

管理者ガイド

iPlanet Web Server, Enterprise Edition

Version 6.0 SP1

816-2140-01
2001 年 8 月

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Part No: 816-1379-01

目次

| | |
|---|-----------|
| このマニュアルについて | 19 |
| 内容の紹介 | 19 |
| マニュアルの構成 | 20 |
| 第 1 部：サーバの基本 | 20 |
| 第 2 部：Administration Server の使用方法 | 20 |
| 第 3 部：構成、監視、パフォーマンスの調整 | 21 |
| 第 4 部：仮想サーバとサービスの管理 | 21 |
| 第 5 部：付録 | 22 |
| このマニュアルで使用する表記規則 | 23 |
| iPlanet Web Server のマニュアルの使用 | 23 |
| その他のドキュメント | 25 |
| テクニカルサポートの連絡先 | 25 |
| | |
| 第 1 部 サーバの基本 | 27 |
| | |
| 第 1 章 iPlanet Web Server の概要 | 29 |
| iPlanet Web Server | 29 |
| iPlanet Web Server の特徴 | 30 |
| iPlanet Web Server の管理と運用 | 31 |
| iPlanet Web Server アーキテクチャ | 31 |
| コンテンツエンジン | 32 |
| サーバ拡張機能 | 32 |
| 実行環境 | 33 |
| アプリケーションサービス | 33 |
| iPlanet Web Server の構成 | 34 |
| iPlanet Web Server コンポーネントオプション | 34 |

| | |
|----------------------------|----|
| iPlanet Web Server の構成ファイル | 34 |
| 動的再構成 | 36 |
| 単一サーバの構成 | 36 |
| 全プラットフォーム共通 | 36 |
| UNIX と Linux プラットフォーム | 39 |
| 仮想サーバの構成 | 39 |
| 複数サーバの構成 | 40 |
| Administration Server | 40 |
| Server Manager | 41 |
| リソースピッカーの使用 | 42 |
| リソースピッカーで使用するワイルドカード | 42 |
| Class Manager | 43 |
| Virtual Server Manager | 44 |

| | |
|-------------------------------------|-----------|
| 第 2 章 iPlanet Web Server の管理 | 45 |
| Administration Server へのアクセス | 45 |
| UNIX や Linux プラットフォーム | 45 |
| Windows NT プラットフォーム | 46 |
| 複数のサーバの稼動 | 47 |
| 仮想サーバ | 47 |
| サーバの複数のインスタンスをインストールする | 47 |
| サーバの削除 | 48 |
| 以前のバージョンからのサーバの移行 | 49 |

第 2 部 Administration Server の使用方法 51

| | |
|--|-----------|
| 第 3 章 Administration Server の設定 | 53 |
| Administration Server のシャットダウン | 53 |
| 待機ソケット設定の編集 | 54 |
| ユーザアカウントの変更 (UNIX/Linux) | 55 |
| スーパーユーザ設定の変更 | 56 |
| 複数の管理者の許可 | 57 |
| ログファイルオプションの指定 | 59 |
| ログファイルの表示 | 59 |
| アクセスログファイル | 59 |
| エラーログファイル | 60 |
| ログファイルの保管 | 60 |
| Cron ベースのログローテーションの使用 (UNIX/Linux) | 60 |
| ディレクトリサービスの構成 | 61 |
| サーバへのアクセスの制限 | 62 |
| JRE/JDK パスの構成 | 63 |

| | |
|------------------------------------|-----------|
| 第4章 ユーザとグループの管理 | 65 |
| LDAP を使用してユーザとグループを管理する | 65 |
| 識別名 (DN) の理解 | 66 |
| LDIF の使用 | 66 |
| ユーザの作成 | 67 |
| ユーザエントリ作成のガイドライン | 67 |
| 新規ユーザエントリの作成方法 | 68 |
| Directory Server のユーザエントリ | 68 |
| ユーザの管理 | 70 |
| ユーザ情報の検索 | 70 |
| カスタム検索照会の構築 | 71 |
| ユーザ情報の編集 | 73 |
| ユーザのパスワードの管理 | 73 |
| ユーザライセンスの管理 | 74 |
| ユーザ名の変更 | 74 |
| ユーザの削除 | 75 |
| グループの作成 | 76 |
| 静的グループ | 76 |
| 静的グループ作成のガイドライン | 76 |
| 静的グループを作成するには | 77 |
| グループの管理 | 77 |
| グループエントリの検索 | 77 |
| 「Find all groups whose」フィールド | 78 |
| グループ属性の編集 | 78 |
| グループメンバーの追加 | 79 |
| グループメンバーリストへのグループの追加 | 80 |
| グループメンバーリストからのエントリの削除 | 80 |
| 所有者の管理 | 81 |
| See Also の管理 | 81 |
| グループの削除 | 82 |
| グループの名前の変更 | 82 |
| 組織単位の作成 | 83 |
| 組織単位の管理 | 83 |
| 組織単位の検索 | 83 |
| 「Find all units whose」フィールド | 84 |
| 組織単位の属性の編集 | 84 |
| 組織単位名の変更 | 85 |
| 組織単位の削除 | 85 |
| Preferred Language List の管理 | 86 |
| | |
| 第5章 Web サーバのセキュリティ | 87 |
| 認証の要求 | 88 |
| 認証に証明書を使用する | 88 |

| | |
|--|-----|
| サーバ認証 | 88 |
| クライアント認証 | 88 |
| 仮想サーバ証明書 | 88 |
| 信頼データベースの作成 | 89 |
| 信頼データベースを作成する | 89 |
| password.conf の使用 | 90 |
| SSL 有効サーバを自動的に起動させる | 90 |
| VeriSign 証明書の要求およびインストール | 91 |
| VeriSign 証明書を要求する | 91 |
| VeriSign 証明書をインストールする | 92 |
| 他のサーバ証明書の要求およびインストール | 92 |
| 必要な CA 情報 | 92 |
| 他のサーバ証明書を要求する | 94 |
| 他のサーバ証明書をインストールする | 95 |
| 証明書をインストールする | 96 |
| アップグレード時の証明書の移行 | 98 |
| 証明書を移行する | 98 |
| 組み込みルート証明書モジュールの使用 | 99 |
| 証明書を管理する | 100 |
| CRL と CKL のインストールと管理 | 102 |
| CRL または CKL をインストールする | 102 |
| CRL と CKL の管理 | 103 |
| セキュリティに関する詳細設定 | 103 |
| SSL と TLS プロトコル | 105 |
| SSL を使用して LDAP と通信する | 105 |
| 接続グループのセキュリティを有効にする | 105 |
| セキュリティ機能をオンにする | 105 |
| 接続グループのサーバ証明書を選択する | 107 |
| 符号化方式の選択 | 108 |
| セキュリティをグローバルに構成する | 110 |
| SSLSessionTimeout | 110 |
| SSLCacheEntries | 111 |
| SSL3SessionTimeout | 111 |
| 外部暗号化モジュールの使用 | 111 |
| PKCS#11 モジュールをインストールする | 111 |
| modutil を使用して PKCS#11 モジュールをインストールする | 112 |
| pk12util を使用する | 112 |
| 接続グループの証明書名を選択する | 114 |
| FIPS-140 標準 | 116 |
| クライアントセキュリティの要件を設定する | 117 |
| クライアント認証を要求する | 117 |
| クライアントの認証を要求するには | 118 |
| クライアント証明書を LDAP ヘマップする | 119 |

| | |
|------------------------------|-----|
| certmap.conf ファイルの使用 | 120 |
| カスタムプロパティを作成する | 123 |
| マッピング例 | 123 |
| Stronger Ciphers を設定する | 125 |
| その他のセキュリティに関する問題 | 127 |
| 物理的アクセスを制限する | 127 |
| 管理アクセスを制限する | 128 |
| 確実なパスワードを選択する | 128 |
| 破られにくいパスワードを作成する | 128 |
| パスワードまたは PIN を変更する | 129 |
| パスワードを変更する | 129 |
| サーバ上で他のアプリケーションを制限する | 130 |
| UNIX と Linux | 130 |
| Windows NT | 130 |
| クライアントによる SSL ファイルのキャッシングを防ぐ | 131 |
| ポートを制限する | 131 |
| サーバの限界を知る | 131 |
| サーバを保護するためその他の追加変更を行う | 131 |
| 仮想サーバクラスに chroot を指定する | 132 |
| 仮想サーバに chroot を指定する | 133 |

| | |
|-------------------------|------------|
| 第 6 章 サーバクラスタの管理 | 135 |
| クラスタについて | 135 |
| サーバクラスタの使用に関するガイドライン | 136 |
| クラスタの設定 | 137 |
| クラスタへのサーバの追加 | 138 |
| サーバ情報の変更 | 139 |
| クラスタからのサーバの削除 | 140 |
| サーバクラスタの制御 | 140 |
| 変数の追加 | 141 |

第 3 部 構成、監視、パフォーマンスの調整 143

| | |
|--|------------|
| 第 7 章 サーバの詳細設定 | 145 |
| サーバの起動と停止 | 145 |
| 終了タイムアウトの設定 | 146 |
| サーバの再起動 (UNIX/Linux) | 147 |
| SSL が有効なサーバを自動的に起動 | 148 |
| inittab を使用した再起動 (UNIX/Linux) | 148 |
| 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) | 148 |
| 手動によるサーバの再起動 (UNIX/Linux) | 148 |

| | |
|---|------------|
| 手動によるサーバの停止 (UNIX/Linux) | 149 |
| サーバの再起動 (Windows NT) | 149 |
| 自動再起動ユーティリティの使用 (Windows NT) | 150 |
| サーバのパフォーマンスの調整 | 151 |
| magnus.conf ファイルの編集 | 152 |
| 待機ソケットの追加と編集 | 152 |
| MIME タイプの選択 | 153 |
| アクセスの制限 | 153 |
| 構成の復元 | 154 |
| ファイルキャッシュの構成 | 154 |
| スレッドプールの追加と使用 | 155 |
| ネイティブスレッドプールと汎用スレッドプール (Windows NT) | 155 |
| スレッドプール (UNIX/Linux) | 155 |
| スレッドプールの編集 | 156 |
| スレッドプールの使用 | 156 |
| 第 8 章 サーバへのアクセス制御 | 157 |
| アクセス制御とは | 158 |
| ユーザ - グループのアクセス制御の設定 | 158 |
| デフォルト認証 (Default) | 159 |
| 基本認証 (Basic) | 159 |
| SSL 認証 (SSL) | 160 |
| ダイジェスト認証 (Digest) | 162 |
| その他の認証 (Other) | 165 |
| ホスト - IP のアクセス制御の設定 | 165 |
| アクセス制御ファイルの使用 | 166 |
| ACL ユーザキャッシュの構成 | 166 |
| アクセス制御の実行方法 | 167 |
| アクセス制御の設定 | 169 |
| グローバルなアクセス制御の設定 | 169 |
| サーバインスタンスに対するアクセス制御の設定 | 173 |
| アクセス制御オプションの選択 | 178 |
| アクションの設定 | 178 |
| ユーザとグループの指定 | 178 |
| 「From Host」の指定 | 180 |
| プログラムへのアクセス制限 | 181 |
| アクセス権の設定 | 182 |
| カスタマイズされた式の作成 | 183 |
| アクセス制御の解除 | 183 |
| アクセスが拒否された場合の応答 | 184 |
| サーバの一部へのアクセス制御 | 185 |
| サーバ全体に対するアクセス制限 | 185 |
| ディレクトリ (パス) へのアクセス制限 | 186 |

| | |
|--|------------|
| URI (パス) へのアクセス制限 | 187 |
| ファイルタイプに対するアクセス制限 | 187 |
| 時刻に基づくアクセス制限 | 188 |
| セキュリティに基づくアクセス制限 | 189 |
| 動的アクセス制御ファイルの使用 | 190 |
| .htaccess ファイルの使用 | 190 |
| ユーザインタフェースからの .htaccess の有効化 | 191 |
| magnus.conf からの .htaccess の有効化 | 192 |
| 既存の .nsconfig ファイルの .htaccess ファイルへの変換 | 193 |
| htaccess-register の使用 | 194 |
| .htaccess ファイルの例 | 195 |
| サポートされる .htaccess 指令 | 195 |
| allow | 195 |
| deny | 196 |
| AuthGroupFile | 196 |
| AuthUserFile | 196 |
| AuthName | 197 |
| AuthType | 197 |
| <Limit> | 197 |
| <LimitExcept> | 198 |
| order | 198 |
| require | 199 |
| .htaccess セキュリティに関する注意事項 | 199 |
| 仮想サーバへのアクセス制御 | 200 |
| 仮想サーバからデータベースへのアクセス | 200 |
| ユーザインタフェースでの LDAP データベースの指定 | 201 |
| 仮想サーバのアクセス制御リストの編集 | 202 |
| 第 9 章 ログファイルの使用 | 203 |
| ログファイルについて | 203 |
| アクセスログファイルの参照 | 204 |
| エラーログファイルの参照 | 206 |
| ログファイルの保管 | 207 |
| 内部デーモンログローテーション | 207 |
| Cron ベースのログローテーション | 208 |
| ログの詳細設定 | 209 |
| Cookie を使用した簡易ロギング | 210 |
| ログアナライザの実行 | 211 |
| イベントの表示 (Windows NT) | 213 |
| 第 10 章 サーバの監視 | 215 |
| 統計情報によるサーバの監視 | 216 |

| | |
|--|------------|
| 統計情報を使用可能にする | 216 |
| 統計情報の使用法 | 217 |
| サービス品質の使用法 | 217 |
| サービス品質の例 | 218 |
| サービス品質の設定 | 219 |
| obj.conf で必要な変更 | 220 |
| サービス品質に関する既知の制限事項 | 221 |
| SNMP の基本 | 223 |
| iPlanet Web Server の MIB | 224 |
| SNMP の設定 | 229 |
| プロキシ SNMP エージェントの使用法 (UNIX または Linux) | 231 |
| プロキシ SNMP エージェントのインストール | 231 |
| プロキシ SNMP エージェントの起動 | 232 |
| ネイティブ SNMP デーモンの再起動 | 232 |
| SNMP ネイティブエージェントの再構成 | 233 |
| SNMP マスターエージェントのインストール | 233 |
| SNMP マスターエージェントを使用可能にして起動する | 234 |
| マスターエージェントを別のポートで起動する | 235 |
| SNMP マスターエージェントを手動で構成する | 235 |
| マスターエージェントの CONFIG ファイルの編集 | 235 |
| sysContact 変数と sysLocation 変数の定義 | 236 |
| SNMP サブエージェントの構成 | 237 |
| SNMP マスターエージェントの起動 | 237 |
| 手動による SNMP マスターエージェントの起動 | 237 |
| Administration Server を使用して SNMP マスターエージェントを起動する | 238 |
| SNMP マスターエージェントの構成 | 239 |
| コミュニティ文字列の設定 | 239 |
| トラップ送信先の設定 | 239 |
| サブエージェントを使用可能にする | 240 |
| SNMP メッセージについて | 240 |
| | |
| 第 11 章 サーバのパフォーマンスの調整 | 243 |
| | |
| 第 12 章 検索機能の使い方 | 245 |
| 検索について | 245 |
| テキスト検索の構成 | 246 |
| 検索アクセスの制御 | 247 |
| URL のマッピング | 247 |
| 検索からの単語の除外 | 249 |
| 検索のオンまたはオフ | 250 |
| 検索パラメータの構成 | 251 |
| 検索パターンファイルの構成 | 252 |

| | |
|-----------------------------------|-----|
| 手動によるファイルの構成 | 253 |
| 構成ファイル | 254 |
| 属性の最大数の調整 | 254 |
| インデックス作成に使用するメモリの制限 | 254 |
| インデックスファイルのサイズ制限 | 254 |
| ドキュメントのインデックス作成 | 255 |
| コレクションについて | 255 |
| コレクションの属性について | 257 |
| 新規コレクションの作成 | 258 |
| コレクションの構成 | 261 |
| コレクションの更新 | 263 |
| コレクションの保守 | 264 |
| 定期保守のスケジュール設定 | 265 |
| コレクションの保守スケジュールの削除 | 266 |
| 検索の実行: 基本 | 267 |
| 検索ホームページ | 268 |
| 検索照会 | 268 |
| ガイド付き検索 (Guided Search) | 269 |
| 拡張検索 (Advanced HTML Search) | 270 |
| 検索結果 | 271 |
| 検索条件を満たすドキュメントのリスト表示 | 272 |
| 結果のソート | 272 |
| 強調表示されているドキュメントの閲覧 | 273 |
| コレクションのコンテンツの表示 | 273 |
| 照会演算子の使用 | 274 |
| デフォルトの想定 | 275 |
| 検索規則 | 275 |
| 山括弧 (<>) | 275 |
| 演算子を組み合わせる | 276 |
| 照会演算子を検索語として使用する | 276 |
| 派生語検索の取り消し | 276 |
| 演算子の修飾 | 276 |
| 使用する演算子の決定 | 276 |
| ワイルドカードの使用 | 280 |
| 英数字以外の文字 | 281 |
| 検索インタフェースのカスタマイズ | 281 |
| 動的に生成されるヘッダーとフッター | 282 |
| HTML パターンファイル | 283 |
| 検索関数の構文 | 284 |
| URL エンコーディング | 285 |
| 必要な検索回数 | 286 |
| パターン変数の使用 | 287 |
| ユーザ定義のパターン変数 | 287 |

| | |
|-----------------------|-----|
| 構成ファイルの変数 | 289 |
| マクロと生成されるパターン変数 | 292 |

第 4 部 仮想サーバとサービスの管理 295

| | |
|--|------------|
| 第 13 章 仮想サーバの使用 | 297 |
| 仮想サーバの概要 | 297 |
| 複数のサーバインスタンス | 298 |
| 仮想サーバクラス | 299 |
| obj.conf ファイル | 299 |
| クラスに属する仮想サーバ | 300 |
| デフォルトのクラス | 300 |
| 待機ソケット | 300 |
| 接続グループ | 301 |
| 仮想サーバ | 301 |
| 仮想サーバの種類 | 302 |
| IP アドレスベースの仮想サーバ | 302 |
| URL ホストベースの仮想サーバ | 302 |
| デフォルトの仮想サーバ | 303 |
| 要求を処理する仮想サーバの選択 | 304 |
| ドキュメントルート | 304 |
| ログファイル | 305 |
| 前のリリースから仮想サーバを移行する | 305 |
| 仮想サーバで iPlanet Web Server の機能を使用する | 306 |
| 仮想サーバで SSL を使用する | 306 |
| 仮想サーバでアクセス制御を使用する | 307 |
| 仮想サーバで CGI を使用する | 307 |
| 仮想サーバで構成スタイルを使用する | 307 |
| 仮想サーバのユーザインタフェースの使用法 | 307 |
| Class Manager | 308 |
| Virtual Server Manager | 308 |
| 変数の使用法 | 309 |
| 動的再構成 | 309 |
| 仮想サーバの設定 | 310 |
| 待機ソケットの作成 | 310 |
| 接続グループの作成 | 311 |
| 仮想サーバクラスの作成 | 311 |
| 仮想サーバクラスの編集または削除 | 312 |
| 仮想サーバクラスと関連付けるサービスの指定 | 312 |
| 仮想サーバの作成 | 312 |
| 仮想サーバと関連付ける設定の指定 | 313 |

| | |
|--|------------|
| 個々の仮想サーバをユーザが監視できるようにする | 313 |
| アクセス制御 | 316 |
| ログファイル | 316 |
| 仮想サーバの導入 | 316 |
| 例 1: デフォルトの構成 | 317 |
| 例 2: セキュリティ保護されたサーバ | 319 |
| 例 3: イントラネットホスティング | 320 |
| 例 4: マスホスティング | 323 |
| 第 14 章 仮想サーバの作成と構成 | 325 |
| 仮想サーバの作成 | 325 |
| 仮想サーバの設定内容の変更 | 326 |
| Virtual Server Manager を使用した変更 | 326 |
| 仮想サーバのレポートの生成 | 327 |
| Class Manager を使用した変更 | 329 |
| 仮想サーバの設定内容の変更 | 329 |
| 仮想サーバの MIME の設定 | 330 |
| 仮想サーバの ACL の設定 | 330 |
| 仮想サーバのセキュリティの構成 | 330 |
| 仮想サーバのサービス品質の設定 | 331 |
| 仮想サーバのログの設定 | 332 |
| 仮想サーバの Java Web アプリケーションの設定 | 333 |
| 仮想サーバの削除 | 333 |
| 第 15 章 プログラムによるサーバの拡張 | 335 |
| サーバサイドプログラムの概要 | 335 |
| サーバで実行するサーバサイドアプリケーションのタイプ | 336 |
| サーバへのサーバサイドアプリケーションのインストール方法 | 336 |
| Java サブレットと JavaServer Pages (JSP) | 337 |
| サブレットと JavaServer Pages の概要 | 337 |
| サーバでサブレットや JSP を実行するための要件 | 338 |
| Web アプリケーションの使用 | 339 |
| web-apps.xml ファイルの使用 | 339 |
| wdeploy を使用した Web アプリケーションの導入 | 340 |
| ユーザインタフェースでの Web アプリケーションの導入と編集 | 342 |
| Web アプリケーションになっていないサブレットと JSP の導入 | 344 |
| JVM 属性の構成 | 344 |
| バージョンファイルの削除 | 345 |
| CGI プログラムのインストール | 346 |
| CGI の概要 | 346 |
| CGI ディレクトリの指定 | 348 |
| 各仮想サーバに固有の CGI 属性を構成する | 349 |

| | |
|---|------------|
| ファイルタイプとして CGI を指定 | 349 |
| 実行可能ファイルのダウンロード | 350 |
| Windows NT CGI プログラムのインストール | 351 |
| Windows NT CGI プログラムの概要 | 351 |
| Windows NT CGI ディレクトリの指定 | 352 |
| ファイルタイプとして Windows NT CGI を指定 | 353 |
| Windows NT でのシェル CGI プログラムのインストール | 354 |
| Windows NT でのシェル CGI プログラムの概要 | 354 |
| シェル CGI ディレクトリの指定 (Windows NT) | 354 |
| ファイルタイプとして シェル CGI を指定 (Windows NT) | 355 |
| 照会ハンドラの使用 | 356 |
| | |
| 第 16 章 コンテンツ管理 | 359 |
| プライマリドキュメントディレクトリの設定 | 360 |
| 追加ドキュメントディレクトリの設定 | 361 |
| ユーザ公開情報ディレクトリのカスタマイズ (UNIX/Linux) | 362 |
| コンテンツ発行の制限 | 363 |
| 起動時のパスワードファイル全体の読み込み | 363 |
| 構成スタイルの使用 | 364 |
| リモートファイル操作の有効化 | 364 |
| ドキュメント設定の構成 | 365 |
| ドキュメント設定の変更 | 365 |
| インデックスファイル名の入力 | 365 |
| ディレクトリのインデックス作成を選択 | 366 |
| サーバのホームページの指定 | 366 |
| デフォルト MIME タイプの指定 | 366 |
| Accept-Language ヘッダーの解析 | 367 |
| URL 転送の構成 | 367 |
| エラー応答のカスタマイズ | 368 |
| 文字セットの変更 | 369 |
| ドキュメントのフッターの変更 | 370 |
| htaccess の使用 | 371 |
| シンボリックリンクの制限 (UNIX/Linux) | 371 |
| サーバが解析する HTML の設定 | 372 |
| キャッシュ制御指令の設定 | 373 |
| Stronger Ciphers の使用 | 374 |
| | |
| 第 17 章 構成スタイルの適用 | 375 |
| 構成スタイルの作成 | 375 |
| 構成スタイルの割り当て | 377 |
| 構成スタイルの割り当ての一覧表示 | 378 |
| 構成スタイルの編集 | 378 |

| | |
|-----------------|-----|
| 構成スタイルの削除 | 379 |
|-----------------|-----|

第 5 部 付録 381

| | |
|--|------------|
| 付録 A コマンド行ユーティリティ | 383 |
| LDIF エントリの書式設定 | 383 |
| ldapmodify を使用したデータベースエントリの修正 | 383 |
| HttpServerAdmin (仮想サーバの管理) | 384 |
| HttpServerAdmin の構文 | 384 |
| control コマンド | 385 |
| オプション | 385 |
| 構文 | 386 |
| パラメータ | 386 |
| 例 | 386 |
| create コマンド | 387 |
| オプション | 387 |
| 仮想サーバクラスの作成 | 387 |
| 接続グループの作成 | 388 |
| 待機ソケットの作成 | 389 |
| 仮想サーバの作成 | 390 |
| delete コマンド | 391 |
| オプション | 391 |
| クラスの削除 | 391 |
| 接続グループの削除 | 392 |
| 待機ソケットの削除 | 393 |
| 仮想サーバの削除 | 393 |
| list コマンド | 394 |
| 構文 | 394 |
| オプション | 394 |
| 例 | 394 |
| | |
| 付録 B HTTP (HyperText Transfer Protocol) | 397 |
| ハイパーテキスト転送プロトコル (HTTP) について | 397 |
| 要求 | 398 |
| 要求メソッド | 398 |
| 要求ヘッダー | 398 |
| 要求データ | 399 |
| 応答 | 399 |
| ステータスコード | 399 |
| 応答ヘッダー | 400 |
| 応答データ | 401 |

| | |
|--|------------|
| 付録 C ACL ファイルの構文 | 403 |
| ACL ファイルの構文 | 403 |
| 認証メソッド | 404 |
| 承認文 | 405 |
| 承認文の階層 | 406 |
| 属性式 | 407 |
| 式の演算子 | 408 |
| デフォルト ACL ファイル | 409 |
| 汎用構文の項目 | 409 |
| obj.conf での ACL ファイルの参照 | 410 |
| | |
| 付録 D 国際化された iPlanet Web Server | 411 |
| 一般情報 | 411 |
| サーバのインストール | 411 |
| UTF-8 データの入力 | 412 |
| ファイル名またはディレクトリ名 | 412 |
| LDAP ユーザとグループ | 412 |
| Accept-Language ヘッダーの使用 | 412 |
| その他の言語設定の使用 | 413 |
| 検索情報 | 414 |
| 多言語検索 | 414 |
| 日本語での検索 | 414 |
| 照会演算子 | 414 |
| ドキュメント形式 | 415 |
| 日本語での検索 | 415 |
| サブレットの国際化 | 416 |
| auto | 417 |
| none | 417 |
| 有効な任意のエンコーディング | 418 |
| JSP への送信 | 418 |
| | |
| 付録 E Microsoft FrontPage のサーバ拡張機能 | 419 |
| 概要 | 419 |
| FrontPage Web の種類 | 420 |
| ドメイン名と FrontPage Web | 421 |
| セキュリティについて | 421 |
| 拡張機能のダウンロード | 422 |
| 必要なディスク容量 | 423 |
| 準備作業 | 423 |
| その他の注意事項 | 423 |
| FrontPage Server Extensions のインストール | 424 |
| Windows NT システムに FrontPage Server Extensions をインストールする | 424 |
| UNIX システムまたは Linux システムに FrontPage97 Server Extensions をインストールする | 428 |

| | |
|--|------------|
| UNIX システムまたは Linux システムに FrontPage98 Server Extensions をインストールする . | 432 |
| UNIX システムまたは Linux システムに FrontPage2000 Server Extensions をインストールする . | 433 |
| 詳細情報 | 435 |
| 用語集 | 437 |
| 索引 | 447 |

このマニュアルについて

このマニュアルは、iPlanet™ Web Server, Enterprise Edition 6.0 の構成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、次の節が記述されています。

- 内容の紹介
- マニュアルの構成
- このマニュアルで使用する表記規則
- iPlanet Web Server のマニュアルの使用
- その他のドキュメント
- テクニカルサポートの連絡先

内容の紹介

このマニュアルは、iPlanet Web Server の構成および管理の方法について説明します。サーバの構成が終了したら、このマニュアルはサーバの保守管理に使用します。

サーバのインストール終了後、このマニュアルは、サーバ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、サーバのルートディレクトリは、`C:\iPlanet\Servers\` または `/usr/iplanet/servers` になります。

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。iPlanet Web Server, Enterprise Edition 6.0 をはじめて使用する場合、製品の概要を理解するために第1部「サーバの基本」から始めます。すでに iPlanet Web Server のこのバージョンに慣れている場合、第1部「サーバの基本」の部分にはざっと目を通すのみにし、第2部「Administration Server の使用方法」へ進みます。

Administration Server の基本的な使用方法を理解したら、第3部「構成、監視、パフォーマンスの調整」を参照します。ここには、iPlanet Web Server の構成や監視の方法についての例があります。第4部「仮想サーバとサービスの管理」には、プログラムの使用方法と構成スタイルに関する情報が提供されています。

最後に、第5部「付録」では、次の項目を含むさまざまな事項についてのリファレンスを提供しています。ハイパーテキスト転送プロトコル (HyperText Transfer Protocol, HTTP)、サーバ構成ファイル、ACL ファイル、国際化に関する問題、サーバ拡張、iPlanet Web Server ユーザ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：サーバの基本

第1部では、iPlanet Web Server の概要を説明します。次の章が記述されています。

- 第1章「iPlanet Web Server の概要」では、iPlanet Web Server の概要を説明します。
- 第2章「iPlanet Web Server の管理」では、Administration Server による iPlanet Web Server の運用方法を説明します。

第2部：Administration Server の使用方法

第2部では、iPlanet Web Server を管理する Administration Server 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「Administration Server の設定」では、iPlanet Web Server を構成するため、Administration Server の Preferences と Global Settings フォームの使用方法を説明します。
- 第4章「ユーザとグループの管理」では、iPlanet Web Server を構成するため、Administration Server の Users and Groups フォームの使用方法を説明します。

- 第5章「Web サーバのセキュリティ」では、iPlanet Web Server のセキュリティの構成方法を説明します。この章を読む前に、公開鍵暗号法と SSL 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、符号化方式、SSL 接続 (ハンドシェイク) の主な手順が含まれます。
- 第6章「サーバクラスタの管理」では、サーバのクラスタ化の概念を解説し、その使用方法と各サーバ間で構成を共有する方法について説明します。

第3部：構成、監視、パフォーマンスの調整

第3部では、Server Manager を使用して iPlanet Web Server を構成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7章「サーバの詳細設定」では、iPlanet Web Server のサーバ設定の構成方法を説明します。
- 第8章「サーバへのアクセス制御」では、サーバの一部へアクセス可能なユーザを指定する方法を説明します。
- 第9章「ログファイルの使用」では、ログファイルの記録や表示、およびオペレーティングシステムに付属のパフォーマンス監視ツールの使用により、ハイパーテキスト転送プロトコル (Hypertext Transfer Protocol、HTTP) を使用して iPlanet Web Server を監視する方法を説明します。
- 第10章「サーバの監視」では、SNMP (Simple Network Management Protocol) を使用して iPlanet Web Server を監視する方法を説明します。
- 第11章「サーバのパフォーマンスの調整」では、次の Web サイトに収録されているオンラインドキュメント「Performance Tuning and Sizing Guide for iPlanet Web Server」を紹介しています。
<http://docs.iplanet.com/docs/manuals/enterprise.html> (英語)
- 第12章「検索機能の使い方」では、サーバにあるコンテンツの検索方法およびドキュメント属性の検索方法を説明します。さらにこの章では、ユーザ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も説明します。

第4部：仮想サーバとサービスの管理

第4部では、プログラムや構成スタイルについて Server Manager 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「仮想サーバの使用」では、iPlanet Web Server を使用して、仮想サーバをセットアップし管理する方法について説明します。

- 第 14 章「仮想サーバの作成と構成」では、個々の仮想サーバの生成と構成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、その他のプラグインをサーバにインストールする方法を説明します。
- 第 16 章「コンテンツ管理」では、サーバのコンテンツを構成したり管理したりする方法を説明します。
- 第 17 章「構成スタイルの適用」では、iPlanet Web Server を使用しての、構成スタイルの使用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、ユーザインタフェース画面の代わりに使う、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。
- 付録 B「HTTP (HyperText Transfer Protocol)」では、いくつかの HTTP の基本概念について、簡単に紹介します。
- 付録 C「ACL ファイルの構文」では、ACL (アクセス制御リスト) ファイルとその構文について説明します。
- 付録 D「国際化された iPlanet Web Server」では、iPlanet Web Server の国際化版について説明します。
- 付録 E「Microsoft FrontPage のサーバ拡張機能」では、Microsoft FrontPage のサポートを提供する iPlanet Web Server における、サーバの拡張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、iPlanet Web Server 管理者にはあまりなじみがないかもしれないものについて解説しています。

このマニュアルで使用する表記規則

このマニュアルで使用する表記規則は、次のとおりです。

斜体 (*Italic*)

この書体は、強調、および可変部分 (お使いのシステムに合わせて置き換える必要のある部分) に使用します。たとえば、URL でサーバのポート番号を示す場合、この URL では *portnumber* のようにポート番号を斜体で表記しています。斜体の語は、サーバの実際の値と置き換えてください。

クーリエ (Courier) フォント

この書体は、ユーザが入力する必要があるテキスト、および関数、例、URL、ファイル名、ディレクトリパスなどに使用します。

iPlanet Web Server のマニュアルの使用

次の表に、iPlanet Web Server の印刷版 (注を参照) マニュアルやオンラインの README ファイルに説明されているタスクと概念を一覧記載します。特定のタスクを達成しようとしたら、また、特定の概念についてもっと詳しく知りたい場合は、該当するマニュアルを参照してください。

| | |
|----------|--|
| 注 | 印刷版のマニュアルは、PDF や HTML 形式のオンラインファイルとしても提供しています。 |
|----------|--|

表 1 iPlanet Web Server のマニュアル

| 次に関する情報は | 参照先 |
|----------------------------|---|
| ソフトウェアとマニュアルの最新情報 | http://docs.iplanet.com |
| iPlanet Web Server のインストール | iPlanet Web Server の『インストールと移行』 |

表 1 iPlanet Web Server のマニュアル (続き)

| 次に関する情報は | 参照先 |
|---|---|
| <p>次のタスクを実行するために、Administration Server を使用してサーバの管理および構成を行い、1 つまたは複数の iPlanet Web Server を管理する</p> <ul style="list-style-type: none"> サーバのセキュリティを設定する ログファイル、SNMP、または OS 付属のツールにより、HTTP を使用してサーバを監視する パフォーマンスのニーズに合わせて、サーバの作業負荷を定義し、システムサイズを指定する 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にインストールする コンテンツやサーバのドキュメントの属性を検索する：テキスト検索インタフェースを生成する | <p>iPlanet Web Server の『管理者ガイド』</p> |
| <p>管理サーバと、暗号化、アクセス制御、パフォーマンス監視などの内容に関するグローバル情報</p> | <p>『Managing Servers with iPlanet Console』</p> |
| <p>ディレクトリサービスの計画。2 ～ 3 百人のユーザといくつかの主要なサーバアプリケーションからなる簡単なシステムをサポートするためのディレクトリサーバの使用法について、また、何百万人もユーザをサポートするためのディレクトリサーバを拡張する方法について。また、ディレクトリサービスの基本的な概念についてと、本稼動用レベルのディレクトリサービスの導入に必要な特定のガイドラインについても紹介します。</p> | <p>iPlanet Directory Server の『導入ガイド』</p> |
| <p>クライアントの要求に応じてコンテンツを動的に生成したりサーバのコンテンツを変更したりするために iPlanet Web Server の拡張および変更に使用することができる、プログラミングテクノロジーおよび API の概要。オンライン版では、各 API について述べているそれぞれのマニュアルへのリンクが提供されています。このマニュアルは、iPlanet Web Server, Enterprise Edition 6.0 の開発者レベルの情報のスターティングポイントとして使用してください。また、このマニュアルでは構成ファイルの目的と用途についても説明し、これらの構成ファイルで使用可能な指令および関数の包括的なリストも提供しています。</p> | <p>iPlanet Web Server の『プログラマーズガイド』</p> |
| <p>iPlanet Web Server でのサーブレットおよび JavaServer Pages (JSP) の有効化と実装方法</p> | <p>iPlanet Web Server の『サーブレットに関するプログラマーズガイド』</p> |

表 1 iPlanet Web Server のマニュアル (続き)

| 次に関する情報は | 参照先 |
|---|--|
| iPlanet Web Server を拡張および変更するためにプラグインを構築するための、NSAPI (Netscape Server Application Programming Interface) の使用方法。また、このマニュアルでは NSAPI 関数のリファレンスを提供しており、新規のプラグインの定義に使用できます。 | iPlanet Web Server の『NSAPI プログラマーズガイド』 |

その他のドキュメント

iPlanet のドキュメントのサイトには、管理者、ユーザ、および開発者向けのドキュメントが提供されています。次の文書も含まれています。

- iPlanet Web Server のリリースノート
- 『Netscape Internet Service Broker programmer's guides』、Java および C++ に関するリファレンスガイド

これらのドキュメントにアクセスするには、次の URL を使用します。

<http://docs.iplanet.com> (英語)

テクニカルサポートの連絡先

テクニカルサポートについては、次の、iPlanet Web Server の Technical Support のページを参照してください。

<http://www.iplanet.com/support/> (英語)

テクニカルサポートの連絡先

サーバの基本

第 1 章 「iPlanet Web Server の概要」

第 2 章 「iPlanet Web Server の管理」

iPlanet Web Server の概要

この章では、iPlanet Web Server を紹介し、いくつかのサーバの基本的な概念を説明します。この章を読み、iPlanet Web Server の機能の概要を理解してください。

この章には、次の節が記述されています。

- iPlanet Web Server
- iPlanet Web Server アーキテクチャ
- iPlanet Web Server の構成
- Administration Server
- Server Manager
- Class Manager
- Virtual Server Manager

iPlanet Web Server

iPlanet Web Server, Enterprise Edition 6.0 は、マルチプロセス、マルチスレッドの Web サーバで、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、次の内容について説明します。

- iPlanet Web Server の特徴
- iPlanet Web Server の管理と運用

iPlanet Web Server の特徴

iPlanet Web Server は、主に、業務用途の HTML ファイルにアクセスするために設計されています。加えて、次のような機能を提供します。

- **企業全体での管理性** : 代理管理機能、クラスタ管理、LDAP (Lightweight Directory Access Protocol) のサポートを含みます。ディレクトリサーバに LDAP が統合されているため、ユーザやグループを1つのディレクトリに集中させて保存できます。さらに、*Simple Network Management Protocol (SNMP)* を使用してサーバをリアルタイムに監視できます。SNMP は、ネットワークアクティビティに関するデータを交換するのに使用するプロトコルの1つです。iPlanet Web Server にユーザやグループを追加するには、iPlanet Directory Server のようなディレクトリサーバを事前にインストールしている必要があります。詳細は、iPlanet Web Server の『インストールと移行』を参照してください。
- **セキュリティ** : ユーザは、Secure Sockets Layer (SSL) 3.0 プロトコルを使用することにより、クライアントとサーバ間で、暗号化され、認証されたトランザクションを確立できます。さらに、iPlanet Web Server は、次のセキュリティベースの標準を採用しています。Public Key Cryptography Standard (PKCS) #11 (SSL と PKCS #11 モジュール間の通信用インタフェースを定義する)、Federal Information Processing Standards (FIPS)-140、およびクライアントの機能に応じて、56、128、または 168 ビットで動作する特別な証明書です。
- **アクセス制御** : ユーザ名、パスワード、ドメイン名、IP アドレスによる、アクセス制御 (表示、編集、バージョン制御) を実装することで、機密のファイルやディレクトリを保護することができます。この機能はまた、NSAPI Content Management プラグインにも利用できます。つまり、管理者に依頼しなくても、エンドユーザ (ドキュメント所有者) 本人が、ドキュメントにアクセス制御を自分で設定できるということです。
- **高性能** : HTTP1.1 やマルチスレッド対応、SSL ハードウェアアクセラレータのサポートなどの機能を備え、動的コンテンツに高い性能を提供します。
- **標準ベース** : iPlanet Web Server は、次のような、広範囲な Web ソフトウェア標準をサポートしています。JDK 1.2、Servlets 2.2、JavaServer Pages 1.1、HTTP 1.1 などです。また、PKCS #11、FIPS-140、168 ビットセットアップ証明書などを含むさまざまなセキュリティベースの標準もサポートしています。
- **サーバサイド Java サーブレットおよび JavaServer Pages のサポート** : サーバプラグインの開発や、動的なコンテンツ、プレゼンテーションロジック、JDBC データベースアクセスを可能にします。
- **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。

iPlanet Web Server の管理と運用

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

- iPlanet Web Server Administration Server
- Server Manager
- Class Manager
- Virtual Server Manager

以前のリリースでは、本サーバやその他の Netscape サーバは、Administration Server という単一のサーバで管理されていました。4.x リリースでは、「administration server」は、iPlanet Web Server の単なる追加のインスタンスの 1 つとなり、iPlanet Web Server Administration Server または Administration Server と呼ばれます。Administration Server を使用して、iPlanet Web Server のインスタンスをすべて管理することになります。詳細は、40 ページの「Administration Server」を参照してください。

注 構成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。

iPlanet Web Server の各インスタンスの管理については、Server Manager を使用します。詳細は、41 ページの「Server Manager」を参照してください。

仮想サーバを管理するには Class Manager を使用します。詳細は、39 ページの「仮想サーバの構成」を参照してください。

iPlanet Web Server アーキテクチャ

iPlanet Web Server は、モジュール方式のアーキテクチャを採用しており、iPlanet ファミリー系のサーバ全製品をシームレスに統合しています。さらに、iPlanet Web Server には、管理サーバインタフェースが組み込まれており、各 Web サーバ間の管理機能を調整できます。この管理機能のインタフェース自体も、iPlanet Web Server の別個のインスタンスであるということを認識しておいてください。

iPlanet Web Server には、次のソフトウェアモジュールが含まれます。

- コンテンツエンジン
- サーバ拡張機能
- 実行環境
- アプリケーションサービス

サーバモジュールについては、以降の節で説明しています。

コンテンツエンジン

iPlanet Web Server コンテンツエンジンは、カスタマのデータを処理するために開発されました。iPlanet Web Server アーキテクチャの Web Publishing レイヤーは、次の3つのコンテンツエンジンで構成されています。HTTP (Web Server)、コンテンツ管理、および検索エンジンです。

HTTP エンジンは、iPlanet Web Server のコアとなるものです。機能的に見れば、iPlanet Web Server アーキテクチャのその他の部分は、この HTTP エンジンを基盤に性能や統合の機能を発揮します。

コンテンツ管理エンジンを使用して、サーバコンテンツの管理が可能です。HTML ページ、JavaServer Page、その他のグラフィックス、テキスト、サウンド、ビデオなどのファイルを作成し、サーバに保存します。アクセス権があれば、クライアントはサーバに接続してこれらのファイルを閲覧することができます。

検索エンジンにより、iPlanet Web Server のユーザは、サーバに格納されている文書の内容や属性を検索することが可能です。サーバ管理者の場合、HTML、Microsoft Word、Adobe PDF、WordPerfect のようなさまざまな形式のドキュメントの検索に対応した、カスタマイズしたテキスト検索インタフェースを作成できます。iPlanet Web Server は、さまざまなタイプの非 HTML 文書を HTML に変換しインデックスを付けるため、ユーザは、検索したドキュメントを Web ブラウザを使って閲覧できます。

サーバ拡張機能

iPlanet Web Server 拡張機能により、サーバの機能を拡張したり交換したりして、業務内容や規模に最適なシステムを構築できます。iPlanet Web Server のコアアーキテクチャとなる、サーバ拡張機能の一部を次に示します。

- CGI (Common Gateway Interface)
- NSAPI (Netscape Server Application Programming Interface)
- Java サブレットおよび JavaServer Page

CGI (Common Gateway Interface) は、スタンドアロンのアプリケーション開発インタフェースで、クライアントの要求を動的に処理するプログラムを作成できます。

NSAPI (Netscape Server Application Programming Interface) は、要求処理時にサーバが呼び出す関数 (サーバアプリケーション関数) を実装するのに使用します。これにより、iPlanet Web Server のコアとなる機能および拡張機能が提供されます。NSAPI は、サーバが要求を処理する際、要求を細かいステップに分割し、さらにそれらをさまざまな方法で整理することで、高速処理と柔軟な構成を可能にします。

Java サブレットおよび JavaServer Pages 拡張機能により、Java サブレットおよび JavaServer Page のメタ機能が有効になります。インスタンス化、初期化、破棄、その他のコンポーネントからのアクセス、構成管理などが使用可能です。Java サブレットおよび JavaServer Page は、再利用が可能な Java アプリケーションで、Web ブラウザではなく Web サーバで実行されます。

実行環境

さまざまなサーバ拡張機能に加えて、iPlanet Web Server には一連の実行環境も用意されており、サーバ拡張機能をサポートします。実行環境には次のものが含まれます。

- CGI プロセッサ
- NSAPI エンジン
- Java 仮想マシン (JVM)

アプリケーションサービス

最後に、iPlanet Web Server アーキテクチャには、各種のアプリケーション固有の機能のための、一連のアプリケーションサービスが含まれています。アプリケーションサービスには次のものが含まれます。

- セキュリティとアクセス制御
- セッション管理サービス
- ファイルシステムサービス
- メールサービス

iPlanet Web Server の構成

iPlanet Web Server の構成を変更することで、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、サーバで稼動しサーバとの対話を行うプログラムを書いたりすることが可能となります。これらのオプションを示す命令（指令と呼ばれる）は、構成ファイルに保存されます。iPlanet Web Server は、起動時やクライアントからの要求時に、この構成ファイルを読み込み、目的のサーバの動作に合わせてユーザの選択をマップします。構成ファイルについての詳細は、34 ページの「iPlanet Web Server の構成ファイル」を参照してください。

サーバはコンピュータにインストールされた時点で、複数の構成ファイルを取り込みます。構成ファイルの保存場所は、`server_root/https-server_id/config` および `server_root/https-admserv/config` です。

この節では、次の内容について説明します。

- iPlanet Web Server コンポーネントオプション
- iPlanet Web Server の構成ファイル
- 単一サーバの構成
- 複数サーバの構成

iPlanet Web Server コンポーネントオプション

次のコンポーネントオプションが、iPlanet Web Server をインストールすると使用可能になります。

- iPlanet Web Server コア
- JRE (Java Runtime Environment)
- Java とサーブレット

iPlanet Web Server の構成ファイル

iPlanet Web Server には、さまざまな構成ファイルがあるため、多様なグローバル変数を設定したり、特定のイベントやクライアント要求に対するサーバの応答方法をカスタマイズしたりすることが可能です。構成ファイルの変更は、Administration Server、Server Manager、Class Manager のユーザインタフェースを使用して自動的に行うか、または、テキストエディタでファイルを直接編集することにより行えます。

主な iPlanet Web Server の構成ファイルは次のとおりです。magnus.conf、obj.conf、mime.types、server.xml、および admpw です。主な構成ファイルは、この節で説明します。

magnus.conf: このファイルには、グローバルサーバ構成情報（セキュリティやデフォルトの言語選択など）が格納されています。このファイルで、初期化時にサーバを構成する変数の値を設定します。iPlanet Web Server は起動時に、このファイルを読み込んで変数設定を実行します。サーバは、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、その度にサーバを再起動する必要があります。

詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照してください。

obj.conf: オブジェクト構成ファイル。仮想サーバクラスごとにまたは仮想サーバグループごとに、1つの obj.conf ファイルがあります。このマニュアルで「obj.conf ファイル」という場合、それは、すべての obj.conf ファイルを意味するか、または、説明する仮想サーバクラス用の obj.conf ファイルのどちらかを意味します。すべての obj.conf ファイルの保存場所は、*server_root/server_id/config* です。ファイル名は、通常、*vsclass.obj.conf* となります。vsclass は、仮想サーバクラス名です。

obj.conf ファイルには、サーバのカスタマイズの設定と、クライアント（ブラウザなど）からの要求に対するサーバの処理方法を指示したものが格納されます。各仮想サーバは、クライアントの要求を処理する毎に、このファイルを読み出します。

obj.conf と magnus.conf 構成ファイルが使用する実際のファイル構文や特定の指令についての詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照してください。

server.xml: このファイルは、サーバの待機するアドレスとポートを設定し、仮想サーバクラスや仮想サーバをこれらの待機ソケットに割り当てます。詳細は、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照してください。

mime.types: MIME (Multi-purpose Internet Mail Extension) タイプの構成ファイル。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。

詳細は、第 16 章「コンテンツ管理」の 366 ページの「デフォルト MIME タイプの指定」を参照してください。このファイルを変更した場合は、その都度、サーバを再起動する必要があります。

admpw: Administration Server のスーパーユーザのユーザ名とパスワードのファイル。詳細は、第 3 章「Administration Server の設定」の 56 ページの「スーパーユーザ設定の変更」を参照してください。

動的再構成

動的再構成を利用すると、変更を有効にするために Web サーバを停止および再起動することなく、稼働中の Web サーバの構成変更を行うことができます。server.xml およびその関連ファイル内の構成に関する設定および属性のすべてを、サーバを再起動することなく動的に変更できます。

動的再構成 (DR) の画面を表示し新規の構成を動的にインストールするには、Server Manager、Class Manager、および Virtual Server Manager 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。新しい構成を組み込む際にエラーがある場合は、前の構成が復元されます。

単一サーバの構成

iPlanet Web Server を 1 台のサーバマシンにインストールする場合、インストールプロセスにより、インストール時に指定したサーバのルートディレクトリにすべてのファイルが保存されます。

全プラットフォーム共通

すべてのプラットフォームについて、次のディレクトリがサーバのルートディレクトリに作成されます。

- **alias** には、iPlanet servers の鍵および証明書のファイルがあります。
(例: https-admserv-server_id-cert7.db および secmod.db)
- **bin** には、サーバのバイナリファイルがあります。実際のサーバや Administration Server フォームなどのバイナリファイルです。さらに、このディレクトリには https/install フォルダが作成され、サーバの設定の移行に必要なファイルや、下位互換に必要なデフォルトの構成ファイルが格納されます。
- **docs** は、サーバのデフォルトプライマリドキュメントディレクトリで、ここに、通常、サーバのコンテンツファイルが収納されます。既存のサーバの設定を移行している場合、このディレクトリは、移行処理が終了した時点ではじめて表示されません。
- **extras** には、ログアナライザやログ分析ツールが収納されます。
 - flexanlg ディレクトリには、コマンド行ログアナライザが 1 つ収納されます。このログアナライザは、フレックスログ形式のファイルを分析します。
 - log_anly ディレクトリには、Server Manager を介して動作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。

- **httpacl** には、アクセス制御構成情報を保持する `generated.server-id.acl` ファイルおよび `genwork.server-id.acl` ファイルがあります。`generated.server-id.acl` ファイルには、**Server Manager** のアクセス制御フォームを使用して変更した、変更保存後の変更情報があります。`genwork.server-id.acl` ファイルには、変更保存前の変更情報があります。
- **https-admserv** には、**Administration Server** 用のディレクトリがあります。このディレクトリには、次のサブディレクトリとファイルがあります。
 - **UNIX/Linux** プラットフォームでは、このディレクトリには、サーバを起動、停止、再起動させるシェルスクリプトと、ログファイルをローテーションさせるスクリプトがあります。
 - **ClassCache** には、**JavaServer page** のコンパイルの結果として生成される、クラスや **Java** のファイルがあります。
 - **conf_bk** には、**Administration Server** の構成ファイルのバックアップコピーがあります。
 - **config** には、次のサーバの構成ファイルがあります。`admpw`、`admin.conf`、`cluster.xml`、`contexts.properties`、`cron.conf`、`dsgw.conf`、`dsgwfilter.conf`、`dsgw-language.conf`、`dsgw-orgperson.conf`、`dsgwsearchprefs.conf`、`jvm12.conf`、`magnus.conf`、`magnus.conf.clfilter`、`mime.types`、`ns-cron.conf`、`obj.conf`、`obj.conf.clfilter`、`server.dtd`、`servers.lst`、`server.xml`、`server.xml.clfilter`、`servlets.properties`、`ssl.xml`、`user-apps.xml`、`userclass.obj.conf`、および `web-apps.xml`。作業用のコピーはここに保存されます。`magnus.conf` と `obj.conf` については、iPlanet Web Server の『NSAPI プログラマーズガイド』を参照してください。
 - **logs** には、エラーやアクセスのログファイルが収納されます。
 - **SessionData** には、**MMapSessionManager** からのセッションデータベースデータがあります。
 - **startsvr.bat** は、**Server Manager** を **Windows NT** マシンで起動するスクリプトです。**Server Manager** は、サーバのルートディレクトリにインストールされたすべてのサーバの構成を可能にします。
 - **stopsvr.bat** は、**Windows NT** マシンの **Server Manager** を停止するスクリプトです。
- **https-server_id.domain** は、マシンにインストールした各サーバ用のディレクトリです。各サーバのディレクトリには、次のサブディレクトリとファイルがあります。
 - **ClassCache** には、**JavaServer page** のコンパイルの結果として生成される、クラスや **Java** のファイルがあります。
 - **conf_bk** には、サーバの構成ファイルのバックアップコピーがあります。
 - **config** には、サーバインスタンスの構成ファイルがあります。
 - **logs** には、サーバインスタンスのログファイルがあります。

- `reconfig` は、サーバの動的再構成に使用されるスクリプトです。サーバに非グローバルな変更を加える場合は、このスクリプトを使用して、サーバを一旦停止して再起動することなく、サーバを再構成することができます。ただし、ACL ファイルや `magnus.conf` を変更する場合は、サーバを停止し、再起動する必要があることに注意してください。
- `restart` は、サーバを再起動するスクリプトです。
- `rotate` は、サーバに接続中のユーザに影響をおよぼすことなく、サーバのログファイルをローテーションさせます。
- `search` には、次のディレクトリがあります: `admin` および `collections`。
- `SessionData` には、`MMapSessionManager` からのセッションデータベースデータがあります。
- `startsvr.bat` は、`Server Manager` を起動するスクリプトです。`Server Manager` は、サーバルートディレクトリにインストールされたすべてのサーバの構成を可能にします。
- `stopsvr.bat` は、`Server Manager` を停止するスクリプトです。
- `manual` には、この製品のオンラインマニュアルがあります。
- `plugins` には、Java、検索、その他のプラグイン用のディレクトリがあります。このディレクトリには、次のサブディレクトリがあります。
 - `htaccess` には、`.htaccess` アクセス制御のサーバプラグインと `.nsconfig` を `.htaccess` に変換するコンバータである `htconvert` プラグインがあります。
 - `digest` には、iPlanet Directory Server 5.0 の `Digest Authentication Plugin`、およびプラグイン関連情報があります。
 - `servlets` には、`web-apps` アプリケーションの情報およびその例があります。
 - `include` には、さまざまなインクルードファイルがあります。
 - `lib` には、共用ライブラリがあります。
 - `nsacl` には、サーバのアクセス制御リストに関する情報があります。
 - `loadbal` には、`Resonate` ロードバランサ統合プラグインに必要なファイルがあります。
 - `nsapi` には、ヘッダーファイルや、NSAPI を使用して独自の関数を作成するために例として使用するコードがあります。詳細は、次の、ドキュメントの `Web` サイトを参照してください。
<http://docs.iplanet.com/docs/manuals/enterprise.html> (英語)
 - `search` には、サーバの検索プラグインに関する情報があります。
 - `snmp` には、サーバの SNMP プラグインに関する情報があります。
- `setup` には、`setup.log` および `uninstall.inf` ファイルを含む、さまざまな iPlanet Web Server セットアップファイルがあります。

- `userdb` には、ユーザデータベースと関連情報があります。
- `LICENSE.txt` は、ライセンスファイルです。
- `README.txt` は、`readme` ファイルで、iPlanet Web Server の *Note* へのリンクが記述されています。

UNIX と Linux プラットフォーム

「全プラットフォーム共通」の節で説明したファイルやディレクトリ以外にも、次のファイルが、UNIX および Linux プラットフォームの `server-root` ディレクトリに作成されます。

- `startconsole` は、ブラウザを Administration Server ページに起動します。

次のファイルは、UNIX および Linux プラットフォームの `server-root/https-admserv` ディレクトリに作成されます。

- `ClassCache` には、JavaServer page のコンパイルの結果として生成される、クラスや Java のファイルがあります。
- `conf_bk` には、サーバの構成ファイルのバックアップコピーがあります。
- `config` には、Administration Server の構成ファイルがあります。
- `logs` には、Administration Server のログファイルがあります。
- `SessionData` には、MMapSessionManager からのセッションデータベースデータがあります。
- `restart` は、Server Manager を再起動するスクリプトです。
- `start` は、Server Manager を起動するスクリプトです。Server Manager は、サーバルートディレクトリにインストールされたすべてのサーバの構成を可能にします。
- `stop` は、Server Manager を停止するスクリプトです。

仮想サーバの構成

仮想サーバを使用すると、インストールされた単一のサーバで、複数の会社または個人のドメイン名、IP アドレス、およびいくつかのサーバ管理機能を提供できます。仮想サーバの構成には、Server Manager の「Virtual Server Class」タブや、Class Manager インタフェース、`server.xml` ファイルなどを使用します。仮想サーバの設定は、`server_root/server_id/config` ディレクトリの `server.xml` ファイルに格納されます。

詳細は、第 13 章「仮想サーバの使用」を参照してください。

複数サーバの構成

1 台のサーバマシン上で複数の Web サーバを稼働させることができます。複数 Web サーバは、Administration Server という単一サーバ用の管理インタフェースから構成できます。

Administration Server

Administration Server は、Web ベースのサーバで、Java フォームが含まれており、これを使用して iPlanet Web Server のすべてを構成します。

iPlanet Web Server をインストールしたら、ブラウザを使用して、Administration Server ページを開き、フォームに必要事項を入力して iPlanet Web Server を構成します。フォームを送信すると、Administration Server が管理しているサーバの構成を変更します。

