

管理者ガイド

iPlanet™ Application Server

Version 6.5

816-5273-01
2002 年 2 月

Copyright © 2002, Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, U.S.A. All rights reserved.

Sun Microsystems, Inc. は、この製品に含まれるテクノロジーに関する知的所有権を保持しています。特に限定されることなく、これらの知的所有権は <http://www.sun.com/patents> に記載されている 1 つ以上の米国特許および米国およびその他の国における 1 つ以上の追加特許または特許出願中のものが含まれている場合があります。

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

フォントテクノロジーを含む第三者のソフトウェアの著作権は Sun の提供者により保護されており、ライセンス許諾されています。

Sun、Sun Microsystems、Sun のロゴマーク、Java、Solaris、iPlanet、および 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。

すべての SPARC の商標はライセンスの基づいて使用され、米国およびその他の国における SPARC International, Inc. の商標もしくは登録商標です。SPARC の商標に関連する製品は Sun Microsystems, Inc. によって開発されたアーキテクチャに基づいています。

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

この製品には Apache Software Foundation (<http://www.apache.org/>) により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.

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

目次

はじめに	11
マニュアルの使用方法	11
このマニュアルについて	13
このマニュアルの構成	14
第 1 部「入門」	14
第 2 部「1 台の iPlanet Application Server の管理」	14
第 3 部「複数の iPlanet Application Server の管理」	15
マニュアルの表記規則	16
第 1 部 入門	19
第 1 章 基本管理タスクの実行	21
iPlanet Application Server Administration Tool (iASAT) について	22
iASAT による管理タスクの実行	24
コマンドラインツールについて	26
コマンドラインツールの使用	27
コマンドラインツールによる管理タスクの実行	27
iascontrol	27
iascontrol start の例	29
iascontrol stop の例	29
Solaris での環境変数の設定	31
iPlanet レジストリエディタについて	32
iPlanet レジストリエディタによる管理タスクの実行	33
Directory Server 内の IP アドレスを変更するには	42
ローカルコンフィグレーションファイルで IP アドレスを変更するには	43
複数の LDAP ドメインを構成するには	45

第 2 部 1 台の iPlanet Application Server の管理 49

第 2 章 サーバの稼動状況の監視	51
iPlanet Application Server の監視	51
プロセス属性の監視	52
プロセスデータをファイルに記録するには	57
プロセスデータプロットを変更するには	57
プロセスデータプロットを削除するには	58
イベント通知の受信	59
イベントについて	59
サーバがダウンした場合は	59
再起動数を越えた場合は	60
異常なクラスタが検出された場合は	60
イベントに関する電子メール通知を設定するには	60
イベント起動スクリプトを指定するには	62
第 3 章 iPlanet Application Server をサードパーティツールを使って監視するよう SNMP を設定する	63
SNMP について	63
SNMP 統計収集を有効にする	65
MIB (Management Information Base) について	66
MIB エントリの書式設定	66
サードパーティの SNMP 管理ソフトウェアを使って MIB を有効にする	67
マスターエージェントおよびサブエージェントの設定	68
SNMP マスターエージェントを設定するには	68
SNMP マスターエージェントを起動するには	70
SNMP 設定を確認するには	71
第 4 章 サーバメッセージの記録	73
ロギングサービスについて	73
記録するメッセージタイプを決定するには	74
ログメッセージの書式設定	75
iPlanet Application Server ログファイルの監視	76
出力ウィンドウの垂直スクロールバーを有効にするには	78
アプリケーションイベントのロギングについて	80
ログ送信先の決定	81
ログの送信先を指定するには	82
データベースへの記録	83
コンソールへの記録	85
サーバイベントのファイルへの記録	85
ログファイルの管理	86
HTTP リクエストのロギングについて	89

Web リクエストの記録方法	89
Web サーバリクエストを記録するには	90
DSync メッセージのロギングについて	92
DSync メッセージの記録方法	92
クラスタダンプファイルの書式設定	92
DSync ノードダンプファイルの書式設定	93
第 5 章 iPlanet Application Server の保護	97
セキュリティについて	97
このマニュアルの内容	98
LDAP とは	98
iPlanet Console とは	99
ユーザおよびグループの保存と管理	100
セキュリティの実装	100
J2EE アプリケーションの認証	101
C++ アプリケーションの認証	101
iPlanet Console による Directory Server へのエントリの追加	101
識別名 (DN) とは何か	101
SSL 認証と識別名	102
iPlanet Console を使ったデータベースエントリの変更	104
LDIF による Directory Server へのエントリの追加	104
LDIF エントリの書式化	105
ldapmodify によるデータベースエントリの変更	105
プログラムによるエントリの作成	106
アプリケーションコンポーネントへのアクセスに対する認証の設定	106
ロールベースの認証の設定 (J2EE アプリケーションの場合)	106
EJB および Servlets に関するロールの管理	107
アクセス制御リスト認証の設定 (C++ アプリケーションの場合)	109
アクセス制御リストの作成	109
アクセス制御リストの変更	112
Web サーバと Application Server 間の暗号化の有効化	114
ファイアウォールを使ったセキュリティ	116
ネットワークセキュリティの基本	117
ファイアウォールの種類	117
iPlanet Application Server アーキテクチャ	119
プロセス間通信プロトコル	120
エフェメラルポート	120
Transmission Control Protocol/Internet Protocol (TCP/IP)	120
IP マルチキャスト	121
User Datagram Protocol (UDP)	122
データチャネルの暗号化	123
iPlanet Application Server でのファイアウォールの設定	124
単独のファイアウォール	124

二重のファイアウォール - DMZ 設定	125
三重のファイアウォール - データベース保護を持つ DMZ	126
第 6 章 サーバリソースの可用性の向上	129
サーバプロセスの追加と調整について	130
リクエスト処理のスレッド数の調整	131
実行プロセスのリクエスト数の指定	134
Administration Server オプションの設定	137
実行時に EJB コンテナパラメータを指定する	137
最大エンジン再起動回数を指定する	139
国際化サポートを有効にする	140
JavaServer Pages (JSP) をキャッシュする	140
サーバおよびエンジンの最大シャットダウン時間を指定する	141
マルチプロセスのシングルスレッド環境の実現	142
Directory Server のフェールオーバーの設定	143
サプライヤ開始レプリケーションの設定	143
第 7 章 Web コネクタプラグインの設定	161
Web コネクタプラグインについて	161
Web サーバの手動設定	162
iPlanet Web Server の設定	163
Apache Web サーバの設定	165
Windows での Apache のインストール	168
iPlanet Application Server レジストリの設定	169
Microsoft Internet Information Server の設定	170
Web サーバが記録するための Web コネクタプラグインの設定	171
データベースフィールドへの HTTP 変数の割り当て	171
ログへの HTTP 変数の追加	172
cookie と隠しフィールドの設定	173
URL 書き換えの設定	174
CGI リクエストのための CGI フラグの設定	174
Web コネクタのポート番号の変更	175
アプリケーションコンポーネントへの入力のための HTTP 変数の指定	175
第 8 章 データベース接続の管理	179
データアクセスドライバについて	179
サードパーティ JDBC ドライバの設定	180
インストール時	181
インストール後	181
サードパーティ JDBC ドライバの設定	181
動作確認済みのサードパーティドライバ	182
コマンドラインツールを使ってサードパーティ JDBC ドライバを設定する	186

サードパーティ JDBC ドライバのレジストリ設定	190
ドライバのエントリ	190
データソースの設定	191
新しいデータソースの機能	191
データソースの登録	192
データソース設定パラメータ	201
データソースパラメータ	201
コネクションプールパラメータ	201
コネクション妥当性パラメータ	203
データソースレジストリ設定	204
サードパーティデータソースの変更	205
データソースの監視	208
現在の統計	209
累積統計	209
詳細な統計情報	210
プールを破壊	210
プールをフラッシュ	210
データソース XML フォーマット	211
第 9 章 トランザクションの管理	213
トランザクションマネージャについて	213
新しい機能	214
iASAT からの分散トランザクションの管理	215
「トランザクション」ウィンドウについて	215
サーバごとにトランザクションを設定する	216
サーバごとにトランザクションを表示する	217
サーバごとにトランザクションを管理する	219
プロセスごとにトランザクションを管理する	219
選択したプロセス上のトランザクションを表示する	219
プロセスごとにトランザクションを管理する	221
トランザクションの監視	222
設定パラメータ	222
インダウトトランザクションの解決	224
第 10 章 CORBA ベースクライアントサポートの有効化	227
Javasoftware ORB サポートの設定	227
ORBIX2000 サポートの設定	229
第 11 章 メッセージ駆動 Bean の管理	237
JMS プロバイダの設定	238
メッセージ駆動 Bean の管理	242

第 3 部 複数の iPlanet Application Server の管理 245

第 12 章 複数のサーバの設定	247
複数サーバ企業での Web コネクタプラグイン	247
複数サーバ向けの Web コネクタプラグインの設定	248
リクエストの送信先となるアプリケーションサーバの指定	249
ログの記録先となるアプリケーションサーバの指定	250
分散データ同期とロードバランス	251
分散データ同期環境の設定	251
マルチキャスト通信	252
ロードバランスでのマルチキャストの適用方法	252
第 13 章 複数サーバアプリケーションの管理	253
iASAT について	253
複数サーバ上でアプリケーションをローカルで管理する	254
複数サーバ上でパーティション分割したアプリケーションを管理する	256
パーティション分割されたアプリケーションの設定	256
アプリケーションコンポーネントの有効/無効を切り替える	257
ロードバランスを行うためにアプリケーションを管理および配置する	260
分散アプリケーションコンポーネントの属性を変更する	262
第 14 章 ユーザリクエストのロードバランス	265
ロードバランスの動作	265
ロードバランスの必要条件	266
固定 (スティッキー) ロードバランスとは	267
固定ロードバランスの適用	267
固定ロードバランスの有効化	268
固定ロードバランスと外部 Sticky cookie	269
ロードバランス方式の選択	270
Web コネクタプラグインによるロードバランス	270
iPlanet Application Server によるロードバランス	271
コンポーネントの応答時間によるロードバランス	271
サーバの応答時間によるロードバランス	272
ラウンドロビンロードバランス	273
ユーザ定義条件ロードバランス	277
「サーバ負荷クライテリア」のウェイト要因の調整	277
アプリケーションコンポーネントパフォーマンスクライテリアのウェイト要因の調整	279
更新およびブロードキャスト間隔の調整	282
ロードバランスのマルチキャストホストアドレスの変更	284
第 15 章 分散のデータ同期の管理	287
分散のデータ同期について	287

フェールオーバーがデータアクセスを維持する方法	288
クラスタについて	289
クラスタ内でのサーバロール	290
クラスタの通信方法	292
クラスタ内の情報の流れ	292
クラスタの設定	293
iASAT を使ったクラスタの設定	293
iPlanet レジストリエディタを使ったクラスタの設定	297
Sync Server の優先度の決定	302
クラスタパラメータの設定	304
クラスタへのシンクロナイザの割り当て	309
優先度を設定した Sync Server の調整	311
例 : 7 台のサーバで構成されるクラスタ内の調整	311
第 16 章 開発用 iPlanet Application Server の設定	315
クラスパスの設定	315
Java エンジンのヒープサイズの変更	316
デフォルトのアプリケーションパスの変更	317
ダイナミッククラス再読み込みの有効化	318
J2EE コンポーネントのセッションタイムアウト値とキャッシュタイムアウト値の指定	319
RMI/IIOP サポートの有効化	322
データベースの環境変数の設定	323
Sybase	323
Oracle	323
DB2	323
INFORMIX	324
環境サイズの調整	324
付録 A トラブルシューティング	325
クラスパスの設定	325
サーバサイドコネクション数の設定	325
索引	337

この章には次のトピックがあります。

- はじめに
- このマニュアルについて
- このマニュアルの構成
- マニュアルの表記規則

マニュアルの使用方法

次の表は、iPlanet Application Server のマニュアルおよび「リリースノート」に記載されているタスクと概念を示しています。特定のタスクを行う場合や特定の概念について調べる場合は、該当するマニュアルを参照してください。

注：印刷版マニュアルは、<http://docs.iplanet.com/docs/manuals/ias.html> から入手できます。

情報の内容	参照するマニュアル	添付されている製品
ソフトウェアおよびマニュアルの最新情報	リリースノート	iPlanet Web サイト http://doc.iplanet.com/docs/manuals/ias.html から入手可能
iPlanet Application Server およびそのコンポーネント (Web コネクタプラグイン、iPlanet Application Server Administrator) のインストールと、サンプルアプリケーションの設定	インストールガイド	iPlanet Application Server 6.5

情報の内容	参照するマニュアル	添付されている製品
<p>次のタスクによる、オープン Java 標準モデル (Servlet、EJB、JSP、および JDBC) に準拠した iPlanet Application Server 6.5 アプリケーションの作成</p> <ul style="list-style-type: none"> • アプリケーションのプレゼンテーション層および実行層の作成 • EJB (Enterprise JavaBeans) コンポーネントへのビジネスロジックの個別部分およびエンティティの配置 • JDBC を使ったデータベースと通信 • 反復テスト、デバッグなどアプリケーションの調整機能を使用した、正確かつ高速に動作するアプリケーションの生成 	開発者ガイド (Java)	iPlanet Application Server 6.5
<p>次のタスクを行うための、iPlanet Application Server Administrator Tool による 1 台または複数台のアプリケーションサーバの管理</p> <ul style="list-style-type: none"> • サーバの稼動状況の監視およびログ記録 • iPlanet Application Server へのセキュリティの実装 • サーバリソースの高利用度の実現 • Web コネクタプラグインの設定 • データベース接続の管理 • トランザクションの管理 • 複数のサーバの設定 • 複数のサーバでのアプリケーションの管理 • サーバのロードバランス • 分散データ同期の管理 • 開発用の iPlanet Application Server のセットアップ 	管理者ガイド	iPlanet Application Server 6.5

情報の内容	参照するマニュアル	添付されている製品
iPlanet Application Server に付属するオンラインバンクアプリケーションの移行サンプルを含む、Netscape Application Server バージョン 2.1 から新しい iPlanet Application Server 6.5 プログラミングモデルへのアプリケーションの移行	移行ガイド	iPlanet Application Server 6.5
C++ アプリケーションを作成する場合の iPlanet Application Server クラスライブラリの共有クラスとインターフェイス、およびそれらのメソッドの使用	Server Foundation Class Reference (C++)	別注文

このマニュアルについて

『管理者ガイド』では、1 台または複数台の iPlanet Application Server マシンの管理者として行う作業について説明します。このマニュアルでは少なくとも 1 台のマシンに iPlanet Application Server がインストールされていることが前提となります。iPlanet Application Server のインストールの詳細については、『インストールガイド』を参照してください。

ほとんどの管理作業は、サーバおよびアプリケーション管理用の GUI ベースのツールである iPlanet Application Server Administration Tool を使って行います。このツールについては、「iPlanet Application Server Administration Tool (iASAT) について」に記載されています。

このマニュアルの構成

このマニュアルは3部に分かれています。iPlanet Application Server マシンの管理に関する知識がない場合は、第1部入門を一読することをお勧めします。第1部には、サーバと Administration Tool を起動する方法についての概要が記載されています。アプリケーションサーバの管理に関する知識がある場合は、第1部にさっと目を通してから第2部「1台の iPlanet Application Server の管理」に進みます。

複数のアプリケーションサーバを管理する場合は、引き続き第3部「複数の iPlanet Application Server の管理」に進んでください。複数のサーバを備えた企業特有の情報について記載されています。

第1部「入門」

『管理者ガイド』の第1部では iPlanet Application Server の環境について記載しています。

第1部には次の章があります。

- 第1章「基本管理タスクの実行」では iPlanet Application Server Administration Tool を紹介し、iPlanet Application Server を設定するために行う基本的な作業を説明しています。

第2部「1台の iPlanet Application Server の管理」

『管理者ガイド』の第2部では、iPlanet Application Server マシンが1台の場合のサーバおよびアプリケーションの管理手順について説明します。

第2部には次の章があります。

- 第2章「サーバの稼動状況の監視」では、Executive Server、Java サーバ、および C++ サーバプロセスのさまざまな属性をチャートにすることができる監視サービスについて説明します。
- 第3章「iPlanet Application Server をサードパーティツールを使って監視するよう SNMP を設定する」では、SNMP (Simple Network Management Protocol) を設定する方法について説明します。サードパーティの SNMP 管理ツールを使って iPlanet Application Server を監視できます。
- 第4章「サーバメッセージの記録」では、iPlanet Application Server によって提供されるメッセージロギングサービスについて説明します。

- 第 5 章「iPlanet Application Server の保護」では、アプリケーションのセキュリティを確保するためにユーザとグループを設定する方法について説明します。
- 第 6 章「サーバリソースの可用性の向上」では、アプリケーションのパフォーマンスを向上させる方法について説明します。
- 第 7 章「Web コネクタプラグインの設定」では、iPlanet Application Server 上にあるアプリケーションにユーザのリクエストを送信する Web コネクタプラグインについて説明します。
- 第 8 章「データベース接続の管理」では、データアクセスドライバを設定し、設定値をデータベースコネクションパラメータに適用する方法について説明します。
- 第 9 章「トランザクションの管理」では、iPlanet Application Server Administration Tool を使ってトランザクションを管理するために必要な作業と概念について説明します。
- 第 10 章「CORBA ベースクライアントサポートの有効化」では、既存の JavaSoft ORB および ORBIX2000 というサードパーティ ORB を使用して CORBA 要素用の RMI/IIOP ORB を設定する方法について説明します。
- 第 11 章「メッセージ駆動 Bean の管理」では、JMS プロバイダの設定方法とメッセージ駆動 Bean サポートの管理方法について説明します。

第 3 部 「複数の iPlanet Application Server の管理」

『管理者ガイド』の第 3 部では、複数の iPlanet Application Server マシンを管理する方法について説明します。より複数のサーバを備えた企業に適用する、広範囲にわたる管理手順と概念を説明します。これらの手順は複数のサーバの管理だけに焦点を当てており、第 2 部で説明する 1 台のサーバの手順とともに使います。

第 3 部には次の章があります。

- 第 12 章「複数のサーバの設定」では、iPlanet Application Server Administration Tool を使って、複数の iPlanet Application Server マシンに対して Web コネクタプラグイン、分散データ同期、およびマルチキャスト通信を設定する方法について説明します。
- 第 13 章「複数サーバアプリケーションの管理」では、iPlanet Application Server Administration Tool を使って複数の iPlanet Application Server マシンを同時に管理する方法について説明しています。
- 第 14 章「ユーザリクエストのロードバランス」では、複数のアプリケーションサーバ間でリクエストのバランスを保つことによって各 iPlanet Application Server マシンのユーザリクエスト処理機能を最適化するロードバランスについて説明します。

- 第 15 章「分散のデータ同期の管理」では、iPlanet Application Server マシンをグループ化してデータ同期クラスタにする方法について説明します。
- 第 16 章「開発用 iPlanet Application Server の設定」では、アプリケーションを開発する目的で iPlanet Application Server をセットアップする方法について説明します。この章には、開発者向けの情報が記載されています。
- 付録 A「トラブルシューティング」には、iPlanet Application Server マシンのトラブルシューティングに関する情報が記載されています。

マニュアルの表記規則

このマニュアルでは次の表記規則を採用しています。

表 1

表記規則	説明
ファイルとディレクトリパス	Windows の場合 : ディレクトリパス名は円記号 (¥) で区切ります。(例 : Images¥Screens¥ser.gif)。 Solaris の場合 : ディレクトリパスはスラッシュで区切ります(例 : Images/Screens/ser.gif)。
メニューオプション	2 番目のレベルのメニューオプションは「より大きい」(>) 記号で区切ります。たとえば、Windows で「スタート」メニューからアプリケーションを起動するには次の手順を実行します。 「スタート」メニューから「プログラム」>「iPlanet Application Server 6.5」を選択します。その後、「iPlanet Administration Tool」を選択します。
URL	このマニュアルでは次のように URL 形式を使います。 <i>http://server.domain:port/path/file.html</i> ここで、 <ul style="list-style-type: none"> • <i>server</i> は、アプリケーションを実行しているサーバの名前です。 • <i>domain</i> は、インターネットのドメイン名です。 • <i>path</i> は、サーバ上のディレクトリ構造です。 • <i>file</i> は、個々のファイル名です。 注 : URL の斜体文字部分はプレースホルダです。

表 1 (続き)

表記規則	説明
フォント	<p>次の場合には、クーリエ (Courier) フォントを使います。</p> <ul style="list-style-type: none"> • サンプルコードとコードの一覧表 • APIと言語要素 (関数の名前、クラス名など) • ファイル名、パス名、ディレクトリ名、および HTML タグ
URL	<p>このマニュアルでは次のように URL 形式を使います。</p> <p><code>http://server.domain:port/path/file.html</code></p> <p>ここで、</p> <ul style="list-style-type: none"> • <i>server</i> は、アプリケーションを実行しているサーバの名前です。 • <i>domain</i> は、インターネットのドメイン名です。 • <i>path</i> は、サーバ上のディレクトリ構造です。 • <i>file</i> は、個々のファイル名です。 <p>注: URL の斜体文字部分はプレースホルダです。</p>
フォント	<p>次の場合には、クーリエ (Courier) フォントを使います。</p> <ul style="list-style-type: none"> • サンプルコードとコードの一覧表 • APIと言語要素 (関数の名前、クラス名など) • ファイル名、パス名、ディレクトリ名、および HTML タグ <p>次の場合には、イタリック体を使います。</p> <ul style="list-style-type: none"> • 変数とプレースホルダ

第 1 章「基本管理タスクの実行」

基本管理タスクの実行

この章では、iPlanet Application Server Administration Tool (iASAT)、iPlanet レジストリエディタ、およびコマンドラインツールを使って iPlanet Application Server を管理する方法について説明します。

また、この章では iPlanet Application Server Administration Tool (iASAT)、各コマンドラインツール、および iPlanet レジストリエディタを使って実行できる基本的な iPlanet Application Server 設定タスクについても説明します。

iASAT、コマンドラインツール、および iPlanet レジストリエディタを使って実行できる基本的な管理および設定タスクは、次のトピックに分類されます。

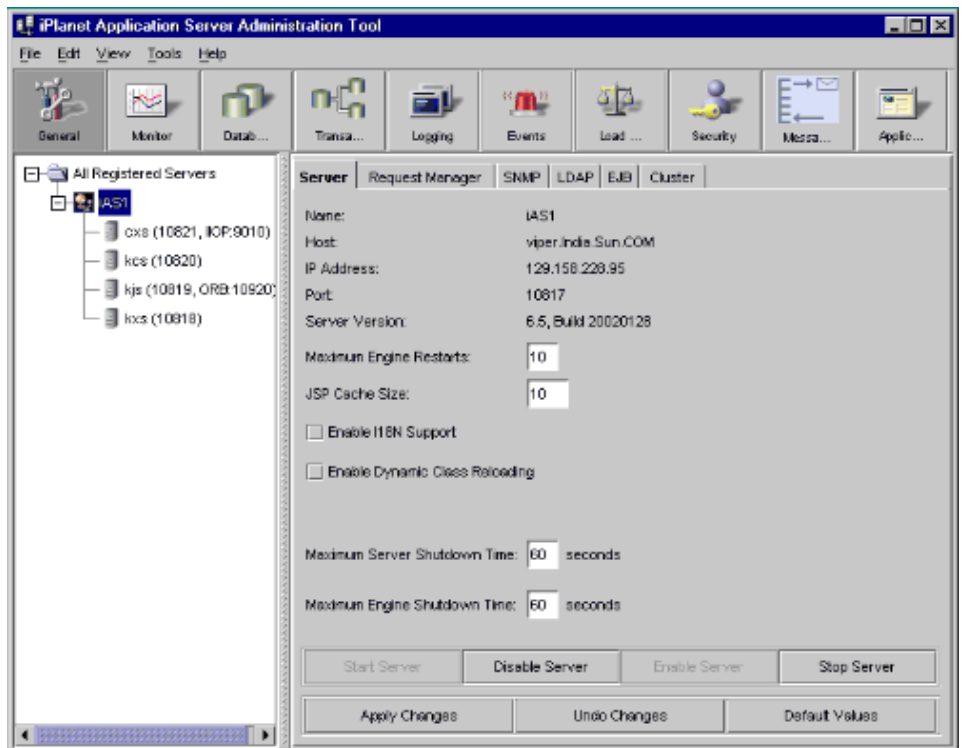
- iPlanet Application Server Administration Tool (iASAT) について
- iASAT による管理タスクの実行
- コマンドラインツールについて
- コマンドラインツールの使用
- コマンドラインツールによる管理タスクの実行
- Solaris での環境変数の設定
- iPlanet レジストリエディタについて
- 複数の LDAP ドメインを構成するには

iPlanet Application Server Administration Tool (iASAT) について

iPlanet Application Server Administration Tool (iASAT) は、グラフィカルユーザーインターフェイスを持つスタンドアロン Java アプリケーションであり、1 つまたは複数の iPlanet Application Server のインスタンスを管理できます。iPlanet Application Server の管理には、データベースコネクションスレッドおよびロードバランスパラメータの調整などのパフォーマンス関連のタスクがあります。また、サーバ管理者は、Web サーバなどのアプリケーションサーバが使うコンポーネントを個別に設定する必要があります。

アプリケーションコンポーネントの管理にも iASAT を使います。iASAT を使うと、アプリケーションコンポーネントをグループ化して有効にし、パーティションに分割できるので、パフォーマンスが向上します。iPlanet Application Server アプリケーションの中核となるアプリケーションコンポーネントはコードに含まれ、アプリケーションサーバに保存されます。Enterprise JavaBeans (EJB)、Servlet、JavaServer Pages (JSP)、および AppLogic オブジェクトはすべてアプリケーションコンポーネントです。それぞれの詳細は、『開発者ガイド』を参照してください。

管理タスクは iASAT を使って実行されます。iASAT のデフォルトの「一般」ウィンドウを開くと、ツールバー、左右にパネルがあるメインウィンドウ、およびメニューバーが次の図のように表示されます。



メインウィンドウの左側のパネルには、IASAT によって登録されたすべての iPlanet Application Server が表示されます。右側のパネルには、登録されている iPlanet Application Server の個々の機能が表示されます。

IASAT を起動するには

1 台または複数台の iPlanet Application Server マシンを管理するには、次のいずれかの方法で IASAT を起動します。

- Windows では、「スタート」メニューから、「プログラム」>「iPlanet Application Server 6.5」>「iPlanet Application Server Administration Tool」を選択します。
- Solaris システムでは、「Terminal」ウィンドウを開き、パス <IASInstallDir>/ias/bin/ に移動し、コマンドプロンプトに次のように入力します。

```
ksvradmin &
```

iASAT による管理タスクの実行

この節では、iASAT の使用に関連する概念およびタスクについて説明します。この節には次のトピックがあります。

- iPlanet Application Server を登録するには
- iPlanet Application Server の登録を解除するには
- iASAT によるサーバの起動と終了を行うには

iPlanet Application Server を登録するには

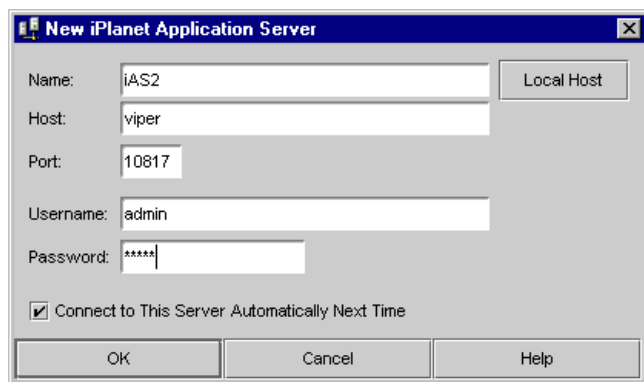
iPlanet Application Server を登録するには、サーバを Administration Tool の範囲に追加します。これは、サーバまたはサーバのグループを企業のネットワークに追加した後に行います。

iPlanet Application Server をインストールすると、ローカルホストは自動的に iASAT に登録されます。

注 管理する前に、iPlanet Application Server を iASAT に登録する必要があります。

iPlanet Application Server マシンを登録するには、次のタスクを実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「ファイル」メニューから、「新規」>「サーバー」をクリックします。
「新規 iPlanet Application Server」ダイアログボックスが表示されます。



3. 次のいずれかの手順を実行します。

- 「名前」テキストボックスでサーバの名前を指定します。
これはほかのアプリケーションサーバと区別するために使う任意の名前です。たとえば、Prod-Node-A などです。さらに、Prod-Node-B、Dev-Node などのアプリケーションサーバを設定できます。
 - 「ローカルホスト」をクリックしてローカルマシンで実行中のサーバを登録します。
サーバ名およびユーザのマシン名が自動的に入力されます。
4. 「ホスト」テキストボックスでサーバのホスト名を指定します。
これはユーザのサーバマシンの DNS 名です。IP アドレスを入力することもできます。
 5. 「ポート」テキストボックスで **Administration Server** のポート番号を指定します。
デフォルトインストールでは、このポート番号は 10817 に設定されます。
 6. ローカルホストを登録する場合は、サーバインストール時または「ユーザーおよびグループ」の変更時に入力したユーザ名とパスワードを、「ユーザー名」および「パスワード」テキストボックスで指定します。

登録しているサーバがローカルホストでない場合は、追加するマシンのユーザ名とログインパスワードを入力する必要があります。たとえば、ネットワーク上で使用可能な Solo という名前のマシンを登録するには、「ユーザー名」および「パスワード」テキストボックスに Solo のユーザ名とパスワードを入力する必要があります。
 7. (オプション) 常にこのサーバに接続して「Administration Tool」ウィンドウに表示するには、「次回このサーバに自動的に接続」チェックボックスをオンにします。
 8. 「OK」をクリックしてサーバを登録します。

iPlanet Application Server の登録を解除するには

アプリケーションサーバが使用不可能になったときに、そのアプリケーションサーバを Administration Tool から削除できます。

iPlanet Application Server インスタンスの登録解除または削除を行うには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで「すべての登録済みサーバ」をダブルクリックします。

すべての登録済みアプリケーションサーバが表示されます。
3. 削除するサーバを選択します。
4. 「編集」メニューから「削除」を選択します。

選択したサーバは iASAT の範囲から削除されます。

iASAT によるサーバの起動と終了を行うには

iPlanet Application Server をインストールするときにサーバの自動起動を選択できます。そのあと、iPlanet Application Server はシステムの起動時に自動的に起動します。ただし、iPlanet Application Server を手動で起動および停止する場合は、iASAT から次の手順を実行します。

1. iASAT ツールバーの「一般」メニューをクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、起動するサーバを選択します。
3. 「一般」ウィンドウの右側のペインで「サーバを起動」をクリックします。

サーバを停止するには、停止するサーバを選択して、「一般」ウィンドウの右側のペインで「サーバの停止」をクリックします。

注 サーバの実行時だけ、階層ツリーでサーバを展開できます。

コマンドラインツールについて

iPlanet Application Server には、さまざまなコマンドラインツールおよび実行可能ファイルがあり、これらはコマンドラインプロンプト (Windows) や、シェルプロンプト (Solaris) から実行可能です。

コマンドラインツールを使うと、基本設定からアプリケーションの配置まで、さまざまなタスクを直接実行できます。

コマンドラインツールの完全な説明を表示するには、コマンドプロンプトに `command -help` と入力します。たとえば、`iascontrol` コマンドで使用できるすべてのオプションの完全なリストを表示するには、コマンドプロンプトで「`iascontrol-help`」と入力します。

コマンドラインツールの使用

使いやすくするため、ほとんどのコマンドラインツールは iPlanet Application Server Administration Tool (iASAT) および iPlanet Application Server Deployment Tool (iASDT) と統合されています。

iPlanet Application Server のコマンドラインツールで使用可能なオプションは、Windows プラットフォームと Solaris プラットフォームでは異なります。

Solaris では、iASAT と iASDT が使用可能であっても、コマンドラインツールを幅広く使用できます。Solaris では iPlanet Application Server の複数のインスタンスを持つことができるので、特定のインスタンスのインストールディレクトリからコマンドラインツールを使用して、そのインスタンスのみの属性を変更することが時には必要になる場合があります。

Windows では、コマンドラインツールは実行可能 (.exe) ファイルの形式になっています。

Solaris システムと Windows システムのどちらも、コマンドラインツールはパス `iASInstallDir/ias/bin` にあります。

次の節では、iPlanet Application Server でのコマンドラインツールの使用について説明します。

コマンドラインツールによる管理タスクの実行

この節では、コマンドラインツールを使った iPlanet Application Server の停止手順と起動手順について説明します。

iascontrol

Windows システムと Solaris システムの両方で、次の作業を実行することによってコマンドラインツールによるサーバの停止と起動を行うことができます。

- Windows では、「スタート」メニューから「ファイル名を指定して実行」を選択します。
- 「ファイル名を指定して実行」ダイアログボックスのフィールドに「iascontrol」と入力します。
- Solaris システムでは、`iASInstallDir%ias%bin%` ディレクトリに移動します。

次のスクリプトを実行します。

```
iascontrol subcommand -instance <instance> -user <user> -password <password> -host <host> -port <port> -eng <engineId>
```

ここで、サブコマンドは次のいずれかになります。

start	アプリケーションサーバインスタンスを起動します。このコマンドをローカルホストから実行したときに、 Administration Server (KAS) が起動されていないときは起動されます。
stop	アプリケーションサーバのエンジンを停止します。管理サーバは停止しません。
kill	ローカルホスト上で実行中のアプリケーションサーバおよび iPlanet Application Server Administration Server (KAS) のすべてのプロセスを、ただちに強制終了します。

次のパラメータが適用されます。

instance	停止または起動するサーバインスタンスの名前。サーバインスタンスは iASAT に登録されている必要があります。
user	サーバインスタンスを起動、停止、または強制終了する権限が与えられているユーザの名前。このユーザは iASAT に登録されている必要があります。
password	ユーザに関連付けられたパスワード
host	起動、停止、または強制終了するサーバインスタンスのホスト名または IP アドレス
port	アプリケーションサーバの Administration Server のポート番号。ポート 10817 はデフォルトです。
eng	起動または停止するエンジンの ID。eng オプションを使用すると、 iPlanet Application Server のインスタンスに関連付けられている特定のエンジンを停止できます。eng オプションを使用しない場合は、目的のサーバに関連付けられているすべてのエンジンが停止します。 このコマンドを使用すると、サーバエンジンは適切にシャットダウンされますが、 KAS は稼働し続けます。 制御するエンジン ID は、 iPlanet レジストリ内の次のキーから取得できます。 Software/iPlanet/Application Server/6.5/CCSO/Eng

iascontrol start の例

1. iPlanet Application Server エンジンの特定のインスタンスを起動する

```
iascontrol start -instance <instance> -user <user> -password
<password> -host <host> -port <port> [-eng <engineid>]
```

このコマンドは、指定されたインスタンスのエンジンを起動します。iPlanet Application Server のローカルインスタンスの場合、KAS がまだ起動していなければ、KAS が起動されます。

2. iPlanet Application Server の指定されたインスタンス内のエンジンを起動する

```
iascontrol start -eng 1
```

このコマンドは、指定されたエンジンと、ローカルホストの KAS (まだ起動されていない場合) を起動します。

3. 指定されたインスタンスに関連付けられているエンジンを起動する

```
iascontrol start -instance prodnode1 -eng 2
```

このコマンドは、指定されたインスタンスに関連付けられているエンジンを起動します。この例では、指定されたインスタンスは *prodnode1* です。

注 このコマンドを実行する前に、指定されたインスタンス上で KAS が稼働している必要があります。

4. ローカルホスト上の特定のインスタンスに関連付けられているエンジンを起動する

```
iascontrol start -user admin1 -password admin -host prodserver -port
10817 -eng 1
```

このコマンドは、ホスト *prodserver* 上のインスタンスに関連付けられているエンジンを起動します。KAS はすでに起動され、指定されたポートで待機している必要があります。

iascontrol stop の例

1. 指定されたインスタンスのエンジンを停止する

```
iascontrol stop [-instance <instance> [-user <user> -password
<password>] [-host <host> -port <port>] ] [-eng <engineid>]
```

このコマンドは、指定されたインスタンスのエンジンを停止します。エンジンは適切にシャットダウンされます。ただし、KAS は稼働し続けます。

2. ローカルホスト上の特定のエンジンを停止する

```
iascontrol stop -eng 1
```

このコマンドは、ローカルホスト上の指定されたエンジンを停止します。ただし、KAS は稼動し続けます。

3. iPlanet Application Server の特定のインスタンスを停止する

```
iascontrol stop -instance prodnodel -eng 2
```

このコマンドは、*prodnodel* というインスタンスに関連付けられている特定のエンジンを停止します。ただし、KAS は稼動し続けます。

4. ローカルホスト上のインスタンスに関連付けられている特定のエンジンを停止する

```
iascontrol stop -user admin1 -password admin -host prodserver -port  
10817 [-eng 2]
```

このコマンドは、ローカルホスト上のインスタンスに関連付けられているエンジンを停止します。ただし、KAS は稼動し続けます。

4. iPlanet Application Server インスタンスを停止するコマンドは次のようになります。

```
iascontrol stop -instance prodnodel -user admin1 -password admin  
-host prodserver -port 10817
```

これらのコマンドを実行すると、ターミナルウィンドウに次のコードのいずれかが表示されます。

0 - 正常に終了しました。

1 - 1 つまたは複数の指定したサーバと通信できませんでした。

2 - 1 つまたは複数の指定したサーバに対する認証 / 承認が失敗しました。

注 これらのコマンドを使用する前に、iASAT を使って iPlanet Application Server を登録する必要があります。

ユーザ ID (iPlanet Application Server のインストール中に入力した ID) が 8 文字より長いと、-stop および -kill コマンドは正しく動作しません。

Solaris での環境変数の設定

Solaris プラットフォームでは、共通環境変数の設定はスクリプトファイル `iasenv.ksh` に記録されます。`iasenv.ksh` スクリプトファイルには、`CLASSPATH`、`GX_ROOTDIR`、および `LD_LIBRARY_PATH` などの共通環境変数の設定が記録されます。

つまり、これらの設定は、`ias/bin` または `ias/jms/bin` ディレクトリにある個別のスクリプトファイルには設定されません。

`iasenv.ksh` スクリプトは `<iASInstallDir>/ias/env` ディレクトリにあります。

注

- 現在のアプリケーションが `kjs`、`kxs` などの環境情報を読み込んでいる場合は、`iasenv.ksh` からこれらを読み込むように変更する必要があります。
- 下位互換性の問題が発生した場合は、`kxs` や `kjs` などの個々のスクリプトを変更する代わりに `iasenv.ksh` 内に必要な変数を定義できます。

`iasenv.ksh` スクリプトと、`iasenv.ksh` を参照するほかのスクリプト (`kjs` など) の両方に同じ値が存在する場合は、これらのスクリプト内で変数が定義されている位置によって優先される設定が異なります。たとえば、`kjs` 内で `iasenv.ksh` スクリプトの呼び出しの前に変数が定義されている場合は、`iasenv.ksh` スクリプトの設定が考慮され、`kjs` 内で `iasenv.ksh` の呼び出しのあとに変数が定義されている場合は、別のスクリプト内に定義されている設定が考慮されます。

警告

`iasenv.ksh` スクリプトの内の値は特に必要な場合を除き変更しないことをお勧めします。

iPlanet レジストリエディタについて

iPlanet レジストリエディタ (kregedit) は、iPlanet Application Server のレジストリ情報を表示および編集できるスタンドアロン GUI ツールです。このエディタは、iPlanet Application Server の各インスタンスとともにインストールされ、Windows のレジストリエディタに似た外観と機能を持ちます。iPlanet レジストリエディタには、ローカルマシンのレジストリと Directory Server の両方に保存されている値が表示されます。

iPlanet レジストリエディタを使うと、iPlanet Application Server のさまざまな機能および処理のバックエンドリソースを形成するエントリを効果的に管理できます。キーの追加、キー値の変更、および不要なキーの削除ができます。kregedit には包括的な Find/Replace オプションがあり、これを使ってアプリケーションサーバのレジストリ内に出てくる同じ値を検出して置換できます。

kregedit を起動するには

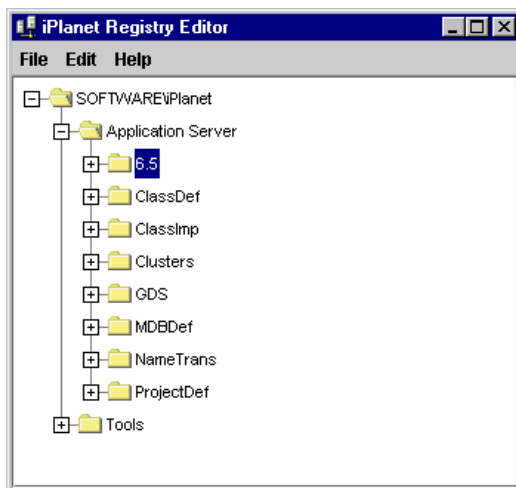
kregedit を起動するには、次の手順を実行します。

1. Solaris で iPlanet レジストリエディタを起動するには、*iASInstallDir/ias/bin* に移動してコマンドラインで *kregedit* と入力します。

注 Solaris では iPlanet Application Server の複数のインスタンスがサポートされるので、各インスタンスは固有のレジストリを持ちます。目的のインスタンスがインストールされている場所から *kregedit* を起動してください。

1. Windows では、「スタート」>「ファイル名を指定して実行」をクリックします。「kregedit」と入力して「OK」をクリックします。

次のウィンドウが表示されます。



iPlanet レジストリエディタによる管理タスクの実行

iPlanet Application Server Administration Tool および Deployment Tool によって、必要に応じて iPlanet レジストリにバックエンドエントリが作成されます。ただし、アプリケーションサーバのレジストリで特定の設定を手動で作成または編集しなければならない場合もあります。

さまざまなキーとその値については、『iPlanet Application Server レジストリガイド』を参照してください。

これらの設定タスクの多くは、このマニュアルのいくつかの章で説明されています。この章では、kregedit を使って実行できる次の基本的な設定作業について説明します。

次の節では、kregedit を使って基本的な管理作業を実行する方法について説明します。

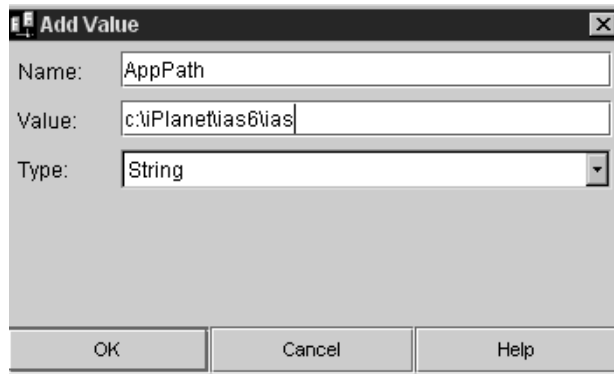
- iPlanet レジストリに値を追加するには
- iPlanet レジストリの値を変更するには
- iPlanet レジストリ内の値の検索と置換を行うには
- インストールキーを更新するには
- IP Address を変更するには

iPlanet レジストリに値を追加するには

iPlanet レジストリには各フォルダを示す複数のキーが含まれています。これらのキーに含まれている値は Administration Tool、Deployment Tool、Web コネクタプラグイン、Directory Server などのさまざまなツールやサーバによってアクセスされます。

iPlanet レジストリにキーを追加するには、次のタスクを実行します。

1. 新しいキーを作成するキーを選択します。
2. 「編集」 > 「キーを追加」をクリックします。キーの名前を入力し、「OK」をクリックします。
3. このキーに値を追加するには、キーを選択して「編集」メニューから「値を追加」を選択します。次の図のように「値を追加」ダイアログボックスが開きます。



