマンドパラメータの値に関わらず、表 A-1 に示されるパラメータは、`HttpServerAdmin` コマンドのすべての使用に適用できます。

表 A-1　HttpServerAdmin パラメータ

パラメータ	値
`-d` *server_root*	(必須)。このパラメータはサーバールート (サーバーがインストールされている場所) へのパスを指定する
`-sinst` *http_instance*	(必須)。このパラメータは、`HttpServerAdmin` によって影響を受けるインスタンスを指定する

control コマンド

control コマンドを使用して、クラスと仮想サーバーの開始、停止、無効化を実行します。仮想サーバーを指定しない場合、クラス内のすべての仮想サーバーに対してコマンドの開始、停止、または無効化を行います。

オプション

表 A-2 に示すオプションとともに control コマンドを使用すると、クラスと仮想サーバーを制御できます。

表 A-2 control コマンドのオプション

オプション	値
-start	指定された仮想サーバーを開始する。仮想サーバーが指定されていない場合は、クラス内のすべての仮想サーバーを開始する
-stop	指定された仮想サーバーを停止する。仮想サーバーが指定されていない場合は、クラス内のすべての仮想サーバーを停止する
-disable	指定された仮想サーバーを無効にする。仮想サーバーが指定されていない場合は、クラス内のすべての仮想サーバーを無効にする

構文

HttpServerAdmin control -cl *classname*, *-control_option* [-id *virtual_server*] -d *server_root* -sinst *http_instance*

パラメータ

これらのパラメータをコマンドオプションとともに使用して、仮想サーバーを制御します。

表 A-3 control コマンドのパラメータ

パラメータ	値
-cl *classname*	仮想サーバークラスを指定する
-id *virtual_server*	(省略可能) 制御している仮想サーバーの ID を指定する

例

```
HttpServerAdmin control -cl myclass -start -id myvirtualserver -d
/usr/sun/servers -sinst https-sun.com

HttpServerAdmin control -cl myclass -stop -id myvirtualserver -d
/usr/sun/servers -sinst https-sun.com

HttpServerAdmin control -cl myclass -disable -id myvirtualserver
-d /usr/sun/servers -sinst https-sun.com
```

create コマンド

create コマンドを使用して、仮想サーバーのクラス、仮想サーバー、および待機ソケットを作成します。

オプション

表 A-4 に示すオプションとともに create コマンドを使用すると、クラス、待機ソケット、仮想サーバー、リソースを作成できます。

表 A-4　create コマンドのオプション

オプション	値
-c	仮想サーバークラスを作成する
-l	待機ソケットを作成する
-v	仮想サーバーを作成する
-r	リソースを作成する

オプションにはそれぞれパラメータがあります。次の節で各パラメータについて説明します。

仮想サーバークラスの作成

create コマンドのこのオプションを使用して、仮想サーバークラスを作成します。

構文

```
HttpServerAdmin create -c -cl classname -docroot document_root [-obj
obj.conf_file] [-acptlang accept_language] -d server_root -sinst http_instance
```

パラメータ

表 A-5 に示すパラメータを `create -c` コマンドオプションとともに使用し、クラスを作成します。

表 A-5 仮想サーバークラス作成のパラメータ

パラメータ	値
`-cl` *classname*	作成するクラス名
`-docroot` *document_root*	クラスのドキュメントルート。これは絶対パスにする必要がある
`-obj` *obj.conf_file*	(省略可能) クラスの `obj.conf` ファイル。このパラメータを指定しない場合、サーバーは *classname*`.obj.conf` という名前で `obj.conf` ファイルを作成する。クラスの `obj.conf` ファイルに別の名前を付ける場合は、ここで指定する
`-acptlang` *accept_language*	(省略可能) このパラメータを指定しない場合、acptlang はデフォルトでオフになる

例

```
HttpServerAdmin create -c -cl myclass1 -docroot /docs -d
/export/sun/servers -sinst https-sun.com
```

待機ソケットの作成

`create` コマンドのこのオプションを使用して、待機ソケットを作成します。

構文

```
HttpServerAdmin create -l -ip ip_address -port port_number -sname
server_name -id default_virtual_server [-sec security] [-acct
number_of_accept_threads] -d server_root -sinst http_instance
```

パラメータ

表 A-6 に示すパラメータを create -l コマンドオプションとともに使用し、待機ソケットを作成します。

表 A-6　待機ソケット作成のパラメータ

パラメータ	値
-ip *ip_address*	待機ソケットの IP アドレス
-port *port_number*	待機ソケットのポート番号
-sname *server_name*	待機ソケットに関連付けるサーバー名
-id *default_virtual_server*	デフォルトの仮想サーバーの ID。待機ソケットを作成できるようにするには、この仮想サーバーがすでに存在することが必要
-acct *number_of_accept_threads*	(省略可能) 待機ソケットの受け入れスレッド数
-sec *on*	(省略可能) 指定されている場合は、待機ソケットに対するセキュリティを有効にするために使用する。指定されていない場合は、セキュリティが有効になっていない

例

```
HttpServerAdmin create -l -id ls3 -ip 0.0.0.0 -port 1333 -sname
austen -defaultvs vs2 -sec on -acct 4 -d /export/carey/server6
-sinst https-austen.com
```

仮想サーバーの作成

create コマンドのこのオプションを使用して、仮想サーバーを作成します。

省略可能なパラメータの一部に値を指定しない場合、デフォルト値が使用されます。仮想サーバーが作成されたあと、いつでもデフォルト値を変更することができます。

構文

HttpServerAdmin create -v -id *virtual_server* -cl *classname* -urlh *urlhosts* [-state *state*] [-docroot *document_root*] [-mime *mime_types_file*] [-aclid *acl_ID*] -d *server_root* -sinst *http_instance*

パラメータ

表 A-7 に示すパラメータを `create -v` コマンドオプションとともに使用し、仮想サーバーを作成します。

表 A-7 待機ソケット作成のパラメータ

パラメータ	値
`-id` *virtual_server*	作成する仮想サーバーの ID
`-cl` *classname*	仮想サーバーがメンバーとなるクラス
`-urlh` *URL_hosts*	仮想サーバーの URL ホスト。コンマで区切ることによって、複数の URL ホストを指定できる
`-state` *state*	(省略可能) 有効な値は、「on」、「off」、および「disable」
`-docroot` *document_root*	(省略可能) 仮想サーバーのドキュメントルートを指定する場合は、このパラメータを使用する。絶対パスを指定することが必要
`-mime` *mime_types_file*	(省略可能) 仮想サーバーの MIME タイプのファイル名
`-aclid` *acl_ID*	(省略可能) `server.xml` ファイルで使用される ACL ファイルの ID <ACLID>

例

```
HttpServerAdmin create -v -id vs3 -cl class1 -urlh annh -d
/export/sun/server6 -sinst https-sun.com

HttpServerAdmin create -v -id vs4 -cl class1 -urlh annh,annh2
-state off -mime mime.types -d /export/sun/server6 -sinst
https-sun.com
```

JDBC 接続プールの作成

`create -r` コマンドを使用して、コマンド行インタフェースから JDBC 接続プールを新規作成します。

構文

`HttpServerAdmin -create -r -jdbcconnectionpool -poolname` *jdbcpoolname* `-classname` *classname* [`-steadypoolsize` *steadypoolsize*] [`-maxpoolsize` *maxpoolsize*] [`-poolresizequantity` *poolresizequantity*] [`-idletimeout` *idletimeout*] [`-maxwaittime` *maxwaittime*] [`-connectionvalidation` *true/false*] [`-connectionvalidationmethod` *connectionvalidationmethod*] [`-validationtablename` *validationtablename*] [`-failall` *true/false*] [`-desc` *description*] [[`-property` *propertyname=value*],...]

オプション

次の表は、`create -r` コマンドオプションを使用して接続プールを作成するときに必要なすべてのオプションを示しています。

表 A-8 接続プール作成のパラメータ

パラメータ	値
`poolname` *jdbcpoolname*	JDBC 接続プールのプール名
`classname` *classname*	データソースを実装するベンダー固有のクラス名
`steadypoolsize` *steadypoolsize*	プール内で維持する必要がある最小接続数
`maxpoolsize` *maxpoolsize*	プールに許容される最大接続数
`poolresizequantity` *poolresizequantity*	`steadypoolsize` 値に達した場合にプールサイズを変更するバッチのサイズ
`idletimeout` *idletimeout*	接続がプール内でアイドル状態でいられる最大時間 (秒)
`maxwaittime` *maxwaittime*	接続がタイムアウトになるまで、呼び出し側が待機する時間
`connectionvalidation` *true/false*	アプリケーションに渡す前に接続を検証するかどうかを指定する
`connectionvalidationmethod` *connectionvalidationmethod*	データベース接続の検証方法。有効な値は `auto-commit`、`meta-data`、`table`
`validationtablename` *validationtablename*	`connectionvalidationmethod` を `table` に設定した場合のテーブル名
`failall` *true/false*	プールの単一の接続が不良であることが確認される場合、そのプールのすべての接続を無効にして再確立するかどうかを指定する
`desc` *description*	プールの説明
`property` *propertyname=value*	JDBC 接続プールの標準プロパティと独自のプロパティを指定する、名前と値のペア

例

```
HttpServerAdmin create -r -jdbcconnectionpool -poolname testpool
-classname "oracle.jdbc.pool.OracleDataSource" -property
"URL=jdbc:oracle:thin:@dbhost:1521:ORCL,user=scott,password=tige
r" -d /opt/Sun/S1WS6.1 -sinst testinstance
```

JDBC リソースの作成

`create -r` コマンドを使用して、コマンド行インタフェースから JDBC リソースを新規作成します。

構文

`HttpServerAdmin -create -r -jdbc -jndiname` *jndiname* `-poolname` *poolname* [`-desc` *description*] [`-enabled` *true*/*false*]

オプション

次の表は、`create -r` コマンドオプションを使用して JDBC リソースを新規作成するときに必要なすべてのオプションを示しています。

表 A-9 JDBC リソース作成のパラメータ

パラメータ	値
`jndiname` *jndiname*	リソースの JNDI 名
`poolname` *poolname*	JDBC 接続プールのプール名
`desc` *description*	プールの説明
`enabled` *true/false*	実行時にリソースを有効にするか、無効にするかを指定する
	JDBC リソースが無効な場合、どのアプリケーションコンポーネントもこのリソースに接続できないが、設定はサーバーインスタンスで維持される

例

```
HttpServerAdmin create -r -jdbc -jndiname "jdbc/testjdbcresource"
-poolname testpool -d /opt/Sun/S1WS6.1 -sinst testinstance
```

カスタムリソースの作成

`create -r` コマンドを使用して、コマンド行インタフェースからカスタムリソースを新規作成します。

構文

`HttpServerAdmin -create -r -custom -jndiname` *jndiname* `-resourcetype` *resourcetype* -factoryclass *factoryclassname* [`-enabled` *true/false*] [`-desc` *description*] [[`-property` *propertyname=value*],...]

オプション

次の表は、`create -r` コマンドオプションを使用して JDBC リソースを新規作成するときに必要なすべてのオプションを示しています。

表 A-10　カスタムリソース作成のパラメータ

パラメータ	値
`jndiname` *jndiname*	リソースの JNDI 名
`resourcetype` *resourcetype*	リソースのタイプ
`factoryclassname` *factoryclassname*	オブジェクトファクトリのクラス名
`enabled` *true/false*	実行時にリソースを有効にするか、無効にするかを指定する
`desc` *description*	プールの説明
`property` *propertyname=value*	カスタムリソースのプロパティを指定する、名前と値のペア

例

```
HttpServerAdmin create -r -custom -jndiname "testcustomresource"
-resourcetype "java.lang.String" -factoryclass
"com.mycom.test.StringFactory" -d /opt/Sun/S1WS6.1 -sinst
testinstance
```

外部 JNDI リソース

`create -r` コマンドを使用して、コマンド行インタフェースから外部 JNDI リソースを新規作成します。

構文

`HttpServerAdmin -create -r -external -jndiname` *jndiname* `-jndilookupname` *jndilookupname* `-restype` *restype* -factoryclass *factoryclass* [`-enabled` *true/false*] [`-desc` *description*] [[`-property` *propertyname=value*],...]

オプション

次の表は、`create -r` コマンドオプションを使用して外部 JNDI リソースを新規作成するときに必要なすべてのオプションを示しています。

表 A-11 外部 JNDI リソース作成のパラメータ

パラメータ	値
`jndiname` *jndiname*	リソースの JNDI 名
`jndilookupname` *jndilookupname*	リソースの JNDI 検索名
`restype` *restype*	リソースのタイプ
`factoryclass` *factoryclass*	オブジェクトファクトリのクラス名
`enabled` *true/false*	実行時にリソースを有効にするか、無効にするかを指定する
`desc` *description*	プールの説明
`property` *propertyname=value*	カスタムリソースのプロパティを指定する、名前と値のペア

例

```
HttpServerAdmin create -r -external -jndiname
"testexternalresource" -jndilookupname "rmiconverter" -restype
"samples.rmi.simple.ejb.ConverterHome" -factoryclass
"com.sun.jndi.cosnaming.CNCtxFactory" -property
"java.naming.provider.url=iiop://localhost:3700" -d
/opt/Sun/S1WS6.1 -sinst testinstance
```

メールリソースの作成

create -r コマンドを使用して、コマンド行インタフェースからメールリソースを新規作成します。

構文

HttpServerAdmin -create -r -mail -jndiname *jndiname* -host *host* -user *user* -from *from* [-storeprotocol *storeprotocol*] [-storeprotocolclass *storeprotocolclass*] [-transportprotocol *transportprotocol*] [-transportprotocolclass *transportprotocolclass*] [-enabled *true/false*] [-desc *description*] [[-property *propertyname=value*]...]