Administration Server ページを開くのに使用する URL は、iPlanet Web Server のインストール時に指定した、Administration Server 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、Administration Server 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.mozilla.com:1234
```

Administration Server 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。ユーザ名とパスワードを入力する必要があります。お使いのコンピュータに iPlanet Web Server をインストールする時に、「スーパーユーザ」のユーザ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、複数のユーザが Administration Server 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 3 章「Administration Server の設定」の 57 ページの「複数の管理者の許可」を参照してください。

Administration Server 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、iPlanet Web Server の管理、追加、削除、移行を実行します。Administration Server 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 サーバの構成に必要な 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、Administration Server の使用方法については、第 2 章「iPlanet Web Server の管理」を参照してください。

Server Manager

Server Manager 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、iPlanet Web Server の各インスタンスを構成します。

iPlanet Web Server の Server Manager にアクセスするには、次の手順を実行します。

1. iPlanet Web Server をインストールして、起動します。
Administration Server に「Servers」ページが表示されます。
2. 「Manage Servers」領域から、目的のサーバを選択し、「Manage」をクリックします。
iPlanet Web Server に、Server Manager の「Preferences」ページが表示されます。

注 サーバの構成に必要な 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 サーバのオン、オフを実行したりします。

さらに、Server Manager には次のタブが用意してあり、iPlanet Web Server 管理タスクを実行できます。

- Security
- Logs
- Monitor
- Virtual Server Class
- Java
- Legacy Servlets
- Search

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

リソースピッカーの使用

Server Manager と Class Manager のページの多くは、iPlanet Web Server 全体やクラス全体を構成するのに使用します。ただし、サーバ(またはクラス)全体、またはサーバ(またはクラス)が管理するファイルとディレクトリ、のどちらかを構成できるページもあります。これらのページには、図 1-1 に示すような、リソースピッカーが上部に表示されています。

図 1-1 リソースピッカー



リソースピッカーの表示されているページは、数多くあり、Server Manager の「Log Preferences」ページや Class Manager 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、構成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ディレクトリを表示します。他のディレクトリを指定するには、「Options」をクリックします。「Wildcard」をクリックすると、特定の拡張子の付いたファイルを構成することができます。

リソースピッカーで使用するワイルドカード

サーバ構成の大部分で、ワイルドカードパターンを指定して、構成する 1 つ、または、複数の項目を示すことができます。ただし、アクセス制御やテキスト検索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が使用されます。こうした特殊文字を特殊な意味を持たせずに使用したい場合は、バックスラッシュ (\) を文字の前に付けます。

Class Manager

Class Manager 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 iPlanet Web Server を構成します。仮想サーバのユーザインタフェースには、Server Manager と Class Manager の 2 つの部分があります。Class Manager は、単一クラスや単一仮想サーバに影響を与える設定を行います。Class Manager でクラスにサービスを設定したり、仮想サーバ(クラスのメンバ)を追加したり、個々の仮想サーバの設定を構成することができます。

iPlanet Web Server の Class Manager にアクセスするには、次の手順を実行します。

1. Server Manager から、「Virtual Server Class」タブをクリックします。
Server Manager に「Manage Class of Virtual Servers」ページが表示されます。
2. ドロップダウンリストから、仮想サーバクラスを選択し、「Manage」をクリックします。
iPlanet Web Server に、Class Manager の「Manage Virtual Servers」ページが表示されます。

Class Manager には、画面の右上隅にある Class Manager 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。

Class Manager には、iPlanet Web Server 仮想サーバ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。

Virtual Server Manager

Virtual Server Manager にアクセスするには、Class Manager 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を選択し、「Manage」をクリックします。またはツリービューで仮想サーバへのリンクをクリックします。

Virtual Server Manager のページから、ステータスや設定を確認したり、Java Web アプリケーション状態をオンにしたり、指定する仮想サーバについてのレポートを生成したりすることができます。

Virtual Server Manager には、iPlanet Web Server 仮想サーバ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

iPlanet Web Server の管理

この章では、iPlanet Web Server, Enterprise Edition 6.0 の iPlanet Web Server Administration Server による管理方法を説明します。Administration Server を使用して、サーバの管理、サーバの追加や削除、以前のリリースからのサーバの移行を行うことができます。

この章には、次の節が記述されています。

- Administration Server へのアクセス
- 複数のサーバの稼働
- サーバの複数のインスタンスをインストールする
- サーバの削除
- 以前のバージョンからのサーバの移行

Administration Server へのアクセス

この節では、UNIX/Linux や Windows NT プラットフォームで Administration Server にアクセスする方法を説明します。

UNIX や Linux プラットフォーム

UNIX や Linux プラットフォームで Administration Server にアクセスするには、`server_root/https-admserv/` ディレクトリに移動します (例：`/usr/iplanet/servers/https-admserv/`)。そして、`./start` と入力します。このコマンドで、Administration Server 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 NT プラットフォーム

Windows NT プラットフォームでは、iPlanet 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 Notes
- Start Administration Server
- Uninstall iPlanet Web Server 6.0
- Administer Web Server

Administration Server 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 NT 4.0 で Administration Server にアクセスするには、次の手順を実行します。

1. 「Start Administration Server」アイコンをダブルクリックするか、または、Administration Server 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`

iPlanet Web Server が起動し、ユーザ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名とパスワードを入力します。

iPlanet 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」を参照してください。

| | |
|---|---|
| 注 | サーバの構成に必要な 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。 |
|---|---|

Administration Server 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。Administration Server 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に接続できるマシンならばどのマシンからでも Administration Server にアクセスできます。

複数のサーバの稼動

お使いのシステムで Web サーバを稼動させるには、2つの方法があります。

- 仮想サーバを使用する
- サーバの複数のインスタンスをインストールする

仮想サーバ

仮想サーバを使用すると、インストールされた単一のサーバで、複数の会社または個人のドメイン名、IP アドレス、およびいくつかのサーバ管理機能を提供できます。ユーザにとってはまるで自分の Web サーバ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 サーバの基本的なメンテナンスを提供するのはユーザではありません。

仮想サーバの設定は、`server_root/server_id/config` ディレクトリの `server.xml` ファイルに格納されます。仮想サーバ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、『NSAPI プログラマーズガイド』を参照してください。

仮想サーバについての詳細は、第 13 章「仮想サーバの使用」を参照してください。

サーバの複数のインスタンスをインストールする

iPlanet Web Server の以前のリリースでは、仮想サーバには独自の構成情報がありませんでした。サーバに別個の構成情報を持たせる唯一の方法は、新規にサーバインスタンスを生成することでした。しかし、iPlanet Web Server, Enterprise Edition 6.0 では、仮想サーバには別個の構成情報を持つことができるため、複数のサーバ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、複数のサーバを持つには仮想サーバを使用することが推奨されます。

Web サーバの複数のインスタンスのインストールを選択する場合、Administration Server を使用して次を実行します。

- NT 上のサーバの複数のコピーを、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を構成します。

システムを複数の IP アドレスで待機するように構成している場合は、インストールするサーバごとに、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、すでにサーバをインストールしていた場合は、システムの構成を別の複数の 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をインストールするか、または、**Server Manager** を使用してサーバ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。

別のサーバインスタンスを追加するには、次の手順を実行します。

1. **Administration Server** 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希望する情報を入力します。

サーバ識別子は、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。

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。

サーバの削除

Administration Server を使用して、システムからサーバを削除できます。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 NT サーバには、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や関連の管理サーバ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。

お使いのマシンからサーバを削除するには、次の手順を実行します。

1. **Administration Server** 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。

Administration Server は、続いて、サーバの構成ファイル、**Server Manager** 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の移行

iPlanet Web Server は、バージョン 4.x から 6.0 へ移行することができます。前の 4.x サーバは保持され、新たに、6.0 サーバ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.x サーバの稼働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のサーバの移行方法についての詳細は、『インストールと移行』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。

以前のバージョンからのサーバの移行

Administration Server の使用方法

第 3 章 「Administration Server の設定」

第 4 章 「ユーザとグループの管理」

第 5 章 「Web サーバのセキュリティ」

第 6 章 「サーバクラスタの管理」

Administration Server の設定

Administration Server は、「Preferences」タブと「Global Settings」タブのページを使用して構成できます。サーバの構成に必要な 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。

- Administration Server のシャットダウン
- 待機ソケット設定の編集
- ユーザアカウントの変更 (UNIX/Linux)
- スーパーユーザ設定の変更
- 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構成
- サーバへのアクセスの制限
- JRE/JDK パスの構成

Administration Server のシャットダウン

サーバがインストールされると、サーバは常時稼働して HTTP 要求を待機し、受け取ります。しかし、サーバを停止し再起動する必要が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を停止できます。

- Administration Server 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。

- Windows NT: 「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。
- stop を使用して、サーバを完全にシャットダウンします。サービスは、サーバが再起動するまで中断されます。

サーバ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

サーバが要求を処理する前に、サーバは待機ソケットから要求を受け取り、その要求を適切な接続グループと仮想サーバに振り向ける必要があります。iPlanet Web Server をインストールすると、ls1 という 1 つの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。

サーバの待機ソケットの設定は、Administration Server 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. Administration Server 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。

詳細は、第 13 章「仮想サーバの使用」、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 サーバのユーザアカウントを変更できます。サーバの処理はすべて、このユーザアカウントとして実行されます。

1024 より上のポート番号が指定されており、root ユーザ以外のユーザアカウントで稼働している場合は、サーバユーザを指定する必要はありません (この場合、サーバを起動するために root でログオンする必要はありません)。ここでユーザアカウントを指定しない場合、サーバは、起動時のユーザアカウントで稼働します。サーバを起動するときは、必ず適正なユーザ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、お使いのシステムのマニュアルを参照してください。

サーバを root で起動した場合でも、常時 root でサーバを稼働すべきではありません。サーバに、システムリソースへのアクセスを一部制限させたり、非特権ユーザとして稼働させたりしたい場合もあります。サーバユーザとして入力したユーザ名は、すでに、通常の UNIX/Linux ユーザアカウントとして存在しているはずですが、サーバの起動後は、このユーザ名で稼働します。

ユーザアカウントを新規に作成したくない場合は、ユーザとして nobody を選択するか、または、同じホストで稼働している、別の HTTP サーバ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 nobody は、ファイルを所有することはできませんが、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. Administration Server 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。

スーパーユーザ設定の変更

Administration Server のスーパーユーザのアクセスを設定できます。この設定は、スーパーユーザアカウントにのみ影響します。つまり、Administration Server 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。

注意 ユーザやグループを管理するのに iPlanet Directory Server を使用する場合、スーパーユーザ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、Administration Server 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 Administration Server にアクセスするか、または iPlanet Directory Server のコンソールや構成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。

Administration Server のスーパーユーザ設定を変更するには、次の手順を実行します。

1. Administration Server 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 Administration Server のユーザを、root からオペレーティングシステム上の別のユーザに変更できます。これにより、この同じグループに属する複数のユーザが構成ファイルを編集、管理できるようになります。ただし、UNIX/Linux の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構成ファイルの「rw」(読み書き)を許可しますが、Windows NT のプラットフォームの場合は、ユーザは「administrators」グループに属している必要があります。

スーパーユーザのユーザ名とパスワードは、`server_root/https-admserv/config/admpw` ファイルに格納されています。ユーザ名を忘れた場合は、このファイルを開いてユーザ名を確認できます。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。

| | |
|-----------|--|
| 注意 | <p>admpw ファイルは編集可能なため、サーバ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。</p> <ul style="list-style-type: none">• 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、どんなシステムユーザでも Administration Server デーモンを実行できるようになってしまいます。• NT システムでは、ファイル所有権は、Administration Server が使用するユーザ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の特定の部分を変更することができます。分散管理では、ユーザは3つのレベルに分類されます。

- **スーパーユーザ**とは、`server_root/https-admserv/config/admpw` ファイルに記載されているユーザです。これは、インストール時に指定したユーザ名 (およびパスワード) になります。このユーザは、Administration Server のすべてのフォーム (ただし、Users & Groups フォームは除く) へフルアクセスできます。Users & Groups フォームへは、iPlanet Directory Server のような LDAP サーバで有効なアカウントを持つスーパーユーザがアクセスできます。
- **管理者**は、Administration Server を含む、特定のサーバの 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の規則 (通常はスーパーユーザにより設定される) 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、ユーザの追加、アクセス制御の変更などその他のユーザに影響する項目を変更できます。
- **エンドユーザ**は、データベースに保存されている読み取り専用データを閲覧できます。ただし、エンドユーザは、特定のデータに関してのみ、変更を許可される場合があります。

iPlanet Web Server のアクセス制御についての詳細は、第 8 章「サーバへのアクセス制御」の 158 ページの「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、iPlanet Web Server の『インストールと移行』および iPlanet Directory Server の『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. Administration Server 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、Administration Server (または、サーバルートにインストールされたその他のサーバ) を構成することを許可したいユーザの名前を追加します。「administrators」グループ内のすべてのユーザは、Administration Server へのフルアクセスを保持していますが、アクセス制御を使用して、それらのユーザが構成できるサーバやフォームを制限することもできます。

注意 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

Administration Server のログファイルには、サーバ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アクセスに関する情報が記述されます。ログを閲覧することで、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、Administration Server 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、Administration Server 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、Administration Server 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9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

Administration Server のログファイルは、サーバ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 NT の場合、ログファイルへのパスは、`c:\iPlanet\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、iPlanet Web Server コンソールから閲覧したり、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、サーバへの要求やサーバ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. Administration Server 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 Access Log」ページ」、および第9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が遭遇したエラーすべてが列記されます。このファイルには、サーバの起動時刻や、サーバへのログインに失敗したユーザ名などのサーバ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. Administration Server 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 Error Log」ページ」、および第9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。一定の時刻に、または一定の間隔で、iPlanet Web Server は、アクセスログをローテーションさせます。iPlanet Web Server は、古いログファイルを保存し、このファイルに保存日時を含む名前を付けます。

たとえば、ログファイルを 1 時間ごとにローテーションさせるように設定すると、iPlanet Web Server は、ログファイルに「access.199907152400」という名前を付けて保存します。ここでは「名前 | 年 | 月 | 日 | 24 時間表示」が 1 つの文字列に連結されています。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。

アクセスログローテーションは、サーバ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、iPlanet 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、iPlanet Web Server は、アクセスログファイルに記録する必要が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。

Cron ベースのログローテーションの使用 (UNIX/Linux)

iPlanet Web Server のいくつかの機能を構成して、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。Cron デーモンがコンピュータのクロックを確認し、一定の時刻に処理を開始します。(これらの設定は、ns-cron.conf ファイルに格納されます。)

この Cron デーモンは、iPlanet Web Server のスケジュールされているタスクを制御し、Administration Server 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、サーバの種類に依存します。(NT プラットフォームでは、スケジューリングは、個々のサーバ内で行われることに注意してください。)

Cron デーモンが制御できるタスクのなかには、コレクションのメンテナンス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 Cron 制御を再起動する必要があります。

Cron 制御を再起動、起動、または停止するには、次の手順を実行します。

1. Administration Server 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、Cron 制御を変更します。

Cron 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構成

ユーザ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、サーバを構成して、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設定を構成するには、次の手順を実行します。

1. Administration Server 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、「Save Changes」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。

サーバへのアクセスの制限

サーバへのアクセスの制御は、サーバ全体に対して、またはサーバの一部 (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。サーバが受信した要求を評価する場合、アクセス制御エントリ (access-control entries、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 (access-control list、ACL) と呼ばれます。要求がサーバに着信すると、サーバは `vsclass.obj.conf` (`vsclass` は仮想サーバのクラス名) で ACL への参照を検索します。この参照はアクセス権を決定するために使用されます。デフォルトでは、サーバには、複数の ACL が含まれている 1 つの ACL ファイルがあります。

アクセス制御は、Administration Server を使用してすべてのサーバに対して全体的に設定したり、Server Manager を使用して特定のサーバ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 8 章「サーバへのアクセス制御」の 169 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。

iPlanet Web Server へのアクセスを制御するには、次の手順を実行します。

1. Administration Server 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. 目的のサーバを選択し、「Edit ACL」をクリックします。

Administration Server には、指定したサーバのアクセス制御の規則が表示されません。

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。

JRE/JDK パスの構成

iPlanet Web Server をインストールするとき、iPlanet Web Server に同梱されている JRE (Java Runtime Environment) をインストールするか、または、別途インストールする必要のある JDK (Java Development Kit) へのパスを指定するか、のどちらかを選択できます。詳細は、iPlanet Web Server の『インストールと移行』を参照してください。

インストール時に JRE をインストールする場合も、JDK へのパスを指定する場合も、次の手順を実行して iPlanet Web Server に通知すれば、いつでも、JRE または JDK の使用を切り替えられます。

1. iPlanet Web Server の Administration Server へアクセスします。
2. 「Global Settings」タブを選択します。
3. 「Configure JRE/JDK Paths」リンクをクリックします。
「Configure JRE/JDK Paths」ページが表示されます。
4. 有効にするオプションに対応するラジオボタンをクリックします。

たとえば、マシンにインストールされている JDK へのパスを指定するには、「JDK」をクリックします。

5. 正しい情報を入力し、「OK」をクリックします。

変更を有効にするには、サーバを再起動する必要があります。

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

ユーザとグループの管理

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

この章には、次の節が記述されています。

- LDAP を使用してユーザとグループを管理する
- ユーザの作成
- ユーザの管理
- 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
- Preferred Language List の管理

LDAP を使用してユーザとグループを管理する

Administration Server を使用して、ユーザアカウント、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。

ユーザやグループの情報は、iPlanet Directory Server 5.0 などのディレクトリサーバに格納されています。iPlanet Directory Server は、LDAP (Lightweight Directory Access Protocol) をサポートします。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拡張可能です。

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

識別名 (DN) の理解

Administration Server の「Users and Groups」タブを使用して、ユーザ、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、グループ、および組織単位については、この章で後程詳細を説明します。

企業内のユーザやグループは、それぞれ、識別名 (Distinguished Name、DN) 属性で表されます。DN 属性は、関連するユーザ、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。iPlanet Console のユーザやグループインタフェースを使用すると、簡単に DN を作成、変更できます。

次の例は、Netscape Communications Corporation の社員の一般的な DN を表しています。

```
uid=doe,e=doe@netscape.com,cn=John Doe,o=Netscape Communications Corp.,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 ID
- e: 電子メールアドレス
- cn: ユーザの共通名
- o: 組織
- 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 Administration Server 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。エントリーは、`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ーを追加するには、まず、LDIF ファイル内のエントリーを定義し、次に、Directory Server から LDIF ファイルをインポートします。詳細は、付録 A 「コマンド行ユーティリティ」の 383 ページの「LDIF エントリーの書式設定」を参照してください。

ユーザの作成

Administration Server の「Users and Groups」タブを使用して、ユーザエントリーを作成したり、変更したりします。ユーザエントリーには、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。

この節では、次の内容について説明します。

- ユーザエントリー作成のガイドライン
- 新規ユーザエントリーの作成方法
- Directory Server のユーザエントリー

ユーザエントリー作成のガイドライン

新規のユーザエントリーの作成に `administrator` フォームを使用する際には、次のガイドラインを考慮します。

- 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のフルネームとユーザ ID が自動的に入力されます。ユーザ ID は、ユーザのファーストネームの頭文字の後にユーザのラストネームを組み合わせて生成されます。たとえば、ユーザの名前が `Billie Holiday` の場合、ユーザ ID は、自動的に `bholiday` となります。このユーザ 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 ID は一意である必要があります。Administration Server 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 ID が使われていないかを調べて、ユーザ 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)、ユーザを作成する場合は、ユーザ 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 ID が存在していた場合、該当するユーザは、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、iPlanet Web Administration Server のエントリー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内のエントリーの名前の文字列表現です。
- 新規にユーザエントリーを作成する場合、少なくとも次のユーザ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 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編集のテキストフィールドは、Administration Server と iPlanet Console では異なります。iPlanet Console では、タグのない cn フィールドのほかに、preferred language (希望言語) の cn フィールドがありますが、Administration Server にはこのフィールドはありません。

新規ユーザエントリーの作成方法

ユーザエントリーを作成するには、67 ページの「ユーザエントリー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. Administration Server 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、表示されたページに関連情報を追加します。

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 のユーザエントリー

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

- ユーザエントリーは、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」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 4-1 LDAP 属性

| ユーザフィールド | 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 ID) | uid |
| Password (パスワード) | userPassword |
| Email Address (電子メールアドレス) | mail |

ユーザ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 4-2 ユーザエントリ LDAP 属性

| ユーザフィールド | 対応する LDAP 属性 |
|----------------|-----------------|
| Title (役職) | title |
| Telephone (電話) | telephoneNumber |

- ユーザの名前は、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、ユーザの preferred language (希望言語) を選択して、ユーザ名をその言語の文字で表示することができます。ユーザの 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。

ユーザの管理

ユーザの属性は、Administration Server 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、ユーザエントリの検索、変更、名前の変更、削除を行ったり、ユーザ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。

Netscape/iPlanet サーバの中には、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が Administration Server 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。

- ユーザ情報の検索
- ユーザ情報の編集
- ユーザのパスワードの管理
- ユーザライセンスの管理
- ユーザ名の変更
- ユーザの削除

ユーザ情報の検索

ユーザ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。特定のユーザ情報を検索するには、次の手順を実行します。

1. Administration Server 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 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」フィールドのプル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。
ユーザ編集フォームが表示されます。
8. 表示された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。

カスタム検索照会の構築

「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。

- 一番左側のプルダウンリストを使用して、検索の基準とするエントリの属性を指定します。

使用可能な検索属性のオプションを、次の表に記述します。

表 4-3 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 ID) | 各エントリのユーザ ID で一致しているものを検索します。 |

表 4-3 検索属性オプション (続き)

| オプション名 | 説明 |
|------------------------------|---------------------------------|
| 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 (記述) | 各エントリの記述で一致しているものを検索します。 |

- 中央のプルダウンリストで、実行したい検索のタイプを指定します。

使用可能な検索タイプのオプションを、次の表に記述します。

表 4-4 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、ユーザ名が「John Smith」でない、ディレクトリ内のすべてのユーザを検索したい場合、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、または表音による検索を実行します。属性のおよその値はわかっているけれども、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、ユーザ名のスペルが、「Saret」、「Surette」、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、ユーザ名が「Miles」で始ま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、このオプションを使用します。 |

表 4-4 検索タイプのオプション (続き)

| オプション名 | 説明 |
|-----------|--|
| 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。

ユーザ情報の編集

ユーザエントリを変更するには、次の手順を実行します。

1. Administration Server にアクセスして、「Users & Groups」タブを選択します。
2. 70 ページの「ユーザ情報の検索」の説明に従って、ユーザ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 <code>edit user</code> 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、 <code>Directory Server</code> の <code>ldapmodify</code> 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、74 ページの「[ユーザ名の変更](#)」を参照してください。

ユーザのパスワードの管理

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

ユーザ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. Administration Server にアクセスして、「Users & Groups」タブを選択します。
2. 70 ページの「ユーザ情報の検索」の説明に従って、ユーザ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 Administration Server のユーザを、root からオペレーティングシステム上の別のユーザに変更できます。これにより、同じグループに属する複数のユーザが構成ファイルを編集、管理できるようになります。ただし、UNIX/Linux の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構成ファイルの「rw」(読み書き)を許可しますが、Windows NT のプラットフォームの場合は、ユーザは「Administrators」グループに属している必要があります。

ユーザのパスワードは、「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、そのユーザのディレクトリエントリを削除せずに、そのユーザがサーバへログインできなくすることができます。このユーザ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。

ユーザライセンスの管理

Administration Server を使用して、ユーザが使用許可のライセンスを持っている iPlanet サーバ製品を調べることができます。

ユーザ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. Administration Server にアクセスして、「Users & Groups」タブを選択します。
2. 70 ページの「ユーザ情報の検索」の説明に従って、ユーザ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。

ユーザ名の変更

名前の変更機能は、ユーザ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、ユーザの古い名前は残されたままなので、古い名前でも検索しても新しいエントリが表示されます。

ユーザエントリ名を変更する場合、変更できるのはユーザ名だけです。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. Administration Server にアクセスして、「Users & Groups」タブを選択します。
2. 70 ページの「ユーザ情報の検索」の説明に従って、ユーザ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 Administration Server 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。

ユーザの削除

ユーザエントリを削除するには、次の手順を実行します。

1. Administration Server にアクセスして、「Users & Groups」タブを選択します。
2. 70 ページの「ユーザ情報の検索」の説明に従って、ユーザ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。iPlanet Web Server グループは、共通する属性を共有する複数のユーザ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。

静的グループは、メンバーオブジェクトを明示的に列挙します。静的グループは 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。静的グループでは、メンバーは、CN=<Groupname> 属性以外の共通の属性は共有しません。

静的グループでは、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やメンバーを追加することができます。

この節では、グループを作成するための、次の内容について説明します。

- 静的グループ

静的グループ

Administration Server を使って、複数ユーザの DN の中に、同じグループ属性を指定して、静的グループを作成できます。静的グループは、ユーザ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。

静的グループ作成のガイドライン

新規の静的グループを作成するために Administration Server 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。

- 静的グループには、その他の静的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、78 ページの「グループ属性の編集」を参照してください。

静的グループを作成するには

静的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. Administration Server 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、78 ページの「グループ属性の編集」、および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

Administration Server 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. Administration Server 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、現在ディレクトリにあるグループ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」フィールドのプル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、71 ページの「カスタム検索照会の構築」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. Administration Server 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、77 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 Administration Server のユーザを、root からオペレーティングシステム上の別のユーザに変更できます。これにより、同じグループに属する複数のユーザが構成ファイルを編集、管理できるようになります。ただし、UNIX/Linux の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構成ファイルの「rw」(読み書き)を許可しますが、Windows NT のプラットフォームの場合は、ユーザは「Administrators」グループに属している必要があります

グループ属性の編集については、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. Administration Server 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. 77 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。

iPlanet 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。リストにユーザエントリを追加する場合、「Users」が「Find」のプル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。
 - ユーザエントリを検索する場合は、ユーザ 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?」カラム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については、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、ユーザエントリと同様に、グループを追加します。詳細は、79 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. Administration Server 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、77 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。

- リストから削除したい各メンバーについて、「Remove from list?」カラムの下の、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については、79 ページの「グループメンバーの追加」を参照してください。
- 「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 4-5 追加情報

| タスク | 参照先 |
|-------------------|--------------------------------|
| グループに所有者を追加する | 79 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 | 80 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 | 80 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 4-6 追加情報

| タスク | 参照先 |
|----------------------|-------------------------------|
| 「See Also」にユーザを追加する | 79 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 | 80 ページの「グループメンバーリストへのグループの追加」 |

表 4-6 追加情報 (続き)

| タスク | 参照先 |
|------------------------|---------------------------------|
| 「See Also」からエントリーを削除する | 80 ページの「グループメンバーリストからのエントリーの削除」 |

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. Administration Server 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、77 ページの「グループエントリーの検索」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 Administration Server は、グループエントリーだけを削除します。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. Administration Server 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、77 ページの「グループエントリーの検索」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。

グループエントリー名を変更する場合、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、エントリーを 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せんが、Marketing という組織単位の下に Online Sales を、Product Management という組織単位の下に Online Sales にするようエントリー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. Administration Server 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」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. Administration Server 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」フィールドのプル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、71 ページの「カスタム検索照会の構築」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、Administration Server にアクセスし、次の手順を実行します。

1. 83 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。

organizational unit edit フォームが表示されます。

- 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、Administration Server にアクセスし、次の手順を実行します。

- 名前を変更したい組織単位の下ディレクトリには、他のエントリが何も入っていないことを確認します。
- 83 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。
- 「Rename」ボタンをクリックします。
- 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「グループの名前の変更」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、Administration Server にアクセスし、次の手順を実行します。

- 削除したい組織単位の下ディレクトリには、他のエントリが何も入っていないことを確認します。
- 83 ページの「組織単位の検索」で説明されているように、削除したい組織単位を特定します。
- 「Delete」ボタンをクリックします。
- 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。

Preferred Language List の管理

iPlanet Web Server では、Preferred Language List (希望言語リスト) を表示したり管理したりすることができます。

Preferred Language List を管理するには、次の手順を実行します。

1. Administration Server にアクセスして、「Users & Groups」タブを選択します。
2. 「Manage Preferred Language List」リンクをクリックします。
3. 「Display Language Selection List」フィールドで、「Yes」または「No」をクリックして、iPlanet Web Server に Language Selection List を表示させるかどうかを指定します。
4. 「Language in the Selection List」フィールドで、「Add to list」チェックボックスをクリックして、Preferred Language List に指定したい各言語を追加します。
5. Preferred Language List 内でデフォルト言語として指定する言語の「Default value」をクリックします。
6. 「Save Changes」をクリックします。

Web サーバのセキュリティ

この章では、データを保護し、侵入者のアクセスを拒絶し、必要なユーザがアクセスできるように設計された各種セキュリティ機能を有効にする方法について説明します。iPlanet Web Server 6.0 には、すべての iPlanet サーバのセキュリティアーキテクチャが組み込まれています。iPlanet Web Server 6.0 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、次の URL に記述された「Introduction to SSL」を参照してください。

<http://docs.ipplanet.com/docs/manuals/security/sslin/index.htm> (英語)

Web サーバをセキュリティ保護する手順について詳細は、次の各節で説明します。

- 認証の要求
- 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
- 証明書を管理する
- 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の要件を設定する
- Stronger Ciphers を設定する
- その他のセキュリティに関する問題

認証の要求

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。

認証に証明書を使用する

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の両方に証明書を持たせることができます。

証明書は、証明機関 (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、事前にサーバ証明書をインストールしておく必要があります。

サーバ認証

サーバ認証とは、クライアントによる、サーバの確実な 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。

仮想サーバ証明書

仮想サーバごとに、それぞれ異なる証明書データベースを設けることができます。各仮想サーバデータベースには、複数の証明書を格納することができます。仮想サーバ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。iPlanet Web Server では、Administration Server と各サーバのインスタンスが、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、129 ページの「パスワードまたは PIN を変更する」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.
```

Administration Server は、信頼データベースを 1 つしか所有できません。各サーバのインスタンスは、それぞれ専用の信頼データベースを所有できます。仮想サーバは、そのサーバインスタンスに対して作成された信頼データベースによって網羅されます。

信頼データベースを作成する

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. Administration Server または Server Manager にアクセスし、「Security」タブを選択します。
「Server Manager」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. Server Manager を使用の場合には、「Apply」をクリックし、その後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、Web サーバが起動する前に、管理者に対してキーデータベースパスワードの入力を求めるプロンプトが表示されます。Web サーバを人的介入なしで自動的に再起動するには、password.conf ファイルにパスワードを保存する必要があります。このファイルとキー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十分にセキュリティ保護されている場合に限りです。

通常、サーバ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も SSL 有効サーバ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。サーバの password.conf ファイルは、root またはサーバをインストールしたユーザだけが所有し、その所有者のみが password.conf 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。なぜなら、ファイルにアクセスできるユーザなら誰でも、SSL が有効なサーバのパスワードにアクセスできるからです。したがって、SSL が有効なサーバのパスワードを 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。

NT 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 password.conf ファイルの保存されているディレクトリをプロテクトしてください。ただしこのディレクトリには、管理サーバのユーザと Web サーバのユーザに対して読み取り / 書き込み許可を持たせる必要があります。ディレクトリをプロテクトしておくとも、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をプロテクトすることはできません。

SSL 有効サーバを自動的に起動させる

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。次に例を示します。

```
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も、Web サーバ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、iPlanet Web Server の推奨する証明機関です。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、直接サーバに対して証明書を返せるという利点があります。

サーバに証明書信頼データベースを作成後、証明書を要求し、証明機関 (Certificate Authority、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、利用可能な証明機関のリストは、「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」(証明機関リスト)に記載されています。

Administration Server は、サーバ証明書を 1 つしか所有できません。各サーバのインスタンスは、専用のサーバ証明書を所有できます。各仮想サーバについては、サーバインスタンスの証明書を選択することができます。

VeriSign 証明書を要求する

VeriSign 証明書を要求するには、次の手順を行います。

1. Administration Server または Server Manager にアクセスし、「Security」タブを選択します。
Server Manager 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. 「Get Certificate」をクリックします。

5. VeriSign の手順に従います。

VeriSign 証明書をインストールする

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. Administration Server または Server Manager にアクセスし、「Security」タブを選択します。
Server Manager 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. 「Install」をクリックします。
7. Server Manager を使用の場合には、「Apply」をクリックし、その後変更内容を有効にするため「Restart」をクリックします。

他のサーバ証明書の要求およびインストール

VeriSign のほかに、他の証明機関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「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

要求プロセスに入る前に、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.iplanet.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**は、ユーザがビジネスで使用する電子メールアドレスです。これは、ユーザと CA との間の連絡に使用されます。
- **組織**は、ユーザの会社、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**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も使用することもできます。
- **地域**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。
- **州名**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 では州名の省略形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.iplanet.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、現在カスタマと係争中の訴訟は無いことを 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。

他のサーバ証明書を要求する

証明書を要求するには、次の手順を行います。

1. Administration Server または Server Manager にアクセスし、「Security」タブを選択します。

Server Manager 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つけ、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 は、証明書の要求を扱う証明書サーバ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。サーバは、このパスワードを使用して、ユーザ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、ユーザの公開鍵と符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」(証明機関)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せん。

9. 「OK」をクリックします。

10. **Server Manager** を使用の場合には、「Apply」をクリックし、その後変更内容を有効にするため「Restart」をクリックします。

サーバは、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。証明書サーバに URL を指定した場合は、サーバがその 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を使用している場合には、証明書サーバ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を運用することになります。

他のサーバ証明書をインストールする

CA から証明書を受け取る際には、ユーザ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、ユーザのサーバの証明書
- 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、連続した証明機関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、証明機関 (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に送信しない場合には、ユーザはそれを要求する必要があります。多くの CA は、ユーザの証明書を電子メールで送信する際に CA 証明書を含めており、ユーザのサーバは、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際には、ユーザ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。

証明書をインストールする

証明書をインストールするには、次の手順を行います。

1. **Administration Server** または **Server Manager** にアクセスし、「**Security**」タブを選択します。

Server Manager の場合には、はじめにドロップダウンリストからサーバ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、お使いのサーバだけに関係する 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に使用する場合。
サーバ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. Server Manager を使用の場合には、「Apply」をクリックし、次に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>-cert7.db` となります。次に例を示します。

```
https-serverid-hostname-cert7.db
```

アップグレード時の証明書の移行

iPlanet Web Server 4.x からアップグレードする場合には、ユーザのファイル (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。iPlanet Web Server 6.0 Administration Server のユーザが古い 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。Administration Server 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。iPlanet Web Server 6.0 では、Administration Server と各サーバ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。

サーバ証明書とすべての含まれている証明機関を含む、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 Administration Server を、サーバインスタンスについては Server Manager 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インスタンスが同じエイリアスを共有していた場合は、移行されると、証明書と鍵ペアファイルは新しいサーバインスタンスの名前をとって名前変更されます。

サーバ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証明機関はすべて、iPlanet Web Server 6.0 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。

証明書を移行する

証明書を移行するには、次の手順を行います。

1. ローカルマシンから、Administration Server または Server Manager のどちらかにアクセスし、「Security」タブを選択します。

Server Manager 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インスタンスを選択する必要があります。

2. 次のいずれかを選択します。
 - Administration Server の「Migrate 3.X Certificates」リンク

- Server Manager の「Migrate Certificate」リンク
- 3. 「3.6 Server Root」を入力します。
- 4. 「Alias」を入力します。
- 5. パスワードを入力します。
- 6. 「OK」をクリックします。
- 7. Server Manager を使用の場合には、「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。

組み込みルート証明書モジュールの使用

動的に読み込み可能なルート証明書モジュールが、iPlanet Web Server 6.0 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。iPlanet Web Server 6.0 に対して、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 iPlanet 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は削除してはなりません。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 (NT 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 (NT) 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。

証明書を管理する

ユーザのサーバ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。

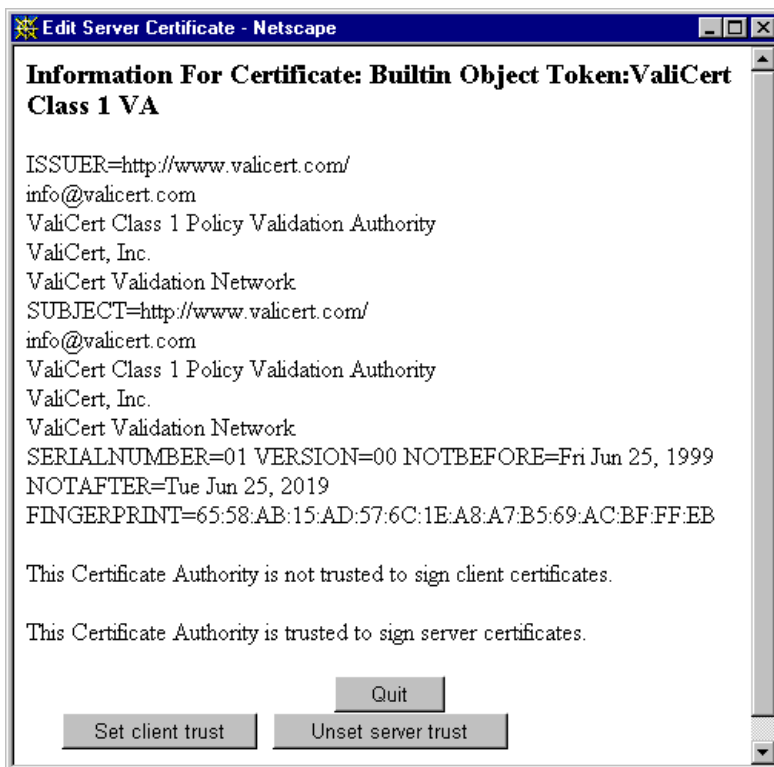
1. Administration Server または Server Manager にアクセスし、「Security」タブを選択します。

Server Manager 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はじめに入力し、「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。

図 5-1 「Edit Server Certificate」 ページ



4. 「Edit Server Certificate」 ウィンドウでは、次を選択できます。
 - 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. Server Manager を使用の場合には、「Apply」をクリックし、その後変更内容を有効にするために「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、サーバの信頼情報の設定を解除したりできます。LDAP サーバ証明書の場合は、サーバ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

証明書の取消しリスト (Certificate Revocation List、CRL) および危殆化鍵リスト (Compromised Key List、CKL) は、クライアントまたはサーバのユーザ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 をインストールする

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. Administration Server または Server Manager にアクセスし、「Security」タブを選択します。

Server Manager 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. Server Manager を使用の場合には、「Apply」をクリックし、その後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. Administration Server または Server Manager にアクセスし、「Security」タブを選択します。

Server Manager 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. 次のいずれかの操作を行います。
 - 「Delete CRL」
 - 「Delete CKL」
5. Server Manager を使用の場合には、「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。

セキュリティに関する詳細設定

証明書を取得すると、サーバのセキュリティ保護を開始できます。iPlanet 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読めるように変換し直すプロセスのことです。iPlanet Web Server 6.0 では、SSL および TLS 暗号化プロトコルをサポートしています。

符号化方式とは、暗号化または復号化に使用する暗号アルゴリズム (数学的関数) です。SSL と TLS プロトコルには、多数の符号化方式のセットが含まれています。符号化方式には、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、符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符号化方式を使用する必要があります。多数の符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は、通信に、その双方が持てる最も強力な符号化方式を使用します。SSL2、SSL3 および TLS プロトコルから符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 符号化方式での動作が保証されていないからです。

暗号化プロセスだけでは、サーバ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。

各種の符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。

サーバが使用できる符号化方式を指定するには、リスト内で符号化方式にチェックマークを付けます。特定の符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。

注意 「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符号化方式を利用できない場合には、サーバ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

iPlanet Web Server 6.0 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、この上により高レベルのプロトコルを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符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、別の符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、どの符号化方式のセットを通信に使用するかをサーバとクライアントが交渉する方法を決定します。

SSL を使用して LDAP と通信する

Administration Server 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。Administration Server で SSL を有効にするには、次の手順を行います。

1. Administration Server 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。

接続グループのセキュリティを有効にする

次の方法で、サーバの接続グループ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。
- 接続グループのサーバ証明書を選択します。
- 符号化方式を選択します。

セキュリティ機能をオンにする

接続グループ用に他のセキュリティ構成を行うには、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. **Server Manager** 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. 「**Add Listen Socket**」を選択します。
「**Create a Listen Socket**」ページが表示されます。
4. 必要な情報を入力してから、デフォルトの仮想サーバを選択します。
5. ドロップダウンリストを使用して、「**Security**」をオンにします。
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。

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

Administration Server または **Server Manager** 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. **Administration Server** または **Server Manager** にアクセスし、「**Security**」タブを選択します。
Server Manager の場合には、はじめにドロップダウンリストからサーバ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。
「**Listen Sockets Table**」ページが表示されます。
4. セキュリティ保護したい接続グループについて、まだ表示されていない場合には、「**Action**」ドロップダウンリストを使用して「**Edit**」を選択します。
5. 「**Security**」カラムのドロップダウンリストを使用して、接続グループに対するセキュリティ機能をオンにします。
6. 「**OK**」をクリックします。
これで、「**Security**」カラムに「**Attributes**」リンクが表示されます。

7. Server Manager 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。

接続グループのサーバ証明書を選択する

Administration Server または Server Manager のどちらかで、ユーザが要求しインストールしたサーバ証明書を使用するよう、接続グループを構成できます。

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。

接続グループが使用するサーバ証明書を選択するには、次の手順を行います。

1. Administration Server または Server Manager にアクセスし、「Preferences」タブを選択します。

Server Manager 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。
「Listen Socket Table」ページが表示されます。
3. 証明書を選択する接続グループについて、まだ表示されていない場合には、「Action」ドロップダウンリストを使用して「Edit」を選択します。
4. その接続グループに対する「Security」が、まだオフになっている場合には、ドロップダウンリストを使用してオンにします。
5. 「Attributes」リンクをクリックします。
「Security Settings of Listen Socket」ページが表示されます。

注 外部モジュールがインストールしてある場合には、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。

6. 「CertificateName」ドロップダウンリストから接続グループのサーバ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。
7. 「OK」をクリックします。
8. Server Manager を使用の場合には、「Apply」をクリックし、その後変更内容を有効にするため「Restart」をクリックします。

符号化方式の選択

Web サーバ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。SSL 2.0、SSL 3.0 および TLS 暗号化プロトコルを有効にして、各種の符号化方式セットを選択することができます。Administration Server の接続グループで、SSL および TLS を有効にできます。Server Manager の接続グループで SSL と TLS を有効にすると、その接続グループに関連するすべての仮想サーバに対して、これらのセキュリティの指定が設定されます。

セキュリティ保護されていない仮想サーバにするには、それらをすべてセキュリティ機能をオフにした同じ接続グループに構成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符号化方式が許可されています。特定の符号化方式セットを回避した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。特定の符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。

SSL と TLS を有効にするには、次の手順を行います。

1. Administration Server または Server Manager にアクセスし、「Preferences」タブを選択します。

Server Manager 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。

「Listen Socket Table」ページが表示されます。

3. セキュリティを有効にしたい接続グループについて、まだ表示されていない場合には、「Action」ドロップダウンリストを使用して「Edit」を選択します。

4. その接続グループに対する「Security」を、まだオフになっている場合には、ドロップダウンリストを使用してオンにします。

5. 「OK」をクリックします。

これで「Attributes」リンクが表示されます。

6. 「Attributes」リンクをクリックします。

「Security Settings of Listen Socket」ページが表示されます。

注 外部モジュールがインストールしてある場合には、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。

7. 次のいずれかを選択します。
 - 「Cipher Default」
 - 「SSL2」
 - 「SSL3/TLS」
8. (省略可能) SSL2 または SSL3/TLS を選択した場合には、「Security Features」ウィンドウで次のいずれかを実行します。
 - 「Allow」を選択し、デフォルトの符号化方式を受け入れます。
 - 「Allow」を選択し、必要な符号化方式にだけチェックマークを付け、不必要な符号化方式のチェックマークを外します。
 - 「Allow」のチェックマークを外して、このプロトコルとそのすべての符号化方式を無効にします。

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。Microsoft Internet Explorer 5.0 および 5.5 の場合には、「TLS Rollback」オプションを使用します。TLS は、ユーザのサーバへのアクセスを求めるブラウザでも有効にする必要があります。「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。

9. 「OK」をクリックして、「Security Features」ウィンドウを閉じます。
10. 「OK」をクリックします。
11. Server Manager 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 接続グループ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、その接続グループに関連するすべての仮想サーバに自動的にデフォルトのセキュリティパラメータが割当てられます。

サーバで 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.iplanet.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。

セキュリティをグローバルに構成する

SSL 有効サーバ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 (サーバのメイン構成ファイル) 内に作成されます。仮想サーバ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。仮想サーバの SSL のプロパティは、`server.xml` ファイルの `SSLPARAMS` 要素内にサーバごとに記述されています。

SSL 構成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. **Server Manager** にアクセスし、仮想サーバのサーバ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. 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**」リンクを選択します。
4. 値を設定する待機ソケットに対して、まだオンにされていない場合は「**Security**」を「On」にします。
5. 「OK」をクリックします。
6. 「**Magnus Editor**」リンクに進みます。
7. 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。
8. 次の項目の値を入力します。
 - `SSLSessionTimeout`
 - `SSLCacheEntires`
 - `SSL3SessionTimeout`
9. 「OK」をクリックします。
10. 「**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 構成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ングを制御します。

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 セッションの数を指定します。

SSL3SessionTimeout

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

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 指令が指定された場合には、秒数値は暗黙のうちに 5 ~ 86400 秒の間であると想定されます。

注 1つの待機ソケット上の1つの接続グループは、同じ SSLPARAMS を持つ必要があります。複数のグループがそれぞれ異なる SSLPARAMS を持つことができます。

外部暗号化モジュールの使用

iPlanet Web Server 6.0 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の使用に、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 モジュールをインストールする

iPlanet 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 を使用して PKCS#11 モジュールをインストールする

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. Administration Server を含むすべてのサーバ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。次に例を示します。

- 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
- NT では、PATH に次を追加します。

```
LD_LIBRARY_PATH server_root/bin/https/bin
```

お使いのマシンの 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

オプションが一覧されます。

7. 必要な操作を行います。

たとえば、UNIX に PKCS#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 (PKCS#11 ファイルの名前) -libfile (PKCS#11 用のユーザの  
libfile) -nocertdb -dbdir (db ディレクトリ)
```

pk12util を使用する

pk12util を使用して、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、cert7.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 でエクスポートする

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。次に例を示します。
 - 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`
 - NT では、`PATH` に次を追加します。


```
LD_LIBRARY_PATH server_root/bin/https/bin
```

お使いのマシンの `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`
オプションが一覧されます。
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 を使用してインポートする

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。次に例を示します。
 - 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
- NT では、PATH に次を追加します。

```
LD_LIBRARY_PATH server_root/bin/https/bin
```

お使いのマシンの 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 のあとに続け、また最後の引数でなくてはなりません。

引用符記号の中の大文字とスペースを含む、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、サーバはその証明書の使用を開始できません。

サーバは常に、「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を起動するには、稼動する接続グループの証明書名を指定する必要があります。

接続グループの証明書名を選択する

接続グループの証明書名を選択するには、次の手順を行います。

1. Administration Server または Server Manager にアクセスし、「Preferences」タブを選択します。

Server Manager 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。
「Listen Socket Table」ページが表示されます。
4. セキュリティを有効にしたい接続グループについて、まだ表示されていない場合には、「Action」ドロップダウンリストを使用して「Edit」を選択します。
5. その接続グループに対する「Security」を、まだオフになっている場合には、ドロップダウンリストを使用してオンにします。
6. 「OK」をクリックします。
これで「Attributes」リンクが表示されます。
7. 「Attributes」リンクをクリックします。
「Security Settings of Listen Socket」ページが表示されます。
8. 「CertificateName」ドロップダウンリストを使用して、外部サーバ証明書を選択します。
9. 「OK」をクリックします。
10. Server Manager 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。

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証明書を使用して起動することをサーバ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きに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、符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 をサーバ上にインストールすると、Federal Information Processing Standards (FIPS) - 140 に準拠するよう iPlanet Web Server を構成できます。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. Administration Server または Server Manager にアクセスし、「Preferences」タブを選択します。
Server Manager 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。
「Listen Socket Table」ページが表示されます。
4. FIPS-140 を有効にしたい接続グループについて、まだ表示されていない場合には、「Action」ドロップダウンリストを使用して「Edit」を選択します。
5. その接続グループに対する「Security」を、まだオフになっている場合には、ドロップダウンリストを使用してオンにします。
6. 「OK」をクリックします。
これで「Attributes」リンクが表示されます。
7. 「Attributes」リンクをクリックします。
8. 「Security Settings of Listen Socket」ページが表示されます。
9. 「SSL3/TLS」リンクをクリックします。
「Security Feature」ウィンドウが表示されます。
10. チェックマークが付いていない場合には、「Allow: SSL version 3」にチェックマークを付けます。
11. 次のうち、適切な FIPS-140 符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
12. 「OK」をクリックして、「Security Features」ウィンドウを閉じます。
13. 「OK」をクリックします。

14. Server Manager 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の要件を設定する

サーバ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。

クライアント認証を要求する

Administration Server と各サーバインスタンスの接続グループ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、照会に対してサーバ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。

クライアント証明書内の CA とクライアント証明書へ署名することが信頼されている CA を照合することによって、iPlanet Web Server は、クライアント証明書の認証をサポートします。Administration Server の「Security」の下にある「Manage Certificates」ページで、クライアントの証明書へ署名することが信頼されている 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 サーバを構成できます。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、100 ページの「証明書を管理する」を参照してください。

iPlanet 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、Administration Server 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エン트리と照合するようにサーバを構成できます。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、119 ページの「クライアント証明書を LDAP へマップする」を参照してください。

証明書のあるユーザは、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、166 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。詳細は、『NSAPI プログラマーズガイド』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. Administration Server または Server Manager にアクセスし、「Preferences」タブを選択します。
Server Manager 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。
「Listen Socket Table」ページが表示されます。
3. クライアント認証を要求したい「Connection Group」について、まだ表示されていない場合には、「Action」ドロップダウンリストを使用して「Edit」を選択します。
4. その「Connection Group」に対する「Security」が、まだオフになっている場合には、ドロップダウンリストを使用してオンにします。
5. 「Attributes」リンクをクリックします。
「Security Settings of Listen Socket」ページが表示されます。
6. 「Client Auth」の「Off」をクリックして、これをオンにします。
7. 「OK」をクリックします。
8. Server Manager 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、各 Web サーバ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は、別個の信頼データベースを使用して、異なるサーバインスタンスで稼動する必要があります。

クライアント証明書を LDAP へマップする

この節では、iPlanet 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 8 章「サーバへのアクセス制御」を参照してください。

サーバは、Administration Server 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、iPlanet Web Server はその接続を終了します。一致した場合、サーバ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) 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 (エンドユーザ名、電子メールアドレスなど) をサーバに通知します。サーバは、これらの値を使用して LDAP ディレクトリ内にユーザ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に通知します。

サーバが、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 5-1 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 iPlanet Web Server がユーザ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、166 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 5-1 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ーなし | 認証失敗 | 認証失敗 |
| 検出されたエントリーが 1 つのみ | 認証失敗 | 認証成功 |
| 検出されたエントリーが複数 | 認証失敗 | 認証失敗 |

サーバが LDAP ディレクトリ内で一致するエントリー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、一部のサーバでは、サーバ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エントリーをサーバ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 エントリーにマップする方法を構成できます。このファイルを編集し、エントリーを追加して、ユーザの LDAP ディレクトリの組織に一致するようにし、ユーザが持っているべき証明書をリストにするようにします。ユーザは、subjectDN 内で使用されているユーザ 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 ツリー内でサーバが検索を開始する場所
- LDAP ディレクトリ内のエントリーを検索するときにサーバが検索条件として使用するべき証明書の属性
- サーバ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構成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ません。たとえば、次の 2 つの `issuerDN` 行は、属性間にスペースがあるかどうかという点が異なるだけですが、サーバは、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 iplanet1 ou=iPlanet Certificate Authority,o=iPlanet,c=US
certmap iplanet2 ou=iPlanet Certificate Authority,o=iPlanet, c=US
```

ヒント `iPlanet 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 を構成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。サーバが LDAP ディレクトリ内で、証明書から収集したユーザ情報に一致する 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が検証を行います。

たとえば、電子メール属性とユーザ ID 属性を使用するよう `FilterComps` を設定すると (`FilterComps,uid`)、電子メールとユーザ 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の情報と一致するエントリを、サーバ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 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 5-2 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 または Linux のユーザ 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、値が共有ライブラリまたは 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 プログラマーズガイド*』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、123 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を作成する

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 プログラマーズガイド*』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>
```

次に例を示します。

```
certmap default1 o=Netscape Communications, c=US
default1:library /usr/netscape/enterprise/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 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
```

サーバが US Postal Service 以外から証明書を取得している場合、サーバ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 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。ただし、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。

注意 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リストされている発行者 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 属性の間にスペース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 を設定する

「Stronger Ciphers」オプションでは、アクセスするための秘密鍵のサイズに 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、iPlanet 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」の下にある現在の符号化方式の設定と整合しないサイズを選択すると、iPlanet Web Server が、符号化方式でより大きいサイズの秘密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、秘密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秘密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` のステータスと一緒に `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. **Server Manager** にアクセスし、サーバ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. 秘密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。

その他のセキュリティに関する問題

他人が暗号を解読しようとする以外にも、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、サーバ本体やサーバ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マシンを設置し、信頼できない個人にサーバ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ングを防ぐ
- ポートを制限する
- サーバの限界を知る
- サーバを保護するためその他の追加変更を行う

物理的アクセスを制限する

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マシンを設置します。このようにすると、サーバ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。

管理アクセスを制限する

リモート構成を使用している場合、必ず、数人のユーザ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。Administration Server からエンドユーザの権限で LDAP サーバやローカルディレクトリにアクセスさせる場合、2 台の Administration Server でクラスタ構成を整え、1 台では SSL を有効にしてマスタとなる Administration Server を構成し、もう 1 台のサーバでエンドユーザ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、135 ページの「クラスタについて」を参照してください。

Administration Server 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。

確実なパスワードを選択する

サーバ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の構成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が第三者に入手された場合、第三者によって偽のサーバが作成され、ユーザ自身のものであるように見せかけることができたり、サーバとの間の通信を傍受されたり、改ざんされたりする可能性があります。

優れたパスワードは、ユーザ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。

破られにくいパスワードを作成する

安全性の高いパスワードを作成するのに役立つ、いくつかのシンプルなガイドラインを示します。

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、パスワードを破られにくくなります。

- 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 (*、"、スペース)
- 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 を変更する

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。Administration Server で SSL を有効にしている場合、サーバ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、サーバのセキュリティ保護のレベルが高まります。

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るガイドラインのリストは、128 ページの「破られにくいパスワードを作成する」を参照してください。

パスワードを変更する

Administration Server またはサーバ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. Administration Server または Server Manager にアクセスし、「Security」タブを選択します。

Server Manager 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. Server Manager を使用の場合には、「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。

鍵ペアファイルが必ずセキュリティ保護されるようにします。Administration Server 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 iPlanet 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と同等に、完全にプロテクトする必要があります。

サーバ上で他のアプリケーションを制限する

サーバ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、サーバのセキュリティが回避される可能性があるからです。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` をサーバマシン上で起動させないでください。また、サーバマシン上に `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るからです)。

Windows NT

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、管理者がサーバ上に置いているプログラムや、サーバ上で他のユーザ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ングするのを前もって防止することができます。

```
<meta http-equiv="pragma" content="no-cache">
```

ポートを制限する

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構成を使用して、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構成します。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。

サーバの限界を知る

サーバは、サーバ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。サーバは、情報のセキュリティを一度クライアントに取得されると、もうそれを制御することはできず、サーバ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。

サーバを保護するためその他の追加変更を行う

保護されているサーバと保護されていないサーバの両方が必要な場合には、保護されているサーバとは異なるマシンで保護されていないサーバ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と同じマシン上で保護されていないサーバを稼動しなくてはならない場合には、次のようにしてください。

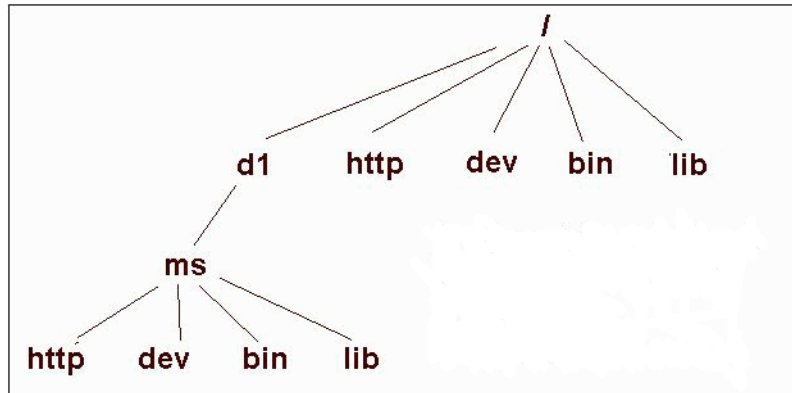
- 適切なポート番号を割り当てます。保護されているサーバと保護されていないサーバに、必ず異なるポート番号が割り当てられる必要があります。登録されているデフォルトのポート番号は、次のとおりです。
 - 443 (保護されているサーバ用)

- 80 (保護されていないサーバ用)
- 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の保護対策としてこの機能を使用することができます。たとえば、ルートディレクトリが /d1/ms であるということもできます。そのとき、Web サーバが、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に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、iPlanet 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セットアップする必要があります。

図 5-2 chroot ディレクトリ構造の例



仮想サーバクラスに chroot を指定する

次の手順で、仮想サーバクラスに chroot ディレクトリを指定できます。

1. Server Manager にアクセスし、サーバのサーバ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. 「Option」が、chroot を指定したいクラスについて、「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、動的に適用します。

仮想サーバに chroot を指定する

次の手順で、特定の仮想サーバに chroot ディレクトリを指定できます。

1. Server Manager にアクセスし、サーバのサーバ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、動的に適用します。

「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。

仮想サーバに chroot ディレクトリを指定する方法についての詳細は、iPlanet Web Server の『プログラマーズガイド』を参照してください。

その他のセキュリティに関する問題

サーバクラスタの管理

この章では、iPlanet Web Server のクラスタリングの概念を解説し、クラスタを使用してサーバ間で構成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の追加
- サーバ情報の変更
- クラスタからのサーバの削除
- サーバクラスタの制御
- 変数の追加

クラスタについて

クラスタとは、1つの Administration Server から管理することができる、複数の iPlanet Web Server で構成されたグループのことです。各クラスタには、管理サーバとして指定された 1つのサーバ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」Administration Server 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 iPlanet 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 iPlanet Web Server を集中して管理する
- 1つ、または複数の構成ファイルをサーバ間で共有する

- 1つの「マスター」Administration Server から、すべてのサーバの起動または停止を行う
- 指定したサーバの、アクセスログやエラーログを表示する

iPlanet Web Server をクラスタリングすることで、マスター Administration Server 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 Administration Server は、クラスタを構成するすべてのサーバの情報を持っており、クラスタ内のすべての Administration Server にアクセスできるように設定する必要があります。

サーバクラスタの使用に関するガイドライン

クラスタを構成する際、すべてのクラスタの情報を持つマスター Administration Server は、クラスタの各 Administration Server と通信します。クラスタの各 Administration Server には、マスター Administration Server と同じ管理ユーザ名とパスワードを登録しておく必要があります。

クラスタを構成する場合は、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 iPlanet 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をインストールし、各サーバが、マスター Administration Server と同じ管理ユーザ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 iPlanet Web Server を Administration Server として構成します。
3. クラスタに含まれる管理サーバのうち 1 つを選択し、すべてのクラスタに対するマスター Administration Server として構成します。マスター Administration Server として選択するのは、どの Administration Server でもかまいません。

注意 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は、UNIX または NT 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 NT のサーバが混在していると、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がすべて、iPlanet Web Server version 6.0であることを確認します。
- すべてのクラスタに固有の Administration Server が、マスター Administration Server 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 Administration Server 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、NT 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 Administration Server を、マスター Administration Server に指定することができます。
- マスター Administration Server がクラスタ固有の各 Administration Server にアクセスできることを確認します。マスター Administration Server は、すべてのインストールされている iPlanet Web Server に関する情報を取得します。
- Administration Server はすべて、iPlanet Web Server version 6.0 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 iPlanet Web Server version 6.0 だけです。
- 1つのクラスタ内の1つの Administration Server のプロトコルを変更する場合は、すべての残りの Administration Server 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の設定を変更できます。

クラスタの設定

iPlanet 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. iPlanet 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 Administration Server が、マスター Administration Server 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 Administration Server に使用するサーバ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. サーバをクラスタリストに追加します。

4. リモートサーバの管理は、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の構成ファイルを別のサーバにコピーして行います。

注 リモートサーバの構成を変更したら、リモートサーバを再起動します。

クラスタへのサーバの追加

クラスタにサーバを追加する際は、そのクラスタを管理している **Administration Server** とポート番号を指定します。追加する **Administration Server** が複数のサーバ情報を持っている場合、すべてのサーバが、そのクラスタに追加されます。個々のサーバは、後から削除することができます。

注 リモート **Administration Server** 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は追加されません。マスター **Administration Server** に追加するサーバは、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だけです。

クラスタにリモートサーバを追加するには、次の手順を実行します。

1. マスター **Administration Server** が起動していることを確認します。
2. マスター **Administration Server** 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 **Administration Server** が使用するプロトコルを選択します。
 - http (通常の **Administration Server** の場合)
 - https (セキュリティ保護された **Administration Server** の場合)
5. 「**Admin Server Hostname**」フィールドに、リモートサーバの `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 : `jodib.iplanet.com`
6. リモート **Administration Server** 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 **Administration Server** は、リモートサーバ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識別子とホスト名が表示されます。サーバ識別子とホスト名が両方とも同じサーバが存在する場合には、ポート番号も表示されます。

サーバ情報の変更

「Modify Server」オプションは、スレーブサーバ上で、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 Administration Server 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 Administration Server の情報も変更する必要があります。スレーブ Administration Server に対するその他の変更の場合は、一旦そのサーバを削除し、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 Administration Server 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 Administration Server 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。
一意のサーバ識別子でリストされた、すべてのサーバが表示されます。
3. 次のどちらかで、変更するサーバを1つまたは複数選択します。
 - 特定のサーバにチェックマークを付ける
 - 「Select All」をクリック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の削除

クラスタからサーバを削除するには、次の手順を実行します。

1. マスター Administration Server 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を1つまたは複数選択します。
 - 特定のサーバ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、そのサーバの Administration Server からのみとなります。

サーバクラスタの制御

iPlanet Web Server 6.0 を使用すると、クラスタ内のリモートサーバを、次のように制御することができます。

- リモートサーバを起動または停止する
- アクセスログやエラーログを閲覧する
- 構成ファイルをサーバに転送する

注意 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は、UNIX または NT のいずれかで統一されていなければなりません。構成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を制御するには、次の手順を実行します。

1. マスター Administration Server の「Server Manager」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、制御するサーバを、1つまたは複数、選択します。

- 特定のサーバ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. 構成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、転送する構成ファイルを選択します。
 - b. ドロップダウンメニューから、構成ファイルの転送元であるサーバを選択します。
 - 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に複数の異なる値を構成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だけ影響します。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 Administration Servers は影響を受けません。変数が定義されると、Administration Server 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 Administration Server 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。

例: Port

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の構成ファイルにも追加する必要があります。次に例を示します。

`SERVERPORT $Port` (ポートの変数が追加された場合)

構成ファイルで、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。

構成、監視、パフォーマンスの調整

第 7 章「サーバの詳細設定」

第 8 章「サーバへのアクセス制御」

第 9 章「ログファイルの使用」

第 10 章「サーバの監視」

第 11 章「サーバのパフォーマンスの調整」

第 12 章「検索機能の使い方」

サーバの詳細設定

この章では、iPlanet Web Server のサーバの詳細設定の方法について説明します。

この章は、次の節で構成されます。

- サーバの起動と停止
- サーバ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
- 構成の復元
- ファイルキャッシュの構成
- スレッドプールの追加と使用

サーバの起動と停止

UNIX では、iPlanet 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。

サーバがインストールされると、サーバは常時稼働して HTTP 要求を待機し、受け取ります。

サーバのステータスが「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 NT: 「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` を使用します。これによって、サーバが完全にシャットダウンされ、再起動するまでサービスは中断されます。`etc/inittab` ファイルを設定して自動的に再起動する (`respawn` を使用して) よう設定している場合は、サーバをシャットダウンする前に `etc/inittab` 内の Web サーバに関連する行を削除する必要があります。そうしないと、サーバが自動的に再起動します。

サーバ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、iPlanet 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が待機する時間が、より長くなることです。ただし、サーバ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。

サーバの再起動 (UNIX/Linux)

以下のいずれかの方法で、サーバを再起動できます。

- `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 が有効なサーバを自動的に起動できますが、この方法は推奨されません。

| | |
|-----------|--|
| 注意 | SSL が有効なサーバ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なら誰でも、SSL が有効なサーバのパスワードにアクセスできます。SSL が有効なサーバ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がサーバをインストールした場合は、そのユーザのアカウントが所有している)、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 が有効なサーバを自動的に起動

セキュリティ上の危険性が問題にならない場合は、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:2:respawn:server_root/type-identifier/start -start -i
```