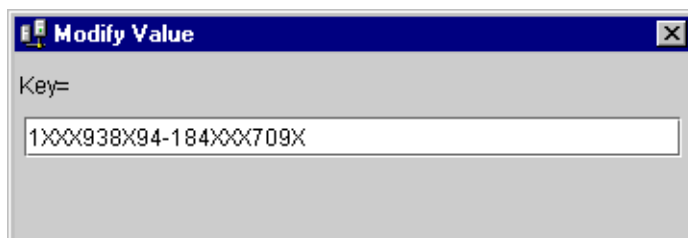
4. 「名前」フィールドに、値の名前を指定します。
例 1: アプリケーションパスを示す値を作成する場合は、「AppPath」と入力します。
例 2: サーバの自動起動回数を指定する場合は、「AutoStart」と入力します。
5. 必要に応じて「値」フィールドにキー値を入力します。
前述の例 1 で示したようにクラスパスを指定する場合は、次の例のようにクラスパスを入力します。
`c:\iPlanet\ias6\ias`
前述の例 2 のように、サーバの自動起動回数を指定する場合は、「1」のように数字を入力します。

- 指定した値のタイプに基づいて、「入力」フィールドで「String」か「Integer」を選択します。
たとえば、値がクラスパスの場合は、「入力」ドロップダウンリストから「String」を選択します。
たとえば、この値がサーバの自動再起動回数を示す数値である場合は、「入力」ドロップダウンリストから「Integer」を選択します。
- iPlanet レジストリにその値を登録する場合は「OK」をクリックします。

iPlanet レジストリの値を変更するには

iPlanet レジストリの値を変更するには、次のタスクを実行します。

- 変更する値を選択します。
- そのエントリをダブルクリックするか、「編集」>「値を変更」をクリックします。次のダイアログボックスが表示されます。



- 値を変更して「OK」をクリックします。

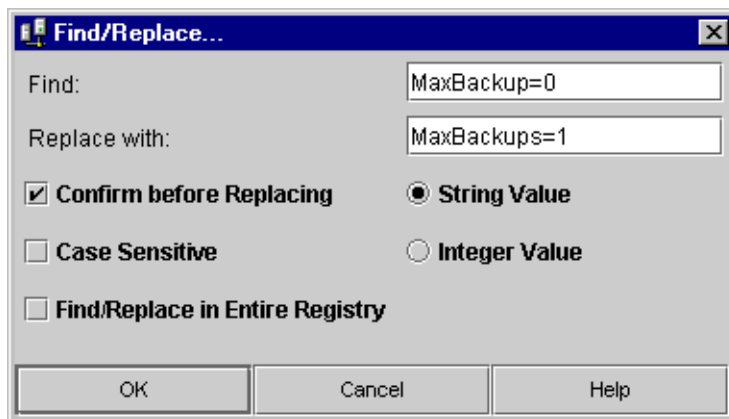
注 「値を変更」ダイアログボックスでは、キーに指定された値だけを変更できます。値のタイプ（「String」または「Integer」）の変更はできません。値のタイプの指定を変更するには、値を削除してからもう一度追加する必要があります。

レジストリから値を削除するには、削除する値を選択して **Delete** キーを押すか、「編集」>「削除」をクリックします。削除を確認するダイアログボックスが表示されます。削除する場合は「はい」をクリックします。

警告 一度削除すると元に戻すことはできません。

iPlanet レジストリ内の値の検索と置換を行うには

「検索 / 置換」メニューオプションによって、iPlanet レジストリ内の値を検索して置換できます。特定のサブツリー内に存在する値を検索して置換したり、レジストリ全体でその値をすべて置換したりできます。



iPlanet レジストリ内の値を検索して置換するには、次のタスクを実行します。

1. 「編集」メニューから、「検索 / 置換」を選択します。
2. 「検索」テキストフィールドで、iPlanet レジストリで検索する値を指定します。値の一部だけを変更するには、その部分だけを指定します。たとえば、iPlanet.com を Sun.com に置換する場合は、「iPlanet」だけを入力します。
3. 「置換」フィールドに、既存の値と置換する値を指定します。たとえば、iPlanet.com を Sun.com に置換するには、「Sun」だけを入力します。残りの値は自動的に保持されます。
4. 検索する値がクラスパスなどの文字列の場合は、「文字列値」をクリックします。
5. 検索する値が 1 などの整数の場合は、「整数値」をクリックします。
6. 入力した検索値を大文字と小文字を区別して検索するには、「大文字と小文字を区別」テキストボックスをオンにします。
7. iPlanet レジストリ内で検索値を置換する前に確認メッセージを表示するには、「置換前に確認」チェックボックスをオンにします。このオプションをオンにすると、iPlanet レジストリで値を置換する前に必ず置換するかどうか尋ねられます。

注 iPlanet レジストリには、さまざまなエンジンやプロセスが依存している重要な多くの情報が格納されています。検索 / 置換操作を確認する前に、このオプションをオンにしておくことをお勧めします。

8. レジストリ全体で検索した値すべてを検索して置換するには、「Find/Replace in Entire Registry」チェックボックスをオンにします。このチェックボックスをオンにすることによって、1 つのツリーやサブツリーに限定されない検索 / 置換機能を使用できます。

注 このチェックボックスをオフにすると、選択したサブツリー内でのみ、値が検索されます。

9. 検索 / 置換操作を開始するには、「OK」をクリックします。

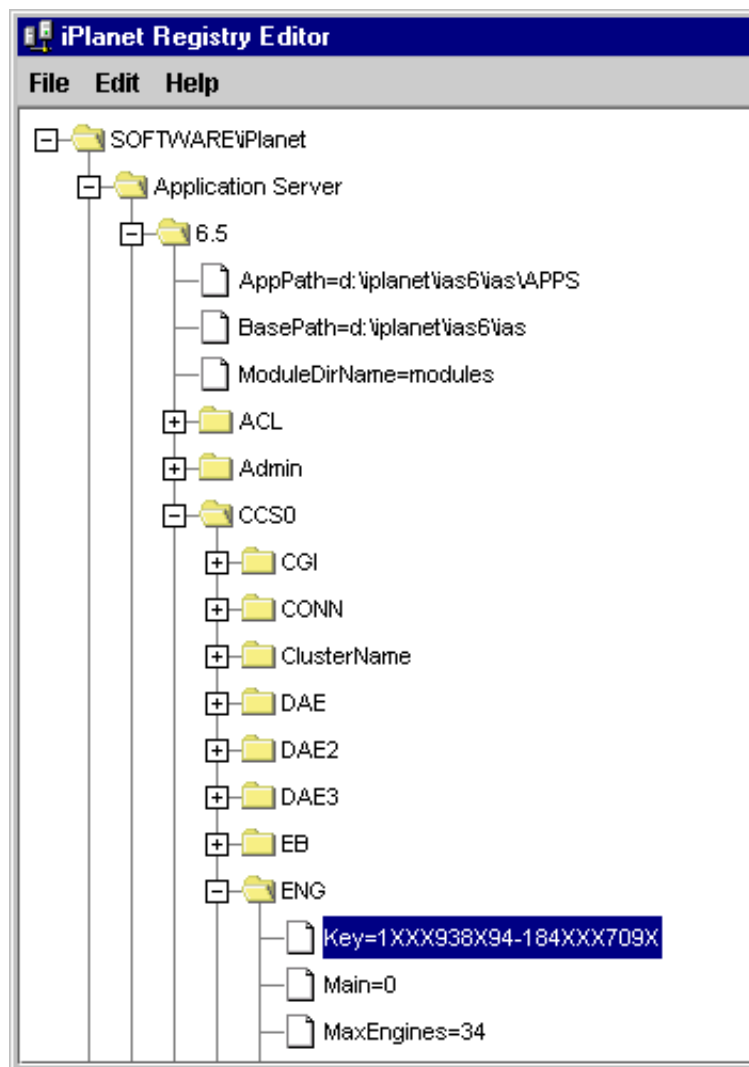
検索 / 置換操作が終了すると、検索結果と実行された置換を示す確認ダイアログが表示されます。このダイアログには、値が置換されたキーの完全なツリー構造とともに、実行されたすべての置換の完全なリストが表示されます。

インストールキーを更新するには

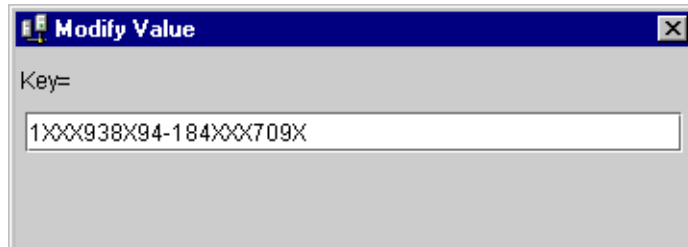
iPlanet Application Server を評価用ライセンスでインストールした場合、評価期間が終了するとサーバは停止します。評価期間を延長したときやサーバを購入したときは、インストールキーを更新する必要があります。インストールキーを更新すると、サーバソフトウェアをインストールし直さなくてもよくなります。

インストールキーを再設定するには、次のタスクを実行します。

1. iPlanet Application Server をシャットダウンします。
2. iPlanet レジストリエディタを開きます。
(「iPlanet レジストリエディタについて」を参照してください。)
次のウィンドウが表示されます。



3. 次のキーを開きます。
SOFTWARE\iPlanet\Application Server\6.5\CCS0\ENG
4. 「key」文字列値をダブルクリックして、新しいインストールキーの値を入力します。



5. 「OK」をクリックします。
6. レジストリエディタを閉じます。
7. iPlanet Application Server を再起動します。

IP Address を変更するには

マシンの移動や異なるネットワークへの割り当てなどで、iPlanet Application Server がインストールされているマシンの IP アドレスが変わった場合は、以前のアドレスを新しい IP アドレスに更新する必要があります。

iPlanet Application Server の各インスタンスのローカル iPlanet レジストリにある IP アドレスを置き換える必要があります。これは、ネットワークにマシンの現在の IP アドレスを通知するために行います。

マシンに iPlanet Console 5.0 がインストールされている場合は、ネットワーク上の別のマシンに、iPlanet Console Administration Server にどこからアクセスできるかを知らせる必要があります。

IP アドレスを変更する手順については、次の節で説明します。

- ローカル iPlanet レジストリで IP アドレスを変更するには
- iPlanet Console Administration Server の IP アドレスを更新するには

注 マシンに iPlanet Console Administration Server をインストールしていない場合は、2 番目の節を読む必要はありません。

ローカル iPlanet レジストリで IP アドレスを変更するには

iPlanet レジストリエディタである kregedit を使って、iPlanet レジストリの新しい IP アドレスを更新できます。これを実行すると、ローカルレジストリとマシンの設定を保存している Directory Server の両方で IP アドレスが更新されます。

注 Solaris マシンでは、複数の iPlanet Application Server のインスタンスがサポートされています。複数の iPlanet Application Server のインスタンスをインストールしている場合、各インスタンスの iPlanet レジストリ内の古い IP アドレスを新しい IP アドレスに置き換える必要があります。

iPlanet レジストリの IP アドレスを変更するには、次のタスクを実行します。

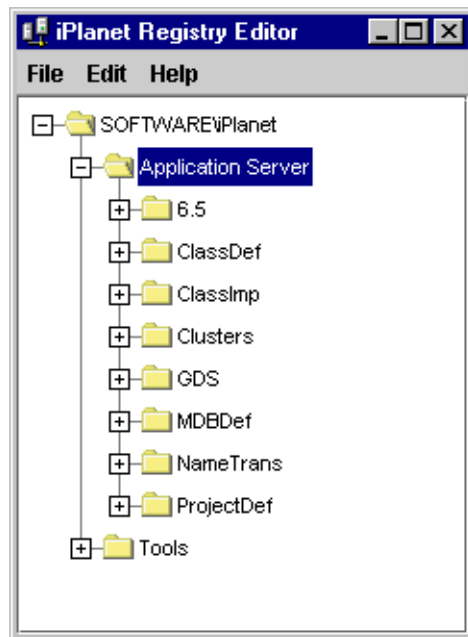
1. iPlanet レジストリエディタを起動します

「iPlanet レジストリエディタについて」を参照してください。

2. 次のキーを選択します。

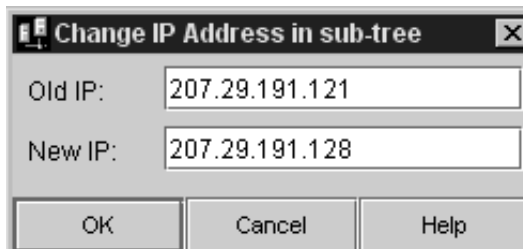
SOFTWARE\iPlanet

これはルートキーです。レジストリ内の古い IP アドレスをすべて更新するには、このキーを選択する必要があります。



3. 「編集」メニューから、サブツリーの「サブツリーの IP アドレスの変更」を選択します。

「サブツリーの IP アドレスの変更」ダイアログボックスが表示されます。



4. 「古い IP」テキストフィールドに古い IP アドレスを入力します。
5. 「新規 IP」テキストフィールドに新しい IP アドレスを入力します。

注 IP アドレスは 129.29.191.128 の形式で入力する必要があります。
数字の各シーケンスをドットで区切ります。

6. 「OK」をクリックして変更を保存します。

注

- 「サブツリーの IP アドレスの変更」ダイアログボックスで IP アドレスとポート番号を入力した場合 (たとえば、129.29.191.128:10818)、iPlanet レジストリ内の IP アドレスだけが抽出、認証、および置換されます。
- レジストリ内のすべての古い IP アドレスが置き換えられたことを確認するには、「編集」メニューの「検索 / 置換」オプションを利用します。このオプションによって IP アドレスの形式が正しいことを確認できますが、iPlanet レジストリ内で置き換える前に IP アドレスの検証は行われないので注意してください。

iPlanet Console Administration Server の IP アドレスを更新するには

iPlanet Console 6.5 は、一般的なサーバ管理機能を持っています。サーバーの停止および起動、新しいサーバインスタンスのインストール、またディレクトリサーバの LDAP サービスを介してユーザおよびグループ情報の管理を行うことができます。iPlanet Console を使用すると、ネットワーク上にあるディレクトリサーバのリモートインスタンスを制御することができます。

iPlanet Console とともに固有の Administration Server も提供されますが、これは iPlanet Console によって内部的に使われます。マシンに iPlanet Console がインストールされている場合は、ネットワーク上のすべてのマシンに正しい nsserveraddress (iPlanet Console Administration Server の IP アドレス) と nsadminaccessaddress (iPlanet Console Administration Server にアクセスできるマシンの IP アドレス) が通知されていることを確認する必要があります。

iPlanet Console Administration サーバの nsserveraddress は Directory Server および iPlanet Console のローカルコンフィグレーションファイルの 2 箇所に保存されています。

nsadminaccessaddress は iPlanet Console のローカルコンフィグレーションファイルだけに保存されています。

次の節では、nsserveraddress および nsadminaccessaddress を変更する方法について説明します。

- Directory Server 内の IP アドレスを変更するには
- ローカルコンフィグレーションファイルで IP アドレスを変更するには

Directory Server 内の IP アドレスを変更するには

Directory Server の nsserveraddress を変更するには、次のタスクを実行します。

1. <iASInstallDir>/slapd-<hostname> に移動して、次のスクリプトを実行します。

```
db2ldif
```

このスクリプトによって、LDIF (LDAP Data Interchangeable Format) ファイルが作成されます。このファイル名の先頭に、YYYY-MM-DD 形式で作成された日付と時刻が記録されます。たとえば、2001 年 6 月 3 日 14 時 2 分 38 秒に作成された LDIF ファイルは次のようになります。

```
2001_06_03_140238.ldif
```

2. <iASInstallDir>/slapd-<hostname>/ldif に移動し、最新の日付および時刻スタンプを持つ LDIF ファイルを探します。
3. LDIF ファイルを開き (テキストエディタを使用)、nsserveraddress エントリを検索します。古い IP アドレスを新しい IP アドレスに置き換えます。変更を完全に行うには、すべての古い IP アドレスを検索して新しいアドレスに置き換える必要があることに注意してください。
4. 次のタスクを実行して、Directory Server を停止します。

Windows の場合

- a. 「スタート」メニューから、「設定」>「コントロールパネル」を選択します。

- b. 「サービス」アイコンをダブルクリックします。
- c. 「サービス」ウィンドウで、iPlanet Directory Server を選択します。
- d. 「停止」をクリックしてサーバを停止します。

Solaris の場合

- a. パス <iASInstallDir>/slapd-<hostname> に移動します。
- b. 次のスクリプトを実行します。

```
stop-slapd
```

Directory Server が停止します。

5. 次に Directory Server に LDIF ファイルを書き込みます。これを実行するには、<iASInstallDir>/slapd-<hostname> に移動し、次のスクリプトを実行します。

```
ldif2db -i ldif/<path>/<filename>.ldif
```

このスクリプトを実行すると、ファイルに行った変更が Directory Server に記録されます。

注 nserveraddress 値を変更した LDIF ファイルの絶対パスを指定する必要があります。

6. Directory Server を起動する

Directory Server を起動するには、上記の手順 4 に説明されている手順に従ってください。Windows では、「サービス」ウィンドウで「開始」をクリックし、サーバを起動します。Solaris では、スクリプト start-slapd を実行してサーバを起動します。

注 Directory Server 内のすべての古い IP アドレスが置き換えられたことを確認するため、次のパスに保存されているログファイルを確認します。

```
<iASInstallDir>/slapd-<hostname>/logs
```

ローカルコンフィグレーションファイルで IP アドレスを変更するには
iPlanet Console Administration Server のローカルコンフィグレーションファイルで IP アドレスを変更するには、次のタスクを実行します。

1. <iASInstallDir>/admin-serv/config に移動し、テキストエディタを使って local.conf ファイルを開きます。

2. `nsserveraddress` エントリを探し、それらの IP アドレスの値を変更します。エントリは次のようになります。
`configuration.nsserveraddress.129.158.228.63.`
3. 存在する場合は `nsadminaccessaddress` エントリを探し、IP アドレス値を変更します。エントリは次のようになります。
`configuration.nsadminaccessaddress.129.158.228.63.`
4. 次のタスクを実行し、**iPlanet Console Administration Server** を停止して起動します。

Windows の場合

- a. `<iASInstallDir>/` に移動して `stop-admin` 実行可能ファイルをダブルクリックしてサーバを停止します。
- b. サーバを起動するには、`start-admin` 実行可能ファイルをダブルクリックします。

Solaris の場合

- a. `<iASInstallDir>/` に移動し、`stop-admin` スクリプトを実行してサーバを起動します。
- b. サーバを起動するには、同じパスから `start-admin` スクリプトを実行します。

-
- 注**
- 1つのホストに複数の **iPlanet Application Server** のインスタンスがインストールされている場合 (Solaris マシンの場合のみ可能) は、各インスタンスに1つのローカルコンフィグレーションファイルが存在します。IP アドレスが変わった場合は、**iPlanet Application Server** の各インスタンスについてローカルコンフィグレーションファイル内の必要なエントリを更新する必要があります。
 - マシンに複数の **iPlanet Application Server** のインスタンスがインストールされている場合は、変更を有効にするために、各インスタンスごとに **iPlanet Application Server** を停止して起動する必要があります。
-

複数の LDAP ドメインを構成するには

iPlanet Application Server では、Nortel LDAP スキーマに基づいて複数の LDAP ドメイン名がサポートされています。

アプリケーションにログインする際に、john.doe@acme.com のような別のドメイン名を使うこともできます。ユーザに関連づけられた各々のドメイン名は、ディレクトリサーバに入力されます。

複数のドメイン名を使用するには、iPlanet レジストリに新しいドメイン名を追加し、ディレクトリサーバ内に、対応するバックエンドエントリを作成する必要があります。これは、Solaris および Windows の両方で実行できます。

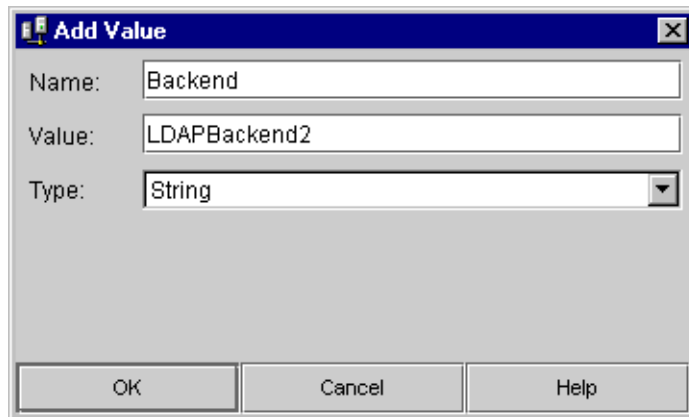
この節には次のトピックがあります。

- iPlanet レジストリにドメイン名を追加するには
- ディレクトリサーバに新しいドメイン名を設定するには

iPlanet レジストリにドメイン名を追加するには

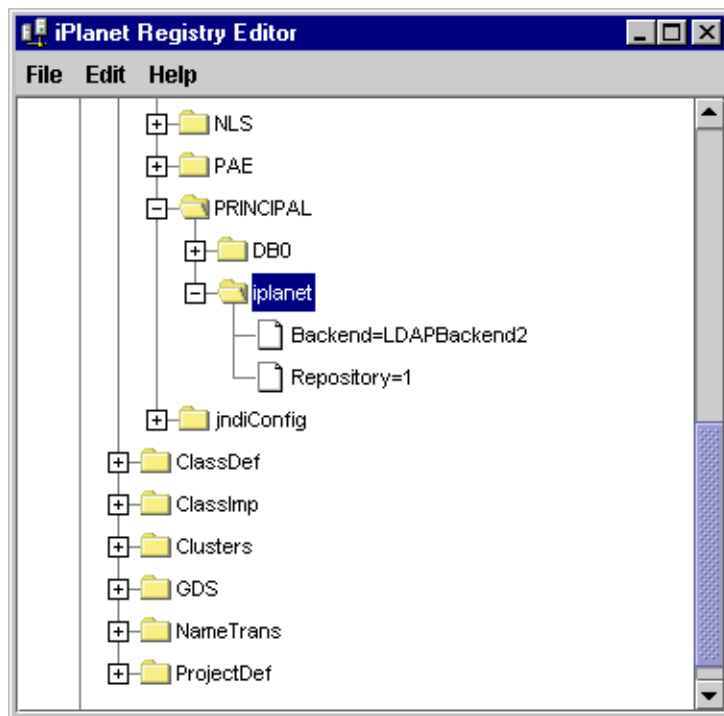
iPlanet レジストリにドメイン名を追加するには、次のタスクを実行します。

1. iPlanet レジストリエディタを開きます。
「iPlanet レジストリエディタについて」を参照してください。
2. 次のキーを開きます。
`Software/iPlanet/Application Server/6.5/Principal`
3. たとえば「iPlanet」のような新しいドメイン名を使用して、サブキーを作成します。
4. 新しいドメイン名を選択し、「編集」メニューから「編集」>「値を追加」を選択します。
次に示す「値を追加」ダイアログボックスが表示されます。



5. 「名前」テキストフィールドに値の名前を入力します。たとえば、「Backend」と入力します。
6. 「値」フィールドに値を指定します。たとえば、「LDAPBackend2」と指定します。
ここで指定した値は、ディレクトリサーバ内に、同じ名前の対応する値がなければなりません。
詳細は、「ディレクトリサーバに新しいドメイン名を設定するには」を参照してください。
7. iPlanet レジストリにその値を登録する場合は「OK」をクリックします。
次に、次の手順に従って、このドメインサブキーの 2 番目の値を作成する必要があります。
8. 新しいドメインキーを選択し、「編集」メニューから「値を追加」を選択します。
9. 「値を追加」ダイアログボックスで、たとえば、「Repository」と名前を指定します。
10. 「値」には、たとえば、「1」と指定します。「入力」ドロップダウンリストから「Integer」を選択します。

これで、iPlanet レジストリに新しいドメイン名が定義されました。iPlanet レジストリへの新しいドメイン名の定義が終了すると、エントリが次の例のようになります。



ディレクトリサーバに新しいドメイン名を設定するには

iPlanet レジストリで新しいドメイン名を定義したら、Directory Server のドメイン名も設定する必要があります。iPlanet レジストリのドメイン名エントリは、Directory Server で作成したエントリを指します。

Directory Server で新しいドメイン名を設定するには、次のタスクを実行します。

1. iPlanet レジストリで、キー Software/iPlanet/Application Server/GDS/Backends/LDAP/ を開きます。
2. Software/iPlanet/Application Server/6.5/Principal の下位に指定した値と同じ値 (つまり、LDAPBackend1) を持つサブキーを作成します。
3. このサブキーの値を作成します。この値によって、Directory Server で定義した共通名 (cn) の属性を定義します。次のようにします。
root=cn=global,cn=iasconfig,cn=iAScluster,O=NetscapeRoot.
4. キー LDAPBackend1 の下に新しいサブキーを作成します。たとえば、0 を作成します。
5. このサブキー 0 に次の値を作成します。

複数の LDAP ドメインを構成するには

- GroupPath = ou=Groups, o=<new domain>
- Host = <hostname>
- Password = <encrypted password>。
- Port = <Directory Server Port>。デフォルトのポートは 389 です。
- User = cn=Directory Manager
- UserPath = ou=People, O=<new domain>

この手順によって、Directory Server 内の新しいドメイン名のバックエンドエントリが設定されます。

注 新しいドメイン名を使ってログインする場合は、Directory Server による認証を受けるための暗号化されたパスワードが必要です。暗号化されたパスワードを作成するには、次のスクリプトを実行します。

```
<iASInstallDir>/ias/bin/kencrypt2 <text to be encrypted>
```

このスクリプトによって、入力した英数文字列から暗号化されたパスワードが作成されます。パスワードには、文字数の制限はありません。

1 台の iPlanet Application Server の管理

第 2 章 「サーバの稼動状況の監視」

第 3 章 「iPlanet Application Server をサードパーティツールを使って監視するよう SNMP を設定する」

第 4 章 「サーバメッセージの記録」

第 5 章 「iPlanet Application Server の保護」

第 6 章 「サーバリソースの可用性の向上」

第 7 章 「Web コネクタプラグインの設定」

第 8 章 「データベース接続の管理」

第 9 章 「トランザクションの管理」

サーバの稼動状況の監視

この章では、iASAT によって提供される監視サービスについて説明します。このサービスを使うと、Executive Server、Java サーバ、C++ サーバ、およびブリッジサーバプロセスのさまざまな属性をチャートにできます。

この章には次のトピックがあります。

- iPlanet Application Server の監視
- イベント通知の受信

iPlanet Application Server の監視

iPlanet Application Server Administration Tool (iASAT) は、iPlanet Application Server を構成する Executive Server、Java サーバ、C++ サーバ、およびブリッジサーバの稼動状況をグラフにする監視サービスを提供します。その情報をファイルに記録することもできます。このようにサーバの稼動状況をグラフィカルに表示したり、ファイルにデータとして記録したりすることによって、1つのアプリケーションサーバまたはサーバグループのパフォーマンスを追跡して確認し、向上させるように調整することができます。たとえば、アプリケーションサーバにメモリを追加する場合や、新しいアプリケーションを配置する場合は、アプリケーションサーバのパフォーマンスを監視して、これらの変更の影響について調査できます。

iPlanet Application Server の監視サービスでは、指定された間隔でアプリケーションサーバをポーリングします。サーバは監視している情報を常に更新するのではなく指定された間隔で更新するので、サーバリソースを節約できます。この間隔は「モニタ」ウィンドウで指定できます。間隔の設定については、57 ページの「プロセスデータプロットを変更するには」を参照してください。

監視するプロセスをクリックすると、Administration Tool から「モニタ」ウィンドウが「ポップアップ」します。これによって、Administration Tool を使って他の管理タスクの実行を継続しながら、独立したウィンドウでサーバの稼動状況を監視できます。

この節では、次のトピックについて説明します。

- プロセス属性の監視
- プロセスデータをファイルに記録するには
- プロセスデータプロットを変更するには
- プロセスデータプロットを削除するには

プロセス属性の監視

チャートにできるサーバの状態つまり属性は、どのサーバ、つまりどのプロセスをモニタしているかによって異なります。

Executive Server (KXS) プロセスには、ロードバランスサービスなどのシステムレベルのサービスを管理しホスティングする役割やアプリケーションコンポーネントの作成言語に応じて Java サーバまたは C++ サーバのどちらかのアプリケーションプロセスの 1 つにリクエストを転送する役割があります。

次の表は、チャートにできる Executive Server プロセスの属性を示します。

表 2-1 Executive Server の監視属性

Executive Server プロセス属性 (KXS)	説明
CPU 負荷	ロードバランスサービスによって計算された、この Executive Server プロセスが実行している CPU 負荷の量
ディスク入出力	ロードバランスサービスによって計算された、この Executive Server が稼動しているシステムで発生した読み書き動作のレート
メモリスラッシュ	参照時にメモリになかったページへのメモリ参照を解決するために、ハードディスクドライブから読み込まれるかまたはそのドライブに書き込まれるページの数
キューに入っているリクエスト	処理のキューで待っている現在のリクエストの数
キャッシュ結果	結果キャッシュに保存されているエントリの数
平均実行時間	リクエストを実行する Executive Server プロセスの平均時間
リクエスト / 間隔	最後のポーリングのあとに受信した新しいリクエストの数
リクエスト数の合計	起動時にプロセスが受信したリクエストの数の合計 (サーバまたはプロセスを起動するときはこの値を 0 にリセットする)。実行プロセスの場合、この値は、サーバがすべてのサーバプロセスで実行したリクエストの数の合計になる

表 2-1 Executive Server の監視属性 (続き)

Executive Server プロセス属性 (KXS)	説明
現在のリクエスト	現在のサーバによって処理されているリクエストの数 (KJS/KCS エンジン内で分散され、処理されているすべてのリクエストを含む)
リクエスト待機	サービスを待っているキュー内のリクエストの数
リクエスト準備完了	サービスの準備ができたキュー内のリクエストの数
現在のリクエストスレッド	プロセスによって割り当てられるリクエストスレッド数 (アイドルスレッドとリクエストを処理中のスレッドの両方を含む)。この数は、このプロセスに設定されている最大数スレッドを超えたり、最小数スレッドより少なくなったりすることはない。これらの値は「一般」ウィンドウで設定する
リクエストスレッド待機	新しい受信リクエストを実行するために使用可能なリクエストスレッド数。この数は、現在のリクエストスレッドの監視属性のサブセットになる
合計スレッド	プロセスによって使われるスレッド数
送信バイト / 間隔	最後にポーリングしてから送信した新しいバイト数
受信したバイト / 間隔	最後にポーリングしてから受信した新しいバイト数
現在のセッション	現在処理されているセッションの数

注 CPU 負荷、ディスク I/O、またはメモリスラッシュを監視する場合は、これらのプロセス属性の統計を更新する間隔を指定する必要があります。間隔を設定するには、「ロードバランス」ボタンを選択します。「ユーザ定義クライテリア (iAS 駆動)」を選択して、「変更の適用」タブをクリックします。

Java サーバ (KJS) および C++ サーバ (KCS) プロセスには、記述言語に応じてアプリケーション要素をホスティングする役割があります。Java サーバは Java で書かれたアプリケーションコンポーネントを管理し、C++ サーバは C++ で書かれたコンポーネントを管理します。さらに、CORBA Executive Server (CXS) またはブリッジプロセスを使うと、独立した Java クライアント (リッチクライアント) は、Java サーバ上でホストされる Enterprise JavaBeans と直接通信できます。リッチクライアントについては、『開発者ガイド (Java)』を参照してください。

次の表は、チャートにできる Java サーバ、C++ サーバ、およびブリッジサーバのプロセスの属性を示します。

表 2-2 Java、C++、および IIOP ブリッジサーバの監視属性

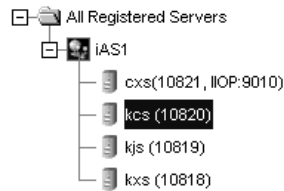
Java/C++、およびブリッジサーバのプロセス (KJS、KCS、および CXS) の監視属性	説明
平均実行時間	リクエストを実行するプロセスの平均時間
リクエスト / 間隔	時間間隔内に受信した新しいリクエストの数
リクエスト数の合計	最後の起動後にプロセスが受信したリクエストの数の合計。サーバまたはプロセスを起動するときはこの値を 0 にリセットする
現在のリクエスト	現在このプロセスによって処理されているリクエスト数
リクエスト待機	サービスを待っているキュー内のリクエストの数
準備完了リクエスト	サービスの準備ができたキュー内のリクエストの数
現在のリクエストスレッド	プロセスによって割り当てられるリクエストスレッド数 (アイドルスレッドとリクエストを処理中のスレッドの両方を含む)。この数は、このプロセスに設定されている最大数スレッドを超えたり、最小数スレッドより少なくなったりすることはない。これらの値は「一般」ウィンドウで設定する
リクエストスレッド待機	新しい受信リクエストを実行するために使用可能なリクエストスレッド数。この数は、現在のリクエストスレッド監視属性のサブセットになる
アクティブデータコネクション	現在アクティブなデータコネクションの数
キャッシュされたデータコネクション	現在キャッシュされているデータコネクションの数
クエリ / 間隔	時間間隔内に実行されたクエリの数
コミットされたトランザクション / 間隔	時間間隔内にコミットされたトランザクションの数
ロールバックされたトランザクション / 間隔	時間間隔内にロールバックされたトランザクションの数
合計スレッド	プロセスによって使われるスレッド数
送信バイト / 間隔	最後にポーリングしてから送信した新しいバイト数
受信したバイト / 間隔	最後にポーリングしてから受信した新しいバイト数

各プロセスに対して、1つまたは複数の属性をチャートにできます。複数のサーバを備えた企業の場合は、いくつかのアプリケーションサーバの属性を同時にチャートにすることもできます。

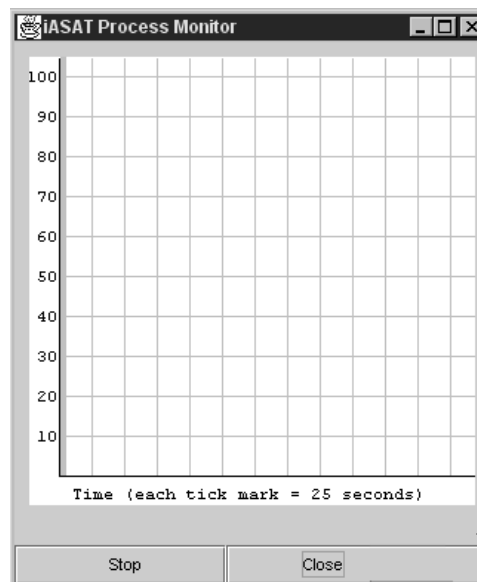
プロセス属性を監視するには

プロセス属性を監視するには、次の手順を実行します。

1. iASAT ツールバーの「モニタ」をクリックして「モニタ」ウィンドウを開きます。
2. 「モニタ」ウィンドウの左側のペインで、次の図のように属性をチャートにするプロセスをクリックします。

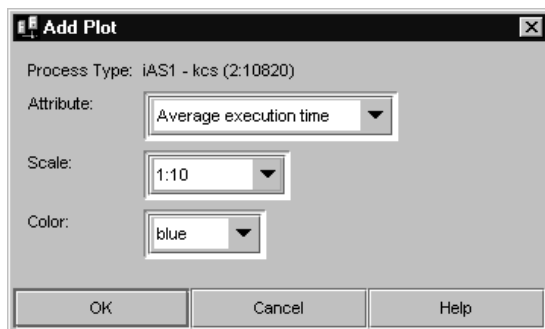


次の図のように、iASAT から別の監視パネルがポップアップします。



3. iASAT の「モニタ」ウィンドウの右側のペインで、ウィンドウのいちばん下にある「プロットの追加」ボタンをクリックします。

「プロットの追加」ダイアログが開き、強調表示されたプロセスについてどの属性を監視するかを指定します。

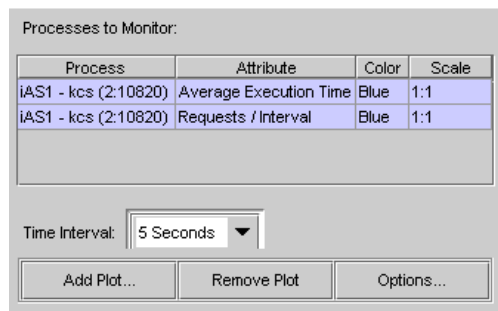


4. 「属性」ドロップダウンリストからチャートにする属性を選択します。
5. 「スケール」ドロップダウンリストから属性をプロットする割合（スケール）を選択します。

値の範囲は 10:1 ~ 1:1,000,000 です。10:1 のスケールは、「プロセスモニタ」ウィンドウで 10 ユニットが各属性カウントに対してプロットされることを示します。

6. 「色」ドロップダウンリストからチャートでプロセス属性を表す色を選択します。
7. チャートにする各プロセスまたは属性について、手順 2 ~ 6 を繰り返します。

チャートにするように選択した各プロセス属性が「モニタ」ウィンドウに表示されます。



- 「時間間隔」ドロップダウンリストから「iASAT プロセス モニタ」ウィンドウの更新間隔を選択します。

この設定は「モニタ」ウィンドウに表示されるすべてのプロセス属性に適用されます。

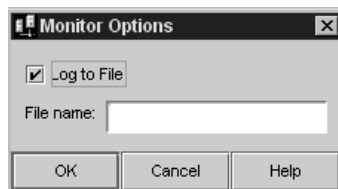
プロセスデータをファイルに記録するには

プロセス属性の監視を開始すると、監視サービスによって収集されたデータをファイルに送信できます。

プロセスデータをファイルに記録するには、次の手順を実行します。

- iASAT ツールバーの「モニタ」をクリックして「モニタ」ウィンドウを開きます。
- 「モニタ」ウィンドウの左側のペインで、データをチャートにするプロセスをクリックします。
- 「モニタ」ウィンドウの右側のペインで「オプション」をクリックします。

次のダイアログボックスが表示されます。



- 「ファイルにログ」チェックボックスをクリックしてロギングサービスを有効にします。
- 「ファイル名」テキストフィールドにデータの書き込み先のファイル名を入力します。
- 「OK」をクリックして変更を保存します。

プロセスデータプロットを変更するには

属性データプロットがプロセス (KCS、KJS、および KXS) に指定されると、「属性」、「色」、および「スケール」ドロップダウンボックスを使ってプロットを調整することができます。

プロセス属性のプロット方法を変更するには、次の手順を実行します。

1. iASAT ツールバーの「モニタ」をクリックして「モニタ」ウィンドウを開きます。
2. 「モニタ」ウィンドウの右側のペインで、属性を変更するプロセス行をクリックします。
3. 属性を変更するには、「属性」カラムをクリックして、ドロップダウンリストからプロットする新しい属性を選択します。
4. 同様に、「色」および「スケール」カラムをクリックして属性のプロット方法を変更します。

プロセスデータプロットを削除するには

プロセス属性のプロットが不要になった場合は、プロットからプロセスプロットを削除します。

プロセスプロットを削除するには、次の手順を実行します。

1. iASAT ツールバーの「モニタ」をクリックして「モニタ」ウィンドウを開きます。
2. 「モニタ」ウィンドウの左側のペインで、右側のペインにプロセス行を表示するプロセスを選択します。
3. 「モニタ」ウィンドウの右側のペインで、属性を削除するプロセスをクリックします。
4. 「プロットの削除」をクリックします。
「モニタ」ウィンドウから属性が削除されます。

イベント通知の受信

iPlanet Application Server をアクティブに監視できないときにはイベント通知が役に立ちます。この受動モニタシステムは、プロセスが失敗したときなど、重大な状況のときだけ実行されます。

関係者の電子メールアドレスを入力することによって、重大な状況が発生したときに、1人または複数の関係者に電子メールで警告するようにシステムを設定できます。さらに、特定のイベントが発生したときに、自動的に実行するスクリプトを指定できます。

この節には次のトピックがあります。

- イベントについて
- イベントに関する電子メール通知を設定するには
- イベント起動スクリプトを指定するには

イベントについて

次の重大なイベントに関して、通知する関係者や実行するスクリプトを指定できます。

- Executive Server (KXS) のダウン
- Java サーバ (KJS) のダウン
- C++ サーバ (KCS) のダウン
- プロセス自動再起動数の超過
- 異常なクラスタの検出

この節では次のトピックについて詳しく説明します。

- サーバがダウンした場合は
- 再起動数を越えた場合は
- 異常なクラスタが検出された場合は

サーバがダウンした場合は

1つまたは複数の Executive Server、Java サーバ、または C++ サーバプロセスがダウンすると、Administrative Server は各プロセスの再起動を試みます。Administration Server によってプロセスを再起動できないと、アプリケーションが実行を停止してトランザクションを失う可能性があります。

失敗を繰り返すのは通常、アプリケーションコード内に問題があるためですが、他の失敗も発生する可能性があります。プロセスの失敗原因に関係なくすぐに通知することをお勧めします。

プロセスが再起動したときは、失敗の原因を調べて今後の失敗を避けるための調整が可能かどうかを判断します。プロセスが再起動しないときは、ログを調べて失敗の原因を見つけてください。

再起動数を超えた場合は

Administration Server がプロセスを再起動する回数を超えた場合も通知できます。エンジンの最大再起動回数値は、「一般」ウィンドウの「サーバ」タブで設定されています。

Administration Server の再起動オプションの回数が少ないときは再起動の回数を増やして、プロセスが失敗した原因を調べてください。

異常なクラスタが検出された場合は

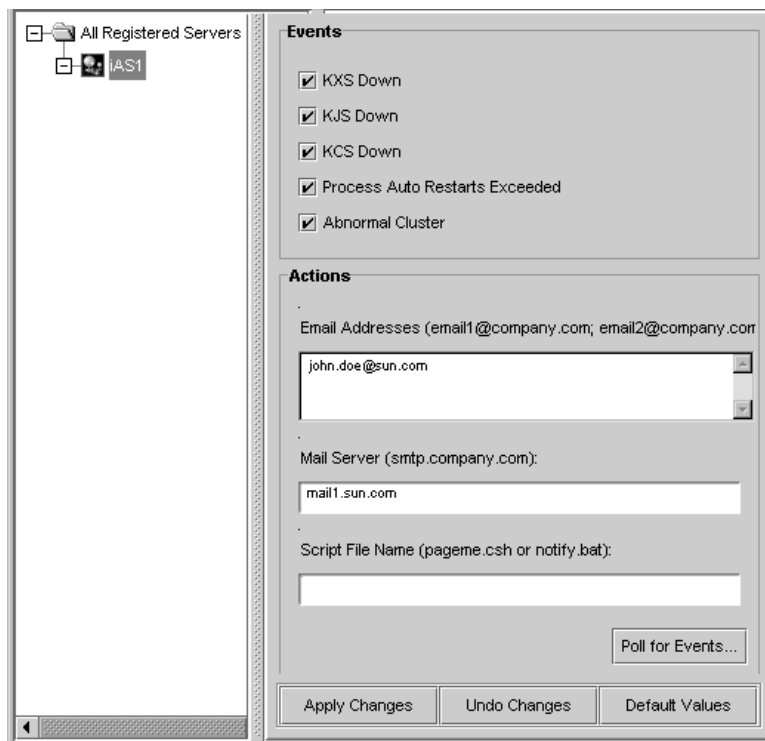
異常なクラスタ状態が検出された場合も通知できます。動作している通常のクラスタ内には、プライマリデータが格納される 1 つの Sync Primary iPlanet Application Server があります。この Sync Primary iPlanet Application Server を使って、すべてのクラスタメンバーが最新の分散データ情報を得ます。異常なクラスタはプライマリの重複またはプライマリなしの状態が検出された場所にあります。