オプション

次の表は、create -r コマンドオプションを使用してメールリソースを新規作成するときに必要なすべてのオプションを示しています。

表 A-12　メールリソース作成のパラメータ

パラメータ	値
jndiname *jndiname*	リソースの JNDI 名
host *host*	メールサーバーのホスト名
user *user*	メールサーバーのユーザー名
from *from*	メールサーバーがメッセージ送信者の指定に使用する電子メールアドレス
storeprotocol *storeprotocol*	メールサーバーへの接続、メッセージの取得、フォルダへのメッセージの保存を行うストレージプロトコルサービス。たとえば、imap や pop3 などの値
storeprotocolclass *storeprotocolclass*	ストレージのサービスプロバイダ実装クラス このクラスは、次のサイトで特定できる • http://java.sun.com/products/javamail/ • http://java.sun.com/products/javabeans/glassgow/jaf.html
transportprotocol *transportprotocol*	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

表 A-12　メールリソース作成のパラメータ (続き)

パラメータ	値
transportprotocolclass *transportprotocolclass*	トランスポートのサービスプロバイダ実装クラス このクラスは、次のサイトで特定できる • http://java.sun.com/products/javamail/ • http://java.sun.com/products/javabeans/glasgow/jaf.html
enabled *true/false*	実行時にこのリソースを有効にするかどうかを指定する。有効な値は、on、off、yes、no、1、0、true、false
desc *description*	リソースの説明
property *propertyname=value*	カスタムリソースのプロパティを指定する、名前と値のペア

例

```
HttpServerAdmin create -r -mail -jndiname "localmail" -host
localhost -user mailid -from mailid@mailhost -d /opt/Sun/S1WS6.1
-sinst testinstance
```

delete コマンド

delete コマンドを使用して、仮想サーバーのクラス、仮想サーバー、および待機ソケットを削除します。

オプション

表 A-13 に示すオプションを delete コマンドとともに使用し、クラス、待機ソケット、および仮想サーバーを削除します。

表 A-13　delete コマンドのオプション

オプション	値
-c	指定された仮想サーバークラスを削除
-l	指定された待機ソケット ID を削除
-v	指定された仮想サーバーを削除

表 A-13　delete コマンドのオプション

オプション	値
-r	指定されたリソースを削除

クラスの削除

delete コマンドのこのオプションを使用して、仮想サーバークラスを削除します。

構文

HttpServerAdmin delete -c -cl *classname* -d *server_root* -sinst *http_instance*

パラメータ

表 A-14 に示すパラメータを delete コマンドとともに使用し、クラスを削除します。

表 A-14　クラス削除のパラメータ

パラメータ	値
-c l *class*	削除するクラス名

例

```
HttpServerAdmin delete -c -cl class1 -d /export/sun/server6
-sinst https-sun.com
```

待機ソケットの削除

delete コマンドのこのオプションを使用して、待機ソケットを削除します。

構文

HttpServerAdmin delete -l -id *listen_socket* -d *server_root* -sinst *http_instance*

パラメータ

表 A-15 に示すパラメータを delete コマンドとともに使用し、クラスを削除します。

表 A-15　クラス削除のパラメータ

パラメータ	値
`-id` *listen_socket*	削除する待機ソケットの ID

例

```
HttpServerAdmin delete -l -id ls3 -d /export/sun/server6 -sinst
https-sun.com
```

仮想サーバーの削除

`delete` コマンドのこのオプションを使用して、仮想サーバーを削除します。

構文

`HttpServerAdmin delete -v -id` *virtual_server* `-cl` *classname* `-d` *server_root* `-sinst` *http_instance*

パラメータ

表 A-16 に示すパラメータを `delete` コマンドとともに使用し、仮想サーバーを削除します。

表 A-16　仮想サーバー削除のパラメータ

パラメータ	値
`-id` *virtual_server*	削除する仮想サーバーの ID
`-cl` *class*	仮想サーバーが属しているクラス

例

```
HttpServerAdmin delete -v -id vs3 -cl class1 -d
/export/sun/server6 -sinst https-sun.com
```

JDBC 接続プールの削除

delete コマンドのこのオプションを使用して、接続プールを削除します。

構文

HttpServerAdmin delete -r *jdbcconnectionpoolname*

パラメータ

表 A-17 に示すパラメータを delete コマンドとともに使用し、接続プールを削除します。

表 A-17　接続ループ削除のパラメータ

パラメータ	値
connectionpoolname	削除する接続プールのプール名

例

```
HttpServerAdmin delete -r connpool
```

JNDI リソースの削除

delete コマンドのこのオプションを使用して、JNDI リソースを削除します。

構文

HttpServerAdmin delete -r *jndiname*

パラメータ

表 A-18 に示すパラメータを delete コマンドとともに使用し、JNDI リソースを削除します。

表 A-18　JNDI リソース削除のパラメータ

パラメータ	値
jndiname	削除するリソースの JNDI 名

例

```
HttpServerAdmin delete -r testresource
```

list コマンド

list コマンドを使用して、仮想サーバーのクラス、仮想サーバー、待機ソケット、リソースをリスト表示します。

構文

HttpServerAdmin list *-command_option* -d *server_root* -sinst *http_instance*

オプション

表 A-19 List コマンドのオプション

オプション	値
-c	すべての仮想サーバークラスをリスト表示する
-l	すべての待機ソケットをリスト表示する
-v	すべての仮想サーバーをリスト表示する
-r	指定されたリソースをリスト表示する

例

```
HttpServerAdmin list -c -d /export/sun/server6 -sinst
https-sun.com

HttpServerAdmin list -l -d /export/sun/server6 -sinst
https-sun.com
```

情報のリストは、コマンドウィンドウに表示されます。

付録 B

Hypertext Transfer Protocol

この付録では、ハイパーテキスト転送プロトコル (Hypertext Transfer Protocol、HTTP) の基本を簡単にご紹介します。HTTP の詳細は、次の「Internet Engineering Task Force (IETF)」ホームページを参照してください。

```
http://www.ietf.org/home.html
```

この付録は、次の節で構成されています。

- HTTP (HyperText Transfer Protocol) について
- 要求
- 応答

HTTP (HyperText Transfer Protocol) について

HTTP (HyperText Transfer Protocol) は、ネットワーク上での情報の交換方法を記述する一連のルールであるプロトコルの 1 つです。このプロトコルによって、Web ブラウザと Web サーバーはヨーロッパ言語用に拡張した ASCII である ISO Latin1 アルファベットを使用して、互いに「対話する」ことができます。

HTTP は、要求 / 応答モデルに基づいています。クライアントはサーバーに接続し、サーバーに要求を送信します。要求には、要求メソッド、URI、およびプロトコルバージョンが含まれています。次に、クライアントはヘッダー情報を送信します。サーバーの応答では、プロトコルバージョン、状態コード、サーバー情報を含むヘッダー、要求されたデータの順に返信されます。このあと接続は終了します。

iPlanet Web Server 4.x は HTTP 1.1 をサポートします。それより前のバージョンのサーバーも HTTP 1.0 をサポートしていました。このサーバーは、IESG (Internet Engineering Steering Group) および IETF (Internet Engineering Task Force) の HTTP ワーキンググループ承認の HTTP 1.1 規格案に、条件付きで準拠しています。条件付き準拠の基準については、次の場所にある「Hypertext Transfer Protocol-HTTP/1.1 specification (RFC 2068)」を参照してください。

```
http://www.ietf.org/html.charters/http-charter.html
```

要求

クライアントからサーバーへの要求に、次の情報が含まれます。

- 要求メソッド
- 要求ヘッダー
- 要求データ

要求メソッド

クライアントは、多数のメソッドを使用して情報を要求することができます。一般的に使われるメソッドには、次のものがあります。

- `GET` : 指定されたドキュメントを要求する
- `HEAD` : ドキュメントのヘッダー情報のみを要求する
- `POST` : CGI プログラムのフォーム入力など、クライアントからのデータ受信をサーバーに対して要求する
- `PUT` : サーバーのドキュメントの内容をクライアントからのデータに置換する

要求ヘッダー

クライアントはヘッダーフィールドをサーバーに送信することができます。ただし、ほとんどのものは任意 (省略可能) です。一般的に使われる要求ヘッダーのいくつかを、表 B-1 に示します。

表 B-1　一般的な要求ヘッダー

要求ヘッダー	説明
`Accept`	クライアントが受信できるファイルの種類

表 B-1　一般的な要求ヘッダー (続き)

要求ヘッダー	説明
Authorization	クライアントがクライアント自身をサーバーに認証させる場合に使用される、ユーザー名、パスワードなどの情報
User-agent	クライアントのソフトウェアの名前とバージョン
Referer	ユーザーがリンクをクリックしたドキュメントの URL
Host	要求されているリソースのインターネットホストとポート番号

要求データ

クライアントが POST または PUT を要求する場合、要求ヘッダーと空白行のあとにデータを送信することができます。クライアントが GET 要求または HEAD 要求を送信する場合、データは送信されず、クライアントはサーバーからの応答を待ちます。

応答

サーバーの応答には次のものが含まれます。

- 状態コード
- 応答ヘッダー
- 応答データ

状態コード

クライアントが要求を送信したとき、サーバーが返信する項目の 1 つに状態コードがあります。これは 3 桁の数字コードです。状態コードには、次の 4 つのカテゴリがあります。

- 100 から 199 までの状態コードは、一時的な応答を示す
- 200 から 299 までの状態コードは、正常に行われたトランザクションを示す
- 300 から 399 までの状態コードは、要求したドキュメントが移動されたために URL を取得できなかった場合に返す
- 400 から 499 までの状態コードは、クライアントにエラーがあることを示す

- 500 以上の状態コードは、サーバーが要求を実行できないか、またはエラーが発生したことを示す

表 B-2 に、一般的な状態コードを示します。

表 B-2　一般的な HTTP 状態コード

状態コード	意味
200	OK。送信は成功しました。これはエラーではありません。
302	見つかりました。新しい URL へ、リダイレクトします。元の URL は移動しました。これはエラーではありません。ほとんどのブラウザでは、新しいページを取得できます。
304	ローカルコピーを使用します。ブラウザのキャッシュにすでにページがあり、そのページが再度要求されている場合、ブラウザ (Netscape Navigator など) の種類によっては、ブラウザのキャッシュされたコピーの「最終更新の (last-modified)」タイムスタンプを Web サーバーに中継することがあります。このサーバー上のコピーがブラウザのコピーよりも古い場合、サーバーは、不要なネットワークトラフィックを減らすために、そのページを返すのではなく 304 コードを返します。これはエラーではありません。
401	承認されていません。ユーザーがドキュメントを要求しましたが、有効なユーザー名およびパスワードが指定されていませんでした。
403	禁止。この URL へのアクセスは禁止されています。
404	見つかりませんでした。要求されたドキュメントがサーバー上にありません。このコードは、承認されていないユーザーにドキュメントが存在しないと伝えることによって、サーバーがドキュメントを保護するよう指示されている場合にも、送信されます。
500	サーバーエラー。サーバーに関連するエラーが発生しました。サーバー管理者がサーバーのエラーログを確認して、何が起こったかを確認する必要があります。

応答ヘッダー

応答ヘッダーには、サーバーに関する情報と、その後にドキュメントに関する情報があります。一般的な応答ヘッダーを、表 B-3 に示します。

表 B-3 一般的な応答ヘッダー

応答ヘッダー	説明
`Server`	Web サーバーの名前とバージョン
`Date`	現在の日付 (グリニッジ標準時)
`Last-modified`	ドキュメントが最後に更新された日付
`Expires`	ドキュメントの有効期限切れの日付
`Content-length`	次に続くデータの長さ (バイト)
`Content-type`	次に続くデータの MIME タイプ
`WWW-authenticate`	認証に使用され、認証に必要な情報 (ユーザー名やパスワードなど) をクライアントのソフトウェアに伝える情報を含む

応答データ

サーバーは最後のヘッダーフィールドの後に空白行を送信します。次に、ドキュメントデータを送信します。

応答

付録 C

ACL ファイルの構文

この付録では、アクセス制御リスト (access-control list、ACL) ファイルとその構文について説明します。ACL ファイルはテキストファイルで、Web サーバー上に格納されるリソースにアクセス可能なユーザーを定義するリストが記述されています。デフォルトでは、Web サーバーはサーバーへのアクセスに関するすべてのリストを含む ACL ファイルを 1 つ使用します。ただし、複数の ACL ファイルを作成し、`obj.conf` ファイルでそれらの参照を指定することもできます。

アクセス制御 API を使用してアクセス制御をカスタマイズする場合、ACL ファイルの構文や関数を知っておく必要があります。たとえば、アクセス制御 API を使用して、Oracle や Informix などの別のデータベースと連動させることができます。API の詳細は、次に示す Sun ONE のドキュメントサイトを参照してください。

```
http://docs.sun.com
```

この付録は、次の節で構成されています。

- ACL ファイルの構文
- obj.conf での ACL ファイルの参照

ACL ファイルの構文

すべての ACL ファイルは、特定の書式と構文に従って記述する必要があります。ACL ファイルは 1 つまたは複数の ACL を含むテキストファイルです。すべての ACL ファイルの先頭に、使用するバージョン番号を記述する必要があります。バージョン行は 1 つだけで、何行のコメント行の後にも指定できます。Sun ONE Web Server 6.1 はバージョン 3.0 を使用します。その例を次に示します。

```
version 3.0;
```

ACL ファイルでは、行の先頭に # を付けてコメントを挿入することができます。

ファイル内の各 ACL は、タイプを定義する文で開始されます。ACL は、次の 3 つのうちのいずれかのタイプに従います。

- **パス ACL** は、影響を受けるリソースへの絶対パスを指定します。
- URI (Uniform Resource Indicator) ACL は、サーバーのドキュメントルートに対して相対的なディレクトリまたはファイルを指定します。
- **名前付き ACL** は、`obj.conf` ファイル内のリソースで参照される名前を指定します。サーバーには、すべてのユーザーに読み取りアクセスを許可する、LDAP ディレクトリのユーザーに書き込みアクセスを許可する「default」という名前が付いたリソースが付属しています。Sun ONE Web Server のウィンドウから名前付き ACL を作成することはできますが、`obj.conf` ファイルで、リソースの名前付き ACL を手動で参照する必要があります。

パス ACL と URI ACL では、エントリの末尾にワイルドカードを付けることができます。たとえば、`/a/b/*` のように記述します。エントリの末尾以外の場所にワイルドカードを指定しても機能しません。

タイプの行は `acl` という文字で始まり、タイプ情報は二重引用符で囲まれ、次にセミコロン (;) が続きます。異なる ACL ファイルの間でも、すべての ACL の各タイプ情報に固有の名前を付ける必要があります。以下の行では、ACL の複数のタイプの例を示します。

```
acl "path=C:/sun/Servers/docs/mydocs/";
acl "default";
acl "uri=/mydocs/";
```

ACL のタイプを定義したあと、ACL で使用されるメソッド (認証文) と、アクセスを許可、または拒否されるユーザーやコンピュータ (承認文) を定義する 1 つまたは複数の文を記述することができます。次の節では、このような文の構文について説明します。

この節では、次の内容について説明します。

- 認証メソッド
- 承認文
- デフォルト ACL ファイル

認証メソッド

ACL では、必要に応じて、サーバーが ACL を処理するときに使用する必要のある認証メソッドを指定します。主に次の 3 つのメソッドがあります。

- 基本 (Basic)。デフォルトの指定
- ダイジェスト (Digest)
- SSL

基本認証メソッドとダイジェスト認証メソッドでは、リソースにアクセスしようとしているユーザーに対してユーザー名とパスワードの入力を要求します。

SSL 認証メソッドでは、ユーザーに対してクライアント証明書を持っていることを要求します。Web サーバーで暗号化を有効にする必要があり、認証された信頼されている CA のリストにユーザーの証明書発行元が表示されている必要があります。