`server_root` はサーバをインストールしたディレクトリ、`type-identifier` はサーバのディレクトリです。

`-i` オプションは、サーバ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、サーバがインストールされているディレクトリに変更します。

手動によるサーバの再起動 (UNIX/Linux)

コマンド行からサーバを再起動するには、1024 より小さい番号のポートでサーバを実行している場合は、ルートとしてログインします。1024 以上の番号の場合は、ルートとして、またはそのサーバのユーザ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` モードでサーバが実行されます。`inittab` モードでは、サーバ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` がサーバを自動的に再起動します。また、このオプションは、サーバ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。

手動によるサーバの停止 (UNIX/Linux)

`etc/inittab` ファイルを使用してサーバを再起動した場合は、サーバの停止を試行する前に、`/etc/inittab` からサーバ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、サーバは停止した後で自動的に再起動してしまいます。

サーバを手動で停止するには、`root` として、またはサーバのユーザ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
```

サーバの再起動 (Windows NT)

以下の方法でサーバを再起動できます。

- 「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または `Administration Server` を再起動するようにオペレーティングシステムを構成します。

Windows NT 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、サーバが使用するアカウントを変更することもできます。

デフォルトでは、起動する前に、Web サーバから管理者に対して、鍵データベースパスワードの入力を求めるプロンプトが表示されます。Web サーバ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十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 NT)

サーバに障害が発生した場合、サーバ監視ユーティリティによって、サーバ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 NT)

Windows NT の起動後、サーバ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」メニューの「新規作成」から「DWORD 値」を選択します。
4. データの名前を「MortalityTimeSecs」と入力します。
5. 「編集」メニューの「変更」を選択します。「DWORD 値の編集」ダイアログボックスが表示されます。
6. Windows NT の起動後、サーバが自動的に再起動されるまでに経過する時間間隔 (秒) を入力します。
間隔は、10 進数、または 16 進数の形式にすることができます。
7. 前の手順で入力した値の数値形式 (10 進数、または 16 進数) をクリックします。
8. 「OK」をクリックします。

「レジストリエディタ」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 NT)

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、サーバは再起動されません。

サーバ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. レジストリエディタを起動します。
2. 「レジストリエディタ」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。
「文字列の編集」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、`Server Manager` 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 512 です。

`Server Manager` 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」フィールドに希望する値を入力します。

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

iPlanet Web Server は起動時に、`server_root/server_id/config` ディレクトリ内の `magnus.conf` ファイルを参照し、サーバの動作と構成に影響するグローバル変数の設定を確立します。iPlanet Web Server は `magnus.conf` で定義されているすべての指令を実行します。Server Manager の Magnus 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、テキストエディタを使用したファイルの編集については、『NSAPI プログラマーズガイド』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. Server Manager 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

サーバが要求を処理するには、待機ソケットから要求を受け取り、その要求を適切な接続グループと仮想サーバに振り向ける必要があります。iPlanet 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) 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。

サーバの待機ソケットの設定は、Server Manager 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. Server Manager 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、サーバの MIME ファイルを編集できます。

MIME (Multi-purpose Internet Mail Extension) 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に関連付けます。サーバには `mime.types` という MIME タイプファイルがデフォルトで存在し、削除することはできません。このファイルは絶対パスにすることもできます。

「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. Server Manager 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. 変更が行い、「OK」 をクリックします。

詳細は、オンラインヘルプの「[Mime Types](#)」 ページ、および第 13 章「仮想サーバの使用」を参照してください。

アクセスの制限

Server Manager の「Restrict Access」 ページを使用して、サーバ全体またはサーバの一部 (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。要求がサーバに着信すると、サーバは `vsclass.obj.conf` (`vsclass` は仮想サーバのクラス名) で ACL への参照を確認します。この ACL への参照はアクセス権を判定するために使用されます。デフォルトでは、サーバには、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、Administration Server を介してすべてのサーバに対してグローバルに設定することも、Server Manager を介して特定のサーバインスタンス内のリソースに対して設定することもできます。リソースに対するアクセス制御の設定の詳細は、第 8 章「サーバへのアクセス制御」の 169 ページの「アクセス制御の設定」を参照してください。

注 サーバへのアクセスを制限する前に、分散管理を有効にしておく必要があります。

iPlanet 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. **Server Manager** 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 8 章「サーバへのアクセス制御」とオンラインヘルプの「「**Restrict Access**」ページ」を参照してください。

構成の復元

「**Restore Configuration**」ページでは、構成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構成データに戻すことができます。

注 Windows NT では、構成ファイルに対して自分が行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構成

iPlanet Web Server では、ファイルキャッシュを使用して、静的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、iPlanet Web Server 6.0 バージョンでは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静的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、**Server Manager** を使用して変更することができます。

詳細は、<http://docs.ipplanet.com/docs/manuals/enterprise.html> (英語)で、オンラインの『Performance Tuning, Sizing, and Scaling 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 - 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.ipplanet.com/docs/manuals/enterprise.html> (英語)で、オンラインの『Performance Tuning, Sizing, and Scaling 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 NT)

Windows NT 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、Server Manager 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、Server Manager 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 ユーザは NativePool を使用する必要がなく、この設定を編集するための Server Manager のページもありません。ただし、UNIX/Linux ユーザがスレッドプールを作成することはできます。スレッドプールを作成するには、Server Manager 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、**Server Manager** でスレッドプール設定の値 (最小スレッド数、最大スレッド数など) を変更できます。

また、`magnus.conf` でもスレッドプール設定を編集できます。

`magnus.conf` で、スレッドプールは以下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 NT ユーザは **Server Manager** 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に対する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。

スレッドプールを構成するには、**Server Manager** の「**Preferences**」タブをクリックし、「**Thread Pool**」を選択します。スレッドプールが構成されると、「**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、`magnus.conf` の `load-modules` 関数の `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。

サーバへのアクセス制御

この章では、Administration Server 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、Administration Server 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を完全に制御できるユーザや、1 台以上のサーバを部分的に管理できるユーザを指定できます。Administration Server 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、すでに分散管理を構成していて、LDAP データベースにユーザとグループを定義していることを前提としています。

第 5 章「Web サーバのセキュリティ」で説明されている Web サーバのセキュリティについても、確認する必要があります。

この章は、次の節で構成されています。

- アクセス制御とは
- アクセス制御の実行方法
- アクセス制御の設定
- アクセス制御オプションの選択
- サーバの一部へのアクセス制御
- 動的アクセス制御ファイルの使用
- 仮想サーバへのアクセス制御

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。

- iPlanet Web Administration Server にアクセスできるユーザ
- ユーザ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

サーバ全体やサーバの一部、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、サーバには、複数の ACL が含まれる 1 つの ACL ファイルがあります。iPlanet Web Server は、受信する要求に使用する仮想サーバを指定したあと、その仮想サーバに対して ACL が構成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、iPlanet 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 (ユーザ - 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 - グループのアクセス制御の設定

特定のユーザまたはグループに対して、Web サーバへのアクセスを制限することができます。ユーザ - グループのアクセス制御を設定すると、ユーザはユーザ名とパスワードを入力しないと、サーバへのアクセス権を取得できなくなります。サーバは、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のエントリと比較します。

Administration Server では、基本認証だけを使用します。Administration Server 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。

サーバインスタンスに対するユーザ - グループ認証メソッドには、次の内容が含まれます。なお、() 内は画面上の表示を示しています。

- デフォルト (Default)

- 基本 (Basic)
- SSL (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が必要です。

ユーザ - グループ認証の場合、ユーザは自分自身の認証を行わないと、Administration Server、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、ユーザはユーザ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 5 章「Web サーバのセキュリティ」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 サーバまたは Web サイトにアクセスするユーザに対して、ユーザ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。iPlanet Directory Server などの LDAP データベースにユーザ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 サーバではなく、別のサーバルトにインストールされているディレクトリサーバ、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を使用する必要があります。

Administration Server 内、または Web サイト上のユーザ - グループ認証が設定されているリソースにユーザがアクセスした場合、Web ブラウザにユーザ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名とパスワードがネットワークを經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。

ユーザがサーバ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。

図 8-1 ユーザ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。

- iPlanet Web Administration Server へのアクセスに対する認証を受けている場合は、「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)

サーバは、次の 2 つの方法で、セキュリティ証明書付きのユーザ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を設定した場合、サーバ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、117 ページの「クライアント認証を要求する」を参照してください。
- 証明書が信頼できる認証局 / 認証機関 (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、120 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 サーバでは SSL 符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 5 章「Web サーバのセキュリティ」を参照してください。

SSL で認証されるリソースへのアクセス権を得るには、Web サーバで信頼されている 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のクライアント証明書を比較するように Web サーバの `certmap.conf` ファイルが構成されている場合、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のエントリだけを比較するように、`certmap.conf` ファイルを構成することもできます。たとえば、ブラウザ証明書のユーザ ID および電子メールアドレスとディレクトリサーバのエントリだけを比較するように、`certmap.conf` ファイルを構成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 5 章「Web サーバのセキュリティ」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、ユーザがユーザ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、ユーザのパスワードと Web サーバによって提供される情報の一部を使用し、MD5 アルゴリズムを使って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、ユーザは認証を受けたものと見なします。

このように機能させるためには、ディレクトリサーバをプレーンテキスト形式の (暗号化されていない) ユーザパスワードにアクセスできるようにする必要があります。iPlanet Directory Server 5.0 には、対称暗号化アルゴリズムを使用してデータを暗号化した形式で格納し、あとから復号化して元の形式に戻すことができるリバーシブルパスワードプラグインが組み込まれています。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、iPlanet Web Server 6.0 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 サーバを構成するには、`dbswitch.conf` でデータベース定義の `digestauth` プロパティを設定します。

表 8-1 に示すように、サーバは指定されている 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行します。ACL メソッドを指定しないと、サーバは認証が要求された場合にダイジェスト認証または基本認証のどちらかを使用し、認証が要求されない場合には基本認証を使用します。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 8-1 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 |

表 8-1 ダイジェスト認証要求の生成 (続き)

| 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 |
|----------|-----------------------------|------------------------------|
| 「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、サーバは次の内容を実行して認証を試行します。

- 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、サーバは次の内容を実行します。
 - nonce を確認します。このサーバ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。
 - 領域を確認します。領域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、iPlanet Directory Server がインストールされているのと同じサーバ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
```

NT 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 iPlanet Directory Server を正常に起動できるようにするには、iPlanet 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 iPlanet Directory Server のサーバマシンにいくつかコピーする必要があります。

NT 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 iPlanet 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。

- o nsldap32v50.dll
- o libspnr4.dll
- o 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。

- o \Winnt\system32
- o iPlanet 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ように iPlanet Directory Server を設定する

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 iPlanet Directory Server を設定するには、次の手順を実行します。

1. iPlanet Directory Server Console 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. プラグインの横にある + 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. iPlanet Directory Server のインスタンスを再起動します。

注 `iplanetReversiblePassword` 属性でユーザ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

Administration Server、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、シームレスにユーザに表示されます。このため、各ユーザは、ユーザ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が使用しているため、ホスト - IP 認証は、ユーザ - 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はユーザ名とパスワードが要求されます。

ホスト - IP 認証では、サーバが動作しているマシンで DNS を構成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が構成されている必要があります。サーバに対して DNS を有効にするには、Server Manager 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、サーバで DNS ルックアップを実行する必要があるため、iPlanet Web Server のパフォーマンスが低下します。サーバ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" 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

Administration Server 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 `server_install/httpacl` ディレクトリに格納されます。`server_install` はサーバがインストールされている場所です。たとえば、`/usr/iPlanet/Servers` にサーバをインストールした場合、Administration Server とサーバ上で構成されている各サーバインスタンスの両方の ACL ファイルが、`/usr/iPlanet/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。`server-id` はサーバ ID を示します。iPlanet Administration Server を使用してアクセスを構成する場合は、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、『プログラマーズガイド』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 ユーザキャッシュの構成

デフォルトでは、iPlanet Web Server によるユーザとグループの認証の結果が、ACL ユーザ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。キャッシュ内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` に対してチェックされます。経過時間が `ACLCacheLifetime` と同じか、それよりも長い場合、そ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 iPlanet 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、iPlanet 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構成できます。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、ユーザエン트리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 プログラマーズガイド』を参照してください。

アクセス制御の実行方法

サーバ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、Administration Server (`admin-serv`) に対する 2 つのデフォルトエン트리と、「`admin-reduced`」グループ内のユーザが Administration Server の「Preferences」タブにアクセスできるようにするための追加エン트리があります。

```
version 3.0;
# The following "es-internal" rules protect files such
# as icons and images related to iPlanet 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 iPlanet 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、ユーザが次の URL を要求する場合、
`http://server_name/my_stuff/web/presentation.html`

iPlanet Web Server はまず、サーバ全体へのアクセス制御を確認します。サーバ全体への 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。

Server Manager 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。

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。

Administration Server によって、すべてのサーバ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、分散管理を構成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. Administration Server 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 (https-admserv) を選択します。
 4. 次のどちらかをクリックします。
 - 「Create ACL」
 - 「Edit ACL」
- 「Access Control Rules for uri=/https-admserv/bin/」ページが表示されます。

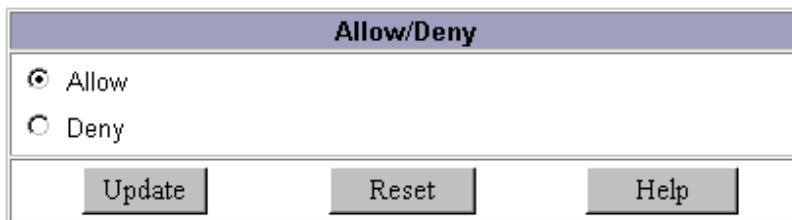
図 8-2 「Access Control Rules」ページ

| Access Control Rules for : default | | | | | | |
|--|------------------------|---|------------------------|-------------------------------------|-------------------------------------|---|
| Action | Users/Groups | From Host | Rights | Extra... | Continue | |
| 1 Allow | anyone | anyplace | r-x-li | x | <input checked="" type="checkbox"/> |  |
| 2 Allow | all | anyplace | -w-d-- | x | <input checked="" type="checkbox"/> |  |
| <input checked="" type="checkbox"/> Access control is on | | <input type="button" value="New Line"/> | | | | |
| Current Access deny response is the default file (redirection off) | | | | | | Response when denied |
| <input type="button" value="Submit"/> | | <input type="button" value="Revert"/> | | <input type="button" value="Help"/> | | |

Administration Server 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. グローバル ACL の作成や編集をするには、「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。

図 8-3 「Allow/Deny」 ページ



Allow/Deny

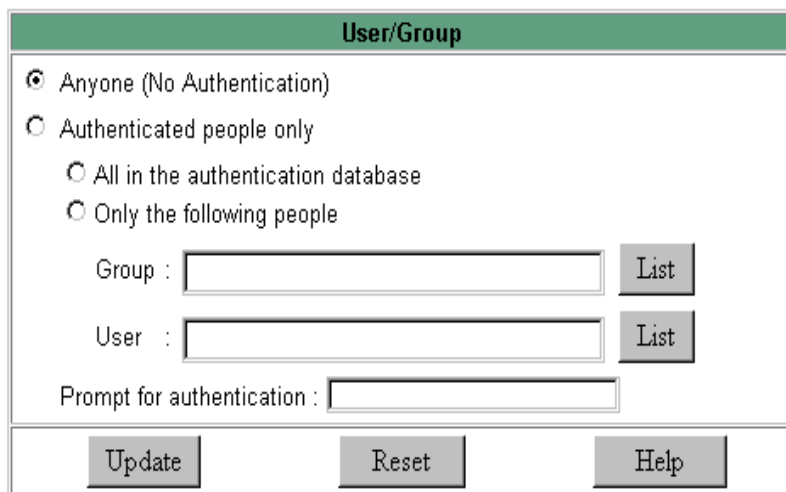
Allow

Deny

Update Reset Help

7. 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。
8. 「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。

図 8-4 「User/Group」 ページ



User/Group

Anyone (No Authentication)

Authenticated people only

- All in the authentication database
- Only the following people

Group : List

User : List

Prompt for authentication :

Update Reset Help

9. アクセスを許可するユーザ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。
10. 「From Host」列の「anyplace」をクリックします。
11. アクセスを許可するホスト名と IP アドレスを入力してから、「Update」をクリックします。
12. 「Programs」列で「All Programs」をクリックします。

図 8-5 「Programs」

13. 「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。
14. (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。
15. 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。
サーバは次の行を評価してから、そのユーザ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。
16. (省略可能) 別の URL または URI にユーザを誘導することを拒否する場合、「Response」をクリックします。
17. 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。
18. 「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削除されます。

サーバインスタンスに対するアクセス制御の設定

Server Manager を使用すると、特定のサーバ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を再起動すると、構文エラーが発生します。

サーバ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. Server Manager にアクセスし、ACL を作成または編集するサーバインスタンスを選択します。
2. Server Manager 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。

図 8-6 「Access Control List Management」 ページ

Access Control List Management

Select an ACL using one of the three methods below:

A. Pick a resource

Editing:

B. Pick an existing ACL

Editing:

C. Type in the ACL name

Editing:

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 8-2 では、使用できるリソースワイルドカードについて説明します。

表 8-2 サーバ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 |
| サーバ全体 | 1 組の規則によって、稼働中の仮想サーバを含めた 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 |



表 8-2 サーバリソースのワイルドカード (続き)

| リソースのワイルドカード | 意味 |
|-------------------------------------|--|
| /usr/iplanet/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 NT 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: *server instance*」が、表示されます。

図 8-7 「Access Control Rules」 ページ

Access Control Rules for : default

| | Action | Users/Groups | From Host | Rights | Extra... | Continue | |
|---|-------------------------|------------------------|--------------------------|------------------------|-------------------|-------------------------------------|---|
| + | 1 Allow | anyone | anyplace | r-x-li | x | <input checked="" type="checkbox"/> |  |
| + | 2 Allow | all | anyplace | -w-d- | x | <input checked="" type="checkbox"/> |  |

Access control is on New Line

Current Access deny response is the default file (redirection off) [Response when denied](#)

Submit
Revert
Help

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。

図 8-8 「Allow/Deny」 ページ

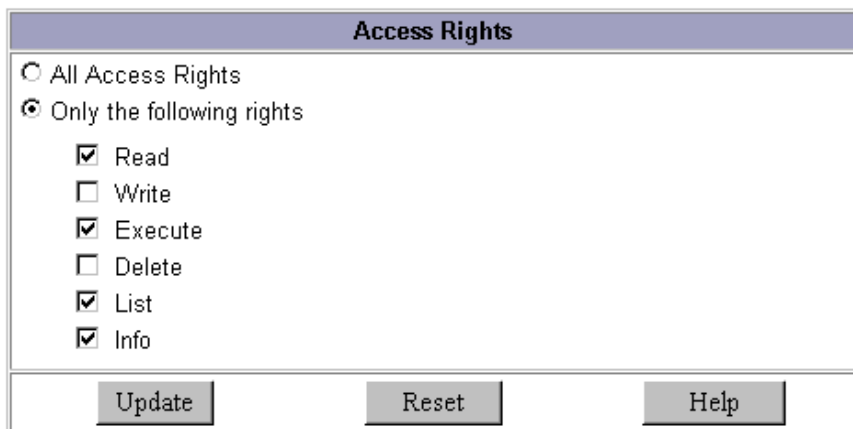
9. デフォルトで選択されていない場合は「Allow」を選択し、「Update」をクリックします。
10. 「Users/Groups」列の「anyone」をクリックします。
下のフレームに「User/Group」ページが表示されます。

図 8-9 「User/Group」 ページ

11. アクセスを許可するユーザ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。

図 8-10 「Access Rights」ページ



The screenshot shows a dialog box titled "Access Rights". It contains two radio buttons: "All Access Rights" (unselected) and "Only the following rights" (selected). Below the second radio button, there are six checkboxes: "Read" (checked), "Write" (unchecked), "Execute" (checked), "Delete" (unchecked), "List" (checked), and "Info" (checked). At the bottom of the dialog, there are three buttons: "Update", "Reset", and "Help".

15. 次のどちらかを選択し、「Update」をクリックします。

- 「All Access Rights」
- 「Only the following rights」をクリックし、このユーザ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。

サーバは次の行を評価してから、ユーザ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、202 ページの「仮想サーバ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。Administration Server 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が実行するアクションを指定できます。

- 「Allow」は、ユーザまたはシステムが、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、サーバはリストの 2 番目の 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、すべてのユーザがリソースへのアクセスを拒否されます。サーバは、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。

ユーザとグループの指定

ユーザとグループの認証が行われる場合、ユーザ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、ユーザ名とパスワードを入力する必要があります。

iPlanet Web Server は、iPlanet Directory Server などの LDAP サーバに格納されているユーザ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、すべてのユーザがユーザ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のアクセスが拒否される場合もあります。Administration Server 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ーがあるユーザに一致します。
 - 「**Only the following people**」では、一致するユーザとグループを指定できます。エントリーをコンマ (,) 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、ユーザとユーザ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に一致します。「**User**」は、指定した個々のユーザに一致します。Administration Server では、ユーザ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、ユーザが入力する必要のある内容を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、プロンプトに最初の約 40 文字が表示されます。Netscape Navigator と Netscape Communicator では、ユーザ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がアクセスする場合は、ユーザ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が再び認証を受けたい場合、そのリソースの ACL に対するプロンプトを変更する必要があります。
- 「**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で使用するメソッドを指定します。Administration Server 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。ユーザ名とパスワードが暗号化されるのは、サーバ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の認証を行います。このメソッドを使用するには、サーバ側で 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、ユーザのパスワードと Web サーバ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、サーバでユーザ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、Server Manager を使用する場合だけです。「Default」を選択した場合、サーバは LDAP ディレクトリ内のユーザ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 を構成する場合、「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、Administration Server 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カ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、特定のドメイン内のすべてのコンピュータに対してアクセスを許可または拒否する場合、*.iplanet.com のように、ドメイン内のすべてのホストが該当するワイルドカードパターンを入力します。Administration Server にアクセスしているスーパーユーザに対しては、その他のユーザ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はなりません。つまり、*.iplanet.com は許容されますが、*users.iplanet.com は許容されません。また、ホスト名で * を使用する場合、この記号が文字列の一番左に配置される必要があります。たとえば、*.iplanet.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、この記号が文字列の一番右に配置される必要があります。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、Administration Server だけです。プログラムへのアクセス制限を適用すると、特定のユーザだけが「Server Manager」ページを参照し、そのサーバを構成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 Administration Server の「Users & Groups」セクションを構成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。

- 「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
 - Preference
 - Global Settings
 - Users & Groups
 - Security

○ Cluster Mgmt

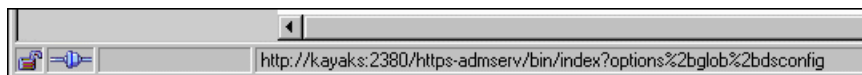
リストに表示されたプログラムグループは、Administration Server のタブに反映されます。たとえば、「Preferences」や「Global Settings」などのタブに反映され、各ページへのアクセスを表します。管理者が Administration Server にアクセスする場合、サーバは管理者のユーザ名、ホスト、および IP を使用して、参照できるページを特定します。

- 「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。

ページ名を調べるには、Administration Server の左側のフレームに表示されるリンクにポインタを置き、ブラウザの下部にあるステータスバーに表示されるテキストを参照します。テキストの中で、最後に出現する %2b よりもあとの部分がページ名です。

たとえば、iPlanet Directory Server を管理しているユーザに「Configure Directory Service」のページへのアクセスを許可する場合、そのユーザ(ホスト、IP など)だけに適用される規則を設定し、「Program Items」フィールドに「dsconfig」と入力します。

図 8-11 ページ名 / プログラムアイテム



アクセス権の設定

サーバインスタンスに対するアクセス権を設定できるのは、Server Manager 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承認または拒否に加えて、一部のアクセス権の承認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、ユーザにファイルの変更や削除を許可します。これには PUT、DELETE、MKDIR、MDIR、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」と 410 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.ac1 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アウトします。

Administration Server からアクセス制御を作成し、特定のサーバインスタンスに対して有効に設定し、ほかのサーバに対しては無効 (デフォルト) のままにしておくことができます。たとえば、Administration Server から「Server Manager」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 Administration Server を構成することはできませんが、ほかのサーバにアクセスして構成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に対して、分散管理のために設定されます。Administration Server はまず、ユーザ (スーパーユーザを除く) 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、iPlanet Web Server はデフォルトメッセージ「FORBIDDEN. Your client is not allowed access to the restricted object.」を送信します。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。

サーバの一部へのアクセス制御

この節では、Web サーバ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、実行する必要がある操作を詳細に説明しています。ただし、173 ページの「サーバ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に対して、サブドメイン内のコンピュータからサーバ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に対して許可したい場合です。

サーバ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. **Server Manager** を使用して、サーバ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.iplanet.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。

サーバのディレクトリへのアクセスを制限するには、サーバ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. **Server Manager** を使用して、サーバ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:\iPlanet\suitespot\docroot1\sales/";
```

URI (パス) へのアクセス制限

URI を使用して、Web サーバ上のシングルユーザ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. **Server Manager** を使用して、サーバ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。
次に例を示します。uri=/my_directory
5. 「**Edit Access Control**」 をクリックします。
6. すべてのユーザ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。

```
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、特定のユーザだけに、サーバ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. **Server Manager** を使用して、サーバ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。
たとえば、「*.cgi」と入力します。
5. 「Edit Access Control」をクリックします。
6. すべてのユーザ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があります。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. **Server Manager** を使用して、サーバ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、「Edit Access Control」をクリックします。
5. すべてのユーザ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、ユーザがファイルやディレクトリの追加、更新、または削除を行いたい場合にはこの規則が適用されず、サーバは該当する別の規則を検索することを意味します。

6. すべてのユーザに対して読み取り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。

次に、その例を示します。

```
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

iPlanet Web Server 6.0 では、1つのサーバインスタンスに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構成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. **Server Manager** を使用して、サーバ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。

次に、入力例を示します。

```
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。

動的アクセス制御ファイルの使用

サーバのすべてのコンテンツが1人のユーザ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、iPlanet Web Server へのアクセス権を与えることなく、一般ユーザが必要な構成を行えるように、構成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。構成オプションのサブセットは、動的構成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

iPlanet Web Server は、動的構成ファイルである .htaccess をサポートします。ユーザインタフェースから、または構成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 - グループ認証が「基本」の場合、ユーザ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

ユーザインタフェースからの .htaccess の有効化

.htaccess を使用するように iPlanet Web Server を設定するには、次の手順を実行します。

1. Server Manager にアクセスし、.htaccess を有効にするサーバ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、編集するサーバを選択します。
 - ドロップダウンリストからサーバ全体または特定のサーバを選択します。
 - 「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 構成を追加するファイル名を入力します。
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 NT 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>/pluglib/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 Administration Server 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、対象ファイルの構成スタイルを作成し、その構成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「構成スタイルの適用」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

iPlanet Web Server 6.0 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。iPlanet Web Server 6.0 では、.nsconfig ファイルがサポートされていません。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 は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 ファイルを検索します。検出された各 .nsconfig ファイルは、.htaccess ファイルに変換されます。構成によっては、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。次に例を示します。

```
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、多数のユーザをグループとして処理できます。多数のユーザ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. ユーザファイルの書式を修正して、ユーザ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

次の内容は、.htaccess ファイルの例です。

```
<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> タグまたは <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、次の形式でユーザ定義が含まれるファイルの名前
username:password
- username はユーザのログイン名で、password は DES で暗号化されたパスワード

<Limit> タグまたは <LimitExcept> タグで囲むことはできません。

効果

指定されたユーザファイルが、`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>` タグまたは `<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名とパスワードのキャッシング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 サーバが理解できるあらゆる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だけに、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限って、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

必ずしも <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

- requires group groupname groupname
- requires user username username
- requires 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

- requires group は、認証を受けるユーザが、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
- requires user は、認証を受けるユーザが、指定したユーザのいずれかであることを要求する
- requires valid-user は、認証されたユーザを要求する

.htaccess セキュリティに関する注意事項

デフォルトでは、サーバによる HTTP PUT のサポートが無効になっています。Class Manager 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。186 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。

仮想サーバへのアクセス制御

iPlanet Web Server 6.0 のアクセス制御についての情報は、各仮想サーバの 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 iPlanet 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。次に例を示します。

```
<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。次に、その例を示します。

```
<VS aclids="standard">
```

この構成では、複数の仮想サーバで同じ ACL ファイルを共有することができます。仮想サーバに対するユーザ - 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>
```

仮想サーバからデータベースへのアクセス

dbswitch.conf ファイルで、ユーザ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 が空白になります。

dcsuffix は dbswitch.conf 内の LDAP データベースの新しい属性で、iPlanet 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 iPlanet LDAP スキーマに準拠し、動作の一部が変更されます。iPlanet LDAP スキーマの詳細と例については、『NSAPI プログラマーズガイド』の第 8 章にある「iPlanet LDAP スキーマ」を参照してください。

仮想サーバごとに、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。仮想サーバに USERDB ブロックがない場合、ユーザ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、仮想サーバの管理者がアクセスするデータベースをサーバ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 の指令の詳細は、『NSAPI プログラマーズガイド』の第8章にある「ユーザデータベースの選択」を参照してください。

ユーザ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認証データベースを定義すると、Class Manager を使用して、各仮想サーバが認証のためにどのデータベースを使用するかを構成できるようになります。また、Class Manger を使用して、仮想サーバでの認証のために `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. Server Manager にアクセスし、「Virtual Server Class」タブを選択します。
2. 「Tree View of the Server」のリストで、仮想サーバクラスのリンクをクリックし、LDAP データベースを指定します。
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. 「Select a Setting」ドロップダウンリストから「ACL Settings」を選択します。
「ACL Settings for Virtual Servers」ページが表示されます。
6. 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。
7. 編集している仮想サーバの「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。
8. 「OK」をクリックします。
9. 「Edit ACL Files」ウィンドウを閉じます。
10. 「Apply」をクリックします。
11. 「dynamically apply」を選択します。

仮想サーバのアクセス制御リストの編集

仮想サーバの ACL は、仮想サーバがあるサーバインスタンスに対して作成されます。仮想サーバの ACL 設定は、サーバ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、各仮想サーバのアクセス制御は **Class Manager** 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に追加します。

仮想サーバの ACL 設定を編集するには、次の手順を実行します。

1. **Server Manager** 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. (省略可能) 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。

ログファイルの使用

複数の方法で、サーバ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、サーバを監視する方法について説明します。組み込み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、第 10 章「サーバの監視」を参照してください。

この章は、次の節で構成されています。

- ログファイルについて
- 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
- ログファイルの保管
- ログの詳細設定
- ログアナライザの実行
- イベントの表示 (Windows NT)

ログファイルについて

サーバのログファイルには、サーバ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を監視すると、障害追跡時に役立ちます。サーバ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、サーバに対する要求とサーバの応答に関する情報が記録されます。iPlanet 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 上で稼働している iPlanet 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。

アクセスログファイルの参照

サーバ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。

Administration Server から Administration Server のアクセスログファイルを参照するには、「Preferences」タブを選択し、「View Access Log」ページを選択します。

Server manager から サーバ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。

Class Manager から個々の仮想サーバ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を選択し、Virtual Server Manager のページで見出しの「Access Log」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 Preference」ウィンドウで指定します。詳細は、209 ページの「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 9-1 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 9-1 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ます。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ます。) |
| RFC 931 information | -(RFC 931 の識別情報は表示されません) |
| 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名) |
| Date/time of request | 29/Mar/2001:4:36:53 -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、209 ページの「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の起動時などのサーバ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、失敗したユーザ認証も記録されます。エラーログを使用して、無効な URL パスや保存場所が見つからないファイルを検索できます。

Administration Server から Administration Server 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。

Server manager からサーバ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。

Class Manager から個々の仮想サーバ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を選択し、Virtual Server Manager のページで見出しの「Error Log」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、エラーログ内のエントリの 2 つの例です。最初の例には、サーバの起動に成功したことを示す情報メッセージを表示します。2 番目の例では、クライアントの wiley.a.com が report.html ファイルを要求したけれども、ファイルがサーバ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):iPlanet-WebServer-Enterprise/6.0
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。指定した時間に、または指定した間隔で、ログがローテーションされます。iPlanet Web Server は古いログファイルを保存し、そのファイルに保存時の日付と時刻が含まれる名前を付けます。たとえば、1 時間おきにアクセスログファイルをローテーションし、iPlanet Web Server でファイルを「access.199907152400」という名前で保存するように設定することができます。ログファイルの名前は、年、月、日、24 時間形式の時間が連結され、1 つの文字列になります。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。

iPlanet 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せん。

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 デーモン内で発生し、起動時にだけ構成を変更できます。内部デーモンログローテーションでは、サーバの再起動を必要とせず、サーバ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.<4 digit year><2 digit month><2 digit day><4 digit 24-hour time>
error.<4 digit year><2 digit month><2 digit day><4 digit 24-hour time>
```

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 12 時で、ローテーションの間隔が 1440 分 (1 日) の場合、現在時刻にかかわらず、保存直後に新しいログファイルが作成され、ローテーションの開始時間まで情報を収集します。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.199907152400 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。

Cron 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 `cron.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。サーバの `cron` 構成オプションは、`server_root/https-admserv/config/` ディレクトリの `ns-cron.conf` に保存されます。`cron` ベースの方法でローテーションされるログは、元のファイル名の後にローテーションされた日時が追加された名前前で保存されます。たとえば、午後 4 時 30 分にローテーションされる `access` は `access.24Apr-0430PM` という名前になります。

ログローテーションは、サーバ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、iPlanet 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、事前にスケジューリングされている「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、iPlanet 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 `cron` ベースの方法のどちらを使用するかを指定するには、Server Manager で「Archive Log Files」ページを使用します。

ログの詳細設定

インストール中、サーバに `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、ロギングに使用する形式、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、リソースに対するアクセスロギングをカスタマイズできます。

複数の仮想サーバに対して1つのログファイルを使用するには、そのログファイルのアクセスログの書式文字列に `%vsid%` を含めて、エラーログ用に `magnus.conf` ファイルで `LogVsId` を有効にする必要があります。次の手順に従って、`magnus.conf` ファイルと、Administration Server の UI でロギングの詳細設定にアクセスし、必要な変更を行います。変更内容は Administration server の再起動後に反映されます。

`LogVsId` を有効にするには、次の手順を実行します。

1. Server Manager にアクセスし、「Preference」タブを選択します。
2. 「Magnus Editor」リンクをクリックします。
3. 「Select a Setting」ドロップダウンリストから「Logging Settings」を選択し、「Manage」をクリックします。
4. 「LogVsId」の値として「On」を選択します。
5. 「OK」ボタンをクリックします。

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. Server Manager にアクセスし、「Logs」タブを選択します。
2. 「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. 「Vsid」チェックボックスをクリックします。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。

`magnus.conf` の `LogVsId` 指令については、『NSAPI プログラマーズガイド』の「エラーログ作成と統計収集」を参照してください。

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、(iPlanet 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、ログの内容を制御するために指定する、パラメータブロックを使用します。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 プログラマーズガイド』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。

ロギングの詳細設定を指定するには、Server Manager の「Log Preferences」ページを使用するか、または `obj.conf` (または `magnus.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 プログラマーズガイド』を参照してください。

Cookie を使用した簡易ロギング

iPlanet 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。構成ファイル `magnus.conf` 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。

ログアナライザの実行

`server-root/extras/log_anly` ディレクトリには、**Server Manager** のユーザ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/flexanlg` 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、**Server Manager** 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、デフォルトサーバの統計情報を生成します。ログアナライザは **iPlanet 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、デフォルトサーバ以外の仮想サーバの統計情報を生成することはできません。ただし、204 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、各仮想サーバの統計情報を参照することはできます。

注 ログアナライザを実行する前に、サーバログをアーカイブする必要があります。サーバログのアーカイブの詳細は、207 ページの「ログファイルの保管」を参照してください。

Server Manager 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. **Server Manager** の「Logs」タブを選択します。
2. 「Generate Report」をクリックします。
3. 各フィールドに必要な事項を入力します。
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、ツール `flexanlg` を実行します。このツールは `server-install/extras/flexanlg` 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)
  h: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 NT)

サーバエラーログへのエラーのロギング (206 ページの「エラーログファイルの参照」を参照)に加えて、iPlanet Web Server は深刻なシステムエラーのログをイベントビューアに記録します。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、基本構成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。iPlanet Web Server でのエラーには、`https-serverid` または `WebServer6.0` 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、お使いのシステムのマニュアルを参照してください。

イベントの表示 (Windows NT)

サーバの監視

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、サーバの監視手法について説明します。

iPlanet 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を監視できます。Windows NT を使用している場合は、SNMP は組み込み済みで、すでに使用可能になっています。

統計機能または SNMP を使用することによって、サーバ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、iPlanet サーバを SNMP 用に構成する必要があります。この章では、Unix または Linux 上で、iPlanet サーバとともに 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の監視
- サービス品質の使用法
- SNMP の基本
- iPlanet 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構成
- 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構成
- 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の監視

統計機能を使用して、サーバの現在の稼働状況を監視できます。統計情報は、サーバ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。個々の仮想サーバに関する統計情報や、サーバインスタンス全体に関する統計情報を表示できます。対話型サーバモニターを通してサーバ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構成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ります。詳細については、

<http://docs.iplanet.com/docs/manuals/enterprise.html> から、オンライン形式の『Performance Tuning, Sizing, and Scaling Guide for iPlanet Web Server』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

対話型サーバモニターで総計をレポートするサーバ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。

注意 統計情報のプロファイリングを使用可能にすると、そのサーバのすべてのユーザ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 プログラマーズガイド』の第3章「定義済み SAF および要求処理プロセス」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. Server Manager 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、サーバインスタンスや仮想サーバ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. Server Manager 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. プル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. プルダウンリストから、表示する統計情報の種類を選択します。
5. 「OK」をクリックします。

サーバ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、選択した種類の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を調整できます。詳細については、<http://docs.iplanet.com/docs/manuals/enterprise.html> で、オンライン形式の『Performance Tuning, Sizing, and Scaling Guide for iPlanet Web Server』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、サーバインスタンスの仮想サーバクラスまたは仮想サーバ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。

Server Manager の「Monitor」タブで、サーバ全体または仮想サーバクラスに関するこれらの設定を有効にできます。ただし、個々の仮想サーバについて、サーバ全体またはクラスレベルの設定を無効にできます。個々のサーバに関するサービス品質の制限設定の詳細については、331 ページの「仮想サーバ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

次の例は、サービス品質の情報を収集および計算する方法を示しています。

サーバの測定時間は 30 秒です。

このサーバは 0 秒で起動します。

1 秒の時点で、HTTP 接続によって、サーバとの間に 5000 バイトのトラフィックが生成されます。

このあと、それ以上の接続は行われません。30 秒の時点で、最後の 30 秒間の合計トラフィックは 5000 バイトになります。

32 秒の時点で、1 秒からのトラフィックが廃棄されます。これは測定時間の 30 秒よりも古いトラフィックであるためです。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。

再計算間隔も同様に動作します。このサーバの再計算間隔は 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きます。

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

サーバインスタンスまたは仮想サーバクラスのサービスの品質を設定するには、ユーザインタフェースを使用して構成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. **Server Manager** 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、サーバインスタンス全体および仮想サーバの各クラスのリストも表示されます。

3. サーバ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、クラス、および仮想サーバ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時間を示す秒数です。デフォルトは 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。

大規模な静的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. サーバインスタンスまたは仮想サーバクラス、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、サーバインスタンスとサーバ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強制するかどうかを選択します。
最大帯域幅を強制する場合、サーバ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。
最大帯域幅を強制しない場合は、最大帯域幅を超えると、サーバ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強制するかどうかを選択します。
最大接続数を強制する場合、サーバ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. 最大接続数を強制しない場合は、最大接続数を超えると、サーバ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 Errorqos-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、この指令が、デフォルトのオブジェクトに設定された最初の AuthTrans である必要があります。サービス品質ハンドラの役割は、仮想サーバ、仮想サーバクラス、およびグローバルサーバの現在の統計情報を調べ、エラーを返して制限値を強制することです。

iPlanet 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、サーバに 503 「Server busy」を返して、NSAPI で処理されるようにします。

iPlanet 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` (`server_root` はサーバルートを示す) にあります。これらのサンプルを使用できますが、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 プログラマーズガイド』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.iplanet.com/docs/manuals/enterprise.html> から、オンライン形式の『Performance Tuning, Sizing, and Scaling Guide for iPlanet Web Server』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、ハンドシェイ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、動的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` が短時間の場合には、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静的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、サーバが動的に再構成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、アクティブでない構成上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、アクティブな構成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がタイムアウトにしないクライアントでは、サーバの動的再構成後、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、仮想サーバの統計情報は仮想サーバクラスおよびグローバルサーバインスタンスとは異なる細分度で計算されます。個々の仮想サーバの接続カウンタは、要求が解析されて仮想サーバ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、仮想サーバクラスおよびグローバルサーバ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。仮想サーバクラスの接続数は、そのクラスのすべての仮想サーバ上の接続の合計数であり、グローバルサーバインスタンスの接続数は、すべての仮想サーバ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、仮想サーバクラスおよびサーバ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管理ステーション (NMS) の間で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、その他のサーバなどです。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、社内のどのサーバが稼働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。iPlanet サーバで 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 とサーバの間で転送されます。

サブエージェントは、サーバに関する情報を収集し、その情報をサーバのマスターエージェントに渡します。Administration Server 以外のすべての iPlanet サーバには、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、各種サブエージェントと NMS の間で情報をやり取りします。マスターエージェントは、Administration Server 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、iPlanet 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、各サーバ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。

iPlanet Web Server の MIB

iPlanet 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 は、Web サーバのネットワーク構成、状態、および統計情報へのアクセスを提供します。SNMP を使用すると、この情報をネットワークマネジメントステーション (NMS) から見ることができます。

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。iPlanet の MIB は、`enterprises (1)` ノードの下に置かれます。

各 iPlanet サーバ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。iPlanet サーバは、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ジメントステーション (NMS) に報告します。NMS では、サーバの 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 iPlanet サーバには、専用の MIB があります。iPlanet の MIB はすべて、次の場所にあります (`server_root` はサーバルートを示す)。

```
server_root/plugins/snmp
```

iPlanet Web Server の MIB は、`iws.mib` という名前のファイルです。この MIB には、iPlanet 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。

iPlanet Web Server 6.0 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 (`server_root` はサーバルートを示す)。

iPlanet Web Server の MIB を使用すると、リアルタイムで Web サーバに関する管理情報を確認し、そのサーバを監視できます。表 10-1 は、`iws.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 10-1 `iws.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 |
|-------------------------------|--------------------------------|
| <code>iwsInstanceTable</code> | iPlanet Web Server インスタンスのテーブル |

表 10-1 iws.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 |
|-------------------------|---|
| iwsInstanceEntry | iPlanet Web Server インスタンスのエントリ |
| iwsInstanceIndex | サーバインスタンスのインデックス |
| iwsInstanceId | サーバインスタンスの識別子 |
| iwsInstanceVersion | バージョンを示す文字列。例: iPlanet-WebServer-Enterprise/ 6.0 BB1-01/24/2001 17:15 (SunOS DOMESTIC) |
| iwsInstanceDescription | サーバインスタンスの説明 |
| iwsInstanceOrganization | サーバインスタンスを担当する組織 |
| iwsInstanceContact | サーバインスタンス担当者の連絡先情報 |
| iwsInstanceLocation | サーバがある場所 |
| iwsInstanceStatus | サーバインスタンスの状態 |
| iwsInstanceUptime | サーバが稼動している時間 |
| iwsInstanceDeathCount | サーバインスタンスのプロセスが機能停止した回数 |
| iwsInstanceRequests | サーバインスタンスが処理した要求数 |
| iwsInstanceInOctets | サーバ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 |
| iwsInstanceCount2xx | サーバインスタンスが発行した 200 番レベル (Successful) の応答数 |
| iwsInstanceCount3xx | サーバインスタンスが発行した 300 番レベル (Redirection) の応答数 |
| iwsInstanceCount4xx | サーバインスタンスが発行した 400 番レベル (Client Error) の応答数 |
| iwsInstanceCount5xx | サーバインスタンスが発行した 500 番レベル (Server Error) の応答数 |

表 10-1 iws.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 |
|-----------------------|---|
| iwsInstanceCountOther | サーバ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 |
| iwsInstanceCount302 | サーバインスタンスが発行した 302 (Moved Temporarily) の応答数 |
| iwsInstanceCount304 | サーバインスタンスが発行した 304 (Not Modified) の応答数 |
| iwsInstanceCount400 | サーバインスタンスが発行した 400 (Bad Request) の応答数 |
| iwsInstanceCount401 | サーバインスタンスが発行した 401 (Unauthorized) の応答数 |
| iwsInstanceCount403 | サーバインスタンスが発行した 403 (Forbidden) の応答数 |
| iwsInstanceCount404 | サーバインスタンスが発行した 404 (Not Found) の応答数 |
| iwsVsTable | iPlanet Web Server 仮想サーバのテーブル |
| iwsVsEntry | iPlanet Web Server 仮想サーバのエントリ |
| iwsVsIndex | 仮想サーバのインデックス |
| iwsVsId | 仮想サーバの識別子 |
| iwsVsRequests | 仮想サーバが処理した要求数 |
| iwsVsInOctets | 仮想サーバが受信したオクテット数 |
| iwsVsOutOctets | 仮想サーバが送信したオクテット数 |
| iwsVsCount2xx | 仮想サーバが発行した 200 番レベル (Successful) の応答数 |
| iwsVsCount3xx | 仮想サーバが発行した 300 番レベル (Redirection) の応答数 |
| iwsVsCount4xx | 仮想サーバが発行した 400 番レベル (Client Error) の応答数 |
| iwsVsCount5xx | 仮想サーバが発行した 500 番レベル (Server Error) の応答数 |
| iwsVsCountOther | 仮想サーバが発行したその他の (2xx、3xx、4xx、5xx のいずれでもない) 応答数 |

表 10-1 iws.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 |
|------------------------------------|---|
| iwsVsCount302 | 仮想サーバが発行した 302 (Moved Temporarily) の応答数 |
| iwsVsCount304 | 仮想サーバが発行した 304 (Not Modified) の応答数 |
| iwsVsCount400 | 仮想サーバが発行した 400 (Bad Request) の応答数 |
| iwsVsCount401 | 仮想サーバが発行した 401 (Unauthorized) の応答数 |
| iwsVsCount403 | 仮想サーバが発行した 403 (Forbidden) の応答数 |
| iwsVsCount404 | 仮想サーバが発行した 404 (Not Found) の応答数 |
| iwsProcessTable | iPlanet Web Server プロセスのテーブル |
| iwsProcessEntry | iPlanet Web Server プロセスのエントリ |
| 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 |
| 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 |
| iwsListenTable | iPlanet Web Server 待機ソケットのテーブル |

表 10-1 iws.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 |
|-------------------------|--|
| iwsListenEntry | iPlanet Web Server 待機ソケットのエントリ |
| 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 | iPlanet Web Server スレッドプールのテーブル |
| iwsThreadPoolEntry | iPlanet Web Server スレッドプールのエントリ |
| iwsThreadPoolIndex | スレッドプールのインデックス |
| iwsThreadPoolEntry | スレッドプールの識別子 |
| iwsThreadPoolEntry | キューに入っている要求数 |
| iwsThreadPoolEntry | これまでに同時にキューに入れた要求の最大数 |
| iwsThreadPoolEntry | 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 |
| iwsVsCount503 | 発行された 503 (Unavailable) の応答数 |
| iwsInstanceCount503 | 発行された 503 (Unavailable) の応答数 |

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 10-2 は、さまざまな条件下での設定手順の概要を示します。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
- すでに稼動している場合、その SNMP エージェントは 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、そのシステムは SMUX をサポートしています)

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 | Administration Server の SNMP の設定を変更したあと、新しいサーバをインストールしたあと、または既存のサーバを削除したあとは、次の手順を実行する必要があります。 <ul style="list-style-type: none"> • Windows NT 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、Administration Server を使用して SNMP マスターエージェントを再起動します。 |
|----------|---|

表 10-2 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

| サーバの条件 | 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 | <ol style="list-style-type: none"> 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のサブエージェントを使用可能にします。 |

表 10-2 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)

| サーバの条件 | 実行手順 (次の節で詳細に説明) |
|--|---|
| <ul style="list-style-type: none"> • 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 | <ol style="list-style-type: none"> 1. Administration Server 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のサブエージェントを使用可能にします。 |
| <ul style="list-style-type: none"> • 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 | <ol style="list-style-type: none"> 1. プロキシ SNMP エージェントをインストールします。 2. プロキシ SNMP エージェントを起動します。 3. 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。 4. マスターエージェントを起動します。 5. システムにインストールされている各サーバのサブエージェントを使用可能にします。 |
| <ul style="list-style-type: none"> • 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 | <ol style="list-style-type: none"> 1. SNMP ネイティブエージェントを再構成します。 2. システムにインストールされている各サーバ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 iPlanet 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、iPlanet 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。233 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。
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。232 ページの「プロキシ SNMP エージェントの起動」および 232 ページの「ネイティブ SNMP デーモンの再起動」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。234 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。240 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構成

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 デーモンの構成を変更する必要があります。

AIX では、いくつかの構成ファイルを使用して通信内容を制限しています。構成ファイルの 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

Server Manager を使用してマスター SNMP エージェントをインストールし、起動するには、サーバが `root` として実行されている必要があります。

Server Manager 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

1. スーパーユーザ (`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、Step 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、デーモンがどの 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. Server Manager 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、239 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ー文字列を入力します。コミュニティー文字列の詳細については、239 ページの「コミュニティー文字列の設定」を参照してください。
8. 「OK」をクリックします。
9. **Server Manager** 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ー文字列を入力します。
11. コミュニティー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構成ファイルに定義されています。**Server Manager** 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た時に、「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構成する
- 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構成
- 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` はサーバルートを示す)。
2. 次の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。手動で起動した場合でも、そのマスターエージェントが実行されていることはユーザインタフェースで検知できます。

SNMP マスターエージェントを手動で構成する

マスター SNMP エージェントを手動で構成するには

1. スーパーユーザ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構成

Server Manager を使用して SNMP サブエージェントを構成するには、次の手順を実行します。

1. Administration Server から、サーバ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、サーバの名前とドメインを入力します。
5. 「Description」フィールドに、オペレーティングシステム情報など、サーバの説明を入力します。
6. 「Organization」フィールドに、そのサーバを担当する組織を入力します。
7. 「Location」フィールドに、サーバの絶対パスを入力します。
8. 「Contact」フィールドに、サーバの担当者名前と連絡先情報を入力します。
9. 「Enable SNMP Statistics Collection」で「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、変更を有効にするためにサーバを再起動します。

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 Administration Server 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ポートマッピングを指定します。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 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。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。

Administration Server を使用して SNMP マスターエージェントを起動する

Administration Server 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. Administration Server にログインします。
2. Server Manager 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構成

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 **Administration Server** を構成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、**Server Manager** 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。**Server Manager** 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 トラップは、SNMP エージェントがネットワークマネジメントステーションに送信するメッセージです。たとえば、SNMP エージェントは、インタフェースの状態が稼働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、**iPlanet 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。**iPlanet 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

Administration Server 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、233 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。Server Manager 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、Server manager の「[SNMP Configuration] ページ」を使用します。次に、「SNMP Subagent Control」ページ (UNIX/Linux) 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該当箇所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 (UNIX/Linux) または Windows NT 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 の構成を変更したあとは、「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。Administration Server 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、サーバからの情報を要求するか、サーバの MIB 内に格納されている変数の値を変更します。次に例を示します。

1. NMS は、Administration Server 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。

サーバ主導の通信：サーバ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。次に例を示します。

1. サブエージェントは、サーバ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。

サーバのパフォーマンスの調整

パフォーマンス向上のためのサーバの調整については、次の場所にある『Performance Tuning, Sizing, and Scaling Guide for iPlanet Web Server』を参照してください。

<http://docs.ipplanet.com/docs/manuals/enterprise.html> (英語)

検索機能の使い方

iPlanet Web Server の検索機能を使用すると、サーバ上のドキュメントの内容と属性を検索できます。サーバ管理者は、ユーザのコミュニティに合わせてカスタマイズしたテキスト検索インタフェースを作成できます。

注 検索機能は、Linux プラットフォームでは使用できません。

この章は、次の節で構成されます。

- 検索について
- テキスト検索の構成
- ドキュメントのインデックス作成
- 検索の実行：基本
- 照会演算子の使用
- 検索インタフェースのカスタマイズ

検索について

サーバドキュメントには、HTML、Microsoft Excel、Adobe PDF、WordPerfect などの各種形式を使用できます。ただし、それぞれのファイル形式で使用できる変換フィルタがあることが前提です。このフィルタにより、サーバは、それらのドキュメントを HTML に変換し、ドキュメントのインデックスを作成し、検索で見つかったドキュメントを Web ブラウザに表示できるようにします。詳細は、255 ページの「コレクションについて」を参照してください。

ユーザは、サーバドキュメントで特定の単語または属性値を検索して、照会と一致するすべてのドキュメントを一覧表示する検索結果セットを取得できます。さらに、結果のリストから1つのドキュメントを選択して、そのドキュメント全体を表示することもできます。このようにして、サーバ上のコンテンツに簡単にアクセスできます。

サーバ管理者は、次の操作を行うことができます。

- テキスト検索を使用する権限が与えられるユーザおよびグループを制限する。
- ユーザおよびグループがアクセスできるドキュメントを決定する。
- テキスト検索の動作を制御する構成ファイルを変更する。
- 検索照会と検索結果のページをカスタマイズする。

サーバで検索機能を使用可能にするには、まず、そのサーバに固有の構成要件を明確にし、いくつかの検索の構成画面を使ってそれらの要件を入力します。次に、検索対象にする1つまたは複数のドキュメントのディレクトリを特定し、そのドキュメント情報のインデックスを作成して、コレクションと呼ばれる検索可能なデータベースに格納します。以下の節で、検索の構成およびコレクションのインデックス作成について詳しく説明します。

テキスト検索の構成

サーバに対する検索機能にはいくつかの項目を設定できます。

- コレクション固有の構成
- すべてのコレクションに適用する構成

コレクション固有の構成では、特定のコレクションに対してドキュメントのインデックスを作成する方法を制御します。この構成は、コレクションを作成する前に定義する必要があります。その他の構成の操作は、検索そのものに影響するだけなので、いつ定義してもかまいません。