「一般」ウィンドウの「クラスタ」タブにある「異常クラスタの場合は再起動」チェックボックスをオンにします。iPlanet Application Server は、クラスタ内に Sync Primary が 1 つだけ存在するように、適切なプロセスを再起動します。クラスタについては、287 ページの「分散のデータ同期の管理」を参照してください。

イベントに関する電子メール通知を設定するには

イベントに関する電子メール通知を送信するには、次の手順を実行します。

1. iASAT ツールバーの「イベント」をクリックして「イベント」ウィンドウを開きます。
2. 「イベント」ウィンドウの左側のペインでイベントを設定するサーバを選択します。
3. 「イベント」ウィンドウの右側のペインで、次の図のように対応するチェックボックスをクリックすることによって、通知するイベントを選択します。



4. 「電子メールアドレス」フィールドでは、通知の送信先の電子メールアドレスを指定します。複数の電子メールアドレスを使用する場合は、各電子メールアドレスをセミコロン (;) で区切ります。
5. 「メールサーバ」フィールドでは、通知がどのメールサーバを経由して送信されるかを指定します。次の形式を使います。

mail.company.com

6. このサーバに関して送信された最新のイベントを見るには、「イベントのポーリング」をクリックします。

「イベントのポーリング」ダイアログボックスには、選択したサーバの最新のイベントが一覧表示されます。

注 「イベントのポーリング」ボタンをクリックするとイベントが削除されます。つまり、ユーザが参照したイベントは、次にイベントセットを表示したときにはもう表示されません。

7. 「変更の適用」をクリックして変更を保存します。

イベント起動スクリプトを指定するには

イベント通知サービスを設定してスクリプトを実行することができます。スクリプトはシステム管理者に問題点を通知して注意を促したり、重大なイベントに直面したときにシステムが引き続き円滑に稼働できるよう他の自動タスクを実行したりできます。

スクリプトは実行されると、引数を渡して発生したイベントのタイプを示します。たとえば、次のコマンドは Java サーバ (KJS) プロセスがクラッシュしたことを示しています。

```
/script location/ crash kjs
```

イベント通知サービスを設定してイベントに応じたスクリプトを実行するには、次の手順を実行します。

1. iASAT ツールバーの「イベント」をクリックして「イベント」ウィンドウを開きます。
2. 「イベント」ウィンドウの左側のペインからイベントを設定するサーバを選択します。
3. 「イベント」ウィンドウの右側のペインで、スクリプトを起動するイベントのチェックボックスをクリックします。
4. 「スクリプト名」フィールドでは実行するスクリプトのパスを指定します。次のようにします。

```
/mydir/scripts/myscript.pl
```

5. 「変更の適用」をクリックして変更を保存します。

iPlanet Application Server をサードパーティツールを使って監視するよう SNMP を設定する

この章では、SNMP (Simple Network Management Protocol) を設定する方法について説明します。サードパーティの SNMP 管理ツールを使って iPlanet Application Server を監視できます。

この章には次のトピックがあります。

- SNMP について
- SNMP 統計収集を有効にする
- MIB (Management Information Base) について
- マスターエージェントおよびサブエージェントの設定

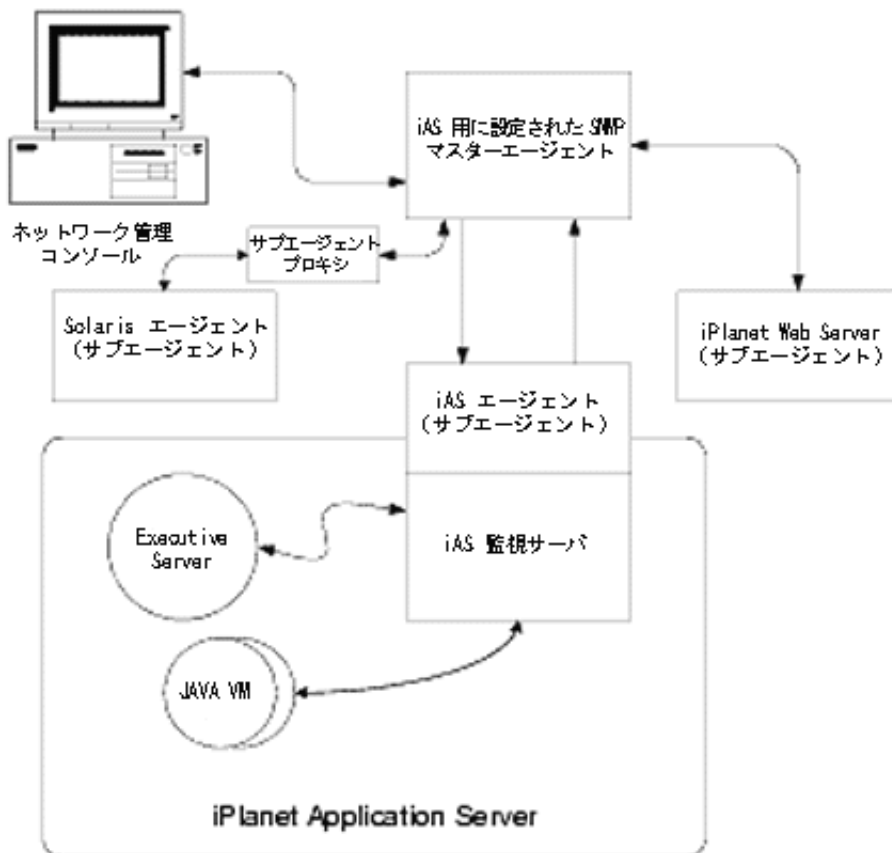
SNMP について

SNMP は、ネットワークの状態に関するデータの交換のために使われるプロトコルです。SNMP によって、データはアプリケーションサーバと、ネットワーク管理ソフトウェアがインストールされているワークステーションの間で移動します。ワークステーションからリモートでネットワークを監視して、サーバ間のネットワークの稼動状況に関する情報を交換できます。たとえば、HP OpenView などのアプリケーションを使って、アプリケーションサーバが受信するエラーメッセージの数やタイプのほかに、どの iPlanet Application Server マシンが実行中かを監視できます。

注 iPlanet Application Server の SNMP サポートは、Windows プラットフォームでは使用できません。

ネットワーク管理ワークステーションは、サブエージェントおよびマスターエージェントの2つのタイプのエージェントを介して、企業内のアプリケーションサーバに関する情報を交換します。サブエージェントは、アプリケーションサーバに関する情報を収集してマスターエージェントにその情報を渡します。マスターエージェントは、各種のサブエージェントとネットワーク管理ワークステーションとの間で情報を交換します。マスターエージェントは、通信するサブエージェントと同じホストマシンで動作します。

図 3-1 SNMP エージェントサポートアーキテクチャ



SNMP 統計収集を有効にする

統計収集を有効にしない限り、iPlanet Application Server SNMP サブエージェントはネットワーク管理ワークステーションに SNMP 統計を報告しません。統計収集が無効の場合は、サブエージェントを起動できません。

注 ネットワーク管理ワークステーションが SNMP 統計を収集できない場合は、次のパスでサーバログ情報を確認します。

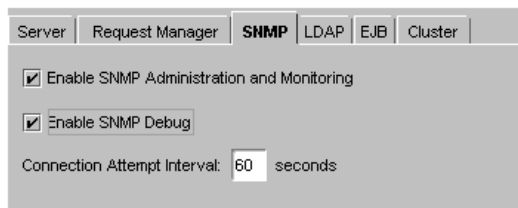
```
<iASInstallDir> /<mail-instanceName>/log/default
```

SNMP データ収集プロセス (snmpcoll) が実行中でない場合は、iPlanet Administration Administration Console を確認して SNMP 有効フラグがオンになっているかどうかを確認します。詳細については、<http://docs.ipplanet.com/> の **Managing Servers with iPlanet Console** を参照してください。

起動サーバを無効にすると、この収集プロセスも無効になります。

データ収集を有効にするには、次の手順を実行します。

1. iPlanet Application Server Administration Tool (IASAT) のツールバーの「一般」ボタンをクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの右側のペインにある「SNMP」タブをクリックします。



3. 「SNMP 管理と監視を有効にする」チェックボックスをオンにします。
これにより、SNMP サブエージェントが有効になり、アプリケーションサーバに関する統計がマスターエージェントに発行されます。
4. マスターエージェントの接続に問題がある場合は、「SNMP デバックを有効にする」をオンにしてエラーメッセージを記録します。
5. 「コネクション試行間隔」テキストフィールドで間隔を指定します。

これは、サブエージェントがマスターエージェントに接続を試みる時の時間間隔です。

注 これらの設定を適用するには iPlanet Application Server サーバを再起動する必要があります。

MIB (Management Information Base) について

iPlanet Application Server はサーバの MIB (Management Information Base) と呼ばれるツリー状の階層構造で、ネットワーク管理に関する変数を格納しています。iPlanet Application Server はこれらの変数を含むメッセージを送信してネットワーク管理ワークステーションに重要なイベントを報告します。ネットワーク管理ワークステーションは、データに関するサーバの MIB のクエリを実行したり、MIB に保存されている変数をリモートで変更したりできます。

この節には次のトピックがあります。

- MIB エントリの書式設定
- サードパーティの SNMP 管理ソフトウェアを使って MIB を有効にする

MIB エントリの書式設定

MIB ファイルには管理されたオブジェクト、つまり変数についての定義が含まれており、サーバに関するネットワーク情報を保存します。各変数の定義には、変数名、データ型、読み込み / 書き込みアクセスレベル、簡単な説明、および永続オブジェクト識別子が含まれています。

次のサンプルエントリは、nsmailEntityDescr 変数の定義を示しています。

```
nasKesMaxThread OBJECT-TYPE      / オブジェクトタイプ
SYNTAX      INTEGER (SIZE (1..512))      / 構文
ACCESS      read-write      / 読み込み / 書き込みアクセス
           レベル
STATUS      mandatory      / ステータス
DESCRIPTION      / 説明
           "The maximum number of threads used to serve requests."
 ::= { kes 4 } / オブジェクト ID
```

この定義には次の情報があります。

- オブジェクトタイプ: 変数名を指定する。この例では、`iasKesMaxThread`
- 構文: ASN.1 表記法で、さまざまなオブジェクトタイプの抽象データタイプを指定する。たとえば、`nasKesMaxThread` 変数の構文は `INTEGER (SIZE (1..512))`
- アクセス: 変数のアクセスレベルを指定する。指定可能なアクセスレベルは、`read-only` (読み込み専用)、`read-write` (読み込み / 書き込み)、`write-only` (書き込み専用)、または `not-accessible` (アクセス不可)
- ステータス: 要素が必須か、オプションか、または古いかを知らせる
- 説明: 二重引用符で囲まれた要素の説明テキスト。たとえば、`nasKesMaxThread` 変数の説明は、`"The maximum number of threads used to serve requests."` となる
- オブジェクト ID: MIB 名前ツリー内で各管理対象オブジェクトの永続 ID として名前空間に割り当てられた名前。SNMP にあるオブジェクトは階層構造になっている。オブジェクト ID は、階層内のオブジェクトを表す一連のラベル。たとえば、`nasKesMaxThread` は `kes 4` として識別される。これは、`nasKesMaxThread` がサブツリー `kes` ではラベル `4` であることを表す

`kes` は、`kesTable subtree` 内でラベル `4` を持つ。

サードパーティの SNMP 管理ソフトウェアを使って MIB を有効にする

MIB の有効化に関する詳しい手順については、SNMP 管理ソフトウェアを参照してください。通常は、iPlanet Application Server の MIB を Network Management マシンにコピーして、その後それを SNMP 管理ソフトウェアの MIB データベースに読み込みます。

iPlanet Application Server の MIB データベースは次の場所から入手できます。

```
<iASInstallDir>/plugins/snmp/
```

マスターエージェントおよびサブエージェントの設定

SNMP マスターエージェントは Solaris オペレーティングシステムにネイティブです。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コンフィグレーションファイルで定義されています。COMFIG ファイルは手動で編集できます。

注 この手順は、推奨パッチを適用した Solaris 2.6 が動作していることを前提とします。iPlanet Web Server がインストールされていることも前提とします。

この節には次のトピックがあります。

- SNMP マスターエージェントを設定するには
- SNMP マスターエージェントを起動するには
- SNMP 設定を確認するには

SNMP マスターエージェントを設定するには

マスター SNMP エージェントを設定するには、次の手順を実行します。

1. root としてログインします。
2. ポート 161 に実行中の Solaris SNMP デーモン (snmpdx) があることを確認します。
SNMP デーモンが実行中の場合は、再起動する方法とサポートする MIB ツリーを確認してください。その後でプロセスを終了します。
3. /etc/rc3.d にある Solaris SNMP デーモンの起動ファイル s76snmpdx を編集して、デーモンが待機するポートを変更します。

開始セクションで、次のどちらかの行、

```
/usr/lib/snmp/snmpdx -y -c /etc/snmp/conf
```

または

```
/usr/lib/snmpdx -p 161 -y -c /etc/snmpconf
```

を

```
/usr/lib/snmp/snmpdx -p 1161 -y -c /etc/snmp/conf に置き換えます。
```

これで、デーモンが待機するポートが 161 から 1161 に変更されました。

4. サーバのルートディレクトリの <iASInstallDir>/ias/snmp にある CONFIG ファイルを編集します。

CONFIG ファイルでは、マスターエージェントとともに動作するコミュニティおよびマネージャを定義します。マネージャの値は、有効なシステム名または IP アドレスにしてください。基本的な CONFIG ファイルの例は次のとおりです。

```

COMMUNITY      public
                ALLOW ALL OPERATIONS

MANAGER        your_manager_station_name
                SEND ALL TRAPS TO PORT 162
                WITH COMMUNITY public
    
```

5. (オプション) CONFIG ファイルにある sysContact および SysLocation 変数を定義します。

CONFIG ファイルを編集して、MIB-II 変数 sysContact および sysLocation を指定する、sysContact および sysLocation の初期値を追加できます。この例の sysContact および sysLocation の文字列が二重引用符で囲まれていることに注意してください。スペース、改行、タブなどが含まれている文字列は、二重引用符で囲む必要があります。16 進数の値を指定することもできます。

この例の CONFIG ファイルでは、sysContract および sysLocation 変数が定義されています。

```

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@iPlanet.com>"
    
```

カプセル化機能は、マスターエージェントからポート 1161 で現在待機している Solaris エージェントにリクエストを転送します。

6. ファイル CONFIG_SAGT を編集して次の行を変更します。

Agent at 1161 with Community Public

これは、ポート 1161 に Solaris エージェントを待機させるようサブエージェントを設定します。

Subtrees <list of oids>

これは、Solaris エージェントによって提供された SNMP サブツリーを設定します。

Forward All Traps

これにより、Solaris エージェントによって送信されたすべてのトラップが確実にマスターエージェントに転送されます。

SNMP マスターエージェントを起動するには

SNMP マスターエージェントをインストールすると、手動または iPlanet Console を使って SNMP マスターエージェントを起動できます。

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 が存在しない場合は、マスターエージェントを最初に起動したときに作成されます。CONFIG ファイルで無効なマネージャ名が指定されている場合は、マスターエージェントの起動に失敗します。

注 INIT にはローカルシステムに関する情報が格納されています。このファイルは、マスターエージェントを最初に起動するときに作成されます。このファイルをほかのマシンにコピーしないでください。

サーバを起動したときにマスターエージェントを自動的に起動するには、次の手順を実行します。

1. ファイル `ias/snmp/k75snmpmagt` および `ias/snmp/s75snmpmagt` を編集します。
2. `$GX_ROOTDIR` 変数がまだルート環境に定義されていない場合は、この変数を iPlanet Application Server インストールディレクトリパスに変更します。

3. `k75snmpmagt` を `/etc/rc2.d` に、`s75snmpmagt` を `/etc/rc3.d` にコピーします。

マスターエージェントを標準以外のポートで手動で起動するには、次の2つの方法のどちらかを使います。

- 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` リクエストを待機する各インターフェイスのトランスポートマッピングを指定します。トランスポートマッピングを使うと、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、1 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決まります。トランスポートマッピングのエントリの例を次に示します。

<code>TRANSPORT</code>	<code>extraordinary SNMP</code>
	<code>OVER UDP SOCKET</code>
	<code>AT PORT 11161</code>

`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、手動でマスターエージェントを起動してください。

```
# 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もコネクションを受け入れることができるようにします。

SNMP 設定を確認するには

この章で説明されている手順を実行すると、`SNMP` セットアップを確認することができます。

`SNMP` を確認するには、次の手順を実行します。

1. `iPlanet Application Server` を終了します。`iPlanet Application Server` の起動と終了についての詳細は、26 ページの「`iASAT` によるサーバの起動と終了を行うには」を参照してください。

また、すべての `iPlanet Application Server` プロセス (`KAS`、`KXS`、`KJS`、および `KCS`) が終了していることを確認します。

2. Directory Server が動作していることを確認します。動作していない場合は、次のコマンドを実行して起動します。

```
<iASInstallDir>/slapd-<hostname>/start-slapd
```

3. iPlanet Web Server (iWS) (例: https-servername) が動作していることを確認します。動作していない場合は、次のようにサーバを起動します。

```
/usr/iplanet/suitespot/https-solsystems/start
```

ここで、solsystem はサーバの名前です。

4. UNIX の ps コマンドで、次のように、Solaris SNMP エージェント (snmpdx) が動作していることを確認します。

```
ps -ef | grep snmpd
```

動作していない場合は、次のようにサーバを起動します。

```
/etc/rc3.d/S76snmpclx start
```

5. UNIX の ps コマンドで、次のように、iPlanet Application Server Master Agent (magt) および encapsulator/proxy subagent (sagt) が動作していることを確認します。

```
ps -ef | grep magt
```

```
ps -ef | grep sagt
```

動作していない場合は、次のようにサーバを起動します。

```
/etc/rc3.d/S75snmpmagt start
```

6. iPlanet Application Server を起動します。
7. サードパーティの SNMP 管理ソフトウェアの MIB ブラウザまたはテストユーティリティ (たとえば snmpwalk) を使って、SNMP データの収集中であることを確認します。

サーバメッセージの記録

この章では、iPlanet Application Server によって提供されるメッセージロギングサービスについて説明します。

この章には次のトピックがあります。

- ロギングサービスについて
- アプリケーションイベントのロギングについて
- HTTP リクエストのロギングについて
- DSync メッセージのロギングについて

ロギングサービスについて

サーバのメッセージを記録するには、iPlanet Application Server のメッセージロギングサービスを使用します。ロギングサービスは iPlanet Application Server Administration Tool の「ログ」メニューで設定します。

ロギングを有効にすると、iPlanet Application Server は、アプリケーションレベルおよびシステムレベルのサービスによって生成されたメッセージを記録します。これらのメッセージは、サービス実行時に発生するイベントについて記述されています。

この節には次のトピックがあります。

- 記録するメッセージタイプを決定するには
- ログメッセージの書式設定
- iPlanet Application Server ログファイルの監視

記録するメッセージタイプを決定するには

iPlanet Application Server サービスによって生成される 3 種類のメッセージを記録できます。次の表で各種類について説明します。

表 4-1 メッセージタイプの記録

メッセージタイプ	説明	表示されるタイミング
情報メッセージ	ステータス更新などのリクエストや通常のサービス状態のプロセスについて記述される	問題が発生していないとき
警告メッセージ	重大ではないが、大きな問題になる可能性がある問題について記述される	プロセスに接続できないときなど、サービスが一時的な問題に遭遇したとき
エラーメッセージ	修復できそうにないサービスの重大な障害について記述される	サービスがパイプ閉鎖などの重大な問題に遭遇したとき

ロギングサービスを使って、エラーメッセージだけ、エラーメッセージと警告メッセージ、またはすべてのメッセージを記録できます。記録するメッセージのタイプを選択するには、次の手順を実行します。

1. iPlanet Application Server Administration Tool (iASAT) ツールバーの「ログ」をクリックして「ログ」ウィンドウを開きます。
2. 「ログ」ウィンドウの左側のペインで、ログ設定を指定する iPlanet Application Server ノードを選択します。
3. 「ログ」ウィンドウの右側のペインで「サーバイベント」タブをクリックします。次のダイアログボックスが表示されます。

The screenshot shows the 'Server Event' configuration window with three tabs: 'Server Event', 'HTTP', and 'DSync'. The 'Server Event' tab is active. It contains the following settings:

- Enable Server Event Log
- Log Target**
 - Log to a Database
 - Data Source: eventlog
 - Username: kdemo
 - Database: ksample
 - Password: [empty]
 - Table Name: eventlog
 - Log to Console
 - Log Errors to WinNT Application Log
 - Log to file
 - File name: logs\ias
- Enable File Rotation: Yes (dropdown)
- Rotation Interval: Every Tuesday (dropdown)

- General**
- Message Type: Errors and Warnings (dropdown)
- Maximum Entries: 100
- Write Interval: 60

4. イベントログを有効にするには、「サーバイベントログを有効にする」チェックボックスをオンにします。
5. 「一般」ペインの「メッセージタイプ」ドロップダウンボックスから「エラー」、「エラーと警告」、または「すべてのメッセージ」を選択します。
6. 「最大エントリ数」テキストフィールドには、データがログに書き込まれるときまで存在可能な最大エントリ数を指定します。
7. 「書き込み間隔」フィールドには、データがログに書き込まれるまでの経過時間(秒単位)を入力します。

これらの設定は、ロギング可能な3つのサービス(アプリケーション、HTTP、およびDSync)のすべてに適用されます。

ログメッセージの書式設定

各ログメッセージには次の4つのコンポーネントがあります。

- メッセージが作成された日時

- 情報、警告、エラーなどのメッセージタイプ
- サービスまたはアプリケーションコンポーネント ID
- メッセージテキスト

ログメッセージをテキストベースの送信先ログに送信すると、次のように書式設定されます。

[Date and time of message] Message type: サービス ID: メッセージテキスト

たとえば、ASCII テキストファイルに送信された次のメッセージは、メッセージの書式設定を示します。

```
[01/18/00 11:11:12:0] info (1):GMS-017:server shutdown (host 0xc0a801ae, port 10818, group 'iAS') - updated host database
```

```
[01/18/00 11:11:18:2] warning (1):GMS-019:duplicate server (host 0xc0a8017f, port 10818) recognized, please contact iPlanet Communications for additional licenses
```

iPlanet Application Server ログファイルの監視

この節では、Windows および Solaris でアプリケーションサーバログを表示する方法について説明します。多くの場合、ログファイルを確認すると、アプリケーションおよびサーバの設定に関する問題を解決できます。次の節では、Windows および Solaris の両方のシステムで iPlanet Application Server ログを監視する方法について詳しく説明します。

この節には次のトピックがあります。

- Solaris で Application Server ログを監視するには
- Windows でアプリケーションサーバログを監視するには

Solaris で Application Server ログを監視するには

Solaris では、コマンドラインツールを使って iPlanet Application Server ログを表示できます。このツールを使うと、メッセージの表示を選択できます。たとえば、特定の期間に作成されたメッセージを表示したり、ファイル内の最初の数個のメッセージだけを表示したりできます。

Solaris で iPlanet Application Server ログを監視するには、次の手順を実行します。

1. 次のパスに移動します。

```
<iasinstsall>ias/logs
```

- 「tail - 」と入力し、表示するログファイルの名前をそのあとに入力します。たとえば、KJS ログを表示するには、「tail -kjs」と入力します。

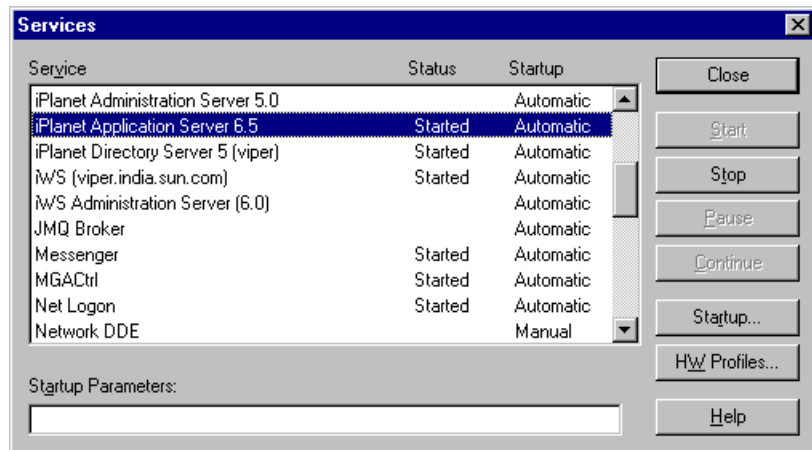
コマンド実行を停止するには、CTRL+C キーを押します。

注 tail コマンドで実行可能なすべてのオプションの完全なリストを表示するには、ターミナルウィンドウを開き、「man tail」と入力します。

Windows で アプリケーションサーバログを監視するには

NT でアプリケーションサーバログの表示を有効にするには、次の手順を実行します。

- 「スタート」メニューから「設定」>「コントロールパネル」を選択します。
- 「サービス」アイコンをダブルクリックします。次の図のように「サービス」ウィンドウが表示されます。



- 「iPlanet Application Server 6.5」を選択します。
- 「スタートアップ」をクリックします。次のように「サービス」ウィンドウが表示されます。



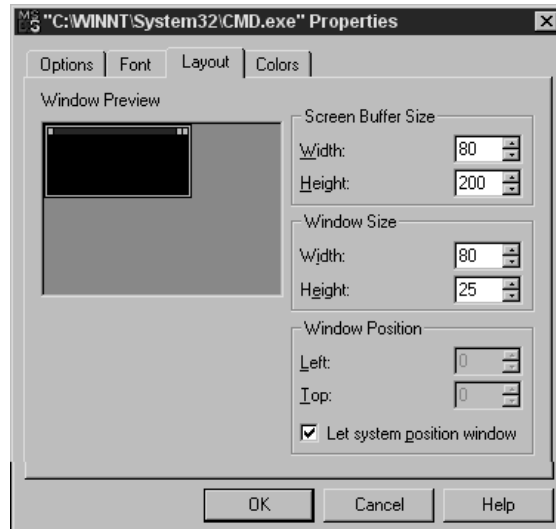
5. 図に示すように、「デスクトップとの対話をサービスに許可」チェックボックスをオンにします。
6. 「OK」をクリックしてウィンドウを閉じます。
7. 「サービス」ウィンドウで「停止」をクリックして iPlanet Application Server を停止します。
8. 「開始」をクリックして iPlanet Application Server を再起動します。

これを実行すると、DOS ウィンドウに iPlanet Application Server ログメッセージが表示されます。iPlanet Application Server の各物理プロセスは独立したウィンドウに表示されます。たとえば、KAS ログ、KJS ログ、および KCS ログを独立したウィンドウに表示できます。

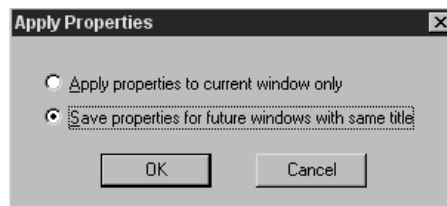
出力ウィンドウの垂直スクロールバーを有効にするには

DOS ウィンドウに表示されるログメッセージは連続してスクロールします。出力ウィンドウで垂直スクロールバーを有効にすると、メッセージ表示を上下にスクロールできます。DOS ウィンドウで垂直スクロールバーを有効にするには、次の手順を実行します。

1. 出力ウィンドウの左上にある DOS アイコンをクリックします。
2. 「プロパティ」を選択します。次の図のように「プロパティ」ウィンドウが表示されます。



3. 「レイアウト」タブを選択します。
4. 「画面バッファのサイズ」の「高さ」を200以上に設定します。
5. 「OK」をクリックして変更を登録します。次のような確認ウィンドウが表示されます。



6. すべての DOS ウィンドウでスクロールバーを有効にするには、「すべてのウィンドウ起動時に設定を有効にする」ラジオボタンをオンにします。

アプリケーションイベントのロギングについて

メッセージログは、アプリケーションエラーを追跡してデバッグするのに有効です。log() 方式を使うことによって、アプリケーション開発者は、サーバ管理者が iPlanet Application Server サービスに設定したものと同一ログの送信先にメッセージを送信できます。

たとえば、アプリケーションがコードのセグメント内で問題に遭遇した場合は、関連するエラーメッセージを記録できます。エラーメッセージよりアプリケーションのステータスについての情報メッセージの方が役立ちます。

iPlanet Application Server の管理サービスは Administration Server プロセス (KAS) で実行されます。Administration Server を使うと、サーバおよびアプリケーションをリモートで管理できます。また、KAS では、アプリケーションのパーティション分割、イベントログ、リクエストの監視、主なサーバ設定値の動的設定などのサービスもサポートしています。管理サービスにアクセスするクライアントには、iASAT、ディレクトリサーバ、およびサードパーティ SNMP エージェントがあります。

KAS によって監視されて記録されるイベントの中には次のものがあります。

- サーバログインおよびログアウトの詳細
- サーバおよびエンジンの初期化の状態、起動、シャットダウン、有効 / 無効の切り替え
- KAS、SNMP、トランザクションマネージャ、およびエンジンに関連する問題
- iPlanet Application Server インストールに関連する問題
- 配置動作および配置時のエラー
- LDAP、データベース、データベースクライアント、トランザクション、リソースマネージャ、ログ、ロードバランス、エンジン、および KAS の設定パラメータの取得時に発生する可能性があるエラー

管理および配置に関するログは、KAS (Administration Server) に保存されます。そのほかに、メッセージおよびイベントログが保存されるサーバとして、KJS (Java サーバ)、KCS (C++ サーバ)、および KXS (Executive Server) があります。KAS はこれらの 3 つのサーバを起動し、それらのサーバの稼動状況を監視します。

この節には次のトピックがあります。

- ログ送信先の決定
- ログファイルの管理

ログ送信先の決定

ロギングサービスを設定して、次の表に示すどれかまたはすべての送信先に、サーバおよびアプリケーションのメッセージを記録できます。

表 4-2 メッセージログの送信先

ログの送信先	説明	使用の目的
プロセスコンソール	ログメッセージが作成されると、iPlanet Application Server プロセスコンソールにログメッセージが表示される	デフォルト値。ロギングが有効で、サーバの自動起動 (UNIX) またはデスクトップとの対話 (NT) が有効な場合は、コンソールが開いてログメッセージが表示される。この機能を無効にするには、「コンソールにログを記録」チェックボックスをオフにする
アプリケーションログ	デフォルトのアプリケーションログファイル。Windows NT の場合は、イベントビューアを使って表示できる	デフォルト値。警告および情報メッセージはアプリケーションログに記録されない。アプリケーションログには、より包括的なサーバおよびアプリケーションエラーメッセージが記録される すべてのメッセージはタイムスタンプによってソートされる
ASCII テキストファイル	ASCII テキストファイル。ユーザが作成して指定する必要がある	サーバおよびアプリケーションメッセージだけを記録するときを使う。すべてのメッセージはタイムスタンプによってソートされる
データベーステーブル	データベーステーブル。ユーザが作成して指定する必要がある	より融通性の高いログの送信先。記録されたメッセージのレポートをソート、グループ化、および作成するときを使う

ロギングを有効にすると、コンソールが開いていてコンソールロギングが有効な限り、ロギングサービスは自動的に Windows および Solaris プラットフォームのプロセスコンソールにメッセージを送信します。Windows では、ロギングサービスはアプリケーションログにもメッセージを送信します。プロセスコンソールに記録しようとしても、メッセージは記録されません。メッセージが画面からスクロールして消えてしまうと、メッセージを取得できません。

注 iPlanet Application Server は、メッセージがアプリケーションログ、ASCII ファイル、データベースログなどに書き込まれる前に、ログバッファを使ってメッセージを保存します。ログが継続的に更新されると、この保存に要する時間がアプリケーションサーバのパフォーマンスに悪影響を及ぼします。

このバッファは、ログが更新されるまでメッセージを保存することによってリソースの使用を制限します。バッファ間隔がタイムアウトになったときや、バッファのエントリ数が許容最大数を越えたときに、バッファは送信先に書き込まれます。

ログの送信先を指定するには

ログングサービスを有効にしてログメッセージの送信先を指定するには、次の手順を実行します。

1. iASAT ツールバーの「ログ」をクリックして「ログ」ウィンドウを開きます。
2. ログ設定を指定する iPlanet Application Server インスタンスを選択します。
3. 「ログ」ウィンドウの右側のペインで「サーバイベント」タブをクリックします。
4. 「サーバイベント」タブの「サーバイベントログを有効にする」チェックボックスをオンにします。
5. 「ログの対象」ボックスでは、必要なチェックボックスをオンにすることによって、有効にするログの種類を選択します。
 - 「データベースにログを記録」
すべてのメッセージとエラーがデータベース形式のファイルに保存されます。
 - 「コンソールにログを記録」
すべてのメッセージとエラーが、それぞれのプロセスのコンソールウィンドウに表示されます。
 - 「Windows NT アプリケーションログにエラーログを記録」
このオプションは Windows でのみ適用されます。メッセージとエラーをイベントビューアで表示できます。イベントビューアを起動するには、「スタート」>「プログラム」>「管理ツール」>「イベントビューア」を選択します。
 - 「ファイルにログ」
すべてのメッセージとエラーは、指定したファイルに記録されます。
ログファイルは、「ファイルにログ」オプションを有効にしたときに作成されます。詳細については、85 ページの「ファイルにログを記録するには」を参照してください。

注 Windows で KAS 監査ログを有効にしてファイルに出力するには、IAS_KASLOGFILE 環境変数を 1 に設定します。デフォルトでは 0 に設定されています。詳細については、86 ページの「Windows でファイルにログを記録するには」を参照してください。

ログファイルの管理の詳細については、86 ページの「ログファイルの管理」を参照してください。

この節には次のトピックがあります。

- データベースへの記録
- コンソールへの記録
- サーバイベントのファイルへの記録

データベースへの記録

アプリケーションサーバのメッセージをデータベースに記録する場合は、イベントログデータベーステーブルを作成する必要があります。次の表は、4 つのフィールド名についての説明と各フィールドのデータタイプを示しています。

注 Windows システムおよび Solaris システムでは、提供されたスクリプトを使って、イベントログ (eventlog) テーブルおよび HTTP ログ (httplog) テーブルを自動的にセットアップできます。

スクリプトはディレクトリ <iASInstallDir>/ias/ias-Samples/dblog/src/schema にあり、スクリプト名は Log_db2.sql、Log_ifmx.sql、Log_mssql.sql、Log_ora.sql、および Log_syb.sql です。

使用しているデータベースに適したスクリプトを選択してください。

表 4-3 データベーステーブルへの記録

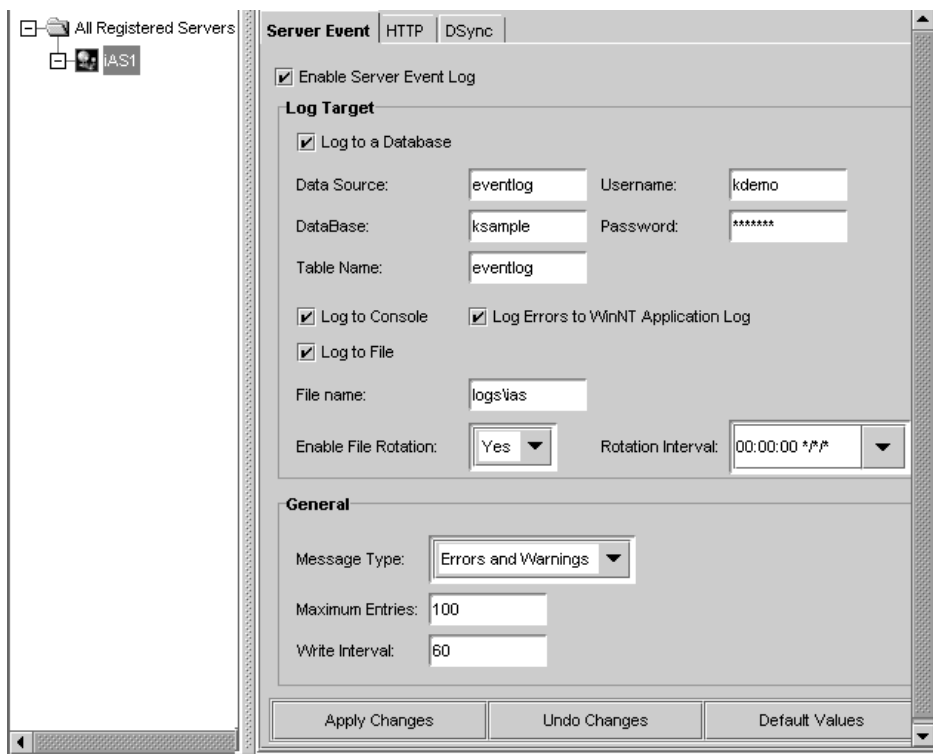
データベースフィールド名	説明	データタイプ
evtttime	メッセージが作成された日時	Date/Time
evttype	情報、警告、エラーなどのメッセージタイプ	Number
evtcategory	サービスまたはアプリケーションコンポーネント ID	Number
evtstring	メッセージテキスト	Text
evtid	イベント ID	Number

ロギングサービスは、表に示されているデータベースフィールドにメッセージコンポーネントを割り当てます。データベーステーブルでは、これらの正確なフィールド名またはカラム名を使う必要があります。

データベースにログを記録するには

データベースにログ記録するには、次の手順を実行します。

1. iASAT ツールバーの「ログ」をクリックして「ログ」ウィンドウを開きます。
2. 「ログ」ウィンドウの左側のペインで、ログ設定を指定する iAS ノードを選択します。
3. 「ログ」ウィンドウの右側のペインで「サーバーイベント」タブをクリックします。
4. 「サーバーイベント」タブの「サーバーイベントログを有効にする」チェックボックスをオンにします。
5. 「ログの対象」ボックスでは、次の図のように「データベースにログを記録」チェックボックスをクリックします。



データベースにアクセスするのに必要なデータソース、データベース名、テーブル名、およびユーザ名とパスワードを指定します。

6. 「一般」ボックスの「メッセージタイプ」ドロップダウンボックスから「エラー」、「エラーと警告」、または「すべてのメッセージ」を選択します。
7. 「変更の適用」をクリックして iASAT への変更を保存します。

コンソールへの記録

生成されたログ出力を表示するときは、「コンソールにログ」チェックボックスをオンにします。モニタのコンソールウィンドウにログメッセージが表示されます。

コンソールにログを記録するには

ログ情報をファイルに記録するには、次の手順を実行します。

1. iASAT ツールバーの「ログ」をクリックして「ログ」ウィンドウを開きます。
2. 「サーバイベント」タブをクリックします。
3. 「サーバイベントログを有効にする」チェックボックスをオンにします。
4. 「ログの対象」セクションで、「コンソールにログ」を選択します。
5. 「一般」ボックスの「メッセージタイプ」ドロップダウンボックスから「エラー」、「エラーと警告」、または「すべてのメッセージ」を選択します。
6. 「変更の適用」をクリックして iASAT への変更を保存します。

サーバイベントのファイルへの記録

すべての KAS メッセージをログファイルに記録するには、ファイルとして保存する KAS ログを有効にする必要があります。

iPlanet Application Server Administration Tool を使ってファイルへのログングを有効にすると、KJS、KXS、および CJS に関するメッセージが記録されます。ただし、KAS はすべてのサーバエンジンと管理機能を管理するので、生成されるメッセージは膨大な量になることがあります。

Solaris では、デフォルトで、すべての KAS メッセージがファイルに記録されます。Windows プラットフォームでは、KAS メッセージのファイルへのログングを手動で有効にする必要があります。詳細については、86 ページの「Windows でファイルにログを記録するには」を参照してください。

ファイルにログを記録するには

iASAT の監視サービスを使うと、サーバの稼動状況に関する情報をファイルに記録できます。

情報をファイルに記録するには、次の手順を実行します。

1. iASAT ツールバーの「ログ」をクリックして「ログ」ウィンドウを開きます。
2. 「ログ」ウィンドウの左側のペインで「サーバイベント」タブをクリックします。
3. 「サーバイベント」タブの「サーバイベントログを有効にする」チェックボックスをオンにします。
4. 「ログの対象」ペインで、「ファイルにログ」チェックボックスをオンにします。
5. 「ファイルにログ」テキストフィールドにログファイルの名前を入力します。
6. 「一般」ボックスの「メッセージタイプ」ドロップダウンボックスから「エラー」、「エラーと警告」、または「すべてのメッセージ」を選択します。
7. 「変更の適用」をクリックして iASAT への変更を保存します。

Windows でファイルにログを記録するには

Windows プラットフォームでファイルへの KAS ロギングを有効にするには、次の手順を実行します。

1. 「コントロールパネル」を開いて「システム」アイコンをダブルクリックします。
2. 「環境」タブに移動します。
3. 「System Variables」リストで IAS_KASLOGFILE を選択します。
4. デフォルト値を 1 に変更し、「設定」をクリックします。

KAS ログメッセージは、次のパスにある KAS.log ファイルに記録されます。