デフォルトでは、サーバーはメソッドを指定しない ACL に対して基本認証メソッドを使用します。サーバーの認証データベースでは、ユーザーから送信されたダイジェスト認証を処理する必要があります。

各認証行では、サーバーが認証を行う属性 (ユーザー、グループ、またはその両方) を指定する必要があります。次に示す ACL タイプ行の後に表示される認証文では、データベースまたはディレクトリ内の各ユーザーと一致するユーザーの基本認証を指定します。

```
authenticate (user) {
     method = "basic";
};
```

次の例では、ユーザーとグループの認証メソッドとして SSL を使用します。

```
authenticate (user, group) {
     method = "ssl";
};
```

次の例では、ユーザー名が `sales` という文字で始まるユーザーを認証します。

```
authenticate (user)

allow (all)

    user = sales*
```

最後の行が `group = sales` に変更された場合、グループの属性が認証されないため、ACL がエラーになります。

承認文

各 ACL エントリには、1 つまたは複数の承認文を指定できます。承認文では、サーバーリソースへのアクセスを許可、または拒否されるユーザーを指定します。承認文を作成する場合、次の構文を使用します。

```
allow|deny [absolute] (right[,right...]) attribute expression;
```

各行の先頭を allow または deny にします。通常の場合、最初の規則ですべてのユーザーに対してアクセスを拒否し、2 番目以降の規則で個別のユーザー、グループ、またはコンピュータに対してアクセスを許可することをお勧めします。これは、規則の階層のためです。すなわち、すべてのユーザーに対して /my_stuff という名前のディレクトリへのアクセスを許可し、サブディレクトリ /my_stuff/personal へのアクセスを一部のユーザーだけに許可する場合、/my_stuff ディレクトリへのアクセスを許可されたユーザーは /my_stuff/personal ディレクトリへのアクセスも許可されるため、サブディレクトリに対するアクセス制御は動作しません。これを避けるには、すべてのユーザーのアクセスを拒否してからアクセスする必要のあるユーザーだけにアクセスを許可する規則をサブディレクトリに対して作成します。

ただし、デフォルトの ACL を設定してすべてのユーザーに対してアクセスを拒否する場合、ほかの ACL 規則では「deny all」規則が必要ありません。

次の行では、すべてのユーザーに対してアクセスを拒否します。

```
deny (all)
    user = "anyone";
```

この節では、次の内容について説明します。

- 承認文の階層
- 属性式
- 式の演算子

承認文の階層

ACL には、リソースに応じて異なる階層があります。たとえば、ドキュメント (URI) /my_stuff/web/presentation.html からの要求を受信する場合、サーバーはこの URI に適用する ACL のリストを構築します。サーバーはまず、obj.conf ファイルの「check-acl」の文にリスト表示される ACL を追加します。次に、一致する URI と PATH ACL を追加します。

サーバー上でも、同じ順序でこのリストを処理します。「無条件な」ACL 文がない場合は、すべての文が順序どおりに評価されます。「無条件に許可」の文または「無条件に拒否」の文が true かどうかを評価する場合、サーバーは処理を停止し、この結果の処理を受け入れます。

一致する ACL が複数ある場合は、一致する最後の文を使用します。ただし、無条件文を使用する場合は、ほかの一致する文の検索を停止し、無条件文のある ACL を使用します。同一のリソースに対する無条件文が 2 つある場合は、ファイル内の最初の文を使用し、一致するほかのリソースの検索を停止します。

```
version 3.0;
acl "default";
authenticate (user,group) {
     prompt="Web Server";
};
allow (read,execute,list,info)
     user = "anyone";
allow (write,delete)
     user = "all";
acl "uri=/my_stuff/web/presentation.html";
deny (all)
     user = "anyone";
allow (all)
     user = "joe";
```

属性式

属性式は、ユーザー名、グループ名、ホスト名、または IP アドレスに基づいて、アクセスを許可、または拒否するユーザーを定義します。次の行は、複数のユーザーまたはコンピュータに対してアクセスを許可する例です。

- `user = "anyone"`
- `user = "smith*"`
- `group = "sales"`
- `dns = "*.sun.com"`
- `dns = "*.sun.com,*.mozilla.com"`
- `ip = "198.*"`
- `ciphers = "rc4"`
- `ssl = "on"`

また、`timeofday` 属性を使用すると、サーバーのローカル時間を基準にした時刻で、サーバーへのアクセスを制御できます。たとえば、`timeofday` 属性を使用すると、特定のユーザーによる特定の時刻のアクセスを制御することができます。

注 時刻を指定するには、24 時間書式を使用します。たとえば、午前 4 時を指定するには 0400、午後 10 時 30 分を指定するには 2230 とします。

次の例では、guests というユーザーのグループによる午前 8 時から午後 4 時 59 分までの間のアクセスを制御します。

```
allow (read)
    (group="guests") and
    (timeofday<0800 or timeofday=1700);
```

また、曜日によってアクセスを制御することもできます。3 文字の省略形、Sun、Mon、Tue、Wed、Thu、Fri、Sat を使用して曜日を指定します。

次の文では、premium グループのユーザーは、曜日や時刻に制限なくアクセスを許可されます。discount グループのユーザーは、週末 (土曜日と日曜日) は時刻に制限なく、平日 (月曜日から金曜日まで) は午前 8 時から午後 4 時 59 分までを除く任意の時間にアクセスできます。

```
allow (read) (group="discount" and dayofweek="Sat,Sun") or
(group="discount" and (dayofweek="mon,tue,wed,thu,fri" and
(timeofday<0800 or timeofday=1700)))
または
(group="premium");
```

式の演算子

属性式では、各種の演算子を使用できます。括弧で演算子の優先度を示します。user、group、dns、ip では、次の演算子を使用できます。

- and
- or
- not
- = (等号)
- != (等しくない)

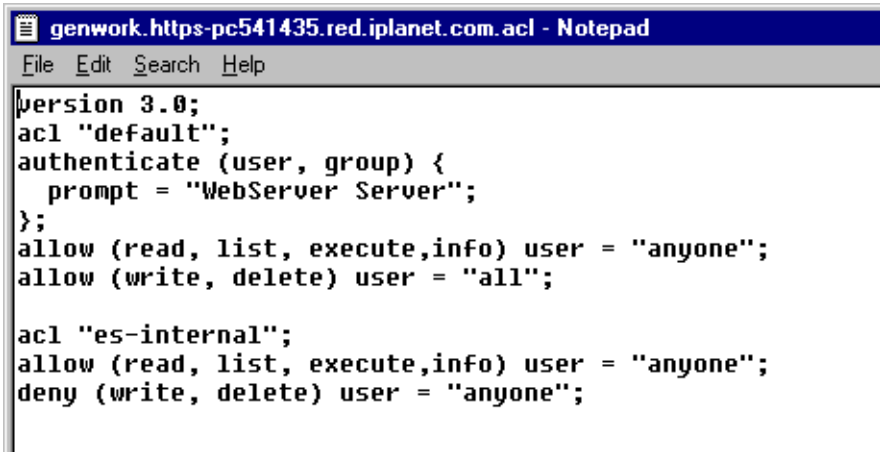
timeofday と dayofweek では、次を使用できます。

- > (より大きい)
- < (より小さい)
- >= (以上)
- <= (以下)

デフォルト ACL ファイル

インストールのあと、*server_root*/httpacl/generated.https-*serverid*.acl ファイルに含まれているサーバーのデフォルト設定を使用できます。ユーザーインタフェースで設定が作成されるまで、サーバーは作業ファイル genwork.https-*serverid*.acl を使用します。ACL ファイルを編集する場合、genwork ファイルに対して変更を加え Sun ONE Web Server を使用して変更を保存して適用します。

genwork ファイル



汎用構文の項目

入力文字列には、次の文字を使用できます。

- a から z までの文字
- 0 から 9 までの数字
- ピリオド (.) と下線 (_)

ほかの文字を使用する場合は、文字を二重引用符で囲む必要があります。

1 つの文は 1 行で表示し、末尾にセミコロンを付けます。複数の文は括弧で囲みます。項目のリストはコンマで区切り、二重引用符で囲む必要があります。

obj.conf での ACL ファイルの参照

名前付き ACL ファイルまたは個別の ACL ファイルがある場合、obj.conf ファイル内で ACL ファイルを参照することができます。このためには、PathCheck 指令で check-acl 関数を使用します。この行には、次の構文があります。

```
PathCheck fn="check-acl" acl="aclname"
```

aclname は、ACL ファイルに表示される、ACL の固有の名前です。

たとえば、testacl という名前の付いた ACL を使用してディレクトリへのアクセスを制限する場合、obj.conf ファイルに次のような行を追加します。

```
<Object ppath="/usr/ns-home/docs/test/*"
PathCheck fn="check-acl" acl="testacl"
</Object
```

上の例では、1 行目が、アクセスを制御するサーバーリソースを示すオブジェクトです。2 行目は PathCheck 指令で、check-acl 関数を使用して名前付き ACL (testacl) を、指令が表示されるオブジェクトにバインドします。testacl ACL は、magnus.conf で参照される ACL ファイルに表示できます。

国際化とローカライズのサポート

Sun ONE Web Server 6.1 の国際化され、ローカライズされたバージョンは、複数の言語および複数のエンコーディングをサポートします。

この付録では、次の主要機能について説明します。

- マルチバイトデータの入力
- 複数文字エンコーディングのサポート
- 言語の設定
- ローカライズされたコンテンツを配信するようにサーバーを設定する

マルチバイトデータの入力

サーバーマネージャまたは管理サーバーのページにマルチバイトデータを入力する場合、次の問題に注意する必要があります。

ファイル名またはディレクトリ名

ファイル名またはディレクトリ名を URL で表示する場合、8 ビット文字やマルチバイト文字は使用できません。

LDAP ユーザーとグループ

電子メール アドレスには、RFC 1700 (`ftp://ds.internic.net/rfc/rfc1700.txt`) で許可された文字のみを使用します。ユーザー ID およびパスワード情報は、ASCII 形式で保存する必要があります。

正しい書式でユーザーとグループの文字情報を入力しているかどうかを確認するには、UTF-8 フォームが有効なクライアント (Netscape Communicator など) を使用して、8 ビットまたはマルチバイトのデータを入力します。

複数文字エンコーディングのサポート

Sun ONE Web Server 6.1 は、次の機能で複数の文字エンコーディングをサポートしています。

- WebDAV
- 検索

WebDAV

Sun ONE Web Server 6.1 は、PROPPPATCH メソッドと PROPFIND メソッドでマルチバイトプロパティの設定と取得をサポートします。要求のエンコーディング形式は問われませんが、サーバーからの応答は常に UTF-8 となります。

検索

Sun ONE Web Server 6.1 は、フルテキストインデックス、および基本となる Java VM が対応しているすべての文字エンコーディングのドキュメント検索をサポートしている Java ベースの検索エンジンを使用します。ドキュメントのデフォルトエンコーディングは、検索コレクションの作成時に指定できます。HTML ドキュメントでは、HTML メタデータからエンコーディングが推測されますが、この推測が不可能な場合はデフォルトエンコーディングが使用されます。

検索インタフェースは JSP タグライブラリに基づいており、あらゆる言語とエンコーディングでカスタマイズおよびローカライズすることができます。タグライブラリのリストは、『Sun ONE Web Server 6.1 Programmer's Guide to Web Applications』に記載されています。詳細は、413 ページの「検索クエリページのカスタマイズ」を参照してください。

言語の設定

エンドユーザー側のエラーメッセージの表示に使用される、サーバーのデフォルト言語は、サーバー設定の「Magnus Editor」を使用して設定できます。Sun ONE Web Server 6.1 のローカライズされたバージョンは、次の 7 言語をサポートします。

- en (英語)
- fr (フランス語)
- de (ドイツ語)
- ja (日本語)
- ko (韓国語)
- zh (簡体字中国語)
- zh_TW (繁体字中国語)

Sun ONE Web Server 6.1 のローカライズされたバージョンのエンドユーザー検索インタフェースは、完全にローカライズされています。

注 この設定は、ローカライズされていない Web サーバーには適用されません。

ローカライズされたコンテンツを配信するようにサーバーを設定する

エンドユーザーは、アクセスするコンテンツの言語設定を説明する Accept-language ヘッダーを送信するようにブラウザを設定できます。サーバー側では、管理サーバーの「Edit Classes」メニューで vs クラスの `acceptlanguage` をオンに設定することで、Accept-language ヘッダーに基づいてコンテンツを配信するように設定できます。この設定により、エンドユーザー側のすべてのエラーメッセージの表示言語も Accept-language ヘッダーに基づいて行われます。

たとえば、`acceptlanguage` を on に設定し、クライアントが次の URL を要求するときに `fr-CH,de` という値を持つ Accept-language ヘッダーを送信するとします。

```
http://www.someplace.com/somepage.html
```

サーバーは、次の順序でファイルを検索します。

1. Accept-language の設定である fr-CH,de

   http://www.someplace.com/fr_ch/somepage.html

   http://www.someplace.com/somepage_fr_ch.html

   http://www.someplace.com/de/somepage.html

   http://www.someplace.com/somepage_de.html

2. 国コードを除いた言語コード (fr-CH の場合は fr)

   http://www.someplace.com/fr/somepage.html

   http://www.someplace.com/somepage_fr.html

3. magnus.conf ファイルに指定されている DefaultLanguage (en など)

   http://www.someplace.com/en/somepage.html

   http://www.someplace.com/somepage_en.html

4. いずれも見つからない場合は、次のようなページ

   http://www.someplace.com/somepage.html

注 ローカライズされたファイルに名前を付けるときは、CH や TW などの国コードは小文字に変換され、ダッシュ (-) はアンダースコア (_) に変換されることに注意してください。

警告 acceptlanguage の設定を有効にすると、サーバーは Accept-language に指定されているすべての言語について前述のアルゴリズムでコンテンツを調べるため、パフォーマンスに影響が生じます。

用語集

admpw　管理サーバーのスーパーユーザーのユーザー名とパスワードを収めたファイル。

CGI　Common Gateway Interface の略。外部プログラムが HTTP サーバーと通信するためのインタフェース。CGI を使用するために記述されたプログラムは、CGI プログラムまたは CGI スクリプトと呼ばれる。CGI プログラムは、サーバーが通常は処理または解析できないフォームの処理または出力の解析を行う。

chroot　サーバーの利用を一定のディレクトリに限定するために作成する補助的なルートディレクトリ。保護されていないサーバーの保護対策としてこの機能を使用する。

ciphertext　暗号化によって隠蔽される情報。あらかじめ決められた受信者のみが復号化できる。

client auth　クライアント認証。

DHCP　Dynamic Host Configuration Protocol の略。システムがネットワーク上の個々のコンピュータに IP アドレスを動的に割り当てることを可能にする Internet Proposed Standard Protocol。