コレクション固有の構成の操作は次のとおりです。

- インデックスを作成するドキュメントディレクトリの URL マッピングの定義
- 特定のコレクションの検索で表示するパターンファイルの定義

すべてのコレクションに影響する構成の操作は次のとおりです。

- ファイルおよびディレクトリに対するアクセス制御の確立
- 検索から除外する単語の定義
- 検索パラメータの定義
- 検索機能のオンとオフの設定
- インデックス作成に使用可能なメモリ量の制限

この節では、次の内容について説明します。

- 検索アクセスの制御
- URL のマッピング
- 検索からの単語の除外
- 検索のオンまたはオフ
- 検索パラメータの構成
- 検索パターンファイルの構成
- 手動によるファイルの構成

検索アクセスの制御

検索機能では、サーバのデフォルトの ACL データベースにアクセスします。アクセス制御リスト (ACL) の規則を明示的に定義することにより、または、デフォルトのアクセス制御定義をそのまま利用することにより、サーバ上のドキュメントおよびディレクトリへのアクセスを制御できます。Administration Server の「Users & Groups」機能を使用して、サーバのアクセス制御データベースにユーザを追加できます。

Server Manager の「Restrict Access」インタフェースで、アクセスを許可または制限することができます。アクセス制御の設定の詳細は、第 8 章「サーバへのアクセス制御」を参照してください。

Server Manager の「Search Configuration」インタフェースを使用して、コレクションのルートを検索したり検索結果を表示したりする前に、アクセス権限を確認するようにサーバを構成できます。その方法は、251 ページの「検索パラメータの構成」で説明されています。このオプションを設定すると、サーバは、ユーザに本人確認を求めて、検索照会の結果を返す前にユーザのアクセス権限を確認します。

URL のマッピング

ユーザがコレクションのファイルを検索する場合、検索結果のドキュメントでは部分的に URI (Uniform Resource Identifier) を使用してファイルを識別します。このセキュリティ機能により、ユーザはファイルの完全な物理パス名を知ることができません。URI を設定するには、追加のドキュメントディレクトリに URL をマッピングします。

たとえば、次のファイルパスがあるとします。

```
server_root/Docs/marketing/bizplans/planB.doc
```

plans という URL 接頭辞を定義して、これを下記のディレクトリにマッピングすると、ユーザは最後のディレクトリ以外の部分を見ることができなくなります。

`server_root/Docs/marketing/bizplans`

このように設定しておけば、ユーザは `/plans/planB.doc` と入力するだけでそのファイルを検索できます。詳細は、第 16 章「コンテンツ管理」を参照してください。

注 デフォルトでは、リダイレクトされる URL は常にエスケープされます。これを防ぐには、`escape="no"` を追加します。次に例を示します。

```
NameTrans fn="redirect" from="/foobar"
url-prefix="index.html" escape="no"
```

iPlanet Web Server には、次の 3 つのデフォルトマッピングが用意されています。

- / (スラッシュ) 最初に `server_root/docs` にマッピングす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とも呼ばれる)
- `/help` ほとんどのヘルプファイル用のディレクトリ
- `/search-ui` ほとんどの検索インタフェースファイル用のディレクトリ

コレクションの作成時に、インデックスを作成するドキュメントディレクトリを指定する必要があります。その場合、URL がマッピングされているディレクトリ、または URL がマッピングされているディレクトリ内のサブディレクトリだけを選択できます。特定のディレクトリを定義する、独自のマッピングを作成することができます。

URL をマッピングするには、次の手順を実行します。

1. **Class Manager** を開き、ドロップダウンリストからサーバインスタンスを選択します。
2. 「Content Mgmt」タブを選択します。
3. 「Additional Document Directories」リンクをクリックします。
Web サーバに「Additional Document Directories」ページが表示されます。
4. (省略可能) 次のどちらかを入力して、別のディレクトリを追加します。
 - URL 接頭辞
例: `plans`
 - URL をマッピングするディレクトリの絶対物理パス
例:
`C:/iPlanet/Servers/docs/marketing/plans`
5. 「OK」をクリックします。

6. 「Apply」をクリックします。
7. 次のどちらかを選択して、リストされている現在の追加ディレクトリのうちの1つを編集します。
 - 「Edit」
 - 「Remove」
8. 編集する場合は、変更しようとしている、リストされているディレクトリの隣にある「Edit」を選択します。
9. 新しい接頭辞を ASCII 形式で入力します。
10. (省略可能) ディレクトリにスタイルを適用する場合は、「Apply Style」ドロップダウンリストからスタイルを選択します。

スタイルの詳細は、第 17 章「構成スタイルの適用」を参照してください。
11. 「OK」をクリックして、新しいドキュメントディレクトリを追加します。
12. 「Apply」をクリックします。
13. 「Apply Changes」を選択し、サーバのハードスタートまたはハードリスタートを実行します。

注 追加のドキュメントディレクトリに基づいてコレクションを作成した後は、URL マッピングを変更してはなりません。変更すると、そのコレクションのエントリの URL マッピングが間違った物理ファイルの場所を指すこととなります。

検索からの単語の除外

検索エンジンでインデックスの対象としない、または検索対象にしない単語を指定できます。通常、これらの単語は「ストップワード」または「検索から除外する単語」と呼ばれ、*at*、*and*、*be*、*for*、*the* などの冠詞、接続詞、および前置詞がこれに含まれます。

ストップワードを指定するには、`style.stp` という名前のファイルを編集します。このファイルは、`server_root\plugins\search\common\style` ディレクトリにある、各コレクションタイプの `html`、`pdf`、`mail`、および `news` の各サブディレクトリにあります。それぞれの `style.stp` ファイルは、そのコレクションタイプのストップワードを制御します。たとえば、`server_root\plugins\search\common\style\html` にある `style.stp` ファイルは、そのコレクションの HTML ファイルのストップワードを制御します。

ストップワードは、1行に1語ずつ、左揃えで `style.stp` に追加します。文字クラスを示す角括弧 ([])、任意の文字を示すピリオド (.), 繰り返しを示すプラス記号 (+) などの演算子を使用できます。たとえば、`style.stp` ファイルに次の行が記述されているとします。

```
.....+
at
and
be
[0-9a-zA-Z]
[0-9] [0-9] [0-9] [0-9]+
```

この例では、最初のピリオドの行(デフォルトでファイルに記述されている)は、40文字以上の語はインデックスに含めないことを示します。さらに、インデックスに含めない単語として *at*、*and*、*be* を指定しています。`[0-9a-zA-Z]` は、1文字の単語は一切インデックスに含めないことを示します。`[0-9] [0-9] [0-9] [0-9]+` は、4桁以上の整数は一切インデックスに含めないことを示します。

指定する語は、大文字と小文字が区別されます。したがって、単語の大文字と小文字によるすべてのバリエーションを入力する必要があります。たとえば、*the* の場合は、*the*、*THE*、および *The* を入力する必要があります。

ストップワードを使用する場合は、コレクションを作成する前に、`style.stp` ファイルを作成する必要があります。コレクションの作成後に `style.stp` ファイルを変更する場合は、次の手順が必要になります。

1. 現在のコレクションを削除します。
2. そのコレクションタイプのストップワードのリストを変更します。
3. そのコレクションを再度作成します。
4. そのコレクション内の全ドキュメントのインデックスを再度作成します。

検索のオンまたはオフ

ユーザがサーバまたは Web サイトを検索できるようにするためには、検索をオンに設定する必要があります。デフォルトの設定では、検索はオフになっています。検索機能をオンまたはオフにするには、*Server Manager* の「*Search State*」インタフェースを使用します。

ユーザが検索を使用しないサーバでは、検索をオフにすると、サーバのパフォーマンスを高めることができます。また、サーバのトラフィックが重くなる場合に検索機能をオフにし、トラフィックが軽くなれば再びオンにすることもできます。検索をオフにしていると、HTTP サーバの起動時に検索プラグインは読み込まれません。

検索パラメータの構成

サーバ管理者は、検索結果を取得するときにユーザに表示される内容を制御するデフォルトのパラメータを設定できます。

検索パラメータを構成するには、次の手順を実行します。

1. Server Manager にアクセスして、「Search」を選択します。
2. 「Search Configuration」リンクをクリックします。
Web サーバに「Search Configuration」ページが表示されます。
3. 「Default Result Set Size」フィールドに、ユーザに一度に表示される検索結果項目のデフォルトの最大数を入力します。
この数は、手順 4 で設定する「Largest Possible Result Set Size」の値よりも大きくしてはなりません。デフォルトは 20 です。
4. 「Largest Possible Result Set Size」フィールドに、結果セットの最大項目数を入力します。
デフォルトは 5000 です。値として 250 を入力すると、検索条件に一致するドキュメントが 1000 見つかったとしても、ユーザがアクセスできるのは、最初の 250 ドキュメント、つまり上位 250 にランク付けされたドキュメントだけになります。
5. 「Date/Time」の文字列を Posix 書式で入力します。
このエントリでは、ユーザに表示される日付および時刻の書式を定義します。リストされている記号を使用します。
6. HTML ドキュメントにタイトルタグが設定されていない場合に使用される、デフォルトの HTML タイトルを入力します。
一般的な HTML のデフォルトは、「(Untitled)」であり、これが HTML ファイルの検索結果ページに表示されます。
7. そのコレクションへのアクセスを制限したい場合は、「Check access permissions on collection root before doing a search ?」に「Yes」を選択します。
8. 検索結果内の各ドキュメントに対してアクセス制御チェックを実行したい場合は、「Check access permissions on search results ?」に「Yes」を選択します。
9. 「OK」をクリックして、新しい検索構成を設定します。
10. 「Apply」をクリックします。
11. 「Apply Changes」を選択し、サーバのハードスタートまたはハードリスタートを実行します。

表 12-1 Posix による共通の日付と時刻の書式

| 書式 | 表示結果 (例) |
|----|-----------------------------|
| %a | 省略形の曜日 (Wed など) |
| %A | 完全形の曜日 (Wednesday など) |
| %b | 省略形の月 (Oct など) |
| %B | 完全形の月 (October など) |
| %c | 現在のロケールの書式による日付と時刻 |
| %d | 十進数による、月内の日 (01 ~ 31) |
| %H | 十進数、24 時間の書式による時間 (00 ~ 23) |
| %m | 十進数による月 (01 ~ 12) |
| %M | 十進数による分 (00 ~ 59) |
| %x | 日付 (09/18/01 など) |
| %X | 時刻 (09:46:40 など) |
| %y | 西暦の下 2 桁による、年 (00 ~ 99) |
| %Y | 完全な西暦による、年 (1999 など) |

検索パターンファイルの構成

パターンファイルは、テキスト検索インタフェースのレイアウトを定義する HTML ファイルです。検索機能や、検索インタフェースの特定の部分を作成するパターン変数セットにパターンファイルに関連付けることができます。パターンファイルでは、テキスト検索インタフェースの外観、操作性、および機能を定義します。パターンファイルでパターン変数を使用することにより、背景色、ヘルプテキスト、バナーなどをカスタマイズできます。パターン変数の値は、その変数が表す実際のテキストやグラフィックスが格納されているファイルのパス名の場合もあれば、テキストや HTML の場合もあります。

デフォルトのパターンファイルを使用することも、独自にカスタマイズしたファイルセットを作成することもできます。単一コレクションまたは複数コレクションの検索で開始パターンファイルと終了パターンファイルが指定されていない場合は、デフォルトの開始パターンファイルと終了パターンファイルが使用されます。ユーザインタフェースの変更方法の詳細は、281 ページの「検索インタフェースのカスタマイズ」を参照してください。

検索の実行時に特定の検索要求に関連付けられたデフォルトのパターンファイルを探す場所を定義するには、そのパターンファイルのパスを指定する必要があります。

パターンファイルを構成するには、次の手順を実行します。

1. Server Manager にアクセスして、「Search」を選択します。
2. 「Search Pattern Files」リンクをクリックします。
Web サーバに「Search Pattern Files」ページが表示されます。
3. パターンファイルが格納されているディレクトリの絶対パスを入力します。
開始 (ヘッダー)、終了 (フッター)、および照会ページのデフォルトのパターンファイルがこのディレクトリに格納されます。
4. 「Default Start Pattern File」に相対パス名を入力します。
このエントリは、コレクションに定義されたヘッダーファイルがない場合、または複数のコレクションを検索する場合の、検索結果のトップページを定義します。
5. 「Default End Pattern File」に相対パス名を入力します。
このエントリは、コレクションに定義されたフッターファイルがない場合、または複数のコレクションを検索する場合の、検索結果ページのフッターを定義します。
6. 「Pattern File for Query Page」に相対パス名を入力します。
このエントリは、検索機能の起動時に表示される検索照会ページを定義します。
7. 「OK」をクリックして、検索パターンファイルを構成します。
8. 「Apply」をクリックします。
9. 「Apply Changes」を選択し、サーバのハードスタートまたはハードリスタートを実行します。

手動によるファイルの構成

検索機能では、いくつかの構成ファイルを調べることにより、サーバの検索がどのように構成されているか判別します。それらの構成ファイルには、システム設定、ユーザが定義した変数、および検索コレクションに関する情報が定義されています。通常、この情報は iPlanet Web Server の「Search」ページから変更しますが、テキストエディタを使用して手動でこれらのファイルを変更することもできます。ユーザインタフェースをカスタマイズするための構成ファイルの変更については、関連情報が 281 ページの「検索インタフェースのカスタマイズ」で説明されています。

注 手動による構成ファイルの変更は、お勧めすることはできません。手動で変更を加える場合は、変更内容を有効にするには、サーバを再起動する必要があります。

構成ファイル

検索を制御する構成ファイルは、次のリストで記述されます。

- `userdefs.ini` - このユーザ定義ファイルでは、ユーザが定義したパターン変数を定義します。これは、使用する言語 (英語、ドイツ語、日本語など) の `userdefs.ini` ファイルにマッピングされます。

独自のパターン変数を作成して `userdefs.ini` ファイルに定義することにより、すべてのパターンファイルの検索インタフェースをカスタマイズできます。詳細は、287 ページの「ユーザ定義のパターン変数」を参照してください。

- `dblist.ini` - このコレクション内容ファイルでは、コレクション固有の情報を記述します。コレクションを作成して管理すると、そのコレクションに関する情報で `dblist.ini` ファイルが更新されます。

属性の最大数の調整

コレクションには、ファイル形式によって異なるデフォルト属性セットがあります。たとえば、HTML ファイルには、Title 属性と SourceType 属性があります。HTML ファイルに META タグ付きの HTML 属性を定義することもできます。PDF のように、非常に多くのデフォルト属性があるファイル形式もあります。各ファイル形式の属性の詳細は、257 ページの「コレクションの属性について」を参照してください。

インデックス作成に使用するメモリの制限

インデックス作成操作に使用できる RAM の容量に制限を設けることができます。これを行うには、`[NS-loader]` ファイルを手動で編集して、最大メモリ量を定義する行を追加します。次に例を示します。

```
NS-max-memory = 32000000
```

サーバのデフォルトでは、使用可能なシステムメモリをすべて使用するように設定されています。通常、次の条件に該当する場合は、インデックス作成に使用する RAM を制限する必要があります。

- RAM が推奨最小値よりも小さいマシンにサーバをインストールしている場合。
- Windows NT サーバのサーバ管理者が、大量のインデックス作成を必要としながらも、ほかのサーバ操作にメモリを必要とする場合。

インデックスファイルのサイズ制限

インデックスファイルに使用できるディスク容量を制限できます。これを行うには、`[NS-loader]` ファイルを手動で編集して、インデックスファイルの最大サイズを定義します。次に例を示します。

```
NS-max-idx-file-size = 1500000
```

通常、インデックス作成操作には、1ファイルにつき約1.5Mバイトが必要です。一時ファイルと合わせて2ファイルあるので、インデックス作成には3Mバイトのディスク容量が必要になります。ファイルサイズを1ファイル当たり1.5Mバイトに設定することで、各ファイルのサイズの上限が設定されます。

ドキュメントのインデックス作成

ユーザが検索を実行するには、検索可能なデータのデータベースが必要です。ドキュメントの内容やファイルプロパティなど、ドキュメントに関する情報のインデックスを作成して保存する、コレクションと呼ばれるデータベースを作成する必要があります。

検索には、検索対象となるファイルのコレクションが必要です。ドキュメントのインデックスを作成すると、ドキュメントの内容や、タイトル、作成日、作成者などのファイルプロパティを検索に使用できます。

コレクションにドキュメントを追加したり、コレクションからドキュメントを削除することができます。これにより、必要に応じて、コレクションの最適化、更新、および管理を行うことができます。

この節では、次の内容について説明します。

- コレクションについて
- コレクションの属性について
- 新規コレクションの作成
- コレクションの構成
- コレクションの更新
- コレクションの保守
- 定期保守のスケジュール設定
- コレクションの保守スケジュールの削除

コレクションについて

サーバ管理者がサーバのドキュメントの全部または一部についてインデックスを作成すると、それらのドキュメントに関する情報が1つのコレクション内に格納されます。コレクションには、次のような情報が格納されます。

- ドキュメントの形式
- ドキュメントの言語

- 検索可能な属性
- コレクション内のドキュメントの数
- コレクションのステータス
- コレクションに関する簡単な説明

詳細は、273 ページの「コレクションのコンテンツの表示」を参照してください。

コレクションの作成時に、次のように、そのコレクションに収めるファイルのタイプを定義する必要があります。

- HTML
- ASCII
- NEWS
- EMAIL
- PDF

インデックスの作成時に、この定義によって、インデックスに含める属性や、何らかのファイル変換が必要かどうかを判別されます。

1つのディレクトリ内にあるすべてのファイルについてインデックスを作成することも、HTML や PDF などの特定の拡張子を持つファイルについてのみインデックスを作成することもできます。

コレクションには、インデックスを作成した各ドキュメントに関する情報を収めたレコードがあります。コレクションからドキュメントを削除すると、そのドキュメントのコレクションのエントリだけが削除されます。元のドキュメントは削除されません。

複数のサービンスタンスがある場合、1つのコレクションは、そのコレクションが作成されたサービンスタンスにだけ関連付けられます。ユーザは、そのサービンスタンス内のコレクションだけを検索できます。

コレクションの属性について

表 12-2 に示すように、特定のファイル形式には、そのタイプのファイルについてインデックスを作成する、デフォルトの属性セットがあります。

表 12-2 各ファイル形式についてインデックスが作成される、デフォルトの属性

| ファイル書式 | 属性 | タイプ | 説明 |
|-------------|----------------------|------------|---|
| ASCII | (なし) | - | - |
| HTML | Title | テキスト | ユーザが定義したファイルのタイトル |
| | SourceType | テキスト | ドキュメントのオリジナルの形式 |
| NEWS | From | テキスト | ニュースアイテムのソース userID |
| | Subject | テキスト | ニュースアイテムのサブジェクトフィールドのテキスト |
| | Keywords | テキスト | ニュースアイテムに定義されるキーワード |
| | Date | 日付 | ニュースアイテムが作成された日付 |
| EMAIL | From | テキスト | 電子メールのソース userID |
| | To | テキスト | 電子メールの宛先の userID |
| | Subject | テキスト | 電子メールのサブジェクトフィールドのテキスト |
| | Date | 日付 | 電子メールが作成された日付 |
| PDF | InstanceID | テキスト | 内部 ID 番号 |
| | PermanentID | テキスト | 内部 ID 番号 |
| | NumPages | 整数 | ドキュメントのページの総数 |
| | DirID | テキスト | PDF ファイルが格納されているディレクトリ |
| | FTS_ModificationDate | 日付 | ドキュメントの最終変更日 |
| | FTS_CreationDate | 日付 | ドキュメントの作成日 |
| | WXEVersion | 整数 | PDF ドキュメントからのテキスト抽出に使用する Adobe Word Finder のバージョン |
| | FileName | テキスト | Adobe の仕様によるファイル名 |
| | FTS_Title | テキスト | ドキュメントのタイトル |
| | FTS_Subject | テキスト | ドキュメントの主題 |
| | FTS_Author | テキスト | ドキュメントの著者 |
| FTS_Creator | テキスト | ドキュメントの作成者 | |

表 12-2 各ファイル形式についてインデックスが作成される、デフォルトの属性 (続き)

| | | |
|--------------|------|--------------------------|
| FTS_Producer | テキスト | ドキュメントのプロデューサ |
| FTS_Keywords | テキスト | ドキュメントのキーワード |
| PageMap | テキスト | ページのワードインスタンスを記述したページマップ |

デフォルトの HTML コレクションでは、Title 属性と SourceType 属性を保持していますが、HTML の <META> タグを使用することにより、ファイル属性を 30 個まで指定して検索およびソートができるようにインデックスを作成できます。254 ページの「属性の最大数の調整」で説明されているように、ファイル属性の最大設定数は変更可能です。

たとえば、次のような HTML コードの行をドキュメントに記述できます。

```
<META NAME="Writer" CONTENT="R. Hunter">
<META NAME="Song" CONTENT="Stella Blue">
```

これらの META タグを抽出してドキュメントのインデックスが作成されている場合、writer (執筆者) や product (製品) のフィールド内の特定の値を検索できます。たとえば、Writer <contains> Hunter or Song <contains> Blue のような照会によって検索を実行できます。

注 META タグフィールド内の属性値に指定できるのは、テキスト文字列のみです。このため日付や時刻などのすべての数値は、テキストとしてソートされます。META タグ属性内の無効な HTML 文字は、ハイフンで置き換えられます。

新規コレクションの作成

サーバに保持できるコレクションは 12 個までです。13 番目のコレクションを使用するには、先に「Search」の「Maintain Collection」を使用して既存のコレクションを削除する必要があります。

コレクション (複数) に収めることができるエント리는、最大で 1,600 万ドキュメントです。1 つのドキュメントのインデックスが複数のコレクションに含まれている場合は、複数のドキュメントとしてカウントされます。10,000 ドキュメント以上の新規コレクションは、低トラフィック時に作成することをお勧めします。高トラフィック時に作成すると、インデックス作成操作がシステムのパフォーマンスに影響する可能性があります。

1つのディレクトリ内のすべてのファイルまたは一部のファイルの内容のインデックスを収めるコレクションを作成することができます。1種類のファイルだけを収めるコレクションを定義できます。また、さまざまな形式のドキュメントのコレクションを作成して、インデックス作成時に自動的にHTMLに変換することもできます。複数の形式によるコレクションを定義して「auto-convert」（自動変換）オプションを設定すると、インデックス作成時に、まずドキュメントがHTMLに変換され、次にその内容のインデックスが作成されます。変換後のHTMLドキュメントは、サーバの検索コレクションフォルダ内のhtml_docディレクトリに置かれます。

選択するファイル形式によって、コレクションで使用されるデフォルトの属性、およびインデックス作成時に内容の自動HTML変換を必要とするかどうかが決まります。各ファイル形式の属性については、257ページの「コレクションの属性について」を参照してください。

選択したファイルタイプに関係なく、ファイルの内容は常にインデックス作成されます。ファイルタイプとしてHTMLを選択した場合は、サーバによりHTMLのデフォルト属性によるコレクションが作成され、HTML以外のファイルのインデックスを作成しようとしてもHTMLへの変換は行われません。HTMLファイルをASCIIコレクションへインデックス作成して収める場合、HTMLのマークアップタグもファイルの内容の一部としてインデックスが作成され、その内容は生テキストとして表示されません。

| | |
|---|--|
| 注 | 1つのコレクションを作成するには、システム上に3Mバイト以上の利用可能なディスク容量が必要です。インデックスファイルのサイズを制限する方法については、254ページの「インデックスファイルのサイズ制限」を参照してください。 |
|---|--|

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. **Server Manager** にアクセスし、コレクションを作成するサーバをドロップダウンリストから選択します。
2. 「Search」タブを選択します。
3. 「New Collection」リンクをクリックします。

Webサーバに「Create a Collection」ページが表示されます。

次のいずれかを選択します。

- 「Directory to Index」フィールドの現在のドキュメントディレクトリ
- サーバのドロップダウンリストに定義されている別のドキュメントディレクトリ
- ファイルとサブディレクトリの一覧を表示する場合は「View」

追加のドキュメントディレクトリの詳細は、247 ページの「URL のマッピング」を参照してください。

4. 「Documents Matching」フィールドのデフォルトの *.html をそのまま使用するか、独自のワイルドカード式を定義します。

1 つの式に複数のワイルドカードを定義できます。次に例を示します。

```
(* .htm|*.html or *(.htm|.html)
```

ワイルドカードパターンの構文の詳細は、280 ページの「ワイルドカードの使用」を参照してください。

注 名前にセミコロン (;) が含まれているファイルのインデックスは作成できません。これらのファイル名は、変更してください。

5. 指定されたディレクトリ内のサブディレクトリをインデックスに含める場合は、「Yes」を選択します。

6. 「Collection Name」フィールドにコレクション名を入力します。

コレクション名は、コレクションの保守に使用されます。これは、ファイルの物理ファイル名なので、使用するオペレーティングシステムの名前付け規則に従ったファイル名にしてください。最大で 128 文字まで使用できます。空白文字は下線に変換されます。

注 コレクション名では、アクセント記号付きの文字は使用しないでください。アクセント記号付きの文字を使用する必要がある場合は、コレクション名からはアクセント記号を削除し、ラベルでは使用してください。このラベルは、検索インタフェースでユーザに表示されます。

7. (省略可能) 「Collection Label」フィールドに、コレクションのユーザが定義した名前を入力します。

この名前は、ユーザがテキスト検索を実行するときに表示されます。できるだけ、内容がわかりやすい適切なラベル名にします。最大で 128 文字の、一重引用符または二重引用符を除くすべての文字を使用できます。

8. (省略可能) 「Description」フィールドに、コレクションの説明を入力します (最大 1024 文字)。

この説明は、コレクションの内容のページに表示されます。

9. コレクションに収めるファイルのタイプを選択します。

- ASCII
- HTML

- News
 - Email
 - PDF
10. インデックス作成時に HTML ファイルから META タグ属性を抽出するかどうかを選択します。
- このオプションは、HTML コレクションに対してのみ選択してください。META タグ属性を抽出すると、その値を検索できます。1つのドキュメントで、最大 30 種類のユーザが定義した META タグのインデックスを作成できます。
11. ドロップダウンリストからコレクションの言語を選択します。
- デフォルトは英語 (「English (ISO-8859-1)」) です。文字セットの詳細は、第 16 章「コンテンツ管理」を参照してください。
12. 「OK」をクリックして新規コレクションを作成します。

注 一度コレクションのインデックス作成を開始すると、インデックス作成が完了するか、システムを再起動しない限り、そのプロセスを停止することはできません。サーバをシャットダウンしても、プロセスは終了しません。

13. 「Apply」をクリックします。
14. 「Apply Changes」を選択し、サーバのハードスタートまたはハードリスタートを実行します。

コレクションの構成

コレクションを作成した後で、そのコレクションの初期設定の一部を変更できます。それらの設定は、コレクション情報ファイルの `dblist.ini` にあります。コレクションを再構成すると、`dblist.ini` ファイルが更新され、変更内容が反映されます。構成ファイルの詳細は、253 ページの「手動によるファイルの構成」を参照してください。次の設定変更が可能です。

- 説明を修正する
- ラベルを変更する
- ドキュメントに別の URL を定義する
- 表示されるドキュメントの強調表示方法を定義する
- 使用するパターンファイルを定義する
- 日付の書式を定義する

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。

コレクションを構成するには、次の手順を実行します。

1. **Server Manager** にアクセスし、目的のコレクションが存在するサーバ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. 「Search」タブを選択します。
3. 「Configure Collection」リンクをクリックします。
Web サーバに「Configure Collection」ページが表示されます。
4. 構成するコレクションを選択します。
5. 次の項目を入力または変更できます。
 - オプションの「Description」フィールドの説明 (1024 文字まで)
 - 「Label」フィールドのユーザが定義した名
 - 「URL for Documents」フィールドの URL (URL を変更した場合)
たとえば、`publisher/help` の URL マッピングを、よりシンプルな `/helpFiles` に変更するなどです。
 - 表示するドキュメント内で検索照会の単語またはフレーズを強調表示するときにサーバが使用する、「Highlight begin」フィールドと「Highlight end」フィールド内の HTML タグ。
デフォルトは、`` タグと `` タグによる太字での表示ですが、さらにタグを追加したり、別のタグに変更することができます。たとえば、`<blink>` と、それに対応する `</blink>` を追加すると、点滅する赤い太字で強調表示できます。
6. 入力日付の書式を選択します。
7. 次の項目の検索結果の表示に使用するデフォルトのパターンファイルを定義または変更します。
 - ヘッダー
 - フッター
 - レコード
8. 「Result Pattern File」フィールドで、検索結果のリストから単一のドキュメントを強調表示する際に使用するパターンファイルの名前を入力または変更します。
9. 「OK」をクリックして、コレクションの構成を変更します。
10. サーバでのコレクションの構成が完了したら、「Apply」をクリックします。
11. 「Apply Changes」を選択し、サーバのハードスタートまたはハードリスタートを実行します。

コレクションの更新

コレクションの作成後に、ファイルの追加または削除が必要になる場合があります。ドキュメントを追加すると、必要な場合はそのファイルの内容のインデックスが作成され、変換されます。ドキュメントを削除する場合は、そのファイルのエントリがメタデータと一緒にコレクションから削除されます。元のドキュメントには影響はなく、コレクションのそのエントリだけが削除されます。

注 コレクションの作成時に「Extract Metatags」オプションを選択した場合は、新しいドキュメントを追加する度に、META タグ HTML 属性のインデックスが作成されます。

コレクションを更新するには、次の手順を実行します。

1. **Server Manager** にアクセスし、目的のコレクションが存在するサービンスタンسをドロップダウンリストから選択します。
2. 「Search」タブを選択します。
3. 「Update Collection」リンクをクリックします。

Web サーバに「Update Collection」ページが表示されます。

4. 更新するコレクションを選択します。

現在選択されているコレクションにインデックスエントリがあるドキュメントのリストが表示されます。各リストには 100 レコードずつ表示されます。100 個以上のファイルがあるコレクションの場合は、「Prev」ボタンおよび「Next」ボタンを使用して前後のリストを表示できます。

5. 「Document Matching」フィールドで、単一のファイル名を入力するか、ワイルドカードを使用して、そのコレクションで追加または削除するファイルのタイプを指定します。

*.html などのようにワイルドカードを入力すると、この拡張子を持つファイルだけを更新できます。サブディレクトリ内のファイルを指定する場合は、ファイルのリストに表示されているパス名を入力します。次に例を示します。

frenchDocs/*.html

注意 ワイルドカード式の入力には注意が必要です。index.html と入力すると、現在のコレクションのインデックスファイルを追加または削除できますが、*/index.html と入力すると、すべての index.html ファイルがそのコレクション内に追加または削除されます。

6. サブディレクトリを含めるかどうかを選択します。
7. 次のどちらかをクリックします。

- 指定されたファイルおよびサブディレクトリを追加する場合は「AddDocs」
 - 指定されたファイルを削除する場合は「RemoveDocs」
8. 「Apply」をクリックします。
 9. 「Apply Changes」を選択し、サーバのハードスタートまたはハードリスタートを実行します。

コレクションの保守

時には、コレクションを保守したい場合もあります。頻繁にコレクションのインデックス作成や更新を行わない限り、通常の使用法では、これらの保守作業は必要ないこともあります。

次のコレクション保守作業を実行できます。

- **コレクションの最適化** - コレクションでドキュメントやディレクトリの追加、削除、または更新を頻繁に行ってコレクションを最適化することにより、パフォーマンスを高めることができます。これは、ハードドライブのデフラグ処理と似ています。最適化は自動的に実行されません。したがって、コレクションのインデックスを再作成または更新した後、手動で最適化する必要があります。コレクションを別のサイトに公開する前や、読み取り専用の CD-ROM に書き込む前に、そのコレクションを最適化した方がよい場合があります。
- **インデックスの再作成** - コレクションのインデックスを再作成できます。コレクションにすでにエントリを持つファイルを探して、その属性および内容のインデックスを再作成することができます。コレクションで最初にファイルのインデックスを作成したときに META タグ属性を抽出するオプションを選択していた場合は、META タグ属性が抽出されます。この場合、*.html などの、元の条件でコレクションを作成し直すのではなく、元の条件に該当する新しいドキュメントを追加します。ソースドキュメントが削除されていて見つからない場合、コレクションのエントリは削除されます。
- **削除** - コレクションを削除できます。コレクションだけが削除され、元のソースドキュメントは削除されません。

注 コレクションの削除には、ローカルのファイルマネージャを使用しないでください。サーバを再起動する前に検索を実行しようとする、検索は失敗します。

これらのコレクション管理作業を実行するには、Server Manager の「Maintain Collection」リンクを使用します。

定期保守のスケジュール設定

定期的なコレクションの保守スケジュールを設定できます。また、最適化とインデックスの再作成の保守スケジュールを個別に設定できます。頻繁にコレクションのインデックス作成や更新を行わない限り、通常の使用法では、これらの保守作業は必要ないこともあります。たとえば、毎日新しいドキュメントが追加されるような非常に変動の多い Web サイトでは、頻繁にインデックスの再作成を行う必要があります。

通常、次の作業の組み合わせにより、定期スケジュールを設定します。

- インデックスの再作成と更新の操作によって削除されたエントリを一掃する。
- コレクションの条件に一致する新しいドキュメントのエントリを追加する。
- コレクションの新しいインデックス作成条件を入力してコレクションを更新する。

コレクションの最適化、インデックス再作成、または更新を行うには、次の手順を実行します。

1. Server Manager から、「Search」を選択します。

2. 「Schedule Collection Maintenance」リンクをクリックします。

Web サーバに「Schedule Collection Maintenance」ウィンドウが表示されます。

3. ドロップダウンリストからコレクションを選択します。

このリストには、作成したすべてのコレクションが表示されます。

4. ドロップダウンリストから、次のいずれかのアクションを選択します。

- 「Reindex」(インデックスの再作成)
- 「Optimize」(最適化)
- 「Update」(更新)

同じコレクションに対して、異なるアクションによる個別のスケジュールを設定できます。コレクションの更新を選択すると、2つの追加フィールドが表示されます。1つは、ドキュメントの一致条件を入力するためのフィールド、もう1つは、条件に一致するドキュメントがサブディレクトリで見つかった場合に、そのドキュメントも含めるためのフィールドです。

5. 「Schedule Time」フィールドに、定期保守を実行する時刻を入力します。

書式(HH:MM)を使用して入力します。HHは24より小さい数、MMは60より小さい数である必要があります。時刻は必ず入力します。

6. 「Schedule Day(s) of the Week」のセクションで、1つまたは複数の曜日にチェックマークを付けます。

すべての曜日を選択することもできます。少なくとも1つの曜日を選択する必要があります。

7. 「OK」をクリックして保守スケジュールを設定します。

UNIX または Linux では、新しい保守スケジュールの設定を有効にするには、Administration Server から ns-cron プロセスを再起動する必要があります。

ns-cron プロセスを再起動するには、次の手順を実行します。

1. Administration Server から、「Global Settings」を選択します。
2. 「Cron Control」リンクをクリックします。
3. ns-cron がすでにオンになっている場合は、「Restart」をクリックして再起動します。ns-cron がオンになっていない場合は、「Start」をクリックして起動します。
どちらの場合も、これで、定期保守のスケジュールが自動的に実行されます。

コレクションの保守スケジュールの削除

コレクションの定期保守が不要になった場合は削除できます。

コレクションの保守スケジュールの設定を解除するには、次の手順を実行します。

1. Server Manager から、「Search」を選択します。
2. 「Remove Scheduled Collection Maintenance」リンクをクリックします。
Web サーバに「Remove Scheduled Collection Maintenance」ウィンドウが表示されます。
3. 「Choose Collection」のドロップダウンリストからコレクションを選択します。
このリストには、定期保守を設定しているすべてのコレクションが表示されます。
4. ドロップダウンリストから、「Reindex」、「Optimize」、「Update」のいずれかのアクションを選択します。
フレームの下方に、現在保守スケジュールが実行されるように設定されている時刻と曜日が表示されます。
5. 「OK」をクリックしてその保守スケジュールを削除します。

UNIX または Linux では、保守スケジュールの削除を有効にするには、ns-cron プロセスを再起動する必要があります。

ns-cron プロセスを再起動するには、次の手順を実行します。

1. Administration Server から、「Global Settings」を選択します。
2. 「Cron Control」リンクをクリックします。
3. ns-cron がすでにオンになっている場合は、「Restart」をクリックして再起動します。ns-cron がオンになっていない場合は、「Start」をクリックして起動します。

どちらの場合も、これで、定期保守スケジュールは実行されなくなります。

検索の実行：基本

ユーザ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。iPlanet Web Server をインストールすると、検索照会と検索結果のデフォルトのフォームセットもインストールされます。これらのフォームにより、ユーザにシンプルな検索方法を提供できます。

テキスト検索は、次の4つの部分で構成されています。

- **照会の作成** - ユーザが検索条件を入力します。
- **検索結果の表示** - 条件に一致するドキュメントのリストがサーバにより表示されます。
- **ドキュメントの表示** - ユーザは、検索結果リストから、強調表示されたドキュメントを表示できます。
- **コレクションのコンテンツの表示** - ユーザは、各コレクションで維持されている情報を表示できます。

注 検索機能がオフになっている場合、これらの照会フォームは使用できません。

この節では、次の内容について説明します。

- 検索ホームページ
- 検索照会
- ガイド付き検索 (Guided Search)
- 拡張検索 (Advanced HTML Search)
- 検索結果
- コレクションのコンテンツの表示

検索ホームページ

検索ホームページ (http://server_root:port/search を参照) には、3つの検索照会インタフェースと、インタフェースのカスタマイズに関するオンライン形式の QuickStart チュートリアルへのリンクが用意されています。このチュートリアルには、各種のパターンファイルの説明、およびパターンファイルを変更して別の検索結果を出す例が示されています。

検索照会

iPlanet Web Server をデフォルトでインストールすると、標準 HTML 照会 (Standard HTML Search)、拡張 HTML 照会 (Advanced HTML Search)、および Java ベースのガイド付き照会 (Guided Search) の3つの検索照会ページがインストールされます。

標準の検索照会では、検索対象のコレクションを選択し、照会言語演算子を使用して、検索する語または句を入力します。

Java ベースのガイド付き検索インタフェースでは、さまざまなドロップダウンリストを使用することで、照会を簡単に作成できます。この機能を利用するには、ブラウザで Java を有効にしておく必要があります。

拡張 HTML 照会ページでは、検索対象に複数のコレクションを選択する、検索結果のソート順序を設定する、1 ページに表示するドキュメント数を定義するなどの追加オプションを使用できます。通常、「前へ (Prev)」と「次へ (Next)」の矢印をクリックすると、検索結果のページ間を移動できます。

標準検索を実行するには、次の手順を実行します。

1. Web ブラウザの location (アドレス指定) フィールドに次の URL を入力します。
`http://server_root:port/search`
2. 表示された検索照会ページで、「検索対象 (Search In)」フィールドのドロップダウンリストから、検索するコレクションを選択します。
3. 「検索文字列 (For)」フィールドに、検索する語句を入力します。演算子を組み合わせることによって、複雑な照会を作成することもできます。検索演算子の詳細は、274 ページの「照会演算子の使用」を参照してください。
4. 「検索 (Search)」ボタンをクリックして、照会を実行します。

ガイド付き検索 (Guided Search)

Java ベースのガイド付き検索インタフェースを使用することもできます。この機能は、ユーザが照会を作成するのをガイドするものです。これは、いくつかのパーツを含む照会を作成する場合、つまりドキュメントのコンテンツ内の語とともに特定の属性値を検索する場合に、特に有用です。

ブラウザで Java が有効になっていることを確認します。有効にするには、「設定」の「詳細」を使用します。

| | |
|---|--|
| 注 | 「Version Control」と「Link Management」の属性は、iPlanet Web Server では使用されなくなりました。ただし、ガイド付き検索を実行すると、依然として iPlanet Web Server がこれらの属性を返すことがあります。これらの変数は使用しないでください。 |
|---|--|

ガイド付き検索ページを表示するには、次の 2 つの方法があります。

- 検索ホームページから表示する
- 標準検索照会ページから表示する

検索ホームページからガイド付き検索ページにアクセスするには、次の手順を実行します。

1. Web ブラウザの location (アドレス指定) フィールドに次の URL を入力します。
`http://server_root:port/search`
2. ホームページの「ガイド付き検索 (Guided Search)」リンクをクリックします。

標準検索照会ページからガイド付き検索インタフェースにアクセスするには、次の手順を実行します。

1. Web ブラウザの location (アドレス指定) フィールドに次の URL を入力して、標準検索照会ページを表示します。
`http://server_root:port/search`
 2. 標準検索ページの「ガイド付き検索 (Guided Search)」をクリックします。Java ベースのガイド付き照会ページが表示されます。
 3. 「Search in」フィールドのドロップダウンリストで、検索対象のコレクションを選択します。
 4. 「for」ドロップダウンリストで、検索する要素のタイプを選択します。ここでは、「Words」を選択したと仮定して説明を進めます。
 5. 空白のテキストフィールドに、検索する語句を入力します。
- 検索演算子の詳細は、274 ページの「照会演算子の使用」を参照してください。

6. 「Add Line」をクリックして、照会の最初の条件を追加します。フォーム下部の大きなテキスト表示ボックスに、語句が表示されます。
7. 照会に追加する別の要素を、ドロップダウンリストから選択します。ここでは、「Attribute」を選択したと仮定して説明を進めます。
8. 選択されているコレクションで使用可能なすべての属性を新たに表示したドロップダウンリストから、検索する属性を選択します。
9. テキスト入力フィールドの上のドロップダウンリストで、照会に適用する照会演算子 (Contains、Starts、Ends、Matches、Has a substring) または論理演算子 (=、<、>、<=、>=) を選択します。
10. 空白のテキストフィールドに、検索する属性値を入力します。
11. 次のいずれかを選択します。
 - 照会にさらに行を追加する場合は、「Add Line」を選択します。
 - 追加した最後の行を削除する場合は、「Undo Line」を選択します。
 - 照会全体を削除する場合は、「Clear」を選択します。
12. 「Search」 ボタンをクリックして検索を実行します。

拡張検索 (Advanced HTML Search)

拡張 HTML 検索インターフェースを使用すると、照会の作成に役立ちます。このインターフェースは、複数のコレクションを検索対象とする照会を作成する場合や、検索結果を特定の属性値によってソートする場合に特に便利です。

拡張 HTML 検索ページを表示するには、次の 2 つの方法があります。

- 検索ホームページから表示する
- 標準検索照会ページから表示する

検索ホームページから拡張 HTML 検索ページにアクセスするには、次の手順を実行します。

1. Web ブラウザの location (アドレス指定) フィールドに次の URL を入力します。
`http://server_root:port/search`
2. ホームページの「拡張検索 (Advanced HTML Search)」リンクをクリックします。

標準検索照会ページから拡張 HTML 検索ページにアクセスするには、次の手順を実行します。

1. Web ブラウザの location (アドレス指定) フィールドに次の URL を入力して、標準検索照会ページを表示します。

`http://server_root:port/search`

2. ブラウザの「設定」の「詳細」を使用して Java を無効にします。
3. 標準検索ページの「ガイド付き検索 (Guided Search)」をクリックします。Web サーバに拡張 HTML 照会ページが表示されます。
4. 「検索文字列 (For)」フィールドに、検索する語句を入力します。

演算子を組み合わせることによって、複雑な照会を作成することもできます。検索演算子の詳細は、274 ページの「照会演算子の使用」を参照してください。
5. 結果をソートする基準となる 1 つまたは複数の属性を入力します。

デフォルトでは昇順でソートされますが、マイナス記号を使用して降順にソートされるように指定することもできます。ソートの詳細は、272 ページの「結果のソート」を参照してください。
6. 検索結果ページに表示する各ドキュメントのフィールド数、または一度に表示するフィールド数を指定すると、1 回の検索で返されるドキュメントの数を調整できます。

結果が複数のページに表示される場合は、「前へ (Prev)」ボタンと「次へ (Next)」ボタンを使用して、前後のページを表示します。
7. 「検索対象 (Search in)」フィールドのドロップダウンリストで、検索対象のコレクションを選択します。

複数のコレクションを選択する場合は、Ctrl キーを押しながら別のコレクションをクリックします。ただし、照会対象のコレクションは、すべて同じ言語でなければなりません。
8. 「検索 (Search)」ボタンをクリックして、照会を実行します。

検索結果

標準的な検索結果は、次の 2 タイプです。

- 検索条件に一致するすべてのドキュメントのリスト
- 一致するドキュメントのリストから選択した 1 つのドキュメントのテキスト

検索プロセスでは、次の各時点で、アクセス権限が確認されます。

- 検索結果内で強調表示されているドキュメントに対して表示されるアイコンをユーザがクリックするとき。

- 「NS-collection-acl-check」オプションが「yes」に設定されているコレクションを検索するとき。「NS-collection-acl-check」の設定は、すべてのコレクションに適用されます。このオプションが設定されている場合、dblist.ini でそのコレクション用に定義されたプライマリドキュメントディレクトリと一致する URI に対して設定された ACL が適用され、それらのコレクションに対する検索が実行されないようにします。

検索条件を満たすドキュメントのリスト表示

デフォルトでインストールした iPlanet Web Server では、標準検索照会ページまたは拡張検索照会ページから検索を実行すると、検索条件に一致するドキュメントのリストが返されます。このリストには、コレクションの書式に基づいて、各ファイルの標準的な情報が含まれます。たとえば、email コレクションのデフォルトの結果ページでは各エントリの subject、to、from、および date が示され、news コレクションの場合は各エントリの subject、from、および date が示されます。

コレクション内のファイル形式の種数により、検索に使用できるデフォルトの属性が決まります。各ファイル形式の属性については、257 ページの「コレクションの属性について」を参照してください。

検索結果のエントリ間の近似性や一致の厳密性をチェックしてスコアを表示することにより、ファイルをランク付けできます。

一致したドキュメントが 1 ページに収まりきらない場合、「次へ (Next)」クリックすると次の検索結果を表示できます。照会データを新たに入力して「検索 (Search)」をクリックすれば、いつでも新規の検索を実行できます。

結果のソート

デフォルトでは、つまり拡張 HTML 照会ページの「ソート属性 (Sort by)」フィールドに何も入力しなければ、検索条件に一致するすべてのドキュメントが次の基準に従って返されます。

- 相対ランキング (相対ランキングを適用する照会の場合)
- サーバのファイルデータベース内の位置 (それ以外の照会の場合)

「ソート属性 (Sort by)」フィールドに属性名を入力すると、ドキュメントが昇順で表示されます。ドキュメントを降順に表示するには、-keywords や -title のように、属性の前にマイナス記号 (-) を付けます。Author、-PubDate のように複数のフィールドに入力すると、複数の条件でソートできます。

通常、検索結果が比較的少ない照会の場合にはソート順は重要ではありませんが、大量の検索結果が表示される照会になるほど、有用な検索結果を得るためにソート値を設定したい場合があります。ただし、特殊なソート順を指定すると、検索のパフォーマンスに影響をおよぼす場合があります。

注 META タグフィールド内の属性値に指定できるのは、テキスト文字列のみです。このため日付や時刻などのすべての数値は、テキストとしてソートされます。META タグ属性内の無効な HTML 文字は、ハイフンで置き換えられます。

強調表示されているドキュメントの閲覧

デフォルトでインストールした iPlanet Web Server では、検索条件に一致するドキュメントのリストを取得したときに、1つのドキュメントを選択して Web ブラウザに表示できます。パターンファイルでどのように設定されているかにより、ドキュメントを表示する際に、検索照会に入力した語を色、太字、または点滅によって強調表示することができます。

強調表示されているドキュメントを閲覧するには、検索結果の中からそのドキュメントのエントリをクリックします。強調表示されているドキュメントへのアクセスに使用するフィールドは、検索インタフェースでどのように設計されているかによって異なりますが、デフォルトのインストールでは、リストされているそのドキュメントの隣にあるアイコンをクリックします。そのアイコンのリンクに追加されているコードにより、検索照会の語を強調表示して表示するドキュメントの書式が定義されます。

デフォルトの検索結果ページでは、ファイルの URL をクリックすると、そのファイルは強調表示なしでブラウザに表示されます。

HTML に変換されたドキュメントの場合は、オリジナルのドキュメントへの URL が表示されます。変換された HTML ドキュメントを表示するには、ドキュメントのタイトルをクリックします。

コレクションのコンテンツの表示

コレクションデータベースの内容を表示して、各コレクションに設定されている属性を確認できます。デフォルトでインストールした iPlanet Web Server では、`dblist.ini` ファイルで表示可能 (`NS-display-select = YES`) に設定された各コレクションに関する情報の表示には `HTML-description.pat` ファイルが使用されます。通常、コレクションの内容には次の項目が含まれます。

- コレクション名、ラベル、および説明
- コレクションの書式
- コレクション内の属性数および属性名のリスト
- コレクション内のドキュメント数
- コレクションのサイズおよび状態
- 言語および文字セット

- 入力および出力の日付の書式

コレクションデータベースの内容を表示するには、次の URL を使用します。

```
http://server_root:port/search?NS-search-page=c
```

照会演算子の使用

効果的に検索を行うためには、照会演算子の使用方法を理解しておく必要があります。照会演算子を使用した検索を実行できるのは、ブール (**boolean**) 検索だけです。このため、この項目での説明はすべてブール検索規則に基づいたものになります。

注 照会言語は、大文字と小文字を区別しません。この例で大文字を使用しているのは、分かりやすくするためです。

検索エンジンは、一連の構文規則に基づいて検索照会を解釈します。たとえば、「region」という語を入力すると、元の「region」という語に加えて、「regions」や「regional」などの派生語がすべて検索されます。検索結果は、重要度の高い順(つまり、一致した語が入力した元の検索条件にどれほど近いかに)ランク付けされます。region の例では、「region」そのものが、他のどの派生語よりも上位にランク付けされます。

すべての照会で、結果がランク付けされるわけではありません。一致の度合いが一律でない照会に限り、ランク付けが可能です。たとえば、<CONTAINS 照会は、指定された文字列が含まれているか含まれていないかのどちらかですが、<NEAR 照会では、指定されている単語間の近接度によってランク付けすることが可能です。指定されている単語間の距離が近いほど検索結果の上位に示され、離れているほど下位に置かれます。

この節では、次の内容について説明します。

- デフォルトの想定
- 検索規則
- 使用する演算子の決定
- ワイルドカードの使用

デフォルトの想定

検索照会言語には、ユーザの入力を解釈する方法に関する暗黙のデフォルトおよび想定事項がいくつかあります。場合によってはデフォルトを回避できる場合もありますが、検索エンジンは、次の項目を使用して返すべき検索結果を決定します。

- **<STEM>** 検索する語句そのものおよび派生語 (意味が同じ物から派生した) を含むすべてのドキュメントを検索します。検索エンジンは、単なる綴りではなく、語の意味に注目します。たとえば、「plan」という語を検索する場合、「planning」や「plans」を含むドキュメントは検索結果に含まれますが、「plane」や「planet」を含むドキュメントは含まれません。
- **<MANY>** 検索の語句がドキュメント内に登場する回数を計算し、頻度 (または関連性) に基づいて結果をランク付けします。
- **<PHRASE>** 空白で区切られた複数の語を、1つの句の一部であると見なします。たとえば、「Monterey otter」は1つの句として解釈されるため、両方の語が存在し、共に見つかる必要があります。この種の検索では、「sea otter」や「Monterey Bay」を含むドキュメントは一致の対象にはなりません。
2つの語が1つの句と見なされるかどうかは明確でない場合には、括弧 () を使って句であることを明示することができます。(例、<PHRASE>(rise "and" fall))
- **OR** 照会内のコンマで区切られた各語や句は、省略可と見なされます。ただし、少なくとも1つは存在しなければなりません。効果の上では、これは暗黙的に OR と同じになります。たとえば「Monterey, otter」と指定すると、「Monerey」または「otter」のいずれかを含むドキュメントが検索されます。OR を山括弧 (<>) で囲む必要はありません。

検索規則

複雑な検索を作成するには、次のような方法があります。

- 照会演算子を組み合わせる
- 照会の構文を操作する
- ワイルドカード文字を使用する

山括弧 (<>)

AND、OR、NOT、DATE および数値比較演算子を除き、<CONTAINS> や <WILDCARD> のように、照会演算子を山括弧 (<>) で囲む必要があります。

演算子を組み合わせる

1つの照会内で複数の照会演算子を組み合わせ、より厳密な結果を取得できます。たとえば、次の照会を入力すると、「Bay」と「Monterey」を含み、「Aquarium」を含まないドキュメントを検索できます。

```
Monterey AND Bay NOT <CONTAINS> Aquarium
```

暗黙的な句をいくつか含めることにより、さらに精度を高めることができます。たとえば、次の照会では、「Monterey Bay Aquarium」全体および「otters」を含み、かつ「shark」を含まないドキュメントを検索できます。

```
Monterey Bay Aquarium AND otter AND NOT shark
```

照会演算子を検索語として使用する

照会演算子は、引用符で囲むことにより、検索語として使用できます。たとえば、次の照会を入力すると、「ebb and flow」という語を含むドキュメントを検索できます。

```
<CONTAINS> ebb "and" flow
```

派生語検索の取り消し

暗黙の派生語検索を実行しない場合は、語を引用符で囲みます。次に例を示します。

```
"plan"
```

この照会では、「plan」と完全に一致する単語が含まれるドキュメントだけが検索されます。「plans」や「planning」を含むドキュメントであっても、「plan」そのものが含まれていなければ、無視されます。

演算子の修飾

演算子 AND、OR、および NOT を使用して、他の演算子を修飾することができます。たとえば、タイトルに「theme park」という句を含むドキュメントを検索から除外する場合、次の照会を入力します。

```
Title NOT <CONTAINS> theme park
```

使用する演算子の決定

次の情報を参考にして、使用する演算子を決定してください。特に、照会言語では、大文字と小文字は区別されないことに注意してください。このため、<starts> と <STARTS> は等価になります。このドキュメントで大文字を使用しているのは、分かりやすくするためです。

表 12-3 使用する演算子の決定

| 検索のタイプ | 有効な演算子 | 例 |
|---|--|---|
| 日付または数値の比較によりドキュメントを検索します | <ul style="list-style-type: none"> • 大なり (>) • 以上 (>=) • 小なり (<) • 以下 (<=) | <p>DATE >= 06-30-96</p> <p>1996年6月30日以降に作成されたドキュメントを検索します</p> |
| 特定のドキュメントフィールド内またはフィールド内の特定位置内の語句を検索します | <ul style="list-style-type: none"> • <STARTS> • <CONTAINS> • <ENDS> • 等価 (=) | <p>Title <STARTS> Help</p> <p>タイトルが「Help」で始まるドキュメントを検索します</p> |
| ドキュメント内の複数の語を検索します | <ul style="list-style-type: none"> • AND • <NEAR/1> | <p>specifications AND review</p> <p>「specifications」と「review」の両方を含むドキュメントを検索します</p> |

次の表に、よく使用される演算子の説明と使用例を示します。別途明記されていない限り、検索結果はすべて、相対ランク付けされた順序で表示されます。

表 12-4 照会言語演算子

| 演算子 | 説明 | 例 |
|------------|---|--|
| AND | <ul style="list-style-type: none"> • 必須条件を検索に追加します。 • 指定された語すべてを含むドキュメントを検索します | <p>Antarctica AND mountain climb</p> <p>「Antarctica」と「mountain climb」の両方およびその派生語(「mountain climbing」など)を含むドキュメントのみを検索します</p> |
| <CONTAINS> | <ul style="list-style-type: none"> • 指定された語をドキュメントフィールドに含むドキュメントを検索します。複数の語が指定されている場合は、それらの語が指定と同じ順序で連続しているものが検出されます • ワイルドカードを使用できます。ワイルドカードで検索されるのは、英数字のみです • ドキュメントの相対ランク付けは、行われません | <p>Title <CONTAINS> higher profit</p> <p>タイトルに「higher profit」という句を含むドキュメントを検索します。タイトルに「profits higher」を含んでいても無視されません</p> |

表 12-4 照会言語演算子 (続き)

| 演算子 | 説明 | 例 |
|-----------|--|--|
| <ENDS> | <ul style="list-style-type: none"> ドキュメントフィールドが特定の文字列で終わるドキュメントを検索します ドキュメントの相対ランク付けは、行われません | <p>Title <ENDS> draft</p> <p>タイトルが「draft」で終わるドキュメントを検索します</p> |
| 等価 (=) | <ul style="list-style-type: none"> ドキュメントフィールドが特定の日付または数値と一致するドキュメントを検索します | <p>Created = 6-30-96</p> <p>1996年6月30日に作成されたドキュメントを検索します</p> |
| 大なり (>) | <ul style="list-style-type: none"> ドキュメントフィールドの値が特定の日付または数値より大きいドキュメントを検索します | <p>Created > 6-30-96</p> <p>1996年6月30日より後に作成されたドキュメントを検索します</p> |
| 以上 (>=) | <ul style="list-style-type: none"> ドキュメントフィールドの値が特定の日付または数値と等しいか、それより大きいドキュメントを検索します | <p>Created >= 6-30-96</p> <p>1996年6月30日以降に作成されたドキュメントを検索します</p> |
| 小なり (<) | <ul style="list-style-type: none"> ドキュメントフィールドの値が特定の日付または数値より小さいドキュメントを検索します | <p>Created < 6-30-96</p> <p>1996年6月30日より前に作成されたドキュメントを検索します</p> |
| 以下 (<=) | <ul style="list-style-type: none"> ドキュメントフィールドの値が特定の日付または数値と等しいか、それより小さいドキュメントを検索します | <p>Created <= 6-30-96</p> <p>1996年6月30日以前に作成されたドキュメントを検索します</p> |
| <MATCHES> | <ul style="list-style-type: none"> ドキュメントフィールド内に、ユーザが指定した文字列を含むドキュメントを検索します 部分的に一致する文字列を含むドキュメントは無視されます ドキュメントの相対ランク付けは行われません | <p><MATCHES> employee</p> <p>「employee」またはその派生語のいずれか (「employees」 など) を含むドキュメントを検索します</p> |
| <NEAR> | <ul style="list-style-type: none"> 指定された複数の語を含むドキュメントを検索します。各語のドキュメント内での位置に近いほど、ドキュメントは上位にランク付けされます | <p>stock <NEAR> purchase</p> <p>「stock」と「purchase」の両方を含むドキュメントを検索します。ただし、「purchase supplies」と「stock up」を含むドキュメントよりも、「stock purchase」を含むドキュメントの方が上位にランク付けされます</p> |