```
<iASInstallDir>/ias6/ias/logs/
```

ログファイルの管理

設定された期間の終了時にログファイルをローテートできます。たとえば、現在のログだけがログファイルに存在するように、ログファイルのローテーションを 10 日ごとまたは毎週水曜日に指定できます。ログファイルには作成された日時がスタンプされるため、ログファイルローテーションはログファイルを管理可能な単位に構成します。

ASCII ファイルにあるサーバメッセージの記録を選択すると、ログファイルのローテーションを有効にしてエントリを定期的に調整できます。

ログファイルをローテートするには

ログファイルのローテーションを設定するには、次の手順を実行します。

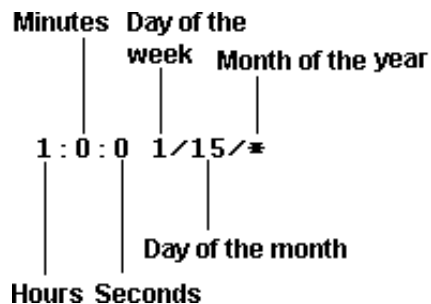
1. iASAT ツールバーの「ログ」をクリックして「ログ」ウィンドウを開きます。

2. ログ設定を指定する iPlanet Application Server ノードを選択します。次の図が表示されます。

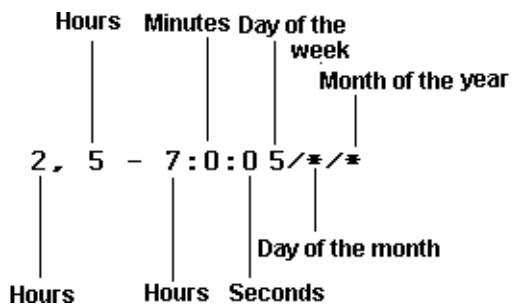
The screenshot shows the configuration interface for the 'Server Event' log. It includes tabs for 'HTTP' and 'DSync'. The 'Log Target' section has several options: 'Log to a Database' (unchecked), 'Log to Console' (unchecked), 'Log Errors to WinNT Application Log' (unchecked), and 'Log to file' (checked). The 'Log to file' option is configured with 'File name: logsvias' and 'Enable File Rotation: Yes'. The 'Rotation Interval' is set to 'Every Hour'. The 'General' section is also visible, with 'Message Type' set to 'Errors and Warnings', 'Maximum Entries' set to 100, and 'Write Interval' set to 60. A dropdown menu for 'Rotation Interval' is open, showing a list of options: 'Every Hour', 'Every Day', 'Every Monday', 'Every Tuesday', 'Every Wednesday', 'Every Thursday', 'Every Friday', and 'Every Saturday'.

3. 「ログ」ウィンドウの右側のペインで「サーバイベント」タブをクリックします。
4. 「サーバイベント」タブの「サーバイベントログを有効にする」チェックボックスをオンにします。
5. 「ファイルにログ」チェックボックスをオンにします。
6. 「ファイルローテーションを有効にする」ドロップダウンリストから「はい」を選択します。
7. 「ローテーション間隔」ドロップダウンリストからログファイルのローテーション間隔を選択します。ログファイルのローテーション時間を示す文字列を入力することもできます。

たとえば、次の文字列は、新しいファイルへの記録が毎月 15 日と、毎週月曜日の午前 1 時に始まることを示しています。



次の文字列は、新しいファイルへの記録が毎週金曜日の午前2時、午前5時、午前6時、および午前7時に始まることを示しています。



8. 「一般」 ペインの「メッセージタイプ」ドロップダウンボックスから「エラー」、「エラーと警告」、または「すべてのメッセージ」を選択します。
9. 「変更の適用」をクリックして iASAT への変更を保存します。

HTTP リクエストのロギングについて

Web サーバリクエストは Web コネクタプラグインによって監視されます。プラグインでは、リクエストを、そのリクエストが処理される iPlanet Application Server インスタンスに送信します。Web サーバリクエストを記録することによって、リクエストパターンおよびほかの重要なリクエストの情報を記録できます。

この節には次のトピックがあります。

- Web リクエストの記録方法
- Web サーバリクエストを記録するには

Web リクエストの記録方法

Web サーバリクエストはコンポーネントに分割されます。これらのコンポーネントは、Web リクエストを管理するために、Web サーバが使用する標準化された HTTP 変数です。iPlanet Application Server には、記録するこれらの HTTP 変数のサブセットが含まれています。追加情報を記録する必要がある場合は、そのリストに変数を追加できます。

注 Windows システムおよび Solaris システムでは、提供されたスクリプトを使って、HTTP ログテーブルおよびイベントログテーブルを自動的にセットアップできます。詳細は、84 ページの「データベースにログを記録するには」を参照してください。

各 HTTP 変数は、ユーザが作成したテーブル内のデータベース フィールド名にマッピングされる必要があります。たとえば、Web サーバリクエストのコンテンツの長さを記録するには、CONTENT_LENGTH 変数を content_length のような名前 text データタイプとして定義されたデータベースフィールドにマッピングします。次の表は、iPlanet Application Server によって使われるデフォルトの HTTP 変数およびそのデータベースのデータタイプを示します。この表は、Web リクエストを記録するデータベーステーブルの作成に役立ちます。

表 4-4 HTTP 変数およびデータベースのデータタイプ

デフォルトの HTTP 変数	デフォルトのデータベースフィールド名	データタイプ
適用なし	logtime	Date/Time
CONTENT_LENGTH	content_length	Number
CONTENT_TYPE	content_type	Text

表 4-4 HTTP 変数およびデータベースのデータタイプ (続き)

デフォルトの HTTP 変数	デフォルトのデータベースフィールド名	データタイプ
HTTP_ACCEPT	accept	Text
HTTP_CONNECTION	connection	Text
HTTP_HOST	host	Text
HTTP_REFERER	referer	Text
HTTP_USER_AGENT	user_agent	Text
PATH_INFO	uri	Text
REMOTE_ADDR	ip	Text
REQUEST_METHOD	method	Text
SERVER_PROTOCOL	protocol	Text

データベーステーブルの logtime 以外のすべてのデータベースフィールド名を変更できます。ロギングサービスは、メッセージが作成された時刻を logtime データベースフィールドに割り当てます。

データベーステーブルのフィールドは、自動的にレジストリの Web サーバ変数に割り当てられます。

インストール済みの正しく設定された NSAPI、ISAPI などの Web サーバ通信プラグインモジュールを持つ必要があります。

Web サーバリクエストを記録するには

Web サーバリクエストを記録する前に、リクエストメッセージを保持するデータベーステーブルを作成する必要があります。このテーブルの作成の詳細は、89 ページの「Web リクエストの記録方法」を参照してください。

Web サーバリクエストを記録するには、次の手順を実行します。

1. iASAT ツールバーの「ログ」をクリックして「ログ」ウィンドウを開きます。
2. 「ログ」ウィンドウの左側のペインから Web サーバリクエストを記録する iPlanet Application Server を選択します。

複数の iPlanet Application Server インスタンスがある場合は、1 つのサーバを指定するとすべての Web サーバリクエストを記録できます。

3. 「ログ」ウィンドウの右側のペインで「HTTP」タブをクリックします。

次のウィンドウが表示されます。

The screenshot shows the 'Log to a Database' dialog box in Oracle Enterprise Manager. The left pane shows a tree view with 'All Registered Servers' and 'iAS1'. The right pane has tabs for 'Server Event', 'HTTP', and 'DSync'. The 'HTTP' tab is selected. A checkbox 'Enable HTTP Log' is checked. Below it, the 'Log to a Database' section contains fields for 'Data Source' (httplog), 'Username' (kdemo), 'DataBase' (ksample), and 'Table Name' (httplog). The 'Password' field is empty. The 'General' section has 'Maximum Entries' set to 100 and 'Write Interval' set to 60. At the bottom are buttons for 'Apply Changes', 'Undo Changes', and 'Default Values'.

4. 「データソース」フィールドに「httplog」と入力します。
5. 「データベース」フィールドには、データベースの接続に使う情報を入力します。たとえば、Oracle データベースについてはこれが Oracle SID になります。
6. 「テーブル名」フィールドに「httplog」と入力します。
7. データベースの接続に使うユーザ名とパスワードを入力します。
8. 「最大エン트리数」フィールドに数値を入力します。この数は、データがログに書き込まれるときまで存在可能な最大エン트리数を表します。
9. 「書き込み間隔」テキストフィールドには必要な数値を入力します。この数は、データがログに書き込まれるまでの経過時間を示します。
10. 「変更の適用」をクリックして変更を保存します。

DSync メッセージのロギングについて

iPlanet Application Server では、パーティション分割されたおよび分散されたアプリケーションについて、複数の iPlanet Application Server にわたる分散データ同期 (DSync) をサポートしています。DSync では、iPlanet Application Server プロセス全体でのクラスタ管理およびデータ同期を提供します。iASAT では DSync メッセージを記録します。

分散データ同期の詳細は、287 ページの「分散のデータ同期について」を参照してください。

この節には次のトピックがあります。

- DSync メッセージの記録方法
- DSync メッセージを記録するには

DSync メッセージの記録方法

DSync は、記録するコンポーネントを選択できる、コンポーネントベースのアーキテクチャを備えています。すべての DSync デバッグメッセージが KXS、KCS、および KJS ログファイルに表示されます。DSync デバッグコンポーネントは次のとおりです。

- モジュール: データ管理機能を提供し、ほかの DSync コンポーネントをログファイルに追加する。有効な場合、DSync によって実行されるメソッドが記録される
- フェールオーバー: クラスタメンバー管理機能を提供する。有効な場合、サーバ間のインタラクションおよびサーバ / エンジン / ネットワーク接続の失敗によるロールの変化が記録される
- トークン: 分散ロック管理機能を提供する。有効な場合、DSync ノードに関連する読み込み / 書き込みトークンのサーバ間におけるインタラクションが記録される
- タイムアウト: タイムアウト指定ごとの DSync ノードのライフサイクル管理機能を提供する。有効な場合、タイムアウトのために削除されるノードが記録される
- メッセンジャ: iPlanet Application Server サーバ間のメッセージ通信を提供する。有効な場合、作成、送信、受信、および処理されるメッセージが記録される

さらに、iasdsync-cluster-XXX.log および iasdsync-node-XXX.log ファイル (XXX はエンジンのポート番号) にクラスタおよび DSync ノードのデータをダンプできます。

クラスタダンプファイルの書式設定

各 iasdsync-cluster-XXX.log ファイルは次のセクションで構成されています。

- クラスタ

- メッセージキュー

クラスタ情報は、次のように、エンジンが現在の DSync クラスタを表示する方法を報告します。

```
*****
*DSync Cluster State
*****
Host:0xd00c3643
Port: 10818
Role:SyncPrimary
Current Engine's order #:1
SyncPrimary:this engine
Is connect to primary? NO
Changing primary? NO
Max number of SyncBackup#=1
SyncLocal [1]:0xd00c3643:10821
SyncLocal [2]:0xd00c3643:10822
```

メッセージキュー情報は、次のように、DSync キューにあるメッセージリストを表示します。

```
*****
*DSync RecvQueue for GXP_DSSYNC protocol
*****
Message [1]:GXDSYNC_MSG_RECLAIM_RDTOKEN(/dsync41test/K/5) from
0xd00c3643:10818
```

...

DSync ノードダンプファイルの書式設定

各 `iasdsync-node-XXX.log` は次のセクションで構成されています。

- メッセージキュー
- ノードデータ
- タイムアウトマネージャ

メッセージキュー情報は、次のように、DSync キューにあるメッセージリストを表示します。

```
*****
*DSync RecvQueue for GXP_DS SYNC protocol
*****
Message [1] :GXDSYNC_MSG_RECLAIM_RDTOKEN(/dsync41test/K/5) from
0xd00c3643:10818
```

...

ノードデータセクションは、次のように、エンジン内に保存されているノードの集まりを表示します。

```
*****
*DSync Token State
*****
[1] ID:/
    Status:without Read or Write Token
    Scope:GLOBAL
[2] ID:/dsync41test
    Status:without Read or Write Token
    Scope:GLOBAL
    Owner Thread:0xf6f040 (Id=0xf78d50)
    Standard wait queue [1] thread 0xf88670 (Id=0xf883a0)
    Standard wait queue [1] thread 0xf89d60 (Id=0xf89a90)
    Child [0]:B
    Child [1]:A
    Child [2]:D
    Child [3]:C
    Attribute [NextPath]:N
[3] ..
```

タイムアウトマネージャセクションは、次のように、現在のエンジン内の DSync タイムアウトマネージャによって管理されるノードセットを表示します。

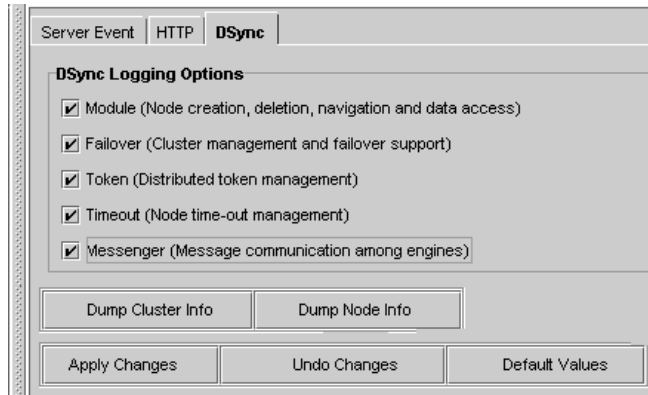
```
*****
*Timeout Manager State
*****
```

```
Entry[0]:ID=/dsync41test/S/4, expired 6 seconds ago
Entry[1]:ID=/dsync41test/U/4, expired 4 seconds ago
Entry[2]:ID=/dsync41test/W/4, expired 3 seconds ago
Entry[3]:ID=/dsync41test/V/4, expired 3 seconds ago
Entry[4]:ID=/dsync41test/X/4, expired 3 seconds ago
Entry[5]:ID=/dsync41test/D/5, expired 2 seconds ago
Entry[6]:ID=/dsync41test/Z/4, expired 2 seconds ago
Entry[7]:ID=/dsync41test/A/5, expired 1 seconds ago
Entry[8]:ID=/dsync41test/B/5, 0 seconds till expiration
Entry[9]:ID=/dsync41test/C/5, 0 seconds till expiration
Entry[10]:ID=/dsync41test/E/5, 0 seconds till expiration
Entry[11]:ID=/dsync41test/F/5, 1 seconds till expiration
Entry[12]:ID=/dsync41test/H/5, 2 seconds till expiration
```

DSync メッセージを記録するには

DSync デバッグメッセージを記録するには、次の手順を実行します。

1. iASAT ツールバーの「ログ」をクリックして「ログ」ウィンドウを開きます。
2. 「ログ」ウィンドウの左側のペインから DSync メッセージを記録する iPlanet Application Server を選択します。
3. 「ログ」ウィンドウの右側のペインで「DSync」タブをクリックします。
次のウィンドウが表示されます。



4. 記録する DSync コンポーネントを指定します。記録する DSync コンポーネントについては、93 ページの「DSync ノードダンプファイルの書式設定」を参照してください。

DSYNC コンポーネントを指定する場合、変更を適用するのに iPlanet Application Server を終了したり再起動したりする必要はありません。

5. (オプション)「ダンプクラスタ情報」をクリックして、DSync 状態のクラスタ情報を `iasdsync-cluster-XXX.log` ファイル (XXX はエンジンのポート番号) にダンプします。

このログファイルの書式設定の詳細は、92 ページの「クラスタダンプファイルの書式設定」を参照してください。

6. (オプション)「ダンプノード情報」をクリックして、DSync 状態のノード情報を `iasdsync-node-XXX.log` ファイル (XXX はエンジンのポート番号) にダンプします。

このログファイルの書式設定の詳細は、93 ページの「DSync ノードダンプファイルの書式設定」を参照してください。

iPlanet Application Server の保護

この章では、iPlanet Application Server セキュリティを実装する方法について説明します。

この章には次のトピックがあります。

- セキュリティについて
- ユーザおよびグループの保存と管理
- アプリケーションコンポーネントへのアクセスに対する認証の設定
- Web サーバと Application Server 間の暗号化の有効化
- ファイアウォールを使ったセキュリティ
- iPlanet Application Server でのファイアウォールの設定

セキュリティについて

アプリケーションセキュリティを実装するには、アプリケーション開発者とサーバ管理者の協力が必要です。アプリケーション開発者は、実装するセキュリティレベルを決定し、そのレベルをアプリケーションに実装する責任があります。サーバ管理者は、アプリケーションを使うユーザおよびグループを管理する責任があります。

管理者には、アプリケーション内のアプリケーションコンポーネントへの認証を管理する責任もあります。J2EE 標準コンポーネントを使う Java アプリケーションでは、ロールを介して認証が実装されます。ロールは、iPlanet Application Server Deployment Tool を使って配置時に作成され、iPlanet Application Server Administration Tool により管理されます。Deployment Tool の詳細は、このツールに付属のオンラインヘルプシステムを参照してください。C++ アプリケーションでは、LDAP に保存されたアクセスコントロールリストを介して認証が実装され、iASAT によって管理されます。

この章では、ユーザおよびグループを設定する方法と、設定されたユーザおよびグループを使ってアプリケーションを保護する方法について説明します。ユーザエントリが iPlanet Directory Server にどのように保存され、iPlanet Console および LDIF を使ってどのように管理されるかについても説明します。この節には次のトピックがあります。

- このマニュアルの内容
- LDAP とは
- iPlanet Console とは

このマニュアルの内容

この章では、iPlanet Console を使ったユーザおよびグループの管理方法に加えて、iPlanet Application Server のインスタンスに関連する Directory Server の設定時に実行が必要な基本的起動タスクについて説明します。iPlanet Directory Server および iPlanet Console の詳細は、各製品のマニュアルを参照してください。

iPlanet Application Server のインスタンスにインストールされた Directory Server のマニュアルは、次の場所にあります。

`iASInstallDir/manual/en/slaped/`

iPlanet Console のマニュアルは、次の iPlanet の Web サイトで入手可能です。

<http://docs.iplanet.com/docs/manuals/console.html>

LDAP とは

iPlanet Application Server の各インスタンスは、Directory Server を使って、ユーザおよびグループについての情報を含む共有サーバ情報を保存します。Directory Server は Lightweight Directory Access Protocol (LDAP) バージョン 2 および 3 をサポートしています。LDAP は TCP/IP 上で動作するオープンディレクトリアクセスプロトコルです。グローバルなサイズおよび多数のエントリに拡張できます。Directory Server を使うと、すべてのアプリケーションサーバがネットワーク経由でアクセスできるディレクトリ情報の単一のセントラルリポジトリに、企業の情報を保存できます。

iPlanet Directory Server は iPlanet Application Server の各インスタンスとともにインストールされます。

iPlanet Application Server に設定可能な Directory Server のタイプは、次のとおりです。

マスタ LDAP サーバ: マスタデータを維持する LDAP サーバ

Consumer LDAP サーバ: マスタ LDAP サーバによって維持されるデータのコピーが保存される LDAP サーバ。iPlanet Application Server に対して複数の Consumer LDAP サーバを設定できる

プライマリ LDAP サーバ: 設定情報について iPlanet Application Server に設定される最初の LDAP サーバ

バックアップ LDAP サーバ: iPlanet Application Server に設定されるセカンダリ LDAP サーバ。プライマリ LDAP サーバが失敗した場合に、iPlanet Application Server はこのサーバに接続する。iPlanet Application Server に対して複数のバックアップ LDAP サーバを設定できる

iPlanet Console とは

iPlanet Console はスタンドアロン Java アプリケーションです。Directory Server に登録されているすべてのリソースおよびアプリケーションを検索して、グラフィカルインターフェイスに表示します。iPlanet Console はすべてのサーバから独立して機能し、企業に接続されているすべてのコンピュータまたはワークステーションから使うことができます。

iPlanet Console は iPlanet Application Server の各インスタンスとともにインストールされます。iPlanet Console を使って iPlanet Application Server のユーザおよびグループを管理します。また、iPlanet Application Server のローカルインスタンス (iPlanet Console と同じマシンにインストールされた iPlanet Application Server のインスタンス) の場合のみ、iPlanet Application Server Administration Tool を起動することもできます。iPlanet Application Server のリモートインスタンスは、コマンドラインまたは Windows NT の「スタート」メニューから起動する必要があります。

ユーザおよびグループの保存と管理

作成する各ユーザおよびグループについて指定する情報は、Directory Server に保存されます。Directory Server に保持された情報は、アプリケーションが複数のサーバによってサポートされているときは、すべてのアプリケーションサーバに共有されます。

この節には次のトピックがあります。

- セキュリティの実装
- iPlanet Console による Directory Server へのエントリの追加
- iPlanet Console を使ったデータベースエントリの変更
- LDIF による Directory Server へのエントリの追加
- プログラムによるエントリの作成

セキュリティの実装

アプリケーションへのアクセスにユーザ名とパスワードの認証が必要な場合、ユーザ名とパスワードは Directory Server に保存されていなければなりません。

アプリケーションは、ユーザ認証を行うアプリケーションコンポーネント (通常は Servlet) を呼び出すことによって、ユーザ認証プロセスを起動します。そのあと、ユーザのログイン権限は、Directory Server に保存されたユーザリストと照合して確認されます。

認証プロセスはユーザ名とパスワードに基づいてアプリケーションへのアクセスを確認します。認証を実装するには、アプリケーションのすべてのユーザに関して、ユーザ名とパスワードを保持するユーザプロファイルを作成する必要があります。この手順については、101 ページの「iPlanet Console による Directory Server へのエントリの追加」で説明します。

ユーザの認証が成功すると、アプリケーションの種類に従って特定のアプリケーションコンポーネント実装にアクセスします。アプリケーションの種類には、J2EE 標準コンポーネントを使う Java アプリケーションと C++ アプリケーションを使う Java アプリケーションがあります。

注 ユーザ名とパスワードの確認以外の認証もあります。たとえば、アプリケーションによっては証明書を介してユーザを認証します。

J2EE アプリケーションの認証

アプリケーションセキュリティを確保するアプリケーションコンポーネントへのアクセスは、配置記述子 XML ファイルに定義された宣言ロール情報に従います。セキュリティは、J2EE によって提供される `isCallerInRole()` のようなセキュリティ API を使って、開発時にプログラムで定義することもできます。詳細については、『iPlanet Application Server 開発者ガイド』を参照してください。

C++ アプリケーションの認証

アプリケーションセキュリティを確保するアプリケーションコンポーネントへのアクセスは、iPlanet Application Server Administration Tool にあるアクセスコントロールリストを使い、宣言によって管理されます。iPlanet Application Server の各インストールに含まれる LDAP JDK を使うことにより、開発時にセキュリティをプログラムで定義することもできます。詳細については、『iPlanet Application Server 開発者ガイド』を参照してください。

iPlanet Console による Directory Server へのエントリの追加

iPlanet Directory Server のユーザエントリおよびグループエントリを作成するには、iPlanet Console を使用します。ユーザエントリには、ディレクトリ内の個人またはオブジェクトについての情報があります。1つのグループは、共通の属性を共有するすべてのユーザで構成されます。たとえば、ある部署のユーザすべてを同じグループに所属させることができます。

識別名 (DN) とは何か

Directory Server では、企業の各ユーザおよび各グループは、識別名 (DN) によって表されます。DN は識別属性を含むテキスト文字列です。ディレクトリ内のユーザおよびグループのデータベースを変更するときには必ず DN を使います。たとえば、ディレクトリエントリを作成または変更するとき、アクセスコントロールを設定するとき、およびメールやパブリッシングなどのアプリケーションに関するユーザアカウントを設定するときは、毎回 DN 情報を指定する必要があります。iPlanet Console のユーザおよびグループのインターフェイスは、DN の作成や変更に役立ちます。

次に iPlanet Communications Corporation の社員の DN の例を示します。

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

この例の各等号 (=) の前の略語には、次の意味があります。

- uid: ユーザ ID

- e: 電子メールアドレス
- cn: ユーザの名前
- o: 組織
- c: 国

DN には、さまざまな名前 - 値ペアが含まれている場合があります。これらは、LDAP をサポートするディレクトリの証明書サブジェクトおよびエントリの両方の識別に使われます。

SSL 認証と識別名

iPlanet Application Server の SSL 相互認証では、X509 SSL クライアント証明書内にある識別名を使用します。アプリケーションサーバは、有効なクライアント証明書が提示されると、そこに含まれるサブジェクト DN を調べて、そのクライアントの認証に使用する候補フィールドを探します。

iPlanet Application Server は、以下のフィールドを一意性の順に探し、選択した証明書の属性の値と一致する uid フィールドを持つ Directory Server エントリを検索します。

ユーザ ID

1. uid
2. oid.2.3.4.45
3. 0.9.2342.19200300.100.1.1

電子メールアドレス

4. emailaddress
5. email
6. e

共通名

7. cn

iPlanet Console を使ってユーザエントリを作成するには

ユーザセキュリティは、アプリケーションのユーザが小人数の場合に最適です。アプリケーションにアクセスする各ユーザに関するユーザプロファイルを作成する必要があります。

ユーザを作成するには、Directory Server 管理者または必要なパーミッションを持つユーザである必要があります。

iPlanet Console を使って iPlanet Directory Server に新しいユーザエントリを作成するには、次の手順を実行します。

1. Windows の場合

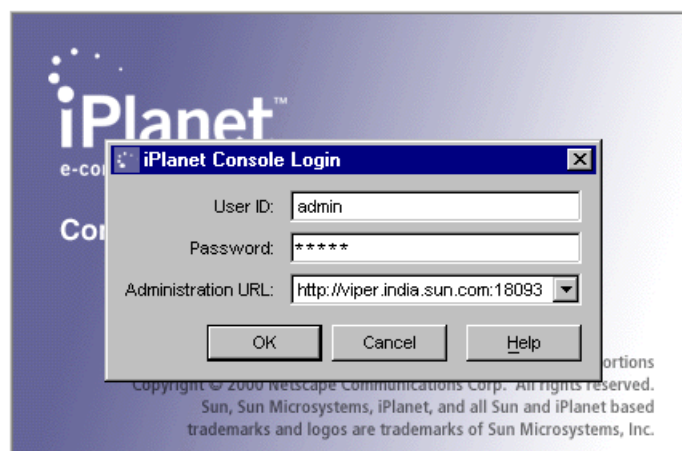
Windows の「スタート」メニューから「プログラム」>「iPlanet Server Products」>「iPlanet Console 5.0」の順にクリックします。

Solaris の場合

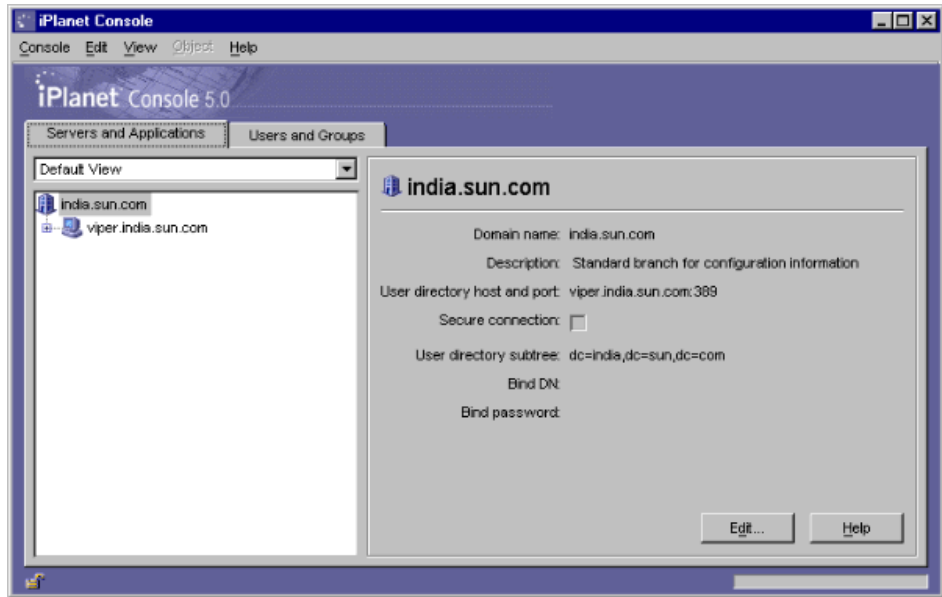
iASInstallDir パスに移動し、次のコマンドを入力します。

```
./startconsole
```

「iPlanet Console」ログインダイアログボックスが表示されます。



2. 「User ID」テキストフィールドに、iPlanet Application Server インストール時に指定したユーザ名を入力します。
3. 「Password」フィールドに、iPlanet Application Server のインストール時に iPlanet Console Administration Server に対して指定したパスワードを入力します。次の図のように、iPlanet Console のメインウィンドウが表示されます。



ユーザおよびグループの作成方法については、iPlanet Console 5.0 に含まれるマニュアルを参照してください。

iPlanet Console を使ったデータベースエントリの変更

ユーザまたはグループデータを変更する前に、まずユーザおよびグループの検索機能を使ってユーザまたはグループエントリをユーザディレクトリで検索する必要があります。次に、メニューツールバーから操作を選択してエントリを変更できます。実行する操作は「Search」リストのすべてのユーザに適用されます。

詳細は、「iPlanet Console」のマニュアルを参照してください。

LDIF による Directory Server へのエントリの追加

Directory Server は、LDAP Data Interchange Format (LDIF) を使って、ディレクトリおよびディレクトリエントリをテキストフォーマットで記述します。LDIF は通常、ディレクトリデータベースを最初にビルドしたり、多数のエントリをまとめてディレクトリに追加したりするときに使います。適切な LDIF 更新ステートメントとともに `ldapmodify` コマンドを使って、エントリを追加または編集することもできます。

LDIF を使ってデータベースにエントリーを追加するには、LDIF ファイルにエントリーを定義したあとに、Directory Server から LDIF ファイルをインポートします。

LDIF エントリーの書式化

LDIF は、空白行によって区切られた 1 つまたは複数のディレクトリエントリーから構成されます。LDIF エントリーは、オプションのエントリー ID、必須の識別名、1 つまたは複数のオブジェクトクラス、および複数の属性定義から構成されます。

LDIF で表されるディレクトリエントリーの基本書式は次のとおりです。

```
dn:distinguished name
objectClass:object class
objectClass:object class
...
attribute type[;subtype]:attribute value
attribute type[;subtype]:attribute value
...
```

DN と 1 つ以上のオブジェクトクラス定義を指定する必要があります。さらに、そのエントリーに関して定義するオブジェクトクラスに必要なすべての属性が含まれている必要があります。他の属性およびオブジェクトクラスはすべてオプションです。オブジェクトクラスおよび属性はどのような順序でも指定できます。コロンのあとの空白もオプションです。標準オブジェクトクラスおよび属性については、次のサイトの iPlanet Directory Server のマニュアルを参照してください。

<http://docs.iplanet.com/docs/manuals/directory.html>

ldapmodify によるデータベースエントリーの変更

既存の Directory Server データベース内のエントリーを変更するには、ldapmodify コマンドラインユーティリティを使います。ldapmodify では、入力した識別名およびパスワードを使って、指定されたサーバへのコネクションを開き、指定されたファイルに含まれている LDIF 更新ステートメントに基づいてエントリーを変更します。

ldapmodify では LDIF 更新ステートメントを使うので、ldapdelete によって実行可能なすべてのことを実行できます。Directory Server のコマンドラインユーティリティのほとんどは 1 か所に保存されています。それらのユーティリティは次のディレクトリにあります。

```
<iASInstallDir>/bin/slapd/server
```

コマンドラインツール ldapdelete、ldapmodify、および ldapsearch は次のディレクトリに保存されています。

```
<iASInstallDir>/shared/bin
```

次の例は、LDIF ファイルにユーザを追加するためのコマンドです。

```
ldapmodify -h myserverhost -p 389 -D "Directory Manager" -w admin -a  
-f MyUsersFile
```

プログラムによるエントリの作成

iPlanet Application Server の各インストールに含まれる LDAP JDK を使うアプリケーション内では、プログラムでエントリを作成することもできます。詳細は、『開発者ガイド (Java)』を参照してください。

アプリケーションコンポーネントへのアクセス に対する認証の設定

アプリケーションコンポーネントへのアクセスに対する認証は、アプリケーションの種類によって異なります。

- Java アプリケーション (J2EE 標準コンポーネントを使う) では、ロールを介して認証が設定される。106 ページの「ロールベースの認証の設定 (J2EE アプリケーションの場合)」を参照してください。
- C++ アプリケーションでは、認証はアクセスコントロールリストのパーミッションによって設定される。109 ページの「アクセス制御リスト認証の設定 (C++ アプリケーションの場合)」を参照してください。

この節には次のトピックがあります。

- ロールベースの認証の設定 (J2EE アプリケーションの場合)
- アクセス制御リスト認証の設定 (C++ アプリケーションの場合)

ロールベースの認証の設定 (J2EE アプリケーションの場合)

アプリケーションコンポーネントのロールが、モジュール内のすべてのアプリケーションコンポーネントにグローバルに設定されます。Administration Tool から、ロールをアプリケーションモジュールに追加して、ロールに属するユーザおよびグループを設定できます。要求者があらかじめ定義されたロールのメンバーである場合は、モジュール内のすべてのアプリケーションコンポーネントにアクセスできます。

ユーザがロールのメンバーでない場合は、アプリケーションがユーザにログインし直すように指示したり、アプリケーションを終了するように要求したり、またはアプリケーションの異なる部分を割り当てたりできます。

EJB および Servlets に関するロールの管理

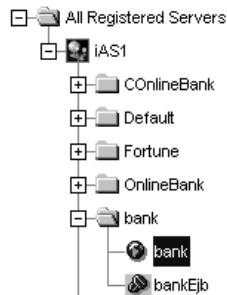
配置されたアプリケーションのロールを管理するには、iPlanet Application Server Administration Tool (iASAT) を使う必要があります。ロールを管理するときに、個々のユーザをメンバーとしてロールに追加しないで、ユーザが属するグループを指定してグループだけをロールに追加できます。これは、個々のユーザベースのセキュリティを使っている場合に有効です。ユーザが変更されたときに更新されたユーザの管理情報をロールに保存できます。

たとえば、Web バンクアプリケーションのユーザを作成して、その後あるユーザがすべてのアカウントを閉じる場合は、グループおよびすべてのロールからユーザを削除するのではなく、該当するグループだけからそのユーザを削除する必要があります。


注 Servlets および EJB のロールは、配置前の配置記述子 XML ファイルに作成されます。詳細は、Deployment Tool に付属されているオンラインヘルプを参照してください。

ロールを管理するには、次の手順を実行します。

1. iASAT ツールバーの「アプリケーション」をクリックして「アプリケーション」ウィンドウを開きます。
2. 「アプリケーション」ウィンドウの左側のペインで、アプリケーションが配置された iPlanet Application Server インスタンスを展開します。
3. 次の図のように、アプリケーションフォルダを開き、Servlet または EJB アイコンを強調表示します。



- 「アプリケーション」ウィンドウの右側のペインで「ロール」タブをクリックして、この EJB/Servlet に対してすでに定義されているロールとロールメンバーを調べます。



Role(s)	Role Member(s)
BankUser	PD Managers (group), HR Managers (group), Accounting Managers (...)
Bank.Admin	HR Managers (group), Accounting Managers (group)

- 管理するロールを強調表示し、ウィンドウの下にある「ロールを編集」ボタンをクリックします。

「ロールを編集」ダイアログボックスが開き、現在このロールのメンバーであるすべてのユーザおよびグループが表示されます。



- ロールにグループおよびユーザを追加するには、次の手順を実行します。
 - グループをロールに追加するには、「利用可能なグループ」ボックス内で、1つまたは複数のグループを強調表示し、右矢印ボタンをクリックします。

注 「利用可能なグループ」ボックスから複数のグループを選択すると、「グループ内のユーザ」ボックス内のユーザは表示されません。

- b. ユーザをロールに追加するには、まず、「利用可能なグループ」リストからユーザが現在属するグループを強調表示し、次に「グループ内のユーザ」ボックスのユーザを強調表示します。最後に右矢印ボタンをクリックし、ユーザをロールに追加します。
7. ロールからグループまたはユーザを削除するには、「ロール内のユーザ / グループ」ボックス内のユーザまたはグループを強調表示し、左矢印ボタンをクリックします。

アクセス制御リスト認証の設定 (C++ アプリケーションの場合)

アクセス制御リスト (ACL) を使うと、ユーザおよびグループにパーミッションを設定できます。パーミッションは、読み込みや書き込みなど、ユーザが実行を許可されるアクションに関連付けられています。

iPlanet Application Server には、デフォルトのパーミッションがありますが、ユーザ独自のアプリケーションに固有のパーミッションおよび ACL を作成することもできます。ACL の情報は、ユーザが行うアクションに対して現在のユーザまたはグループのパーミッションを確認するために、アプリケーションによって使われます。

ユーザがあるパーミッションを持っていない場合は、アプリケーションがユーザにログインし直すように指示したり、アプリケーションを終了するように求めたり、あるいはアプリケーションの異なる部分を割り当てたりできます。

アクセス制御リストの作成

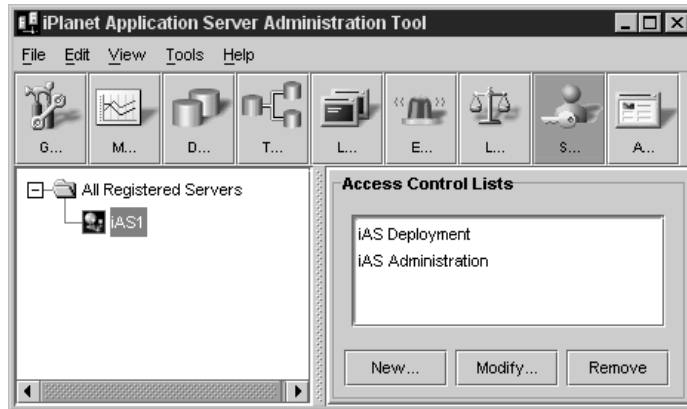
アクセス制御リスト (ACL) を作成して管理するには、iASAT を使う必要があります。ACL を作成するときに、個々のユーザを ACL にメンバーとして追加せずに、ユーザが属するグループを作成してグループだけを ACL に追加することができます。これは、個々のユーザベースのセキュリティを使っている場合に有効です。ユーザが変更されたときに更新されたユーザの管理情報を ACL に保存できます。

たとえば、イントラネットアプリケーションのユーザを作成して、その後あるユーザが退職する場合は、グループおよびすべての ACL からユーザを削除するのではなく、該当するグループだけからそのユーザを削除する必要があります。

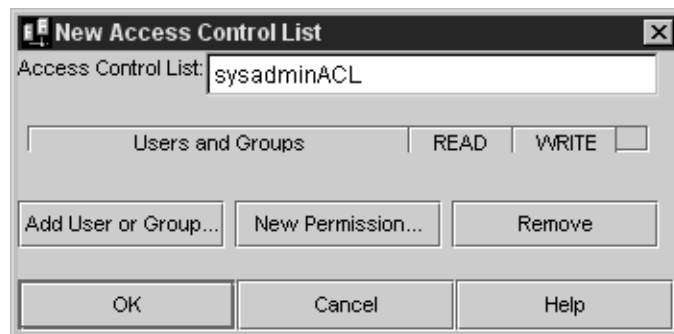
アクセス制御リストを作成するには、次の手順を実行します。

1. iASAT ツールバーの「セキュリティ」をクリックして「セキュリティ」ウィンドウを開きます。

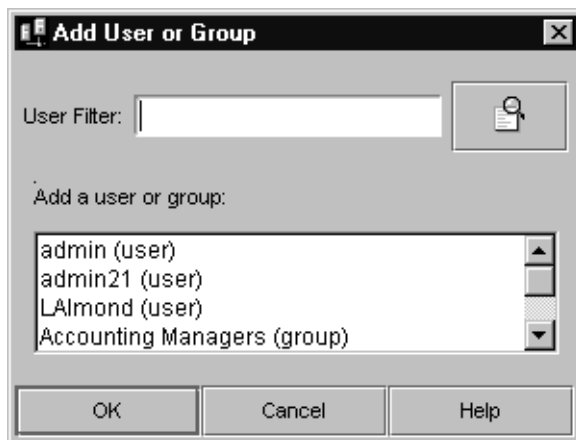
次のウィンドウが表示されます。



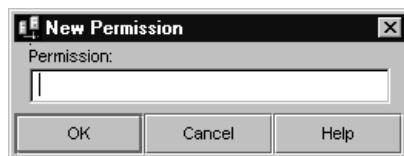
2. 「新規」をクリックします。次の図のように、「新規アクセス制御リスト」ダイアログボックスが表示されます。



3. 「アクセス制御リスト」テキストフィールドに ACL の名前を入力します。
他の ACL と区別できる名前を付けます。
4. ACL にユーザまたはグループを追加するには、「ユーザまたはグループの追加」ボタンをクリックします。
「ユーザまたはグループの追加」ダイアログボックスが表示されます。



5. ACL に追加するユーザまたはグループ、あるいはその両方を選択します。
「ユーザフィルタ」テキストボックスに文字列を入力することによって、結果セットに表示されるユーザのリストをフィルタできます。たとえば、「F」で始まるユーザ ID だけを表示するには、「ユーザフィルタ」テキストボックスに「F*」と入力し、そのテキストボックスの右側にある「ユーザフィルタ」ボタンをクリックします。フィルタ条件と一致するユーザ ID が下のリストボックスに表示されます。ユーザフィルタはグループではなくユーザだけに適用されます。
6. 「OK」をクリックします。
7. ACL に新しいパーミッションを追加するには、「新規パーミッション」をクリックします。
「新規パーミッション」ダイアログボックスが表示されます。



8. 新しいパーミッションを入力します。
パーミッションは、ユーザまたはグループが特定のアプリケーションまたはアプリケーションの一部に対して持っているアクセスのレベルを定義します。
9. 「OK」をクリックします。
10. ACL にグループの適切なパーミッションを設定するには、そのグループの各パーミッションを確認します。

アクセス制御リストの変更

次の ACL プロパティを変更することができます。

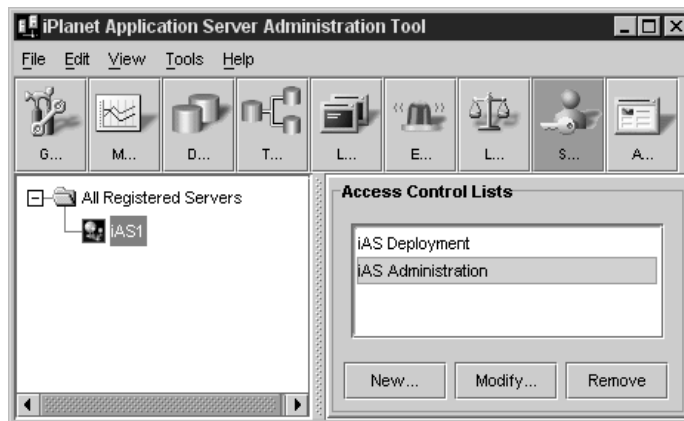
- グループの追加
- 新規パーミッションの作成
- パーMISSIONの編集

システムからグループを削除することもできます。

アクセス制御リストを変更するには、次の手順を実行します。

1. iASAT ツールバーの「セキュリティ」をクリックして「セキュリティ」ウィンドウを開きます。

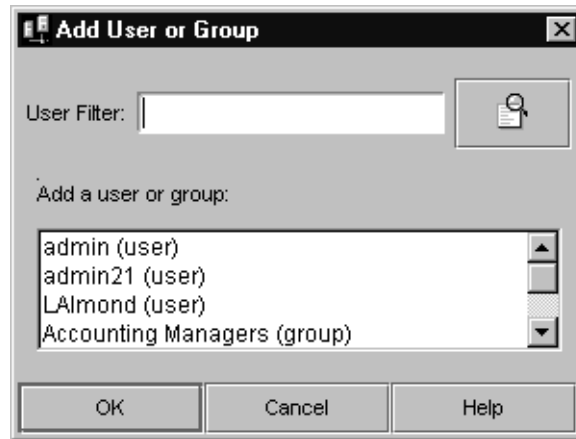
次のウィンドウが表示されます。



2. 変更するアクセス制御リストを強調表示します。
3. 「変更」をクリックします。「アクセス制御リストの変更」ダイアログボックスが表示されます。



- 新しいユーザまたはグループを追加するには、「ユーザまたはグループの追加」ボタンをクリックします。「ユーザまたはグループの追加」ダイアログボックスが表示されます。

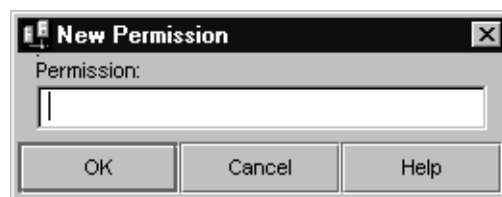


- ACL に追加する 1 つまたは複数のグループを選択します。

「ユーザフィルタ」テキストボックスに文字列を入力することによって、リストに表示されるユーザをフィルタできます。たとえば、「F」で始まるユーザ ID だけを表示するには、「ユーザフィルタ」テキストボックスに「F*」と入力し、そのテキストボックスの右側にある「ユーザフィルタ」ボタンをクリックします。フィルタ条件と一致するユーザ ID が下のリストボックスに表示されます。ユーザフィルタはグループではなくユーザだけに適用されます。

- 「OK」をクリックします。
- 「アクセス制御リストの変更」ウィンドウで「新規パーミッション」をクリックして新しいパーミッションを作成します。

「新規パーミッション」ダイアログボックスが表示されます。



- 「パーミッション」テキストフィールドに Read や Write などの新しいパーミッションを入力し、「OK」をクリックします。

9. 「アクセス制御リストの変更」ウィンドウで、パーミッションの隣りにあるチェックボックスのオン / オフを切り替えることによって、グループのパーミッションを編集できます。
10. グループを削除するには、グループを選択し、「アクセス制御リストの変更」ウィンドウで「削除」をクリックします。

Web サーバと Application Server 間の暗号化の有効化

Web サーバと Application Server 間のトラフィックを暗号化するかどうかを選択できます。Executive Server (KXS) ではサーバ間のトラフィックを管理します。クレジットカード情報を収集する Servlets やログイン Servlets など、高度なセキュリティが必要なコンポーネントの暗号化を有効にしなければならない場合があります。コンポーネントは、128 ビットキーおよび RSA BSafe 3.0 ライブラリを使って暗号化されるので、情報が安全に転送されます。

この節では、KXS ログで暗号化の詳細を確認する方法と、暗号化の cryptext キーの作成方法について説明します。

この節には次のトピックがあります。

- Web サーバと Application Server 間の暗号化を有効にするには
- 暗号化ログメッセージを確認するには
- 個々の J2EE コンポーネントの暗号化を有効にするには

Web サーバと Application Server 間の暗号化を有効にするには

Web サーバと Application Server 間のトラフィックの暗号化を有効にするには、次の手順を実行します。

1. iPlanet レジストリエディタを開きます。
(32 ページの「iPlanet レジストリエディタについて」を参照)。
2. 次のキーを開きます。
SOFTWARE/iPlanet/Application Server/6.5/CCSO/Security/

3. `EnableEncryption=0` のような値が表示されます。デフォルトでは、暗号化は無効になっています。暗号化を有効にするには、この値を `EnableEncryption=D` に変更します。D は内部 128 ビットデータタイプ文字列を意味します。これで暗号化が有効になりました。

暗号化ログメッセージを確認するには

KXS ログファイルで暗号化ログメッセージを確認するには、次の手順を実行します。

1. iPlanet レジストリエディタを開きます。
(32 ページの「iPlanet レジストリエディタについて」を参照)。
2. 次のキーを開きます。
`SOFTWARE/iPlanet/Application Server/6.5/CCSO/Security/`
3. 「編集」メニューから「値を追加」を選択します。次の図のように、「値を追加」ダイアログボックスが表示されます。



4. 「名前」フィールドに「LogEncryption」と入力します。
5. 「値」フィールドに「1」と入力します。
6. 「入力」ドロップダウンリストから「Integer」を選択します。

このキーを作成すると、暗号化ログメッセージが KXS ログに記録されます。

個々の J2EE コンポーネントの暗号化を有効にするには

JSP や Servlets などの個々の J2EE コンポーネントの暗号化を有効にするには、次の手順を実行します。EJB については暗号化を有効にできません。

1. コンポーネントの暗号化を有効にするアプリケーションの `ias-web.xml` ファイルを開きます。

2. 暗号化を有効にする Servlet または JSP を検索して選択します。
3. コンポーネントの `<encrypt>>false</encrypt>` タグを選択します。 *false* を *true* に変更します。
4. `iasdeploy` を使ってアプリケーションを登録します。

暗号化が有効になっていて機能していることを確認するには、KXS ログを開いて、次のようなメッセージを検索します。