DNS　Domain Name System の略。ネットワーク上のマシンが、198.93.93.10 のような標準 IP アドレスを、`www.sun.com` のようなホスト名に関連付けるための仕組み。各マシンは通常、この変換済みの情報を DNS サーバーから取得するか、またはそれぞれのシステムで管理されているテーブルから検索する。

DNS エイリアス　DNS サーバーが管理している、別のホストを参照するためのホスト名 (具体的には DNS の CNAME レコード)。マシンは必ず 1 つの実名を持つが、1 つまたは複数のエイリアスを持つことができる。たとえば、`www.`*yourdomain.domain* というエイリアスによって、現在サーバーが存在する `realthing.`*yourdomain.domain* という実際のマシンを参照できる (*yourdomain.domain* は実際のドメインを示す)。

Expires ヘッダー　リモートサーバーによって指定される、返されたドキュメントの有効期限。

FORTEZZA　米国政府機関が、要注意ではあるが機密扱いではない情報を管理するために使用する暗号化システム。

FTP　File Transfer Protocol の略。ネットワークを介してコンピュータ間でファイルを転送するためのインターネットプロトコル。

GIF　Graphics Interchange Format の略。CompuServe によって開発された、プラットフォーム間で利用できるイメージ形式。GIF ファイルは通常、他のグラフィックファイルタイプ (BMP、TIFF) よりもかなりサイズが小さい。GIF は、もっとも一般的な画像変換フォーマットの 1 つである。GIF イメージは、UNIX、Microsoft Windows、および Apple Macintosh の各システムでそのまま表示できる。

HTML　Hypertext Markup Language。WWW (World Wide Web) 上のドキュメントに使用する文書整形言語。HTML ファイルは、テキストの表示方法、グラフィックやフォーム項目の配置方法、他のページへのリンクの表示方法などを Netscape Navigator などのブラウザに知らせるための書式設定コードが記述されたプレーンテキストファイル。

HTTP　HyperText Transfer Protocol の略。HTTP サーバーとクライアントの間で情報をやり取りするための手法。

HTTP-NG　次世代の HTTP。

HTTPD　HTTP デーモンまたはサービスの略。HTTP プロトコルを使用して情報を提供するプログラム。Sun ONE Web Server を HTTPD と呼ぶこともある。

HTTPS　セキュリティ機能を備えた HTTP。SSL (Secure Sockets Layer) を使用して実装される。

inittab (UNIX)　何らかの理由で停止した場合に再起動する必要があるプログラムのリストを収めた UNIX ファイル。このファイルにより、プログラムが継続的に稼動することを保証する。ファイルが保存されている場所から、`/etc/inittab` とも呼ばれる。このファイルを使用できない UNIX システムもある。

IP アドレス　Internet Protocol アドレス。ドットで区切られた一連の数字 (たとえば 198.93.93.10 など) で示され、インターネット上のマシンの実際の場所を表す。

ISDN　サービス総合デジタル網 (Integrated Services Digital Network)。

ISINDEX　クライアント内で検索機能を有効にする HTML タグ。ドキュメントでネットワークナビゲータの機能を使用することにより、フォームを使用せずに検索文字列を受け取ってサーバーに送信し、検索可能なインデックスにアクセスできる。<ISINDEX> を使用するには、照会ハンドラを作成する必要がある。

ISMAP　指定されたイメージがイメージマップであることをサーバーに知らせるために HTML で使用される、IMG SRC タグの拡張機能。

ISP　インターネットサービスプロバイダ。インターネット接続を提供する組織。

Java　Sun Microsystems が開発したオブジェクト指向のプログラミング言語。アプレットと呼ばれるリアルタイムの対話型プログラムの作成に使用される。

JavaScript　コンパクトな、オブジェクトベースのスクリプト言語。クライアントサーバーインターネットアプリケーションの開発に使用される。

JavaServer Pages　インスタンス化、初期化、破棄、他のコンポーネントからのアクセス、設定管理など、JavaServer ページのすべてのメタファンクションを使用可能にする拡張機能。JavaServer ページは、Web ブラウザではなく Web サーバーで実行される再利用可能な Java アプリケーションである。

Java サーブレット　インスタンス化、初期化、破棄、他のコンポーネントからのアクセス、設定管理など、Java サーブレットアプリケーションのすべてのメタファンクションを使用可能にする拡張機能。Java サーブレットは、Web ブラウザではなく Web サーバーで実行される再利用可能な Java アプリケーションである。

Last-Modified ヘッダー　サーバーから HTTP 応答で返される、ドキュメントファイルの最終変更日時。

LDAP データベース　認証に使用するためのユーザーとグループのリストが格納されているデータベース。

magnus.conf　主要な Sun ONE Web Server 設定ファイル。このファイルには、サーバーのグローバルな設定情報 (ポート、セキュリティなど) が格納されている。このファイルで、初期化時にサーバー設定変数の値を設定する。Sun ONE Web Server は、起動時にこのファイルを読み取り、変数の設定を実行する。サーバーは、再起動されるまでこのファイルを読み直すことはないため、このファイルに変更を加えた時は、その度にサーバーを再起動する必要があります。

MD5　RSA Data Security を使用したメッセージ要約アルゴリズム。MD5 を使用すると、高い確率で他と重複しない、短く要約したデータを作成できる。計算上、作成される電子メールのメッセージ要約が同一のデータが作成される可能性は非常に低い。

MD5 シグネチャ　MD5 アルゴリズムによって生成されたメッセージ要約。

MIB　マネージメントインフォーメーションベース (Management Information Base)。

MIME　Multi-Purpose Internet Mail Extensions の略。マルチメディアによる電子メールやメッセージングの標準となりつつある。

mime.types　MIME (Multi-purpose Internet Mail Extension) タイプの設定ファイル。このファイルで MIME タイプにファイル拡張子を割り当てることにより、要求された内容の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示し、`.gif` 拡張子を持つリソースの要求は、クライアントが GIF 形式のイメージファイルを要求していることを示します。

modutil　外部の暗号化デバイスまたはハードウェアアクセラレータデバイス用の PKCS#11 モジュールをインストールするのに必要なソフトウェアユーティリティ。

MTA　メッセージ転送エージェント (Message Transfer Agent)。サーバーの MTA Host を、サーバー上のエージェントサービスを使用するように設定する必要がある。

NIS (UNIX)　ネットワーク情報サービス (Network Information Service)。コンピュータのネットワークを通して、マシン、ユーザー、ファイルシステム、およびネットワークパラメータに関する個別情報を収集、照合、および共有するために UNIX マシンで使用するプログラムとデータファイルのシステム。

NNTP　ニュースグループ用の Network News Transfer Protocol。ニュースサーバーホストを、サーバー上のエージェントサービスを使用するように設定する必要がある。

NSAPI　「Server Plug-in API」を参照。

obj.conf　サーバーのオブジェクト設定ファイル。このファイルには、初期化情報、サーバーのカスタマイズに必要な設定、クライアント (ブラウザなど) からの要求を処理する際にサーバーが使用する命令が含まれる。Sun ONE Web Server は、クライアントの要求を処理するたびにこのファイルを読み取る。

pk12util　証明書データベースと鍵データベースを内部のマシンからエクスポートし、外部の PKCS#11 モジュールにインポートするために必要なソフトウェアユーティリティ。

RAM　ランダムアクセスメモリ。コンピュータ内の物理的な半導体メモリ。

rc.2.d (UNIX)　マシンの起動時に実行するプログラムが置かれた、UNIX マシン上のディレクトリ。ディレクトリが存在している場所から、`/etc/rc.2.d` とも呼ばれる。

RFC　Request For Comments の略。一般に、インターネットコミュニティに提示される、手法または標準仕様に関する文書。標準仕様として承認されるまでに、技術に関するコメントを送付できる。

root (UNIX)　UNIX マシンで最高の権限を持つユーザー。root ユーザーは、マシン上のすべてのファイルに対するすべてのアクセス権を持つ。

Server Plug-in API　Sun ONE サーバーのコア機能の拡張やカスタマイズ、および HTTP サーバーとバックエンドアプリケーションの間のインタフェースを構築するための、スケーラブルで効率的なメカニズムの提供を可能にする拡張機能。NSAPI とも呼ばれる。

SOCKS　ファイアウォールの内側と外側の接続を確立するファイアウォールソフトウェア。ファイアウォールのためのソフトウェアまたはハードウェア (ルーター設定など) によって直接の接続が防止されている場合に利用する。

SSL　セキュアソケットレイヤの略。二者 (クライアントとサーバー) の間でセキュリティ保護された接続を確立するソフトウェアライブラリ。HTTPS (セキュリティ機能を備えた HTTP) を実装する場合に使用される。

SSL 認証　本物であることの証明としてクライアント証明書の情報を使用するか、または LDAP ディレクトリに記載されたクライアント証明書を確認することによって、ユーザーがセキュリティ証明書の当人であることを確認する。

strftime　日付および時刻を文字列に変換する関数。トレーラを追加する時にサーバーによって使用される。`strftime` は、日時用の特別な形式の言語を持ち、サーバーはこれをトレーラで使用してファイルの最終変更日を示すことができる。

Sun ONE Web Server 管理コンソール　企業ネットワーク内であればどこでも、中枢となる 1 箇所からすべての Sun ONE サーバーを管理できるグラフィカルインタフェースをサーバー管理者に提供する Java アプリケーション。従来は iPlanet Console と呼ばれていた。インストールした Sun ONE Web Server 管理コンソールのどのインスタンスからでも、自社ネットワーク上にあってアクセス権を与えられているすべての Sun ONE サーバーを表示し、アクセスできる。

Sym-links (UNIX)　シンボリックリンクの略。UNIX オペレーティングシステムで使用されるリダイレクションの種類の 1 つ。Sym-links を利用すると、ファイルシステムのある部分から、そのファイルシステムの別の部分にある既存のファイルまたはディレクトリへのポインタを作成できる。

TCP/IP　Transmission Control Protocol/Internet Protocol の略。インターネットおよび企業内ネットワーク用の主要なネットワークプロトコル。

telnet　ネットワーク上の 2 台のマシンを互いに接続し、リモートログインの端末エミュレーションをサポートするプロトコル。

TLS (Transport Layer Security)　二者 (クライアントとサーバー) の間でセキュリティ保護された接続を確立するソフトウェアライブラリ。HTTPS (セキュリティ機能を備えた HTTP) を実装する場合に使用される。

top (UNIX)　一部の UNIX システム上で使用される、システムリソースの現在の使用状態を表示するプログラム。

uid (UNIX)　UNIX システムで、ユーザーごとに割り当てられる固有番号。

URI　Uniform Resource Identifier の略。省略された URL を使用することによってセキュリティの層を追加するファイル識別子。URL の先頭部分を URL マッピングに置き換えて、ファイルの完全な物理パス名をユーザーから隠す。「URL マッピング」も参照。

URL　Uniform Resource Locator の略。ドキュメントを要求するためにサーバーおよびクライアントで使用されるアドレス指定システム。URL は、場所と呼ばれることもある。URL は、*protocol://machine:port/document* の形式で表される。

URL の例 : `http://www.sun.com/index.html`

URL データベース修復　ソフトウェアの障害、システムクラッシュ、ディスクの故障、ファイルシステムの容量オーバーなどによって壊れた URL データベースを修復して更新するプロセス。

URL マッピング　ドキュメントディレクトリの物理パス名をユーザー定義のエイリアスにマッピングする手法。これにより、ドキュメントディレクトリ内のファイルを参照する時に、完全な物理パス名の代わりに、そのディレクトリのエイリアスを参照するだけでよい。たとえば、`usr/sun/servers/docs/index.html` としてファイルを指定する代わりに、`/myDocs/index.html` のように指定できる。ユーザーがサーバーファイルの物理的な場所を知る必要をなくすことで、サーバーのセキュリティを高めることができる。

WAR (Web Application Archive)　完全な Web アプリケーションを圧縮形式で保管するアーカイブファイル。Sun ONE Web Server は、WAR ファイル内のアプリケーションにはアクセスできない。Sun ONE Web Server が Web アプリケーションにアクセスするには、Web アプリケーション (`wdeploy` ユーティリティを使って配備) の圧縮を解除する必要がある。

Web アプリケーション　サーブレット、JavaServer Pages (JSP)、HTML ドキュメント、およびその他の Web リソース (イメージファイル、圧縮アーカイブ、その他のデータなど) の集まり。Web アプリケーションは、アーカイブ (WAR ファイル) にパッケージ化される場合と、オープンディレクトリ構造に置かれる場合がある。

Windows CGI (Windows)　Visual Basic のような Windows ベースのプログラミング言語で記述された CGI プログラム。

アクセス制御エントリ (Access Control Entries、ACE)　Web サーバーが受信したアクセス要求を評価するために使用する規則の階層。

アクセス制御リスト (Access Control List、ACL)　ACE の集まり。ACL は、サーバーへのアクセス権を持つユーザーを指定するためのメカニズムである。個々のファイルやディレクトリ別に ACL 規則を定義して、単一または複数のユーザーおよびグループに対してアクセスを許可または拒否できる。

暗号化　あらかじめ決められた受信者のみが復号化して読むことができるように情報を変換するプロセス。

暗号化方式　暗号化または復号化に使用する暗号アルゴリズム (数学関数)。

イメージマップ　イメージの各部に機能を持たせて、ユーザーがイメージの各部をマウスでクリックすることによって、ナビゲートや情報の取得を行えるようにする方法。他の HTTPD でイメージマップ機能を扱うために使用する「imagemap」という CGI プログラムを指すこともある。

インテリジェントエージェント　サーバー内で、ユーザーに代わってさまざまな要求 (HTTP、NNTP、SMTP、FTP などの要求) を実行するオブジェクト。たとえば、インテリジェントエージェントは、サーバーに対するクライアントの働きをし、サーバーが遂行する要求を作成する。

エージェント　ルーター、ホスト、X 端末などのネットワークデバイスでネットワーク管理ソフトウェアを実行するソフトウェア。「インテリジェントエージェント」も参照。

エクストラネット　企業のイントラネットをインターネット上に拡張したもの。顧客、仕入先、および遠隔地の従業員がデータにアクセスできる。

仮想サーバー　単一サーバーに複数のドメイン名、IP アドレス、およびサーバー監視機能を設定する手法。

仮想サーバークラス　obj.conf ファイル内の同じ基本設定情報を共有する仮想サーバーの集まり。

管理サーバー　使用するすべての Sun ONE Web Server の設定に使用するフォームを持つ Web ベースのサーバー。

危殆化鍵リスト (Compromised key list、CKL)　危殆化された鍵を持つユーザーに関する鍵情報のリスト。このリストも CA が提供する。

キャッシュ　ローカルに保存されたオリジナルデータのコピー。キャッシュされたデータが要求された時は、再度リモートサーバーから取得する必要がない。

共通ログファイル形式　サーバーがアクセスログに情報を記入する時に使用する形式。Sun ONE Web Server など、すべての主要サーバーで同じ形式が使用される。