表 12-4 照会言語演算子 (続き)

| 演算子 | 説明 | 例 |
|--------------------|---|--|
| <NEAR/N> | <ul style="list-style-type: none"> 指定された複数の語が N 語以内の位置にあるドキュメントを検索します。N には、1000 までの整数を指定できます。各語のドキュメント内での位置が近いほど、ドキュメントは上位にランク付けされます | <p>stock <NEAR/1> purchase</p> <ul style="list-style-type: none"> 「stock purchase」および「purchase stock」という句を含むドキュメントを検索します 「purchase supplies and stock up」のように、「stock」と「purchase」が隣接していない句が含まれていても、そのドキュメントは無視されます N に 2 以上の値を指定すると、指定された語をその範囲内に含むドキュメントが検索され、各語の位置が近いほどドキュメントが上位にランク付けされます |
| NOT | <ul style="list-style-type: none"> 特定の語や句を含まないドキュメントを検索します <p>注：NOT を使用して、OR または AND 演算子を修飾できます</p> | <p>surf AND NOT beach</p> <p>「surf」という語を含み、かつ「beach」という語を含まないドキュメントを検索します</p> |
| OR | <ul style="list-style-type: none"> 検索に省略可能な条件を追加します 少なくとも 1 つの検索値を含むドキュメントを検索します | <p>apples OR oranges</p> <p>「apples」または「oranges」のいずれかを含むドキュメントを検索します</p> |
| <PHRASE> | <ul style="list-style-type: none"> 指定された句を含むドキュメントを検索します。句とは、複数の語をグループ化したもの (特定の順序を持つ) を指します | <p><PHRASE> (rise "and" fall)</p> <p>「rise and fall」という句を含ドキュメントを検索します。引用符で囲むことにより、and は、検索実行時に演算子ではなく文字列として解釈されます</p> |
| <STARTS> | <ul style="list-style-type: none"> ドキュメントフィールドが特定の文字列で始まるドキュメントを検索します ドキュメントの相対ランク付けは行われません | <p>Title <STARTS> Corp</p> <p>「Corporate」や「Corporation」のように、タイトルが「Corp」で始まるドキュメントを検索します</p> |
| <STEM> (英語のみ) | <ul style="list-style-type: none"> 指定された語およびその派生語 (意味が同じものから派生した) を含むドキュメントを検索します | <p><STEM> plan</p> <p>「plan」、「plans」、「planned」、「planning」など、同じ意味の語幹の語を含むドキュメントを検索します。「planet」や「plane」のように、綴りが似ていても意味が異なる語は無視されます</p> |

表 12-4 照会言語演算子 (続き)

| 演算子 | 説明 | 例 |
|-------------|---|---|
| <SUBSTRING> | <ul style="list-style-type: none"> ドキュメントフィールド内に、指定された文字列の全体または一部分を含むドキュメントを検索します <MATCHES> に似ていますが、この演算子では部分文字列にも一致しません ワイルドカードは、共に使用しても機能しません ドキュメントの相対ランク付けは行われません | <p><SUBSTRING> employ</p> <p>「employ」の全体または一部と一致するドキュメントを検索します。この場合、「ploy」も検索に当てはまります</p> |
| <WILDCARD> | <ul style="list-style-type: none"> 検索文字列にワイルドカード文字を含めて、ドキュメントを検索します。綴りが類似しているが、特定の語の意味的派生語としては検索できない語を指定する場合に、これを使用します * や ? などの特定の文字を指定すると、ワイルドカードによる検索が自動的に行われます。この場合、<WILDCARD> という語を含める必要はありません | <p><WILDCARD> plan*</p> <ul style="list-style-type: none"> 「plan」、「plane」、「planet」、その他「plan」で始まる全ての語 (「planned」、「plans」、「planetopolis」 など) を含むドキュメントを検索します 詳細および例は、次の節を参照してください |
| <WORD> | <ul style="list-style-type: none"> 指定された語を含むドキュメントを検索します | <p><WORD> theme</p> <p>「theme」そのものの他に、「thematic」、「themes」など、「theme」を語幹に持つ語を含むドキュメントを検索します</p> |

ワイルドカードの使用

ワイルドカードを使用すると、特殊な結果を取得できます。たとえば、綴りが似ているが、同じ意味の語幹からの派生ではない語を含むドキュメントも検索できます。たとえば、*plan* の意味的派生語には *plans*、*planning* が含まれますが、*plane* や *planet* は含まれません。ワイルドカードを使用すると、これらの語もすべて検索できます。

サポートされているワイルドカード文字は、「*」および「?」のみです。これらの語を指定すると、自動的にワイルドカードベースの検索が行われます。このため、式に <WILDCARD> 演算子を指定する必要はありません。

表 12-5 ワイルドカード演算子

| 文字 | 説明 |
|----|--|
| * | <ul style="list-style-type: none"> 0 個以上の英数字を意味します。たとえば、<code>air*</code> を指定すると、「air」そのものを含めて、「airline」や「airhead」などを含むドキュメントが検索されます ただしこのワイルドカードは、式の最初の文字として指定することはできません このワイルドカードが (<code>[]</code>) または (<code>{ }</code>) で囲まれている場合、ワイルドカードは無視されます このワイルドカードを使用すると、<code><WILDCARD></code> 演算子が暗黙的に指定されます |
| ? | <ul style="list-style-type: none"> 1 つの英数字を意味します。「?」を複数個使用して、複数の文字を指定することもできます。たとえば、<code>?at</code> を指定すると、「cat」および「hat」などを含むドキュメントを検索します。<code>??at</code> を指定すると、「that」および「chat」などを含むドキュメントを検索します このワイルドカードが (<code>[]</code>) または (<code>{ }</code>) で囲まれている場合、ワイルドカードは無視されます このワイルドカードを使用すると、<code><WILDCARD></code> 演算子が暗黙的に指定されます |

英数字以外の文字

英数字以外の文字を検索するには、コレクションの作成に使用された `style.lex` ファイルが、英数字以外の文字を認識するように設定されている必要があります。このファイルは、`server_root\plugins\search\common\style\` ディレクトリの `html`、`news`、および `mail` のサブディレクトリにあります。

検索インターフェースのカスタマイズ

サーバ管理者は、ユーザの特定の要件に合わせて検索インターフェースをカスタマイズすることができます。ユーザに表示される HTML ベースのフォームはすべて、パターンファイルのセットによって次のことを行うように定義されます。

- 検索結果ページのヘッダーとフッターの書式を表示する
- 照会に応答して表示される各検索結果のレコードを表示する

検索の入力および出力に使用されるフォームを作成するための一連のパターン変数があります。この変数の多くは、システム構成ファイルおよびユーザ構成ファイルの `userdefs.ini` および `dblist.ini` に定義されています。これらのファイルについては、253 ページの「手動によるファイルの構成」を参照してください。

注 `http://server_root:port/search`にある検索ホームページでも、検索インタフェースの紹介、およびインタフェースのカスタマイズに関するオンライン形式の **QuickStart** チュートリアルを参照できます。このチュートリアルには、各種のパターンファイルの説明、およびパターンファイルを変更して別の方法で結果を表示する例が示されています。

この節では、次の内容について説明します。

- 動的に生成されるヘッダーとフッター
- HTML パターンファイル
- 検索関数の構文
- パターン変数の使用

動的に生成されるヘッダーとフッター

動的に生成されるヘッダーとフッターを指定できます。そのためには、`add-headers` および `add-footers` 指令を **Service** 関数として `obj.conf` ファイルに追加します。これらの指令には、パスまたは URI のどちらかのパラメータが必要です。ヘッダーまたはフッターとして静的ファイルを指定する場合は、パスのパラメータを使用します。次に例を示します。

```
Service fn="add-headers" path="/export2/docs/header.html"  
Service fn="add-footer" path="/export2/docs/footer.html"
```

CGI プログラムなどのように動的に生成されるファイルをヘッダーまたはフッターとして指定する場合は、URI パラメータを使用します。次に例を示します。

```
uri="/cgi-bin/header.cgi"
```

これらの **Service** 関数は、`send-file` や `send-cgi` など、要求に応答する実際の **Service** 関数の前に置く必要があります。

HTML パターンファイル

インターフェースのカスタマイズは、まず、既存のパターンファイルを変更することから始めるとよいでしょう。パターン変数とその働きについて理解すれば、独自のパターンファイルを作成し、それらのファイルをポイントするように構成ファイルやほかのパターンファイルを変更することができます。デフォルトでインストールした iPlanet Web Server では、パターンファイルは `server_root\plugins\search\ui\text` ディレクトリにあります。後で復元できるように、元のパターンファイルのコピーを作成しておくことをお勧めします。

email、news、ASCII、PDF、HTML など、各種のコレクション用のパターンファイルがあります。また、汎用タイプのパターンファイルもいくつかあり、それぞれに特定の用途があります。ASCII-record.pat、EMAIL-record.pat などのように、ファイルの接頭辞によって、どのタイプのファイルに使用するパターンファイルであるかわかります。汎用パターンファイルには、次の種類があります。

- `NS-query.pat` は、標準照会ページおよび拡張照会ページを表示します。検索照会ページ上に Web 検索 (「Web 検索 (Search the Web)」ボックス) を呼び出す HTML が記述されています。
- `tocstart.pat` は、検索結果ページの上部にヘッダーを表示します。
- `tocrec.pat` は、検索結果ページにリストされた各ドキュメントを表示します。
- `tocend.pat` は、検索結果ページの下部にフッターを表示します。
- `record.pat` は、検索結果ページから、強調表示された単一のドキュメントを表示します (詳細は、273 ページの「強調表示されているドキュメントの閲覧」を参照)。
- `descriptions.pat` は、コレクションの内容を表示します。

パターンファイルには、要素の表示方法を定義する HTML フォーマット命令と、表示されるテキストラベルや値を定義する HTML 検索の引数および変数が記述されています。

パターン変数には次の 3 種類があります (詳細は、287 ページの「パターン変数の使用」を参照)。

- **ユーザが定義する変数** : `userdefs.ini` ファイルに定義され、`$$` 接頭辞が付きます (287 ページの「ユーザ定義のパターン変数」を参照)。
- **構成ファイルに定義される変数** : `dblist.ini` ファイルに定義され、`$$NS-` 接頭辞が付きます (詳細は、289 ページの「構成ファイルの変数」を参照)。
- **検索マクロとパターンファイルによって生成される変数** : `$$NS-` 接頭辞が付きます (詳細は、292 ページの「マクロと生成されるパターン変数」を参照)。

標準照会のパターンファイルである `NS-query.pat` の次の行は、これらの変数がどのように連携して機能するかを示しています。

```



```

各行には、標準の HTML タグと、\$\$ 接頭辞または \$\$NS- 接頭辞が付いた 1 つまたは複数の変数が記述されています。各行についてさらに詳細に調べるためには、253 ページの「手動によるファイルの構成」で説明されている、構成ファイルを調べる必要があります。

- NS-max-records: このフィールドは非表示のため、ユーザはこの値を変更できません。この値は、一度に返す一致ドキュメント数を定義します。ただし、拡張 HTML 照会パターンファイルの NS-advquery.pat では、ユーザが変更できる入力フィールドになります。
- \$\$NS-max-records: 検索を実行すると、このフィールドから変数が生成され、後続の検索で、一度に表示する結果レコード数を計算するためにその変数を使用できます。拡張照会では、この値を照会ごとに変えることができます。
- \$\$logo: userdefs.ini ファイルに定義されます。ユーザがフォームに表示したいイメージまたはテキストを定義できます。
- \$\$sitename: \$\$NS-host 検索マクロから提供されるサーバのホスト名として、userdefs.ini ファイルに定義されます。
- \$\$queryLabel: 照会入力フィールドのテキストラベルとして、userdefs.ini ファイルに定義されます。この場合、フォーム上のラベルは「検索文字列 (For):」です。
- NS-query: 入力フィールド名として、このパターンファイルに定義されます。
- \$\$NS-display-query: userdefs.ini ファイルに定義されます。検索を実行すると、このフィールドから変数が生成され、後続の検索で、一致ドキュメントの全文を表示するときどの語または句を強調表示するかを決定するためにその変数を使用できます。

検索関数の構文

検索関数では、標準の URL 構文を、一連の検索引数の「名前 - 値」ペアとともに使用します。基本的な構文を次に示します。

```
http://server_root/search?name=value [&name=value] [&name=value]
```

HTML の検索照会ページと検索結果ページを使用する場合は、ブラウザの URL フィールドに、検索関数と引数が表示されます。URL フィールドに直接入力する場合は、修飾 URL と呼ばれることもあります。また、HREF タグを使用してパターンファイルに組み込むこともできます。

パターンファイル内の HREF 要素として、完全な検索関数を作成できます。次の例は、HTML-descriptions.pat ファイルの一部であり、コレクション情報の表示方法が定義されています。この例では、ラベル (「コレクション (Collection):」) で各コレクションのヘッダーを作成し、dblist.ini ファイルに定義されているコレクションのラベル (NS-collection-alias) により、実際のコレクションファイルへのリンクを提供します。

```
<td colspan=6><font size=+2><b>$$collectionLabel</b>
<a href=$$NS-server-url/search?NS-collection=$$NS-collection>
$$NS-collection-alias</a>
</font></td>
```

この例の HREF では、次の各要素を使用して、完全な検索関数を記述しています。

- \$\$NS-server-url : ユーザのサーバ URL を特定する検索マクロ。
/search : 検索コマンド。
- ? : 照会文字列インジケータ。? の後の文字列はすべて、検索関数に使用される情報です。
- NS-collection=\$\$NS-collection : \$\$NS-collection 検索マクロを使用して、コレクションのファイル名を定義します。

検索で、条件付きで変数が使用されるように、つまり、変数に対応する値がない場合は何も表示されないように設定することができます。その構文を次に示します。

```
variableName[conditionalized output]
```

たとえば、ドキュメントのタイトルが存在する場合は出力されるように要求できます。このドキュメントにタイトルがない場合は、「Title:」のラベルも表示されないようにすることができます。その場合は、次のように入力します。

```
$$Title [<P>Title:<B>$$Title</B>]
```

URL エンコーディング

HTML 命令を作成するときは、修飾 URL 内でもパターンファイル内でも、URL エンコーディングの規則に従う必要があります。URL の一部と誤って解釈される可能性のある文字は、%nn の書式でコード化する必要があります。nn は、16 進コードを表します。空白文字は、照会では+ (プラス記号) に、出力では %20 に変換されます。次の表は、もっともよく使用される URL コードを示しています。

表 12-6 一般的な URL エンコーディング

| 文字 | 説明 | コード |
|----|--------|-----|
| | 空白文字 | %20 |
| ; | セミコロン | %3B |
| / | スラッシュ | %2F |
| ? | 疑問符 | %3F |
| : | コロン | %3A |
| @ | アットマーク | %40 |
| = | 等号 | %3D |
| & | アンパサンド | %26 |

必要な検索引数

照会ページと結果ページのほとんどすべての要素をカスタマイズできますが、検索関数で各種の検索ページを表示するために必要な引数がいくつかあります。検索関数を修飾 URL として入力する場合も、HREF としてパターンファイルに組み込む場合も、それらの引数は必要です。

検索照会ページを表示する検索関数では、次の引数を必要とします。

- 検索照会 (検索する語、句、または属性)
- コレクション (複数コレクションの検索では複数回指定できる)

検索結果ページを表示する検索関数では、次の引数を必要とします。

- NS-search-page=results (または、大文字か小文字の r)
- コレクション (複数コレクションの検索では複数回指定できる)
- 検索照会

強調表示されたドキュメントを表示する検索関数では、次の引数を必要とします。

- NS-search-page=document (または、大文字か小文字の d)
- ドキュメントのパス
- コレクション (1 回だけ指定できる)
- 検索照会 (照会データを強調表示したい場合は、必要)

コレクションの内容を表示する検索関数では、次の引数だけが必要です。

- NS-search-page=contents (または、大文字か小文字の c)

パターン変数の使用

パターン変数を使用して、検索テキストインターフェースをカスタマイズできます。この方法では、ユーザの要求が変わっても、実際の HTML ページを更新する必要がありません。たとえば、インターフェースのグラフィックスやテキスト要素を定期的に変更する場合、そのグラフィックスまたはテキストが保守および保存されている場所のパス名をポイントするパターン変数を定義することができます。

パターン変数には次の 3 つのカテゴリがあります。

- `userdefs.ini` ファイルに定義される変数。修飾 URL およびパターンファイルで使用する場合は、`$$` 接頭辞を追加します。たとえば、`uidir`、`logo`、および `title` は、`$$uidir`、`$$logo`、および `$$title` になります。
- `dblist.ini` 構成ファイルに定義される変数。構成ファイル内の定義では `NS-` 接頭辞を付け、修飾 URL およびパターンファイル内で使用する場合は `$$NS-` 接頭辞を付けます。たとえば、`NS-max-records`、`NS-doc-root`、および `NS-date-time` は、`$$NS-max-records`、`$$NS-doc-root`、および `$$NS-date-time` になります。
- 検索マクロとパターンファイルによって生成される変数。常に `$$NS-` 接頭辞が付きます。たとえば、`$$NS-host`、`$$NS-get-next`、`$$NS-sort-by` などです。

ユーザ定義のパターン変数

ユーザ定義ファイルの `userdefs.ini` では、独自のユーザ定義のパターン変数をいくつでも作成できます。また、既存の定義を変更することもできます。これらの変数をパターンファイルで使用する場合は、`$$` 接頭辞を追加します。変数名には、最大で 32 の文字または数字、あるいは両方の組み合わせを使用できます。文字には、大文字または小文字の A-Z、ハイフン (-)、および下線 (_) を使用できます。変数名は、大文字と小文字が区別されます。

iPlanet Web Server に付属するデフォルトの `userdefs.ini` ファイルには、次の項目を定義するのに使用される変数が記述されています。

- 検索照会ページ (ファイル内の `[query]`)
- 結果リスト (`[toc]`)
- ドキュメント表示ページ (`[record]`)
- コレクションの内容ページ (`[contents]`)

各行は変数名で始まり、その変数の定義が続きます。多くは画面要素のラベルですが、ほかのファイルへのパスや、もっと複雑な内容のものもあります。例として、このファイルの `[query]` セクションからの行を次に示します。

```

[query]
NS-character-set=EUC-JP
uidir = $$NS-server-url/search-ui
icondir = $$uidir/icons
l10nicondir = $$uidir/icons
htmldir = $$uidir/text
logo = <b><font
size=+1>iPlanet&nbsp;S</font><font size=+2>S</font><font
size=+1>earch</font></b>
sitename = $$NS-host
help = /manual/ug/search.htm
title = 検索インタフェースのサンプル
searchButtonLabel = 検索
searchNote = 検索を行うには、コレクションを選択し、検索語句をコンマで区切って入力します。
<br>(例： search, jet engines, basketball)
advSearchNote = 検索を行うには、コレクションを選択し、検索語句をコンマで区切って入力しま
す。<br>(例： search, jet engines, basketball)<p>指定した属性値によってソートされま
す。降順にソートする場合は「-」を属性の前に付けます。<br>(例： Title, -Author, +Date)
queryLabel = 検索文字列：
queryLabelSJIS = $$queryLabel
queryLabelEUC = $$queryLabel
queryLabelJIS7 = $$queryLabel
collectionLabel = 検索対象：
booleanLabel = ブール検索
sortByLabel = ソート属性：
sortByLabelSJIS = $sortByLabel
sortByLabelEUC = $sortByLabel
sortByLabelJIS7 = $sortByLabel
freetextLabel = フリーテキスト (使用不可)
maxDocumentsLabel = 表示するドキュメント数：
maxDocumentsLabelSJIS = $$maxDocumentsLabel
maxDocumentsLabelEUC = $$maxDocumentsLabel
maxDocumentsLabelJIS7 = $$maxDocumentsLabel
copyright = Copyright &#169; 1999-2001 Sun Microsystems, Inc. Copyright &#169;
1997-2000 Netscape Communications Corporation. All Rights Reserved.
advancedButtonLabel = 詳細ボタンラベル
helpButtonLabel = ヘルプボタンラベル

```

このファイルには、\$\$NS-server-urlなどの検索マクロへの参照も記述されており、次の各行のように、別のユーザ定義の変数を参照することもできます。

```

uidir = $$NS-server-url/search-ui
icondir = $$uidir/icons

```

検索マクロの詳細は、292 ページの「マクロと生成されるパターン変数」を参照してください。

変数の定義では、サポートされているすべての HTML 文字エンティティを使用できます。`&name;` の書式で定義されたエンティティ名および `&#nnn;` の書式で 3 桁のコードにより定義されたエンティティ名を使用できます。userdefs.ini のコーディング例では、` ` エンティティで非区切りスペースを挿入し、`©` エンティティで著作権記号を挿入しています。一般的に使用されるエンティティの一部を次の表に示します。

表 12-7 一般的な HTML 文字エンティティ

| 数値コード | エンティティ名 | 説明 |
|-------------------------|-------------------------|----------|
| <code>&#032;</code> | | 空白文字 |
| <code>&#034;</code> | <code>&quot;</code> | 引用符 |
| <code>&#036;</code> | <code>\$</code> | ドル記号 |
| <code>&#058;</code> | <code>-</code> | コロン |
| <code>&#060;</code> | <code>&lt;</code> | より小さい |
| <code>&#062;</code> | <code>&gt;</code> | より大きい |
| <code>&#153;</code> | <code>-</code> | 商標記号 |
| <code>&#160;</code> | <code>&nbsp;</code> | 非区切りスペース |
| <code>&#169;</code> | <code>&copy;</code> | 著作権記号 |
| <code>&#174;</code> | <code>&reg;</code> | 登録商標 |

構成ファイルの変数

いくつかの変数は、システム構成ファイルおよびコレクション構成ファイルに定義されます。これらの変数は、HTML ページのほかのマークアップタグと区別するために、構成ファイルでは NS- 接頭辞を使用します。これらの変数を検索関数の引数として使用する場合は、`$$NS-date-time` や `$$NS-max-records` などのように、変数にさらに `$$` 接頭辞を追加します。

1 つのサーバ上のすべての検索に対してデフォルトを定義する変数は、システム構成ファイルに定義されます。

```

NS-max-records = 20
NS-query-pat = /text/NS-query.pat
NS-ms-tocstart = /text/HTML-tocstart.pat
NS-ms-tocend = /text/HTML-tocend.pat
NS-default-html-title = (Untitled)
NS-HTML-descriptions-pat = /text/HTML-descriptions.pat
NS-date-time = %b-%d-%y %H:%M

```

各サーバの構成によってインストール内容が異なる場合がありますが、もっとも一般的に使用される変数を次の表に示します。

表 12-8 一般的に使用される変数

| 変数 | 説明 |
|--------------------------|---|
| NS-default-html-title | HTML ドキュメントにユーザ定義のタイトルが設定されていない場合に与えられる名前。通常は (Untitled) に設定します |
| NS-date-time | 結果を表示する際に使用する日付と時刻の書式 |
| NS-date-input-format | 日付を入力する際の書式 (デフォルトは MMDDYY) |
| NS-HTML-descriptions-pat | コレクションの内容を表示する際に使用するパターンファイル |
| NS-largest-set | 検索条件に一致するものとして処理できる最大レコード数。レコードは NS-max-records のグループに表示されます |
| NS-max-records | 一度に表示する結果セットの最大サイズ |
| NS-ms-tocend | 複数のコレクションを検索する場合に、検索結果ページの一番下に表示するフッターに使用するパターンファイル |
| NS-ms-tocstart | 複数のコレクションを検索する場合に、検索結果ページの一番上に表示するヘッダーに使用するパターンファイル |
| NS-query-pat | 照会ページの作成時に使用される照会パターンファイル |
| NS-search-type | 実行する検索のタイプ。ブール検索だけが可能です |

コレクション固有の変数は、dblist.ini ファイルに定義されます。このファイルに定義される変数は次のとおりです。

```

NS-doc-root = C:/iPlanet/Servers/docs
NS-url-base = /
NS-display-select = YES

```

dblist.ini ファイルに定義される変数は、使用しているコレクションのタイプによって異なります。表 12-9 は、よく使用されるコレクション固有の変数のいくつかを示しています。

表 12-9 dblist.ini でよく使用される変数

| 変数 | 説明 |
|---------------------|---|
| NS-collection-alias | コレクションのラベル。複数のコレクションを検索する場合は複数回指定できます |
| NS-doc-root | コレクション内のドキュメントのルートディレクトリ |
| NS-display-select | NS-search-page=contents の場合に、そのコレクションをコレクション情報リストの一部として表示するかどうかを指定します。デフォルトは YES です |
| NS-highlight-start | 表示されるドキュメント内で、この位置から強調表示を開始します。通常は、検索照会条件を強調表示します |
| NS-highlight-end | 表示されるドキュメント内で、この位置で強調表示を終了します |
| NS-language | コレクション内のドキュメントの言語 |
| NS-record-pat | 強調表示されているドキュメントのページを表示する際に使用するパターンファイル |
| NS-tocend-pat | 検索結果の書式設定に使用される、コレクションに関連付けられているフッターのパターンファイル |
| NS-tocrec-pat | 検索結果の書式設定に使用される、コレクションに関連付けられているレコードのパターンファイル |
| NS-tocstart-pat | 検索結果の書式設定に使用される、コレクションに関連付けられているヘッダーのパターンファイル |
| NS-url-base | ファイルの場所の特定に使用されるリンクの作成時に使用されるベース URL |

マクロと生成されるパターン変数

パターンファイルまたは修飾 URL 内で使用できる検索マクロがいくつかあります。検索関数自体によって生成されるパターン変数がいくつかあり、後続の検索要求でその変数を使用して出力の表示方法を定義できます。それらのマクロおよび変数には、その用途を示す `$$NS-` 接頭辞が付きます。

たとえば、最初の検索照会を実行して 24 個のドキュメントが結果ページに表示された後、検索によって生成された `$$NS-docs-matched` 変数と `$$NS-doc-number` 変数を、ドキュメントの 1 つを詳細に表示するドキュメントページの定義に再利用できます。この方法により、このドキュメントは最初の検索で返された 24 ドキュメントのうちの 3 番目のドキュメントであることをユーザに知らせることができます。

次の表は、後続のパターンファイルまたは修飾 URL 内で使用できる検索マクロと生成される変数を示しています。

表 12-10 マクロと生成されるパターン変数

| 変数 | 説明 |
|--|---|
| <code>\$\$NS-collection-list</code> | <code>NS-display-select</code> が YES に設定されている場合に表示する、 <code>dblist.ini</code> 内の全コレクションの HTML の複数選択リスト |
| <code>\$\$NS-collection-list-dropdown</code> | <code>NS-collection-list</code> の HTML のドロップダウンリスト |
| <code>\$\$NS-collections-searched</code> | この要求に対して検索するコレクションの数 |
| <code>\$\$NS-display-query</code> | 結果ページ用に生成される、HTML で表示可能な照会 |
| <code>\$\$NS-doc-href</code> | そのドキュメントの HTML HREF タグ。元のソースドキュメントへの URL を提供します。email の場合は <code>mailbox:/boxname?id=messageID</code> の書式、news の場合は <code>news:messageID</code> の書式を使用します |
| <code>\$\$NS-doc-name</code> | ドキュメントの名前 |
| <code>\$\$NS-doc-number</code> | 結果ページのリスト内での、ドキュメントのシーケンス番号 |
| <code>\$\$NS-doc-path</code> | ドキュメントへの絶対パス |
| <code>\$\$NS-doc-score</code> | ドキュメントのランク付け (範囲: 0 ~ 100) |
| <code>\$\$NS-doc-score-div10</code> | ドキュメントのランク付け (範囲: 0 ~ 10) |
| <code>\$\$NS-doc-score-div5</code> | ドキュメントのランク付け (範囲: 0 ~ 5) |
| <code>\$\$NS-doc-time</code> | 結果リストにあるドキュメントの作成時刻。この値を取得するには、 <code>NS-use-system-stat = YES</code> に設定する必要があります。システムの統計処理は負荷が大きいため、デフォルトでは NO に設定されています |

表 12-10 マクロと生成されるパターン変数 (続き)

| 変数 | 説明 |
|---|---|
| <code>\$\$NS-doc-size</code> | もっとも近い K バイトに丸めた、ドキュメントのサイズ。この値を取得するには、 <code>NS-use-system-stat = YES</code> に設定する必要があります。システムの統計処理は負荷が大きいため、デフォルトでは <code>NO</code> に設定されています |
| <code>\$\$NS-docs-found</code> | この要求に対して検索エンジンが検出した、実際のドキュメント数 |
| <code>\$\$NS-docs-matched</code> | この要求に対して検索から返されたドキュメント数 (<code>NS-max-records</code> の値を最大とする) |
| <code>\$\$NS-docs-searched</code> | この要求で検索対象となったドキュメント数 |
| <code>\$\$NS-get-highlighted-doc</code> | 強調表示を伴う HTML テキストとしてドキュメントを表示できるようにするための、強調表示されたドキュメントの URL を提供します |
| <code>\$\$NS-get-next</code> | この変数は、次に表示する検索結果セットを取得します。このセットは <code>NS-max-records</code> に等しく、 <code>NS-search-offset</code> を使用して配置されます |
| <code>\$\$NS-get-prev</code> | この変数は、表示済みの前の検索結果セットを取得します。このセットは <code>NS-max-records</code> に等しく、 <code>NS-search-offset</code> を使用して配置されます |
| <code>\$\$NS-host</code> | ホスト名 |
| <code>\$\$NS-insert-doc</code> | ソースドキュメントの挿入位置を示すために HTML 用の <code>NS-record-pat</code> パターンファイルで使用されるプレースホルダ |
| <code>\$\$NS-rel-doc-name</code> | ドキュメントページを作成する際に使用する、表示するドキュメントの相対名 |
| <code>\$\$NS-search-offset</code> | 検索結果として返されるレコードセットのオフセット。 <code>NS-get-next</code> および <code>NS-get-prev</code> を使用する場合に、表示するレコードセットを決定するために使用されます |
| <code>\$\$NS-server-url</code> | サーバの URL |
| <code>\$\$NS-sort-by</code> | 結果ページの項目のソート順序。コレクションで使用可能な属性を 1 つ以上選択できます。デフォルトでは、昇順にソートされます |

仮想サーバとサービスの管理

第 13 章「仮想サーバの使用」

第 14 章「仮想サーバの作成と構成」

第 15 章「プログラムによるサーバの拡張」

第 16 章「コンテンツ管理」

第 17 章「構成スタイルの適用」

仮想サーバの使用

この章では、iPlanet Web Server を使用して仮想サーバの設定および管理を行う方法を説明します。

この章は、次の節で構成されています。

- 仮想サーバの概要
- 仮想サーバで iPlanet Web Server の機能を使用する
- 仮想サーバのユーザインタフェースの使用法
- 仮想サーバの設定
- 個々の仮想サーバをユーザが監視できるようにする
- 仮想サーバの導入

仮想サーバの概要

仮想サーバを使用すると、インストールされた 1 つのサーバで、複数の会社または個人に対して、ドメイン名、IP アドレス、およびサーバ監視機能を提供できます。仮想サーバを使用してハードウェアと基本的な Web サーバの維持管理を提供しますが、ユーザからは、それぞれが専用の Web サーバを所有しているように見えます。

注 仮想サーバを使用しない場合でも、Web サーバ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構成するときは **Class Manager** の項目を使用します。Web サーバ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が作成されます。仮想サーバのユーザインタフェースを使用して、デフォルトの仮想サーバ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。

- 仮想サーバクラス
- 待機ソケット
- 接続グループ
- 仮想サーバ

仮想サーバの設定は、`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、『NSAPI プログラマーズガイド』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。

- 複数のサーバインスタンス
- 仮想サーバクラス
- 待機ソケット
- 接続グループ
- 仮想サーバ
- 要求を処理する仮想サーバの選択
- 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を移行する

複数のサーバインスタンス

これまでのリリースの iPlanet Web Server では、仮想サーバごとに固有の構成情報を柔軟に設定することができませんでした。サーバごとに個別の構成情報を簡単に設定する方法として、ほとんどの場合、ユーザは個別のサーバインスタンスを作成していました。iPlanet Web Server 6.0 では、仮想サーバクラスごとに個別の構成情報が存在します。複数のサーバインスタンスを使用することは現在も可能ですが、個別の構成情報を持つサーバを多数使用する場合は、仮想サーバの使用をお勧めします。

仮想サーバクラス

仮想サーバはクラスにグループ分けされます。クラスを使用すると、類似するサーバを一度に構成できるので、各サーバを個別に構成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は同じ基本構成情報を共有しますが、仮想サーバごとに変数を設定して構成を変更することもできます。仮想サーバ間で構成情報を共有させたくない場合は、各仮想サーバクラスに1つずつ仮想サーバを作成します。一方、複数の仮想サーバ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構成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クラスを設定できます。あるクラスの仮想サーバでは 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、304 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。仮想サーバクラスでは、次の種類のサービスを有効にするか構成することができます。

- プログラム。第 15 章「プログラムによるサーバの拡張」を参照してください。
- コンテンツ管理。第 16 章「コンテンツ管理」を参照してください。
- 構成スタイル。第 17 章「構成スタイルの適用」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、各仮想サーバの稼動時に、それぞれのサーバ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 プログラマーズガイド』を参照してください。ユーザインタフェースでの変数の使用方法については、309 ページの「変数の使用方法」を参照してください。

クラスに属する仮想サーバ

同一のクラスに属する仮想サーバを、そのクラスのメンバーと呼びます。仮想サーバの設定項目には、クラス内のすべての仮想サーバに対して構成する項目と、個別に構成する項目があります。仮想サーバの設定項目は、Class Manager の「Virtual Servers」タブで設定します。詳細は、第 14 章「仮想サーバの作成と構成」を参照してください。

デフォルトのクラス

iPlanet 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インスタンス用の仮想サーバメンバーが 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を追加できますが、デフォルトの仮想サーバ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。作成する待機ソケットにはそれぞれ、IP アドレス、ポート番号、サーバ名、およびデフォルトの仮想サーバが割り当てられます。デフォルトの仮想サーバは、待機ソケットに対して自動的に作成される接続グループに関連付けられます。1 台のマシンの特定のポートで構成済み 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。

iPlanet 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。仮想サーバを使用する場合は、各仮想サーバ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。ただし、マシンが両方のアドレスに対応するように構成されていることを前提とします。1 つのポートですべての IP アドレスを待機する 0.0.0.0 または ANY の IP アドレスを使用する場合、同じポートで特定の IP アドレスを待機する待機ソケットは追加できません。たとえば、0.0.0.0:80 (ポート 80 ですべての IP アドレスを待機する) 待機ソケットがある場合、1.2.3.4:80 の待機ソケットを追加することはできません。

さらに、待機ソケットの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。

接続グループ

各待機ソケットには、少なくとも1つの接続グループが関連付けられます。待機ソケットを作成すると、その待機ソケット用に指定したデフォルトの仮想サーバをメンバーとする接続グループも作成されます。この接続グループのIPアドレスはdefaultです。

待機ソケットのIPアドレスを0.0.0.0またはANYに設定すると、特定のIPアドレスに応答する複数の接続グループを追加できます。この機能を使用して、専用のIPアドレスを持つ仮想サーバを設定できます。

各待機ソケットには必ず、デフォルトの接続(IPアドレスがdefaultと示される)による接続グループが存在します。待機ソケットに特定のIPアドレスを設定すると、使用できる接続グループはデフォルトの接続グループだけになります。待機ソケットにanyのIPアドレスを設定した場合、その待機ソケットの他の接続グループの中に該当するIPアドレスが見つからないときは、デフォルトの接続グループが使用されます。

サーバのインストール時に、デフォルトの待機ソケットls1に、1つの接続グループがデフォルトで作成されます。デフォルトの接続グループのIPアドレスはdefault、ポートは80であり、デフォルトの仮想サーバは、インストール時に作成されたデフォルトのサーバになります。

仮想サーバごとに、その仮想サーバが対応する接続グループを1つ以上選択します。クラス全体に対して接続グループを設定することはできません。接続グループ情報は、仮想サーバクラスとは無関係です。

仮想サーバ

仮想サーバを作成するには、まず、その仮想サーバ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の種類を決める必要があります。仮想サーバを作成するには、仮想サーバID、1つ以上の接続グループ、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。

- 仮想サーバの種類

- IP アドレスベースの仮想サーバ
- URL ホストベースの仮想サーバ
- デフォルトの仮想サーバ

仮想サーバの種類

以前のバージョンの iPlanet Web Server には、ハードウェアとソフトウェアの 2 種類の仮想サーバがありました。ハードウェア仮想サーバには、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。

iPlanet Web Server 6.0 では、このような概念が当てはまらなくなっています。すべての仮想サーバに URL ホストが割り当てられます。ただし、待機ソケットおよび接続グループの情報に基づいて、仮想サーバに 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に送るかを決定します。サーバは、最初に IP アドレスを評価します。詳細は、304 ページの「要求を処理する仮想サーバの選択」を参照してください。

IP アドレスベースの仮想サーバ

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 NT 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 を使用するための方法は、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。

通常、IP アドレスベースの仮想サーバを作成するには、any の IP アドレスで待機する待機ソケットを作成し、各待機ソケットの追加接続グループを作成します。各接続グループに、固有の IP アドレスを割り当てます。次に、各接続グループに、デフォルトの仮想サーバを関連付けます。ただし、IP アドレスごとに個別の待機ソケットを作成することもできます。仮想サーバの導入方法の詳細は、316 ページの「仮想サーバの導入」を参照してください。

URL ホストベースの仮想サーバ

URL ホストベースの仮想サーバを設定するには、各仮想サーバに固有の URL ホストを割り当てます。サーバは、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 を構成します。次に、各仮想サーバの URL ホストを正しく設定します (*www.aaa.com* など)。

1 つの接続グループに関連付ける URL ホストベースの仮想サーバ数はいくつでもかまいません。

URL ホストベースの仮想サーバは、Host 要求ヘッダーを使用してユーザ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があります。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、接続グループのデフォルトの仮想サーバを受け取ります。

デフォルトの仮想サーバ

URL ホストベースの仮想サーバは、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 ヘッダーをサーバが見つけれない場合は、デフォルトの仮想サーバがその要求を処理します。

IP アドレスベースの仮想サーバでも、指定された IP アドレスを iPlanet Web Server が見つけられない場合は、デフォルトの仮想サーバがその要求を処理します。デフォルトの仮想サーバを、エラーメッセージまたは専用のドキュメントルートからのサーバページを送るように構成できます。

| | |
|----------|--|
| 注 | 接続グループのデフォルトの仮想サーバと、サーバのインストール時に作成されるデフォルトのクラスおよび仮想サーバを混同しないでください。デフォルトのクラスは、サーバのインストール時に自動的に作成され、そのサーバインスタンスの仮想サーバがそのクラスのメンバーになります。接続グループのデフォルトの仮想サーバは、デフォルトとして指定する仮想サーバです。 |
|----------|--|

デフォルトの仮想サーバは、接続グループごとに設定します。待機ソケットの作成時に、デフォルトの仮想サーバを指定します。これが、その待機ソケット用にデフォルトで作成される接続グループのデフォルトの仮想サーバになります。デフォルトの仮想サーバはいつでも変更できます。特定の IP アドレスが割り当てられている待機ソケットの場合は、この接続グループが、その待機ソケットの唯一の接続グループになります。any の IP アドレスで待機する待機ソケットの場合は、作成する接続グループごとに、デフォルトの仮想サーバを指定します。

要求を処理する仮想サーバの選択

サーバが要求を処理するには、待機ソケットから要求を受け取り、その要求を適切な接続グループと仮想サーバに振り向ける必要があります。

まず、次のように接続グループが選択されます。

- 待機ソケットが特定の IP アドレスに対して待機するように構成されている場合、接続グループは1つしかないので、その接続グループが選択されます。
- 待機ソケットが ANY で待機するように構成されている場合は、クライアントが接続した IP アドレスが、その待機ソケットに属する接続グループの IP アドレスと照合されます。一致する IP アドレスがない場合は、default の IP アドレスを持つデフォルトの接続グループが選択されます。

次のように、仮想サーバが選択されます。

- 接続グループがデフォルトの仮想サーバのみで構成されている場合は、デフォルトの仮想サーバが選択されます。
- 接続グループが複数の仮想サーバ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は、サーバの起動時に、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。

仮想サーバが決まったら、サーバは、その仮想サーバが属する仮想サーバクラスの `obj.conf` ファイルを実行します。サーバが `obj.conf` で実行する指令を決定する方法については、『NSAPI プログラマーズガイド』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:\iplanet\servers\docs` の場合、`http://www.iplanet.com/products/info.html` という要求を受け取ると、サーバは `C:\iplanet\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。

iPlanet Web Server のインストール時に、Web サーバインスタンスのドキュメントルートが指定されます。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入力すると、そのクラスに属するすべての仮想サーバ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 `$id` 変数を付けると、仮想サーバごとに、`class_doc_root/virtual_server_ID` というドキュメントルートがデフォルト設定されます。たとえば、クラスのドキュメントディレクトリが `/iplanet/servers/docs/$id` の場合、そのクラスに属する仮想サーバ `vs1` のデフォルトのドキュメントディレクトリは、`/iplanet/servers/docs/vs1` になります。

変数についての詳細は、309 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

新しい仮想サーバを作成すると、デフォルトでは、ログファイルはサーバ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、各仮想サーバのログパスを変更します。

詳細は、332 ページの「仮想サーバのログの設定」を参照してください。

前のリリースから仮想サーバを移行する

バージョン 4.x 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、『インストールと移行』を参照してください。

仮想サーバで iPlanet Web Server の機能を使用する

iPlanet Web Server には、SSL やアクセス制御など、仮想サーバ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、1つのサーバインスタンス、仮想サーバクラス、または個別の仮想サーバの構成に関係します。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。

- 仮想サーバで SSL を使用する
- 仮想サーバでアクセス制御を使用する
- 仮想サーバで CGI を使用する
- 仮想サーバで構成スタイルを使用する

仮想サーバで SSL を使用する

仮想サーバで 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を使用します。ポートは通常、443 を使用します。iPlanet 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で SSL を使用するのには困難です。サーバが要求を読み取ると、セキュリティ情報をやり取りする最初のハンドシェイクが発生済みになります。

唯一の例外は、URL ホストベースの仮想サーバのすべてが、「ワイルドカード証明書」を使用して、同一のサーバ証明書など、同じ SSL 構成を持つ場合です。詳細は、第 5 章「Web サーバのセキュリティ」を参照してください。

仮想サーバで SSL を使用方法の 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。ユーザは通常、SSL を使用しない待機ソケットから仮想サーバにアクセスします。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せん。

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。

iPlanet Web Server や仮想サーバ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 5 章「Web サーバのセキュリティ」を参照してください。仮想サーバでの SSL の設定例は、319 ページの「例 2: セキュリティ保護されたサーバ」の図を参照してください。

仮想サーバでアクセス制御を使用する

仮想サーバでは、仮想サーバ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、各仮想サーバごとにユーザ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、200 ページの「仮想サーバへのアクセス制御」を参照してください。

仮想サーバで CGI を使用する

仮想サーバで 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、346 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。

仮想サーバで構成スタイルを使用する

構成スタイルを使用すると、さまざまな仮想サーバが保持する個々のファイルまたはディレクトリに、一連のオプションを簡単に適用できます。構成スタイルの詳細は、第 17 章「構成スタイルの適用」を参照してください。

仮想サーバのユーザインタフェースの使用法

仮想サーバの作成および編集には、ユーザ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。

仮想サーバを管理するためのユーザインタフェースは、次の 3 つの部分で構成されます。

- **Server Manager** では、サーバ全体 (またはすべての仮想サーバ) に影響する項目を設定します。
- **Class Manager** では、単一のクラスおよびクラス内の仮想サーバに影響する項目を設定します。

- **Virtual Server Manager** では、個々の仮想サーバに関する項目を設定します。

また、個々の仮想サーバを所有するエンドユーザ用のユーザインタフェースも使用できます。詳細は、313 ページの「個々の仮想サーバをユーザ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。

- **Class Manager**
- **Virtual Server Manager**
- 変数の使用法
- 動的再構成

Class Manager

Class Manager にアクセスするには、次の手順に従います。

1. **Server Manager** 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。

サーバのツリービューでクラス名をクリックする、または **Server Manager** 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。

Virtual Server Manager

Virtual Server Manager にアクセスするには、次の手順に従います。

1. **Class Manager** から「**Virtual Servers**」タブをクリックします。
2. 「**Manage Virtual Servers**」をクリックします。
3. 仮想サーバを選択して、「**Manage**」をクリックします。

サーバのツリービューで仮想サーバ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` を使用しても、ユーザインタフェースを使用する場合と同じ仮想サーバ関連操作を実行できます。コマンド行ユーティリティの `HttpServerAdmin` の詳細は、384 ページの「**HttpServerAdmin (仮想サーバの管理)**」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ごとに固有の値を供給する変数を使用すると、それぞれの値を個別に指定しなくて済みます。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。ユーザインタフェースでもっとも便利な変数は、仮想サーバの ID を表す `$id` 変数です。この変数を入力すると、サーバは、その値に各仮想サーバの ID を代入します。

`$accesslog` (各仮想サーバのアクセスログのパス) や `$docroot` (各仮想サーバ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。

変数の詳細は、『NSAPI プログラマーズガイド』を参照してください。

動的再構成

動的再構成を利用すると、稼働中の Web サーバの構成を変更して、Web サーバ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構成に関する設定および属性のすべてを、サーバを再起動することなく動的に変更できます。仮想サーバのユーザインタフェースで加えたすべての変更が、サーバを再起動することなく適用されます。再構成スクリプトまたはユーザインタフェースによる変更のあと、サーバを動的に再構成できます。

UNIX プラットフォームの場合、動的再構成スクリプトは、各インスタンスのディレクトリにある `reconfig` という名前の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。この再構成スクリプトを実行するには、サーバインスタンスのディレクトリから「`reconfig`」と入力します。

NT の場合は、動的再構成スクリプトは、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。この再構成スクリプトを実行するには、サーバインスタンスのディレクトリから「`reconfig`」または「`reconfig.bat`」と入力します。

このスクリプトを実行すると、ユーザインタフェースと同様にサーバの動的再構成が起動し、再構成に関するサーバメッセージが表示されます。

動的再構成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。新しい構成を組み込む際にエラーが発生する場合は、以前の構成が復元されます。

仮想サーバの設定

仮想サーバを設定するには、次の手順に従います。

1. 待機ソケットを作成します。
2. 接続グループを作成します。
3. 仮想サーバクラスを作成します。
4. クラスのサービスを構成します。
5. 仮想サーバクラス内の仮想サーバを作成します。
6. 仮想サーバを構成します。

待機ソケットを作成する場合、デフォルトの仮想サーバのフィールドには、既存する仮想サーバを入力する必要があります。サーバのインストール時に作成された仮想サーバをデフォルトの仮想サーバとして使用し、追加の仮想サーバを作成したあとで、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. **Server Manager** 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な事項を入力します。

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。このあと、接続グループで特定の IP アドレスを指定できます。

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、サーバが自動的に生成する URL に影響し、サーバに保存されるディレクトリおよびファイルの URL には影響しません。この名前は、エイリアスを使用するサーバの場合は、エイリアス名にする必要があります。

デフォルトの仮想サーバは、他の仮想サーバが見つからなかった場合に、待機ソケットのデフォルトの接続グループに対する要求に応える仮想サーバです。

詳細は、304 ページの「要求を処理する仮想サーバの選択」を参照してください。

4. 「OK」をクリックします。

接続グループの作成

待機ソケットを追加すると、デフォルトの接続グループが自動的に追加されます。待機ソケットが any の IP アドレスに応答する場合は、接続グループをさらに追加できます。

接続グループを作成するには、次の手順に従います。

1. Server Manager から、「Preferences」タブをクリックします。
2. 「Edit Listen Sockets」をクリックします。
3. 接続グループを追加する待機ソケットに続く行で、「Groups」をクリックします。
その待機ソケットに関連付けられているグループのリストが表示されます。
4. グループを追加するには、画面の一番上の行でアクションを「Add」に設定し、各フィールドに必要な事項を入力します。

待機ソケットに単一の IP アドレスが設定されている場合は、接続グループを追加することはできません。

仮想サーバクラスの作成

仮想サーバクラスを作成するには、次の手順に従います。

1. Server Manager 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 /sid 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 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、第 16 章「コンテンツ管理」を参照してください。

仮想サーバクラスの編集または削除

仮想サーバクラスの設定を編集するには、次の手順に従います。

1. **Server Manager** 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. この仮想サーバクラスで **Accept-Language** ヘッダーの解析を使用する場合は、「**Accept Language**」フィールドに「**On**」を入力します。

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。

仮想サーバクラスと関連付けるサービスの指定

仮想サーバクラス間の違いを示す特性に、それぞれの仮想サーバ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クラスと、使用できない仮想サーバクラスを設定できます。サービスの設定方法の詳細は、第 16 章「コンテンツ管理」を参照してください。

仮想サーバの作成

仮想サーバクラスの設定が完了したら、仮想サーバを作成できます。仮想サーバは、仮想サーバクラスのメンバーであるため、**Class Manager** で作成します。

詳細は、325 ページの「仮想サーバの作成」を参照してください。

仮想サーバと関連付ける設定の指定

クラスの設定を仮想サーバ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、Class Manager で行います。

詳細は、第 14 章「仮想サーバの作成と構成」を参照してください。

個々の仮想サーバをユーザが監視できるようにする

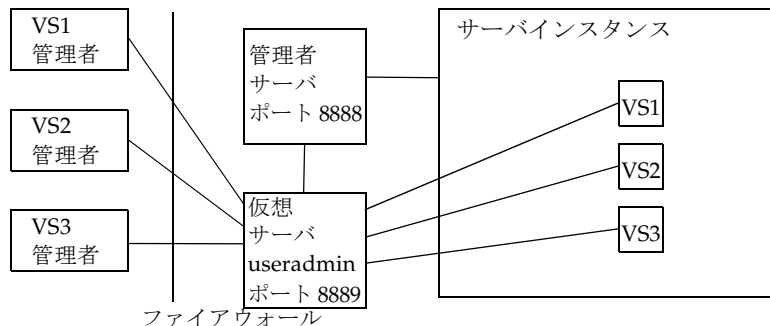
個々の仮想サーバを管理するための特別なユーザインタフェースがあります。これを利用すると、各仮想サーバの管理者は、その仮想サーバの設定を確認したり、アクセスログやエラーログを表示することができます。たとえば、3つの部門用に3つの仮想サーバを持つイントラネットの場合、それぞれの部門で、設定およびログファイルを個別に表示できます。

セキュリティ上の理由により、この管理ユーザインタフェースは、管理サーバポートまたは Web サーバインスタンスポートとは別のポートにあります。

このユーザインタフェースは、管理サーバ内の仮想サーバで稼働します。この仮想サーバは、デフォルトで設定され、**useradmin** という名前が付けられます。ユーザが管理サーバポートへのアクセス権を持たなくても仮想サーバの管理ユーザインタフェースにアクセスできるように、管理サーバが稼働する待機ソケットとは別の待機ソケットを仮想サーバに設定する必要があります。

図 13-1 は、各仮想サーバの管理者が、**useradmin** 仮想サーバから各自の仮想サーバの情報にアクセスする様子を示しています。

図 13-1 仮想サーバ管理者のユーザインタフェースの構成



個々の仮想サーバをユーザが監視できるようにする

仮想サーバをオンにすると、ユーザは、その仮想サーバを次の URL から管理できます。

`server_name:port/user-app/server_instance/virtual_server_ID`

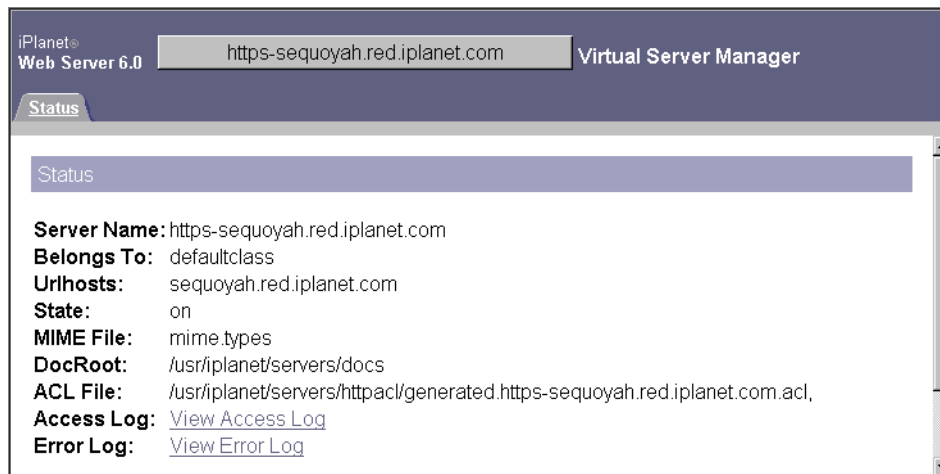
次に例を示します。

`iplanet:8889/user-app/iplanet/vs2`

サーバインスタンスには、サーバインスタンス名の「https」の部分は含めません。

図 13-2 は、エンドユーザに表示されるユーザインタフェースを示します。

図 13-2 仮想サーバの管理ユーザインタフェース



この機能を使用できるようにサーバを構成するには、次の手順に従います。

1. 管理サーバ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、管理サーバを保護できます。

セキュリティ上の理由により、ユーザ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. 新しい待機ソケットおよび接続グループをファイルに追加します。

IP アドレスは 0.0.0.0 または ANY に、ポート番号は、管理サーバのポートとは別のポートにする必要があります。デフォルトの仮想サーバは `useradmin` であることが必要です。

コード例 13-1 新しい待機ソケット

```
<LS id="ls2" ip="0.0.0.0" port="8889" security="off" acceptorthreads="1"
blocking="no">
<CONNECTIONGROUP id="group2" matchingip="default" servername="iplanet.com"
defaultvs="useradmin"/>
</LS>
```

この例では、`ls2` は、接続グループの `group2` とともに作成する待機ソケットです。

4. `useradmin` 仮想サーバ (`userclass` クラスにある) が、作成した接続グループを使用するように、`server.xml` ファイルを編集します。
5. `useradmin` 仮想サーバの状態を「on」に設定します。

コード例 13-2 編集後の useradmin

```
<VSCLASS id="userclass" objectfile="userclass.obj.conf" rootobject="default" >
<VS id="useradmin" connections="group2" state="on" mime="mime1"
urlhosts="user-app" aclids="acl1">
<VARS webapps file="user-apps.xml" webapps_enable="on"/>
<USERDB id="default" database="default" />
</VS>
</VSCLASS>
```

この例では、接続グループは前に作成した `group2` グループに設定され、状態は `on` に設定されています。

6. 変更を `server.xml` に保存します。
7. Administration Server を再起動して変更を適用します。
8. これで、どのサーバインスタンスのどの仮想サーバでも、次の URL で管理ユーザ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`

アクセス制御

権限を持たないユーザによる仮想サーバ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 8 章「サーバへのアクセス制御」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がサーバインスタンスのログファイルを共有します。ユーザ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、332 ページの「仮想サーバのログの設定」を参照してください。

仮想サーバの導入

iPlanet Web Server の仮想サーバアーキテクチャは、非常に柔軟性に富んでいます。サーバインスタンスには、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。各待機ソケットには、接続グループを通して任意の数の仮想サーバを関連付けることができます。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を設定できます。

さらに、設定が類似する仮想サーバを、任意の数の仮想サーバクラスにグループ分けすることもできます。仮想サーバクラスに属するすべての仮想サーバは、`obj.conf` 内の同じ要求処理命令を共有します。

仮想サーバごとに、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を柔軟に構成できます。次の例では、iPlanet 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

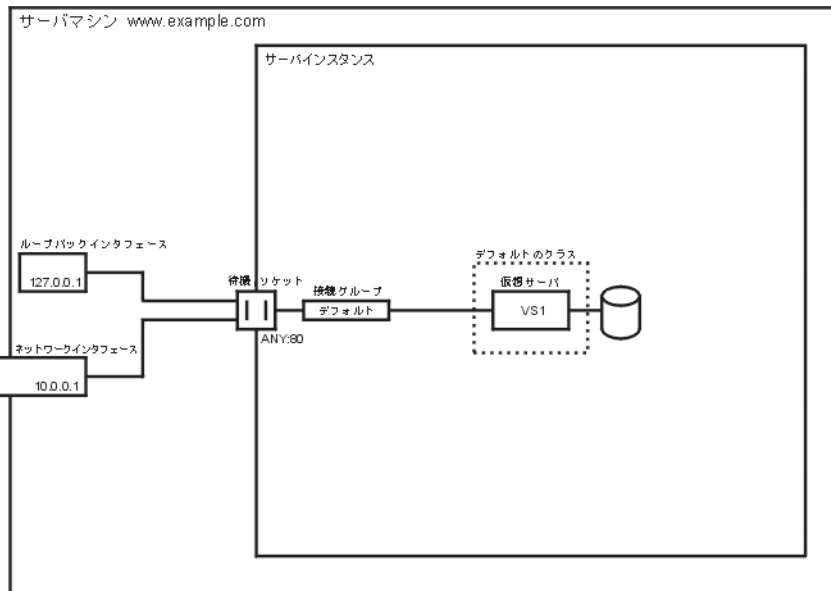
iPlanet Web Server を新規にインストールすると、1つのサーバインスタンスができます。このサーバインスタンスは、コンピュータに設定されているすべての IP アドレスのポート 80 (またはインストール時に選択したポート) で待機する待機ソケットを1つだけ持ちます。

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 で待機するように構成されます (「ANY:80」または「0.0.0.0:80」)。

デフォルトの構成では、IP アドレスベースの仮想サーバは存在しないので、デフォルトの接続グループが唯一の接続グループになります。すべての接続は、仮想サーバ VS1 を通過します。

図 13-3 デフォルトの構成



DNS

| | |
|-----------------|----------|
| www.example.com | 10.0.0.1 |
| | |
| | |
| | |

この構成では、次の場所への接続がサーバに到達し、仮想サーバ 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/

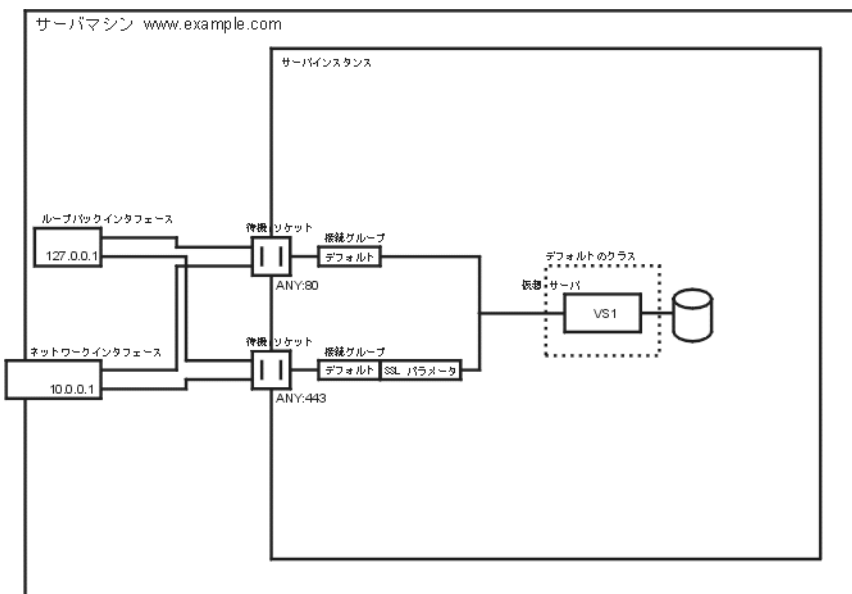
通常の Web サーバの使用法では、この構成を使用します。仮想サーバや待機ソケットをさらに追加する必要はありません。サーバの設定を行うには、defaultclass (VS1 は defaultclass のメンバー) および VS1 そのもので設定を変更します。