```
[11/Jan/2001 19:58:43:0] info:CRPT-001:Encrypting 2309 bytes,  
keysize = 128 bits
```

```
[11/Jan/2001 19:58:43:5] info:NSAPICLI-012:plugin reqstart, tickct:  
1903570535
```

```
[11/Jan/2001 19:58/:43:5] info:NSAPICLI-009:plugin  
reqexit:Os+.12995s. (198114 0537)
```

```
[11/Jan/2001 19:58:52:2] info:CRYPT-004:Decrypting 1897 bytes,  
keysize = 128 bits
```

コンポーネントの暗号化を有効にすると、iPlanet レジストリで暗号化を無効に設定しても暗号化が行われます。

ファイアウォールを使ったセキュリティ

iPlanet Application Server は通常、Web サーバおよびクライアントデータベースアプリケーションとともに配置されます。このような場合、プロセス間通信は非常にレベルが高いため、データの完全性およびセキュリティの問題が非常に重要になります。アプリケーションサーバ間、Web サーバ間、およびアプリケーションサーバで実行するクライアントアプリケーション間の円滑で安全な通信を確保するために、システムのセキュリティコンポーネントがファイアウォールによって区切られる場合があります。

この節では、ファイアウォールで保護された環境で iPlanet Application Server を効果的に計画して配置する方法について説明します。この節には次のトピックがあります。

- ネットワークセキュリティの基本
- iPlanet Application Server アーキテクチャ
- プロセス間通信プロトコル
- データチャネルの暗号化

注 このマニュアルに記載されているアドレスおよびポート番号は、すべてデフォルト値です。これらの値は、対応するレジストリに保存されている値の変更によって、インストール時、またはその後に変更される場合があります。変更された場合は、対応する変更をファイアウォール設定パラメータに適用する必要があります。

ネットワークセキュリティの基本

企業のローカルエリアネットワーク (LAN) などの専用ネットワークは、インターネットなどの公衆網に直接接続されるのではなく、アクセスを制御するファイアウォールを通過します。公衆網へのアクセスは、特定のマシンの承認されたポート番号で承認されたプロトコルを使って制御されます。

ファイアウォールの種類

ファイアウォールには種類があり、それぞれに長所と短所があります。ファイアウォールは、1つの種類のマシンで構成されるとはかぎりません。通常、存在している他のマシンを必要とします。これらの他のシステムは、基本的なトラフィック転送を行うように設定されることはありますが、ファイアウォール機能を提供するように設定されている場合とそうでない場合があります。次の節では、3つの主要なファイアウォールについて説明します。

ルータ

ルータは高速かつ簡単なファイアウォールであり、比較的low価格です。基本的に、ルータは2つのネットワークの間に置かれ、それらのネットワーク間の接続を中継します。設定パラメータは、個々の IP アドレスまたはある範囲の IP アドレスとの接続を許可または拒否するように指定できます。

より高度なバージョンのルータでは、データを検査して、使われているセキュリティプロトコル、およびそのセキュリティプロトコルがアドレス指定されているポート番号を調べます。制御は特定のプロトコルを使う接続で実行され、特定のポート (フィルタ) に転送されます。

ルータはそのタスク専用に構築されたマシンであり、通常 UNIX オペレーティングシステムが動作する、複数のネットワークインターフェイスを持つコンピュータを使って実装できます。汎用マシンを専用ルータとして使う長所は、ルータがプロキシサーバも実行できることです。

プロキシサーバ

プロキシサーバは通常、データアクセスがプロキシサーバだけを通してのように、何らかの形のルータと一緒に使われます。プロキシサーバは、ファイアウォールにまたがるマシンに置かれるか、またはプロキシサーバと専用ネットワークの間に他のルータを持つ、非武装地帯 (DMZ) と呼ばれる小さな専用ネットワーク上でプライマリルータの後ろに置かれます。後者は一般的な配置です。一般的に、プロキシシステムは余分な装備をすべて除いた基本的なシステムです。標準のユーザアプリケーションがほとんどなく、さまざまなプロトコルのプロキシサーバとして機能するプログラムだけがあります。

プロキシサーバは、専用ネットワーク内で一部のクライアントのサーバとして機能するように設計されています。たとえば、HTTP プロキシサーバは、専用ネットワーク上の Web ブラウザに関するかぎり HTTP サーバとして動作します。ブラウザは HTTP プロキシサーバに接続し、ドキュメントのリクエストを送信します。その結果、HTTP プロキシサーバはブラウザとして機能し、ドキュメントが要求されたサーバに接続します。リモート HTTP サーバはプロキシサーバを認識しません。HTTP プロキシサーバは要求されたドキュメントを受信すると、専用ネットワーク上のユーザのブラウザに渡す前に、ウイルスがないかどうかドキュメントをスキャンします。HTTP プロキシサーバがドキュメントをブラウザに渡す前に実行するアクションは、HTTP プロキシサーバで指定できます。これによって追加のセキュリティが有効になり、情報の質および安全性も保証されます。

ステートフルインスペクションファイアウォール

一般的になってきたファイアウォールの形態は、単に情報を転送するのではなく情報パッケージのコンテンツも検査するマシンです。取得した情報を使って、より高いレベルのプロトコルを決定できます。これによって、アプリケーションプロトコルのレベルまで拡張可能です。通信プロトコルが確立されると、プロトコルが新しいステートを入力するまで、システムではステートテーブルを使って、どの操作を許可するかを決定します。

たとえば、受信した telnet コネクションが送信先マシンのポート番号 23 に転送される TCP/IP (Transmission Control Protocol/Internet Protocol) コネクションであると仮定します。telnet サーバは、telnet クライアントによって接続されるポート番号を使って応答します。ステートフルインスペクションファイアウォールシステムは、ステートが開いている telnet を認識し、返されたデータに、クライアントによって接続されるポート番号があるかどうかを調べます。ホスト (コネクションを開始したマシン) がコネクションを送信先ポートに転送すると、ステートフルインスペクションファイアウォールシステムによってコネクションが許可されます。標準以外のポートへの他のランダムなコネクションは通常拒否されます。

ステートフルインスペクションファイアウォールは、すべてのデータを調べる必要がないため高速です。このファイアウォールは、指定されたプロトコルを処理するために特別に作成する必要のあるプロキシサーバと違って、汎用的なセキュリティに使うことができます。新しいプロトコルのプロキシの作成には、長時間かかる可能性があります。ステートフルインスペクションファイアウォールシステムを使うと、適切なステートトランザクションおよびアクションを決定する新しいプロトコルを監視するだけで、新しいステートテーブルをすばやく簡単に作成できます。

次の節では、iPlanet Application Server のアーキテクチャと iPlanet Application Server 内の通信方法について説明します。

iPlanet Application Server アーキテクチャ

iPlanet Application Server は複数のモジュールから構成されます。基本コンポーネントは Executive Server (KXS) です。Executive Server はアプリケーションのコンポーネントを作成し、負荷の監視および均衡を iPlanet Application Server の他のインスタンスとともにセッションデータごとに管理します。

アプリケーションコードは KXS によって作成されるマルチスレッドプロセスで動作します。プロセスには、C++ Server (KCS) と Java Server (KJS) の 2 つの種類があります。

システムは Administration Server (KAS) によって管理されます。Web サーバは iPlanet Application Server と同じマシンにインストールされることもありますが、通常は、別のマシンにインストールされます。

Web サーバと iPlanet Application Server 間の通信には、iPlanet Application Server と直接通信する Web サーバにあるプラグインを使います。iPlanet Application Server の複数のインスタンスがインストールされている場合、プラグインではロードバランスシステムによって選択された iPlanet Application Server と通信します。使用している Web サーバがプラグインを使うことができない場合、通信モジュールは CGI (Common Gateway Interface) アプリケーションとして呼びだされることがあります。これは iPlanet Application Server とのコネクションを確立します。CGI モデルの使用は効率的でないので、通常は最後の手段として使われます。

また、OCL (Object Constraint Language) を使う方法もあります。OCL は CORBA を使って必要なサービスを検索し、iPlanet Application Server を介してそれらのサービスと通信します。これは、IIOP (Internet Inter-Object Protocol) コネクションについて標準化されたセキュリティがないため、イントラネットだけにお勧めします。

次の節では、ファイアウォールの構成方法と、サーバ、アプリケーション、およびセキュリティプロトコル間の通信方法について説明します。

プロセス間通信プロトコル

サーバ、アプリケーション、およびセキュリティプロトコルの間のプロセス間通信は、さまざまな方法で行われます。関連する通信リンクおよびプロトコルは次のとおりです。

エフェメラルポート

エフェメラルポートは、クライアントからサーバへのリクエストに応じて作成される匿名ポートです。UDP (User Datagram Protocol) の場合、これらは返信データグラムに使われるポート番号になります。TCP/IP の場合は、クライアントからのリクエストに応じてサーバで作成される接続のポート番号になります。

エフェメラルポートは現在使われておらず、予約された範囲 (0-1023) 外のポートにすることができます。通常、エフェメラルポートは、範囲 (1024-5000) から割り当てています。ただし、一部のシステム、特に Solaris では、別の範囲にあるエフェメラルポート番号を割り当てます。多くの接続を持つ大規模なシステムでは、ポート番号の制限された範囲が不十分な場合があるという認識から、通常、エフェメラルポートとして 32768 より大きなポート番号を割り当てることができます。

エフェメラルポートを使用できるファイアウォールではエフェメラルポートについて、使われている値を確認することと、適切な範囲が確実に有効になっていることが重要です。この範囲は通常、現在の Solaris システムでは >32768、Windows システムでは 1023 ~ 5000 です。

Transmission Control Protocol/Internet Protocol (TCP/IP)

次の表に示すように、さまざまな種類のトラフィックを送る複数の TCP/IP コネクションがあります。表に示されているアドレスおよびポートはデフォルト値です。

コネクション	トラフィック	ポート / アドレス	目的
Web コネクタプラグインから KXS	リクエストデータ	10818	ユーザ I/O
KXS から KCS/KJS	内部データ	10819 (*)	リクエスト送信
CGI から KXS	リクエストデータ	テキスト参照	ユーザ I/O
KCS/KJS から KXS	内部データ	10818	結果 / 管理
管理 UI から KAS	内部データ	10817	管理者

注 (*) この番号で始まり、各 KCS/KJS によって増加します。

これらの通信チャネルの大部分は複数の種類のデータ転送に使われます。データ転送は同じ TCP/IP コネクション上で多重化されます。

プラグインをサポートしていない Web サーバが使われている場合、KXS との通信は小さな CGI アプリケーションを通過します。

CGI と KXS の間の通信に使われるポートは固定されていませんが、次のように割り当てられます。

10819 + (KCS/JKS エンジンの数)

したがって、4つの KJS エンジンと1つの KCS エンジンがインストールされている場合、CGI ポートはポート番号 10824 に設定されます。これは、iPlanet Application Server インストール時のデフォルトの CGI ポート番号 10819+3 です。

Web サーバの TCP/IP 通信は、HTTP または HTTPS (Secure HTTP) のどちらを使うかによって、ポート 80 または 443 に割り当てられます。

IP マルチキャスト

複数の iPlanet Application Server インスタンスがある場合、iPlanet Application Server では、2つの IP マルチキャストチャネルを使って次の通信タスクを処理します。

- IP マルチキャストチャネル1 では KXS プロセス間でロードバランス情報を通信する
- IP マルチキャストチャネル2 では KAS プロセス間で管理情報およびコマンドを通信する

KXS チャネルのトラフィックは重くなる可能性があり、システム負荷によって増加します。次の表は KXS チャネルのデフォルトのアドレスを示しています。

マルチキャスト	アドレス	Port
KXS <-> KXS	228.8.18.71	9607
KAS <-> KAS	228.8.18.71	9608

注 1つの TCP/IP チャネルでの内部サーバデータの多重化は、Kiva Communications Protocol (KCP) と呼ばれる場合があります。このプロトコルは内部データ転送だけを意味します。

User Datagram Protocol (UDP)

Web コネクタプラグインでは UDP を使います。ping プロトコルを使って、Web コネクタプラグインが現在関連付けられている KXS プロセスがまだ接続されているかどうかを確認します。KXS 内で実行中のスレッドが ping に応答します。ping に対するレスポンスは KXS が実行中であることを示します。レスポンスがない場合は、代替の KXS に切り替えるためにフェールオーバーを開始することを示します。次の表は、デフォルトの UDP ポートの指定を示しています。

UDP パケット	Port
ping リクエスト	9610
ping レスポンス	エフェメラル

UDP が許容されない場合は、Web コネクタプラグインの iPlanet レジストリを変更すると ping を停止できます。ping リクエストへのレスポンスを無効にするには、次の手順を実行します。

1. iPlanet レジストリを開始します。
(32 ページの「iPlanet レジストリエディタについて」を参照)。
2. 次のキーを開きます。
SOFTWARE/iPlanet/Application Server/6.5/CCS0/CONN/
3. DisableEcho キーを変更します。値を 1 に設定します。

ヒント iPlanet レジストリで値を変更するには、次の手順を実行します。

- a. キー名をダブルクリックするか、あるいはキーを選択し、「編集」メニューから「値を変更」を選択すると、「値を変更」ダイアログボックスが表示されます。
 - b. ダイアログボックスに新しい値を入力します。
 - c. 「OK」をクリックして iPlanet レジストリに変更を登録します。
-

UDP の ping をオフにすると、現在選択されている iPlanet Application Server が失敗して、TCP/IP コネクションタイムアウトまで遅延が発生する可能性があります。TCP/IP 設定パラメータによって異なりますが、これには数分かかる場合があります。

ネットワークで UDP が無効になっている場合は、動作する Web コネクタプラグインの Web コネクタプラグインレジストリで ping を無効にする必要があります。UDP ping は、CGI インターフェイスではなく Web サーバプラグインだけで使われます。

次の表に、iPlanet Application Server コンポーネントのプロトコル、アドレス、および目的をまとめます。

コンポーネント	プロトコル	ポート/アドレス	目的
Web コネクタプラグイン	TCP/IP	10818	Web コネクタプラグインから KXS
Web コネクタプラグイン/CGI	TCP/IP	> 1024	コネクションの返送
Web コネクタプラグイン	UDP/IP	9610	キーブアライブ ping
Web コネクタプラグイン	UDP/IP	エフェメラル	ping 返送
KXS から KCS/KJS	TCP/IP	10819	KXS から KCS/KJS
CGI から KXS	TCP/IP	最後の KCS/KJS + 1	CGI アプリケーションから KXS
KCS/KJS から KXS	TCP/IP	10818	KCS/KJS から KXS
KXS	IP マルチキャスト	228.8.28.71:9607	KXS から KXS
KAS	IP マルチキャスト	228.8.18.71:9608	KAS から KAS
管理 UI から KAS	TCP/IP	10817	管理者

データチャネルの暗号化

Web コネクタプラグインと KXS 間で実行される通信を暗号化できます。暗号化を行うときは、同じポートとアドレスが継続して使われますが、データ転送が暗号化されます。安全な暗号を使うデータ暗号化に関連するオーバーヘッドのため、絶対的なニーズに基づいて使うことをお勧めします。「Web サーバと Application Server 間の暗号化の有効化」を参照してください。

iPlanet Application Server でのファイアウォールの設定

この節では、いくつかの一般的なファイアウォール設定と、正しく機能するためのパラメータ設定について検討します。これらの例では、iPlanet Application Server 実装内へのファイアウォールの合理的な配置を示します。

iPlanet Application Server インスタンス (KXS/KJS/KCS プロセスグループである iPlanet Application Server インスタンス) 間にファイアウォールを置くことは可能ですが、お勧めしません。そのようなファイアウォールを実装する場合は、ファイアウォールを越えて IP を許可する IGMP (Internet Group Management Protocol) を実装することが重要です。

iPlanet Application Server インスタンス間にファイアウォールを置くことはお勧めしませんが、環境によっては iPlanet Application Server マシン間に IP マルチキャストトラフィックの高速専用ネットワークを実装することをお勧めします。

iPlanet Application Server システムは LAN 速度で動作するネットワーク用に設計されているので、iPlanet Application Server インスタンスを WAN (広域ネットワーク) 全体に分散させると、パフォーマンスの問題が発生する場合があります。

単独のファイアウォール

最も簡単で一般的なファイアウォール構成は、Web サーバとインターネットの間にファイアウォールを 1 つ置くものです。

ファイアウォールが 1 つの設定は単純です。適切にポート 80 や HTTPS 443 への HTTP コネクションを許可するようにファイアウォールを設定する必要があります。返されるデータが 1024 よりも大きいポートにあるため、これらのポートを送信 (返信) コネクションに開く必要があります。

この設定は簡単であることが利点です。最大の短所は防御が 1 つしかないことです。ファイアウォールが破られると、専用ネットワーク内にある各マシンのセキュリティが唯一の防御になります。

次の表は、ファイアウォールを正しく機能させるためのプロトコルとポートの設定を要約したものです。

プロトコル	方向	Port	理由
TCP/IP	受信	80	HTTP リクエスト
TCP/IP	受信	443	HTTPS リクエスト
TCP/IP	返信 (送信)	>1024	HTTP(S) レスポンス

Web サーバをファイアウォールの外側で実行しなければならない場合があります。

この設定によって Web サーバマシンはインターネットに公開されるため、お勧めできません。前の例のように、防御が1つしかありません。外部 Web サーバは通常、保護されたネットワーク内での安全性しかないので、有害な攻撃の第一の標的となります。

このファイアウォール設定では、次の表のように、Web コネクタプラグイン TCP/IP データコネクションおよび UDP ping コネクションを許可する必要があります。

プロトコル	方向	Port	理由
TCP/IP	受信	10818	Web から KXS
TCP/IP	受信	次の「注」を参照	CGI から KXS
TCP/IP	返信 (送信)	> 1024	KXS から Web
UDP	受信	9610	Web から KXS
UDP	返信 (送信)	エフェメラル	KXS から Web

注 使用している Web サーバで Web コネクタプラグインを使用できない場合は、KXS との通信に CGI アプリケーションが使われます。インストール時にデフォルトで割り当てられた CGI ポートは、10819 + (KCS/JKS インスタンスの数) です。

二重のファイアウォール - DMZ 設定

この設定は、多くの企業が DMZ を介して提携先や顧客に専用ネットワークへの制限されたアクセスを解放するにつれて一般的になってきています。

この設定で提供されるセキュリティは、前に示した単独のファイアウォールの例よりもはるかに改善されています。各ファイアウォールおよび DMZ でアクティビティをアクティブに監視する二重レイヤの保護によって、内部ネットワークへの侵入のための試みがほとんど検出されます。

この場合、外側のファイアウォールは、上の例のように HTTP および HTTPS トランザクションを許可するように設定する必要があります。内側のファイアウォールは、Web サーバプラグインがファイアウォールの後ろの iPlanet Application Server サーバと通信できるように設定する必要があります。

KXS インスタンスと通信する Web プラグインで使われるデフォルトのポートは 10818 です。同様に、フェールオーバーのキープアライブ ping に使われるデフォルトの UDP ポートは 9610 です。これを変更するには、ファイアウォール設定の適切な変更が必要です。次の表は、KXS ポートと UDP ポートのデフォルトのポート番号を示します。

プロトコル	方向	Port	理由
TCP/IP	受信	10818	Web から KXS
TCP/IP	返信 (送信)	> 1024	KXS から Web
UDP	受信	9610	フェールオーバー ping
UDP	送信	エフェメラル	フェールオーバー ping 返信
TCP/IP	受信	389	Web から LDAP

三重のファイアウォール - データベース保護を持つ DMZ

企業の設定には、データベースが自社のネットワーク内にあるものもあります。この設定では、もっとも重要な会社のリソース、つまり企業データベース内に保存されているデータを重点的に保護します。

LAN とデータベースシステム間のファイアウォールでは、外部だけでなく内部の脅威からも保護します。

データベースへのコネクションでは、ODBC (Open DataBase Connectivity)、JDBC (Java DataBase Connectivity)、データベースベンダーから供給されたコネクタライブラリなどの標準的なアクセスメカニズムを使います。データベースへのコネクションは他のアプリケーションと同じなので、データベース保護レイヤのファイアウォール設定は、使用している特定のデータベースへのアクセスに必要な標準設定に従います。

注**Web サーバへの FTP アクセス**

アプリケーションの配置時に、通常、いくつかの HTML ページを Web サーバドキュメント階層にインストールする必要があります。これらのページのインストールは手動で行い、通常、FTP (File Transfer Protocol) を介して Web サーバにページを転送します。iPlanet Application Server とその Web サーバ間にファイアウォールを設定するときは、iPlanet Application Server から Web サーバへの FTP (またはファイルをコピーする他の手段) を注意して許可する必要があります。これはアプリケーション配置時だけ許可する必要があり、その他の場合は無効にできます。

リモート管理

iPlanet Application Server Administration Tool をリモートで実行するために、ネットワーク接続は、iASAT と KAS 間の接続についてポート 10817 で TCP/IP を許可する必要があります。

サーバリソースの可用性の向上

この章では、効果的なシステムパフォーマンスを確保するために、iPlanet Application Server リソースの可用性を向上させる方法について説明します。

スレッド数、プロセス数、再起動試行回数などの iPlanet Application Server のリソースを増やすと、サーバ上で実行するアプリケーションのパフォーマンスが向上し、アプリケーションのダウンタイムの発生頻度が減少します。

サーバリソースを増やす際は、iPlanet Application Server インスタンスのリソースを考慮する必要があります。たとえば、iPlanet Application Server インスタンスが最大能力で作業している場合に負荷を増やすと、アプリケーションのパフォーマンスに悪影響を及ぼすことがあります。同様に、プロセスに追加のスレッドを割り当てると、システム全体のスレッドプールから使用可能なスレッドが削除され、データベースアクセスなどのスレッドを利用したほかのリクエストを処理するためのシステムの能力が制限されます。

Directory Server には、iPlanet Application Server の使用に必要な大量の情報が格納されています。このため、何らかの理由で Directory Server がエラーになってもこの情報が失われないようにする必要があります。大量の情報および設定データの高可用性を実現するため、プライマリ Directory Server がエラーになった場合にバックアップ Directory Server に引き継ぐように設定する必要があります。

この章では主に、iPlanet Application Server のリソースを効果的に構成し、データの可用性を向上させる方法について説明します。

この章には次のトピックがあります。

- サーバプロセスの追加と調整について
- リクエスト処理のスレッド数の調整
- 実行プロセスのリクエスト数の指定
- Administration Server オプションの設定
- マルチプロセスのシングルスレッド環境の実現
- Directory Server のフェールオーバーの設定

サーバプロセスの追加と調整について

Java サーバ (KJS) あるいは C++ サーバ (KCS) のプロセスを追加することによって、可用性を向上させることができます。1 つまたは 2 つのプロセスを追加すれば、アプリケーションはユーザのリクエストにより確実に応答するようになります。たとえば、1 つのプロセスがエラーになっても 2 番目、3 番目のプロセスが引き続き実行されるので、アプリケーションを使用できない時間が減ります。この方法は、プロセスエラーの原因となる問題が判明しているアプリケーションで特に効果的です。

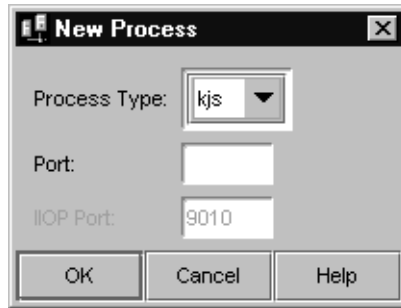
さらに、ブリッジプロセスを追加すると、RMI/IIOP を使用することにより、iPlanet Application Server 上の KJS プロセスでホストされるアプリケーションコンポーネントと直接通信できます。CORBA ベースクライアントからのリクエストは、ブリッジプロセスを経由して iPlanet Application Server に送られます。これにより、CORBA ベースクライアントは iPlanet Application Server 上のアプリケーションコンポーネントと直接通信できます。ブリッジプロセスの追加の詳細は、第 10 章「CORBA ベースクライアントサポートの有効化」を参照してください。

Java および C++ サーバプロセスの追加と調整

Java アプリケーションには KJS プロセスを、C++ アプリケーションには KCS プロセスを追加できます。どちらのタイプのアプリケーションにも、3 つ以上のプロセスを追加する必要はありません。1 つまたは 2 つのプロセス上でアプリケーションを実行できない場合は、コード内にエラーがある可能性があります。アプリケーション開発者に確認してください。

Java サーバまたは C++ サーバプロセスを追加するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、KJS プロセスを追加する iPlanet Application Server インスタンスを選択します。
3. 「ファイル」メニューから、「新規」>「プロセス」をクリックします。
「新規プロセス」ダイアログボックスが表示されます。



4. 「処理タイプ」ドロップダウンリストで、「KJS」または「KCS」を選択します。
5. 「ポート」テキストボックスで、追加のプロセスを実行する未使用のポート番号を指定します。
6. 「OK」をクリックして、新しいプロセスを追加します。
7. このプロセスをシングルスレッド環境で使う場合は、次の手順を実行します。
 - a. 「一般」ウィンドウの左側のペインで必要なプロセスを選択します。
 - b. ウィンドウの右側のペインで、「最小スレッド」と「最大スレッド数」を1に設定します。
8. 「変更の適用」をクリックして変更を保存します。

リクエスト処理のスレッド数の調整

リクエストスレッドはアプリケーションコンポーネントへのユーザのリクエストを処理します。iPlanet Application Server はリクエストを受け取ると、利用可能なスレッドにそのリクエストを割り当てます。スレッドはシステムに対するリクエストのニーズを管理します。たとえば、現在ビジー状態のシステムリソースが必要な場合、スレッドはそのリソースが解放されるのを待ってからリクエストにそのリソースの使用を許可します。

アプリケーションからのリクエスト用に確保するスレッドの最小数と最大数を指定することができます。スレッドプールはこれらの2つの値の間で動的に調整されます。アプリケーションリクエストに備えて、ユーザが指定した最小のスレッド数が維持されます。スレッドは、ユーザが指定した最大値まで増加します。

プロセスに使用可能なスレッドの数を増やすと、そのプロセスは同時により多くのアプリケーションリクエストに応答できるようになります。サーバレベルで、各プロセスのスレッドの追加や調整を行ったり、そのサーバのすべてのプロセスのスレッド数を定義したりできます。

デフォルトでは、各プロセスは iPlanet Application Server に割り当てられているスレッドを使います。たとえば、iPlanet Application Server が最小 8、最大 64 のスレッドを使う場合、個々のプロセスは最小 8、最大 64 のスレッドを使います。

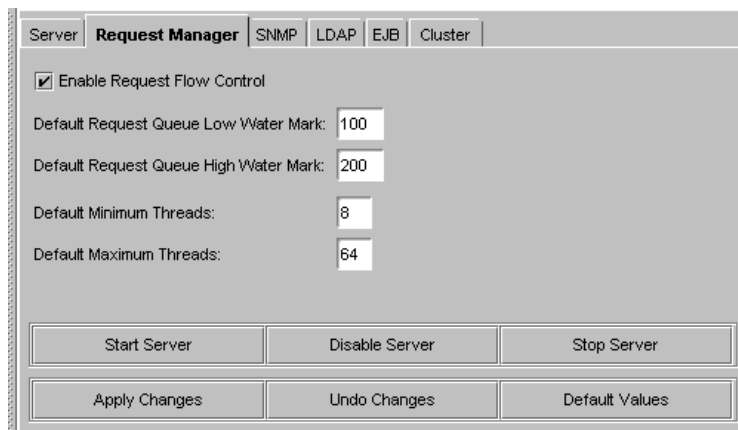
この節では、次のトピックについて説明します。

- サーバレベルでスレッドを調整するには
- プロセスレベルでスレッドを調整するには

サーバレベルでスレッドを調整するには

サーバに対し、すべてのプロセス (KJS/KCS/KXS および IIOP) のリクエストスレッド数を調整するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、スレッドの数を調整するサーバを選択します。
3. 「一般」ウィンドウの右側のペインにある「リクエスト マネージャ」タブをクリックします。
4. 「デフォルト最小スレッド数」テキストボックスに、選択した iPlanet Application Server の各プロセスに使用可能な最小スレッド数を入力します。



5. 「デフォルト最大スレッド数」テキストボックスに、選択した iPlanet Application Server の各プロセスに使用可能な最大スレッド数を入力します。
6. 「変更の適用」をクリックして変更を保存します。

プロセスレベルでスレッドを調整するには

サーバの下にある各プロセスに別のスレッド設定を指定することもできます。プロセスに指定した数とそのプロセスのデフォルト値となることに注意してください。プロセスレベルの設定値はサーバレベルの設定値より優先されます。

サーバの下にあるプロセスに使用可能なスレッド数を調整するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、プロセスを編集するサーバを選択します。サーバを展開してプロセスを表示します。サーバの階層ツリーはサーバの実行時だけ展開できます。
3. スレッド数を調整するプロセスを選択します。
4. 「一般」ウィンドウの右側のペインにある「リクエスト マネージャ」タブをクリックします。
5. 「最小スレッド」テキストボックスに、選択したプロセスに使用可能な最小スレッド数を入力します。

6. 「最大スレッド数」テキストボックスに、プロセスに使用可能な最大スレッド数を入力します。
7. 「変更の適用」をクリックして変更を保存します。

注 プロセスレベルで指定した設定値は、サーバレベルで設定した値より優先されます。

実行プロセスのリクエスト数の指定

Web コネクタプラグインは、iPlanet Application Server アプリケーションへのユーザーリクエストを実行プロセス (KXS) に送ります。これらのリクエストは実行プロセスのリクエストキューに記録されます。

次の手順を実行できます。

- Web コネクタプラグインがプロセスリクエストに対して使う最大スレッド数の制御。これにより、リクエストキューが処理能力以上のリクエストを受け付けないようにする
- リクエストのフローを制御するための、リクエストキューに記録する最大リクエスト数の設定。最大数は「高水位」と呼ばれる
- ログが再開するキューのリクエスト数の設定。この数は「低水位」と呼ばれる

この節には次のトピックがあります。

- サーバレベルでリクエストフローを制御するには
- プロセスレベルでリクエストフローを制御するには

サーバレベルでリクエストフローを制御するには

サーバレベルでのリクエストフローを制御するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、リクエストフローを制御するサーバを選択します。
3. 「一般」ウィンドウの右側のペインにある「リクエスト マネージャ」タブをクリックします。
4. 「リクエスト フロー制御を有効にする」チェックボックスをオンにして、リクエスト フローの制御を有効にします。

5. 「デフォルト リクエスト キュー低水位」テキストボックスで、リクエストログをトリガするキューのリクエスト数を指定します。
この数は、キューのリクエストが最大数に達した場合にのみ適用できます。次の手順を参照してください。
6. 「デフォルト リクエスト キュー高水位」テキストボックスで、キューの最大リクエスト数を指定します。
この数値に達すると、リクエストキューが低水位として指定された数値に減少するまで、ユーザのリクエストは受け取られません。
7. 「変更の適用」をクリックして変更を保存します。

プロセスレベルでリクエストフローを制御するには

プロセスに対してリクエストフローをカスタマイズすることもできます。プロセスに指定した数があるプロセスのデフォルト値となることに注意してください。プロセスレベルの設定値はサーバレベルの設定値より優先されます。

プロセスのリクエストフローを調整するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、リクエストフローを制御するサーバを選択します。サーバを展開してプロセスを表示します。

注 階層ツリーでサーバを展開できるのは、サーバの実行時だけです。

3. 「一般」ウィンドウの右側のペインにある「リクエスト マネージャ」タブをクリックします。
4. 「リクエスト フロー制御を有効にする」チェックボックスをオンにして、リクエスト フロー制御を有効にします。

The screenshot shows a configuration window for the Request Manager. It has two tabs: 'Engine Information' and 'Request Manager'. The 'Request Manager' tab is selected. Inside this tab, there is a checked checkbox labeled 'Enable Request Flow Control'. Below this, there are four input fields: 'Request Queue Low Water Mark' with the value 200, 'Request Queue High Water Mark' with the value 600, 'Minimum Threads' with the value 8, and 'Maximum Threads' with the value 64. At the bottom of the window, there are two rows of buttons. The first row contains 'Start Process', 'Disable Process', and 'Stop Process'. The second row contains 'Apply Changes', 'Undo Changes', and 'Default Values'.

5. 「リクエスト キュー低水位」テキストボックスに、ログを再開するキューのリクエスト数を入力します。