クライアント　Netscape Navigator など、WWW (World Wide Web) ページにアクセスして表示するためのソフトウェア。ブラウザプログラムとも呼ばれる。

クラスタ　「マスター」および管理サーバーによって追加、制御されるリモートの「スレーブ」管理サーバーのグループ。クラスタ内のすべてのサーバーは、同じプラットフォームを使用し、同じユーザー ID とパスワードを持つ必要がある。

検索から除外する単語　「ストップワード」を参照。

公開鍵　公開鍵暗号化方式で使用する暗号鍵。

公開情報ディレクトリ (UNIX)　ドキュメントルート内ではなく UNIX ユーザーのホームディレクトリにあるディレクトリ、またはユーザーの制御下にあるディレクトリ。

コレクション　単語リストやファイルのプロパティなど、ドキュメントに関する情報を格納するデータベース。コレクションは、指定した検索条件に該当するドキュメントを取得するために検索を実行する際に使用される。

サーバーデーモン　稼動中は常に、クライアントからの要求を待機して受け取るプロセス。

サーバールート　サーバープログラム、設定ファイル、メンテナンスファイル、および情報ファイルを格納するためのサーバーマシン上のディレクトリ。

サービス品質　サーバーインスタンス、仮想サーバークラス、または仮想サーバーに対して設定するパフォーマンス制限。

証明書　通信者の双方によって信頼済みの第三者が発行する、譲渡や偽造ができないデジタルファイル。

証明書失効リスト (certificate revocation list、CRL)　CA が提供する、破棄された証明書のリスト。

スーパーユーザー (UNIX)　UNIX マシンで使用できる最高の権限を持つユーザー (root とも呼ばれる)。スーパーユーザーは、マシン上のすべてのファイルに対するすべてのアクセス権を持つ。

ストップワード　検索機能で検索しない単語として指定された単語。通常は、the、a、an、and などの単語が含まれる。「検索から除外する単語 (ドロップワード)」とも呼ばれる。

ソフトリスタート　サーバーの再起動方法の 1 つであり、サーバーを内部的に再起動させる、つまり、設定ファイルを再度読み込ませる。ソフトリスタートでは、プロセスに `HUP` 信号 (信号番号 1) が送られる。ハードリスタートとは異なり、プロセス自体は終了しない。

待機ソケット　ポート番号と IP アドレスの組み合わせ。サーバーとクライアントの間の接続は待機ソケットを通して確立されます。

ダイジェスト認証　ユーザー名とパスワードをクリアテキストとして送信せずにユーザーの認証を可能にする。ブラウザは、MD5 アルゴリズムを使用してダイジェスト値を作成する。サーバーは、Digest Authentication プラグインを使用して、クライアントから提供されたダイジェスト値を比較する。

タイムアウト　ハングしたと思われるサービスルーチンを完了させようとする試みを、サーバーが放棄するまでの指定時間。

デーモン (UNIX)　個別のシステムタスクを担当するバックグラウンドプロセス。

ドキュメントルート　サーバーにアクセスするユーザーに表示するファイル、イメージ、およびデータを格納する、サーバーマシン上のディレクトリ。

トップレベルドメイン機関　ホスト名分類の最上位カテゴリ。通常、組織の種類 (`.com` は会社、`.edu` は教育機関など) または所在地の国名 (`.us` は米国、`.jp` は日本、`.au` はオーストラリア、`.fi` はフィンランドなど) を示す。

認証　クライアントが各自の識別情報をサーバーに照合する。基本認証、またはデフォルトの認証では、Web サーバーまたは Web サイトにアクセスするためのユーザー名とパスワードの入力をユーザーに要求する。LDAP データベース内のユーザーおよびグループのリストを必要とする。「ダイジェスト認証」および「SSL 認証」も参照。

サーバー全体、またはサーバー上の特定のファイルおよびディレクトリへのアクセス権を付与すること。ホスト名、IP アドレスなどの条件によって認証を制限することもできる。

認証局 (certification authority、CA)　暗号化トランザクションで使用するデジタルファイルを発行する内部または第三者の組織。

ネットワーク管理ステーション (NMS)　ユーザーがリモートでネットワークを管理するために使用できるマシン。ホスト、ルーター、Sun ONE サーバーなど、SNMP を使用するすべてのデバイスが管理対象となる。NMS は通常、1 つまたは複数のネットワーク管理アプリケーションがインストールされた強力なワークステーションである。

ハードリスタート　プロセスまたはサービスの終了に続く再起動。「ソフトリスタート」も参照。

パスワードファイル (UNIX)　UNIX ユーザーのログイン名、パスワード、およびユーザー ID 番号が格納された、UNIX マシン上のファイル。ファイルが保存されている場所から、`/etc/passwd` とも呼ばれる。

非公開鍵　公開鍵暗号化方式で使用する復号鍵。

ファイアウォール　組織内のネットワークコンピュータを外部のアクセスから保護するネットワーク設定。通常はハードウェアとソフトウェアの両方で設定される。ファイアウォールは一般に、物理的な建物または組織の敷地内におけるネットワークの電子メールやデータファイルなどの情報を保護するために使用される。

ファイル拡張子　ファイル名の最後の部分で、通常はファイルのタイプを表す。たとえば、`index.html` というファイル名のファイル拡張子は `html` である。

ファイルタイプ　ファイルの形式。たとえば、グラフィックファイルには、テキストファイルと同じファイルタイプはない。ファイルタイプは通常、ファイル拡張子 (`.gif`、`.html` など) によって識別される。

プライマリドキュメントディレクトリ　「ドキュメントルート」を参照。

ブラウザ　「クライアント」を参照。

フレキシブルログ形式　サーバーがアクセスログに情報を記入する時に使用する形式。

プロトコル　ネットワーク上のデバイスが情報をやり取りする方法を示す一連の規則。

ホームページ　サーバー上に存在し、そのサーバーのコンテンツのカタログまたはエントリポイントの役割をするドキュメント。このドキュメントの格納場所は、サーバーの設定ファイル内で指定される。

ホスト名　*machine.domain.dom* の形式によるマシン名。この名前が IP アドレスに変換される。たとえば、`www.sun.com` は、`com` ドメイン内の `sun` サブドメインにある `www` というマシンを示す。

リソース　サーバーからアクセスして要求元のクライアントに送信できるドキュメント (URL)、ディレクトリ、プログラムなど。

リダイレクション　特定の URL にアクセスするクライアントを、同じサーバーまたは別のサーバー上の別の場所に転送する仕組み。この仕組みを利用すると、リソースが移動した場合に、クライアントが物理的な場所の移動を意識せずに、新しい場所を使用できる。また、最後のスラッシュを入力せずにディレクトリにアクセスした場合でも、相対リンクを正しく機能させるために使用することもできる。

索引

記号

!= (等しくない), 476
$TOKENNAME, 138
$、ワイルドカードの , 60, 64, 65, 143, 198
%vsid%、ログファイルの書式文字列に追加 , 245
*、ワイルドカードの , 60, 64, 65, 143, 198
= (等号), 476
>= (以上), 476
> (より大きい), 476
?、ワイルドカードの , 60, 64, 65, 143, 198
^、ワイルドカードの , 60, 64, 65, 143, 198
|, 145
~、ワイルドカードの , 60, 64, 65, 143, 198

数字

200 - 500 状態コード , 468
4.x サーバーを 6.0 サーバーに移行する , 49

A

Accept, 466
Accept Language ヘッダー
  使用 , 481
ACL
  obj.conf、参照 , 478
  URI へのアクセス制限 , 212
  アクション、設定 , 203
  アクセス拒否メッセージの変更 , 210
  仮想サーバー , 331
  仮想サーバー、設定 , 342
  仮想サーバーへのアクセス制限 , 227
  仮想サーバー用の設定を編集 , 229
  サーバー全体へのアクセス制限 , 211
  サーバーのダイジェスト認証プロシージャ , 186
  時刻に基づくアクセス制限 , 214
  承認文 , 474
  セキュリティに基づくアクセス制限 , 215
  属性式 , 475
  ディレクトリへのアクセス制限 , 211
  デフォルトファイル , 477
  認証文 , 473
  ファイル、構文 , 471
  ファイルタイプに対するアクセス制限 , 213
  分散管理と , 104
  無効化 , 209
  ユーザーとグループの指定 , 204
.acl
  アクセス制御設定を格納するファイルのファイル拡張子 , 190
ACLCacheLifetime, 190
ACLFILE, 227
-aclid, 453
aclname, 478
ACLUserCacheSize, 190
ACL ユーザーキャッシュ

サーバーがユーザーとグループの認証結果を格納, 190
admin/logs
ログファイルの場所, 105
admpw, 103
スーパーユーザーのユーザー名とパスワードのファイル, 102
AIX
SNMP 関連事項, 274
alias ディレクトリ, 120
allow, 223
and, 476
ansi_x3.4-1968, 381
ansi_x3.4-1986, 381
API リファレンス
JSP, 352
サーブレット, 351
ascii, 381
AuthGroupFile, 222, 224
AuthName, 224
Authorization, 467
AuthTrans qos-handler, 261
AuthType, 225
AuthUserFile, 224

B

bong-file, 149

C

c, 145
CA
種類, 140
承認プロセス (1 日から 2 か月), 118
信頼できる, 119
定義 (認証局), 110
certmap.conf, 143, 185
LDAP 検索, 142
使用, 143
デフォルトのプロパティ, 144
マッピング例, 146
certSubjectDN, 147
CGI, 380
Windows, 364
Windows ディレクトリの指定, 365
Windows でのシェルプログラムのインストール, 367
Windows ファイルタイプの指定, 366
Windows プログラム、概要, 364
インストール, 359
インストールプログラム, 360
概要, 360
仮想サーバー、一意の属性を設定する, 362
仮想サーバーで使用する, 322
シェル, 367
シェルディレクトリの指定、Windows, 368
実行可能ファイルのダウンロード, 363
定義済み (Common Gateway Interface), 349
ディレクトリの削除, 362
ディレクトリの指定, 361
ファイル拡張子, 361
ファイルタイプ, 363
ファイルタイプ、Windows でのシェルの指定, 368
ファイルタイプの指定, 363
プログラム、サーバーへのインストール方法, 350
プログラム、サーバーへの格納方法, 360
CGI (Common Gateway Interface)
概要, 360
CGIStub
CGI の実行を補助するためのプロセス, 360
check-acl, 478
chroot, 155
仮想サーバーにディレクトリクラスを指定する, 155
仮想サーバーにディレクトリを指定する, 156
CKL (危殆化鍵リスト)
インストールおよび管理, 124
ClassCache, 358, 359
classpathsuffix, 288

CmapLdapAttr, 145, 147
cn, 60, 145
common-log, 246
CONFIG, 273, 275
　マスターエージェント、編集 , 277
CONFIG ファイル , 277
contains
　検索タイプのオプション , 65
Content-length, 469
Content-type, 469
cookie
　CGI プログラムを起動できるようにする , 38
　ロギング、簡易 , 246
COPY, 427
cp367, 381
cp819, 381
CRL (証明書の失効リスト)
　インストールおよび管理 , 124
Cron ベースのログローテーション , 244

D

Date, 469
dayofweek, 476
dbswitch.conf, 228
dbswitch.conf ファイル , 205
dcsuffix, 228
defaultclass
　仮想サーバークラス , 316
DELETE, 208
delete アクセス , 208
deny, 223
DES アルゴリズム
　Directory Server の設定 , 188
DES 暗号化方式 , 139
digestauth, 186
DigestStaleTimeout, 187
Directory Server
　DES アルゴリズムの設定 , 188
　ldapmodify コマンド行ユーティリティ , 58
　分散管理に必要な , 103
　ユーザーエントリ , 59
　ユーザーとグループの管理 , 101
DN
　ディレクトリサーバーのエントリ名を表す文字列 , 58
DNComps, 144
DNS
　サーバーパフォーマンスに対する検索の影響を小さくする , 189
-docroot, 453
Domain Name System
　エイリアス、定義 , 483
　定義 , 483

E

e, 145
ends with
　検索タイプのオプション , 65
Error qos-error, 261
execute アクセス , 208
Expires, 469
Expires ヘッダー、定義 , 483

F

FAT ファイルシステム
　セキュリティ (ディレクトリやファイルはアクセス制限ではプロテクトできない), 112
Federal Information Processing Standards (FIPS) -140, 138
FilterComps, 144
FIPS-140
　有効化 , 138
flex_anlg, 251
flexanlg
　使用と構文 , 252
flex-init, 246

flex-log, 246

G

genwork ファイルを示す図 , 477
GET, 208, 466
  SNMP メッセージ , 282
GIF、定義 , 484
givenName, 60
groups-with-users, 221

H

HEAD, 208, 466
home.html, 377
Host, 467
HP OpenView ネットワーク管理ソフトウェア
  SNMP との使用 , 255
.htaccess
  magnus.conf による有効化 , 219
  .nsconfig ファイルからの変換 , 221
  サポートされる指令 , 223
  セキュリティに関する注意事項 , 227
  ダイナミック設定ファイル , 218
  ユーザーインタフェースからの有効化 , 219
  例 , 222
htaccess-register
  独自の認証メソッドを作成するための関数 , 222
htconvert, 221
HTML
  サーバーによる解析、設定 , 384
  定義 , 484
HTML、サーバーで解析
  ファイルキャッシュ , 177
HTTP
  1.1 に準拠 , 466
  応答 , 467
  状態コード , 467
  定義 , 484
  要求 , 466
http_head, 208
httpacl, 190
HTTPD, 484
HTTP (Hypertext Transfer Protocol)
  概要 , 465
HTTPS
  定義 , 484
HttpServerAdmin, 324
  control コマンド , 449
  create コマンド , 450
  delete コマンド , 459
  list コマンド , 463
  仮想サーバーの設定 , 447
  構文 , 448
Hypertext Transfer Protocol (HTTP)
  概要 , 465
Hypertext Transfer Protocol HTTP/1.1 仕様
  URL 参照 , 466

I

ibm367, 381
ibm819, 381
INDEX, 208
index.html, 377
inetOrgPerson、オブジェクトクラス , 59
info アクセス , 208
INIT, 279
init-clf, 246
InitFn, 146
inittab, 112, 169, 171
  サーバーの起動 , 169
  サーバーの再起動 , 170
  定義 , 484
  編集 , 170
iplanetReversiblePassword, 189
iplanetReversiblePasswordobject, 189
IP アドレス
  アクセスの制限 , 182