例 2：セキュリティ保護されたサーバ

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 iPlanet 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のデフォルトの接続グループに VS1 を関連付けることもできます。その場合、この仮想サーバは 2 つの接続グループを持ち、1 つの接続グループではセキュリティ保護された待機ソケットを使用し、もう 1 つの接続グループではセキュリティ保護されていない待機ソケットを使用します。これにより、サーバは、SSL を使用する場合も使用しない場合も同じコンテンツを提供します。つまり、`http://example.com/` と `https://example.com/` は、同じコンテンツを配信します。

図 13-4 セキュリティ保護されたサーバ



DNS

| | |
|------------------------------|-----------------------|
| <code>www.example.com</code> | <code>10.0.0.1</code> |
| | |
| | |
| | |

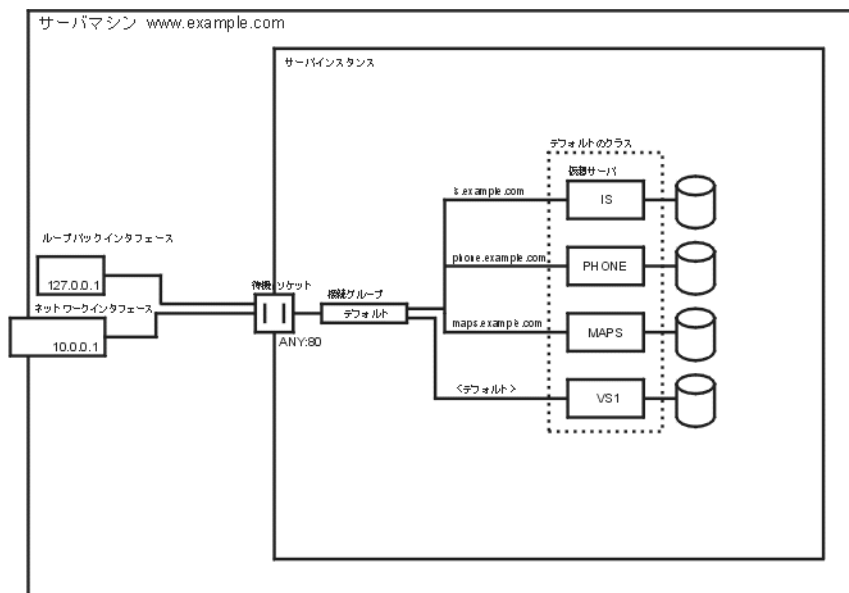
SSL パラメータは接続グループに対して設定します。つまり、1 つの接続グループ内のすべての仮想サーバに対して 1 つの SSL パラメータセットを設定します。

例 3：イントラネットホスティング

さらに複雑な iPlanet Web Server の構成として、イントラネットで、サーバが複数の仮想サーバ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。

ハードウェアおよび管理の間接費を最小限にするために、この3つのサイトを、example.com マシンで稼動する1つの Web サーバに統合します。これには、URL ホストベースの仮想サーバを使用する方法と、IP アドレスベースの仮想サーバを使用する方法があります。両者には、それぞれに長所と短所があります。

図 13-5 URL ホストベースの仮想サーバによるイントラネットホスティング



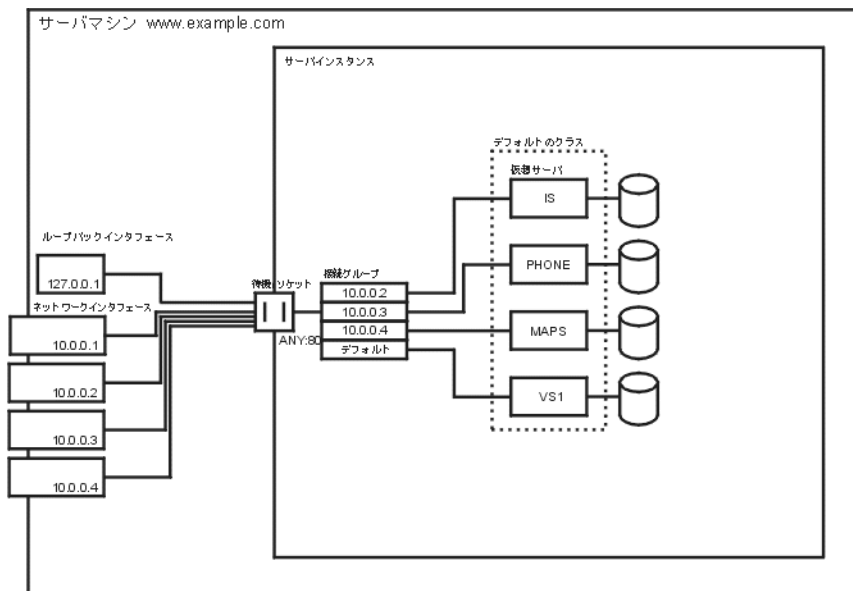
DNS

| | |
|-------------------|----------|
| www.example.com | 10.0.0.1 |
| is.example.com | 10.0.0.1 |
| phone.example.com | 10.0.0.1 |
| maps.example.com | 10.0.0.1 |
| | |

URL ホストベースの仮想サーバ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 5 章「Web サーバのセキュリティ」を参照してください。
- URL ホストベースの仮想サーバは、古いバージョンの HTTP クライアントでは動作しません。

図 13-6 IP アドレスベースの仮想サーバによるイントラネットホスティング



DNS

| | |
|-------------------|----------|
| www.example.com | 10.0.0.1 |
| is.example.com | 10.0.0.2 |
| phone.example.com | 10.0.0.3 |
| maps.example.com | 10.0.0.4 |
| | |

IP アドレスベースの仮想サーバ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。

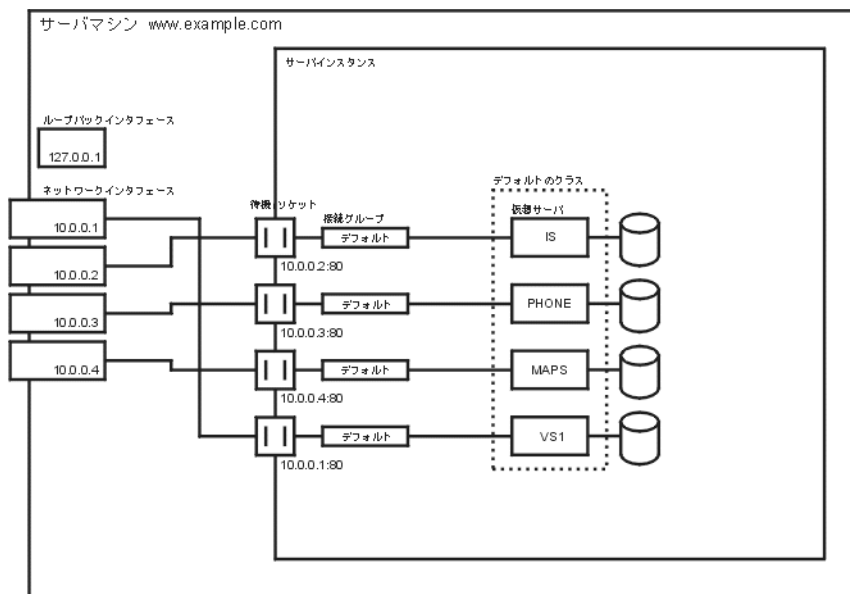
- ホストコンピュータの構成 (実際のネットワークインタフェースまたは仮想ネットワークインタフェースの構成) を変更する必要があります。

- 何千もの仮想サーバ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。

さらに別の方法として、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を設定することもできます。

図 13-7 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



DNS

| | |
|-------------------|----------|
| www.example.com | 10.0.0.1 |
| is.example.com | 10.0.0.2 |
| phone.example.com | 10.0.0.3 |
| maps.example.com | 10.0.0.4 |

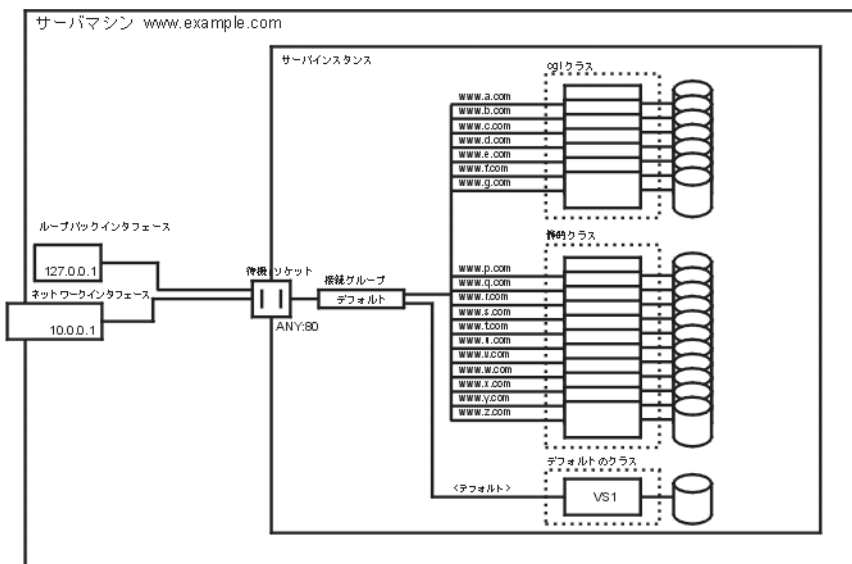
ANY:80の待機ソケットが1つだけ設定されているIPアドレスベースの仮想サーバの構成と比較すると、複数の待機ソケットを持つ構成では、パフォーマンスが少し向上する場合があります。これは、サーバが、要求を受信したアドレスを見つける必要がないためです。ただし、複数の待機ソケットを使用すると、受け入れスレッドが増えるため、オーバーヘッド(メモリおよびスケジューリング)も増えます。

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を使用可能にする構成です。たとえば、低トラフィックの個人ホームページを多数ホスト処理する 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、複数の仮想サーバクラスの 1 つに属します。たとえば、静的コンテンツのみを使用できるクラスと、静的コンテンツと CGI を使用できるクラスを作成できます。

図 13-8 マスホスティング



DNS

| | |
|-----------------|----------|
| www.example.com | 10.0.0.1 |
| www.a.com | 10.0.0.1 |
| www.b.com | 10.0.0.1 |
| www.c.com | 10.0.0.1 |
| ... | |
| www.p.com | 10.0.0.1 |
| ... | |

この場合も、サーバのインストール時にインストールされた仮想サーバ VS1 は、defaultclass に存在しています。

仮想サーバの作成と構成

仮想サーバのクラスには、仮想サーバ(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、個々の仮想サーバを作成して構成する方法について説明します。仮想サーバクラスの構成については、第 16 章「コンテンツ管理」を参照してください。仮想サーバの概要については、第 13 章「仮想サーバの使用」を参照してください。

この章は、次の節で構成されます。

- 仮想サーバの作成
- 仮想サーバの設定内容の変更
- Virtual Server Manager を使用した変更
- Class Manager を使用した変更
- 仮想サーバの削除

仮想サーバの作成

仮想サーバを使用すると、インストールされた 1 つのサーバで、複数の会社または個人に対して、ドメイン名、IP アドレス、およびサーバ監視機能を提供できます。仮想サーバの紹介および iPlanet Web Server での仮想サーバの設定方法については、第 13 章「仮想サーバの使用」を参照してください。

仮想サーバを作成するには、次の手順に従います。

1. Class Manager 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. 仮想サーバの名前を選択します。
4. 仮想サーバの接続グループを選択します。

5. 仮想サーバの 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。
6. 「OK」をクリックします。

仮想サーバ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の設定内容を構成できます。

仮想サーバの設定内容の変更

仮想サーバを設定したあと、設定内容を変更できます。仮想サーバの設定を変更する方法は 2 とおりあります。1 つは **Class Manager** を使用する方法で、もう 1 つは **Virtual Server Manager** を使用する方法です。

Class Manager 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。

Virtual Server Manager では、ページは 1 つの仮想サーバだけに関係します。したがって、その仮想サーバのすべての設定を表示したり変更したりできます。

Virtual Server Manager を使用した変更

Virtual Server Manager には、「Preferences」、「Logs」、および「Web Applications」の 3 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
- ACL ファイル
- MIME タイプファイル

- CGI 設定

1 つの仮想サーバの設定を変更する場合は、Virtual Server Manager 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のレポートを生成するためのページがあります。

「Web Applications」タブには、次のページがあります。

- 「Java Web Applications Settings」
- 「Deploy Web Application」
- 「Edit Web Applications」

「Java Web Applications Settings」ページには、Web アプリケーションが設定されているファイルが表示されます。web-apps.xml がデフォルトのファイルとして設定されます。このページでは、Web アプリケーションの状態をオンまたはオフに設定することもできます。デフォルトではオンに設定されています。Web アプリケーションを導入または編集するには、この設定をオンにする必要があります。

「Deploy Web Application」ページでは、パス、URL、およびディレクトリを指定して、ローカルマシンまたはリモートサーバに WAR ファイルを導入できます。

「Edit Web Applications」ページでは、選択した仮想サーバの WAR ファイルを編集、削除、または無効にすることができます。

Web アプリケーションファイルの導入および編集の詳細は、第 15 章「プログラムによるサーバの拡張」の 342 ページの「ユーザインタフェースでの Web アプリケーションの導入と編集」を参照してください。

仮想サーバのレポートの生成

Virtual Server Manager を使用して、1 つの仮想サーバに関するレポートを生成できます。そのためには、以下に示すように、まず、仮想サーバが使用する新しいアクセスログを作成し、作成したアクセスログを仮想サーバの設定に追加する必要があります。

仮想サーバのレポートを生成するには、次の手順に従います。

1. サーバインスタンスの Server Manager の「Preferences」タブに移動して、「Magnus Editor」を選択します。
2. 「Logging Settings」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。
3. ドロップダウンリストを使用して、LogVsId の値をオンに設定します。

また、手動で LogVsId をオンに設定するには、magnus.conf ファイルに LogVsId on を追加します。

4. 「OK」をクリックします。
5. 「Apply」をクリックします。
6. 変更を有効にするために、「Apply Changes」をクリックします。
7. サーバインスタンスの Server Manager の「Logs」タブに移動して、「Log Preferences」を選択します。
8. 「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。

また、手動で新しいアクセスログを作成するには、`magnus.conf` ファイルを次のように変更します。

```
Init fn=init access="$accesslog" を Init fn=init  
access="newaccesslog" に変更する
```

9. 「Format」の「only log」を選択して、「Vsid」にチェックマークを付けます。
カスタムフォーマットの場合は、「custom format」を選択し、その行の最後に `%vsid%` を追加します。

`%vsid%` は、複数の仮想サーバを使用する場合に役立ちます。このエントリは、アクセスログに `vsid` を記録します。

また、`magnus.conf` ファイルの `Init fn` の後に手動で `%vsid%` を追加することもできます。

10. 「OK」をクリックします。
11. 「Apply」をクリックします。
12. 変更を有効にするために、「Apply Changes」をクリックします。
13. レポートを生成する仮想サーバを選択し、Virtual Server Manager に移動します。
14. 「Preferences」タブに移動し、「Settings」を選択します。

「Access Log」フィールドのアクセスログを、新しく作成したアクセスログに変更します。

15. 「OK」をクリックします。
16. 「Apply」をクリックします。
17. 変更を有効にするために、「Apply Changes」をクリックします。
18. 「Logs」タブを選択します。

「Generate Reports」ページが表示されます。

前述のとおり、仮想サーバが作成済みで、`LogVsId` がオンになっていなければ、このページは表示されません。

19. (省略可能) 必要に応じて設定を変更します。

20. 「OK」をクリックしてレポートを生成します。

Class Manager を使用した変更

次に示す Class Manager の各ページを使用して、仮想サーバの設定を変更します。

仮想サーバの設定内容の変更

仮想サーバ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. Class Manager 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。

デフォルトの仮想サーバは、編集のみ可能で、削除はできません。

4. 「State」を、「On」、「Off」、または「Disable」に設定します。

管理者は仮想サーバの状態を「Disable」に設定しても再びオンに戻すことができませんが、その仮想サーバのエンドユーザはオンにすることができません。

これは仮想サーバの状態であり、サーバ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の状態がオンと表示されている場合、サーバインスタンスもオンのときだけ、その仮想サーバ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インスタンスのデフォルトの仮想サーバについても同じことがいえます。サーバインスタンスをオフにすると、デフォルトの仮想サーバ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。

サーバインスタンスのデフォルトの仮想サーバをオフまたは無効にすることはできません。

5. 「Connections」列のリストから接続グループを選択します。
6. 「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、その URL ホストを入力します。

複数の URL ホストを、空白文字で区切って入力できます。

7. 仮想サーバの変更作業が終了したら、「OK」をクリックします。

仮想サーバの MIME の設定

個々の仮想サーバに 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。

仮想サーバまたは仮想サーバ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 タイプファイルだけを作成し、それを仮想サーバに関連付けます。サーバには 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、または MIME タイプファイル内の定義を編集する場合は、153 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. Class Manager 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. 目的の仮想サーバの横にあるプルダウンメニューから、MIME タイプファイルを選択します。
4. 「OK」をクリックします。

仮想サーバの ACL の設定

ACL を使用して、仮想サーバへのアクセスを制御できます。各仮想サーバは、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、各仮想サーバは、iPlanet 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、200 ページの「仮想サーバへのアクセス制御」を参照してください。

仮想サーバのセキュリティの構成

仮想サーバの接続グループでセキュリティ保護された待機ソケットを使用する場合は、その仮想サーバ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 5 章「Web サーバのセキュリティ」を参照してください。

仮想サーバのサービス品質の設定

サービス品質は、仮想サーバ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。

サーバ全体または仮想サーバクラスに対してこの設定を有効にするには、**Server Manager** の「**Monitor**」タブを使用します。ただし、個々の仮想サーバでは、サーバ全体またはクラスレベルの設定を無効にすることができます。

仮想サーバに対してサービス品質を有効にする前に、まず、サーバ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。217 ページの「サービス品質の使用法」を参照してください。

仮想サーバのサービス品質を設定するには、次の手順に従います。

1. **Class Manager** 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. 仮想サーバのサービス品質を有効にするには、プルダウンリストから「**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、サーバのオーバーヘッドが少し増えます。
4. その仮想サーバ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強制するかどうかを選択します。

最大帯域幅を強制する場合、サーバ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。

最大帯域幅を強制しない場合は、最大帯域幅を超えると、サーバ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強制するかどうかを選択します。

最大接続数を強制する場合、サーバ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。

最大接続数を強制しない場合は、最大接続数を超えると、サーバ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、217 ページの「サービス品質の使用法」を参照してください。

仮想サーバのログの設定

仮想サーバ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. Class Manager 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。

特定の仮想サーバのログを表示するには、次の手順に従います。

1. Virtual Server Manager から、「Status」ページを表示します。
2. アクセスログまたはエラーログのリンクをクリックします。
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まりです。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。クラス内のすべての仮想サーバに対して Web アプリケーションを設定する場合は、Class Manager の「Java Web Applications Settings」ページを使用します。特定の仮想サーバに対して Web アプリケーションを導入または編集する場合は、Virtual Server Manager の「Web Applications」タブを使用します。

Web アプリケーションおよび `web-apps.xml` ファイルの詳細は、第 15 章「プログラムによるサーバの拡張」を参照してください。

仮想サーバの削除

仮想サーバを削除するには、次の手順に従います。

1. Class Manager 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. 目的の仮想サーバの横にあるプルダウンリストから、「Delete」を選択します。
サーバ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は削除できません。また、接続グループのデフォルトの仮想サーバも削除できません。
4. 「OK」をクリックします。仮想サーバが削除されます。

プログラムによるサーバの拡張

この章では、クライアントからの要求に応じて HTML ページを動的に生成するプログラムを iPlanet 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、次の節が記述されています。

- サーバ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 NT CGI プログラムのインストール
- Windows NT でのシェル CGI プログラムのインストール
- 照会ハンドラの使用

サーバ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。次のリストでは、このようなサーバ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の間で情報をやりとりします。

サーバで実行するサーバサイドアプリケーションのタイプ

iPlanet Web Server では、サーバサイドアプリケーションの次のタイプを実行して、コンテンツを動的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、iPlanet Web Server では、サーバ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、『NSAPI プログラマーズガイド』を参照してください。

サーバへのサーバ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導入できます。詳細は、338 ページの「サーバでサーブレットや JSP 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を設定することができます。詳細については、346 ページの「CGI プログラムのインストール」、351 ページの「Windows NT CGI プログラムのインストール」、および 354 ページの「Windows NT 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 サブレットと JavaServer Pages (JSP)

この節では、iPlanet Web Server での Java サブレットおよび 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。

- サブレットと JavaServer Pages の概要
- サーバでサブレットや JSP を実行するための要件
- Web アプリケーションの使用
- web-apps.xml ファイルの使用
- wdeploy を使用した Web アプリケーションの導入
- ユーザインタフェースでの Web アプリケーションの導入と編集
- Web アプリケーションになっていないサブレットと JSP の導入
- JVM 属性の構成
- バージョンファイルの削除

サブレットと JavaServer Pages の概要

iPlanet Web Server 6.0 はサブレット 2.2 API 仕様をサポートします。これによって、Web アプリケーションにサブレットや JSP を組み込むことができます。

Web アプリケーションは、サブレット、JavaServer Pages、HTML ドキュメント、その他の Web リソースの集合です。Web リソースには画像ファイル、圧縮済みアーカイブ、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 | サブレット API のバージョン 2.2 にはバージョン 2.1 との完全な下位互換性があるため、既存のすべてのサブレットを修正や再コンパイルすることなく引き続き使用できます。 |
|----------|--|

サブレットを開発するには、Sun Microsystems の Java サブレット API を使用します。Java サブレット API の使用についての詳細は、Sun Microsystems が提供しているマニュアルを次の Web サイトで参照してください。

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

JSP は HTML ページとよく似たページで、Web ブラウザで表示できます。ただし、HTML タグに加えて、Java コードと組み合わせた JSP タグと指令のセットを指定することができます。これによって、Web ページに動的なコンテンツを取り入れるための機能が拡張されます。この追加機能で、プロパティ値を表示したり、単純な条件を使用したりすることができます。iPlanet Web Server 6.0 は JavaServer Pages (JSP) 1.1 API 仕様をサポートします。

JSP の作成についての詳細は、次に示す Sun Microsystems の JavaServer Pages の Web サイトを参照してください。

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

iPlanet Web Server と一緒に使用するサブレットや JSP の開発についての詳細は、『サブレットに関するプログラマーズガイド』を参照してください。

サーバでサブレットや JSP を実行するための要件

サブレットを有効にするには、Server Manager で「Java」タブを選択し、「Enable/Disable Servlets/JSP」リンクを選択します。「Enable Java Globally」チェックボックスにチェックマークを付け、サーバ全体に対してサブレットを有効にします。「Enable Java for Class」チェックボックスにチェックマークを付け、単一の仮想サーバクラスに対してサブレットを有効にします。Java がグローバルに有効になっていない場合は、クラスに対してサブレットを有効にすることができません。デフォルトでは、Java はグローバルに有効で、各仮想サーバのクラスに対しても有効です。

JSP を有効にするには、まず、サブレットを有効にします。また、web-apps.xml ファイルに enable="true" とともに jsp-servlet 要素を記述し、JVM クラスパスに tools.jar を追加する必要があります。詳細については、『サブレットに関するプログラマーズガイド』を参照してください。

iPlanet Web Server に Java Runtime Environment (JRE) は組み込まれていますが、Java Development Kit (JDK) は組み込まれていません。サーバはサブレットと、JRE を使用してコンパイル済みの JSP を実行できますが、コンパイルされていない JSP を実行するには JDK が必要です。JSP を開発したい場合は、iPlanet Web Server でカスタム JDK を使用する必要があります。

iPlanet Web Server 6.0 では、JDK の正式バージョンを使用する必要があります。『サブレットに関するプログラマーズガイド』で説明されているように、プラットフォームによって異なるバージョンが必要です。

必要な JDK バージョンの更新情報については、iPlanet Web Server の『インストールと移行』および最新のリリースノートで確認してください。

JDK 1.2 (およびその他の JDK バージョン) は、Sun Microsystems の次のサイトから入手できます。

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

JDK へのパスは、次のどちらかの方法で指定します。

- サーバのインストール時にパスを指定する
iPlanet Web Server をインストールするとき、インストールプロセスのダイアログボックスの 1 つで、カスタム Java Development Kit (JDK) を使用するかどうかを尋ねられます。使用する場合は、パスを指定できます。
- サーバをインストールしたあとでパスを指定する
JDK へのパスを指定するには、Web Server Administration Server に切り替え、「Global Settings」タブを選択し、63 ページの「JRE/JDK パスの構成」で説明されているように「Configure JRE/JDK Paths」ページを使用します。また、このページで JDK へのパスを変更することもできます。

JDK へのパスをインストール時に指定する場合も、インストール後に指定する場合も、パスは、JDK をインストールしたフォルダです。

Web アプリケーションの使用

次の節では、wdeploy コマンド行ユーティリティまたはユーザインタフェースを使用して、Web アプリケーションを導入、編集、および削除する方法について説明します。

web-apps.xml ファイルの使用

Web アプリケーションを導入するには、その前に iPlanet Web Server に固有の web-apps.xml ファイルを修正する必要があります。各仮想サーバには、固有の web-apps.xml ファイルがあります。このファイルは、仮想サーバで実行する Web アプリケーションのコンテキストを定義します。コンテキスト情報には、Web アプリケーションのコンテキストパス、セッション管理や認証の処理方法などのその他のプロパティが含まれます。

Web アプリケーションは導入されると、デフォルトで有効になります。導入された Web アプリケーションを手動で無効にするには、web-app.xml ファイルを次のように修正する必要があります。

```
<vs>
<web-app uri="/mywebapp" dir="/webappdir" enable = "false" >
</web-app>
```

複数の Web アプリケーションに同じ説明を付けて導入したり編集したりした場合、そのいずれかのアプリケーションが無効になっていると、サーバは `enable = "false"` を無視し、デフォルト設定の `enable = "true"` を使用します。

`web-apps.xml` ファイルについての詳細は、『サブレットに関するプログラマーズガイド』を参照してください。

wdeploy を使用した Web アプリケーションの導入

Web アプリケーションを手動で導入するには、パスに `server_root/bin/https/httpadmin/bin` ディレクトリが含まれていることと、`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する必要があります。

コマンド行で `wdeploy` ユーティリティを使用して、WAR ファイルを仮想サーバの Web アプリケーション環境に導入することができます。

```
wdeploy deploy -u uri_path -i instance -v vs_id [-d directory] war_file -n
```

仮想サーバの Web アプリケーションを削除することができます。

```
wdeploy delete -u uri_path -i instance -v vs_id hard|soft -n
```

仮想サーバの Web アプリケーション URI とディレクトリのリストを表示できます。

```
wdeploy list -i instance -v vs_id
```

コマンドパラメータには、次の意味があります。

| | |
|--------------------------|--|
| <code>uri_path</code> | Web アプリケーションの URI 接頭辞 |
| <code>instance</code> | サーバのインスタンス名 |
| <code>vs_id</code> | 仮想サーバの ID |
| <code>idirectory</code> | (省略可能) アプリケーションが導入されるディレクトリ、またはアプリケーションが削除されるディレクトリ。導入されるディレクトリが指定されていない場合、アプリケーションはドキュメントルートディレクトリに導入される |
| <code>hard soft</code> | ディレクトリと <code>web-apps.xml</code> エントリが削除されるか (<code>hard</code>)、 <code>web-apps.xml</code> エントリだけが削除されるか (<code>soft</code>) を指定する |
| <code>war_file</code> | WAR ファイル名 |

注意 Web アプリケーションを導入する場合に *directory* を指定しないと、アプリケーションはドキュメントルートディレクトリに導入されます。そのあと、*hard* パラメータを使用してアプリケーションを削除する場合は、ドキュメントルートディレクトリが削除されます。

wdeploy deploy コマンドを実行すると、次の 3 つの内容が実行されます。

- 指定した *uri_path* と *directory* とともに Web アプリケーションが、web-apps.xml ファイルに追加される
- WAR ファイルがターゲット *directory* に抽出される
- サーバが動的に再構成されて、新しい Web アプリケーションがロードされる

次に例を示します。

```
wdeploy deploy -u /hello -i server.iplanet.com -v acme.com
-d /iws60/https-server.iplanet.com/acme.com/web-apps/hello
/iws60/plugins/servlets/examples/web-apps/HelloWorld/HelloWorld.war
```

このユーティリティの結果は、次の web-apps.xml エントリとなります。

```
<vs>
  <web-app uri="/hello"
    dir="/iws60/https-server.iplanet.com/acme.com/web-apps/hello"/>
</vs>
```

/iws60/https-server.iplanet.com/acme.com/web-apps/hello ディレクトリには、次のコンテンツがあります。

```
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

iPlanet Web Server 6.0 SP1 では、Web アプリケーションの導入や削除を行ったあと、その Web アプリケーションの読み込みや読み込み解除を行うように wdeploy でサーバを動的に再構成します。変更も反映するには、次のいずれかを事前に実行して、サーバを明示的に再構成しておく必要があります。

- 再構成スクリプトを使用する
- サーバを再起動する
- 管理ユーザインタフェースの「Apply」リンクをクリックする

wdeploy コマンドの実行に成功すると自動的に、新しい Web アプリケーションへの要求に対するサービス提供が有効になるか、削除された Web アプリケーションへの要求に対するサービス提供が停止されます。

-n オプションを指定すると、wdeploy によって再構成コマンドが Web サーバに自動的に送信されることを防止できます。スクリプトなどで、複数の Web アプリケーションの導入または導入解除を行う場合で、最後の Web アプリケーションの導入後に一度だけサーバを再起動したいときに、コマンドに -n オプションを指定します。

導入された Web アプリケーションへのアクセス

アプリケーションを導入したあと、次のようにブラウザからそのアプリケーションにアクセスできるようになります。

```
http://vs_urlhost[:vs_port]/uri_path/[index_page]
```

URL の各部には、次の意味があります。

| | |
|-------------------|---|
| <i>vs_urlhost</i> | 仮想サーバの <code>urlhosts</code> 値の 1 つ |
| <i>vs_port</i> | (省略可能) 仮想サーバでデフォルト以外のポートを使用する場合にだけ必要 |
| <i>uri_path</i> | アプリケーションを導入するために使用した URI と同じ。これは、コンテキストパスでもある |
| <i>index_page</i> | (省略可能) エンドユーザがアプリケーション内で最初にアクセスするページ |

次に例を示します。

```
http://acme.com:80/hello/index.jsp
```

または

```
http://acme.com/hello/
```

ユーザインタフェースでの Web アプリケーションの導入と編集

iPlanet Web Server 6.0 SP1 では、指定された仮想サーバに対して Web アプリケーションの導入、編集、削除、無効化、および有効化を実行できます。

Web アプリケーションの導入

最初に、Virtual Server Manager の「Web Applications」タブの下の「Java Web Applications Settings」リンクを選択し (手順の 1)、次に、「Deploy Web Application」リンクを選択して「Deploy Web Application」ページにアクセスします (手順の 2 以降)。

Web アプリケーションを導入するには、次の手順に従います。

1. 「Web Applications File」フィールドに、Web アプリケーションファイルへのパスを入力します。デフォルトの仮想サーバでは、`web-apps.xml` が設定されています。パスは、そのインスタンスの構成ディレクトリ内の絶対パスまたはファイル名にすることができます。入力したファイル名が存在しない場合は、新規に作成されます。
2. 「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。
WAR ファイルを自分のサーバにアップロードする場合は、「Local Machine」を選択します。WAR ファイルがすでにサーバのマシンに存在する場合は、「Server Machine」を選択します。
3. 「WAR File Path」フィールドに、Web アプリケーションを格納している WAR ファイルへの、ローカルマシンまたはサーバマシン上のパスを入力します。
サーバマシンの場合は、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、使用可能なパスを参照できます。「参照 (browse)」をクリックすると、「ファイルアップロード」ウィンドウが表示されて、サーバにアップロードする WAR ファイルを選択できるようになります。
4. Application URI フィールドに、仮想サーバ上における、Web アプリケーションの URI を入力します。
5. WAR ファイルのコンテンツの抽出先である、サーバマシン上のディレクトリへの絶対パスを入力します。入力したディレクトリが存在しない場合は、新規に作成されます。
6. 「OK」をクリックします。
7. 「Apply」をクリックします。
8. 導入する Web アプリケーションに対して「Dynamic Reconfiguration」を選択します。

Web アプリケーションの編集

すでに導入されている Web アプリケーションの編集、削除、無効化、または有効化を実行できます。Virtual Server Manager の「Web Applications」タブの下の「Edit Web Applications」を選択し、「Edit Web Applications」ページにアクセスします。

すでに導入されている Web アプリケーションの編集、削除、無効化、または有効化を実行するには、次の手順に従います。

1. 編集する Web アプリケーションの横の「Action」列で、ドロップダウンリストから実行するアクションを選択します。
 - 「Edit」 - Web アプリケーションにアクセスするための URI を変更する
 - 「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、そのアプリケーションが導入されているディレクトリを削除する
 - 「Disable」 - URI からその Web アプリケーションにアクセスできなくなるようにする。ただし、削除されない
 - 「Enable」 - 無効にしたアプリケーションを再び有効にする

注意 Web アプリケーションを削除すると、そのアプリケーションが導入されているディレクトリも削除されることに注意してください。

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. 導入を行なう Web アプリケーションに対して「Dynamic Reconfiguration」を選択します。

Web アプリケーションになっていないサーブレットと JSP の導入

Web アプリケーションになっていない 4.x のサーブレットや JSP を導入できますが、デフォルトの仮想サーバ内だけに導入できます。詳細は、『サーブレットに関するプログラマーズガイド』を参照してください。

JVM 属性の構成

Server Manager の「Java」タブの「Configure JVM Attributes」ページで、Java 仮想マシン (Java Virtual Machine、JVM) の属性を構成できます。

これらのオプションについての詳細は、『サーブレットに関するプログラマーズガイド』を参照してください。

バージョンファイルの削除

Server Manager 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 ディレクトリの下に JSP クラスキャッシュに保存します。

サーバは JavaServer Pages (JSP) とサ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

サーバは JavaServer Pages (JSP) とサブレットの情報をキャッシュに書き込むために、次のディレクトリを使用します。

```
server_root/https-server_id/ClassCache/virtual_server_id/webapp_uri/
```

サーバは「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 ディレクトリの下に JSP クラスキャッシュに保存します。

- 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、キャッシュの中身を消去できます。

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。次に「JSP」ページを処理するときにサーバは、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 NT 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 NT CGI プログラムのインストール
- Windows NT 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。サーバを停止すると、これらのプロセスは消滅します。

Windows NT 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 NT 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と連動して動作します。このようなプログラムは、Windows NT CGI プログラムと呼ばれます。Windows NT CGI についての詳細は、351 ページの「Windows NT 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。

- iPlanet Web Server の『プログラマーズガイド』
- 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://www.iplanet.com/docs>

サーバマシンに 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。同じ様に、ファイルタイプの方法を選択した場合、サーバ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、Server Manager 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. Class Manager 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 接頭辞を入力します。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 接頭辞として「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 接頭辞は、次の手順で指定する実際の 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 接頭辞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 接頭辞または 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。

各仮想サーバに固有の CGI 属性を構成する

単一の仮想サーバに CGI 属性を指定するには、次の手順を実行します。

1. Class Manager で仮想サーバを選択し、「Manage」ボタンをクリックします。
2. Virtual Server Manager の「Preferences」タブで「Settings」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、サーバは `nice` 値 0 で稼働し、`nice` 値の増分は 0 (CGI プログラムはサーバと同じ優先度で動作) ~ 19 (CGI プログラムはサーバよりかなり低い優先度で動作) 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よりも優先することは可能ですが、この方法はお勧めしません。
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. Class Manager 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。Server Manager 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/kb/server/960513-130.html>

Windows NT CGI プログラムのインストール

この節では、Windows NT 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 NT CGI プログラムの概要
- Windows NT CGI ディレクトリの指定
- ファイルタイプとして Windows NT CGI を指定

Windows NT CGI プログラムの概要

Windows NT CGI プログラムは、ほかの CGI プログラムと同様に処理されます。Windows NT 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 NT 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 NT CGI プログラムを格納するディレクトリを作成し、Windows NT CGI ファイルの拡張子を 1 つ指定します。

Windows NT 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 NT 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 NT 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。

iPlanet Web Server は次の点を除いて、Windows NT 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、この値はサーバの非公開鍵の生成に使用されるビット数を示す
- サーバ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 NT 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 NT CGI ディレクトリの指定

Windows NT 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. Class Manager 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 接頭辞を入力します。

つまり、入力したテキストが Windows NT 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 接頭辞として「wcgi-programs」と入力する場合、Windows NT CGI プログラムへの URL はすべて次の構造になります。

`http://yourserver.domain.com/wcgi-programs/program-name`

注 指定した URL 接頭辞は、手順 5 で指定する実際の Windows NT 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 パラメータがサーバから Windows NT CGI プログラムへとファイルを介して渡されます。このファイルは通常、Windows NT CGI プログラムを実行したあとサーバ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 NT CGI プログラムで開いたウィンドウは表示されたままになります。
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 接頭辞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 NT CGI ディレクトリを削除するには、「CGI Directory」フォームで Windows NT 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 接頭辞または Windows NT 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 NT 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 NT 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 NT CGI を指定

Windows NT 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. Server Manager 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 の詳細については、153 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: サーバに関連付けたい Windows NT CGI のファイル接尾辞を入力します。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接尾辞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接尾辞 `.exe` を使用することはできません。必要に応じて、接尾辞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 NT でのシェル CGI プログラムのインストール

この節では、Windows NT 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 NT でのシェル CGI プログラムの概要
- シェル CGI ディレクトリの指定 (Windows NT)
- ファイルタイプとしてシェル CGI を指定 (Windows NT)

Windows NT でのシェル CGI プログラムの概要

シェル CGI は Windows NT のファイル関連付けを使用して CGI アプリケーションを実行できるサーバ構成です。

たとえば、サーバは `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 NT 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 を構成するには、サーバ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、iPlanet 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を構成することもできます。

注 Windows NT でのファイル拡張子の設定方法については、Windows NT 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 NT)

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. Server Manager の「Class Manager」タブを選択します。

3. 次に、「Programs」タブを選択します。
「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 接頭辞を入力します。
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 接頭辞として「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。

注意 サーバには、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 NT の場合は、サーバを実行するユーザアカウント (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 NT のファイル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、サーバ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 NT)

iPlanet 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に関連付けることができます。これは、Windows NT での関連付けの作成とは異なります。

サーバのシェル CGI 機能とファイル拡張子に関連付ける場合、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。サーバは、この拡張子を持つファイルの要求を受けると、Windows NT 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. Server Manager 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、153 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: サーバを関連付けたいシェル CGI のファイル接尾辞を入力します。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接尾辞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接尾辞 .exe を使用することはできません。必要に応じて、接尾辞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。

照会ハンドラの使用

注 照会ハンドラは現在、使用されなくなっています。iPlanet 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照会を送信する方法がより一般的です。

デフォルトの照会ハンドラ CGI プログラムを指定できます。照会ハンドラは、HTML ファイル内の ISINDEX タグによって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、ユーザが Enter キーを押すとすぐに送信されます。デフォルトの照会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。

照会ハンドラを設定するには、次の手順を実行します。

1. Class Manager 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、デフォルトの照会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照会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。

コンテンツ管理

この章では、仮想サーバのクラスと仮想サーバについて、コンテンツを構成して管理する方法を説明します。

この章は、次の節から構成されています。

- 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
- ドキュメント設定の構成
- URL 転送の構成
- 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
- サーバが解析する HTML の設定
- キャッシュ制御指令の設定
- Stronger Ciphers の使用

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ぶ) 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。Class Manager 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、`$id` 変数を使用して、クラス内のすべての仮想サーバに対して、仮想サーバの ID を名前に使用したディレクトリを作成することができます。クラスのドキュメントルートを `class_doc_root/$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 `/iplanet/servers/docs/$id` の場合、そのクラスに属している仮想サーバ `vs1` のデフォルトのドキュメントディレクトリは `/iplanet/servers/docs/vs1` です。

ドキュメントディレクトリと、サーバインスタンス、クラス、および仮想サーバ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、304 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. Class Manager 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末尾に変数 `$id` を組み込む場合、すべての仮想サーバのデフォルトのドキュメントルートが `class_doc_root/virtual_server_ID` になります。たとえば、クラスのドキュメントディレクトリが `/iplanet/servers/docs/$id` の場合、クラスに属している仮想サーバ `vs1` のデフォルトドキュメントディレクトリは `/iplanet/servers/docs/vs1` です。

変数の詳細については、309 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。

注 通常、各仮想サーバには固有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、またはサーバ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があります。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によって使用されます。

クラス内の個々の仮想サーバ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 接頭辞がマッピングされるディレクトリを仮想サーバごとに変えるため、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. Class Manager 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 接頭辞を選択します。
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、既存の構成スタイルを使用し、このディレクトリの構成方法を指定します。
6. 「OK」をクリックします。

詳細は、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、次の接頭辞が付いています。

- /manual
- /servlet

これらのディレクトリへのアクセスを制限して、ユーザが書き込めないようにする必要があります。サンプル ACL は次のとおりです。

```
deny (all) anyone;  
allow (rxli) all;  
allow (wd) privileged_user;
```

ユーザ公開情報ディレクトリのカスタマイズ (UNIX/Linux)

ユーザが独自の Web ページを維持管理する場合もあります。サーバ上のすべてのユーザが、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構成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に認識されている特定の URL を使用してサーバにアクセスできます。たとえば、接頭辞 ~ とディレクトリ `public_html` を選択するとします。

`http://www.iplanet.com/~jdoe/aboutjane.html` が要求された場合、サーバは ~jdoe がユーザの公開情報ディレクトリを参照していると認識します。サーバはシステムのユーザデータベースの `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を構成するには、次の手順に従います。

1. Class Manager 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. ユーザの URL 接頭辞を選択します。

チルド文字がユーザのホームディレクトリにアクセスするための標準の UNIX/Linux 接頭辞であるため、通常の接頭辞は ~ です。

4. サーバが HTML ファイルを検索する、ユーザ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。

サーバでは、システム上のユーザ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。サーバはこのファイルを使用して、ユーザ名が有効であるかどうかを判断し、そのユーザ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、サーバは標準のライブラリコールを使用してユーザを検索します。あるいは、別のユーザファイルを作成して、ユーザを検索することもできます。ユーザファイルを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、363 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. 構成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アカウントを制限したい場合もあります。ユーザによる発行を制限するには、`/etc/passwd` ファイルのユーザのホームディレクトリのパスに最後の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、iPlanet Web Server はこのユーザ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

また、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、サーバは起動時にパスワードファイルをメモリに読み込んで、ユーザの検索速度を大きく向上させます。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。

構成スタイルの使用

サーバに構成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がシンボリックリンクを作成するのを防止できます。構成ファイルの詳細については、第 17 章「構成スタイルの適用」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をサーバユーザ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. Class Manager 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。

ドキュメント設定の構成

「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、次の内容を説明します。

- 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を選択
- サーバのホームページの指定
- デフォルト MIME タイプの指定
- Accept-Language ヘッダーの解析

これらの設定はすべて、個々の仮想サーバではなく、クラスに対して構成されます。

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. Class Manager 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を選択

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ります。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多くの場合に便利です。

サーバは `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、維持管理するファイルです。詳細は、365 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。

| | |
|-----------|--|
| 注意 | サーバ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。 |
|-----------|--|

サーバのホームページの指定

エンドユーザが最初にサーバ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時、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

Accept-Language ヘッダーの解析

クライアントが HTTP 1.1 を使用してサーバに接続する場合、受け入れる言語を説明したヘッダー情報を送信できます。この言語情報を解析するようにサーバを構成できます。

たとえば、ドキュメントを日本語または英語で保存する場合、Accept-Language ヘッダーを解析するように選択できます。Accept-Language ヘッダーとして日本語が設定されたクライアントがサーバに接続する場合、日本語版のページを受信します。Accept-Language ヘッダーとして英語が設定されたクライアントがサーバに接続する場合、英語版のページを受信します。

複数の言語をサポートしていない場合は、Accept-Language ヘッダーを解析する必要はありません。

Accept-Language ヘッダー使用についての詳細は、412 ページの「Accept-Language ヘッダーの使用」節を参照してください。

URL 転送の構成

URL 転送を使用すると、ドキュメント要求を別のサーバ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、サーバがユーザに 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サーバに移動した場合）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、あるサーバのドキュメントを要求するユーザをスムーズに別のサーバ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.iplanet.com/info/movies` を接頭辞 `film.iplanet.com` に転送する場合、`http://www.iplanet.com/info/movies` という URL は `http://film.iplanet.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、309 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 の接頭辞を <http://www.iplanet.com/explain.html> にリダイレクトできます。

URL 転送を構成するには、次の手順に従います。

1. Class Manager 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 接頭辞を入力し、リダイレクト先を別の接頭辞にするか、または静的な URL にするかを指定します。
4. 「OK」をクリックします。

詳細は、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、Class Manager 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. Class Manager 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「The entire server」を選択し、クラス全体に対して変更を適用するか、特定の仮想サーバに対するドキュメントルートまたは特定の仮想サーバ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-8859-1`
- `iso-2022-jp`
- `x-sjis`
- `x-euc-jp`
- `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
- iso_8859-1:1987
- iso-ir-100
- ibm819
- cp819

文字セットを変更するには、次の手順に従います。

1. Class Manager 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「The entire server」を選択し、クラス全体に対して変更を適用するか、特定の仮想サーバに対するドキュメントルートまたは特定の仮想サーバ内の特定のディレクトリを指定します。
4. サーバ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. Class Manager 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「The entire server」を選択し、クラス全体に対して変更を適用するか、特定の仮想サーバに対するドキュメントルートまたは特定の仮想サーバ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の最大文字数は 765 文字です。ドキュメントが最後に修正された日付を挿入する場合は、「:LASTMOD:」という文字列を入力します。

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、190 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ポイントする 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、元のファイルをポイント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. Class Manager 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「The entire server」を選択し、クラス全体に対して変更を適用するか、特定の仮想サーバに対するドキュメントルートまたは特定の仮想サーバ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、両方を有効にすることを選択し、開始ディレクトリを選択します。
5. 「OK」をクリックします。

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

サーバが解析する 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、サーバに介入されずに、クライアントに送信されます。ただし、サーバ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. Class Manager 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「The entire server」を選択し、クラス全体に対して変更を適用するか、特定の仮想サーバに対するドキュメントルートまたは特定の仮想サーバ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` タグは有効にしないで 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。

解析される HTML を受け入れるためのサーバ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。

サーバが解析する HTML の使用についての詳細は、『プログラマーズガイド』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、プロキシサーバによってキャッシュに保存される情報を iPlanet 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、デフォルトのプロキシのキャッシングがオーバーライドされ、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. Class Manager 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. フィールドに必要な事項を入力します。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。

Stronger Ciphers の使用

Stronger Ciphers の設定についての詳細は、125 ページの「Stronger Ciphers を設定する」を参照してください。

構成スタイルの適用

構成スタイルは、さまざまな仮想サーバ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構成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構成スタイルを適用すれば、仮想サーバ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構成する必要はありません。

この章は、次の節から構成されています。

- 構成スタイルの作成
- 構成スタイルの割り当て
- 構成スタイルの割り当ての一覧表示
- 構成スタイルの編集
- 構成スタイルの削除

構成スタイルの作成

構成スタイルを作成するには、次の手順を実行します。

1. Class Manager 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. 構成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。
iPlanet Web Server によって「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、編集する構成スタイルを選択し、「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、スタイルを構成するカテゴリをクリックします。

表 17-1 にリストされている情報を構成できます。

7. 表示されるフォームに必要な事項を入力し、「OK」をクリックします。
8. 構成スタイルのほかの構成を変更するには、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構成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「iPlanet Web Server の概要」の 42 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 構成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の拡張」の 346 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。 |
| 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 369 ページの「文字セットの変更」を参照してください。 |
| Default Query Handler (デフォルトの照会ハンドラ) | サーバリソースに対してデフォルトの照会ハンドラを設定できます。照会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の拡張」の 356 ページの「照会ハンドラの使用」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。 |
| Dynamic Configuration (動的構成) | Server Manager へのアクセス権を与えることなく、ほかのユーザに構成オプションのサブセットを与えることができるようになります。 |
| Error Responses (エラー応答) | サーバ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、カスタマイズできるようになります。 |
| Log Preferences (ログ設定) | アクセスログの詳細を設定できるようになります。ログの詳細設定については、第 9 章「ログファイルの使用」の 209 ページの「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) | クライアントが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、サーバ上のファイルの名前変更などを実行できるようにします。 |

表 17-1 構成スタイルのカテゴリ (続き)

| カテゴリ | 説明 |
|---|---|
| Require Stronger Security (より強力なセキュリティを 要求) | より強力なセキュリティ要件を要求できます。 |
| Restrict Access (アクセスの制限) | サーバ全体またはサーバの一部へのアクセスを制限できます。アクセス制御の詳細については、第 8 章「サーバへのアクセス制御」を参照してください。 |
| Server Parsed HTML (サーバが解析する HTML) | ファイルがクライアントに送信される前に、サーバが解析するかどうかを指定できます。 |

詳細については、オンラインヘルプの「[「Create a New Style」ページ](#)」を参照してください。

構成スタイルの割り当て

構成スタイルを作成すると、仮想サーバ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。

構成スタイルを割り当てるには、次の手順を実行します。

1. Class Manager 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. この構成スタイルを適用する URL の接頭辞を入力します。

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、構成スタイルがそのディレクトリ内のすべてのファイルに適用されます。

5. 適用する構成スタイルを選択します。

以前にリソースに適用した構成スタイルを削除するには、「None」構成スタイルを適用します。「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「[「Assign a Style」ページ](#)」を参照してください。

構成スタイルの割り当ての一覧表示

構成スタイルを作成し、ファイルやディレクトリに適用すると、構成スタイルとその適用先の一覧を表示できます。

構成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. Class Manager 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。

iPlanet Web Server によって、サーバーリソースに適用された構成スタイルを示す「List Assignments」ページが表示されます。

4. 構成スタイルの割り当てを編集するには、構成スタイル名の隣にある「Edit」リンクをクリックします。

詳細については、オンラインヘルプの「「List Assignments」ページ」を参照してください。

構成スタイルの編集

構成スタイルを編集するには、次の手順を実行します。

1. Class Manager 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. 編集する構成スタイルを選択し、「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、スタイルを構成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、375 ページの「構成スタイルの作成」節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. 構成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構成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「iPlanet Web Server の概要」の 42 ページの「リソースピッカーの使用」を参照してください。

詳細については、オンラインヘルプの「[「Edit a Style」ページ](#)」を参照してください。

構成スタイルの削除

構成スタイルを削除する前に、構成スタイルが適用された割り当てを削除します。構成スタイルを削除する前にこれを実行しない場合は、仮想サーバのクラスの `obj.conf` ファイルを手動で編集し、ファイル内の構成スタイルを検索し、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、適用されていた構成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ユーザは、サーバの構成が間違っているというエラーメッセージを受信します。

構成スタイルを削除するには、次の手順を実行します。

1. **Class Manager** 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。

詳細については、オンラインヘルプの「[「Remove a Style」ページ](#)」を参照してください。

構成スタイルの削除

付録 A 「コマンド行ユーティリティ」

付録 B 「HTTP (HyperText Transfer Protocol)」

付録 C 「ACL ファイルの構文」

付録 D 「国際化された iPlanet Web Server」

付録 E 「Microsoft FrontPage のサーバ拡張機能」

コマンド行ユーティリティ

この付録では、ユーザ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を説明します。

この付録は、次の節で構成されています。

- LDIF エントリの書式設定
- HttpServerAdmin (仮想サーバの管理)

LDIF エントリの書式設定

LDIF は、空白行で区切られた 1 つ以上のディレクトリエントリで構成されます。各 LDIF エントリはオプションエントリ ID、要求される識別名、1 つ以上のオブジェクトクラス、複数の属性定義で構成されます。

LDIF エントリの書式設定についての詳細は、

<http://docs.ipplanet.com/docs/manuals/directory.html> から iPlanet Directory Server 5.0 の『構成、コマンド、およびファイルのリファレンス』または『Netscape Schema Reference, Directory Server 4.0』を参照してください。

ldapmodify を使用したデータベースエントリの修正

ldapmodify コマンド行ユーティリティを使用して、既存のディレクトリサーバデータベース内のエントリを修正します。ldapmodify では、入力された識別名とパスワードを使用して、指定されたサーバへの接続を開き、指定されたファイルに含まれている LDIF 更新文に基づいてエントリを修正します。ldapmodify は LDIF 更新文を使用しているため、ldapmodify は ldapdelete で実行できることをすべて実行できます。

ディレクトリサーバのデータベースエントリで使用されるコマンド行ユーティリティについての詳細は、<http://docs.iplanet.com/docs/manuals/directory.html> の iPlanet Directory Server 5.0 の『構成、コマンド、およびファイルのリファレンス』または『Netscape Schema Reference, Directory Server 4.0』を参照してください。

HttpServerAdmin (仮想サーバの管理)

HttpServerAdmin は、Server Manager と Class Manager で の仮想サーバのユーザ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を設定する場合は、HttpServerAdmin を使用します。

HttpServerAdmin は `server_root/bin/https/httpadmin/bin` 内にあります。

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/iplanet/servers
```

Windows NT 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 に表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 |
|-----------------------------------|---|
| <code>-d server_root</code> | (必須)。このパラメータはサーバルート (サーバがインストールされている場所) へのパスを指定する |
| <code>-sinst http_instance</code> | (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の開始、停止、無効化を実行します。仮想サーバを指定しない場合、クラス内のすべての仮想サーバ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 |
|---------------------|---|
| <code>-start</code> | 指定された仮想サーバを開始する。仮想サーバ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を開始する |
| <code>-stop</code> | 指定された仮想サーバを停止する。仮想サーバ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を停止する |

表 A-2 Control コマンドのオプション

| オプション | 値 |
|----------|---|
| -disable | 指定された仮想サーバを無効化する。仮想サーバ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を無効化する |

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 <i>classname</i> | 仮想サーバクラスを指定する |
| -id <i>virtual_server</i> | (省略可能) 制御している仮想サーバの ID を指定する |

例

```
HttpServerAdmin control -cl myclass -start -id myvirtualserver -d
/usr/iplanet/servers -sinst https-iplanet.com
```

```
HttpServerAdmin control -cl myclass -stop -id myvirtualserver -d
/usr/iplanet/servers -sinst https-iplanet.com
```

```
HttpServerAdmin control -cl myclass -disable -id myvirtualserver
-d /usr/iplanet/servers -sinst https-iplanet.com
```

create コマンド

create コマンドを使用して、仮想サーバのクラス、仮想サーバ、待機ソケット、接続グループを作成します。

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、接続グループ、待機ソケット、および仮想サーバを作成できます。

表 A-4 create コマンドのオプション

| オプション | 値 |
|-------|---------------|
| -c | 仮想サーバクラスを作成する |
| -g | 接続グループを作成する |
| -l | 待機ソケットを作成する |
| -v | 仮想サーバを作成する |

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 仮想サーバクラス作成のパラメータ

| パラメータ | 値 |
|-------------------------------|--------------------------------------|
| -cl <i>classname</i> | 作成するクラス名 |
| -docroot <i>document_root</i> | (省略可能) 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 |

表 A-5 仮想サーバクラス作成のパラメータ (続き)

| パラメータ | 値 |
|-------------------------------------|--|
| -obj <i>obj.conf_file</i> | (省略可能) クラスの <i>obj.conf</i> ファイル。このパラメータを指定しない場合、サーバは <i>classname.obj.conf</i> という名前で <i>obj.conf</i> ファイルを作成する。クラスの <i>obj.conf</i> ファイルに別の名前を付ける場合は、ここで指定する |
| -acptlang <i>accept_language</i> | (省略可能) このパラメータを指定しない場合、 <i>acptlang</i> はデフォルトでオフ |

例

```
HttpServerAdmin create -c -cl myclass1 -d /export/iplanet/servers
-sinst https-iplanet.com
```

接続グループの作成

`create` コマンドのこのオプションを使用して、接続グループを作成します。

構文

```
HttpServerAdmin create -g group_ID -lsid listen_socket -ip IPaddress -sname
server_name -defaultvs default_virtual_server -d server_root -sinst http_instance
```

パラメータ

表 A-6 に示すパラメータを `create -g` コマンドオプションとともに使用し、接続グループを作成します。

表 A-6 接続グループ作成のパラメータ

| パラメータ | 値 |
|---|---|
| -g <i>connection_group</i> | 作成する接続グループの ID |
| -lsid <i>listen_socket</i> | この接続グループに関連付けられる待機ソケットの ID |
| -ip <i>IP_address</i> | この接続グループに関連付けられた IP アドレス |
| -sname <i>server_name</i> | サーバ名 |
| -defaultvs <i>default_virtual_server</i> | 要求された URL ホストが見つからない場合、接続グループが接続するデフォルトの仮想サーバ |

例

```
HttpServerAdmin create -g conngroup2 -lsid ls1 -ip 1.1.1.1 -sname
iplanet -defaultvs vs2 -d server_root -sinst https-iplanet.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 -id listen-socket -ip ip_address -port port_number
-sname server_name -defaultvs default_virtual_server [-sec security] [-acct
number_of_accept_threads] -d server_root -sinst http_instance
```

パラメータ

表 A-7 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-7 待機ソケット作成のパラメータ

| パラメータ | 値 |
|---|--|
| -id <i>listen-socket</i> | 作成する待機ソケットの ID |
| -ip <i>ip_address</i> | 待機ソケットの IP アドレス |
| -port <i>port_number</i> | 待機ソケットのポート番号 |
| -sname <i>server_name</i> | 待機ソケットに関連付けるサーバ名 |
| -defaultvs <i>default_virtual_server</i> | デフォルトの仮想サーバの ID。待機ソケットを作成できるようにするには、この仮想サーバがすでに存在することが必要 |
| -acct <i>number_of_accept_threads</i> | (省略可能) 待機ソケットの受け入れスレッド数 |
| -sec <i>on</i> | (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
-conngroupid connection_group_ID [-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-8 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、仮想サーバを作成します。

表 A-8 仮想サーバ作成のパラメータ

| パラメータ | 値 |
|--|--|
| -id <i>virtual_server</i> | 作成する仮想サーバの ID |
| -cl <i>classname</i> | 仮想サーバがメンバーとなるクラス |
| -urlh <i>URL_hosts</i> | 仮想サーバの 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 |
| -conngroupid <i>connection_group_ID</i> | 待機ソケットの接続グループ |
| -state <i>state</i> | (省略可能) 有効な値は、「on」、「off」、および「disable」 |
| -docroot <i>document_root</i> | (省略可能) 仮想サーバ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 <i>mime_types_file</i> | (省略可能) 仮想サーバの MIME タイプのファイル名 |
| -aclid <i>acl_ID</i> | (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
-conngroupid group1 -d /export/iplanet/server6 -sinst
https-iplanet.com