6. 「リクエスト キュー高水位」テキストボックスに、キューの最大リクエスト数を入力します。
これらの設定は、サーバレベルのデフォルト設定より優先されます。
7. 「変更の適用」をクリックして変更を保存します。

注 プロセスレベルで指定した設定値は、サーバレベルで設定した値より優先されます。

Administration Server オプションの設定

Administration Server (KAS) は iPlanet Application Server 内のすべての管理プロセスとタスクを管理します。Administration Server について、サーバリソースの高可用性を実現する複数のオプションを設定できます。これらのオプションを設定することにより、サーバ上で実行するアプリケーションのパフォーマンスの向上や、アプリケーションダウンタイムの発生頻度を減らす対策が可能です。設定可能なオプションは次のとおりです。

- 実行時に EJB コンテナパラメータを指定する
- 最大エンジン再起動回数を指定する
- 国際化サポートを有効にする
- JavaServer Pages (JSP) をキャッシュする
- サーバおよびエンジンの最大シャットダウン時間を指定する

実行時に EJB コンテナパラメータを指定する

iPlanet Application Server では、ユーザ独自の EJB コンポーネントおよびほかの提供者コンポーネントを使って、分散アプリケーションを構築できる EJB コンテナを提供します。企業向けに iPlanet Application Server を設定する場合は、EJB コンテナの宣言パラメータを設定する必要があります。たとえば、これらのパラメータは、EJB が指定された時間 (秒単位) アクティブでなかった場合に削除されるセッションタイムアウトを決めます。iASAT のエディタを使ってこれらのパラメータを設定します。

エディタにアクセスするには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの右側のペインで「EJB」タブをクリックして EJB コンテナ宣言パラメータエディタを開きます。

次のウィンドウが表示されます。

Server	Request Manager	SNMP	LDAP	EJB	Cluster
Default Session Timeout:	<input type="text" value="14400"/>	seconds			
Default Passivation Timeout:	<input type="text" value="60"/>	seconds			
Meta Data Cache Size:	<input type="text" value="30"/>	beans			
Implementation Cache Size:	<input type="text" value="10"/>	beans (per bean type)			
Timer Interval:	<input type="text" value="10"/>	seconds			
Failover Save Interval:	<input type="text" value="10"/>	seconds			

ここでは、次の値を設定できます。

- 「デフォルトセッションタイムアウト」
指定した時間 (秒単位) EJB へのアクセスがないと EJB は削除されます。ステートフルセッション EJB に適用されます。
 - 「デフォルト不活性化タイムアウト」
EJB のステートがディスクに書き込まれるまでの経過時間 (秒単位)。この値はセッションタイムアウトの値より小さくする必要があります。
 - 「メタデータのキャッシュサイズ」
EJB のメタデータのキャッシュ。この値は EJB の数以内です。
 - 「実装キャッシュサイズ」
最大キャッシュサイズは EJB の数以内です。
 - 「タイマー間隔」
EJB プールが EJB を不活性化または削除する必要があるかどうかをチェックする間隔 (秒単位)。
 - 「フェールオーバー保存間隔」
EJB ステートを保存する頻度 (秒単位)。サーバが失敗した場合、最後に保存された EJB のステートが復元されます。保存されたデータはクラスタ内にあるすべてのエンジンからアクセスできます。この値はサーバごとに設定され、Deployment Tool EJB 記述子エディタの「一般」タブで有効にしたフェールオーバーオプションで配置された EJB に適用されます。
3. 「変更の適用」をクリックして変更を保存します。
 4. 変更を有効にするには、iPlanet Application Server を再起動します。

最大エンジン再起動回数を指定する

Executive Server (KXS)、Java サーバ (KJS)、C++ サーバ (KCS)、CORBA Executive Server (CXs) などのプロセスが失敗すると、Administration Server によって再起動されます。再起動オプションを使うと、プロセスが再起動される回数を増減できます。フォールトトレランスとアプリケーションの可用性がすべてのプロセスがスムーズに実行されているときに向上します。

Administration Server の再起動オプションを調整するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、再起動オプションを調整する Administration Server がある iPlanet Application Server インスタンスを選択します。
3. 「一般」ウィンドウの右側のペインにある「サーバー」タブを開きます。
4. 「最大エンジン再起動」テキストフィールドに新しい再起動回数値を入力します。

Server	Request Manager	SNMP	LDAP	EJB	Cluster
Name:	iAS1				
Host:	viper.India.Sun.COM				
IP Address:	129.158.228.95				
Port:	10817				
Server Version:	6.5, Build 20020301				
Maximum Engine Restarts:	<input type="text" value="10"/>				
JSP Cache Size:	<input type="text" value="10"/>				
<input type="checkbox"/> Enable I18N Support					
<input type="checkbox"/> Enable Dynamic Class Reloading					
Maximum Server Shutdown Time:	<input type="text" value="60"/> seconds				
Maximum Engine Shutdown Time:	<input type="text" value="60"/> seconds				

5. 「変更の適用」をクリックして変更を保存します。

国際化サポートを有効にする

iPlanet Application Server が、さまざまな地域の言語でアプリケーションをサポートするように設定できます。

国際化を有効にするには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、国際化を有効にする iPlanet Application Server インスタンスを選択します。
3. 「一般」ウィンドウの右側のペインにある「サーバー」タブをクリックします。
4. 「I18N サポートを有効にする」チェックボックスをオンにします。
5. 「変更の適用」をクリックして変更を保存します。

注 変更を有効にするには、サーバを停止して再起動する必要があります。詳細は、33 ページの「iPlanet レジストリエディタによる管理タスクの実行」を参照してください。

JavaServer Pages (JSP) をキャッシュする

各 iPlanet Application Server インスタンスの各 KJS エンジンによってキャッシュされる JSP ページの数を指定できます。JSP をキャッシュすることにより、アプリケーションの応答時間が最適化されます。

Administration Server の JSP キャッシュ値を設定するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、JSP キャッシュ値を設定する iPlanet Application Server インスタンスを選択します。
3. 「一般」ウィンドウの右側のペインにある「サーバー」タブをクリックします。
4. 「キャッシュサイズ」テキストフィールドに JSP キャッシュサイズを入力します。キャッシュサイズはページごとに設定されます。
5. 「変更の適用」をクリックして変更を保存します。

サーバおよびエンジンの最大シャットダウン時間を指定する

Administrative Server のシャットダウン値は、iPlanet Application Server とエンジンプロセス両方に、設定できます。たとえば、エンジンシャットダウン時間を 60 秒に設定すると、処理中のアプリケーションタスクは処理を終了するのに 60 秒まで認められます。この期間の経過後は、新しいリクエストを受け付けません。シャットダウン値を指定することによって、クライアントにエラーを返す「ハード」シャットダウンを回避します。

Administration Server のサーバおよびエンジンのシャットダウン時間を設定するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、シャットダウン時間を指定する iPlanet Application Server インスタンスを選択します。
3. 「一般」ウィンドウの右側のペインにある「サーバー」タブをクリックします。「サーバシャットダウン最大時間」を指定します。

「サーバシャットダウン最大時間」は、iPlanet Application Server をシャットダウンする最大時間です。この時間を過ぎると、実行中のすべてのエンジンが強制終了されます。通常、負荷が大きくない限り、サーバはすぐにシャットダウンされます。

4. 「最大エンジンシャットダウン時間」を入力します。

「最大エンジンシャットダウン時間」は、iPlanet Application Server がエンジンのシャットダウンを待つ最大時間です。この時間を過ぎると、エンジンは強制終了され、次のエンジンがシャットダウンされます。

5. 「変更の適用」をクリックして変更を保存します。

マルチプロセスのシングルスレッド環境の実現

Java サーバ (KJS) または C++ サーバ (KCS) プロセスを追加することにより、マルチプロセスのシングルスレッド環境を実現できます。全てがシングルスレッドリクエストモードの複数の KJS プロセスを実行することによって、ユーザのリクエストを同時に処理できる「マルチスレッド」環境を効果的に作成することができます。

この環境を実現すると、各プロセスは一度に 1 つのリクエストしか受け取ることができなくなります。これは、サードパーティユーティリティを統合するとき効果的です。iPlanet Application Server マルチスレッドリクエスト環境でサードパーティユーティリティを実行すると、スレッドの安全性の問題など、アプリケーションサーバでは処理できないエラーが発生する場合があります。マルチプロセスのシングルスレッド環境を実現すれば、iPlanet Application Server を拡張したときでも、このような問題を回避することができます。

たとえば、スレッドに対して安全でないサードパーティのユーティリティを KJS プロセス内で実行する場合は、KJS のリクエストスレッドを 1 に調整して、ユーティリティの安全性の問題を解決できます。ただし、この方法では、KJS によってリクエストが 1 つずつ処理されるのをリクエストが待つ間に、リクエストのバックログが作成されます。シングルスレッドリクエストモードで KJS プロセスを実行することによってこの問題を回避し、ユーザのリクエストを同時に処理できる「マルチスレッド」環境を効果的に作成することができます。

Executive Server (KXS) は iPlanet Application Server に送られたすべてのリクエストを分散させるので、KXS プロセスの場合は複数のリクエストスレッドを管理する必要があります。

マルチプロセスのシングルスレッド環境を実現するには、次の手順を実行します。

1. KJS または KCS プロセスを追加します。
130 ページの「Java および C++ サーバプロセスの追加と調整」を参照してください。
2. これらのプロセスに割り当てられたリクエストスレッドを 1 に調整します。
131 ページの「リクエスト処理のスレッド数の調整」を参照してください。

Directory Server のフェールオーバーの設定

iPlanet Application Server インスタンスに接続されている Directory Server は、Directory Server のクラスタ内にあるすべてのアプリケーションサーバで共有されるグローバルな情報を保持します。Directory Server のクラスタは、1 台の Directory Server を共有する 1 つまたは複数の iPlanet Application Server インスタンスです。LDAP サーバの詳細は、98 ページの「LDAP とは」を参照してください。

このグローバルに共有される情報を保護するには、プライマリサーバが失敗した場合にバックアップとして機能するようにセカンダリ Directory Server を設定する必要があります。プライマリ Directory Server が設定されているマシン以外に、バックアップ用の Directory Server をインストールする必要があります。

フェールオーバー用にバックアップ Directory Server を設定するには、次の手順を実行します。

1. 別のマシンに iPlanet Directory Server 5.0 をインストールします。インストールの詳細については、iPlanet Directory Server に付属のマニュアルを参照してください。
2. バックアップ Directory Server をコンシューマ (Consumer) として設定します。
3. バックアップ Directory Server のホスト名およびポート番号が含まれるように、プライマリ Directory Server の設定を変更します。
4. 2 つの Directory Server 間のレプリケーション・アグリーメント (Replication Agreement) を作成します。
5. iPlanet Application Server の Web なしインストールの場合は、Web コネクタの iPlanet レジストリエントリを変更します。

サプライヤ開始レプリケーションの設定

サプライヤ開始レプリケーションは、ディレクトリツリーとサブツリーのマスターコピーを保持しているサーバが、コンシューマ Directory Server にディレクトリデータをレプリケートするレプリケーション設定です。サプライヤ開始レプリケーションを設定するには、まず、レプリケーション設定の更新を受け取るバックアップ Directory Server を設定する必要があります。次に、プライマリ Directory Server の特定のディレクトリツリーのレプリケーション・アグリーメントを設定する必要があります。

バックアップ Directory Server が、レプリケーション設定を受け入れるように設定すると、プライマリ Directory Server に行った変更はバックアップ Directory Server にレプリケートされます。

これらのトピックについては、次の節で詳細に説明します。

- バックアップ Directory Server を設定するには

- プライマリ Directory Server を設定するには
- レプリケーション・アグリーメントを作成するには
- iASAT でバックアップ Directory Server を設定するには
- Web コネクタプラグインのレジストリエントリを設定するには

バックアップ Directory Server を設定するには

バックアップ Directory Server (Consumer) を、サプライヤからのレプリケーションの更新を受け入れるように設定するには、次の手順を実行します。

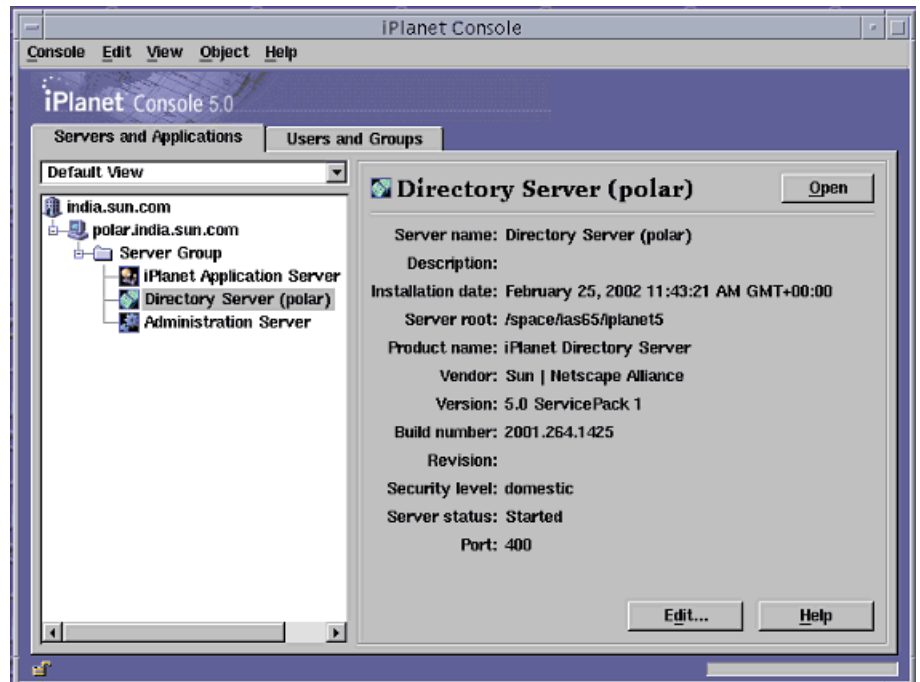
1. バックアップ Directory Server がインストールされているマシン上で iPlanet Console を起動します。

Windows システムでは、「スタート」メニューを開き「プログラム」 > 「iPlanet Server Products」 > 「iPlanet Console 6.5」を選択します。

Windows システムでも Solaris システムでも、パス <iASInstallDir> に移動し、コマンドラインプロンプトに「startconsole」と入力します。

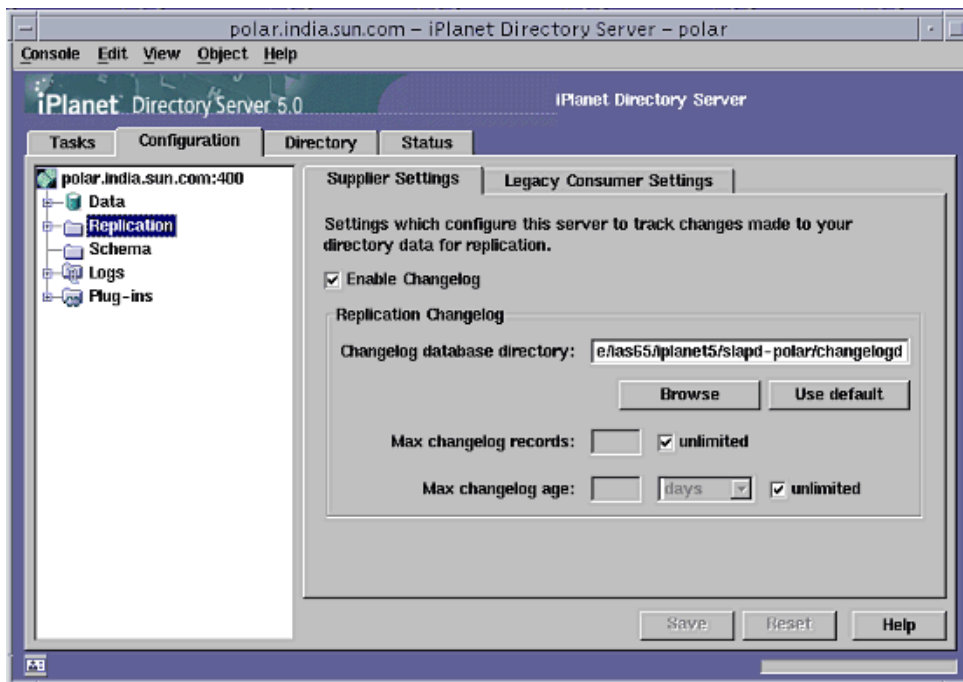
2. 次の図に示すように、Directory Server を選択します。

図 6-1 iPlanet Console 画面



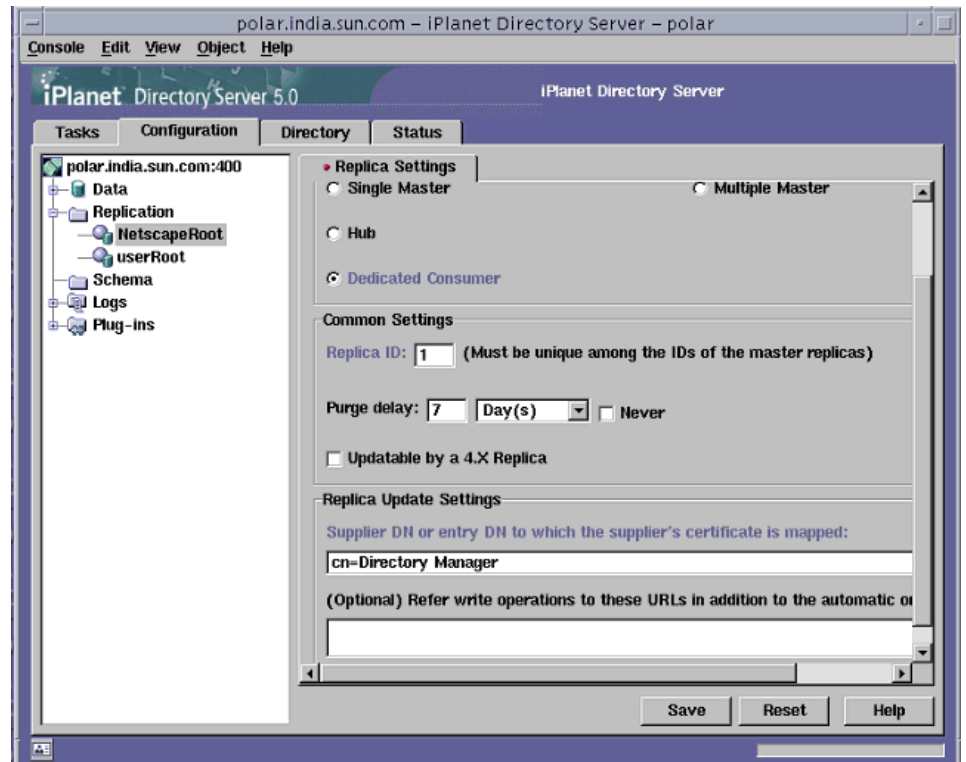
3. 「開く」をクリックして iPlanet Directory Server 設定画面にアクセスします。
4. 「設定」をクリックします。
次のウィンドウが表示されます。

図 6-2 バックアップ Directory Server



5. 左側のペインで、「Replication」を選択します。
6. 左側のペインで、「Replication」を展開して「NetscapeRoot」を選択します。
次のダイアログが表示されます。

図 6-3 「バックアップ Directory Server の設定」



7. 「Enable Replica」を選択します。
8. 「Dedicated Consumer」を選択して、この Directory Server をバックアップ Directory Server に設定します。
9. 「Replica ID」に番号を入力します。1 ～ 255 の間の番号を入力できます。
10. バックアップ Directory Server のエントリが期限切れになるまでの日数を入力します。
サブライヤが作成したエントリを保存しない場合は、「違う」を選択します。
11. 「Supplier DN」テキストフィールドに有効な識別名を入力します。たとえば、「cn=Directory Manager」のように入力します。

サブライヤ Directory Server で 4.x バージョンの iPlanet Directory Server を使用している場合は、「Updatable by a 4.X Replica」を有効にします。その場合、「Legacy Consumer Settings」タブで追加設定を行う必要があります。

12. 「保存」をクリックして変更を確定します。

これで、バックアップ Directory Server が設定されました。次に、ここで設定したコンシューマに更新情報を送信するためにプライマリ Directory Server、あるいはサブライヤを設定する必要があります。

プライマリ Directory Server を設定するには

プライマリ Directory Server (サブライヤ) がバックアップ Directory Server を更新するように設定するには、次の手順に従います。

1. バックアップ Directory Server がインストールされているマシン上で iPlanet Console を起動します。

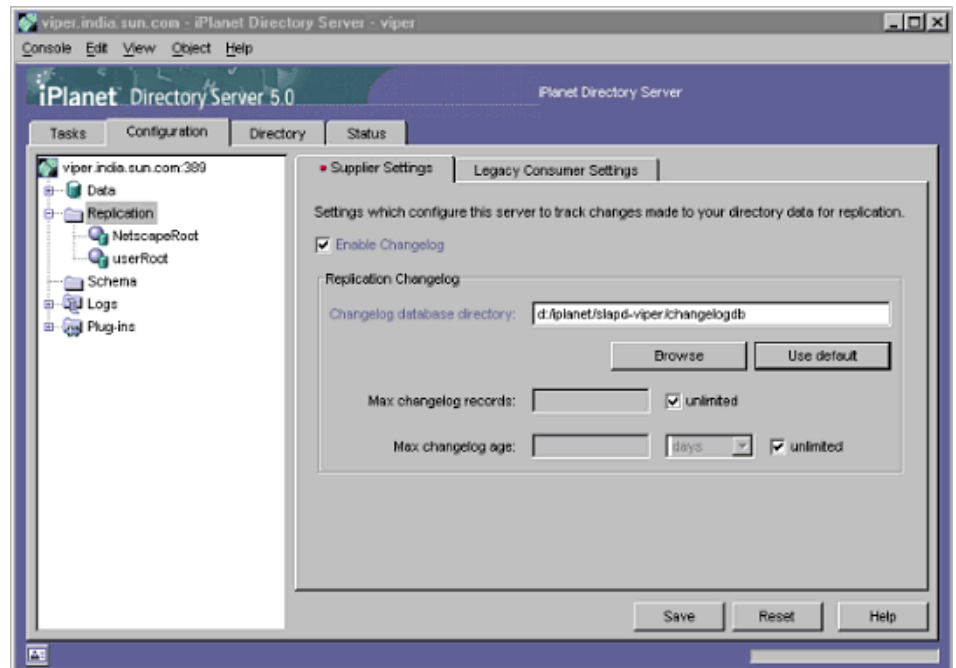
Windows システムでは、「スタート」メニューを開き、「プログラム」>「iPlanet Server Products」>「iPlanet Console 6.5」を選択します。

Windows システムでも Solaris システムでも、パス <iASInstallDir> に移動し、コマンドラインプロンプトに「startconsole」と入力します。

2. 「Directory Server」を選択して「開く」をクリックします。
3. 左側のペインで、「Replication」を選択します。

次のダイアログが表示されます。

図 6-4 「プライマリ Directory Server」



4. 「Enable Changelog」をクリックします。

レプリケーション・アグリーメントの処理を行うには、このオプションを選択する必要があります。

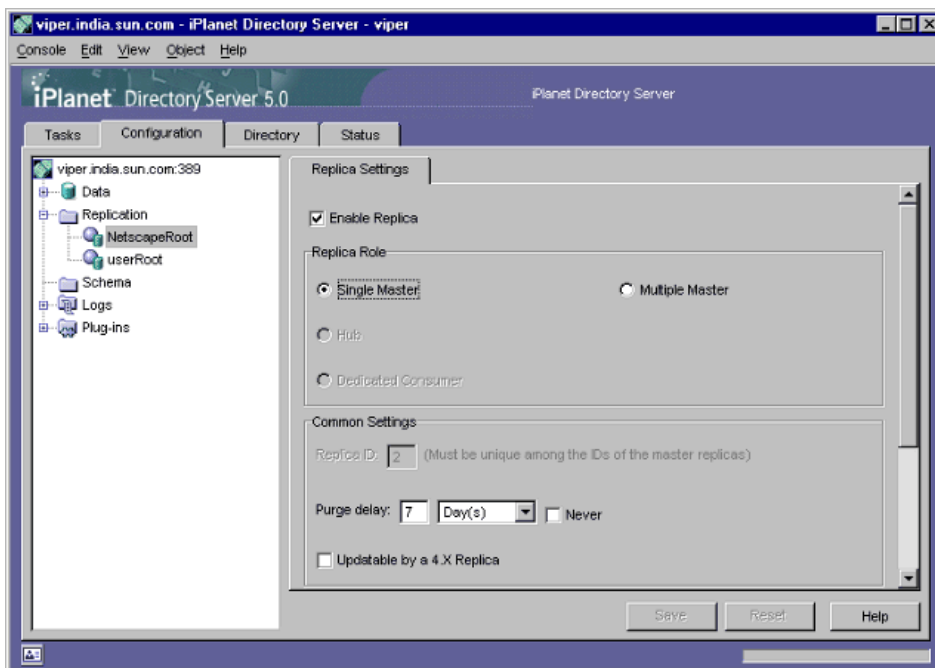
5. 変更ログデータを保存するディレクトリを選択します。

これは、この Directory Server の設定値に対するすべての変更が記録されるディレクトリです。「Changelog database directory」テキストフィールドに変更ログデータベースディレクトリを指定します。ディレクトリを選択するには「Browse」をクリックし、デフォルトの変更ログディレクトリを設定するには「Use default」をクリックします。

デフォルトのディレクトリは `iASInstallDir/slappd-machine_name/changelogdb` に設定されます。

6. 「保存」をクリックして変更を確定します。
7. 左側のペインで、「Replication」を展開して「NetscapeRoot」を選択します。次のダイアログが表示されます。

図 6-5 「プライマリ Directory Server の設定」



8. 「Enable Replica」を選択します。
9. 「Single Master」を選択して、この Directory Server をプライマリ Directory Server (サブライヤ) に設定します。

1つのコンシューマに対して複数のサブライヤ Directory Server を設定し、それぞれが同じサブツリーを保持する場合は、「Multiple Master」オプションを選択します。複数マスターによるレプリケーション設定の詳細については、iPlanet Directory Server のマニュアルを参照してください。

10. 「Replica ID」に番号を入力します。1～255の間の番号を入力できます。
11. バックアップ Directory Server のエントリが期限切れになるまでの日数を入力します。

サブライヤが作成したエントリを保存しない場合は、「違う」を選択します。

12. 「Supplier DN」テキストフィールドに有効な識別名を入力します。たとえば、「cn=Directory Manager」のように入力します。

Supplier Directory Server で 4.x バージョンの iPlanet Directory Server を使用している場合は、「Updatable by a 4.X Replica」を有効にします。その場合、「Legacy Consumer Settings」タブで追加設定を行う必要があります。

13. 「保存」をクリックして変更を確定します。

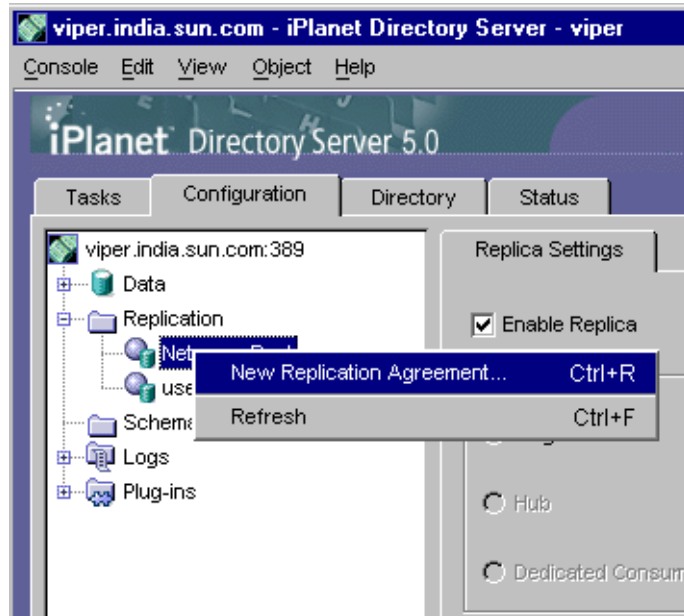
これで、プライマリ Directory Server のサブライヤとしての設定は完了です。次に、サブライヤ Directory Server とコンシューマ Directory Server の間のレプリケーション・アグリーメントを作成する必要があります。

レプリケーション・アグリーメントを作成するには

2つの Directory Server 間のレプリケーション・アグリーメントを作成するには、次の手順に従います。

1. プライマリ Directory Server のコンソールで「NetscapeRoot」>「New Replication Agreement...」を右クリックします。

図 6-6 レプリケーション・アグリーメントの設定



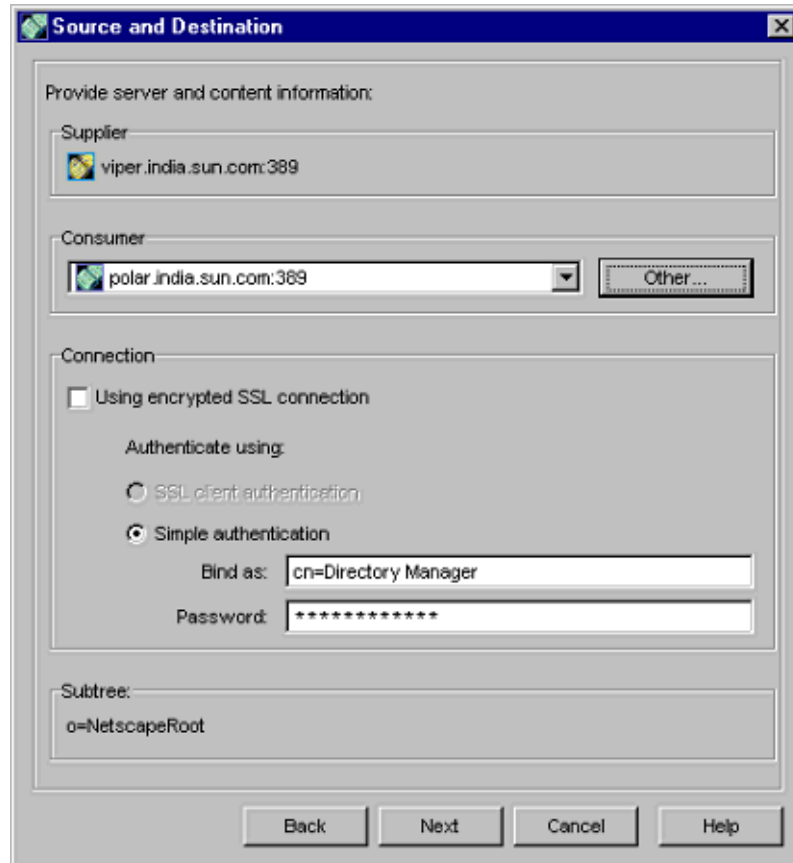
2. このレプリケーション・アグリーメントを識別するための名前と説明を入力します。

たとえば、「Name」に「ReplicaOne」と入力し、「説明」に「iAS LDAP failover」と入力します。

「次へ」をクリックします。

3. バックアップ (コンシューマ) Directory Server の URL と認証証明書を入力します。

図 6-7 Consumer とユーザ証明書の指定



- a. 「Consumer」テキストフィールドに、バックアップ Directory Server がインストールされているマシンのホスト名を入力します。
- b. 「Other」をクリックします。次の図のように、「Consumer Server Information」ポップアップウィンドウが開きます。
- c. 「Host name」テキストフィールドに、バックアップ Directory Server がインストールされているマシンのホスト名を入力します。
「Port number」テキストフィールドのポート番号がプライマリ Directory Server のマシンのポート番号と異なる場合は、ポート番号を入力します。
- d. 「OK」をクリックして確認します。これで、「Source and Destination」ウィンドウに Consumer 名が表示されます。

- e. 簡単に直接的な認証の場合は、「Simple authentication」を選択します。

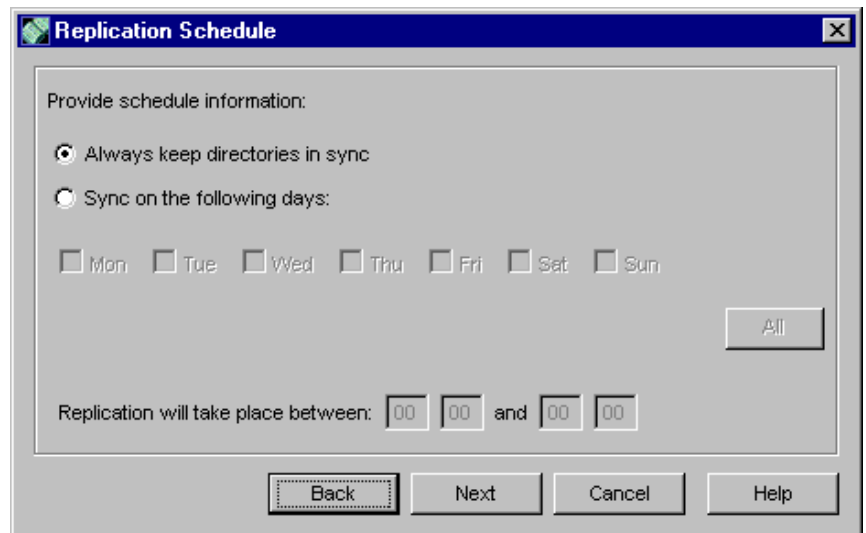
注 より高レベルのセキュリティが必要な場合は、暗号化された SSL コネクションを選択できます。このオプションを選択すると、ユーザは SSL (Secure Socket Layer) プロセスによって認証されます。認証サーバのサブジェクト DN の指定が必要になります。詳細については、iPlanet Directory Server のマニュアルを参照してください。

- f. 「Bind as」テキストフィールドに有効な識別名を指定します。プライマリサーバでは、これを使ってバックアップ Directory Server にバインドします。
- g. 「Password」テキストフィールドに、iPlanet Application Server のインストール時に使ったパスワードを入力します。

注 Solaris システムでは、iPlanet Application Server の複数のインスタンスを保持できます。入力したパスワードが、設定するバックアップ Directory Server の iPlanet Application Server インスタンスのパスワードであることを確認してください。

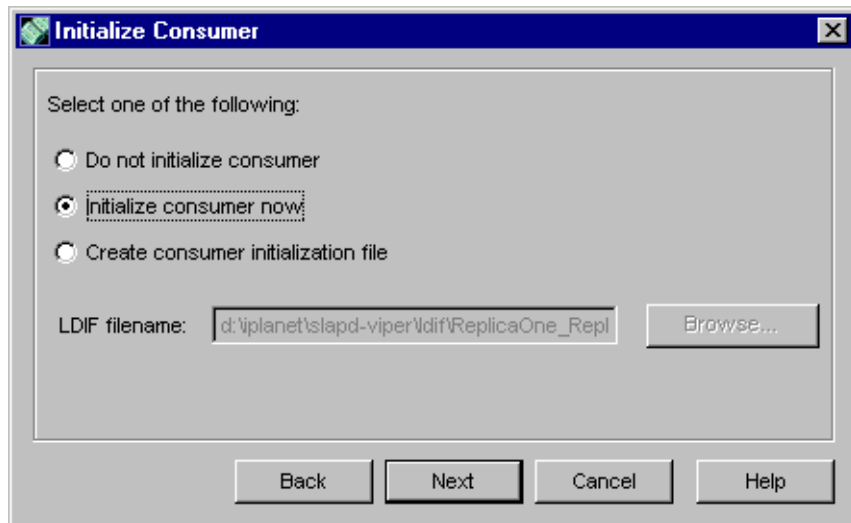
4. 必要なレプリケーションスケジュールを選択します。

図 6-8 レプリケーションスケジュールの指定

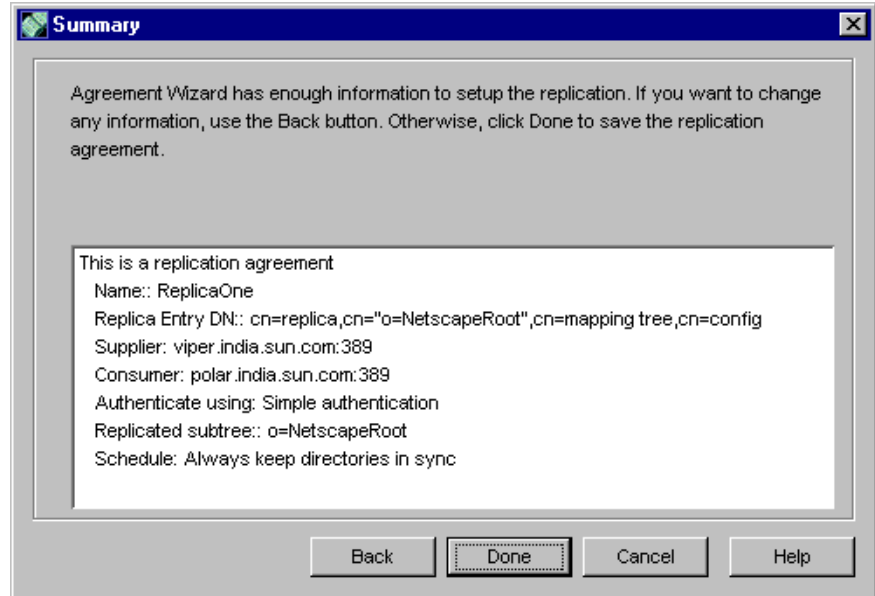


- 両方の Directory Server データを常に最新の状態にするには、「Always keep directories in sync」を選択します。
 - プライマリ Directory Server とバックアップ Directory Server 間のデータの同期をスケジュールする場合は、「Sync on the following days」を選択します。選択した曜日の指定した時刻に、データのレプリケーションが行われます。
 - 毎日、データの同期をとる場合は、「All」をクリックします。
5. コンシューマへの接続の起動に使用する方法を選択します。

図 6-9 コンシューマ初期化オプション



- この時点でレプリケーションを中止するには、「Do not initialize consumer」を選択します。このオプションを選択すると、データはバックアップ Directory Server にレプリケートされません。
 - 「Initialize consumer now」ラジオボタンをオンにします。このオプションによって、バックアップ Directory Server がただちに初期化されます。これでバックアップ Directory Server のレプリケーションの準備ができました。
 - ファイルにデータを書き込む場合は、「Create consumer initialization file」を選択します。データを保存するための .ldif ファイルを選択します。このファイルはあとでバックアップ Directory Server にコピーされます。詳細は、104 ページの「LDIF による Directory Server へのエントリの追加」を参照してください。
6. 選択した値が summary 画面に表示されます。



7. 「完了」をクリックしてレプリケーション・アグリーメントの手順を完了します。変更がある場合は、「戻る」をクリックします。

「Initialize Consumer」画面で「initialize consumer now」オプションを選択した場合は、この時点でバックアップ Directory Server がプライマリ Directory Server によって更新されます。

注 `cn=iasRoot` をレプリケートする必要があります。このノードもレプリケーションスキーマに追加するには、151 ページの「レプリケーション・アグリーメントを作成するには」の手順に従ってください。

8. レプリケーションが正しく完了すると、確認メッセージが表示されます。レプリケーションの完了後、バックアップ内のプライマリ Directory Server のエントリを確認できます。

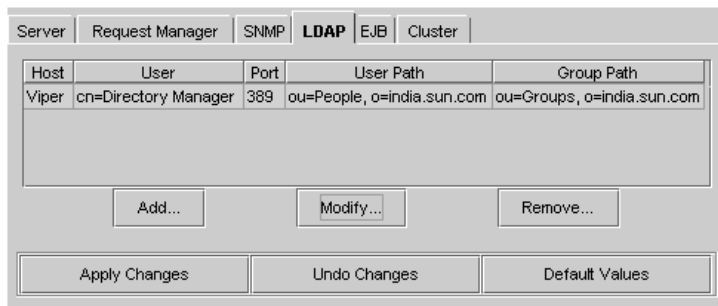
注 証明書ベースの認証の詳細は、『iPlanet Directory Server 管理者ガイド』の「レプリケーションの管理」の章を参照してください。このマニュアルは、Directory Server のインストール時に次の場所にインストールされます。

`<iASInstallDir>/manual/en/slapped/ag/replicat.htm`

iASAT でバックアップ Directory Server を設定するには

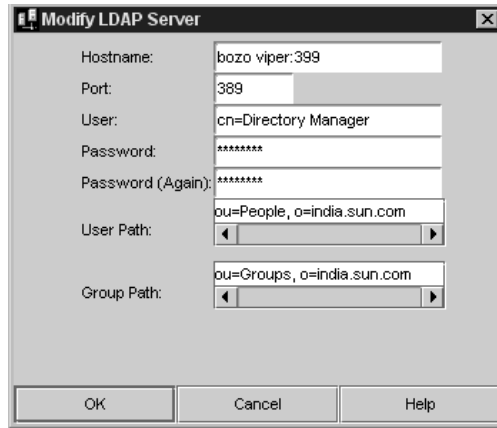
バックアップ Directory Server をインストールしてレプリケートしたら、iASAT で、プライマリ Directory Server の設定を変更し、バックアップ Directory Server の名前とポート番号を入力する必要があります。これを行うには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. バックアップ Directory Server を設定する iPlanet Application Server インスタンスを選択します。
3. 「一般」ウィンドウの右側のペインにある「LDAP」タブを開きます。次のウィンドウが表示されます。



使用している iPlanet Application Server インスタンスに関連付けられている Directory Server がウィンドウに表示されます。

4. プライマリ Directory Server を選択し、「変更」をクリックします。次のダイアログボックスが表示されます。



5. 上の図に示すように、「ホスト名」フィールドで、既存のホスト名の後ろに空白を挿入し、バックアップ Directory Server がインストールされているマシンの名前を入力します。バックアップ Directory Server マシンのポート番号が異なる場合は、ホスト名の隣にコロン(:)を挿入し、ポート番号を指定します (bozo viper:399 など)。

ポート番号の指定が必要なのは、ホストのポート番号と異なる場合だけです。

6. 「OK」をクリックしてウィンドウを閉じます。
7. 「変更の適用」をクリックし、iASAT で変更を登録します。
8. 変更を有効にするには、iPlanet Application Server を停止して、再起動する必要があります。

これでバックアップ Directory Server が設定できました。iPlanet Application Server と連動するように設定した Directory Server が、常に1つ以上必要であることに注意してください。

注 Directory Server を削除するには、iASAT の「一般」ウィンドウの右側のペインで削除する Directory Server を選択し、「削除」をクリックします。

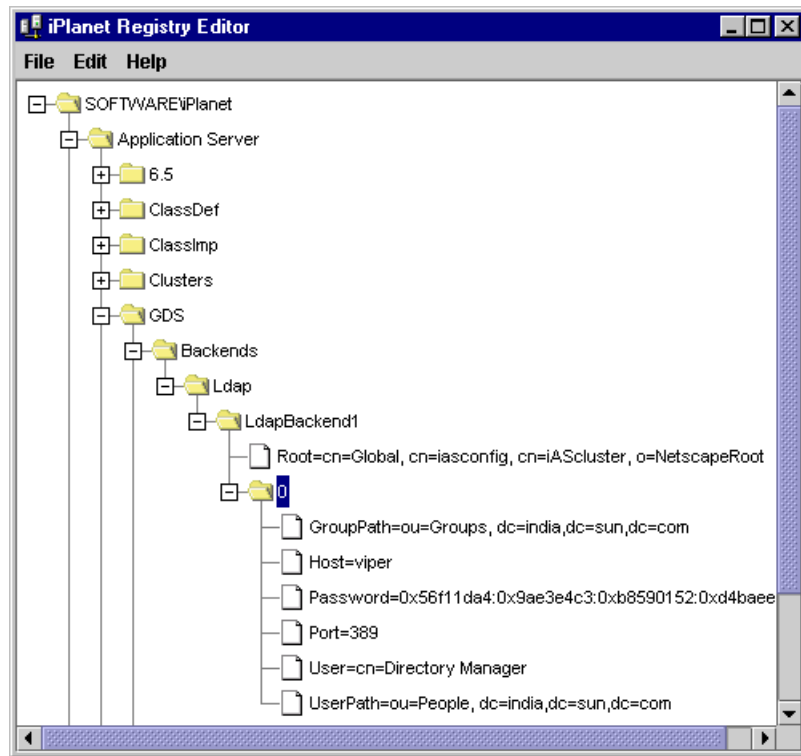
Web コネクタプラグインのレジストリエントリを設定するには

バックアップ Directory Server を設定すると、iPlanet Application Server によって、iPlanet レジストリに必要なバックエンドエントリが作成されます。ただし、iPlanet Application Server の Web なしインストールの場合、つまり iPlanet Application Server および Web コネクタが別のマシンにインストールされている場合は、iPlanet レジストリのインスタンスを 2 つの持っていることとなります。バックアップ Directory Server がインストールされているマシンのホスト名とポート数で、Web コネクタ上の iPlanet レジストリエントリを更新する必要があります。

バックアップの Directory Server の詳細で iPlanet レジストリを更新するには、次の手順を実行します。

1. Web コネクタがインストールされているマシン上で iPlanet レジストリを起動します。
2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application  
Server¥GDS¥Backends¥Ldap¥LdapBackend1¥0¥
```

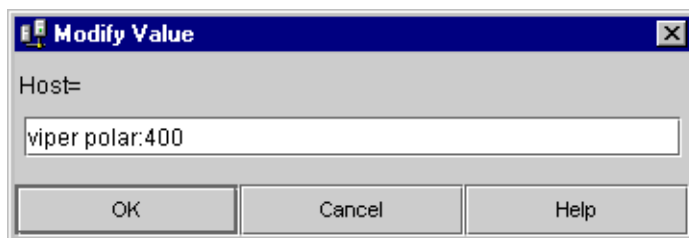


3. Host=<hostname> 値を選択します。

この値をダブルクリックして開くか、または、「編集」メニューから「値を変更」を選択します。「値を変更」ダイアログボックスが表示されます。

4. 既存のホスト名の後ろに空白を挿入し、バックアップ Directory Server がインストールされているマシン名を入力します。バックアップ Directory Server マシンのポート番号が異なる場合は、ホスト名の隣にコロン (:) を挿入し、ポート番号を指定します。

図 6-10 コンシューマ Directory Server のレジストリへの追加



5. 変更を有効にするには、iPlanet レジストリエディタを停止して、再起動します。

Web コネクタプラグインの設定

この章では、iPlanet Application Server 上にあるアプリケーションにユーザのリクエストを送信する Web コネクタのプラグインについて説明します。

この章には次のトピックがあります。

- Web コネクタプラグインについて
- Web サーバの手動設定
- Web サーバが記録するための Web コネクタプラグインの設定
- cookie と隠しフィールドの設定
- URL 書き換えの設定
- CGI リクエストのための CGI フラグの設定
- Web コネクタのポート番号の変更
- アプリケーションコンポーネントへの入力のための HTTP 変数の指定

Web コネクタプラグインについて

Web コネクタプラグインは、iPlanet Application Server のインストール時に、使用している Web サーバにインストールされます。

Web サーバがインストールされているマシンに iPlanet Application Server をインストールすると、Web コネクタプラグインが同時にインストールされ、Web サーバが自動的に設定されます。

Web サーバがインストールされていないマシンに iPlanet Application Server をインストールする場合は、その Web サーバマシンに Web コネクタプラグインを手動でインストールする必要があります。Web コネクタを手動でインストールする方法については、『インストールガイド』を参照してください。

次の Web コネクタプラグインの機能を設定できます。

表 7-1 設定可能な Web コネクタプラグインの機能

コネクタの機能	説明	詳細
Web サーバのリクエストの記録	Web サーバのリクエストコンポーネントをデータベースフィールドに割り当て、HTTP 変数をログに追加する	171 ページの「Web サーバが記録するための Web コネクタプラグインの設定」
cookie と隠しフィールドのセキュリティ	Web サーバから iPlanet Application Server への通信時に cookie と隠しフィールドの有効 / 無効を切り替える	173 ページの「cookie と隠しフィールドの設定」
CGI リクエストを処理するための CGI フラグ	必要に応じて CGI モードでリクエストを処理するためのフラグを設定する	174 ページの「CGI リクエストのための CGI フラグの設定」
プラグインのポート番号	プラグインが使うポート番号を再設定する	175 ページの「Web コネクタのポート番号の変更」
アプリケーションコンポーネントへの入力として HTTP 変数を設定する	アプリケーションコンポーネントがどの HTTP 変数にアクセスできるかを決定する	175 ページの「アプリケーションコンポーネントへの入力のための HTTP 変数の指定」

Web サーバの手動設定

iPlanet Application Server をインストールすると、使用している Web サーバは Web コネクタプラグイン用に自動的に設定されます。つまり、Web サーバ上の必要なディレクトリと設定がすべて更新されます。ただし、Web コネクタプラグインをインストールしたときなどには、Web サーバを手動で再設定する必要があります。この手順は iPlanet Application Server と使用している Web サーバ間の接続に問題がある場合のみお勧めします。

次の手順では、使用している Web サーバと、iPlanet Application Server がインストールされているマシンが同一であるかにかかわらず、Web コネクタプラグインを使うために Web サーバを手動で設定する方法について説明します。

次の手順の中で手順 1 (CGI を有効にする) だけを実行する場合、Web コネクタプラグインは CGI スクリプトとして実行されます。手順全体を実行する場合、Web コネクタプラグインはプラグインとして実行されます。プラグインは、CGI スクリプトより高速で効率的です。