定義 , 484
IP アドレスとホスト名
指定 , 206
IP アドレスベースの仮想サーバー , 317
is
検索タイプのオプション , 65
ISINDEX, 370
isn't
検索タイプのオプション , 65
iso_646.irv
1991, 381
iso_8859-1, 381
1987, 381
iso-2022-jp, 381
iso646-us, 381
iso-8859-1, 381
iso-ir-100, 381
iso-ir-6, 381
issuerDN, 144
IWS_SERVER_HOME
HttpServerAdmin の実行 , 447
環境変数 , 354
iwsCpuId, 270
iwsCpuIdleTime, 270
iwsCpuIndex, 270
iwsCpuUserTime, 270
iwsInstanceContact, 266
iwsInstanceCount2xx - 5xx, 266
iwsInstanceCountOther, 267
iwsInstanceDeathCount, 266
iwsInstanceDescription, 266
iwsInstanceEntry, 266
iwsInstanceId, 266
iwsInstanceIndex, 266
iwsInstanceInOctets, 266
iwsInstanceLoad15MinuteAverage, 270
iwsInstanceLoad1MinuteAverage, 270
iwsInstanceLoad5MinuteAverage, 270
iwsInstanceLocation, 266
iwsInstanceNetworkInOctets, 270
iwsInstanceNetworkOutOctets, 270
iwsInstanceOrganization, 266
iwsInstanceOutOctets, 266
iwsInstanceRequests, 266
iwsInstanceStatus, 266
iwsInstanceStatusChange, 269
iwsInstanceTable, 266
iwsInstanceUptime, 266
iwsInstanceVersion, 266
iwsKernelTime, 270
iwsListenAddress, 269
iwsListenEntry, 269
iwsListenId, 269
iwsListenIndex, 269
iwsListenPort, 269
iwsListenSecurity, 269
iwsListenTable, 269
iwsProcessConnectionQueueCount, 268
iwsProcessConnectionQueueMax, 269
iwsProcessConnectionQueueOverflows, 269
iwsProcessConnectionQueuePeak, 268
iwsProcessConnectionQueueTotal, 269
iwsProcessEntry, 268
iwsProcessFractionSystemMemoryUsage, 269
iwsProcessId, 268
iwsProcessIndex, 268
iwsProcessKeepaliveCount, 269
iwsProcessKeepaliveMax, 269
iwsProcessSizeResident, 269
iwsProcessSizeVirtual, 269
iwsProcessTable, 268
iwsProcessThreadCount, 268
iwsProcessThreadIdle, 268
iwsThreadPoolEntry, 269
iwsThreadPoolIndex, 269
iwsThreadPoolTable, 269
iwsVsCount200, 268
iwsVsCount2xx - 5xx, 267
iwsVsCount302, 268
iwsVsCount304, 268
iwsVsCount400, 268
iwsVsCount401, 268
iwsVsCount403, 268
iwsVsCount404, 268

iwsVsCount503, 268
iwsVsCountOther, 268
iwsVsEntry, 267
iwsVsId, 267
iwsVsIndex, 267
iwsVsInOctets, 267
iwsVsOutOctets, 267
iwsVsRequests, 267
iwsVsTable, 267

J

J2EE
  Java メールセッション , 291
  JNDI ネーミングサービス , 292
  アプリケーション環境エントリ , 295
  初期ネーミングコンテキスト , 297
  ネーミングサービスとリソース , 289
  ファクトリ、リソースファクトリ , 295
  リソース , 285
  リソースの管理 , 289
J2EE/ サーブレットベースのアクセス制御
  概要 , 88
  用途 , 98
JAAS (Java Authentication and Authorization Service), 86
Java
  Java の有効化と無効化 , 285
  Java メールセッション , 291
  特定仮想サーバーでの有効化 , 286
JavaServer Pages
  概要、インストール方法 , 351
Java Servlet API, 351
JDBC
  JDBC API, 290
  JDBC 接続プールの新規作成 , 299
  JDBC リソースの作成 , 303
  JDBC リソースの設定 , 303
  外部リソースの作成 , 304
  カスタムリソース , 291
  カスタムリソースの作成 , 304
  遮断レベルの保証 , 302
  接続の検証 , 301
    検証方法 , 301
      autocommit, 301
      meta-data, 301
      table, 301
    接続を必ず検証する , 301
    テーブル名 , 301
      すべての接続の無効化 , 302
  接続プール , 290
  通常プールサイズ , 300
  データソース , 290
  データソース名 , 300
  トランザクションの遮断 , 302
    read-committed, 302
    read-uncommitted, 302
    repeatable-read, 302
    serializable, 302
    ダーティリード , 302
  プールの設定 , 300
    アイドルタイムアウト , 301
    最大待機時間 , 301
    最大プールサイズ , 300
    プールのサイズ変更量 , 301
  プール名 , 300
JDBC 接続プール , 290
JDBC 接続プールの新規作成 , 299
JNDI
  JNDI 検索と関連参照 , 294
  JNDI について , 292
  JNDI ネーミングコンテキスト , 293
  接続ファクトリ , 297
  ネーミングサービス , 292
  ネーミング参照 , 294
  ネーミング参照とバインド情報 , 294
  リソース参照名 , 294
JSP
  API リファレンス , 352
  Web サーバーで実行するための要件 , 352
  概要、インストール方法 , 351
  キャッシュディレクトリ , 358
  バージョンファイルの削除 , 358
JSP タグの仕様 , 421

JVM
Java Virtual Machine の設定 , 287
JVM オプションの設定 , 288
JVM のパス設定 , 288
JVM プロファイラの設定 , 289
デバッグオプション , 288
ネイティブライブラリパス , 288

K

keepOldValueWhenRenaming パラメータ , 68

L

Language ヘッダー、Accept
使用 , 481
Last-modified, 469
latin1, 381
LDAP
クライアント証明書をマップする , 142
検索結果、の表 , 142
ディレクトリサービスの設定 , 107
ユーザーインタフェースでのデータベースの指定 , 228
ユーザーとグループの管理 , 53
ユーザー名とパスワードによる認証 , 183, 491
LDAP (Lightweight Directory Access Protocol)
ユーザーとグループの管理 , 53
ldapmodify
Directory Server コマンド行ユーティリティ , 58
Directory Server ユーティリティ , 66
group edit フォームで表示されていない属性値を変更するために使用 , 76
LDAP 検索
certmap.conf を使用する , 142
LDAP 検索フィルタ , 75
LDAP ディレクトリ、アクセス制御 , 205
LDIF
インポートとエクスポート機能について , 57
データベースエントリの追加 , 57
libdigest-plugin.ldif, 187
libdigest-plugin.lib, 187
libnssckbi.sl, 121
libnssckbi.so, 121
Limit, 225
LimitExcept, 225
list アクセス , 208
load-modules, 179
LOCK, 427
log_anly, 251
Look Within ディレクトリ
含まれているユーザーエントリをすべて表示するには , 65

M

magnus.conf, 132
ACLCacheLifetime 指令 , 190
.htaccess の有効化 , 219
起動時のグローバル変数設定 , 174
終了タイムアウト , 168, 187
スレッド制限の調整 , 173
セキュリティの発行 , 132
mail, 60, 145
Manage Servers
サーバーマネージャ、設定変更リスト , 38
Management Information Base (MIB)
場所、Netscape、iPlanet, 265
MaxProcs, 261
MaxThreads, 179
MD5、定義 , 485
memberCertDescriptions, 69
memberURLs, 69
memberURL フィルタ , 69
MIB
場所、Netscape、iPlanet, 265
MIB (management information base)
管理対象オブジェクトの定義 , 264

MIME
charset パラメータ , 381
octet-stream, 364
仮想サーバー、設定 , 341
-mime, 453
MIME (Multi-purpose Internet Mail Extension) タイプ
定義およびページへのアクセス , 175
MIME タイプ
デフォルトの指定 , 378
MIME、定義 , 485
MinThreads, 179
MKCOL, 427
MKDIR, 208
MMappedSessionManager, 359
modutil
PKCS#11 モジュールをインストールする , 134
MortalityTimeSecs, 173
MOVE, 208, 427
MTA
定義 , 486
my_stuff
アクセス制御 , 193

N

NativePool, 178
ndex_page, 357
netscape-http.mib
管理対象オブジェクトと説明 , 266
NIS、定義 , 486
NMS 主導の通信 , 282
NNTP
定義 , 486
nobody ユーザーアカウント , 101
nonce, 187
not, 476
.nsconfig ファイル
.htaccess ファイルへの変換 , 221
nsfc.conf
ファイルキャッシュ設定 , 177
nssckbi.dll, 121
NTFS ファイルシステム
パスワードの保護 , 112

O

o, 145
obj.conf, 107, 246, 472
ACL ファイルの参照 , 478
仮想サーバー , 315
サービス品質使用のための SAF の設定 , 259
スタイルの削除 , 396
デフォルト認証 , 183
octet-stream, 364
OpenView、HP ネットワーク管理ソフトウェア
SNMP ユーザー , 255
or, 476
order, 226
organizationalPerson、オブジェクトクラス , 59
ou, 145

P

password.conf, 112, 172
PathCheck, 218, 220, 478
鍵サイズ制限 , 148
person、オブジェクトクラス , 59
pk12util
証明書と鍵をインポートする , 136
証明書と鍵をエクスポートする , 135
PKCS#11
modutil を使用してインストールする , 134
pk12util で証明書と鍵をインポートする , 136
pk12util で証明書と鍵をエクスポートする , 135
モジュール、追加する , 134
pool パラメータ , 179
POST, 208, 466
pragma no-cache, 154

PROPFIND, 427
PROPPATCH, 427
PROTOCOL_FORBIDDEN, 149
PR_Recv()/net_read, 262
PR_Send()/net_write, 262
PR_TransmitFile, 262
Public Key Cryptography Standard (PKCS) #11
  モジュール、追加する , 134
PUT, 208, 466

Q

qos-error、Error, 261
qos-handler、AuthTrans, 261
QueueSize, 179

R

RAM
  定義 , 486
rc.2.d, 486
  サーバーの起動 , 169
rc.local, 112
read アクセス , 208
Referer, 467
REG_DWORD, 172
REQ_ABORTED, 149
REQ_NOACTION, 149
REQ_PROCEED, 149
require, 226
RequireAuth, 221
RestrictAccess, 221
RMDIR, 208
root
  サーバーと , 100
  定義 , 486
RqThrottleMinPerSocket, 173

S

SAF サンプル
  場所 , 261
sagt, 273
sagt、プロキシ SNMP エージェントを起動するコマンド , 273
schedulerd, 106
secret-keysize, 149
Secure Sockets Layer (SSL)
  暗号化通信プロトコル , 127
Server, 469
server.policy, 98
server.xml, 132, 227, 314
servercertnickname, 138
SessionData, 359
SET
  SNMP メッセージ , 282
SMUX, 270, 274
sn, 60
SNMP
  AIX デーモンの設定 , 274
  GET メッセージと SET メッセージ , 282
  基本 , 263
  コミュニティ文字列 , 281
  コミュニティ文字列、設定 , 281
  サーバーでの設定 , 270
  サーバーの状態をリアルタイムでチェックする , 255
  サブエージェント , 263
  デーモン
    再起動 , 273
  トラップ , 281
  トラップ送信先、設定 , 281
  ネイティブデーモン
    再起動 , 273
    再設定 , 274
  プロキシエージェント , 272
    インストール , 272
    起動 , 273
  プロキシエージェント、インストール , 272
  プロキシエージェント、起動 , 273
  マスターエージェント , 263

インストール , 272, 274, 275
起動 , 279
手動で設定 , 276
マスターエージェント、インストール , 274
マスターエージェント、起動 , 279
snmpd.conf, 274
snmpd、ネイティブ SNMP デーモンを再起動するコマンド , 273
SNMP マスターエージェント
有効化と起動 , 275
SOCKS、定義 , 486
sounds like
検索タイプのオプション , 65
SSL
仮想サーバーで使用する , 321
管理サーバーで有効にする , 127
準備 , 150
定義 , 486
認証 , 185
パラメータ、仮想サーバーの接続グループごとに 1 つのセット , 334
有効化 , 131
有効にするのに必要な情報 , 115
SSL2 プロトコル , 126, 130
SSL3SessionTimeout (SSL)
指令 , 133
SSL3 プロトコル , 126, 130
SSLCacheEntries
指令 (SSL), 133
SSLPARAMS, 132, 138
SSLSessionTimeout (SSL)
セキュリティ指令 , 133
SSL 設定ファイル指令
値を設定する , 132
SSL 認証メソッド , 473
SSL 有効サーバー
自動起動の手順 , 112
st, 145
StackSize, 179
starts with
検索タイプのオプション , 65
stats-xml, 256
stop コマンド
管理サーバーのシャットダウン , 100
sysContact, 277
sysLocation, 277

T

telephoneNumber, 60
telnet, 487
testacl, 478
timeofday, 476
title, 60
TLS
有効化 , 131
tlsrollback, 131
TLS (Transport Layer Security), 127
TLS 暗号化プロトコル , 130
TLS および SSL3 暗号化方式
Netscape Navigator 6.0, 131
TLS プロトコル , 126
Transport Layer Security (TLS)
暗号化通信プロトコル , 127
Triple DES 暗号化方式 , 139

U

uid, 60, 145
定義 , 487
uniqueMembers, 69
UNIX プラットフォーム
管理サーバーへのアクセス , 45
UNLOCK, 428
uri_path, 355, 357
URI、定義 , 487
URL
SSL 有効サーバーと , 132
管理サーバーへのアクセス , 37
定義 , 487
マッピング、定義 , 488

URL 転送
設定 , 379
URL ホストベースの仮想サーバー , 318
us, 381
us-ascii, 381
useradmin
仮想サーバー , 328
User-agent, 467
USERDB, 227
userPassword, 60

V

verifycert, 145
VeriSign
認証局 , 113
VeriSign 証明書
インストール , 114
要求 , 113
vs_port, 355, 357
vs_urlhost, 355, 357