HttpServerAdmin create -v -id vs4 -cl class1 -urlh annh,annh2
-conngroupid group1 -state off -mime mime.types -d
/export/iplanet/server6 -sinst https-iplanet.com
```

delete コマンド

delete コマンドを使用して、仮想サーバのクラス、仮想サーバ、待機ソケット、および接続グループを削除します。

オプション

表 A-9 に示すオプションを delete コマンドとともに使用し、クラス、接続グループ、待機ソケット、および仮想サーバを削除します。

表 A-9 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クラスを削除 |
| -g | 指定された接続グループを削除 |
| -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-10 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-10 クラス削除のパラメータ

| パラメータ | 値 |
|------------------|----------|
| -cl <i>class</i> | 削除するクラス名 |

例

```
HttpServerAdmin delete -c -cl class1 -d /export/iplanet/server6
-sinst https-iplanet.com
```

接続グループの削除

delete コマンドのこのオプションを使用して、接続グループを削除します。

構文

```
HttpServerAdmin delete -g -id connection_group -lsid listen_socket -d
server_root -sinst http_instance
```

パラメータ

表 A-11 に示すパラメータを delete コマンドとともに使用し、接続グループを削除します。

表 A-11 接続グループ削除のパラメータ

| パラメータ | 値 |
|-----------------------------|-----------------------|
| -id <i>connection_group</i> | 削除する接続グループの ID |
| -lsid <i>listen_socket</i> | 接続グループが属している待機ソケット ID |

例

```
HttpServerAdmin delete -g -id conngroup3 -lsid ls2 -d
/export/iplanet/server6 -sinst https-iplanet.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-12 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-12 待機ソケット削除のパラメータ

| パラメータ | 値 |
|--------------------------|----------------|
| -id <i>listen_socket</i> | 削除する待機ソケットの ID |

例

```
HttpServerAdmin delete -l -id ls3 -d /export/iplanet/server6
-sinst https-iplanet.com
```

仮想サーバの削除

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-13 に示すパラメータを delete コマンドとともに使用し、仮想サーバを削除します。

表 A-13 仮想サーバ削除のパラメータ

| パラメータ | 値 |
|---------------------------|----------------|
| -id <i>virtual_server</i> | 削除する仮想サーバの ID |
| -cl <i>class</i> | 仮想サーバが属しているクラス |

例

```
HttpServerAdmin delete -v -id vs3 -cl class1 -d
/export/iplanet/server6 -sinst https-iplanet.com
```

list コマンド

list コマンドを使用して、仮想サーバのクラス、仮想サーバ、待機ソケット、および接続グループをリストに表示します。

構文

```
HttpServerAdmin list -command_option -d server_root -sinst http_instance
```

オプション

表 A-14 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クラスをリストに表示する |
| -g -lsid listen_socket | 待機ソケットのすべての接続グループをリストに表示する |
| -l | すべての待機ソケットをリストに表示する |
| -v | すべての仮想サーバをリストに表示する |

例

```
HttpServerAdmin list -c -d /export/iplanet/server6 -sinst
https-iplanet.com

HttpServerAdmin list -l -d /export/iplanet/server6 -sinst
https-iplanet.com

HttpServerAdmin list -g -lsid ls1 -d /export/iplanet/server6
-sinst https-iplanet.com
```

コマンドウィンドウに情報のリストが表示されます。

HttpServerAdmin (仮想サーバの管理)

HTTP (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。

- ハイパーテキスト転送プロトコル (HTTP) について
- 要求
- 応答

ハイパーテキスト転送プロトコル (HTTP) について

ハイパーテキスト転送プロトコル (HyperText Transfer Protocol、HTTP) 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に接続し、サーバ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情報を送信します。サーバの応答では、プロトコルバージョン、ステータスコード、サーバ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 6.x は HTTP 1.1 をサポートします。4.0 よりも以前のバージョンでは、HTTP 1.0 をサポートしていました。サーバは、Internet Engineering Steering Group (IESG) および Internet Engineering Task Force (IETF) HTTP ワーキンググループが承認する標準 HTTP 1.1 に条件付きで準拠しています。条件付き準拠の基準の詳細は、次の Web サイトの「Hypertext Transfer Protocol--HTTP/1.1 仕様 (RFC 2068)」を参照してください。

<http://www.ietf.org/>

要求

クライアントからサーバへの要求に、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。次のメソッドが一般的に使用されています。

- 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だけを要求する
- 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に対して要求する
- PUT: サーバ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に送信することができます。ほとんどは省略可能です。一般的に使われる要求ヘッダーを表 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に認証させる場合に使用される、ユーザ名、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 | ユーザがリンク上でクリックしたドキュメントの URL |
| Host | 要求されたインターネットホストとリソースのポート番号 |

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からの応答を待ちます。

応答

次のようなサーバの応答があります。

- ステータスコード
- 応答ヘッダー
- 応答データ

ステータスコード

クライアントが要求を送信したとき、サーバが返信する項目の 1 つにステータスコードがあります。これは 3 桁の数字コードです。ステータスコードには、次の 4 つのカテゴリがあります。

- 100 から 199 までのステータスコードは、一時的な応答を示す
- 200 から 299 までのステータスコードは、正常に行われたトランザクションを示す
- 300 から 399 までのステータスコードは、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 までのステータスコードは、クライアントにエラーがあることを示す
- 500 以上のステータスコードは、サーバ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、一般的なステータスコードを示します。

表 B-2 一般的な HTTP ステータスコード

| ステータスコード | 意味 |
|----------|--|
| 200 | OK。送信は成功しました。エラーではありません。 |
| 302 | 見つかりました。新しい URL にリダイレクトします。元の URL は移動されました。これはエラーではなく、ほとんどのブラウザは新しいページを取得します。 |
| 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ページがあり、そのページが再び要求された場合、Netscape Navigator などの一部のブラウザは、キャッシュ内のコピーの「最後に変更された」タイムスタンプを Web サーバに中継します。サーバ上のコピーがブラウザのコピーから更新されていない場合は、不要なネットワークトラフィックを削減するため、サーバはページを返さず、コード 304 を送信します。これはエラーではありません。 |
| 401 | 認証不可。ユーザはドキュメントを要求しましたが、有効なユーザ名またはパスワードが入力されませんでした。 |
| 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。要求されたドキュメントはサーバ上にありません。認証されていないユーザにドキュメントが存在しないことを告げることで、そのドキュメントを保護するようにサーバが指定されている場合にも、このコードが送信される可能性があります。 |
| 500 | サーバエラー。サーバに関連するエラーが発生しました。サーバ管理者はサーバのエラーログを調べて、何が発生し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 サーバの名前とバージョン |
| Date | 現在の日付 (グリニッジ標準時間による) |

表 B-3 一般的な応答ヘッダー

| 応答ヘッダー | 説明 |
|------------------|---|
| Last-modified | ドキュメントが最後に変更された日付 |
| Expires | ドキュメントの有効期限が切れる日付 |
| Content-length | 後に続くデータの長さ (バイト) |
| Content-type | 後に続くデータの 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

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 サーバ上に格納されるリソースにアクセス可能なユーザ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 サーバはサーバ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 iPlanet の Web サイトを参照してください。

```
http://www.iplanet.com/docs
```

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。記述できるバージョンは 1 行だけで、コメント行の後に表示できます。iPlanet Web Server 6.0 ではバージョン 3.0 を使用します。たとえば、次のように記述します。

```
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。サーバは、すべてのユーザ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。iPlanet 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:/iPlanet/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、次の内容を説明します。

- 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に対してユーザ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 サーバで暗号化を有効にする必要があります、認証された信頼されている CA のリストにユーザ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。サーバの認証データベースでは、ユーザ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、サーバが認証を行う属性 (ユーザ、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と一致するユーザの基本認証を指定します。

```
authenticate (user) {
    method = "basic";
};
```

次の例では、ユーザ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、ユーザ名が sales という文字で始まるユーザを認証します。

```
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、サーバリソースへのアクセスを許可、または拒否されるユーザを指定します。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、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、次の内容を説明します。

- 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、サーバはこの URI に適用する ACL のリストを構築します。サーバはまず、`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を定義します。次の行は、複数のユーザ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 = "*.iplanet.com"
- dns = "*.iplanet.com,*.mozilla.com"
- ip = "198.*"
- ciphers = "rc4"
- ssl = "on"

また、timeofday 属性を使用すると、サーバのローカル時間を基準にした時刻で、サーバ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)))
or
    (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)

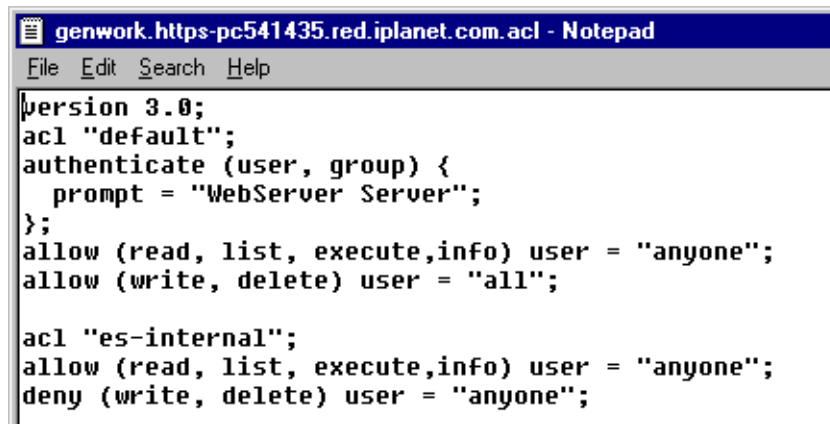
timeofday と dayofweek では、次を使用できます。

- > 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のデフォルト設定を使用できます。ユーザインタフェースで設定が作成されるまで、サーバは作業ファイル `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、iPlanet Web Server を使用して変更を保存して適用します。

図 C-1 genwork ファイル



```
version 3.0;
acl "default";
authenticate (user, group) {
    prompt = "WebServer Server";
};
allow (read, list, execute,info) user = "anyone";
allow (write, delete) user = "all";

acl "es-internal";
allow (read, list, execute,info) user = "anyone";
deny (write, delete) user = "anyone";
```

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。

国際化された iPlanet Web Server

国際化されたバージョンの iPlanet Web Server には、米国以外の環境向けに調整された特別な機能があります。この機能では、ユーザインタフェース言語 (日本語、フランス語、ドイツ語) の選択や、さまざまな言語によるテキスト検索が可能な検索エンジンの選択ができます。

この付録は、次の節から構成されています。

- 一般情報
- 検索情報
- サーブレットの国際化

一般情報

この節では、サーバ機能の国際化に関する、一般的な考慮事項について説明します。

- サーバのインストール
- UTF-8 データの入力
- Accept-Language ヘッダーの使用
- その他の言語設定の使用

サーバのインストール

サーバのインストール時に、インストールする検索エンジンだけでなく、使用するユーザインタフェース言語を選択します。

多言語バージョンのサーバのインストールについては、iPlanet Web Server, Enterprise Edition 6.0 のリリースノートを参照してください。README ファイルのリンクからリリースノートにオンラインでアクセスできます。

UTF-8 データの入力

Server Manager または Administration Server のページに UTF-8 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 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。

正しい書式で文字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 ビットまたはダブルバイトのデータを入力します。

ユーザが各自のユーザとグループの情報にアクセスできるようにする場合、UTF-8 フォームが有効なクライアントを使用する必要があります。

Accept-Language ヘッダーの使用

クライアントが HTTP を使用してサーバに接続する場合、受け入れる言語を説明したヘッダー情報を送信できます。第 16 章「コンテンツ管理」で説明するとおり、この言語情報を解析するようにサーバを構成できます。

server.xml ファイルの `acceptlanguage` 指令に対して、サーバを有効または無効にすることができます。

図 D-1 server.xml の多言語設定

| | | |
|-----------------------------|--------|--|
| <code>acceptlanguage</code> | on、off | Accept-language ヘッダーの解析を有効にしたり無効にしたりする |
|-----------------------------|--------|--|

たとえば、`acceptlanguage` が「on」に設定され、クライアントが `Accept-language` ヘッダーを `fr-CH,de` という値とともに送信することを前提に、次の URL を要求するとします。

`http://www.someplace.com/somepage.html`

サーバは次の順序でファイルを検索します。

1. Accept-language は fr-CH, de をリストします。
 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. 国コードなしの言語コード (fr-CH の場合は fr)
 http://www.someplace.com/fr/somepage.html
 http://www.someplace.com/somepage_fr.html
3. magnus.conf ファイルで定義される en などの DefaultLanguage
 http://www.someplace.com/en/somepage.html
 http://www.someplace.com/somepage_en.html
4. 上記のいずれも見つからない場合、サーバは次の場所を検索します。
 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場合、CH や TW などの国コードは小文字に変換し、ダッシュ (-) はアンダースコア (_) に変換するよう、配慮しておく必要があります。

その他の言語設定の使用

magnus.conf ファイルに含まれる次の指令は、デフォルトの言語を指定します。

表 D-1 magnus.conf の言語設定

| 指令 | 値 | 説明 |
|-----------------|----------------|--|
| ClientLanguage | en, fr, de, ja | 「Not Found」や「Access denied」などのクライアントメッセージを表示する言語を指定する。この値は、ローカライズされたメッセージに使用する ns-httpd.db データベースを指定するために使用される |
| DefaultLanguage | en, fr, de, ja | クライアント言語用のリソースがない場合に使用する言語を指定する |

検索情報

検索機能が次の言語に対してサポートされています。

- 英語
- ドイツ語
- フランス語
- イタリア語
- スペイン語
- スウェーデン語
- オランダ語
- 日本語

多言語検索

複数の文字セットを使用してエンコードされたドキュメントを表示するには、ブラウザ上で文字セットエンコーディングの設定を変更する必要があります。また、テキスト検索機能は、一度に1つの文字セットのエンコーディングに対して実行されるため、多言語検索の機能を使用しているときには、誤った結果を受け取ることがあります。最良の結果を得るには、検索コレクションを作成するとき、すべてのドキュメントに対して特定の1つの文字セットだけを使用することが必要です。

日本語での検索

この項は、日本語での検索についての情報です。

照会演算子

今回のリリースでは、次のような日本語の照会演算子をサポートしています。

表 D-2 日本語の照会演算子

| 演算子 | 日本語の文字のサポートの有無 |
|----------|----------------|
| AND | はい |
| CONTAINS | いいえ |
| ENDS | はい |
| MATCHES | はい |

表 D-2 日本語の照会演算子 (続き)

| 演算子 | 日本語の文字のサポートの有無 |
|------------|----------------|
| NEAR | はい |
| NEAR/N | はい |
| NOT | はい |
| OR | はい |
| PHRASE | はい |
| STARTS | はい |
| STEM | いいえ (英語のみサポート) |
| SUBSTRING | はい |
| WILDCARD * | はい |
| WILDCARD ? | はい |
| WORD | はい |

ドキュメント形式

今回のリリースでは、日本語の次のドキュメント形式をサポートしています。

- HTML
- ASCII
- NEWS
- MAIL

注 日本語では、PDF 形式のドキュメントがサポートされていません。

日本語での検索

次の節では、日本語文字セットでの検索について詳細情報を提供します。

ドキュメントのエンコーディング

今回のリリースでは、次のような日本語のドキュメントのエンコーディングをサポートしています。

- euc-jp

- Shift_JIS

注 ISO-2022-JP はサポートされません。

検索単語

このリリースでは、次の検索単語をサポートします。

- 漢字
- ひらがな
- カタカナ (全角と半角)
- ASCII (全角と半角)

検索エンジンは半角カタカナを全角カタカナに変換し、全角の ASCII 文字を半角の ASCII 文字に変換します。ユーザは全角文字と半角文字を同じ文字のように扱うことができます。

今回のリリースでは、句や文の検索もサポートしています。

サーブレットの国際化

POST を使用して、フォームデータがブラウザからサーバへ送信される場合、ブラウザは次の処理を実行します。

- POST データの url をエンコードする
- Content-Type を application/x-www-form-urlencoded に設定する
- Content-Type ヘッダーで文字セット情報を送信しない

サーバ側で、サーブレットが `getParameter` または `getParameterValues` を使用して、データに POST でアクセスしている場合、`getParameter` 文字列にどのエンコーディングで文字を格納したかという情報は、サーブレットコンテナには格納されません。

POST データ文字列の解釈に使用する文字のエンコード方式を、サーブレットコンテナに知らせるように、iPlanet Web Server 6.0 を構成することができます。このためには、`web-apps.xml` の `parameter-encoding` 要素を使用して、次のように文字のコード化を指定します。

```
<parameter-encoding enc="value"/>
```

value は、次の中から選択できます。

- auto (デフォルト)

- none
- 有効な任意のエンコーディング

次に、これらの値について説明します。

auto

使用される文字のエンコーディングに関するヒントを検索するために、auto ではサーブレットコンテナが必要です。次の内容を使用してヒントを指定することができます。

- `com.iplanet.server.http.servlet.parameterEncoding` という名前の要求の属性。値のタイプは `String` です。要求の属性は、`getParameter()` または `getParameterValues()` を呼び出す前に設定する必要があります。その例を次に示します。

```
request.setAttribute("com.iplanet.server.http.servlet.  
parameterEncoding", "Shift_JIS");  
request.getParameter("test");
```

このオプションは、サーブレットが固定表示データの文字セットを事前に認識している場合に使用されます。

- フォームデータ内の `j_encoding` パラメータです。送信されるフォームには、以下の隠し要素がある場合があります。その例を次に示します。

```
<input type=hidden name="j_encoding" value="Shift_JIS" >
```

このオプションは通常、データを読み込むサーブレットが必ずしも固定表示データの文字セットを事前に認識しているとは限らない場合に使用されます。デフォルトでは、`j_encoding` のヒントパラメータ名は `web-apps.xml` の `parameter-encoding` 要素を使用して変更できます。

none

このオプションは、サーブレットのパラメータデータにプラットフォームのデフォルトのエンコーディングを使用したい場合に使用します。

有効な任意のエンコーディング

上記のオプションがいずれも指定されない場合、サーブレットコンテナはこの文字列自体をエンコーディングと解釈するため、これを Shift_JIS や UTF-8 などと同様に有効なコード化文字列にすることができます。たとえば、フォームの POST データが常に UTF-8 であることがわかっている場合、これに UTF-8 を指定します。

注 サーバは常に、まず要求の Content-Type ヘッダーから文字を解決しようとします。

parameter-encoding の詳細については、『サーブレットに関するプログラマーズガイド』を参照してください。

JSP への送信

サーブレットの代わりに JSP に対してデータを送信する場合も、parameter-encoding が同じように動作するようにサーバを構成することができます。次の例では、日本語 Shift_JIS でのエンコーディングの読み取りパラメータに対して「auto」が設定されている JSP を示します。

```
<%@ page contentType="text/html; charset=Shift_JIS" %>
<html>
<head>
<title>JSP Test Case</title>
</head>
<body>
<%
request.setAttribute("com.iplanet.server.http.servlet.parameterEncoding", "Shift_JIS");
%>
<h1>The Entered Name is :<%= request.getParameter("test") %> </h1>
</body>
</html>
```

Microsoft FrontPage のサーバ拡張機能

この付録では、Microsoft FrontPage をサポートするサーバ拡張機能を iPlanet Web Server 上で使用する方法を説明します。この拡張機能では、FrontPage Web を使用する場合に必要となる、サーバ側での内部的なサポートが提供されます。

この付録は、次の節で構成されます。

- 概要
- 拡張機能のダウンロード
- FrontPage Server Extensions のインストール
- 詳細情報

概要

FrontPage のサーバ拡張機能とは、iPlanet Web Server で FrontPage Web をサポートするための CGI プログラムのことです。クライアントとサーバの通信は、標準の HTTP POST 要求を使用し、この要求は、拡張子に対応する CGI プログラムに転送されます。FrontPage Web を使用する場合、拡張機能により、FrontPage のオーサリングと発行、アクセス権限、および WebBot 機能がサポートされます。たとえば、次のような機能を利用できます。

- ユーザが FrontPage Web のフォルダ間でページを移動すると、Web 内のほかのページにあるそのページへのリンクがすべて自動的に更新されます。
- FrontPage Web の管理、オーサリング、および表示の権限を付与するユーザを指定できます。
- FrontPage Web ユーザがディスカッショングループに参加する場合、使用可能な WebBot により、ディスカッションの記事へのリンクのインデックス、目次、および検索フォームが自動的に管理されます。

サーバ拡張機能を利用すると、インターネット経由のファイル転送を最小限に抑えることができます。たとえば、拡張機能が組み込まれた iPlanet Web Server からユーザが FrontPage Web を開くと、リンクマップなどの Web メタデータがユーザのマシンにダウンロードされますが、Web ページ一式はサーバ上に残ったままです。ページがダウンロードされるのは、編集のために開く場合のみです。

サーバ拡張機能をサーバにインストールすると、インターネット上またはローカルのイントラネット上のすべてのコンピュータから FrontPage Web の公開、管理、およびディスカッショングループの機能を利用できます。ただし、オーサリングおよび管理機能を使用するには FrontPage クライアントプログラムが必要です。

この節では、次の内容について説明します。

- FrontPage Web の種類
- ドメイン名と FrontPage Web
- セキュリティについて

FrontPage Web の種類

FrontPage Web には、次の 2 種類があります。

- ルート Web は、Web サーバ (マルチホスト環境では仮想 Web サーバ) の最上位のコンテンツディレクトリです。Web サーバまたは仮想 Web サーバには、ルート Web が 1 つだけ存在します。ただし、1 つのルート Web では、複数のサブ Web をサポートすることができます。
- サブ Web は、ルート Web のサブディレクトリであり、完全な FrontPage Web です。サブ Web は、ルート Web の 1 つ下のレベルにのみ存在できます。ただし、各サブ Web には、コンテンツを構成する多数のレベルのサブディレクトリを置くことができます。

Web サーバのファイルシステムおよび URL 領域では、サブ Web はルート Web の下に表示されますが、実際には、ルート Web にサブ Web のコンテンツが含まれているわけではありません。このようなコンテンツの分割は、FrontPage Server Extensions によって行われます。

サーバ上のルート Web とすべてのサブ Web には、個別に拡張機能をインストールするか、拡張機能プログラムのスタブ実行可能ファイルを置く必要があります。FrontPage では、サーバにビルトインされるセキュリティメカニズムを使用して、アクセスを制限します。このため、拡張機能の別々のコピーを各 FrontPage Web にインストールすることにより、サーバ管理者は、エンドユーザやコンテンツの制作者、管理者などの権限を、FrontPage Web ごとに付与することができます。

ドメイン名と FrontPage Web

FrontPage Web を、iPlanet Web Server に実装することにより、Web ブラウザから FrontPage Web にアクセスできるようになります。FrontPage の実装には、次の方法があります。

- `www.mycompany.com` のようなプライベートドメイン名として使用します。これらのドメイン名は通常、マルチホストを使用する同じ物理サーバマシン上の仮想サーバとして実装されます。プライベートドメイン名の顧客は、それぞれに専用のルート Web が与えられ、オプションでサブ Web を作成することもできます。
- プライベート仮想サーバでの共通または共有ドメインとして使用します。たとえば、`www.mycompany.myprovider.com` のようなドメインです。`myprovider.com` は共有ドメイン、`www.mycompany` はプライベート仮想サーバを示します。プライベート仮想サーバでの共有ドメインの顧客は、それぞれに専用のルート Web が与えられ、オプションでサブ Web を作成することもできます。
- インターネットサービスプロバイダのサーバマシンの URL として使用します。たとえば、`www.myprovider.com/mycompany` のような URL です。URL の顧客には、単一のサブ Web が与えられます。

セキュリティについて

FrontPage では、各 FrontPage Web に含まれているすべてのファイルとディレクトリを対象とするアクセス制御リスト (ACL) を変更することによって、Web サーバの Web セキュリティを実装します。FrontPage をインストールすると必ず、各 Web の `/_vti_bin` ディレクトリ内にある Server Extensions スタブ実行可能ファイルの ACL が変更されます。FrontPage を新規にインストールすると、Web コンテンツファイルの ACL も変更されます。一方、既存の Server Extensions をアップグレードする場合は、コンテンツファイルの ACL は変更されず、FrontPage のデフォルト設定よりも低いレベルのセキュリティ設定のままになります。Web コンテンツの ACL をアップグレードするには、FrontPage Server Administrator ユーティリティの「Check and Fix」オプションを使用します。

FrontPage は、Web コンテンツファイルのセキュリティ ACL の変更に加えて、FrontPage DLL 呼び出しの結果として使用されるシステム DLL の ACL も変更します。これにより、管理者、コンテンツの制作者、エンドユーザのいずれのアカウントで実行する場合でも、システム DLL に必要なレベルの権限が設定されます。FrontPage ファイルに設定される ACL セットの完全なリスト、Server Extensions のインストール時にセキュリティに関して検討すべき事項の説明、およびシステム DLL の ACL の変更が必要とされる理由については、Ready-to-Run Software や Microsoft 社の Web サイトから利用可能な追加リソースを参照してください。

拡張機能のダウンロード

拡張機能をインストールするための最初のステップは、拡張機能をダウンロードすることです。これには、Microsoft 社の FrontPage のサイトを利用できます。UNIX または Linux の拡張機能をインストールしたい場合は Ready-to-Run Software のサイトも利用できます。Ready-to-Run Software のサイトには、さまざまな情報や使用説明書も用意されています。

- FrontPage 97 Server Extensions (バージョン 2.0)
 - [NT] 実行可能ファイルをダウンロードできます。
 - [UNIX/Linux] Ready-to-Run Software の Web サイトから、インストールスクリプトとサーバ拡張機能セットをダウンロードできます。使用するプラットフォーム用の 2 つの tar ファイルをダウンロードします。Solaris の場合は、WPP Kit Software に含まれている vt20.solaris.tar.z と wpp.solaris.tar.z になります。
 - [UNIX/Linux] Microsoft 社の Web サイトから、インストールスクリプトとサーバ拡張機能セットをダウンロードできます。使用するプラットフォーム用の 2 つの tar ファイルをダウンロードします。Solaris の場合は、WPP Kit Software に含まれている vt20.solaris.tar.z と wpp.solaris.tar.z になります。
- FrontPage 98 Server Extensions (バージョン 3.0)
 - [NT] 実行可能ファイルをダウンロードできます。
 - [UNIX/Linux] Ready-to-Run Software の Web サイトから、インストールスクリプトとサーバ拡張機能セットをダウンロードできます。fp_install.sh ファイルと、使用するプラットフォーム用の tar ファイル (Solaris の場合は fp30.solaris.tar.z) をダウンロードします。
 - [UNIX/Linux] Microsoft 社の Web サイトから、インストールスクリプトとサーバ拡張機能セットをダウンロードできます。fp_install.sh ファイルと、使用するプラットフォーム用の tar ファイル (Solaris の場合は fp30.solaris.tar.z) をダウンロードします。
- FrontPage 2000 Server Extensions (バージョン 4.0)
 - [NT] 実行可能ファイル fp2kserk.exe をダウンロードできます。このファイルは、FrontPage 拡張 Web の設定方法および使用方法に関する情報を提供します。Microsoft 社の Web サイトから、サーバ拡張機能セット fpse2k_x86_ENG.exe をダウンロードできます。
 - [UNIX/Linux] Ready-to-Run Software の Web サイトから、インストールスクリプトとサーバ拡張機能セットをダウンロードできます。fp_install.sh ファイルと、使用するプラットフォーム用の tar ファイル (Solaris の場合は fp40.solaris.tar.Z) をダウンロードします。
 - [UNIX/Linux] Microsoft 社の Web サイトから、インストールスクリプトとサーバ拡張機能セットをダウンロードできます。fp_install.sh ファイルと、使用するプラットフォーム用の tar ファイル (Solaris の場合は fp40.solaris.tar.Z) をダウンロードします。

FrontPage Server Extensions をインストールする前に、ローカルマシンに十分な空きディスク容量があること、ドキュメントルートディレクトリがあること、認証が使用可能になっていること、およびインストール後の重要事項 (アクセス権限など) について検討済みであることを確認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。

- 必要なディスク容量
- 準備作業
- その他の注意事項

必要なディスク容量

Windows NT システムでは、およそ 6M バイトの空きディスク容量が必要です。ダウンロードファイルは 3M バイト、インストールファイルは合計 2.5M バイトになります。

UNIX システムまたは Linux システムでは、サーバ上に 32M バイト以上の空き容量が必要です。UNIX または Linux の FrontPage 拡張機能には、`/usr/local/frontpage` ディレクトリに 9M バイトのディスク容量が必要です。Web コンテンツに拡張機能をインストールする場合は、Web コンテンツが `/usr/local/frontpage` と同じディスクパーティションにない限り、仮想ホストごとに 5M バイトずつ余分のディスク容量が必要です。

準備作業

iPlanet Web Server のドキュメントルートディレクトリが存在している必要があります。このディレクトリは、サーバを最初に起動したときに作成されます。このため、拡張機能をインストールする前に、少なくとも 1 回はサーバを起動する必要があります。

ドキュメントルートが存在する場合、Web サーバ構成ディレクトリの `obj.conf` ファイルで、`NameTrans fn=document root root="$docroot"` の `$docroot` を、そのドキュメントディレクトリの絶対パスに置き換えます。

その他の注意事項

- `.nsconfig` ファイルなど、FrontPage で必要な内部ファイルを削除しないでください。これらの内部ファイルを削除すると、コンテンツアップロードに対するアクセス制御が無効になります。

- 有効なエンドユーザのみに制限するように Web を設定することはできません。そのように設定すると、「This server does not support restricting end user access.」というメッセージが表示されます。
- [UNIX/Linux のみ] スタブ拡張機能をインストールする場合、Web 所有者が iPlanet Web Server ユーザと同じになるように設定する必要があります。そのように設定することで、FrontPage の拡張機能は、一定のディレクトリ、すなわち、`https-instance/config` ディレクトリおよびドキュメントルートに対する書き込み権限を持ちます。fpsrvadm.exe スクリプトで Web にスタブ拡張機能をインストールする場合、Web 所有者の入力が求められます。

FrontPage Server Extensions のインストール

Windows NT、UNIX、または Linux の各プラットフォームには、FrontPage 97、FrontPage 98、または FrontPage 2000 の拡張機能をインストールできます。ここでは、プラットフォームごとに、インストールの手順を説明します。

- Windows NT システム
- UNIX システムまたは Linux システム - FrontPage97 拡張機能
- UNIX システムまたは Linux システム - FrontPage98 拡張機能
- UNIX システムまたは Linux システム - FrontPage2000 拡張機能

Windows NT システムに FrontPage Server Extensions をインストールする

Windows NT システムに FrontPage97、FrontPage98、および FrontPage 2000 の拡張機能をインストールする手順は比較的簡単です。実行可能ファイルをダウンロードして実行すると、必要なファイルおよびフォルダがシステムにインストールされます。拡張機能は特別なディレクトリ構造を必要としますが、それについては、この節の後の方で説明します。インストールのあと、権限の設定および各 Web へのアクセスに関する追加の管理作業を実行する必要があります。

ここに示すインストール手順は、自己解凍型の実行可能ファイルに収められているスタンドアロンの FrontPage Server Extensions をインストールする場合の手順です。この自己解凍型の実行可能ファイルは、Microsoft 社の FrontPage の Web サイトから、ダウンロードできます。

注 FrontPage Server Extensions をインストールするには、Windows NT システムに Administrator としてログオンするか、または Administrator の権限を所有している必要があります。

FrontPage と Server Extension Resource Kit は、Web サーバと同じマシンにインストールする必要があります。Windows NT に FrontPage Server Extensions をインストールするには、次の手順を実行します。

1. システムの言語とプロセッサのタイプに対応するサーバ拡張機能セットアッププログラムを実行します。

たとえば、英語の FrontPage98 拡張機能をインストールする場合は、fp98ext_x86_enu.exe ファイルを実行します。このサーバ拡張機能は、C:\Program Files\MicrosoftFrontPage\Version 3.0 フォルダにコピーされます。英語の FrontPage2000 拡張機能をインストールする場合は、fpse2k_x86_ENG.exe ファイルを実行します。このサーバ拡張機能は、C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40 フォルダにコピーされます。

2. FrontPage 2000 と Server Extensions Resource Kit をインストールしたあと、「スタート」メニューの「プログラム」に含まれる「管理ツール」から、Server Extensions Administrator を選択し、「Console Root」の「FrontPage Server Extensions」から自分のマシンのホスト名を右クリックして「New Web」を選択します。ウィザードの指示に従って、サーバインスタンスを選択し、FrontPage Server Extensions 用に構成します。
3. FrontPage Server Extensions をインストールする仮想サーバを選択し、「OK」をクリックします。
4. 新しい FrontPage 管理者のアカウント名とパスワードを入力します。

サーバ拡張機能をインストールしたあと、別の管理者アカウントを追加するには、FrontPage Explorer の「権限」コマンドを使用します。

各 FrontPage Web へのサーバ拡張機能のインストールには数分間かかる場合があります。また、コンピュータの CPU 負荷が増加することもあります。FrontPage Server Extensions を新規にインストールする場合、各ページのコンテンツが構文解析されて、挿入コンポーネント、代入コンポーネントなどの FrontPage コンポーネントの拡張、FrontPage Web のハイパーリンクマップの作成、およびページタイトルとベース URL の抽出が行われます。

インストールプロセスでは、テキストインデックスの更新、Web 内のリンクの再検討、FrontPage 管理者アカウント、パスワード、および IP アドレス制限の追加が行われます。さらに、obj.conf ファイルに新しい ObjectType 指令が追加された場合は、Web 管理者に対してサーバの再起動を促します。

FrontPage97 拡張機能の場合、コンポーネントは C:\Program Files\Microsoft FrontPage ディレクトリにインストールされます (C はインストール先となるハードドライブを示す)。インストールされるコンポーネントは次のとおりです。

- FrontPage Server Extensions の .dll ファイルおよび .exe ファイルは、\bin サブディレクトリおよびデフォルトの \windows\system ディレクトリにコピーされます。
- ユーザの Web サイトに Server Extensions の機能を実装するために FrontPage が使用する ISAPI (.dll) ファイルまたは CGI (.exe) ファイルは、\isapi ディレクトリおよび _vti_bin ディレクトリにそれぞれコピーされます。これらのファイルは、FrontPage 拡張機能をインストールする各仮想サーバのドキュメントルートにもコピーされます。
- FrontPage Server Administrator (fpsrvwin.exe) およびそのコマンド行バージョン (fpsrvadm.exe) は、\bin ディレクトリにコピーされます。FrontPage Server Administrator は、FrontPage Server Extensions のインストール、更新、確認、または削除を実行するためのツールです。

FrontPage98 拡張機能の場合、コンポーネントは C:\Program Files\Microsoft FrontPage\version 3.0 ディレクトリにインストールされます (C はインストール先のハードドライブを示す)。

- FrontPage Server Extensions の .dll ファイルおよび .exe ファイルは、\bin サブディレクトリおよびデフォルトの \windows\system ディレクトリにコピーされます。
- ユーザの Web サイトに Server Extensions の機能を実装するために FrontPage が使用する 3 つの ISAPI (.dll) ファイルまたは CGI (.exe) ファイルは、\isapi ディレクトリおよび _vti_bin ディレクトリにそれぞれコピーされます。これらのファイルは、FrontPage 拡張機能をインストールする各仮想サーバのドキュメントルートにもコピーされます。
- FrontPage Server Administrator (fpsrvwin.exe) およびそのコマンド行バージョン (fpsrvadm.exe) は、\bin ディレクトリにコピーされます。FrontPage Server Administrator は、FrontPage Server Extensions のインストール、更新、確認、または削除を実行するためのツールです。
- Server Extensions Resource Kit
- HTML 管理フォーム。これは、Web ブラウザから FrontPage Server Extensions をリモートで管理するための一連の HTML フォームです。FrontPage Server Extensions をリモートで管理するためのコマンド行ユーティリティ (fpremadm.exe) も \bin ディレクトリにインストールされます。

FrontPage2000 拡張機能の場合、コンポーネントは C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40 ディレクトリにインストールされます (C はインストール先のハードドライブを示す)。

- FrontPage Server Extensions の .dll ファイルおよび .exe ファイルは、\bin サブディレクトリおよびデフォルトの \WINNT\system32 ディレクトリにコピーされます。

- ユーザの Web サイトに Server Extensions 機能を実装するために FrontPage が使用する 3 つの ISAPI (.dll) ファイルまたは CGI (.exe) ファイルは、\isapi ディレクトリおよび _vti_bin ディレクトリにそれぞれコピーされます。これらのファイルは、FrontPage 拡張機能をインストールする各仮想サーバのドキュメントルートにもコピーされます。
- FrontPage Server Administrator (fpsrvwin.exe) は、FrontPage 2000 では使用しないのでインストールされません。FrontPage 2000 では、Server Extensions Administrator (「スタート」メニューの「プログラム」に含まれる「管理ツール」から選択) を使用するか、または \bin サブディレクトリにインストールされているコマンド行バージョン (fpsrvadm.exe) を使用する必要があります。FrontPage Server Administrator は、FrontPage Server Extensions のインストール、更新、確認、または削除を実行するためのツールです。
- Server Extensions Resource Kit は、\serk サブディレクトリにコピーされます。
- HTML 管理フォーム (Web ブラウザを通して FrontPage Server Extensions をリモートで管理するための一連の HTML フォーム) は、\admcgi サブディレクトリと \admisapi サブディレクトリにコピーされます。FrontPage Server Extensions のリモートで管理するためのコマンド行ユーティリティ (fpremadm.exe) も \bin ディレクトリにインストールされます。

インストールでは、次のファイルおよびディレクトリの変更または追加も行われます。

- magnus.conf ファイルが変更されます。
- サーバの構成ファイル (obj.conf) に ObjectType 指令が追加され、実行可能ファイルを格納する 3 つの _vti_ ディレクトリが作成されます。
- サーバのドキュメントルートの下に、次の 7 つのサブディレクトリが作成されます。

```

\_private
\_vti_bin (shtml.exe を格納)
\_vti_bin\_vti_adm (admin.exe を格納)
\_vti_bin\_vti_aut (author.exe を格納)
\_vti_cnf
\_vti_log
\_vti_pvt
\_vti_txt
\_images

```

- _vti_bin、_vti_adm、_vti_aut の各サブディレクトリおよびドキュメントルートディレクトリに .nsconfig ファイルが作成されます。

インストール作業が完了したあと、次の管理作業も実行する必要があります。

- FrontPage 97 および 98 では、fpsrvwin.exe ファイル (FrontPage ディレクトリの \bin ディレクトリにある) を実行して、サーバポートの設定、拡張機能のテスト、ほかの仮想サーバへの拡張機能のインストール、および拡張機能の更新を行います。

- FrontPage 2000 では、Server Extensions Administrator (「スタート」メニューの「プログラム」の「管理ツール」) またはコマンド行バージョン (fpsrvadm.exe) を実行します。
- 作業対象のサーバおよび Web を選択します。
 - リモートマシンには、FrontPage97、FrontPage98、または FrontPage 2000 のプログラムがインストールされている必要があります (Macintosh または Windows のみ)。FrontPage プログラムを起動すると、ユーザは、開くまたは編集するサーバの名前を入力するよう求められます。
 - ユーザが別のマシン上の Web を編集する場合は、その行で「MoreWebs」をクリックして Web サーバまたはディスクの場所を選択し、編集する Web の *servername:portnumber* を入力して、「OK」をクリックします。
 - ホストマシンの Web リストから編集する Web を選択します。
- 各追加 Web のプロビジョニングを行う必要があります。FrontPage クライアントがルート Web に対する正当な権限 (管理者のユーザ名とパスワード) を所有している場合は、クライアント側からこれを行うことができます。また、fpsrvadm.exe プログラムを使用して個々の Web にパスワードを設定することにより、サーバ側からユーザの Web のプロビジョニングを行うこともできます。新しい FrontPage Web がルート Web から管理者のユーザ名とパスワードを継承しないようにする必要があります。
- fpadmin.htm ファイルを探します。通常は FrontPage プログラムディレクトリの \admin\cgi ディレクトリ (97 および 98 の場合) または \admcgi ディレクトリ (2000 の場合) にあります。このファイルを使用して、FrontPage Web を構成できます。
- ユーザは、FrontPage を起動したときに表示されるローカルの Web を編集できますが、変更を加えるには、有効なユーザ ID とパスワードが必要です。

UNIX システムまたは Linux システムに FrontPage97 Server Extensions をインストール する

UNIX システムまたは Linux システムでインストール作業を行うには、適切なファイル権限およびディレクトリが事前に設定されている必要があります。拡張機能は特別なディレクトリ構造を必要としますが、それについては、この節の後の方で説明します。インストールのあと、権限の設定および各 Web へのアクセスに関する追加の管理作業を実行する必要があります。

ここに示すインストール手順は、tar ファイルに収められているスタンドアロンの FrontPage Server Extensions をインストールする場合のものです。この tar ファイルは、Microsoft 社の FrontPage の Web サイトまたは Ready-To-Run Software の Web サイトからダウンロードできます。

注 インストールを実行するには、root ユーザとしてログインする必要があります。また、この root ユーザは、/usr/local ディレクトリに対する書き込み権限を持っている必要があります。拡張機能を別のディレクトリにインストールする場合も同じです。別のディレクトリに拡張機能をインストールする場合は、/usr/local へのソフトリンクが自動的に追加されます。

拡張機能をインストールするには、次の手順に従います。

1. tar ファイルから FrontPage Server Extensions をインストールできるように、root ユーザとしてログインし、次のコマンドを実行します。

```
cd /usr/local
```

2. ダウンロードした tar ファイルを展開します。

これにより、/usr/local/FrontPage/version2.0 ディレクトリが作成され、その他のいくつかのディレクトリがドキュメントルートディレクトリの下にインストールされます。たとえば、FrontPage 97 拡張機能を Solaris プラットフォームにインストールする場合、vt20.solaris.tar.z ファイルに対して、次のコマンドを実行します。

```
tar xvf /usr/tmp/vt20.solaris.tar
```

3. ディレクトリを /usr/local/frontpage/version2.0 に変更します。

```
cd frontpage/version2.0
```

4. /extensions というディレクトリを作成し、そこに _vti_bin ディレクトリを移動します。

```
mv _vti_bin extensions
```

5. WPP キットを /usr/local/frontpage/version2.0 ディレクトリにインストールします。

Solaris の場合は、次のコードを使用します。

```
tar xvf /usr/tmp/wpp.solaris.tar
```

6. /executables ディレクトリ (/usr/local/frontpage/version2.0/executables) の名前を /_vti_bin に変更します。

```
mv executables _vti_bin
```

7. fpsrvadm.suid.exe ファイルを /bin ディレクトリに移動します。

```
mv fpsrvadm.suid.exe bin
```

8. `fp_install.sh` シェルプログラムを実行し、画面の指示に従って、次の表に示された情報を入力します。

サーバの構成ファイル名を入力するよう求められた場合は、サーバの `magnus.conf` ファイルのパス名を入力します。

表 E-1 インストールパラメータ情報

| | | |
|--|-------|--|
| <code>-fpdir <dir></code> | デフォルト | FrontPage ディレクトリ |
| <code>-httpdconfdir <dir></code> | デフォルト | サーバの構成ファイルが格納されているディレクトリ |
| <code>-web <webname></code> | 必須 | Server Extensions をインストールする Web (ルート <code>web</code> の場合は <code>/</code>) |
| <code>-user <webowner></code> | 必須 | <code>web</code> の所有者のユーザ ID |
| <code>-group <webgroup></code> | 省略可能 | <code>web</code> の所有者のグループ ID |
| <code>-host <host></code> | | Server Extensions をインストールする仮想ホストの名前。指定するホスト名は、サーバの <code>httpd.conf</code> ファイルの Virtual Host 指令で指定されたホスト名と同じであることが必要 |
| <code>-admuser <fpadmin></code> | 必須 | FrontPage Administrator のユーザ名 |
| <code>-admpass <fppass></code> | 必須 | FrontPage Administrator のパスワード |
| <code>-admaddr <ipaddr></code> | 省略可能 | FrontPage Administrator の IP アドレス制限。IP アドレスマスクが指定されていない場合、FrontPage Administrator は、すべての IP アドレスからアクセスできる |

各 FrontPage Web への Server Extensions のインストールには数分間かかる場合があります。また、コンピュータの CPU 負荷が増加することもあります。FrontPage Server Extensions を新規にインストールする場合、各ページのコンテンツが構文解析されて、挿入コンポーネント、代入コンポーネントなどの FrontPage コンポーネントの拡張、FrontPage Web のハイパーリンクマップの作成、およびページタイトルとベース URL の抽出が行われます。

インストールプロセスでは、テキストインデックスの更新、Web 内のリンクの再検討、FrontPage 管理者アカウント、パスワード、および IP アドレス制限の追加が行われます。さらに、`obj.conf` ファイルに新しい `ObjectType` 指令が追加された場合は、Web 管理者に対してサーバの再起動を促します。

インストール中、インストールシエルにより、次のファイルおよびディレクトリの変更または追加が行われます。

- `magnus.conf` が変更されます。
- `/usr/local/frontpage/hostname:port.cnf` という名前の構成ファイルが作成されます。
- サーバの構成ファイル (`obj.conf`) に `ObjectType` 指令が追加され、実行可能ファイルを格納する 3 つの `/_vti_` ディレクトリが作成されます。
- サーバのドキュメントルートの下に、次の 7 つのサブディレクトリが追加されます。

```

/_vti_bin (shtml.exe を格納)
/_vti_bin/_vti_adm (admin.exe を格納)
/_vti_bin/_vti_aut (author.exe を格納)
/_vti_cnf
/_vti_pvt
/_private
/_vti_log
/_vti_txt
/images

```

- `/_vti_bin`、`/_vti_adm`、`/_vti_aut` の各サブディレクトリおよびドキュメントルートディレクトリに `.nsconfig` ファイルが作成されます。

インストール作業が完了したあと、次の管理作業を実行する必要があります。

1. `fpsrvwin.exe` ファイルを実行して、サーバポートの設定、拡張機能のテスト、ほかの仮想サーバへの拡張機能のインストール、および拡張機能の更新を行います。
2. リモートマシンには、FrontPage97、FrontPage98、または FrontPage 2000 のプログラムがインストールされている必要があります (Macintosh または Windows のみ)。

FrontPage プログラムを起動すると、ユーザは、開くまたは編集するサーバの名前を入力するよう求められます。

3. ユーザが別のマシン上の Web を編集する場合は、その行で「MoreWebs」をクリックして Web サーバまたはディスクの場所を選択し、編集する Web の `servername:portnumber` を入力します。「OK」をクリックします。
4. ホストマシンの Web リストから目的の Web を選択して編集します。

UNIX システムまたは Linux システムに FrontPage98 Server Extensions をインストール する

ここに示すインストール手順は、tar ファイルに収められているスタンドアロンの FrontPage Server Extensions をインストールする場合の手順です。この tar ファイルは、Microsoft 社の FrontPage の Web サイトまたは Ready-To-Run Software の Web サイトからダウンロードできます。

インストールのあと、権限の設定および各 Web へのアクセスに関する追加の管理作業を実行する必要があります。

| | |
|----------|---|
| 注 | インストールを実行するには、root ユーザとしてログインする必要があります。また、この root ユーザは、/usr/local ディレクトリに対する書き込み権限を持っている必要があります。拡張機能を別のディレクトリにインストールする場合も同じです。別のディレクトリに拡張機能をインストールする場合は、/usr/local へのソフトリンクが自動的に追加されます。 |
|----------|---|

拡張機能をインストールするには、次の手順に従います。

1. tar ファイルから FrontPage Server Extensions をインストールできるように、root ユーザとしてログインします。
2. cd /usr/local と入力するか、cd コマンドを利用して 2 つのダウンロードファイル (fp30.solaris.tar.Z および fp_install.sh) が格納されているディレクトリに移動します。
3. fp_install.sh シェルプログラムを実行し、画面の指示に従って、パラメータ情報を入力します。

サーバ構成ファイルの名前を入力するよう求められた場合は、サーバの magnus.conf ファイルのパス名を入力します。

各 FrontPage Web への Server Extensions のインストールには数分間かかる場合があります。また、コンピュータの CPU 負荷が増加することもあります。FrontPage Server Extensions を新規にインストールする場合、各ページのコンテンツが構文解析されて、挿入コンポーネント、代入コンポーネントなどの FrontPage コンポーネントの拡張、FrontPage Web のハイパーリンクマップの作成、およびページタイトルとベース URL の抽出が行われます。

インストールプロセスでは、テキストインデックスの更新、Web 内のリンクの再検討、FrontPage 管理者アカウント、パスワード、および IP アドレス制限の追加が行われます。さらに、obj.conf ファイルに新しい ObjectType 指令が追加された場合は、Web 管理者に対してサーバの再起動を促します。

インストール中、インストールシェルにより、次のファイルおよびディレクトリの変更または追加が行われます。

- magnus.conf が変更されます。
- /usr/local/frontpage/hostname:port.conf という名前の構成ファイルが作成されます。
- サーバの構成ファイル (obj.conf) に ObjectType 指令が追加され、実行可能ファイルを格納する 3 つの /_vti_ ディレクトリが作成されます。
- サーバのドキュメントルートの下に、次の 7 つのサブディレクトリが追加されます。

```

/_vti_bin (shtml.exe を格納)
/_vti_bin/_vti_adm (admin.exe を格納)
/_vti_bin/_vti_aut (author.exe を格納)
/_vti_cnf
/_vti_pvt
/_private
/_vti_log
/_vti_txt
/images

```

/_vti_bin、/_vti_adm、/_vti_aut の各サブディレクトリおよびドキュメントルートディレクトリに .nsconfig ファイルが作成されます。

UNIX システムまたは Linux システムに FrontPage2000 Server Extensions をインストールする

ここに示すインストール手順は、tar ファイルに収められているスタンドアロンの FrontPage Server Extensions をインストールする場合の手順です。この tar ファイルは、Microsoft 社の FrontPage の Web サイトまたは Ready-To-Run Software の Web サイトからダウンロードできます。

インストールのあと、権限の設定および各 Web へのアクセスに関する追加の管理作業を実行する必要があります。

注 インストールを実行するには、`root` ユーザとしてログインする必要があります。また、この `root` ユーザは、`/usr/local` ディレクトリに対する書き込み権限を持っている必要があります。拡張機能を別のディレクトリにインストールする場合も同じです。別のディレクトリに拡張機能をインストールする場合は、`/usr/local` へのソフトリンクが自動的に追加されます。

拡張機能をインストールするには、次の手順に従います。

1. `tar` ファイルから FrontPage Server Extensions をインストールできるように、`root` ユーザとしてログインします。
2. `cd /usr/local` と入力するか、`cd` コマンドを利用して 2 つのダウンロードファイル (`fp40.solaris.tar.Z` および `fp_install.sh`) が格納されているディレクトリに移動します。
3. `fp_install.sh` シェルプログラムを実行し、画面の指示に従って、パラメータ情報を入力します。
4. サーバ構成ファイルの名前を入力するよう求められた場合は、サーバの `magnus.conf` ファイルのパス名を入力します。

各 FrontPage Web への Server Extensions のインストールには数分間かかる場合があります。また、コンピュータの CPU 負荷が増加することもあります。FrontPage Server Extensions を新規にインストールする場合、各ページのコンテンツが構文解析されて、挿入コンポーネント、代入コンポーネントなどの FrontPage コンポーネントの拡張、FrontPage Web のハイパーリンクマップの作成、およびページタイトルとベース URL の抽出が行われます。

インストールプロセスでは、テキストインデックスの更新、Web 内のリンクの再検討、FrontPage 管理者アカウント、パスワード、および IP アドレス制限の追加が行われます。さらに、`obj.conf` ファイルに新しい `ObjectType` 指令が追加された場合は、Web 管理者に対してサーバの再起動を促します。

インストール中、インストールシェルにより、次のファイルおよびディレクトリの変更または追加が行われます。

- `magnus.conf` が変更されます。
- `/usr/local/frontpage/hostname:port.cnf` という名前の構成ファイルが作成されます。
- サーバの構成ファイル (`obj.conf`) に `ObjectType` 指令が追加され、実行可能ファイルを格納する 3 つの `/_vti_` ディレクトリが作成されます。
- サーバのドキュメントルートの下に、次の 7 つのサブディレクトリが追加されません。

```
/_vti_bin (shtml.exe を格納)  
/_vti_bin/_vti_adm (admin.exe を格納)  
/_vti_bin/_vti_aut (author.exe を格納)  
/_vti_cnf  
/_vti_pvt  
/_private  
/_vti_log  
/_vti_txt  
/images
```

/_vti_bin、/_vti_adm、/_vti_aut の各サブディレクトリおよびドキュメントルートディレクトリに .nsconfig ファイルが作成されます。

詳細情報

さらに詳細な情報は、Microsoft 社の FrontPage の Web サイトから入手できます。URL は次のとおりです。

<http://www.microsoft.com>

UNIX または Linux に関する情報は、Ready-to-Run Software の Web サイトからも入手できます。URL は次のとおりです。

<http://www.rtr.com>

用語集

Administration Server 使用するすべての iPlanet 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。

admpw Administration Server のスーパーユーザのユーザ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.iplanet.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 サーバとクライアントの間で情報をやり取りするための手法。

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。iPlanet Web Server を HTTPD と呼ぶこともある。

HTTP-NG 次世代の HTTP。

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。

iPlanet Console 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 iPlanet サーバを管理できるグラフィカルインタフェースをサーバ管理者に提供する Java アプリケーション。インストールした iPlanet Console のどのインスタンスからでも、自社ネットワーク上にあつてアクセス権を与えられているすべての iPlanet サーバを表示し、アクセスできる。

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 (JSP) 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、構成管理など、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、構成管理など、Java サ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 サブレットは、Web ブラウザではなく Web サーバ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 iPlanet Web Server 構成ファイル。このファイルには、サーバのグローバルな構成情報 (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、初期化時にサーバを構成する変数の値を設定する。iPlanet 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ので、このファイルに変更を加えた時は、サーバを再起動する必要がある。

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) タイプの構成ファイル。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、要求されたコンテンツのタイプをサーバが判別できるようにする。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)。サーバの MTA Host を、サーバ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、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。ニュースサーバホストを、サーバ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 サーバのオブジェクト構成ファイル。このファイルには、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が使用する付加的な初期化情報、サーバのカスタマイズに必要な設定、および命令が記述されている。iPlanet 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込む。

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。root ユーザは、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 iPlanet サーバのコア機能の拡張やカスタマイズ、および HTTP サーバ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 (ルーター構成など) 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 Transport Layer Security の略。二者 (クライアントとサーバ) 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。

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一つ。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 の略。二者 (クライアントとサーバ) 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、システム資源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.iplanet.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/iplanet/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。ユーザがサーバファイルの物理的な場所を知る必要をなくすことで、サーバのセキュリティを高めることができる。

WAR (Web Application Archive) 圧縮形式で Web アプリケーション全体を収めたアーカイブファイル。iPlanet Web Server では、WAR ファイルのアプリケーションにアクセスできない。iPlanet Web Server で Web アプリケーションを扱えるようにするには、それを圧縮解除する (wdeploy ユーティリティを使用して導入する) 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 NT) 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、サーバへのアクセス権を持つユーザ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。

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 サーバ内で、ユーザ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、サーバに対するクライアントの働きをし、サーバ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 単一サーバに複数のドメイン名、IP アドレス、およびサーバ監視機能を設定する手法。

仮想サーバクラス obj.conf ファイル内の同じ基本構成情報を共有する仮想サーバの集まり。

危殆化鍵リスト (Compromised key list, CKL) 危殆化された鍵を持つユーザ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から取得する必要がない。

共通ログファイル形式 サーバがアクセスログに情報を記入する時に使用する形式。iPlanet Web Server など、すべての主要サーバ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」管理サーバによって追加、制御されるリモートの「スレーブ」管理サーバのグループ。クラスタ内のすべてのサーバは、同じプラットフォームを使用し、同じユーザ 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。

サーバルート サーバプログラム、構成ファイル、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マシン上のディレクトリ。

サービス品質 サーバインスタンス、仮想サーバクラス、または仮想サーバに対して設定するパフォーマンス制限。

証明機関 (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。

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。

証明書の取り消しリスト (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 (UNIX) UNIX マシンで使用できる最高の権限を持つユーザ (**root** と呼ばれる)。スーパーユーザは、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。一般に、*the*、*a*、*an*、*and* などの単語がある。「検索から除外する単語」とも呼ばれる。

ソフトリスタート サーバの再起動方法の 1 つであり、サーバを内部的に再起動させる、つまり、構成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。サーバとクライアントの間の接続は待機ソケット上で行われる。

ダイジェスト認証 ユーザ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 サーバにアクセスするユーザ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 サーバまたは Web サイトにアクセスするためのユーザ名とパスワードの入力をユーザに要求する。LDAP データベース内のユーザ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。

サーバ全体、またはサーバ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。

ネットワークマネージメントステーション (network management station、NMS) ユーザ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、iPlanet サーバなど、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 ユーザのログイン名、パスワード、およびユーザ 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構成。通常はハードウェアとソフトウェアの両方で構成される。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。

符号化方式 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)。

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 サーバ上に存在し、そのサーバ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、サーバの構成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.iplanet.com* は、*com* ドメイン内の *iplanet* 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、同じサーバまたは別のサーバ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。

記号

- , 289
- != (等しくない), 408
- \$, 289
- \$\$logo, 284
- \$\$NS-collection-list, 292
- \$\$NS-collection-list-dropdown, 292
- \$\$NS-collections-searched, 292
- \$\$NS-display-query, 292
- \$\$NS-doc-href, 292
- \$\$NS-doc-name, 292
- \$\$NS-doc-number, 292
- \$\$NS-doc-path, 292
- \$\$NS-doc-score, 292
- \$\$NS-doc-score-div10, 292
- \$\$NS-doc-score-div5, 292
- \$\$NS-docs-found, 293
- \$\$NS-doc-size, 293
- \$\$NS-docs-matched, 293
- \$\$NS-docs-searched, 293
- \$\$NS-doc-time, 292
- \$\$NS-get-highlighted-doc, 293
- \$\$NS-get-next, 293
- \$\$NS-get-prev, 293
- \$\$NS-host, 293
- \$\$NS-insert-doc, 293
- \$\$NS-max-records, 284
- \$\$NS-rel-doc-name, 293
- \$\$NS-search-offset, 293
- \$\$NS-server-url, 285, 293
- \$\$NS-sort-by, 293
- \$\$queryLabel, 284
- \$\$sitename, 284
- \$TOKENNAME, 115
- \$、ワイルドカードの, 23, 69, 71, 72, 120, 174
- %vsid%、ログファイルの書式文字列内, 209
- %vsid%、ログファイルの書式文字列に追加, 209
- ©;, 289
- >;, 289
- <;, 289
- ;, 289
- ";, 289
- ®;, 289
- *、ワイルドカードの, 23, 69, 71, 72, 120, 174
- /helpFiles, 262
- = (等号), 408
- = (等しい), 278
- >= 以上, 408
- > より大きい, 408
- ? ワイルドカード演算子, 281
- ?、ワイルドカードの, 23, 69, 71, 72, 120, 174
- ^、ワイルドカードの, 23, 69, 71, 72, 120, 174

l、ワイルドカードの、23
~、ワイルドカードの、23, 69, 71, 72, 120, 174

数字

200 - 500 ステータスコード、400
4.x サーバを 6.0 サーバに移行する、49
8 ビットテキスト、412

A

Accept, 398
Accept Language ヘッダー
使用、412
AcceptLanguage, 412
Accept Language ヘッダー、解析、367
AcceptTimeout, 162
ACL
URI へのアクセス制限、187
obj.conf、参照、410
アクション、設定、178
アクセス拒否メッセージの変更、184
仮想サーバ、316
仮想サーバ、設定、330
仮想サーバへのアクセス制限、200
仮想サーバ用の設定を編集、202
サーバ全体へのアクセス制限、185
サーバのダイジェスト認証プロシージャ、163
時刻に基づくアクセス制限、188
承認文、404, 405
セキュリティに基づくアクセス制限、189
属性式、407
ディレクトリへのアクセス制限、186
デフォルトファイル、409
ファイル、構文、403
ファイルタイプに対するアクセス制限、187
分散管理と、58
無効化、184
ユーザとグループの指定、178
.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、166