Web サーバをインストールした管理者ユーザとしてログインする必要があります。

この節には次のトピックがあります。

- iPlanet Web Server の設定
- Apache Web サーバの設定
- Microsoft Internet Information Server の設定

iPlanet Web Server の設定

iPlanet Web Server を再設定するには、次の手順を実行します。

1. CGI がまだ有効になっていない場合は CGI を有効にします。
 - a. Windows では、「スタート」メニューから「iPlanet Web Server」 > 「Administer Web Server」を選択します。
 - b. 管理者の ID とパスワードを入力し「OK」をクリックします。
 - c. 「iPlanet Server Selector」画面で、ドロップダウンリストから設定する Web サーバのインスタンスを選択して、「Manage」をクリックします。
 - d. メニューツールバーから「Programs」をクリックします。
 - e. 「URL prefix」の下の「CGI directory」画面に「cgi-bin」と入力します。
 - f. 「CGI directory」の下に cgi-bin のパスを入力します。

iPlanet Web Server 4.1 およびそれ以降の Windows の場合

```
<drive letter>:\Netscape\Server4\docs\cgi-bin
```

iPlanet Web Server 4.1 およびそれ以降の Solaris の場合

```
<iWSInstallDir>/docs/cgi-bin
```

これで、Web コネクタプラグインを設定する準備ができました。

2. Web サーバの設定ディレクトリ内にある obj.conf ファイル (iWS 4.1 まで) または magnus.conf ファイル (iWS 6.0 以降) を編集します。

iWS4.1 では、obj.conf ファイルのみを変更します。iWS6.0 以降では、magnus.conf ファイルも変更します。

iPlanet Web Server 4.1、Windows の場合は、次のパスにある obj.conf ファイルを変更します。

```
drive letter:%Netscape%Server4%https-machinename%config
```

iPlanet Web Server 4.1、Solaris の場合は、次のパスにある obj.conf ファイルを変更します。

```
iWSInstallDir/https-machinename/config
```

ファイルを変更する前にそのファイルをコピーします。obj.conf ファイルの Init セクションの末尾に次の 2 行を追加します。

- Windows の場合

```
Init fn="load-modules"
    funcs=nas_name_trans,gxrequest,gxlog,gxinit,gxredirect,
    gxhtmlrequest shlib="<path to iAS bin dir>%gxnsapi6.dll"

Init fn="gxinit"
```

- Solaris の場合

```
Init fn="load-modules"
    funcs=nas_name_trans,gxrequest,gxlog,gxinit,gxredirect,
    gxhtmlrequest shlib="libgxnsapi6.so"

Init fn="gxinit" LateInit=yes
```

iPlanet Web Server 4.1 の場合は shlib に次の文字列を指定します。

- Windows の場合

```
<iASInstallDir>%bin%gxnsapi6.dll
```

- Solaris の場合

```
<iASInstallDir>/gplib/libgxnsapi6.so
```

次の変更は、すべてのバージョンの iPlanet Web Server の obj.conf ファイルで行う必要があります。

3. Object name=default セクションで type=text/plain セクションの直後に次の行を追加します。

```
Service fn="gxredirect" filename="imagemap" method="(GET|HEAD)"
```

4. Object name=cgi セクションで Service fn="send-cgi" の行の直前に次の行を挿入します。

```
Service fn="gxrequest"
```

さらに、行 Service fn="send-cgi" の直後に次の行を挿入します。

```
AddLog fn="gxlog"
```

5. 現在のバージョンの obj.conf ファイルと magnus.conf ファイルのコピーを作成し、バックアップが現在のバージョンと一貫性を保つように、次のディレクトリ内のバックアップバージョンにコピーします。

Windows の場合

```
drive letter:¥iPlanet¥SuiteSpot¥https-machinename¥conf_bk
```

Solaris の場合

```
iWSIstallDir/https-machinename/conf_bk
```

6. Solaris の場合のみ Web サーバの開始スクリプトと終了スクリプトを次のように変更します。

開始スクリプトの場合

GX_ROOTDIR を iPlanet Application Server がインストールされているディレクトリに設定します。次のようにします。

```
GX_ROOTDIR=iASInstallDir; export GX_ROOTDIR
```

7. Web サーバを再起動します。

Apache Web サーバの設定

Apache Web サーバ 1.3.19 は、iPlanet Application Server 6.5 リリースでサポートされています。iPlanet Application Server の Web コネクタプラグインで Apache Web サーバを使うには、いくつかの設定手順を手動で実行する必要があります。

Web コネクタプラグインをインストールしたマシンに Apache Web サーバをインストールして設定する必要があります。

注 Apache Web サーバを使うには、アプリケーションサーバのインストール時に iPlanet Web Server または Microsoft Internet Information Server (Windows のみ) を選択します。iPlanet Application Server をインストール後、Apache Web サーバをインストールして設定します。

Solaris で Apache Web サーバをインストールするには

1. Apache Web サイト www.apache.org から Apache Web サーバ 1.3.19 ソースファイルをダウンロードし、tar.gz ファイルをアンパックします。

注

- ar はパスに存在しなければなりません。
 - iPlanet Application Server と Apache のオーナーおよびグループは同一にします。デフォルトでは、Apache は nobody:nobody として動作します。
たとえば、iPlanet Application Server ファイルのオーナーおよびグループを nobody:nobody に変更する場合は、次のコマンドを入力します。

```
chown -R nobody:nobody <iASInstallDirectory/ias/>
```
 - Apache では、Web サーバを root として実行しないことをお勧めします。詳細については、Apache Web サイトを参照してください。

root として実行するには、コンパイルする前に、シェル変数 EXTRA_CFLAGS を -DBIG_SECURITY_HOLE に設定し直します。
 - 古いバージョンの GNU binutils (2.7 以前) を使用している場合は、オプション `-export-dynamic` を GNU リンカーに渡します。このオプションがないと、実行時リンカーはプラグインで参照される特定の Apache シンボルを解決できず、Web サーバは起動時にクラッシュします。
-

2. ソースファイルが保存されて実行されるルートディレクトリに移動します。

```
./configure --prefix=<Installation_Directory>
--enable-module=so --with-port=<PortNo.>
make
make install
```

-
- 注
- `enable-module=so` は必須です。
 - ポート番号値はオプションで、デフォルトでは **80** に設定されています。
-

Apache コントロールスクリプトは `apachectl` という名前で、インストールの `bin` ディレクトリにあります。

スクリプトを開始するには、次のコマンドを入力します。

```
apachectl start
```

停止するには、次のコマンドを入力します。

```
apachectl stop
```

Solaris で Apache Web サーバを設定するには

1. シェル変数 `GX_ROOTDIR`, `LD_LIBRARY_PATH` と `LD_PRELOAD` を `apachectl` スクリプトに追加します。

コード例 7-1

```
# iPlanet Application Server への追加
#
GX_ROOTDIR=<iASInstallDir>/ias
export GX_ROOTDIR
LD_LIBRARY_PATH=${GX_ROOTDIR}/gxlib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
LD_PRELOAD=${GX_ROOTDIR}/gxlib/libiasdl.so
export LD_PRELOAD
#
```

-
- 注
- `<iASInstallDir>` を iPlanet Application Server のインストールディレクトリで置き換えます。
 - `<ApacheInstallDir>` を Apache Web サーバのインストールディレクトリで置き換えます。
 - Solaris 2.6 のみ `LD_PRELOAD` の行が必要です。
-

2. <ApacheInstallDir>/conf/httpd.conf の末尾に次の行を追加します。

コード例 7-2

```
# iPlanet Application Server への追加
#
LoadModule nas_module<iASInstallDir>/ias/gxlib/libiASApachePlugin.so
AddModule iASApacheInterface.cpp
<Location "/">
SetHandler ias-handler
</Location>
#
```

3. <iASInstallDir>/ias/GXApp および <iASInstallDir>/ias/ias-samples ディレクトリを <apacheInstallDir>/htdocs/ ディレクトリにコピーします。

Web サーバの最小数、最大数、および開始番号は conf/httpd.conf 内の次のキーによって制御されます。これらは Apache が提示するデフォルト値です。

```
MinSpareServers 5
MaxSpareServers 10

StartServers 5
```

すべての Web サーバが同じログファイルに書き込みます。

4. 169 ページの「iPlanet Application Server レジストリの設定」に示すように、kregedit を使って iPlanet Application Server レジストリを変更します。

Windows での Apache のインストール

Apache Web サイト www.apache.org から Apache 1.3.19 のバイナリをダウンロードしてインストールします。Windows での Apache のインストールの詳細については、Apache Web サイトに掲載されている指示に従ってください。

Windows で Apache Web サーバを設定するには

1. システム変数 PATH で実行できる Apache のパスを追加します。

2. iASApachePlugin.dll (Web コネクタプラグイン) のパスをシステム変数 PATH に追加します。

パスは、<iASInstallDir>/ias/bin になります。

-
- 注
- <iASInstallDir> を iPlanet Application Server のインストールディレクトリで置き換えます。
-

3. <ApacheInstallDir>/conf/httpd.conf の末尾に次の行を追加します。

コード例 7-3

```
# iPlanet Application Server への追加
#
LoadModule nas_module<iASInstallDir>/ias/bin/iASApachePlugin.dll
AddModule iASApacheInterface.cpp
<Location "/">
SetHandler ias-handler
</Location>
#
```

4. <iASInstallDir>/ias/GXApp および <iASInstallDir>/ias/ias-samples ディレクトリを <ApacheInstallDir>/htdocs/ ディレクトリにコピーします。
5. 169 ページの「iPlanet Application Server レジストリの設定」に示すように、kregedit を使って iPlanet Application Server レジストリを変更します。

iPlanet Application Server レジストリの設定

インストール後、kregedit を使って、iPlanet Application Server レジストリ内の次のキーを変更します。

1. SOFTWARE/iPlanet/Application Server/6.5/CCS0/HTTPAPI/HTTPPort を Apache Web サーバのリスナポートに設定します。
2. SOFTWARE/iPlanet/Application Server/6.5/CCS0/HTTPAPI/PATH と SOFTWARE/iPlanet/Application Server/6.5/deployment/LogicalName/WWW_DOCROOT をドキュメントルートディレクトリに設定します。
3. ドキュメントルートディレクトリ <Apache_Installation_Directory>/htdocs を SOFTWARE/iPlanet/Application Server/6.5/CCS0/TEMPLATE/PATH に追加し、セミコロンで区切ります。

Microsoft Internet Information Server の設定

Microsoft IIS を再設定するときは、次のことに注意してください。

- `gxisapi.dll` ライブラリの名前を `gx.dll` に変更し、IIS `wwwroot` の `cgi-bin` ディレクトリに置く (`inetput/wwwroot/cgi-bin/`)
- 次のレジストリエントリ内に ISAPI フィルタファイル `gx.dll` を設定する
My Computer¥HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Services
¥W3SVC¥Parameters¥
Parameters の下に次の値とともに文字列キー Filter DLLs を追加する
<drive letter>:¥inetpub¥wwwroot¥cgi-bin¥gx.dll
- Webconnector プラグインが Windows レジストリへの書き込みパーミッションを持たない場合は例外をスローする
プラグインがレジストリを変更できるようにするには、次の手順を実行します。
 - a. Windows レジストリエディタを開きます。
 - b. ¥¥HKEY_LOCAL_MACHINE¥SOFTWARE¥iPLANET¥Application Server¥6.5¥CCSO¥HTTPAPI に移動します。
 - c. メニューオプションから Permission Definition キーを開きます。
 - d. パーミッションを Special Access から Full Control に変更します。
 - e. サブキーの変更を有効にします。
 - f. 「変更の適用」をクリックして変更を確定します。

Web サーバが記録するための Web コネクタプラグインの設定

Web サーバのリクエストはコンポーネントに分かれています。各コンポーネントは HTTP 変数で表されます。HTTP 変数は Web サーバ全体で標準化されているので、Web サーバの使用に関してユーザが行う設定は、Web サーバに依存しません。

この節では、次のトピックについて説明します。

- データベースフィールドへの HTTP 変数の割り当て
- ログへの HTTP 変数の追加

データベースフィールドへの HTTP 変数の割り当て

Web サーバリクエストの特定のコンポーネントのログを有効にするには、Web サーバリクエストが正しく記録されるように、HTTP 変数を特定のデータベースフィールドに割り当てる必要があります。データベースフィールドに HTTP 変数を割り当てるには、Web サーバマシン上の Web コネクタプラグインを使います。Web サーバマシンと iPlanet Application Server は、同一マシンにインストールされていなくてもかまいません。

HTTP 変数をデータベースフィールドに割り当てるには、次の手順を実行します。

1. コマンドラインに「kregedit」と入力し、iPlanet レジストリエディタを開きます。

エディタツールが開いて、iPlanet Application Server に適用するキーと値が表示されます。Web サーバと iPlanet Application Server が別のマシンにインストールされている場合は、エディタが開いて、Web コネクタプラグインに適用するキーと値が表示されます。

2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥CCS0¥HTTPLOG¥INPUTVARS
```

このキーの下の各値は、HTTP 変数と、その変数が割り当てられるデータベースフィールドを表します。

値の ID は HTTP 変数です。文字列値はデータベースフィールドです。

HTTP 変数は HTTP_REFERER のようにすべて大文字です。また、データベースフィールドはデータベーステーブルに表示されるフィールドと同じです。

3. データベースフィールドに割り当てる HTTP 変数をダブルクリックします。

「文字列」編集ダイアログボックスが表示されます。

4. データベースフィールドの名前を値のデータとして入力し、「OK」をクリックします。
5. 記録しない HTTP 変数はすべて空白にしておきます。
6. エディタを閉じます。

HTTP 変数の説明については、使用している Web サーバのマニュアルを参照してください。

iPlanet レジストリエディタを使って Web コネクタプラグインを変更します。

ログへの HTTP 変数の追加

使用可能な HTTP 変数のリストを変更し、変数をリストに追加してログオプションを拡張することもできます。

HTTP 変数をログに追加するには、次の手順を実行します。

1. コマンドラインに「kregedit」と入力し、iPlanet レジストリエディタを開きます。

エディタが開き、iPlanet Application Server に適用されるキーと値が表示されます。Web サーバと iPlanet Application Server が別のマシンにインストールされている場合は、エディタが開いて、Web コネクタプラグインに適用するキーと値が表示されます。

2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥CCS0¥HTTPLOG¥INPUTVARS
```

このキーの下の各値は、HTTP 変数と、その変数が割り当てられるデータベースフィールドを表します。

値の ID は HTTP 変数です。文字列値はデータベースフィールドです。

HTTP 変数は HTTP_REFERER のようにすべて大文字です。また、データベースフィールドはデータベーステーブルに表示されるフィールドと同じです。

3. 新しい HTTP 変数名を使って新しい文字列値を追加します。
4. 「OK」をクリックします。
5. 新しい HTTP 変数ごとに手順 3 ~ 5 を繰り返します。
6. エディタを閉じます。

すべての使用可能な HTTP 変数のリストと説明については、使用している Web サーバのマニュアルを参照してください。

cookie と隠しフィールドの設定

iPlanet Application Server は、cookie と隠しフィールドセキュリティに関するすべてのモードで Web ブラウザと一緒に動くように設計されています。さまざまなセキュリティモードをサポートするために、Web コネクタプラグインには 3 つの設定が用意されています。次の表でこれらの設定について説明します。

表 7-2 セキュリティモードをサポートする設定

cookie の設定	説明
0	cookie と隠しフィールドが、リクエスト元の Web ブラウザに返される。これはデフォルトの設定である
1	隠しフィールドだけが、リクエスト元の Web ブラウザに返される
2	cookie だけが、リクエスト元の Web ブラウザに返される

cookie と隠しフィールドの使い方を設定するには、次の手順を実行します。

1. コマンドラインに「kregedit」と入力し、iPlanet レジストリエディタを開きます。

エディタツールが開いて、iPlanet Application Server に適用するキーと値が表示されます。Web サーバと iPlanet Application Server が別のマシンにインストールされている場合は、レジストリエディタが開いて、Web コネクタプラグインに適用するキーと値が表示されます。

2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥CCSO¥HTTPAPI
```

3. NoCookie DWORD 値をダブルクリックします。

「値を変更」ダイアログボックスが表示されます。

4. Web ブラウザに渡される cookie を無効にするには、値のデータを 1 に変更します。
5. Web ブラウザに渡される隠しフィールドを無効にするには、値のデータを 2 に変更します。
6. cookie と隠しフィールドを有効にするには、値のデータを 0 に変更します。
7. 終了したらエディタを閉じます。

URL 書き換えの設定

iPlanet Application Server では、ユーザは暗黙的な URL 書き換え (デフォルトの方式) と明示的な URL 書き換えのどちらを使用するかを選択できます。

デフォルトでは、暗黙的な URL 書き換えが有効になっています。明示的な URL 書き換えを有効にするには、kregedit を使って iPlanet Application Server レジストリ内の次のパスにある default=implicit キーを変更します。

```
SOFTWARE/iPlanet/Application Server/6.0/CCSO/HTTPAPI/URLRewrite
```

明示的な URL 書き換えを有効にするには、値 default=implicit を default=explicit に変更します。

暗黙的な URL 書き換えは Webconnector プラグインによって実行されるので、開発者によるコードの変更は必要ありません。Web コンテナによって実行される明示的な URL 書き換えを有効にする場合は、開発者は、encodeURL と encodeRedirectURL を呼び出すように Servlet または JSP のコードを変更する必要があります。

URL 書き換えの詳細については、『iPlanet Application Server 開発者ガイド』を参照してください。

CGI リクエストのための CGI フラグの設定

一部のリクエストは、CGI モードで処理する必要があります。Web コネクタプラグインでフラグを設定すれば、これらのリクエストを識別できます。

CGI リクエストのための CGI フラグを設定するには、次の手順を実行します。

1. コマンドラインに「kregedit」と入力し、iPlanet レジストリエディタを開きます。

エディタが開き、iPlanet Application Server に適用されるキーと値が表示されます。Web サーバと iPlanet Application Server が別のマシンにインストールされている場合は、エディタが開いて、Web コネクタプラグインに適用するキーと値が表示されます。

2. 次のキーを開きます。

```
SOFTWARE\iPlanet\Application Server\6.5\CCSO\HTTPAPI
```

3. AgentToken 文字列値をダブルクリックします。
「文字列」編集ダイアログボックスが表示されます。
4. 値データとして、CGI モードで処理するリクエストにフラグを入力します。
5. 「OK」をクリックします。

6. エディタを閉じます。

Web コネクタのポート番号の変更

設定によっては、Web コネクタのポート番号がほかのソフトウェアパッケージと競合する場合があります。コネクタのポート番号を再設定すれば、この競合を解消できます。

Web コネクタプラグインのポート番号を変更するには、次の手順を実行します。

1. コマンドラインに「kregedit」と入力し、iPlanet レジストリエディタを開きます。

エディタが開き、iPlanet Application Server に適用されるキーと値が表示されます。Web サーバと iPlanet Application Server が別のマシンにインストールされている場合は、エディタが開いて、Web コネクタプラグインに適用するキーと値が表示されます。

2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥CCSO¥HTTPAPI
```

3. ListenPort DWORD 値をダブルクリックして、値データを使用可能なポート番号に変更します。
4. 「OK」をクリックします。
5. エディタを閉じます。

アプリケーションコンポーネントへの入力のための HTTP 変数の指定

HTTP 変数は、アプリケーションリクエストの一部として Enterprise JavaBeans (EJB) などのアプリケーションコンポーネントに渡すことができます。その結果、開発者はリクエストに関する特定の情報を判断し、リクエストの処理時にその情報を使えます。

たとえば、アプリケーションは HTTP_REFERER 変数を参照して、リクエスト元を判断します。この情報を使って、目的に応じたグリーティング画面を表示したり、リクエスト元の統計を保存したりできます。

HTTP 変数は、レジストリのエントリを編集して管理します。必要に応じて HTTP 変数の有効 / 無効を切り替えることができます。iPlanet Web Server のデフォルトの HTTP 変数は、次のとおりです。

HTTPS	HTTP_USER_DEFINED
AUTH_USER	HTTPS_KEYSIZE
CLIENT_CERT	HTTPS_SECRETKEYSIZE
CONTENT_LENGTH	PATH_INFO
CONTENT_TYPE	PATH_TRANSLATED
HOST	QUERY
HTTP_ACCEPT	QUERY_STRING
HTTP_ACCEPT_CHARSET	REMOTE_ADDR
HTTP_ACCEPT_ENCODING	REMOTE_HOST
HTTP_ACCEPT_LANGUAGE	REMOTE_IDENT
HTTP_AUTHORIZATION	REMOTE_USER
HTTP_CONNECTION	REQUEST_METHOD
HTTP_COOKIE	SCRIPT_NAME
HTTP_HOST	SERVER_PORT
HTTP_IF_MODIFIED_SINCE	SERVER_PROTOCOL
HTTP_REFERER	SERVER_SOFTWARE
HTTP_USER_AGENT	SERVER_URL

アプリケーションコンポーネントへの入力として HTTP 変数を指定するには、次の手順を実行します。

1. コマンドラインに「kregedit」と入力し、iPlanet レジストリエディタを開きます。
エディタが開き、iPlanet Application Server に適用されるキーと値が表示されます。Web サーバと iPlanet Application Server が別のマシンにインストールされている場合は、エディタが開いて、Web コネクタプラグインに適用するキーと値が表示されます。
2. 該当するキーを開きます。
 - iPlanet Web サーバの場合は次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application  
Server¥6.5¥CCSO¥HTTPAPI¥INPUTNSAPI
```


- Microsoft Web サーバの場合は次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application
Server¥6.5¥CCSO¥HTTPAPI¥INPUTISAPI
```

- Apache Web サーバの場合は次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application
Server¥6.5¥CCSO¥HTTPAPI¥INPUTAPACHE
```

表示された各値は HTTP 変数を表します。この値によって、HTTP 変数がアプリケーションリクエストとともに iPlanet Application Server に渡されるかどうかが決まります。値が 0 以外の場合に、HTTP 変数がアプリケーションリクエストとともに iPlanet Application Server マシンに渡されます。

HTTP 変数は HTTP_REFERER のようにすべて大文字です。

3. HTTP 変数名である名前を追加します。
4. 新しい HTTP 変数 (名前) をダブルクリックし、値データとして次のどちらかの値を入力します。
 - HTTP 変数を無効にする場合は「0」を入力する
 - HTTP 変数を有効にする場合は「1」を入力する

注 デフォルトの HTTP 変数を無効にするには、その HTTP 変数名を追加してから、その値を 0 に設定します。たとえば、ENTITY_HEADER を追加してその値を 1 に設定し、HTTP_REFERER (デフォルトの HTTP 変数) を追加してその値を 0 に設定すると無効になります。

5. 「OK」をクリックします。
6. 追加 / 有効 / 無効にする HTTP 変数ごとに、手順 4 ~ 6 を繰り返します。
7. エディタを閉じます。変更を有効にするには、Web Server を再起動する必要があります。

アプリケーションコンポーネントへの入力のための HTTP 変数の指定

データベース接続の管理

iPlanet Application Server アプリケーションから、データベースにアクセスし、データを追加、取得、および変更することができます。この章では、データアクセスドライバを設定し、設定値をデータベース接続パラメータに適用する方法について説明します。

この章には次のトピックがあります。

- データアクセスドライバについて
- サードパーティ JDBC ドライバの設定
- サードパーティ JDBC ドライバのレジストリ設定
- データソースの設定

データアクセスドライバについて

iPlanet Application Server アプリケーションでは、頻繁にデータベースにアクセスする必要があります。データベースにはデータアクセスドライバを介してアクセスします。データアクセスドライバは、データベースベンダーまたはサードパーティベンダーによって作成されたソフトウェアです。次のタイプのデータアクセスドライバを iPlanet Application Server に設定し、データベースに接続できます。

- Oracle
- DB2
- Informix
- Sybase
- SQL Server (Windows)
- ODBC

-
- 注
- グローバルトランザクションは、ネイティブ JDBC ドライバではサポートされていません。グローバルトランザクションを使用するには、サードパーティ JDBC ドライバを設定する必要があります。

iPlanet Application Server 6.0 SP1 では、ネイティブドライバは廃止されています。ネイティブ JDBC ドライバのサポートは、下位互換性を維持する目的に限定されています。
 - データベースドライバでトランザクションをサポートするためには、JDBC 2.0 with Extensions の仕様に準拠するサードパーティ JDBC ドライバを使用する必要があります。
-

iPlanet Application Server をインストールする前に、データアクセスドライバをインストールしてください。インストールしたドライバは、iPlanet Application Server によって自動的に設定されます。

JDBC 2.0 Extensions 仕様に準拠する サードパーティ JDBC ドライバはすべて iPlanet Application Server でサポートされています。JDBC 2.0 Extensions フレームワークに準拠していないデータベースドライバは、iPlanet Application Server では動作しません。

サードパーティ JDBC ドライバの設定

サードパーティ JDBC ドライバは、アプリケーションサーバのインストール時か、またはインストール後に登録ツールを使って iPlanet に識別させる必要があります。登録は、サードパーティ JDBC ドライバデータソースを使うアプリケーションをハウジングする各アプリケーションサーバインスタンスで行う必要があります。たとえば、iPlanet Application Server の 2 つのノードクラスタを設定し、クラスタ内の両方のノードでアプリケーションを利用できるようにする場合は、各インスタンスでサードパーティ JDBC ドライバを登録する必要があります (既存の iPlanet Type 2 ドライバには同じ必要条件が適用されます)。

この節では次のトピックについて説明します。

- インストール時
- インストール後
- サードパーティ JDBC ドライバの設定
- コマンドラインツールを使ってサードパーティ JDBC ドライバを設定する
- サードパーティ JDBC ドライバのレジストリ設定

インストール時

サードパーティ JDBC ドライバの設定は、カスタムインストールオプションでのみ行うことができます。高速または標準インストールを使う場合は、アプリケーションサーバのインストール後にサードパーティ JDBC ドライバを設定する方法について、次の節を参照してください。

アプリケーションサーバのカスタムインストール時に、iPlanet Type 2 またはサードパーティ JDBC を設定するか、あるいは JDBC ドライバを使わないように選択できます。インストール時には Type 2 ドライバまたはサードパーティ JDBC ドライバのどちらかしか設定できませんが、インストール後に両方のドライバを設定するように選択できます。

インストール後

インストール後に iPlanet Application Server Administration Tool を使ってサードパーティ JDBC ドライバを設定します。

iPlanet はサポートされているネイティブクライアントライブラリの存在を自動的に認識するので、サポートされているデータベースプラットフォームへの iPlanet Type 2 JDBC ドライバの登録は自動的に行われます。

インストール後にデータベースドライバを登録する方法については、『iPlanet Application Server 管理者ガイド』を参照してください。

サードパーティ JDBC ドライバの設定

JDBC は、Javasoftware 仕様の標準 API で、Java プログラムから Database Management Systems (DBMS) にアクセスするときに利用します。JDBC API は、次の手順の実行に使用する、いくつかのインターフェイスとクラスから構成されています。

- データベースに接続するアプリケーションおよびアプレットの作成
- SQL (Structured Query Language) で書き込まれたクエリの送信
- 結果の処理

iPlanet Application Server をインストールするときに、ネイティブおよびサードパーティの JDBC ドライバを設定できます。インストール後にサードパーティの JDBC ドライバを設定する場合は、iPlanet Application Server Administration Tool を使用します。

コマンドラインツールを使用してサードパーティドライバを設定する必要がある場合は、「コマンドラインツールを使ってサードパーティ JDBC ドライバを設定する」を参照してください。

動作確認済みのサードパーティドライバ

次の表は、iPlanet Application Server 6.5 で動作することが確認されているサードパーティ JDBC ドライバを示します。

表 8-1 Solaris および Windows で動作確認済みのサードパーティドライバ

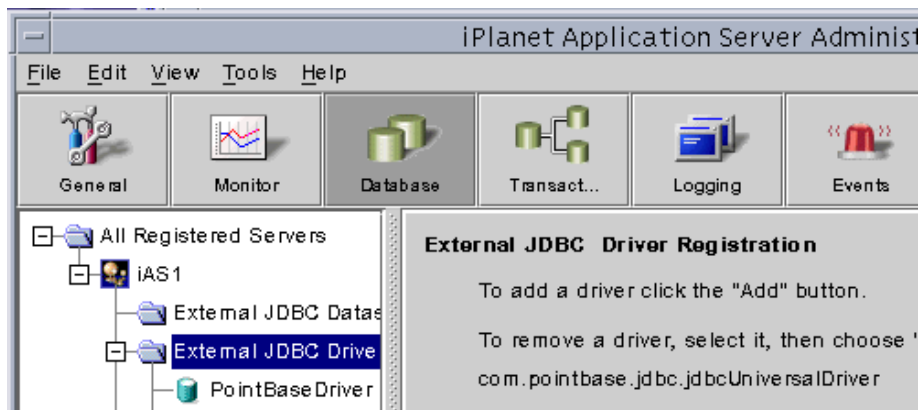
ドライバ	プラットフォーム
Oracle 8.1.7 および 8i Type 2 および Type 4 (thin)	Solaris および Windows
Merant (Intersolv) SequeLink DataDirect Java 5.1	Solaris および Windows (DDJ5.1 がサポートするすべてのデータベース)
Sybase jConnect for JDBC 5.2 Type 4	Solaris および Windows
DB2 JDBC 7.1 client	Solaris および Windows
Access to Microsoft SQL Server	Windows の場合のみ Merant から使用可能
Informix JDBC v2.1 Type 4	Solaris および Windows

サードパーティ JDBC ドライバをセットアップするには

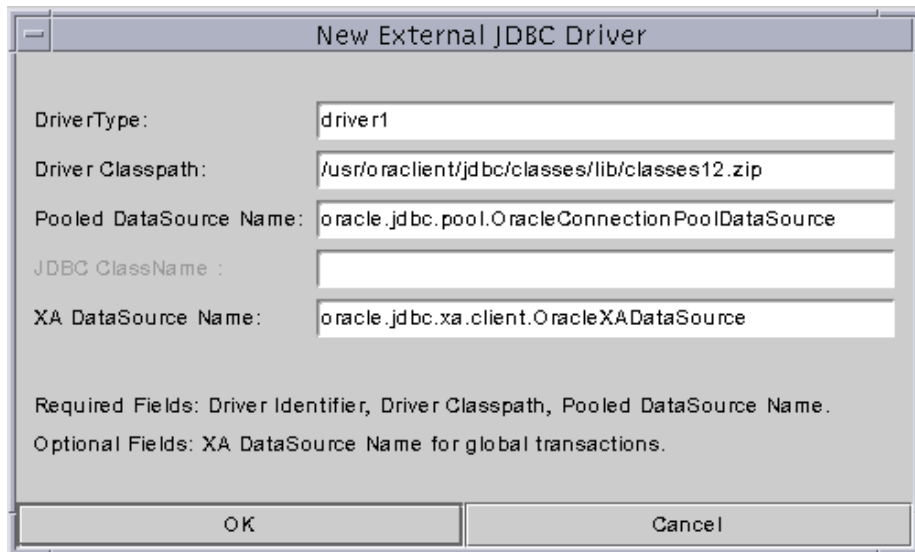
Solaris および Windows でサードパーティ JDBC ドライバを設定するには、次の手順を実行します。

1. iPlanet Application Server Administration Tool を起動します。
 - Solaris の場合は、`iasInstallDir/ias/bin` に移動します。


```
./ksvradmin
```
 - Windows では、「スタート」>「iPlanet Application Server 6.5」>「iAS Administration Tool」を選択します。
Administration Tool が起動します。
2. メニューオプションから「データベース」を選択します。
3. 登録済み iPlanet Application Server の隣の「+」記号をクリックして、すべてのデータベースオプションを表示します。



4. 「外部 JDBC ドライバ」を選択し、「追加」をクリックします。
次のダイアログボックスが表示されます。



5. 設定するドライバのドライバ名を入力します。たとえば、「driver1」のように入力します。

このドライバ名は、iPlanet Application Server にドライバを識別させるための論理名です。この名前はデータソースの定義を物理ドライバタイプにリンクさせるときに使います。この名前には、ユーザが選択した任意の文字列値を指定できません。たとえば、「driver1」、「ora-type4」、「ora-type2」、「jconnect」などです。

6. ドライバのクラスパスを入力します。たとえば、「Oracle_Home/jdbc/lib/classes12.zip」のように入力します。

ドライバクラス、JAR、または ZIP ファイルへの完全修飾パスです。この zip ファイルには、ドライバのライブラリクラスが含まれています。次の例のように、絶対パスを指定します。

```
usr/oraclient/jdbc/classes/lib/classes12.zip
```

Windows では、ドライバのクラスパスは D:¥orant¥jdbc¥lib¥classes12.zip のようになります。

次の表は一般的な Solaris および Windows の CLASSPATH を示しています。

表 8-2 一般的な Solaris および Windows の CLASSPATH

ドライバ	Solaris の CLASSPATH	Windows の CLASSPATH
Oracle Type 2 および 4	/oraclient/jdbc/lib/classes12.zip	D:¥orant¥jdbc¥lib¥classes12.zip
SequeLink Java 5	/sljc/driver/lib/sljc.jar	C:¥sljc¥driver¥lib¥sljc.jar
Sybase jConnect 5.2	/jConnect-5_2/classes/jconn2.jar	D:¥jConnect-5_2¥classes¥jconn2.jar
DB2	/DB2DIR/java/db2java.zip	D:¥SQLLIB¥java¥db2java.zip
Informix	/lib/ifxjdbc.jar	D:¥lib¥ifxjdbc.jar

7. プールされたデータソースのクラス名を入力します。たとえば、「oracle.jdbc.pool.OracleConnectionPoolDataSource」のように入力します。

注 このプールされたデータソースクラス名が `com.iplanet.ias.jdbc.IASConnectionPoolDataSource` の場合は、ドライバに固有の JDBC ドライバクラス名を指定する必要があります。たとえば、「`com.pointbase.jdbc.jdbcUniversalDriver`」のように指定します。

このオプションは、iPlanet Application Server に付属する PointBase などのドライバマネージャをサポートするドライバでのみ必要です。

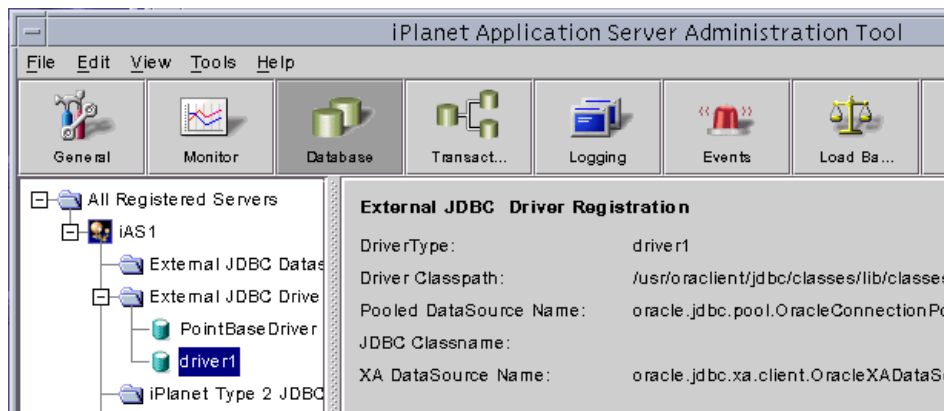
iPlanet Application Server では、そのようなドライバでプールされた接続をサポートするためのラッパーを提供しています。このラッパーの名前は、`com.iplanet.ias.jdbc.iASConnectionPoolDataSource` です。

8. XA データソースのクラス名を入力します。たとえば「`oracle.jdbc.xa.client.OracleXADataSource`」のように入力します。

注 これはオプションであり、グローバルトランザクションを使用する場合に指定する必要があります。

9. 「OK」をクリックしてドライバ登録プロセスを完了します。

Administration Tool の「外部 JDBC ドライバ」の下に新しい JDBC ドライバが表示されます。



JDBC ドライバの設定を変更するには

Administration Tool でドライバのプロパティを変更するには、次の手順を実行します。

1. Administration Tool の左側のペインで、変更するドライバを選択します。
2. 右側のペインで「変更」をクリックします。
3. 必要な変更を加えます。
4. 「OK」をクリックして変更を確定します。

JDBC ドライバの削除

Administration Tool でドライバを削除するには、次の手順を実行します。

1. Administration Tool の左側のペインで、削除するドライバを選択します。
2. 「編集」 > 「削除」をクリックします。
3. 削除の確認を求められたら、「はい」をクリックします。

コマンドラインツールを使ってサードパーティ JDBC ドライバを設定する

次の節では、サードパーティ JDBC ドライバを Solaris マシンおよび Windows マシンに設定する方法について説明します。

- Solaris で `db_setup.sh` を使ってサードパーティ JDBC ドライバを設定するには
- Windows で `jdbcsetup.exe` を使ってサードパーティ JDBC ドライバを設定するには

注 Windows の `dbsetup.exe` ユーティリティは廃止され、iPlanet Application Server のインストールに含まれなくなりました。

Solaris で `db_setup.sh` を使ってサードパーティ JDBC ドライバを設定するには

サードパーティ JDBC ドライバを設定するには、次の手順を実行します。

1. `<iasinstall>/ias/bin` に移動します。次のスクリプトを実行します。

```
db_setup.sh
```

2. インストールするドライバを選択します。次のオプションが表示されます。
 1. iPlanet Type 2 JDBC Drivers
 2. Third Party JDBC Drivers
 ネイティブ JDBC ドライバを設定するときは「2」を入力して、Enter キーを押します。
3. 使用しているマシンで、iPlanet Application Server がインストールされているディレクトリを指定します。たとえば、「/usr/iplanet/ias6」と指定します。Enter キーを押します。
4. 設定するドライバ数を指定し、Enter キーを押します。
設定するデータベースクライアントは、同一である必要はありません。
5. 設定するドライバのドライバ名を入力します。たとえば、「driver1」のように入力します。

このドライバ名は、iPlanet Application Server にドライバを識別させるための論理名です。この名前はデータソースの定義を物理ドライバタイプにリンクさせるときに使います。この名前には、ユーザが選択した任意の文字列値を指定できます。たとえば、「driver1」、「ora-type4」、「ora-type2」、「jconnect」などです。

6. ドライバのクラスパスを入力します。たとえば、「Oracle_Home/jdbc/lib/classes12.zip」のように入力します。
ドライバクラス、JAR、または ZIP ファイルへの完全修飾パスです。この zip ファイルには、ドライバのライブラリクラスが含まれています。次の例のように、絶対パスを指定します。

```
usr/oraclient/jdbc/classes/lib/classes12.zip
```

一般的な Solaris のクラスパスについては、表 8-2 を参照してください。

7. プールされたデータソースのクラス名を入力します。たとえば、「oracle.jdbc.pool.OracleConnectionPoolDataSource」のように入力します。

注 このプールされたデータソースクラス名が `com.iplanet.ias.jdbc.IASConnectionPoolDataSource` の場合は、ドライバに固有の JDBC ドライバクラス名を指定する必要があります。たとえば、「com.pointbase.jdbc.jdbcUniversalDriver」のように指定します。

8. XA データソースのクラス名を入力します。たとえば「oracle.jdbc.xa.client.OracleXADataSource」のように入力します。

注 これはオプションであり、グローバルトランザクションを使用する場合に指定する必要があります。

9. インストール後、*iASInstallDir/ias/env/iasenv.ksh* ファイルにドライバの環境変数を設定するように要求されます。

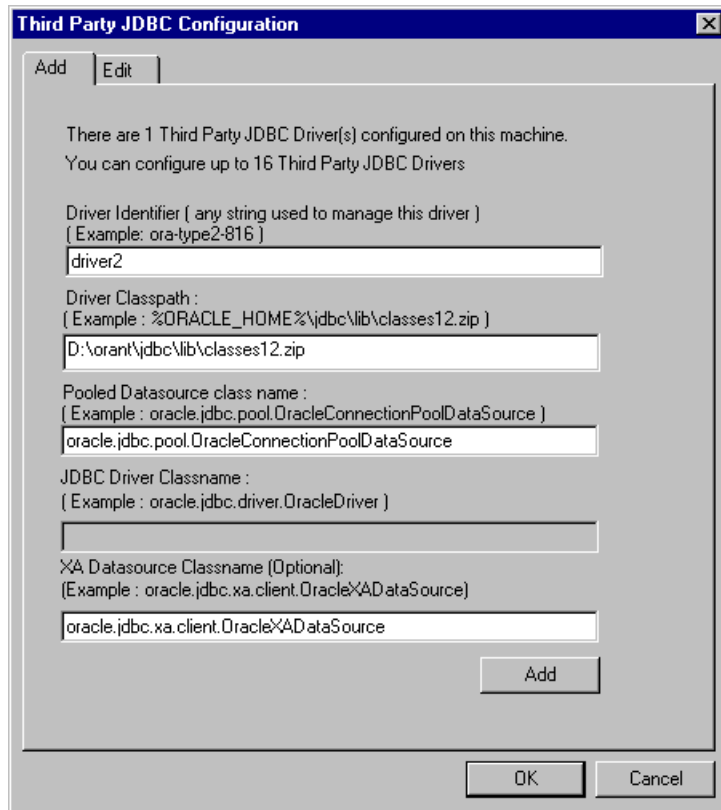
Windows で *jdbcsetup.exe* を使ってサードパーティ JDBC ドライバを設定するには

ネイティブまたはサードパーティの JDBC ドライバを Windows システムに設定するには、次の手順を実行します。

1. *iASInstallDir/ias6/ias/bin* に移動します。次のスクリプトを実行します。

```
jdbcsetup.exe.
```

次のウィンドウが表示されます。



2. ドライバのドライバ識別子を入力します。たとえば、「driver2」のように入力します。

このドライバ名は、iPlanet Application Server にドライバを識別させるための論理名です。この名前はデータソースの定義を物理ドライバタイプにリンクさせるときに使います。この名前には、ユーザが選択した任意の文字列値を指定できます。たとえば、「driver2」、「ora-type4」、「ora-type2」、「jconnect」などです。

3. ドライバのクラスパスを入力します。たとえば、「Oracle_Home/jdbc/lib/classes12.zip」のように入力します。

ドライバのクラスパスは、ドライバクラス、JAR、または ZIP ファイルへの完全修飾パスです。この zip ファイルには、ドライバのライブラリクラスが含まれています。次の例のように、絶対パスを指定します。

d:\oracle\jdbc\lib\classes12.zip

注 ネイティブ JDBC ドライバを設定している場合、このクラスパスは指定しません。

iPlanet Application Server はサポートされているネイティブクライアントライブラリの存在を自動的に認識するので、サポートされているデータベースプラットフォームへの iPlanet Type 2 JDBC ドライバの登録は自動的に行われます。

一般的なクラスパスの例については、表 8-2 を参照してください。

4. プールされたデータソースのクラス名を入力します。たとえば、「oracle.jdbc.pool.OracleConnectionPoolDataSource」のように入力します。

注 このプールされたデータソースクラス名が com.iplanet.ias.jdbc.IASConnectionPoolDataSource の場合は、ドライバに固有の JDBC ドライバクラス名を指定する必要があります。たとえば、「com.pointbase.jdbc.jdbcUniversalDriver」のように指定します。

5. XA データソースのクラス名を入力します。たとえば「Database.Jdbc.xa.client.DatabaseXADatasource」のように入力します。データベースが Oracle の場合は、「oracle.xa.jdbc.client.OracleConnectionXADatasource」と入力します。
6. 「Add」をクリックしてドライバを登録します。
さらに別のドライバを設定することも、あるいは「Cancel」をクリックして処理を中止することもできます。

「OK」をクリックして変更を確定します。

サードパーティ JDBC ドライバのレジストリ設定

iPlanet でサードパーティ JDBC ドライバを登録するとき、次の iPlanet レジストリ設定が表示されます。

ドライバのエントリ