W

WaitingThreads, 173
WAR (Web Application Archive)
定義 , 488
wdeploy ユーティリティ , 354, 488
WebDAV
Sun ONE Web Server によるロック要求の処理 , 440
URI, 424
WebDAV 対応クライアント , 428
WebDAV に必要なアクセス権 , 441
WebDAV 有効リソースへのアクセスの制限 , 441
アクセス制御の有効化 , 441
新しい HTTP ヘッダー , 427
新しい HTTP メソッド , 427
概要 , 423
機能 , 424
共有ロック , 439
コレクション , 425
コレクションとリソースの管理 , 424
コレクションの作成 , 432
コレクションの編集 , 433
セキュリティに関する注意事項 , 442
設定 , 434
URI レベル , 436
仮想サーバークラスレベル , 435
ソース URI, 424
内部メンバー URI, 425
ネームスペースの操作 , 424
排他的ロック , 438
プロパティ , 425
プロパティの操作 , 424
メソッド , 427
COPY, 427
LOCK, 427
MKCOL, 427
MOVE, 427
PROPFIND, 427
PROPPATCH, 427
UNLOCK, 428
メンバー URI, 425
有効化 , 428
仮想サーバークラスレベル , 430
コレクションレベル , 431
サーバーインスタンスレベル , 429
リソースのロックとロック解除 , 438
ロック , 424
ロックの管理 , 439
ロックの最小タイムアウト , 439
ロック要求の例 , 440
WebDAV コレクションの作成 , 432
WebDAV コレクションの編集 , 433
WebDAV 対応クライアント , 428
WebDAV に必要なアクセス権 , 441
WebDAV の設定 , 434
WebDAV の有効化 , 428
webserv61.mib, 266
Web Server へのアクセスを制限する

手順 , 107
Web アプリケーション
定義 , 488
配備 , 354
Web アプリケーションの配備 , 354
Web サーバー
起動と停止 , 167
Web サイト
アクセスの制限 (グローバルとシングルインスタンス), 193
Windows
プログラム、CGI の概要 , 364
Windows CGI, 364
Windows NT プラットフォーム
管理サーバーへのアクセス , 46
write アクセス , 208
WWW-authenticate, 469

X

x509v3 証明書
属性 , 145
x-euc-jp, 381
x-mac-roman, 381
x-sjis, 381

あ

アカウント、ユーザー
変更 , 100
アクセス
delete, 208
execute, 208
info, 208
list, 208
read, 208
Web サイトへ、制限 (グローバルとシングルインスタンス), 193
write, 208
アクセス、サーバー
制限 , 107, 176
アクセス制御
administrators グループ , 104
IP アドレス , 206
LDAP ディレクトリ , 205
my_stuff ディレクトリ , 193
WebDAV, 441
WebDAV 有効リソースへのアクセスの制限 , 441
解除 , 209
概要 , 182
カスタマイズされた式の作成 , 208
仮想サーバーで使用する , 322
拒否の場合の応答 , 210
公開情報ディレクトリ、設定スタイルを使用して制御 , 376
時刻による制限 , 208
説明 , 191
データベース , 205
ファイル , 190
プログラム , 208
分散管理と , 104
分散管理によるアクセス制御のセキュリティ保護 , 216
ホスト名 , 206
ホスト名と IP アドレス , 182
メソッド (基本、SSL), 183
ユーザーおよびグループ , 182, 204
曜日による制限 , 208
リダイレクション , 210
アクセス制御エントリ (ACE), 107, 176, 182
アクセス制御ファイル (ACL)
格納される場所 , 190
アクセス制御リスト (ACL), 107, 176, 182
アクセス、制限
Web Server、手順 , 107
アクセスログ , 245
仮想サーバー、設定 , 343
場所 , 238
アクセスログの詳細設定
設定 , 245

アクセスログファイル , 238, 248
　設定 , 245
　表示 , 105
アクセスログローテーション , 106
アクセラレータ、ハードウェア
　secmod.db に格納された証明書と鍵 , 134
アナライザ、ログ
　実行 (使用する前にサーバーログをアーカイブ), 251
アプリケーション
　クライアントサイド , 349
　サーバーサイド , 349
アプリケーション環境エントリ , 295
アプリケーション、サーバーサイド
　Web サーバーへのインストール方法 , 350
　Web サーバーで実行するタイプ , 350
暗号化
　定義 , 126
暗号化、双方向 , 126
暗号化方式
　Netscape Navigator 6.0 での TLS および SSL3, 131
　オプションを設定する , 148
　定義 , 126
暗号化モジュール、外部
　使用法 , 134

い

移行
　証明書、Enterprise Server 3.x から Web Server 6.0 へ移行する , 120
イベントの表示 , 254
イベントビューア , 254
イベント、表示 (NT), 254
イベント変数
　トラップ , 265
インストール
　CGI プログラム , 359
　複数サーバー , 47

う

受け入れスレッド
　仮想サーバー , 317

え

エージェント
　SNMP, 272
エクストラネット、定義 , 489
エラー
　応答のカスタマイズ , 380
エラー応答、カスタマイズ , 380
エラーログ , 250
　仮想サーバー、設定 , 343
　表示 , 105
　例 , 105
エラーログファイル , 238, 250
　場所 , 238
演算子
　属性式 , 476

お

応答、HTTP, 467
応答データ , 469
応答ヘッダー , 469

か

階層、ACL 承認文 , 474
ガイドライン
　推測しにくいパスワードの作成 , 151
鍵
　pk12util でエクスポートする , 135
　pk12util を使用してインポートする , 136
　定義 , 126

鍵サイズ制限 (obj.conf 内の PathCheck 指令に基づく), 148
鍵データベースパスワード , 112
鍵ペアファイル
  紹介 , 111
  パスワード、変更する , 152
  保護する , 153
カスタムリソース , 291
仮想サーバー , 324
  ACL 設定の編集 , 229
  ACL の設定 , 331, 342
  CGI を使用する , 322
  chroot ディレクトリを指定する , 156
  control コマンド , 449
  create コマンド , 450
  defaultclass, 316
  delete コマンド , 459
  HttpServerAdmin、設定 , 447
  HttpServerAdmin の create コマンドによる作成 , 452
  iWS 4.x バージョンからの移行 , 320
  iWS の機能を使用する , 321
  list コマンド , 463
  MIME の設定 , 341
  obj.conf, 315
  SSL の使用 , 321
  useradmin, 328
  useradmin を使用するように設定する , 330
  Web アプリケーションに含まれていないサーブレットと JSP の配備 , 358
  アクセス制御 , 227
  アクセス制御を使用する , 322
  アクセスログの参照 , 248
  アクセスログ、表示 , 343
  受け入れスレッド , 317
  エラーログの参照 , 250
  仮想サーバーマネージャで設定を変更する , 345
  関連付けるサービス、指定 , 327
  クラスごとに個別の設定情報を持つ , 314
  クラス、作成 , 315, 326
  クラスの設定、編集または削除 , 327
  クラスマネージャで設定を変更する , 340
  公開ディレクトリ、使用するように設定 , 374
  コンテンツ管理 , 320
  サービス品質、設定 , 342
  サービス品質の使用法 , 258
  削除 , 348
  作成 , 339
  作成および編集 , 323
  種類 , 317
  紹介 , 313
  証明書 , 110
  信頼できる別の CA を要求する場合 , 141
  セキュリティ、設定 , 342
  セキュリティの発行 , 132
  接続グループごとに 1 つの SSL パラメータセット , 334
  設定 , 314, 325
  設定スタイルを使用する , 322
  待機ソケット , 316
  ダイナミック再設定 (DR), 324
  追加ドキュメントディレクトリの設定 , 373
  データベースへのアクセス , 228
  デフォルト , 318
  同時接続、サービス品質 , 262
  ドキュメント設定、変更 , 377
  独自の CGI 属性の設定 , 362
  配備 , 332
  複数の Web サーバーの実行 , 47
  変数の使用法 , 324
  ユーザーが監視できるようにする , 328
  例、イントラネットホスティング , 335
  例、セキュリティ保護されたサーバー , 334
  例、デフォルトの構成 , 332
  例、マスホスティング , 337
  ログ、設定 , 343
  ログファイル , 320, 331
仮想サーバークラス
  chroot ディレクトリを指定する , 155
  HttpServerAdmin の create コマンドによる作成 , 450
  サービス品質の使用法 , 258
  スレッドプール , 179
仮想サーバーマネージャ
  UI の概要 , 36

アクセス , 324
管理インタフェース
詳細情報 , 26
管理グループ
作成 , 103
管理サーバー
cron デーモンの起動と停止 , 106
SNMP マスターエージェントの起動 , 279
SSL を有効にする , 127
UI の概要 , 36
URL ナビゲーション , 37
アクセス , 45
インスタンス、Web サーバーの , 36
コントロールパネルのサービスアプレットから起動 , 46
サーバーの削除 , 48
紹介 , 37
セキュリティと , 151
停止 , 99
トップページのメインのタブ , 37
ユーザーエントリ名の変更時に古いフルネームや古い UID 値を削除する方法 , 68
管理サーバーのシャットダウン , 99
管理者
分散管理 , 103
管理者のユーザー ID (スーパーユーザー), 37
管理対象オブジェクト , 264, 282
管理、分散
有効化 , 103
関連項目
管理 , 78

き

起動コマンド
UNIX プラットフォーム , 45
基本認証メソッド , 473
キャッシュ制御指令
設定 , 385
キャッシュ、定義 , 489
キャッシュディレクトリ , 358
共通ログファイル形式
サーバーアクセスログ , 245
定義 , 489
例 , 248
共有ロック , 439

く

クエリ
カスタム構築 , 64
クエリハンドラ
使用 , 370
クライアント
アクセスのリスト , 245
クライアントサイドアプリケーション , 349
クライアント証明書
LDAP へマップする , 142
認証 , 184
クライアント証明書 API
カスタムプロパティを作成する , 146
クライアント認証
定義 , 110
要求するための手順 , 141
クラスタ
管理 , 162
サーバー設定のガイドライン , 158
サーバーの削除 , 162
サーバーの追加 , 160
使用についての定義と見込みタスク , 157
使用のガイドライン , 158
情報の変更 , 161
設定 , 159
変数の追加 , 163
クラスパス
クラスパスを無視する , 288
クラスマネージャ
UI の概要 , 36
アクセス , 39
紹介 , 39
追加タブのリスト , 39

グループ
LDAP データベースにおいてオブジェクトのセットを表現するオブジェクト , 69
アクセスの制限 , 182
エントリの削除 , 78
管理 , 74
グループメンバーリストへ追加 , 77
検索 , 75
削除 , 79
名前の変更 , 79
認証 , 182
認証、ユーザー , 183
編集 , 76
メンバーの追加 , 76
グループ、スタティック
作成のガイドライン , 70
定義 , 69
グループ、ユーザー
について , 56
グローバルセキュリティパラメータ , 132

け

形式、検索オプション
リスト , 64
言語
デフォルト、ユーザーエントリ , 60
検索
JSP タグの仕様 , 421
URI, 399
インタフェースのコンポーネント , 413
仮想サーバーでの検索の無効化 , 400
仮想サーバーでの検索の有効化 , 399
クエリ , 410
検索クエリページのカスタマイズ , 413
検索結果の表示 , 412
検索結果ページのカスタマイズ , 416
検索ページ , 409
検索ページのカスタマイズ , 412
コレクション , 400
SEARCHCOLLECTION エレメント , 402
インデックスの再作成 , 405
エンコーディング , 402, 404, 406
コレクションの更新 , 403
コレクションの削除 , 404
コレクションの作成 , 401
コレクションの設定 , 402
コレクションの保守 , 405
コレクション名 , 401
パターン , 402, 404, 406
表示名 , 401
保守スケジュールの削除 , 408
保守スケジュールの追加 , 406
保守スケジュールの編集 , 407
コレクションの保守スケジュール , 406
最大ヒット数 , 399
詳細検索 , 410
独立したフォームページと結果ページによるカスタマイズ , 420
について , 398
パス , 399
検索クエリ
カスタム、構築 , 64
検索属性オプション
リスト , 64
検索タイプのオプション
リスト , 64
検索のカスタマイズ , 413
検索結果ページのカスタマイズ , 416
独立したフォームページと結果ページのカスタマイズ , 420
検索フィールド
有効エントリ , 63
検索フィルタ
LDAP, 75
検索フィルタ、LDAP
等号 (=) を含む文字列 , 63
検索ベース (ベース DN)
ユーザー ID, 58

こ

公開鍵 , 110, 117

公開情報ディレクトリ
アクセスを制御するための設定スタイルの使用 , 376
公開ディレクトリ
設定 , 374
公開ディレクトリ (UNIX)
カスタマイズ , 374
構文
ACL ファイル , 471
このマニュアルについて
内容 , 27
コマンド行
flexanlg を使用してアクセスログファイルを分析 , 252
コミュニティ文字列
SNMP エージェントが認証に使用する文字列 , 281
コレクション
定義 , 489
コンテンツの圧縮
Vary ヘッダーの挿入 , 387
圧縮レベル , 388
コンテンツのオンデマンド圧縮 , 388
コンテンツを圧縮するための設定 , 386
事前に圧縮されたコンテンツの配信 , 386
フラグメントサイズ , 388
有効化 , 387
コントロールパネル (Windows NT)
管理サーバーのシャットダウンに使用 , 100

さ

サーバー
4.x を 6.0 に移行する , 49
CA の種類 , 140
LDAP ユーザーとグループ、国際化についての考慮事項 , 479
root ユーザー , 100
SNMP を介してリアルタイムでステータスをチェックする , 255
監視で利用可能な統計情報の種類 , 256
起動 , 169, 171
起動時のユーザーアカウント , 100
起動と停止 , 167
クラスタから削除 , 162
コントロールパネルを使用して起動 , 171
再起動 (UNIX), 169
再起動 (Windows), 171
再起動の時間間隔、変更 , 172
削除 , 48
自動的に起動 , 169
手動による再起動 (UNIX), 170
手動による停止 (UNIX), 171
停止 , 171
複数インストール , 47
ポート番号 1024, 100
リモート、クラスタの追加 , 160
ログ (ログアナライザを実行する前にアーカイブ), 251
サーバーアクセス
制限 , 107, 176
サーバーインスタンス
追加 , 48
サーバー、管理
シャットダウン , 99
サーバーサイドアプリケーション , 349
Web サーバーへのインストール方法 , 350
Web サーバーで実行するタイプ , 350
サーバー主導の通信 , 283
サーバー設定
アクセス , 101
サーバーデーモン、定義 , 490
サーバー認証
定義 , 110
サーバーの起動 , 169, 171
必要なユーザーアカウント , 100
サーバーの停止 , 171
サーバー、複数稼動する
仮想サーバーで使用する , 47
サーバーマネージャ
Manage Servers、設定変更リスト , 38
UI の概要 , 36
アクセス , 38

紹介 , 38
スレッド制限の調整 , 173
追加タブのリスト , 38
ログアナライザの実行 (使用する前にサーバーログをアーカイブ), 251
サーバールート、定義 , 490
サービス品質
アプリケーションレベルの HTTP 帯域幅のみを測定 , 261
仮想サーバー、設定 , 342
使用 , 258
使用時の obj.conf の SAF の設定 , 259
同時接続、仮想サーバー , 262
例 , 258
サーブレット
API リファレンス , 351
Web サーバーで実行するための要件 , 352
アクセスの例 , 357
概要、インストール方法 , 351
キャッシュディレクトリ , 358
サーバーへのインストール、方法 , 350
バージョンファイルの削除 , 358
サーブレットと JSP
Web アプリケーションに含まれていない場合の配備 , 358
再起動ユーティリティ、自動 (Windows), 172
再計算の間隔 , 258
削除
Web アプリケーション , 354
作成 , 208
サブエージェント
SNMP, 263
SNMP、使用可能にする , 282