ACLCacheLifetime, 166
ACLFILE, 200
-aclid, 390
aclname, 410
ACLUserCacheSize, 166
ACL ユーザキャッシュ
サーバがユーザとグループの認証結果を格納
、166
admaddr, 430
admin/logs
ログファイルの場所、59
Administration Server
Cron デーモンの起動と停止、61
SNMP マスターエージェントの起動、238
SSL を有効にする、105
UI の概要、31
URL ナビゲーション、40
アクセス、45
インスタンス、Web サーバの、31
コントロールパネルのサービスアプレットから
起動、46
サーバの削除、48
紹介、40
セキュリティと、128
停止、53
トップページのメインのタブ、40
ユーザ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、75
Administration Server のシャットダウン、53
admpass, 430
admpw, 37, 57
構成ファイル、概要、35
スーパーユーザのユーザ名とパスワードのファ
イル、56
admuser, 430
AIX
SNMP 関連事項、233
alias ディレクトリ、36, 98
allow, 195
AND, 277, 414
and, 408
ansi_x3.4-1968, 369
ansi_x3.4-1986, 369

API のリファレンス
 JSP, 338
 サブレット, 337

ASCII, 415
ascii, 369
AuthGroupFile, 194, 196
AuthName, 197
Authorization, 399
AuthTrans qos-handler, 220
AuthType, 197
AuthUserFile, 196

B

bin ディレクトリ, 36
bong-file, 126

C

c, 122

CA

種類, 117
承認プロセス (1 日から 2 か月), 95
信頼できる, 96
定義 (証明機関), 88

certmap.conf, 120, 161
 LDAP 検索, 119
 使用, 120
 デフォルトのプロパティ, 121
 マッピング例, 123

certSubjectDN, 125

CGI, 368
 Windows, 351
 Windows NT ディレクトリの指定, 352
 Windows NT でのシェルプログラムのインストール, 354
 Windows NT ファイルタイプの指定, 353
 Windows NT プログラム、概要, 351
 インストール, 346
 インストールプログラム, 347
 概要, 346

仮想サーバ、一意の属性を構成する, 349
仮想サーバで使用する, 307
サーバ拡張機能、概要, 32
シェル, 354
シェルディレクトリの指定、Windows NT, 354
実行可能ファイルのダウンロード, 350
定義済み (Common Gateway Interface), 335
ディレクトリの削除, 349
ディレクトリの指定, 348
ファイル拡張子, 348
ファイルタイプ, 349
ファイルタイプ、Windows NT でのシェルの指定, 355
ファイルタイプの指定, 349
プログラム、サーバへのインストール方法, 336
プログラム、サーバへの格納方法, 347

CGI.exe, 426

CGI (Common Gateway Interface)
 アーキテクチャの概要, 32
 概要, 346
 サーバ拡張機能、概要, 32

CGIStub

CGI の実行を補助するためのプロセス, 346

CGI プログラム

FrontPage 拡張機能, 419

CGI プロセッサ
 実行環境, 33

check-acl, 410

chroot, 132

仮想サーバにディレクトリクラスを指定する, 132
仮想サーバにディレクトリを指定する, 133

CKL(Compromised Key List)

インストールおよび管理する, 102

CKL (危険化鍵リスト)

インストールおよび管理, 102

ClassCache, 345

ClassCache ディレクトリ, 37

ClassCache ファイル, 39

Class Manager

UI の概要, 31
アクセス, 43
紹介, 43
追加タブのリスト, 43

ClientLanguage, 413
 CmapLdapAttr, 122, 125
 cn, 69, 122
 common-log, 210
 conf_bk ディレクトリ, 37
 conf_bk ファイル, 39
 CONFIG, 232, 234
 マスターエージェント、編集, 235
 config ディレクトリ, 37
 CONFIG ファイル, 235
 config ファイル, 39
 -conngroupid, 390
 CONTAINS, 277, 414
 contains
 検索形式のオプション, 72
 Content Management エンジン、アーキテクチャの
 概要, 32
 Content-length, 401
 Content-type, 401
 cookie
 CGI プログラムを起動できるようにする, 41
 ロギング、簡易, 210
 cp367, 369
 cp819, 370
 CRL(Certificate Revocation List)
 インストールおよび管理, 102
 CRL (証明書の取消しリスト)
 インストールおよび管理, 102
 cron.conf, 37, 208
 Cron デーモン
 Cron 制御の使用, 60
 Cron ベースのログローテーション, 208

D

Date, 257, 400
 dayofweek, 408
 dblist.ini, 272, 281, 285, 291
 dblist.ini ファイル, 254
 dbswitch.conf, 200
 dbswitch.conf ファイル, 180
 dcsuffix, 200
 defaultclass
 仮想サーバクラス, 300
 DefaultLanguage, 413
 DELETE, 183
 「Delete」 アクセス, 183
 deny, 196
 descriptions.pat, 283
 DES アルゴリズム
 Directory Server の設定, 164
 DES 符号化方式, 116
 digestauth, 162
 digest ディレクトリ, 38
 Directory Server
 DES アルゴリズムの設定, 164
 ldapmodify コマンド行ユーティリティ, 67
 Web サーバへのユーザやグループ追加、ディレ
 クトリサーバをインストールする必要
 がある, 30
 分散管理に必要な, 58
 ユーザエントリ, 68
 ユーザとグループの管理, 56
 DirID, 257
 DN
 ディレクトリサーバのエントリ名を表す文字列
 , 68
 DNComps, 121
 DNS
 サーバパフォーマンスに対する検索の影響を小
 さくする, 165
 -docroot, 390
 docs ディレクトリ, 36
 Domain Name System
 エイリアス、定義, 437
 定義, 437
 dsgw.conf, 37
 dsgwfilter.conf, 37
 dsgwlanguage.conf, 37
 dsgw-orgperson.conf, 37
 dsgwserarchprefs.conf, 37

E

e, 122
ENDS, 278, 414
ends with
 検索タイプのオプション, 73
Error qos-error, 220
euc, 415
「Execute」 アクセス, 183
Expires, 401
Expires ヘッダー、定義, 438
extras ディレクトリ, 36

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), 90
Federal Information Processing Standards (FIPS)
 -140, 116
FileName, 257
FilterComps, 121
FIPS-140
 有効, 116
flex_anlg, 211
flexanlg
 使用と構文, 211
flexanlg ディレクトリ, 36
flex-init, 210
flex-log, 210
fpdir, 430
fprsvadm.exe, 426
fprsvwin.exe, 426
FrontPage
 Web、種類, 420
 インストールの準備, 423
 インストールパラメータ, 430
 拡張機能、CGI プログラム, 419
 拡張機能のダウンロード, 422
 サーバ拡張、インストール, 424
 セキュリティについて, 421
 ドメイン名, 421

FTS_Author, 257
FTS_CreationDate, 257
FTS_Creator, 257
FTS_Keywords, 258
FTS_ModificationDate, 257
FTS_Producer, 258
FTS_Subject, 257
FTS_Title, 257

G

GET, 182, 398
 SNMP メッセージ, 240
GIF、定義, 438
givenName, 69
-group, 430
groups-with-users, 194

H

HEAD, 182, 398
home.html, 365
Host, 399
host, 430
HP OpenView ネットワーク管理ソフトウェア
 SNMP との使用, 215
.htaccess
 .nsconfig ファイルからの変換, 193
 magnus.conf による有効化, 192
 サポートされる指令, 195
 セキュリティに関する注意事項, 199
 動的構成ファイル, 190
 ユーザインタフェースからの有効化, 191
 例, 195
htaccess-register
 独自の認証メソッドを作成するための関数
 , 194
htconvert, 193
HTML, 415
 サーバによる解析、設定, 372

- 定義, 438
- パターンファイル, 283
- 文字エンティティ, 289
- html_doc, 259
- HTML コレクション
 - デフォルトの属性 (Title、Sourcetype), 258
- HTML、サーバで解析
 - ファイルキャッシュ, 154
- HTTP
 - 1.1 に準拠, 398
 - 応答, 399
 - ステータスコード, 399
 - 定義, 438
 - 要求, 398
- http_head, 183
- httpacl, 166
- httpacl ディレクトリ, 37
- HTTPD, 438
- httpdconfdir, 430
- HTTP(HyperText Transfer Protocol)
 - 概要, 397
- HTTPS
 - 定義, 438
- https-admserv ディレクトリ, 37
- HttpServerAdmin, 308
 - control コマンド, 385
 - create コマンド, 387
 - delete コマンド, 391
 - list コマンド, 394
 - 仮想サーバの設定, 384
 - 構文, 384
- https-server_id.domain, 37
- HTTP エンジン、アーキテクチャの概要, 32
- Hypertext Transfer Protocol HTTP/1.1 仕様
 - URL 参照, 398

I

- ibm367, 369
- ibm819, 370
- include ディレクトリ, 38
- INDEX, 182

- index.html, 365
- inetOrgPerson、オブジェクトクラス, 68
- 「Info」アクセス, 183
- INIT, 237
- init-clf, 210
- InitFn, 123
- inittab, 90, 147, 149
 - サーバの起動, 147
 - サーバの再起動, 148
 - 定義, 438
 - 編集, 148
- InstanceID, 257
- iplanetReversiblePassword, 165
- iplanetReversiblePasswordobject, 165
- iPlanet Web サイト
 - URL (http
//docs.iplanet.com), 25
- IP アドレス
 - アクセスの制限, 158
 - 定義, 438
- IP アドレスとホスト名
 - 指定, 180
- IP アドレスベースの仮想サーバ, 302
- is
 - 検索タイプのオプション, 72
- ISAPI.dll, 426
- ISINDEX, 356
- isn't
 - 検索タイプのオプション, 72
- iso_646.irv
 - 1991, 369
- iso_8859-1, 370
 - 1987, 370
- iso-2022-jp, 369
- iso646-us, 369
- iso-8859-1, 369
- iso-ir-100, 370
- iso-ir-6, 369
- issuerDN, 121
- IWS_SERVER_HOME
 - HttpServerAdmin の実行, 384
 - 環境変数, 340
- iwsInstanceContact, 225

iwsInstanceCount2xx - 5xx, 225
iwsInstanceCountOther, 226
iwsInstanceDeathCount, 225
iwsInstanceDescription, 225
iwsInstanceEntry, 225
iwsInstanceId, 225
iwsInstanceIndex, 225
iwsInstanceInOctets, 225
iwsInstanceLocation, 225
iwsInstanceOrganization, 225
iwsInstanceOutOctets, 225
iwsInstanceRequests, 225
iwsInstanceStatus, 225
iwsInstanceStatusChange, 228
iwsInstanceTable, 224
iwsInstanceUptime, 225
iwsInstanceVersion, 225
iwsListenAddress, 228
iwsListenEntry, 228
iwsListenId, 228
iwsListenIndex, 228
iwsListenPort, 228
iwsListenSecurity, 228
iwsListenTable, 227
iwsProcessConnectionQueueCount, 227
iwsProcessConnectionQueueMax, 227
iwsProcessConnectionQueueOverflows, 227
iwsProcessConnectionQueuePeak, 227
iwsProcessConnectionQueueTotal, 227
iwsProcessEntry, 227
iwsProcessId, 227
iwsProcessIndex, 227
iwsProcessKeepaliveCount, 227
iwsProcessKeepaliveMax, 227
iwsProcessTable, 227
iwsProcessThreadCount, 227
iwsProcessThreadIdle, 227
iwsThreadPoolEntry, 228
iwsThreadPoolIndex, 228
iwsThreadPoolTable, 228
iwsVsCount2xx - 5xx, 226
iwsVsCountOther, 226

iwsVsEntry, 226
iwsVsId, 226
iwsVsIndex, 226
iwsVsInOctets, 226
iwsVsOutOctets, 226
iwsVsRequests, 226
iwsVsTable, 226

J

Java

ガイド付き検索インタフェース, 269

Java Runtime Environment (JRE), 338
パスの設定, 63

JavaServer Pages

アーキテクチャの概要, 33
概要、インストール方法, 337

Java 仮想マシン (JVM)
実行環境, 33

Java サブレット
アーキテクチャの概要, 33

Java サブレット API, 337

Java サブレットおよび JavaServer Page
サーバ拡張、概要, 33

JDK

ダウンロード, 339

JDK (Java Development Kit)
ダウンロードの場所, 63
パスの設定, 63

JDK、JRE パス
切り替え, 63

JRE、JDK パス
切り替え, 63

JSP

API のリファレンス, 338
Web サーバで実行するための要件, 338
概要、インストール方法, 337
キャッシュディレクトリ, 345
サーバ拡張、概要, 33
バージョンファイルの削除, 345

JVM

属性、構成, 344

K

keepOldValueWhenRenaming パラメータ, 75
Keywords, 257

L

l, 122
Language List、Preferred
管理, 86
Language ヘッダー、Accept
使用, 412
Last-modified, 401
latin1, 370
LDAP
クライアント証明書をマップする, 119
検索結果、の表, 119
ディレクトリサービスの構成, 61
ユーザインタフェースでのデータベースの指定
, 201
ユーザとグループの管理, 65
ユーザ名とパスワードによる認証, 159, 445
LDAP (Lightweight Directory Access Protocol)
ユーザとグループの管理, 65
ldapmodify
Directory Server コマンド行ユーティリティ
, 67
Directory Server ユーティリティ, 73
「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, 79
エントリの修正, 383
説明, 383
LDAP 検索
certmap.conf を使用する, 119
LDAP 検索フィルタ, 78
LDAP ディレクトリ、アクセス制御, 180
LDIF
インポートとエクスポート機能、について, 67
エントリ、書式設定, 383
エントリ、説明, 383
データベースエントリの追加, 67
libdigest-plugin.ldif, 163
libdigest-plugin.lib, 163

libnssckbi.sl, 99
libnssckbi.so, 99
lib ディレクトリ, 38
LICENSE.txt, 39
Limit, 197
LimitExcept, 198
「list」アクセス, 183
loadbal ディレクトリ, 38
load-modules, 156
log_anly, 211
log_anly ディレクトリ, 36
logs ディレクトリ, 37
logs ファイル, 39
LogVsId、有効, 209
Look Within ディレクトリ
含まれているユーザエントリをすべて表示する
には, 73

M

magnus.conf, 37, 110
.htaccess の有効化, 192
AcceptTimeout, 162
ACLCacheLifetime 指令, 166
起動時のグローバル変数設定, 152
言語設定, 412, 413
構成ファイル、概要, 35
終了タイムアウト, 146
スレッド制限の調整, 151
セキュリティについて, 109
magnus.conf.cf.filter, 37
MAIL, 415
mail, 69, 122
Manage Servers
Server Manager、設定変更リスト, 41
manual ディレクトリ, 38
MANY 検索, 275
MATCHES, 278, 414
MaxProcs, 221
MaxThreads, 156
MD5、定義, 439

- memberCertDescriptions, 76
- memberURLs, 76
- META タグ, 258
- MIB
 - 場所、Netscape、iPlanet, 224
- MIB (management information base)
 - 管理対象オブジェクトの定義, 224
 - 場所、Netscape、iPlanet, 224
- MIME
 - charset パラメータ, 369
 - octet-stream, 350
 - 仮想サーバ、設定, 330
- mime, 390
- mime.types, 37
 - 構成ファイル、概要, 35
- MIME (Multi-purpose Internet Mail Extension) タイプ
 - 定義およびページへのアクセス, 153
- MIME タイプ
 - デフォルトの指定, 366
- MIME、定義, 440
- MinThreads, 156
- MKDIR, 183
- MMappedSessionManager, 345
- MMapSessionManager, 37, 38
- modutil
 - PKCS#11 モジュールをインストールする, 112
- MortalityTimeSecs, 151
- MOVE, 183
- MTA
 - 定義, 440
- my_stuff
 - アクセス制御, 169

N

- NativePool, 155
- ndex_page, 342
- NEAR, 278, 415
- NEAR/N, 279, 415
- netscape-http.mib, 224
 - 管理対象オブジェクトと説明, 224
- NEWS, 415
- NIS、定義, 440
- NMS 主導の通信, 240
- NNTP
 - 定義, 440
- nobody ユーザアカウント, 55
- nonce, 163
- NOT, 279, 415
- not, 408
- nsacl ディレクトリ, 38
- NSAPI
 - アーキテクチャの概要, 33
 - サーバ拡張機能、概要, 33
- NSAPI (Netscape Server Application Programming Interface)
 - アーキテクチャの概要, 33
 - サーバ拡張機能、概要, 33
- NSAPI エンジン
 - 実行環境, 33
- nsapi ディレクトリ, 38
- NS-collection=\$NS-collection, 285
- NS-collection-acl-check, 272
- NS-collection-alias, 291
- .nsconfig ファイル
 - .htaccess ファイルへの変換, 193
- ns-cron.conf, 37, 60
- NS-date-input-format, 290
- NS-date-time, 290
- NS-default-html-title, 290
- NS-display-select, 291
- NS-doc-root, 291
- nsfc.conf
 - ファイルキャッシュ設定, 154
- NS-highlight-end, 291
- NS-highlight-start, 291
- NS-HTML-descriptions-pat, 290
- NS-language, 291
- NS-largest-set, 290
- NS-max-records, 284, 290
- NS-ms-tocend, 290
- NS-ms-tocstart, 290
- NS-query, 284

NS-query.pat, 283
NS-query-pat, 290
NS-record-pat, 291
nssckbi.dll, 99
NS-search-page, 286
NS-search-type, 290
NS-tocend-pat, 291
NS-tocrec-pat, 291
NS-tocstart-pat, 291
NS-url-base, 291
NTFS ファイルシステム
 パスワードの保護, 90
NumPages, 257

O

o, 122
obj.conf, 37, 62, 210, 404
 ACL ファイルの参照, 410
 仮想サーバ, 299
 構成ファイル、概要, 35
 サービス品質使用のための SAF の設定, 219
 スタイルの削除, 379
 デフォルト認証, 159
obj.conf.clfilter, 37
octet-stream, 350
OpenView、HP ネットワーク管理ソフトウェア
 SNMP ユーザ, 215
OR, 279, 415
or, 408
order, 198
organizationalPerson、オブジェクトクラス, 68
OR 検索, 275
ou, 122

P

PageMap, 258
password.conf, 90, 150
PathCheck, 191, 192, 410

 鍵サイズ制限, 125
PermanentID, 257
person、オブジェクトクラス, 68
PHRASE, 279, 415
PHRASE 検索, 275
pk12util
 証明書と鍵をインポートする, 113
 証明書と鍵をエクスポートする, 112
PKCS#11
 modutil を使用してインストールする, 112
 pk12util で証明書と鍵をインポートする, 113
 pk12util で証明書と鍵をエクスポートする
 , 112
 モジュール、追加する, 111
plugins ディレクトリ, 38
pool パラメータ, 156
Posix による日付と時刻の書式, 252
POST, 182, 398
PR_Recv()/net_read, 221
PR_Send()/net_write, 221
PR_TransmitFile, 221
pragma no-cache, 131
Preferred Language List
 管理, 86
「Product Support」 ページ
 <http://iplanet.com/support>, 25
PROTOCOL_FORBIDDEN, 126
Public Key Cryptography Standard (PKCS) #11
 モジュール、追加する, 111
PUT, 183, 398

Q

qos-error、Error, 220
qos-handler、AuthTrans, 220
query.pat, 283
QueueSize, 156

R

RAM

定義, 440

rc.2.d, 440

サーバの起動, 147

rc.local, 90

README.txt, 39

「Read」アクセス, 182

record.pat, 283

Referer, 399

REG_DWORD, 150

REQ_ABORTED, 126

REQ_NOACTION, 126

REQ_PROCEED, 126

require, 199

RequireAuth, 194

restart ファイル, 39

RestrictAccess, 194

RMDIR, 183

root

サーバと, 55

定義, 440

RqThrottleMinPerSocket, 151

S

SAF サンプル

場所, 221

sagt, 232

sagt、プロキシ SNMP エージェントを起動するコマンド, 232

search ディレクトリ, 38

secret-keysize, 126

Secure Sockets Layer (SSL)

暗号化通信プロトコル, 105

server.xml, 110, 200, 298

構成ファイル、概要, 35

Server、Administrator

シャットダウン, 53

servercertnickname, 115

Server Manager

Server Manager、設定変更リスト, 41

UI の概要, 31

アクセス, 41

紹介, 41

スレッド制限の調整, 151

追加タブのリスト, 41

ログアナライザの実行 (使用する前にサーバログをアーカイブ), 211

servers.lst, 37

servlets ディレクトリ, 38

SessionData, 37, 345

SessionData ディレクトリ, 38

SessionData ファイル, 39

SET

SNMP メッセージ, 240

setup ディレクトリ, 38

sjis, 416

SMUX, 229, 233

sn, 69

SNMP

AIX デーモンの構成, 233

GET メッセージと SET メッセージ, 240

基本, 223

コミュニティー文字列, 239

コミュニティー文字列、設定, 239

サーバでの設定, 229

サーバの状態をリアルタイムでチェックする, 215

サブエージェント, 223

デーモン

再起動, 232

トラップ, 239

トラップ送信先、構成, 239

ネイティブデーモン

再起動, 232

再構成, 233

プロキシエージェント, 231

インストール, 231

起動, 232

プロキシエージェント、インストール, 231

プロキシエージェント、起動, 232

マスターエージェント, 223

インストール, 231, 233, 234

起動, 237

- 手動で構成, 235
- マスターエージェント、インストール, 233
- マスターエージェント、起動, 237
- snmpd.conf, 233
- snmpd、ネイティブ SNMP デーモンを再起動する
コマンド, 232
- snmp ディレクトリ, 38
- SNMP マスターエージェント
有効化および起動する, 234
- SOCKS、定義, 441
- sounds like
検索タイプのオプション, 72
- SourceType, 257, 258
- SSL
 - Administration Server で有効にする, 105
 - 仮想サーバで使用する, 306
 - 定義, 441
 - 認証, 162
 - の準備, 127
 - パラメータ、仮想サーバの接続グループごとに
1つのセット, 319
 - 有効, 108
 - 有効にするのに必要な情報, 92
- SSL 有効サーバ
自動起動の手順, 90
- SSL 2 プロトコル, 108
- SSL2 プロトコル, 104
- SSL3SessionTimeoutSSL
指令, 111
- SSL 3 プロトコル, 104, 108
- SSL3 プロトコル, 104
- SSLCacheEntries
指令 (SSL), 111
- SSLPARAMS, 110, 115
- SSLSessionTimeout (SSL)
セキュリティ指令, 110
- SSL 構成ファイル指令
値を設定する, 110
- SSL 認証メソッド, 405
- st, 122
- StackSize, 156
- startconsole ファイル, 39
- STARTS, 279, 415

- starts with
検索タイプのオプション, 72
- startsvr.bat, 37, 38
- start ファイル, 39
- stats-xml, 216
- STEM, 279, 415
- STEM 検索, 275
- stopsvr.bat, 37, 38
- stop コマンド
Administration Server のシャットダウン, 54
- stop ファイル, 39
- style.stp, 249
- Subject, 257
- SUBSTRING, 280, 415
- sysContact, 235, 236
- sysLocation, 235, 236

T

- telephoneNumber, 69
- telnet, 441
- testacl, 410
- timeofday, 408
- Title, 258
- title, 69
- TLS
 - 有効, 108
- TLS (Transport Layer Security), 105
- TLS 暗号化プロトコル, 108
- TLS および SSL3 符号化方式
Netscape Navigator 6.0, 109
- TLS プロトコル, 104
- TLS ロールバックオプション
符号化方式 (MS Internet Explorer 5.0 および 5.5
の場合に使用), 109
- tocend.pat, 283
- tocrec.pat, 283
- tocstart.pat, 283
- Transport Layer Security (TLS)
暗号化通信プロトコル, 105

Triple DES 符号化方式, 116

U

uid, 69, 122

定義, 441

Uniform Resource Identifier (URI), 247

uniqueMembers, 76

UNIX プラットフォーム

Administration Server へのアクセス, 45

uri_path, 340, 342

URI (Uniform Resource Identifier), 247

URI、定義, 441

URL

Administration Server にアクセス, 40

SSL 有効サーバと, 109

エンコーディング, 285

定義, 442

マッピング、定義, 442

マッピング方法, 248

URL 転送

構成, 367

URL ホストベースの仮想サーバ, 302

us, 369

us-ascii, 369

user, 430

useradmin

仮想サーバ, 313

User-agent, 399

USERDB, 200

userdb ディレクトリ, 39

userdefs.ini, 281, 287

userdefs.ini ファイル, 254

userPassword, 69

V

verifycert, 122

VeriSign

証明機関, 91

VeriSign 証明書

インストール, 92

要求, 91

Virtual Server Manager

UI の概要, 31

アクセス, 308

vs_port, 340, 342

vs_urlhost, 340, 342

W

WaitingThreads, 151

WAR (Web Application Archive)

定義, 442

wdeploy ユーティリティ, 340, 442

web, 430

Web Servers へのアクセスを制限する

手順, 62

Web アプリケーションの導入, 340

web-apps.xml

使用, 339

WebBot 関数, 419

Web Publishing レイヤー、アーキテクチャの概要
, 32

Web、root, 420

Web アプリケーション

定義, 442

導入, 340

Web サーバ

アーキテクチャ、概要, 31

起動と停止, 146

機能, 30

コンポーネントオプション, 34

ソフトウェアモジュール, 31

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), 169

Web ソフトウェア

標準サポート, 30

WILDCARD, 280, 415

Windows CGI, 351

Windows NT
プログラム、CGI の概要、351
Windows NT プラットフォーム
Administration Server へのアクセス、46
WORD, 280, 415
「Write」アクセス、182
WWW-authenticate, 401
WXEVersion, 257

X

x509v3 証明書
属性、122
x-euc-jp, 369
x-mac-roman, 369
x-sjis, 369

あ

アーカイブ
ログファイル、60, 207
アーキテクチャ、概要、31
アカウント、ユーザ
変更、55
アクセス
Delete, 183
Execute, 183
Info, 183
list, 183
Read, 182
Web サイトへ、制限 (グローバルとシングルイ
ンスタンス)、169
Write, 182
アクセス、サーバ
制限、62, 153
アクセス制御
「administrators」グループ、58
IP アドレス、180
LDAP ディレクトリ、180
my_stuff ディレクトリ、169
解除、183

概要、158
カスタマイズされた式の作成、183
仮想サーバで使用する、307
機能の概要、30
拒否の場合の応答、184
公開情報ディレクトリ、構成スタイルを使用し
て制御、364
時刻による制限、183
説明、167
データベース、180
ファイル、166
プログラム、182
分散管理と、58
ホスト名、180
ホスト名と IP アドレス、158
メソッド (基本、SSL)、159
ユーザとグループ、158, 178
曜日による制限、183
リダイレクション、184

アクセス制御エントリ (ACE)、62, 153, 158

アクセス制御ファイル (ACL)
格納される場所、166

アクセス制御リスト
FrontPage, 421

アクセス制御リスト (ACL)、62, 153, 158

アクセス、制限
Web Server、手順、62

アクセスログ、209
仮想サーバ、構成、332
場所、203

アクセスログファイル、203, 204

構成、209
表示、59

アクセスログロテーション、60

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、111

アナライザ、ログ
実行 (使用する前にサーバログをアーカイブ)
、211

アプリケーション
クライアントサイド、335
サーバサイド、335

アプリケーション、サーバサイド
Web サーバで実行するタイプ、336
Web サーバへのインストール方法、336

アプリケーションサービス
アーキテクチャの概要, 33

暗号化
定義, 103

暗号化、双方向, 104

暗号化モジュール、外部
使用法, 111

い

移行する
証明書、Enterprise Server 3.x から Web Server
6.0 へ移行する, 98

イベント、表示 (NT), 213

イベントの表示, 213

イベントビューア, 213

イベント変数
トラップ, 224

インストール
CGI プログラム, 346
Directory Server, 30
マルチプルサーバ, 47
マルチプルサーバの実行, 47

インデックスファイルのサイズ、制限, 254

う

受け入れスレッド
仮想サーバ, 301

え

英数字以外の文字
検索, 281

エージェント
SNMP, 231

エクストラネット、定義, 443

エラー
応答のカスタマイズ, 368

エラー応答、カスタマイズ, 368

エラーログ, 206
仮想サーバ、構成, 332
表示, 60
例, 60

エラーログファイル, 203, 206
場所, 203

演算子
修飾, 276
照会、組み合わせ, 276
照会言語, 277
属性式, 408
中国語、日本語、韓国語の, 414
どれを使用するか, 276
ワイルドカード, 281

エンドユーザ
分散管理, 57

お

応答、HTTP, 399

応答データ, 401

応答ヘッダー, 400

オプション
インストール時に使用可能なコンポーネント
, 34

か

階層、ACL 承認文, 406

ガイドライン
破られにくいパスワードを作成する, 128

鍵
pk12util でエクスポートする, 112
pk12util を使用してインポートする, 113
定義, 104

鍵サイズ制限 (obj.conf 内の PathCheck 指令に基づ
く), 125

鍵データベースパスワード, 90

鍵ペアファイル
紹介, 89

- パスワード、変更する, 129
- 保護する, 130
- 拡張機能、サーバ
 - アーキテクチャの概要, 32
- 仮想サーバ, 309
 - ACL 設定の編集, 202
 - ACL の設定, 316, 330
 - CGI を使用する, 307
 - chroot ディレクトリを指定する, 133
 - Class Manager で設定を変更する, 329
 - control コマンド, 385
 - create コマンド, 387
 - defaultclass, 300
 - delete コマンド, 391
 - HttpServerAdmin、設定, 384
 - HttpServerAdmin の create コマンドによる作成, 390
 - iWS 4.x バージョンからの移行, 305
 - iWS の機能を使用する, 306
 - list コマンド, 394
 - MIME の構成, 330
 - obj.conf, 299
 - obj.conf ファイル, 35
 - SSL の使用, 306
 - useradmin, 313
 - useradmin を使用するように構成する, 314
 - Virtual Server Manager で設定を変更する, 326
 - web-apps.xml、使用, 339
 - Web アプリケーションになっていないサーバレットと JSP の導入, 344
 - アクセス制御, 200
 - アクセス制御を使用する, 307
 - アクセスログの参照, 204
 - アクセスログ、表示, 332
 - 受け入れスレッド, 301
 - エラーログの参照, 206
 - 関連付けるサービス、指定, 312
 - クラスごとに個別の構成情報を持つ, 298
 - クラス、作成, 299, 311
 - クラスの設定、編集または削除, 312
 - 公開ディレクトリ、使用するように構成, 362
 - 構成スタイルを使用する, 307
 - コンテンツ管理, 305
 - サービス品質、設定, 331
 - サービス品質の使用法, 217
 - 削除, 333
 - 作成, 325
 - 作成および編集, 307
 - 種類, 302
 - 紹介, 297
 - 証明書, 88
 - 信頼できる別の CA を要求する場合, 118
 - セキュリティ、構成, 330
 - セキュリティの発行, 109
 - 接続グループ, 301
 - 接続グループごとに 1 つの SSL パラメータセット, 319
 - 接続グループ、作成, 311
 - 設定, 298, 310
 - 待機ソケット, 300
 - 追加ドキュメントディレクトリの設定, 361
 - データベースへのアクセス, 200
 - デフォルト, 303
 - 同時接続、サービス品質, 222
 - 動的再構成, 309
 - 導入, 316
 - ドキュメント設定、変更, 365
 - 独自の CGI 属性の構成, 349
 - 複数の Web サーバの実行, 47
 - 変数の使用法, 309
 - ユーザが監視できるようにする, 313
 - 要求を処理するための選択プロセス, 304
 - 例、イントラネットホスティング, 320
 - 例、セキュリティ保護されたサーバ, 319
 - 例、デフォルトの構成, 317
 - 例、マスホスティング, 323
 - ログ、構成, 332
 - ログファイル, 305, 316
 - 仮想サーバのクラス
 - chroot ディレクトリを指定する, 132
 - HttpServerAdmin の create コマンドによる作成, 387
 - サービス品質の使用法, 217
 - スレッドプール, 156
 - カタカナ (全角と半角), 416
 - 漢字, 416
 - 管理グループ
 - 作成, 58
 - 管理者
 - 分散管理, 57
 - 管理者のユーザ ID (スーパーユーザ), 40
 - 管理対象オブジェクト, 224, 240
 - 管理、分散

有効化, 57
関連項目
管理, 81

き

起動コマンド
UNIX プラットフォーム, 45
機能、Web サーバ, 30
キャッシュ制御指令
設定, 373
キャッシュ、定義, 443
キャッシュディレクトリ, 345
共通ログファイル形式
サーバアクセスログ, 210
定義, 443
例, 204

く

クライアント
アクセスのリスト, 209
クライアントサイドアプリケーション, 335
クライアント証明書
LDAP へマップする, 119
認証, 160
クライアント証明書 API
カスタムプロパティを作成する, 123
クライアント認証
定義, 88
要求するための手順, 118
クラスタ
管理, 140
構成, 137
サーバ構成のガイドライン, 137
サーバの削除, 140
サーバの追加, 138
使用についての定義と見込みタスク, 135
使用のガイドライン, 136
情報の変更, 139
設定, 137

変数の追加, 141

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, 76
アクセスの制限, 158
エントリの削除, 80
管理, 77
グループメンバーリストへ追加, 80
検索, 77
削除, 82
名前の変更, 82
認証, 158
認証、ユーザ, 159
編集, 78
メンバーの追加, 79

グループ、静的

作成のガイドライン, 76
定義, 76

グループ、ユーザ について, 66

グローバルセキュリティパラメータ, 110

け

形式、検索オプション
のリスト, 72

言語

検索でサポートされる, 414
デフォルト、ユーザエントリ, 69

言語設定

magnus.conf, 412, 413

検索

Java ベースのガイド付きインタフェース, 269
style.stp, 249
Uniform Resource Identifier (URI), 247
URL エンコーディング, 285
URL マッピング, 247
アクセスの制御, 247
一致ドキュメントのリスト, 272
インタフェースのカスタマイズ, 281
インデックス作成に使用するメモリの制限
, 254
英数字以外の文字, 281
演算子、照会言語, 277

- 演算子、ワイルドカード, 281
 - オンとオフ, 250
 - 概要, 245
 - 拡張, 270
 - 強調表示ドキュメントの閲覧, 273
 - 結果, 271
 - 結果のソート, 272
 - 検索規則, 275
 - 構成, 251
 - 構成ファイル, 254
 - 構成ファイルの変数, 291
 - 構文、基本, 284
 - コレクションに固有の変数, 290
 - サポートされる言語のリスト, 414
 - 実行、基本ガイドライン, 267
 - 手動によるファイルの構成, 253
 - 使用, 245
 - 照会演算子、組み合わせ, 276
 - 照会演算子、使用, 274
 - 照会演算子、どれを使用するか, 276
 - 照会演算子の修飾, 276
 - 照会演算子を検索語として使用する, 276
 - 照会言語、デフォルトの想定, 275
 - 使用できる言語, 414
 - ストップワード, 249
 - 生成パターン変数, 292
 - 属性数の調整, 254
 - 中国語、日本語、韓国語, 414
 - 中国語、日本語、韓国語の照会演算子, 414
 - ドキュメント形式、日本語、韓国語、中国語, 415
 - ドキュメントのインデックス作成, 255
 - 日本語, 415
 - 派生語検索、無効にする, 276
 - パターンファイル、構成, 252
 - パターン変数, 293
 - パターン変数、使用, 287
 - パターン変数、ユーザ定義, 287
 - パラメータ、構成, 251
 - 引数、必要な, 286
 - 標準照会の実行, 268
 - ホームページ, 268
 - マクロ, 292
 - メモリの制限, 254
 - ユーザ定義のパターン変数, 289
 - ワイルドカード、使用, 280
 - 検索エンジン、アーキテクチャの概要, 32
 - 検索から除外する単語, 443
 - 検索規則, 275
 - 検索照会
 - カスタム、構築, 71
 - 検索属性オプション
 - のリスト, 71
 - 検索タイプのオプション
 - のリスト, 72
 - 検索、テキスト
 - 構成, 246
 - 検索フィールド
 - 有効エントリ, 70
 - 検索フィルタ
 - LDAP, 78
 - 検索フィルタ、LDAP
 - 等号 (=) を含む文字列, 71
 - 検索ベース (ベース DN)
 - ユーザ ID, 67
- ## こ
- 公開鍵, 88, 94
 - 公開情報ディレクトリ
 - アクセスを制御するための構成スタイルの使用, 364
 - 公開ディレクトリ
 - 構成, 362
 - 公開ディレクトリ (UNIX)
 - カスタマイズ, 362
 - 構成、仮想サーバ、インストーラ, 39
 - 構成、新規
 - 動的インストーラ, 36
 - 構成スタイル, 375
 - 仮想サーバで使用する, 307
 - カテゴリ、CGI のファイルタイプ, 376
 - カテゴリ、アクセスの制限, 377
 - カテゴリ、エラー応答, 376
 - カテゴリ、サーバが解析する HTML, 377
 - カテゴリ、デフォルトの照会ハンドラ, 376
 - カテゴリ、動的構成, 376
 - カテゴリ、ドキュメントのフッター, 376
 - カテゴリ、文字セット, 376

- カテゴリ、より強力なセキュリティを要求 , 377
 - カテゴリ、リモートファイル操作, 376
 - カテゴリ、ログ設定, 376
 - 削除, 379
 - 作成, 375
 - 編集, 378
 - 割り当て, 377
 - 割り当ての一覧表示, 378
 - 構成、単一サーバ
 - インストール済みファイル, 36
 - 高性能
 - 機能の概要, 30
 - 構成ファイル
 - admpw、概要, 35
 - dblist.ini, 254
 - magnus.conf, 413
 - magnus.conf、概要, 35
 - magnus.conf、言語設定, 412, 413
 - mime.types、概要, 35
 - obj.conf, 379
 - obj.conf、概要, 35
 - 「Restore Configuration」ページでのバックアップコピー, 154
 - server.xml、概要, 35
 - SSL、値を設定する, 110
 - userdefs.ini, 254
 - アーキテクチャの概要, 34
 - 検索, 254
 - サーバの root に保存, 37
 - 動的、使用, 190
 - 変数, 289
 - 構成、複数サーバ、インストール, 40
 - 構文
 - ACL ファイル, 403
 - 検索関数、基本, 284
 - コマンド行
 - flexanlg を使用してアクセスログファイルを分析, 211
 - コミュニティー文字列
 - SNMP エージェントが認証に使用する文字列 , 239
 - コレクション
 - style.stp の変更, 250
 - インデックスの再作成, 264
 - 更新, 263
 - 構成, 261
 - 最適化, 264
 - 削除, 264
 - 作成、URL マッピング, 248
 - 新規、作成, 259
 - 属性, 258
 - 定義, 255, 443
 - 定期保守スケジュールの削除, 266
 - 定期保守のスケジュール設定, 265
 - 内容の表示, 273
 - について, 255
 - ファイルタイプ, 256
 - フィルタ, 257
 - 変換フィルタ, 258
 - 保守, 264
 - 保守スケジュールの削除, 266
 - 保守のスケジュール設定, 265
 - コレクション固有の変数, 290
 - コンテンツエンジン
 - ソフトウェアモジュール、Web サーバ, 32
 - コントロールパネル (Windows NT)
 - Administration Server のシャットダウンに使用 , 54
 - コンポーネントオプション
 - Web サーバのインストール時に使用可能, 34
- ## さ
- サーバ, 400
 - 4.x を 6.0 に移行する, 49
 - CA の種類, 117
 - LDAP ユーザとグループ、国際化についての考慮事項, 412
 - root ユーザ, 55
 - SNMP を介してリアルタイムでステータスをチェックする, 215
 - 一般的な機能、国際化についての考慮事項 , 411
 - 監視で利用可能な統計情報の種類, 216
 - 起動, 147, 149
 - 起動時のユーザアカウント, 55
 - 起動と停止, 146
 - クラスタから削除, 140
 - コントロールパネルを使用して起動, 149

- 再起動 (NT), 149
- 再起動 (UNIX), 147
- 再起動の時間間隔、変更、150
- 削除、48
- 自動的に起動、148
- 手動による再起動 (UNIX), 148
- 手動による停止 (UNIX), 149
- 停止、149
- 複数インストール、47
- ポート番号 1024, 55
- リモート、クラスタの追加、138
- ログ (ログアナライザを実行する前にアーカイブ), 211
- サーバアクセス
 - 制限、62, 153
- サーバインスタンス
 - 追加、48
- サーバ拡張
 - ソフトウェアモジュール、Web サーバ、32
- サーバサイドアプリケーション、335
 - Web サーバで実行するタイプ、336
 - Web サーバへのインストール方法、336
- サーバ主導の通信、241
- サーバ設定
 - アクセス、55
- サーバデーモン、定義、443
- サーバ認証
 - 定義、88
- サーバの起動、147, 149
 - 必要なユーザアカウント、55
- サーバの停止、149
- サーバ、複数稼働する
 - 仮想サーバで使用する、47
 - サーバの複数のインスタンスを使用、47
- サーバルート、定義、444
- 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、221
 - 仮想サーバ、設定、331
 - 使用、217
 - 使用時の obj.conf の SAF の設定、219
 - 同時接続、仮想サーバ、222
 - 例、218
- サブレット
 - API のリファレンス、337

- Web サーバで実行するための要件、338
- アクセスの例、342
- 概要、インストール方法、337
- キャッシュディレクトリ、345
- サーバ拡張、概要、33
- サーバへのインストール、方法、336
- バージョンファイルの削除、345
- サブレットと JSP
 - Web アプリケーションになっていないものを導入する、344
- 再起動ユーティリティ、自動 (NT), 150
- 再計算の間隔、217
- 削除
 - Web アプリケーション、340
- 作成、183
- サブ Web、420
- サブエージェント
 - SNMP、223
 - SNMP、使用可能にする、240

し

- シェル CGI、354
- シェルプログラム
 - CGI のインストール、Windows NT, 354
- 時間間隔、サーバの再起動
 - 変更、150
- 式、カスタム、183
- 式、属性
 - 演算子、408
- 識別名
 - LDAP エントリに証明書をマップする、119
 - ユーザの、フォーム、68
- 識別名 (DN) 属性
 - 定義、66
- システムの実行制御スクリプト
 - サーバの再起動、148
- 実行可能ファイル、ダウンロード、350
- 実行環境
 - Java、338
 - ソフトウェアモジュール、Web サーバアーキテクチャの概要、33

- 自動再起動ユーティリティ (NT), 150
 - 終了タイムアウト
 - magnus.conf, 146
 - 設定, 146
 - 照会, 275
 - 英数字以外の文字, 281
 - 演算子、組み合わせ, 276
 - 演算子、修飾, 276
 - 演算子、使用, 274
 - 演算子、どれを使用するか, 276
 - カスタム構築, 71
 - 検索語としての演算子, 276
 - 中国語、日本語、韓国語における演算子, 414
 - 標準検索の実行, 268
 - ワイルドカード、使用, 280
 - 照会言語
 - 演算子, 277
 - 検索、デフォルトの前提条件, 275
 - 照会ハンドラ
 - 使用, 356
 - 詳細設定、ログ
 - 設定, 209
 - 承認文、ACL, 405
 - 証明機関
 - VeriSign, 91
 - 定義, 88
 - 利用可能な CA のリストを入手する, 92
 - 証明書
 - certmap.conf と, 120
 - Enterprise Server 3.x を Web Server 6.0 に移行する, 98
 - iPlanet Web Server 4.x からアップグレードする, 98
 - pk12util でエクスポートする, 112
 - pk12util を使用してインポートする, 113
 - x509v3、属性, 122
 - 移行する, 98
 - 仮想サーバ, 88
 - 管理, 100
 - 組み込みルート証明書モジュールの使用, 99
 - クライアント、LDAP へマップする, 119
 - クライアントマッピング
 - 例, 123
 - 種類, 95
 - 紹介, 88
 - 信頼できる, 96
 - 接続グループの名前を選択する, 114
 - 他のサーバ、インストールする, 96
 - 他のサーバ証明書を要求する, 94
 - 単一、信頼データベース、各 Web サーバインスタンスごと, 118
 - ルート、削除する, 99
 - ルート、復元する, 99
 - 証明書、クライアント
 - 認証, 160
 - 証明書チェーン
 - 定義, 96
 - 証明書マッピングファイル
 - certmap.conf の格納場所, 120
 - certmap.conf の構文, 120
 - 証明書要求、必要な情報, 92
 - 除外する単語
 - 検索, 249
 - 所有者
 - 管理, 81
 - 指令
 - SSL3SessionTimeoutSSL, 111
 - SSLCacheEntries (SSL), 111
 - SSLSessionTimeout (SSL), 110
 - 言語設定, 413
 - シンボリック (ソフト) リンク
 - 定義, 371
 - シンボリックリンク、制限, 371
 - シンボリックリンクの制限, 371
 - 信頼データベース
 - Web サーバインスタンスごとに 1 つの証明書, 118
 - 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, 115
 - 作成, 89
 - パスワード、変更する, 129
 - 信頼できる証明書, 96
- ## す
- スーパーユーザ
 - 管理者のユーザ ID, 40
 - 分散管理, 57

- スーパーユーザ設定
 - 変更, 56
- スーパーユーザ、定義, 444
- スタイル
 - 構成, 375
- スタイル、構成
 - 作成, 375
- ステータスコード
 - HTTP, 399
- ストップワード, 444
 - 検索しない単語の決定, 249
- スレッド制限、調整, 151
- スレッドプール
 - 追加時に指定する情報, 155

せ

- 制御、アクセス
 - 概要, 158
- 生成パターン変数, 292
- 静的グループ
 - 作成のガイドライン, 76
 - 定義, 76
- セキュリティ
 - .htaccess、注意事項, 199
 - FIPS-140 を有効にする, 116
 - FrontPage, 421
 - magnus.conf のグローバルパラメータ, 110
 - 新しい待機ソケットを作成するときに有効にする, 106
 - 新しい待機ソケットを編集するときに有効にする, 106
 - 仮想サーバ、構成, 330
 - 機能の概要, 30
 - 強化, 127
- セキュリティ指令, 110
- セキュリティとアクセス制御
 - アプリケーションサービスの概要, 33
- セッション管理サービス
 - アプリケーションサービスの概要, 33
- 接続グループ
 - 1つのグループ内のすべての仮想サーバに対して1つのSSLパラメータセット, 319

- HttpServerAdmin の create コマンドによる作成, 388
- 仮想サーバで作成, 311
- 証明書名を選択する, 114
- 待機ソケット, 301
- 要求を処理するために選択, 304
- 設定、スーパーユーザ
 - 変更, 56

そ

- 双方向の暗号化、符号化方式, 104
- 属性
 - JVM、構成, 344
 - x509v3 証明書, 122
 - コレクション検索用, 258
 - 最大数の調整, 254
 - 識別名 (DN), 66
 - フィルタ, 257
- 属性、検索オプションのリスト, 71
- 属性式
 - ACL、属性, 407
 - 演算子, 408
- 測定時間, 217
- 組織単位
 - 検索, 83
 - 削除, 85
 - 作成, 83
 - 名前の変更, 85
 - 編集, 84
- ソフトウェアモジュール、Web サーバ, 31
- ソフト (シンボリック) リンク
 - 定義, 371

た

- ダイアログボックス
 - デバッグ
 - 無効化, 151
- 待機ソケット
 - HttpServerAdmin の create コマンドによる作成

- , 389
- ls1, 152, 300
- ls1 (デフォルト待機ソケット), 54
- SSLPARAMS、1つのグループと複数のグループ, 111
- 仮想サーバ, 300
- セキュリティ機能を有効にする, 106
- 接続グループ, 301
- 設定、編集, 54
- 表, 152
- ダイジェスト認証, 162
 - ACLのためのサーバの処理手順, 163
- ダイジェスト認証プラグインインストール, 163
- ダイジェスト認証メソッド, 404
- タイムアウト、終了設定, 146
- タグ、META, 258
- 多言語についての考慮事項
 - LDAPユーザとグループ, 412
 - 一般情報, 411
- 単位、組織
 - 検索, 83
 - 削除, 85
 - 作成, 83
 - 名前の変更, 85
 - 編集, 84

つ

- 追加ドキュメントディレクトリ, 361

て

- ディレクトリ
 - 追加ドキュメント, 361
- ディレクトリサービス
 - 構成, 61
- ディレクトリサービスの詳細設定
 - 構成, 61
- データ、応答, 401

- データベース
 - 仮想サーバを介してアクセス, 200
- データベース、ACL, 180
- データベースエントリ
 - LDIFを使用して追加, 67
- データベース、信頼
 - 作成, 89
 - パスワード、変更する, 129
- データ、要求, 399
- デーモン
 - Cron 制御の使用, 60
 - SNMP
 - 再起動, 232
 - ネイティブ SNMP、再起動, 232
 - ネイティブ SNMP、再構成, 233
- テキスト検索
 - 構成, 246
- テクニカルサポート
 - http
 - //iplanet.com/support, 25
- デバッグダイアログボックス
 - 無効化, 151
- デフォルト待機ソケット (ls1), 54

と

- 統計情報
 - アクセス, 217
 - サーバの監視で利用可能な種類, 216
 - サービス品質の帯域幅はサーバが動的に再構成されると失われる, 222
 - トラフィック測定の設定, 217
- 同時接続
 - 仮想サーバ、サービス品質, 222
- 動的構成ファイル
 - 使用, 190
- 動的再構成, 309
 - 概要, 36
- 動的に生成されるように指定する, 282
- ドキュメント
 - アクセスされたドキュメントのリスト, 209
 - インデックス作成, 255
- ドキュメント形式

- 検索、日本語、韓国語、中国語、415
- ドキュメント設定
 - Accept Language ヘッダーの解析、367
 - インデックスファイル名、365
 - 仮想サーバ、設定、365
 - サーバのホームページ、366
 - ディレクトリのインデックス作成、366
 - デフォルトの MIME タイプ、指定、366
- ドキュメントディレクトリ
 - コンテンツ発行の制限、363
 - 追加、361
 - プライマリ、304
 - プライマリ (ドキュメントルート)、360
- ドキュメントフッター
 - 設定、370
- ドキュメントルート、304
 - 設定、360
- 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、132
- ドキュメントルートディレクトリのリダイレクト、132
- トップレベルドメイン機関、444
- ドメイン名
 - FrontPage、421
- トラップ
 - SNMP、239
 - イベント変数が含まれたメッセージ、224
- トラフィック
 - 設定、統計情報の計算、217

な

- 内部デーモンログ交換、207
- ナビゲーション
 - URL 経由で Administration Server にアクセス、40

に

- 認証
 - SSL、162

- クライアント証明書、160
- ホスト名、165
- ユーザとグループ、158
- 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、160
- 認証、クライアント
 - 要求するための手順、118
- 認証、クライアント、サーバ
 - 定義、88
- 認証、ダイジェスト、162
- 認証データベース、180
- 認証文、ACL 構文、404
- 認証、ホスト -IP、165
- 認証メソッド
 - htaccess-register を使用して独自に作成、194
 - 種類、179
- 認証メソッド (Basic)、404
- 認証、ユーザ - グループ、159、165

ね

- ネイティブ SNMP デーモン
 - 再起動、232
 - 再構成、233
- ネットワーク管理ステーション (NMS)、223

は

- バージョン管理
 - 属性、現在は使用されていない、269
- バージョンファイル
 - 削除、JSP とサーブレット、345
- 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、111
- ハードリンク、定義、371
- パスの構成
 - JRE (Java Runtime Environment) または JDK (Java Development Kit)、63

- パスワード
 - 作成のガイドライン, 128
- パスワードの保護
 - NTFS ファイルシステム, 90
- パスワードファイル, 445
 - 起動時に読み込み, 363
- パスワード、ユーザ
 - 変更または作成するには, 73
- 派生語
 - 検索、無効にする, 276
- パターンファイル
 - HTML, 283
 - 検索、構成, 252
- パターン変数
 - 検索, 293
 - 構成ファイル, 291
 - 使用, 287
 - ユーザ定義, 287, 289
- パターン変数、生成, 292
- パフォーマンス
 - サービス品質の使用法, 217
- パラメータ
 - 検索、構成, 251
- ハンドラ、照会
 - 使用, 356

ひ

- 引数
 - 検索、必要な, 286
- 日付と時刻の書式 (Posix), 252
- ビューア、イベント, 213
- 表記規則、このマニュアルで使用する, 23
- 表示, 206
- 標準
 - Web ソフトウェア、サポート, 30
- ひらがな, 416

ふ

- ファイル
 - certmap.conf, 120
 - アクセス制御, 166
- ファイル拡張子
 - CGI, 348
 - 定義, 445
- ファイルキャッシュ
 - 静的な情報をすばやく提供、サーバで解析される HTML の処理速度を向上, 154
- ファイルシステムサービス
 - アプリケーションサービスの概要, 33
- ファイル操作、リモート
 - 有効化, 364
- ファイルタイプ
 - 定義, 445
- ファイルの変数
 - 構成, 289
- ファイルをキャッシングする, 131
- フォーム、アクセスの制限, 182
- フォント、このマニュアルで使用する, 23
- 復号化
 - 定義, 103
- 符号化方式
 - Netscape Navigator 6.0 での TLS および SSL3, 109
 - TLS ロールバックオプション (MS Internet Explorer 5.0 および 5.5 の場合に使用), 109
 - オプションを設定する, 125
 - 定義, 104
- プライマリドキュメントディレクトリ、設定, 304
- 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), 360
- プロキシ SNMP エージェント, 231
 - インストール, 231
 - 起動, 232
- プロキシエージェント、SNMP, 231
 - インストール, 231
 - 起動, 232
- プログラム
 - CGI
 - サーバへの格納方法, 347

- アクセス制御, 182
- プロトコルデータユニット (PDU), 240
- プロパティ
 - カスタム、を作成する, 123
- 分散管理
 - Directory Server、必要な, 58
 - アクセス制御に必要な, 157
 - グループ
 - ACL と, 58
 - 有効, 57

へ

- ヘッダー、応答, 400
- ヘッダーとフッター, 282
- ヘッダー、要求
 - リスト, 398
- 変数
 - コレクション固有, 290
 - パターン、使用, 287
 - ファイル、構成, 289
- 変数、イベント
 - トラップ, 224
- 変数、グローバル
 - magnus.conf での設定, 152
- 変数、パターン
 - ユーザ定義, 287
- 変数、パターン、生成, 292

ほ

- ポート
 - セキュリティと, 131
- ポート (1024 未満)
 - サーバのユーザを指定する必要はない, 55
- ホスト -IP 認証, 165
- ホスト名
 - アクセスの制限, 158
 - 定義, 446
 - 認証, 165
- ホスト名と IP アドレス

- 指定, 180

ま

- マクロ, 292
- マスターエージェント
 - CONFIG ファイル、編集, 235
 - SNMP, 223
 - SNMP、インストール, 231, 233, 234
 - SNMP、起動, 237
 - SNMP、手動で構成, 235
 - SNMP、使用可能にして起動する, 234
 - 標準以外のポートでの起動, 237
- マスターエージェント、SNMP
 - インストール, 233
 - 起動, 237

め

- メールサービス
 - アプリケーションサービスの概要, 33

も

- 文字エンティティ
 - HTML, 289
- 文字セット
 - iso_8859-1, 370
 - us-ascii, 369
 - 変更, 369
- モジュール
 - PKCS#11、追加する, 111
- モジュール、ソフトウェア, 31

ゆ

- ユーザ
 - アクセスの制限, 158

- 管理, 70
- 認証, 158
- ユーザアカウント
 - nobody, 55
 - 変更, 55
- ユーザインタフェース
 - Administration Server、Server Manager、Class Manager、および Virtual Server Manager, 31
- ユーザエントリ
 - Directory Server, 68
 - 検索, 70
 - 削除, 75
 - 作成のガイドライン, 67
 - 新規作成, 68
 - デフォルト言語, 69
 - 名前の変更, 75
 - 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, 75
 - 変更, 73
- ユーザとグループ
 - ACL、指定, 178
 - LDAP を使用して管理する, 65
 - について, 66
- ユーザとグループの認証, 159, 165
 - ACL ユーザキャッシュに格納される結果, 166
- ユーザ認証データベース
 - dbswitch.conf で定義, 200
- ユーザの削除, 75
- ユーザのディレクトリ
 - 構成, 362
- ユーザのディレクトリ (UNIX)
 - カスタマイズ, 362
- ユーザパスワード
 - 変更または作成するには, 73
- ユーザライセンス
 - 管理, 74
- ユーティリティ、自動再起動 (NT), 150

よ

- 要求
 - HTTP, 398

- 要求 - ダイジェスト, 163
- 要求データ, 399
- 要求ヘッダー
 - リスト, 398

ら

- ライセンス
 - 管理, 74
- ライブラリ, 123

り

- リソース
 - 定義, 446
- リソースのワイルドカードのリスト, 174
- リソースピッカー
 - 概要, 42
 - 構成スタイル, 376
 - 図, 42
 - ワイルドカード, 42
- リダイレクション, 446
- リダイレクション (アクセス制御), 184
- リモートサーバ
 - クラスタの追加, 138
- リモートファイル操作
 - 有効化, 364
- 領域, 163
- リリースノート
 - <http://docs.ipplanet.com>, 25
- リンク管理
 - 属性、旧バージョンの, 269

る

- ルート Web, 420
- ルート証明書
 - 削除, 99

復元する, 99

ろ

ローテーション、アクセスログ, 60

ロギング

cookie、簡易, 210

ログ

アクセス, 209

ログ、アクセス

場所, 203

ログアナライザ

flexanlg、使用と構文, 211

コマンド行から実行, 211

実行 (使用する前にサーバログをアーカイブ)
, 211

ログ、エラー

場所, 203

表示, 206

ログ交換

Cron ベース, 208

内部デーモン, 207

ログの詳細設定

設定, 209

ログファイル

Linux OS での 2G バイトのサイズ制限, 204

アーカイブ, 60, 207

アクセス, 203, 204

エラー, 203, 206

オプションの指定, 59

仮想サーバ, 305, 316

共通形式, 210

構成, 209

柔軟な形式, 210

詳細設定, 209

ログファイル、アクセス

表示, 59

ログファイルの場所

admin/logs, 59

わ

ワード、ストップ

検索しない単語の決定, 249

ワイルドカード

? 演算子, 281

演算子, 281

リソースピッカー, 42

ワイルドカード、使用, 280

ワイルドカード、リソース

のリスト, 174