し

シェル CGI, 367
シェルプログラム
CGI のインストール、Windows, 367
時間間隔、サーバーの再起動
変更 , 172
式、カスタム , 208
式、属性
演算子 , 476
識別名
LDAP エントリに証明書をマップする , 142
ユーザーの、フォーム , 59
識別名 (DN) 属性
定義 , 56
システムの実行制御スクリプト
サーバーの再起動 , 170
実行可能ファイル、ダウンロード , 363
自動再起動ユーティリティ (Windows), 172
終了タイムアウト
magnus.conf, 168, 187
設定 , 168
詳細設定、ログ
設定 , 245
状態コード
HTTP, 467
承認文、ACL, 474
証明書
certmap.conf と , 143
Enterprise Server 3.x を Web Server 6.0 に移行する , 120
iPlanet Web Server 4.1 からの移行 , 120
iPlanet Web Server 6.0 からの移行 , 120
pk12util でエクスポートする , 135
pk12util を使用してインポートする , 136
x509v3、属性 , 145
仮想サーバー , 110
管理 , 122
組み込みルート証明書モジュールの使用 , 121
クライアント、LDAP へマップする , 142
クライアントマッピング
例 , 146
種類 , 118
紹介 , 110
信頼できる , 119
待機ソケット用に選択する , 137
他のサーバー、インストールする , 119
他のサーバー証明書を要求する , 116

単一、信頼データベース、Web サーバーインスタンスごと , 141
ルート、削除する , 121
ルート、復元する , 121
証明書、クライアント
認証 , 184
証明書チェーン
定義 , 118
証明書マッピングファイル
certmap.conf の格納場所 , 143
certmap.conf の構文 , 143
証明書要求、必要な情報 , 115
除外する単語 , 489
初期ネーミングコンテキスト , 297
所有者
管理 , 78
指令
SSL3SessionTimeout (SSL), 133
SSLCacheEntries (SSL), 133
SSLSessionTimeout (SSL), 133
シンボリック (ソフト) リンク
定義 , 383
シンボリックリンク、制限 , 383
信頼データベース
Web サーバーインスタンスごとに 1 つの証明書 , 141
外部 PKCS#11 モジュールの証明書を要求またはインストールしたときに自動作成 , 138
作成 , 111
パスワード、変更する , 152
信頼できる証明書 , 119

す

スーパーユーザー
管理者のユーザー ID, 37
分散管理 , 103
スーパーユーザー設定
変更 , 101
スーパーユーザー、定義 , 490
スタイル
設定 , 391
スタイル、設定
作成 , 391
スタティックグループ
作成のガイドライン , 70
定義 , 69
ストップワード , 490
スレッド制限、調整 , 173
スレッドプール
仮想サーバークラス obj.conf の構文 , 179
追加時に指定する情報 , 178

せ

制御、アクセス
概要 , 182
制限、シンボリックリンク , 383
セキュリティ
file レルム , 90
FIPS-140 を有効にする , 138
.htaccess、注意事項 , 227
J2EE/ サーブレットモデルの使用 , 98
LDAP レルム , 90
magnus.conf のグローバルパラメータ , 132
native レルム , 91
solaris レルム , 91
Sun ONE Web Server 6.1 の新機能 , 85
新しい待機ソケットを作成するときに有効にする , 128
新しい待機ソケットを編集するときに有効にする , 128
概要 , 85
カスタムレルム , 91
仮想サーバー、設定 , 342
強化 , 150
グループマッピング , 93
主体マッピング , 92
証明書レルム , 91
セキュリティレルム , 89
デフォルトレルムの指定 , 97

プログラムによるログイン , 97
レルムの設定方法 , 94
ロールと制限対象領域のマッピング , 92
ロールによるアクセス制御の定義 , 92
ロールベースの認証 , 92
セキュリティ指令 , 133
接続グループ
1 つのグループ内のすべての仮想サーバーに対して 1 つの SSL パラメータセット , 334
接続ファクトリ , 297
設定、スーパーユーザー
変更 , 101
設定スタイル , 391
仮想サーバーで使用する , 322
削除 , 396
作成 , 391
編集 , 395
割り当て , 394
割り当ての一覧表示 , 394
設定ファイル
obj.conf, 396
「Restore Configuration」ページでのバックアップコピー , 177
SSL、値を設定する , 132
ダイナミック、使用 , 218
宣言によるセキュリティ , 86

そ

双方向の暗号化、暗号化方式 , 126
ソース URI, 424
属性
x509v3 証明書 , 145
識別名 (DN), 56
属性、検索オプション
リスト , 64
属性式
ACL、属性 , 475
演算子 , 476
測定時間 , 258
組織単位
検索 , 81
削除 , 82
作成 , 80
名前の変更 , 82
編集 , 82
ソフト (シンボリック) リンク
定義 , 383

た

ダイアログボックス
デバッグ
無効化 , 173
待機ソケット
HttpServerAdmin の create コマンドによる作成 , 451
ls1, 174, 316
ls1 (デフォルト待機ソケット), 100
仮想サーバー , 316
証明書名を選択する , 137
セキュリティ機能を有効にする , 128
設定、編集 , 100
テーブル , 174
ダイジェスト認証 , 186
ACL のためのサーバーの処理手順 , 186
ダイジェスト認証プラグイン
インストール , 187
ダイジェスト認証メソッド , 473
ダイナミック再設定 (DR), 324
ダイナミック設定ファイル
使用 , 218
タイムアウト、終了
設定 , 168
多言語についての考慮事項
LDAP ユーザーおよびグループ , 479
単位、組織
検索 , 81
削除 , 82
作成 , 80
名前の変更 , 82

編集 , 82

つ

追加ドキュメントディレクトリ , 373

て

ディレクトリ
追加ドキュメント , 373
ディレクトリサービス
設定 , 107
ディレクトリサービスの詳細設定
設定 , 55, 107
データ、応答 , 469
データベース
仮想サーバーを介してアクセス , 228
データベース、ACL, 205
データベースエントリ
LDIF を使用して追加 , 57
データベース、信頼
作成 , 111
パスワード、変更する , 152
データ、要求 , 467
デーモン
SNMP
再起動 , 273
ネイティブ SNMP、再起動 , 273
ネイティブ SNMP、再設定 , 274
デバッグダイアログボックス
無効化 , 173
デフォルト待機ソケット (ls1), 100

と

統計情報
アクセス , 257
サーバーの監視で利用可能な種類 , 256
サービス品質の帯域幅はサーバーがダイナミックに再設定されると失われる , 262
トラフィック測定の設定 , 258
同時接続
仮想サーバー、サービス品質 , 262
ドキュメント
アクセスされたドキュメントのリスト , 245
ドキュメント設定
インデックスファイル名 , 377
仮想サーバー、設定 , 377
サーバーのホームページ , 378
デフォルトの MIME タイプ、指定 , 378
ディレクトリのインデックス作成 , 378
ドキュメントディレクトリ
コンテンツ発行の制限 , 375
追加 , 373
プライマリ , 319
プライマリ (ドキュメントルート), 372
ドキュメントのルートディレクトリ , 319
設定 , 372
ドキュメントフッター
設定 , 382
ドキュメントルートディレクトリ
chroot を使用してリダイレクトする , 155
ドキュメントルートディレクトリのリダイレクト , 155
トップレベルドメイン機関 , 490
トラップ
SNMP, 281
イベント変数が含まれたメッセージ , 265
トラフィック
設定、統計情報の計算 , 258

な

内部デーモンログローテーション , 243
内部メンバー URI, 425
ナビゲーション
URL 経由での管理サーバーへのアクセス , 37

に

認証
J2EE/ サーブレットモデルの使用 , 98
SSL, 185
クライアント証明書 , 184
グループマッピング , 93
主体マッピング , 92
ホスト名 , 189
ユーザーおよびグループ , 182
ロールと制限対象領域のマッピング , 92
ロールによるアクセス制御の定義 , 92
ロールベースの認証 , 92
認証、基本
SSL 暗号化とホスト - IP 認証のいずれかまたはその両方と組み合わせた場合にもっとも効果的 , 183
認証局
VeriSign, 113
定義 , 110
利用可能な CA のリストを入手する , 115
認証、クライアント
要求するための手順 , 141
認証、クライアント、サーバー
定義 , 110
認証、ダイジェスト , 186
認証データベース , 205
認証文、ACL 構文 , 473
認証、ホスト - IP, 189
認証メソッド
htaccess-register を使用して独自に作成 , 222
種類 , 204
認証、ユーザー - グループ , 183, 189

ね

ネイティブ SNMP デーモン
再起動 , 273
再設定 , 274
ネットワーク管理ステーション (NMS), 263

は

バージョンファイル
削除、JSP とサーブレット , 358
ハードウェアアクセラレータ
secmod.db に格納された証明書と鍵 , 134
ハードリンク、定義 , 383
排他的ロック , 438
配備記述子 , 86
パスワード
作成のガイドライン , 151
パスワードの保護
NTFS ファイルシステム , 112
パスワードファイル , 491
起動時に読み込み , 375
パスワード、ユーザー
変更または作成するには , 66
パフォーマンス
サービス品質の使用法 , 258
ハンドラ、クエリ
使用 , 370

ひ

ビューア、イベント , 254
表示 , 250

ふ

ファイル
certmap.conf, 143
アクセス制御 , 190
ファイル拡張子
CGI, 361
定義 , 491
ファイルキャッシュ
スタティックな情報をすばやく提供、サーバーで解析される HTML の処理速度を向上 , 177
ファイル操作、リモート
有効化 , 376

ファイルタイプ
定義 , 491
ファイルをキャッシングする , 154
フィルタ
memberURL, 69
フォーム、アクセスの制限 , 208
復号化
定義 , 126
プライマリドキュメントディレクトリ、設定 , 319
プライマリドキュメントディレクトリ、設定 (ドキュメントルート), 372
プロキシ SNMP エージェント , 272
インストール , 272
起動 , 273
プロキシエージェント、SNMP, 272
インストール , 272
起動 , 273
プログラム
CGI
サーバーへの格納方法 , 360
アクセス制御 , 208
プログラムによるセキュリティ , 86
プログラムによるログイン , 97
server.policy ファイル , 98
プロトコルデータユニット (PDU), 282
プロパティ
カスタム、作成する , 146
分散管理
Directory Server、必要な , 103
アクセス制御に必要 , 181
グループ
ACL と , 104
有効化 , 103

へ

ヘッダー、応答 , 469
ヘッダー、要求
リスト , 466
変数、イベント
トラップ , 265
変数、グローバル
magnus.conf での設定 , 174

ほ

ポート
セキュリティと , 154
ポート (1024 未満)
サーバーのユーザーを指定する必要はない , 100
保管
ログファイル , 105, 243
ホスト - IP 認証 , 189
ホスト名
アクセスの制限 , 182
定義 , 492
認証 , 189
ホスト名と IP アドレス
指定 , 206

ま

マスターエージェント
CONFIG ファイル、編集 , 277
SNMP, 263
SNMP、インストール , 272, 274, 275
SNMP、起動 , 279
SNMP、手動で設定 , 276
SNMP、使用可能にして起動する , 275
標準以外のポートでの起動 , 279
マスターエージェント、SNMP
インストール , 274
起動 , 279
マルチバイトデータ , 479

め

メンバー URI, 425

も

文字セット
iso_8859-1, 381
us-ascii, 381
変更 , 381
モジュール
PKCS#11、追加する , 134

ゆ

ユーザー
アクセスの制限 , 182
管理 , 62
認証 , 182
ユーザーアカウント
nobody, 101
変更 , 100
ユーザーインタフェース
管理サーバー、サーバーマネージャ、クラスマネージャ、仮想サーバーマネージャ , 36
ユーザーエントリ
Directory Server, 59
検索 , 63
削除 , 68
作成のガイドライン , 58
新規作成 , 59
デフォルト言語 , 60
名前の変更 , 68
名前の変更時に古いフルネームや古い UID 値を削除する方法 , 68
変更 , 66
ユーザーおよびグループ
ACL、指定 , 204
LDAP を使用して管理する , 53
について , 56
ユーザー - グループの認証 , 183, 189
ユーザーとグループの認証
ACL ユーザーキャッシュに格納される結果 , 190
ユーザー認証データベース
dbswitch.conf で定義 , 228
ユーザーの削除 , 68
ユーザーのディレクトリ
設定 , 374
ユーザーのディレクトリ (UNIX)
カスタマイズ , 374
ユーザーパスワード
変更または作成するには , 66
ユーザーライセンス
管理 , 67
ユーティリティ、自動再起動 (Windows), 172

よ

要求
HTTP, 466
要求 - ダイジェスト , 187
要求データ , 467
要求ヘッダー
リスト , 466

ら

ライセンス
管理 , 67
ライブラリ , 146

り

リソース
定義 , 492
リソースのロック
Sun ONE Web Server によるロック要求の処理 , 440
共有ロック , 439
排他的ロック , 438
例 , 440
ロックの管理 , 439
ロックの最小タイムアウト , 439
リソースのワイルドカード

リスト , 198
リソースピッカー
概要 , 41
図 , 41
設定スタイル , 392
ワイルドカード , 41
リダイレクション , 492
リダイレクション (アクセス制御), 210
リモートサーバー
クラスタの追加 , 160
リモートファイル操作
有効化 , 376

る

ルート証明書
削除 , 121
復元する , 121

れ

レルム
file レルム , 90
LDAP レルム , 90
native レルム , 91
solaris レルム , 91
カスタムレルム , 91
証明書レルム , 91
設定方法 , 94
デフォルトレルムの指定 , 97

ろ

ローテーション、アクセスログ , 106
ロギング
cookie、簡易 , 246
ログ
アクセス , 245
ログ、アクセス
場所 , 238
ログアナライザ
flexanlg、使用と構文 , 252
コマンド行から実行 , 251
実行 (使用する前にサーバーログをアーカイブ), 251
ログ、エラー
場所 , 238
表示 , 250
ログファイル
Linux OS での 2G バイトのサイズ制限 , 238
アクセス , 238, 248
エラー , 238, 250
オプションの指定 , 104
仮想サーバー , 320, 331
共通形式 , 245
柔軟な形式 , 245
詳細設定 , 245
設定 , 245
保管 , 105, 243
ログファイル、アクセス
表示 , 105
ログファイルの場所
admin/logs, 105
ログローテーション
Cron ベース , 244
内部デーモン , 243
ロックの最小タイムアウト , 439

わ

ワイルドカード
リソースピッカー , 41
ワイルドカード、リソース
リスト , 198

わ