```
SOFTWARE\iPlanet\Application Server\6.5\CCS0\DAE3\DRIVERS<driver name>
```

このキーの下にドライバのクラス名があります。

レジストリキー設定の詳細については、『iPlanet Application Server レジストリガイド』を参照してください。

データソースの設定

データソースには、データベースに接続するときデータベースドライバに必要な、データベースクライアントに関する情報が含まれています。iPlanet Application Server 6.5 では、Datasource は JDBC コネクションを取得するためのファクトリクラスです。iPlanet ネイティブの JDBC ドライバとサードパーティの JDBC ドライバ、およびサードパーティドライバ実装の PooledDataSource と XADatasource をサポートします。

データソースは、ドライバマネージャのみのサポートを行うドライバに、プールされたコネクションをサポートするためのラッパーを提供します。

注 データソースを追加する前に、まず、マシンにデータベースドライバをインストールして設定する必要があります。

この節には次のトピックがあります。

- 新しいデータソースの機能
- 外部 JDBC ドライバにデータソースを追加するには
- ネイティブデータベースドライバにデータソースを追加するには
- Deployment Tool を使ってデータソースを追加するには
- コマンドラインツールを使ってデータソースを追加するには
- データソース設定パラメータ
- サードパーティデータソースの変更

新しいデータソースの機能

iPlanet Application Server 6.5 で使用可能なデータソースの新機能は次のとおりです。

- トランザクションで使用する JDBC コネクションのためのコネクションの共有、およびユーザに返されるコネクションの妥当性検査を実行するメカニズムを提供します。プールが破損すると、自動的に破棄されて新しいプールが作成されます。
- テーブルベースの妥当性 (オプション)
- 次に示すようないくつかのデータソースプロパティの動的設定をサポートします。

- sanityRequired
- tableBasedSanity
- tableName

注 iPlanet Application Server 6.0 SPx ベースのデータソース XML フォーマットは廃止されています。古いデータソース XML ファイルを新しいファイルに移行するには、Deployment Tool を使用します。

データソース XML ファイルを 6.5 に移行する方法については、「6.x の配置記述子 XML ファイルを移行するには」を参照してください。

データソースの登録

J2EE JNDI、コネクションプール、およびローカルトランザクションとグローバルトランザクションの管理機能を利用する場合の iPlanet Application Server でのデータソーススペースのアクセスにはサードパーティ JDBC ドライバを使用します。

注 iPlanet による JDBC 統合を行わずにサードパーティ JDBC ドライバを使う必要がある場合は、アプリケーションにドライバを手動で読み込み (通常 DriverManager クラスを使って行う)、正しいコネクション設定をセットアップできるようにします。

正しい CLASSPATH とライブラリパス設定を使ってアプリケーションサーバのランタイム環境を設定します。

アプリケーションサーバでサードパーティ JDBC ドライバを登録したあと、アプリケーションがデータベース管理システムと対話できるように JDBC データソースを定義する必要があります。

この節では、iPlanet Application Server でサードパーティまたはネイティブの JDBC ドライバに関連付けられている JDBC データソースを登録するときに必要な情報について説明します。iPlanet で JDBC データソースを登録すると、Administration Tool からコネクションプール設定を変更できます。

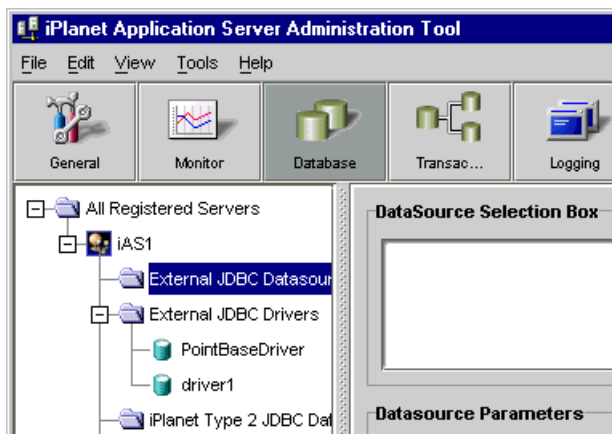
次の節では、外部のサードパーティおよびネイティブの JDBC データベースドライバにデータソースを追加する方法について説明します。

- 外部 JDBC ドライバにデータソースを追加するには
- ネイティブデータベースドライバにデータソースを追加するには

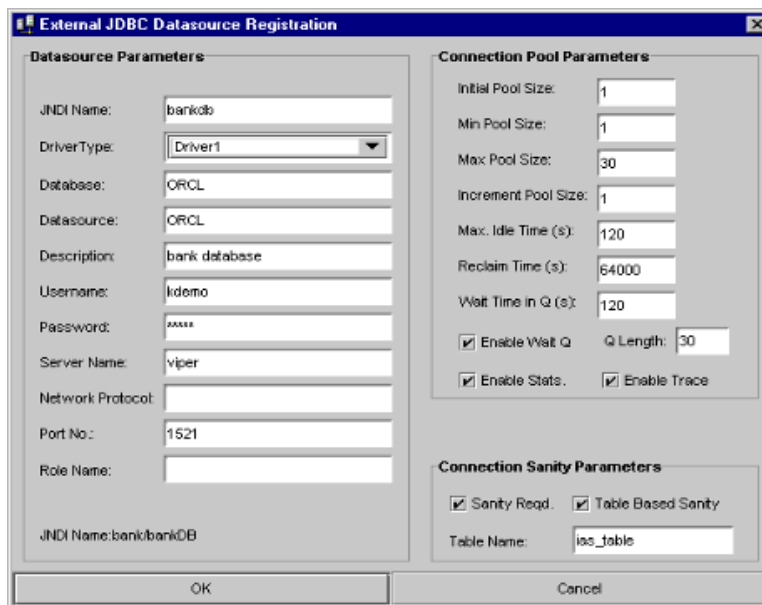
外部 JDBC ドライバにデータソースを追加するには

Type 4 JDBC ドライバなどの外部 JDBC ドライバにデータソースを追加するには、次の手順を実行します。

1. iASAT を起動します。
2. iASAT ツールバーの「データベース」をクリックします。
マシンに設定されているすべてのデータベース接続が表示されます。
3. 次の図のように、「外部 JDBC データソース」ノードを選択します。



4. ウィンドウの右側のペインで「追加」をクリックします。次の図のように「外部 JDBC データソース登録」ウィンドウが表示されます。



データソースを登録すると、この名前で iASAT にノードが作成されます。

○ JNDI 名

「JNDI 名」はアプリケーションサーバの JNDI ネーム空間内でデータソースを一意に識別します。たとえば、dbc/estore/EstoreDB という JNDI 名は J2EE アプリケーションの ias-web.xml および ias-ejb-jar.xml ファイルの <resource-ref> エントリ内で参照されます。

<resource-ref> エントリは、J2EE アプリケーションに用いられるリソース名をアプリケーションサーバ内で定義された JNDI 名にマッピングします。

注 JNDI 名は、追加するデータソースごとに一意の名前を指定する必要があります。

○ ドライバのタイプ

driver-type はサードパーティ JDBC ドライバに割り当てられた論理名にマッピングされます。

○ ドライバ名

iPlanet にドライバを識別させるための論理名。この名前はデータソースの定義を物理ドライバタイプにリンクさせるときに使います。この名前には、ユーザが選択した任意の文字列値を指定できます。たとえば、「ora-type4」、「ora-type2」、「jconnect」などです。

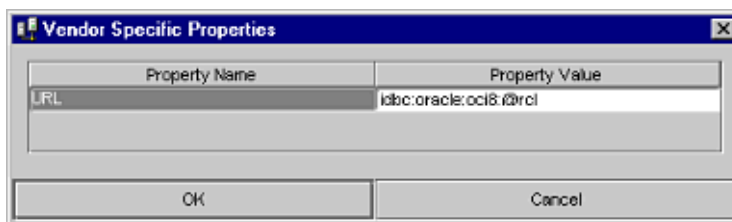
○ ユーザ名

iPlanet がデータベースに接続すると、ユーザ名が RDBMS に渡されます。ユーザ名とパスワードはプログラム上で、またはコネクション URL 上で指定できるので、省略可能です。

○ パスワード

iPlanet Application Server がデータベースに接続すると、パスワードが RDBMS に渡されます。

5. 「ベンダー固有のプロパティ」では、JDBC ドライバのベンダー固有のデータを指定します。たとえば、「プロパティ名」に「URL」を指定し、「プロパティ値」をデータベース URL 形式で指定します。



6. 「OK」をクリックしてベンダーのプロパティ値を確定します。
 7. メイン画面で「OK」をクリックしてデータソースの登録を完了します。

ネイティブデータベースドライバにデータソースを追加するには

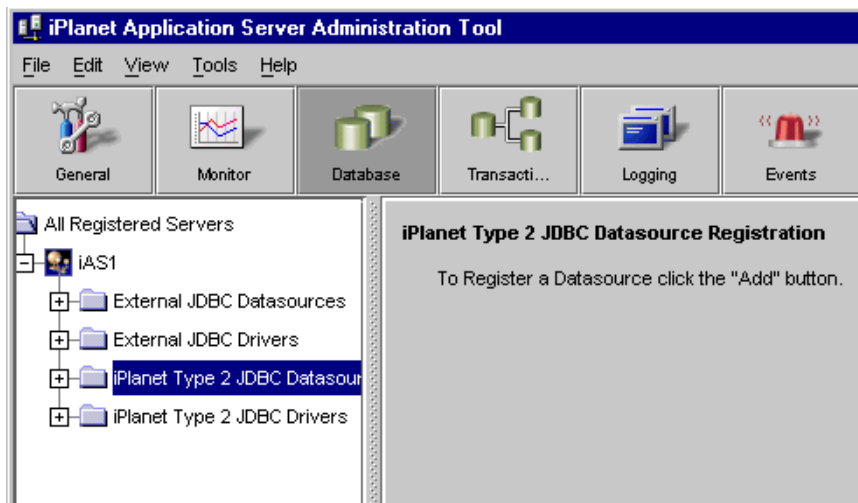
データソースを、Type 2 JDBC ドライバなどのネイティブデータベースドライバに追加するには、次の手順を実行します。

1. iASAT を起動します。Windows では、「スタート」メニューから「プログラム」>「iPlanet Application Server 6,5」>iPlanet Application Server Administration Tool を選択します。

Solaris では、<iASInstallDir>/ias/bin/ に移動して、次のコマンドを入力します。

```
kvsradmin
```

2. iASAT ツールバーの「データベース」をクリックします。
3. ウィンドウの左側のペインで、データソースを追加する iPlanet Application Server インスタンスを選択します。マシンに設定されているすべてのデータベース接続が表示されます。
4. 次の図のように、「iPlanet Type 2 JDBC データソース」フォルダを選択します。



5. データソースを追加するには、「データベース」ウィンドウの右側のペインで「追加」をクリックします。

次の図のように、「iPlanet Type 2 JDBC データソース登録」ウィンドウが表示されます。

The screenshot shows a dialog box titled "iPlanet Type 2 JDBC Datasource Registration". It contains several input fields and a dropdown menu. The fields are filled with the following values: JNDI Name: bankDB, DriverType: ORACLE_OCI, DataSource: orcl, DataBase: ora1, Username: admin, Password: admin. Below the fields, there is a summary section that reads: "JNDI Name: bank/bankDB" and "Datasource: orcl(sqlnet, connect string for oracle)". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

6. 「JNDI名」フィールドに、データソースの名前を入力します。たとえば、「BankDB」と入力します。データソースを登録すると、この名前でノードが作成されます。

注 JNDI名は、追加するデータソースごとに一意の名前を指定する必要があります。

7. 「ドライバのタイプ」ドロップダウンリストで、データソースの追加先ドライバを選択します。

注 ネイティブドライバの場合は、マシンにデータベースクライアントをインストールする必要があります。

8. 「データソース」フィールドに、データソースの追加先データベースクライアント依存の関連する値を入力します。データベースとドライバは、この値を使って接続されます。次の表は、iPlanet Application Server がサポートする、データベースタイプの値を示しています。

表 8-3 データソースの値のリスト

データベースタイプ	「データソース」フィールドの値
Oracle	Oracle データベースの TNS 名。この値は、ORACLE_HOME/network/admin ディレクトリの tnsnames.ora ファイルに格納される

表 8-3 データソースの値のリスト (続き)

データベースタイプ	「データソース」フィールドの値
Sybase	Sybase クライアントがインストールされているサーバの名前
DB2	任意の文字列値。データベースに接続するときは、この文字列値を、ユーザ名およびパスワードとともに入力する必要がある。
Informix	このフィールドに値を入力する必要はない。ログインすると、Informix クライアントによって自動的にデータソースが選択される

9. 「データベース」フィールドに、データベースの名前を入力します。たとえば、「Oracle」と入力します。Sybase の場合は、複数の Sybase サーバに接続するために、データベース名の前にサーバ名を付ける必要があります。
10. 「ユーザ名」フィールドに、データベースクライアントをインストールしたときに入力したユーザ名を入力します。
11. 「パスワード」フィールドに、そのユーザ名に対応するパスワードを入力します。
12. 「OK」をクリックしてデータソースを登録します。変更を有効にするには、iPlanet Application Server を停止して再起動する必要があります。iPlanet Application Server の開始と停止の詳細については、「iASAT による管理タスクの実行」を参照してください。

Deployment Tool を使ってデータソースを追加するには

Deployment Tool を使ってデータソースを登録するには、次の手順を実行します。

1. Deployment Tool (deploytool) を起動します。
2. 「ツール」 > 「データソースを登録」を選択します。
次のダイアログボックスが表示されます。

図 8-1 Deployment Tool によるデータソースの登録

3. データベースドライバパラメータ、接続プールパラメータ、および接続妥当性パラメータを入力します。

これらのフィールドの詳細については、「データソース設定パラメータ」を参照してください。

4. 「名前を付けて保存」をクリックします。
新しいデータソース配置記述子の名前を入力します。
5. 「登録」をクリックします。
6. 「Register」を選択して、1つまたは複数のアプリケーションサーバインスタンスを持つデータソースを登録します。

6.x の配置記述子 XML ファイルを移行するには

iPlanet Application Server 6.5 では、データベース接続インフラストラクチャの機能が拡張されています。新しい機能を利用し、かつ以前のデータソース設定を使用し続けるには、新しいデータソース XML DTD への移行を行う必要があります。

1. 「ツール」 > 「データソースを登録」を選択します。
データソース登録ダイアログが表示されます。
2. 「開く」をクリックし、データソース記述子を格納している XML ファイルを選択します。
Deployment Tool は、古い XML ファイルの値をインポートし、6.5 の XML ファイルで使用可能な新しいフィールドにデフォルト値を追加します。
3. そのデータソースに対し、デフォルト値を受け入れるか、またはデフォルト値を変更します。
4. データベースドライバパラメータ、コネクションプールパラメータ、およびコネクション妥当性パラメータを入力します。
これらのフィールドの詳細については、「データソース設定パラメータ」を参照してください。
5. 「Save」をクリックします。
Deployment Tool は、開いているファイルのデータソース配置記述子を、指定した値で上書きします。
注：更新した XML ファイルを別の名前での別の場所に保存する場合は、「名前を付けて保存」をクリックします。
6. 「登録」をクリックします。
Deployment Tool は、指定した値でデータソース配置記述子を更新します。これで、このデータソースで iPlanet Application Server 6.5 の新機能を利用できます。

コマンドラインツールを使ってデータソースを追加するには

データソースを登録するには、コマンドラインツールの `iasdeploy` および `regdatasource` を使用します。

コマンドラインでデータソースを登録するには、`iasdeploy` ツールをお勧めします。その例を次に示します。

```
iasdeploy resreg EstoreDB.xml
```

ローカルサーバに登録する場合は、`resreg` コマンドを使うことができます。

```
resreg EstoreDB.xml
```


データソース設定パラメータ

これらのパラメータは、JDBC 2.0 仕様に基づいています。これらのパラメータで提供されるプロパティの大部分はオプションです。

次の表は、データソースプロパティの画面に示される各種パラメータを示します。

- データソースパラメータ
- コネクションプールパラメータ
- コネクション妥当性パラメータ
- データソースレジストリ設定

データソースパラメータ

表 8-4 データソース設定パラメータ

設定パラメータ	説明	デフォルト値
ドライバのタイプ	ドライバのタイプ (iAS に登録されているドライバの1つ)	なし
データベース	サーバ上のデータベースの名前	なし
データソース	サーバ上のデータソースの名前	なし
説明	このデータソースの説明	なし
ユーザ名	ユーザのアカウント名	なし
パスワード	パスワード	なし
serverName	データベースサーバ名	なし
NetworkProtocol	サーバとの通信に使うネットワークプロトコル	なし
ポート番号	サーバがリクエストを受けるポート番号	なし
ロール名	初期 SQL ロール名	なし

コネクションプールパラメータ

設定パラメータ	説明	デフォルト値
初期プールサイズ	プール作成時に取得するコネクションの数	1

設定パラメータ	説明	デフォルト値
最小プールサイズ	<p>maxIdleTime の設定に関係なく、プールが保持しなければならない物理コネクションの数</p> <p>プールが minPoolSize に達すると、minPoolSize の条件が適用される</p> <p>minPoolSize の条件はプールが minPoolSize の数に達したときのみ適用されるので、initialPoolSize を minPoolSize よりも小さくすることも可能</p>	1
最大プールサイズ	<p>プールに格納する物理コネクションの最大数</p> <p>プールが maxPoolSize に達すると、waitQueue が有効になっている場合は、コネクションリクエストはキューに入れられる。waitQueue が無効の場合は、SQLException がスローされる</p>	30
増分プールサイズ	<p>プールおよびユーザリクエストのコネクションで使用できる空きコネクションがない場合に取得する物理コネクションの数</p> <p>プールがすでに maxPoolSize に達している場合は、SQLException がスローされる</p>	1
最大アイドル時間	<p>物理コネクションがプールで使用されない状態が続くときに、そのコネクションを閉じるまでの時間 (秒単位)</p> <p>この条件は、プール内で使用されていない、または空きコネクションの数が minPoolSize よりも多い場合にのみ適用される</p>	120
回復時間	アプリケーションが使用していたコネクションをプールが要求し直すまでの時間 (秒単位)	64000

設定パラメータ	説明	デフォルト値
キューでの待機時間	コネクションがデータベースで使用可能になるまでプールがキューにコネクションを保持する時間 (秒単位)	120
待機キューを有効にする	コネクションが使用可能になるまでプールが待機するように設定する	True (有効)
キューの長さ	キューに保持されるコネクションの数	30
統計コレクションを有効にする	コネクションプールに関する統計コレクションを有効にする	Disable
トレースを有効にする	コネクションプールプロパティのトレースを有効にする	Disable

コネクション妥当性パラメータ

表 8-5 データソースコネクション妥当性パラメータ

設定パラメータ	説明	デフォルト値
TableBasedSanity	テーブルベースの妥当性を有効にする (true/false)	False
IsSanityRequired	妥当性を有効にする場合に使用する (true/false)	True
TableName	テーブルベースの妥当性を有効にする場合に使用するテーブル名	ias_table

すべてのプール設定パラメータを動的に変更できるだけでなく、複数のデータソースのプール設定パラメータを同時に変更することもできます。データソースプロパティの変更の詳細については、205 ページの「サードパーティデータソースの変更」を参照してください。

このダイアログボックスのパラメータの詳細は、JDBC 2.0 Extensions 仕様を参照してください。

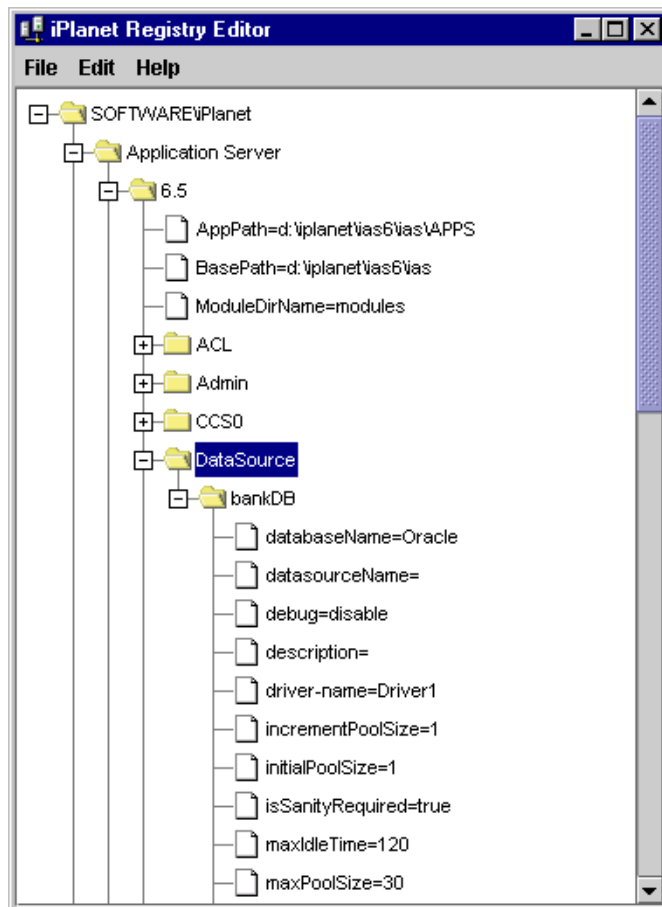
次の表に、データベース URL フォーマットの例をいくつか示します。

表 8-6 データベース URL フォーマット

ドライバ	クラス名
Oracle Type 2 および 4	jdbc:oracle:oci8:@MyHostString jdbc:oracle:thin:@myhost:1521:orcl jdbc:oracle:thin:scott/tiger@myhost:1521:orcl jdbc:oracle:oci8:scott/tiger@myhost
SequeLink Java 5	jdbc:sequelink://sequelinkhost:19996 jdbc:sequelink://192.168.5.96:19996;user=john;password=whatever jdbc:sequelink://192.168.5.96:19996;databaseName=stores7 jdbc:sequelink://192.168.5.96:19996;databaseName=pubs;H User=john;HPassword=whatever jdbc:sequelink://sequelinkhost:4006;databaseName=pubs;D BUser=john;DBPassword=whatever jdbc:sequelink:ssl://mysecurehost:9500;cipherSuites=SSL_D H_anon_WITH_3DES_EDE_CBC_SHA jdbc:sequelink:ssl://mysecurehost:9502;
Sybase jConnect 5.2	jdbc:sybase:Tds:host:port
DB2	db2:jdbc:dbname
Informix	jdbc:informix-sqli://mysecurehost:1528/database:informixse rver=my_server
PointBase 3.5	jdbc:pointbase://localhost:9092/j2eeguide, max.connections=100

データソースレジストリ設定

サードパーティ JDBC ドライバに関連付けられているデータソースを登録するとき、`SOFTWARE\iPlanet\Application Server\6.5\DataSource<jndi-name>`の下に次の iPlanet レジストリ設定が表示されます。



サードパーティデータソースの変更

登録済みデータソースに関連付けられた各種パラメータを変更するには iASAT を使用します。各データソースについて、次のパラメータを変更できます。

- データソースパラメータ
- コネクション妥当性パラメータ
- コネクションプールパラメータ

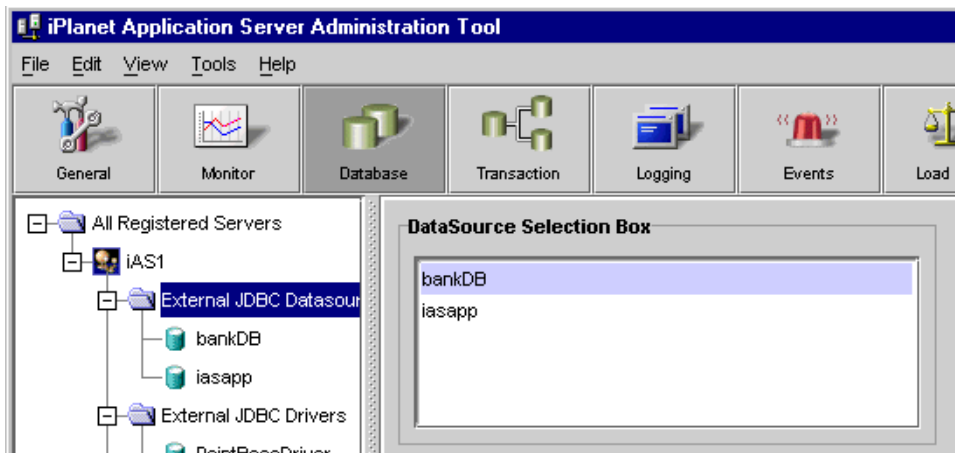
一度に1つのデータソースを変更するか、複数のデータソースを同時に変更するかを選択できます。一度に複数のデータソースを選択して変更すると、選択したデータソースのグローバルコネクションプールパラメータを設定できます。

次の節では、一度に1つまたは複数のデータソースを変更する方法について説明します。

1つのデータソースを変更するには

1つのデータソースを変更するには、次の手順を実行します。

1. iASAT メニューバーから「データベース」を選択します。
2. 左側のペインで「外部 JDBC データソース」を選択します。



3. 「データソース選択ボックス」で変更するデータソースを選択します。
4. 右側のペインで使用可能なプロパティを変更できます。

5. 変更が完了したら、「変更の適用」をクリックして変更を保存します。

複数のデータソースを変更するには

複数のデータソースを変更すると、複数のデータソースのグローバルコネクションプールパラメータを設定できます。

複数のデータソースを変更するには、次の手順を実行します。

1. 206 ページの「1 つのデータソースを変更するには」の最初の 2 つの手順を実行します。
2. Shift キーを押しながら、変更するデータソースを選択します。
「データソースパラメータ」の項目がグレー表示になります。

複数のデータソースを選択するときは、「コネクション妥当性パラメータ」と「コネクションプールパラメータ」のみを変更できます。

注 複数のデータソースを選択すると、「コネクションプールパラメータ」の値がデフォルト値に戻されます。

3. 選択したデータソースに共通する「コネクションプールパラメータ」に必要な変更を加えます。
4. 変更作業が完了したら、「変更の適用」をクリックして変更を保存します。

データソースの監視

iASATには、すべてのアクティブコネクションプールの統計情報を収集する機能があります。

特定のデータソースの統計情報を表示するには、iASATの左側のペインでデータソースを選択します。複数のKJSが設定されている場合は、「エンジンID」ドロップダウンボックスから、統計情報を表示するKJSを選択します。

Select kjs Engine

Engine Id:

Connection Pool Statistics

Cumulative Statistics

Peak Value for Total Conns in Pool:

Peak Value for Connections in Use:

Peak Value for Queue Size:

Cache Hits:

Cache Misses:

Total Connection Requests:

Total Connection Dropped:

Connections Obtained After Wait:

Connections Dropped After Wait:

Connections Reclaimed:

Connections Released After Idle Time:

Current Statistics

Total Connections in Pool:

Connections in Use:

Waiting Queue Size:

管理者は、特定のデータソース用に作成されたコネクショナルプールの監視、設定、および管理を行うことができます。Administration Tool で、プールの統計を有効または無効にできます。メインのデータソースパネルからだけでなく、この画面でも統計を有効にできます。

プールには 2 種類の統計があります。

- 現在の統計
- 累積統計

現在の統計

「現在の統計」で監視されるパラメータは次のとおりです。

- プール内のコネクショナル総数
- 使用中のコネクショナル
- 待機中キューのサイズ

選択したデータソースに関する上記の値は、統計を有効にしているかどうかに関係なく、Administration Tool に表示されます。

累積統計

「累積統計」で監視されるパラメータは次のとおりです。

- プール内の接続総数のピーク値
 - 使用中接続のピーク値
 - キューサイズのピーク値
- キャッシュのヒット数
- キャッシュのミス数
- 接続リクエスト総数
- ドロップした接続総数
- 待機後に取得した接続
- 待機後にドロップした接続
- 接続が回復しました
- アイドル時間後に解放された接続

上記のパラメータの値は、統計を有効にすると Administration Tool に表示されます。各パラメータに表示される値は、そのデータソース用に作成されたすべてのプールの累積値です。

詳細な統計情報

「詳細」ボタンをクリックすると、各プールに関する上記すべてのパラメータの詳細を表示できます。このオプションを使用すると、上記のパラメータだけでなく、プール名とプールタイプ (XA、Normal) も表示されます。

Administration Tool で使用できるその他のオプションは次のとおりです。

- プールを破壊
- プールをフラッシュ

プールを破壊

「プールを破壊」ボタンを選択すると、接続プールを破壊できます。「プールを破壊」オプションは、プールからすべての接続を削除します。

プールをフラッシュ

「プールをフラッシュ」ボタンを選択すると、接続プールをフラッシュできます。「プールをフラッシュ」オプションは、プールからすべてのアイドル接続を削除します。

データソースを削除するには

データソースを削除するには、次の手順を実行します。

1. iPlanet Application Server Administration Tool で、削除するデータソースを選択します。
2. キーボードの Delete キーを押します。
3. 確認を求められたら「はい」をクリックします。

Administration Tool からデータソースが削除されると、対応するデータソースオブジェクトも初期コンテキストから削除されます。

データソース XML フォーマット

iPlanet Application Server 6.5 で提供される新しい XML フォーマットは、柔軟性があり、標準の JDBC データソースプロパティおよびドライバ固有のデータソースプロパティをサポートします。iPlanet Application Server 6.0 SPx のデータソース登録で使用されていた古い XML フォーマットは、自動的に新しい XML フォーマットに変換されます。変換された XML ファイルは、*xmlfilename.new* として保存されます。

新しい XML データソースフォーマットの例を次に示します。

コード例 8-1 新しいデータソース XML フォーマット

```
<ias-resource>
<resource>
<jndi-name>jdbc/dsName</jndi-name>
<jdbc>
<user>admin</user>
<password>admin</password>
<database-url>jdbc:oracle:oci8:@rcl</database-url>
<driver-name>oracle_xa</driver-name>
```

ポート番号の「<port>1050</port>」などの追加データはここに追加します。

```
<conn-pooling>
<initialPoolSize>1</initialPoolSize>
<waitQueueEnabled>>true</waitQueueEnabled>
<reclaimTime>1200</reclaimTime>
```

```
<maxPoolSize>abc</maxPoolSize>  
<maxIdleTime>120</maxIdleTime>  
<queueLength>20</queueLength>  
<trace>false</trace>  
<stat>false</stat>  
<waitTimeInQueue>120</waitTimeInQueue>  
<incrementPoolSize>1</incrementPoolSize>  
<minPoolSize>5</minPoolSize>  
</conn-pooling>  
</jdbc>  
</resource>  
</ias-resource>
```

ほかのデータベースベンダーの例は、『iPlanet Application Server 開発者ガイド』に記載されています。

トランザクションの管理

この章では、iPlanet Application Server Administration Tool (iASAT) を使ったトランザクション管理に必要な作業と概念について説明します。

この章には次のトピックがあります。

- トランザクションマネージャについて
- iASAT からの分散トランザクションの管理
- トランザクションの監視

トランザクションマネージャについて

トランザクションマネージャは、iPlanet Application Server の各インスタンスとともにインストールされ、Java サーバ (KJS) プロセス内のグローバルトランザクションを調整します。グローバルトランザクションは、関連処理の集合で、単位として実行する必要があります。ただし、各処理は同一プロセス内で実行する必要はありません。

グローバルトランザクションでは、が同一グローバルトランザクションに対して、1 つまたは複数の EJB (Enterprise Java Beans) を使用している場合でも、1 つまたは複数の KJS プロセスからデータベースを更新することができます。この現象は、EJB が別の EJB をトリガし、これらの EJB が同じトランザクションを使用している場合に発生します。さまざまな場所に分散されている複数のデータベースを更新したり、さまざまなタイプ (Oracle や Sybase など) の複数のデータベースを更新したりできます。トランザクションマネージャは、KJS プロセス内で動作します。

新しい機能

iPlanet Application Server Enterprise Edition 6.5 リリース以降では、バンドルされているトランザクションマネージャは、運用環境向けに変更が加えられた Java Transaction Service Reference Implementation (JTS RI) に基づいています。

ローカルトランザクションを使用するアプリケーションを最適化するためにローカルトランザクションマネージャも追加されています。

データソースマネージャの機能が拡張されており、XADataSource、PooledDataSource、および DriverManager をサポートします。XA コネクションとプールされたコネクションのプーリングをサポートするための新しいコネクションプーリングインフラストラクチャが実装されています。

トランザクションマネージャに追加された新しい機能は次のとおりです。

- グローバルトランザクションとローカルトランザクションを同時に実行できる柔軟性
- 次の3つのレベルでトランザクションタイプを指定できるユーザに対するプロビジョン
 - インスタンス全体
 - パッケージレベル (J2EE-Module/ パッケージ名にある)
 - コンポーネントレベル

iPlanet Application Server は、まず、パッケージ名の下を参照し、次にコンポーネント <classdef¥<guid>、最後にレジストリ内の <CCSO¥TXNMGR> にあるインスタンス全体の値を参照することによって特定のコンポーネントまたはアプリケーションのトランザクションタイプを決定します。

ヒント コンポーネントレベルでトランザクションタイプを指定すると、パフォーマンスを向上させることができます。

- リカバリのサポート

iPlanet Application Server が起動すると、レジストリに登録されているデータソースが使用するすべてのデータベースから保留中のトランザクションを検索し、リカバリを実行します。リカバリ機能を無効にすることもできます。

新しいトランザクションマネージャでは、管理制御、監視、および追跡の機能が改善されています。iPlanet Application Server Administration Tool は、管理および監視の機能をサポートするために大幅に拡張されています。

iASAT からの分散トランザクションの管理

iPlanet Application Server Administration Tool の「トランザクション」メニューオプションを使ってトランザクションを管理できます。

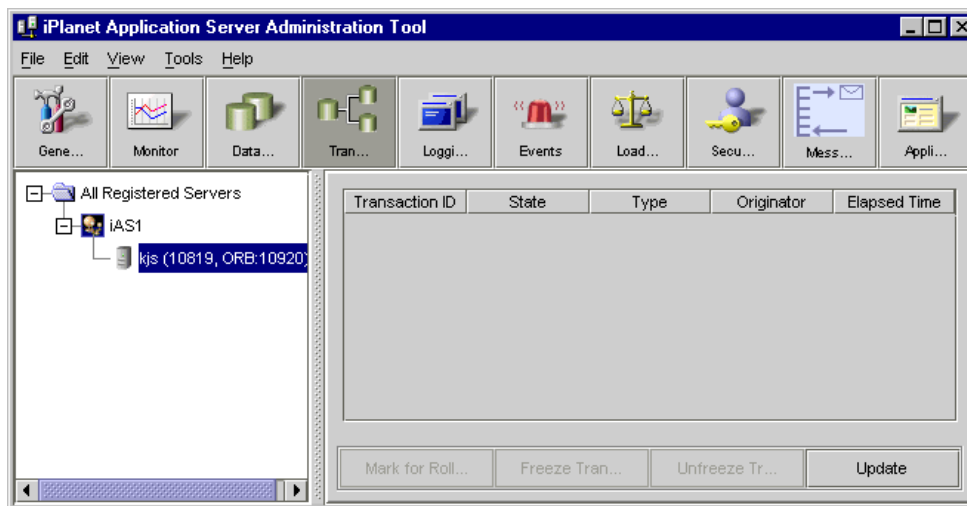
この節では次のトピックについて説明します。

- 「トランザクション」ウィンドウについて
- サーバごとにトランザクションを表示する
- サーバごとにトランザクションを設定する
- プロセスごとにトランザクションを管理する
- 選択したプロセス上のトランザクションを表示する
- プロセスごとにトランザクションを管理する

「トランザクション」ウィンドウについて

「トランザクション」ウィンドウの左側のペインには、次の図のようなノードツリーが表示されます。

図 9-1 トランザクションマネージャの「設定」タブ



ツリーの最上位レベルには、iASAT に登録されているサーバが一覧表示されます。2 番目のレベル、つまり各登録済みサーバ名の下には、1 つまたは複数のプロセスノードが表示されます。ノードには、登録済みサーバ上で実行されているプロセスが表示されます。トランザクションは KJS プロセスだけでサポートされるため、ツリーには KJS プロセスだけが表示されます。

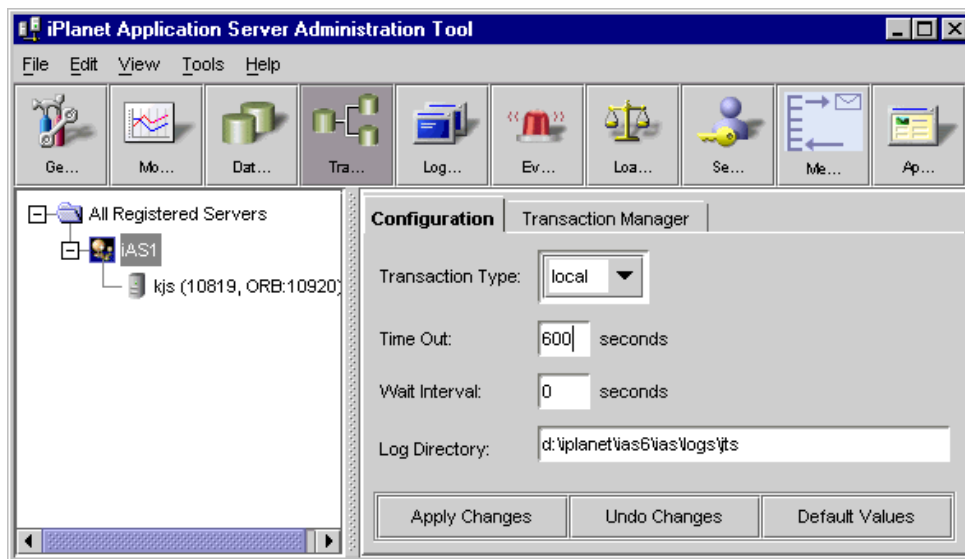
KJS プロセスは、ポート番号 (10819 など)、および JMS リスナへの接続に使用する ORB ポート (10920 など) とともに表示されます。

サーバごとにトランザクションを設定する

iPlanet Application Server インスタンスを選択してメニューバーの「トランザクション」をクリックすると、そのサーバインスタンスのグローバルプロパティが表示されます。すべての KJS インスタンスは、エンジン ID やポート番号などの識別情報以外は iPlanet Application Server のプロパティを共有します。したがって、この「設定」ダイアログボックスで変更を行うと、そのサーバ上で動作するすべてのトランザクションに影響します。

アプリケーションサーバのトランザクション設定を変更するには、「トランザクション」ウィンドウの左側のペインで登録済みサーバをクリックします。次のような「設定」タブが右側のペインに表示されます。

図 9-2 サーバインスタンスのトランザクションプロパティの表示

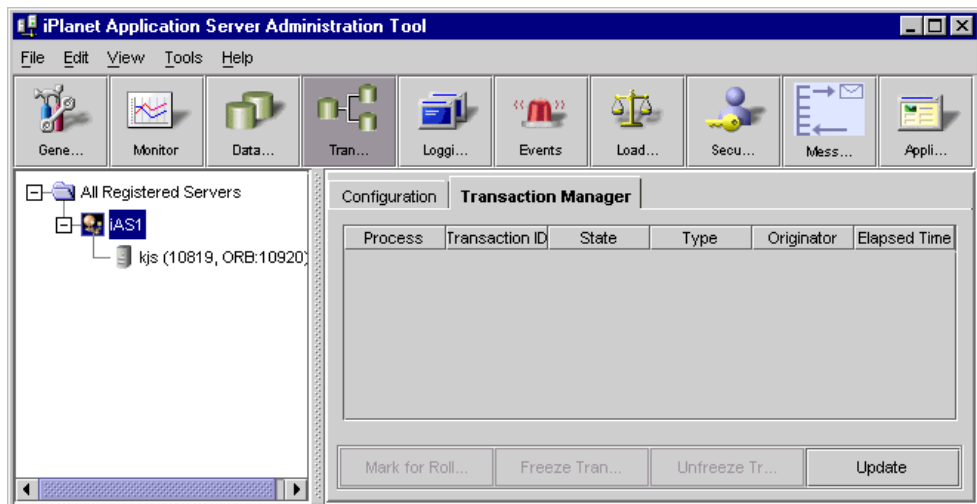


設定できるオプションは次のとおりです。

- トランザクションタイプ
トランザクションタイプをローカルまたはグローバルに設定できます。
グローバルトランザクションを有効にすると、複数の異なるデータベースとプロセスにわたってトランザクションを分散できます。
- タイムアウト
規定された時間内にトランザクションが完了されなかった場合にロールバックするまでの時間 (秒単位) を設定します。これは、トランザクションプロセス (たとえばステートフルセッションビーンなど) が開始されたが、アプリケーション設計にバグがあったために完了できなかった場合に役立ちます。
- 待機間隔
「待機間隔」には、iPlanet Application Server が動かなくなったトランザクションや破壊したトランザクションをクリアするまでに待機する時間 (秒単位) を指定します。
- ログディレクトリ
トランザクションマネージャのログファイルを保存する場所を示すパスを指定します。デフォルトは `iASInstallDir/ias/logs/jts` です。

サーバごとにトランザクションを表示する

トランザクションの詳細を表示するには、「トランザクションマネージャ」タブをクリックします。「トランザクションマネージャ」ダイアログボックスに、進行中の各種トランザクションの状態に関する詳細情報が表示されます。



詳細表示の項目は次のとおりです。

- プロセス
「プロセス」タブには、トランザクションが属する KJS エンジンが表示されます。
- トランザクション ID
トランザクション ID は、コネクションプールマネージャによってトランザクションに割り当てられます。
- 状態
「prepared」、「running」、「rollbacked」、「committing」の 4 つの状態が表示されます。
- タイプ
トランザクションがローカルかグローバルかを示します。
- 作成者
「作成者」は、トランザクションを生成した Servlet または Bean を示します。
- 経過時間
トランザクションが開始されてからの時間を示します。

ヒント 表の見出しをクリックすると、そのデータをソートできます。

サーバごとにトランザクションを管理する

一度に複数のエントリを選択してアクティブなトランザクションを制御できます。進行中のトランザクションを制御するためのオプションは、ダイアログボックスのいちばん下にあります。

- ロールバックのマーク

「ロールバックのマーク」オプションを使うと、アクティブなトランザクションをロールバックできます。目的のトランザクションを選択し、「ロールバックのマーク」をクリックします。

Shift キーを押しながらトランザクションを選択し、「ロールバックのマーク」をクリックすることにより、複数のトランザクションを一度にロールバックすることもできます。

注 トランザクションをロールバックするときに、「トランザクションが存在しません」というメッセージが表示されることがあります。これは、コマンドが実行される前にそのトランザクションが完了してしまうことがあるためです。

- 更新

「更新」ボタンを定期的をクリックして、期限切れのトランザクションをビューから削除し、現在実行されているトランザクションをウィンドウに表示してください。

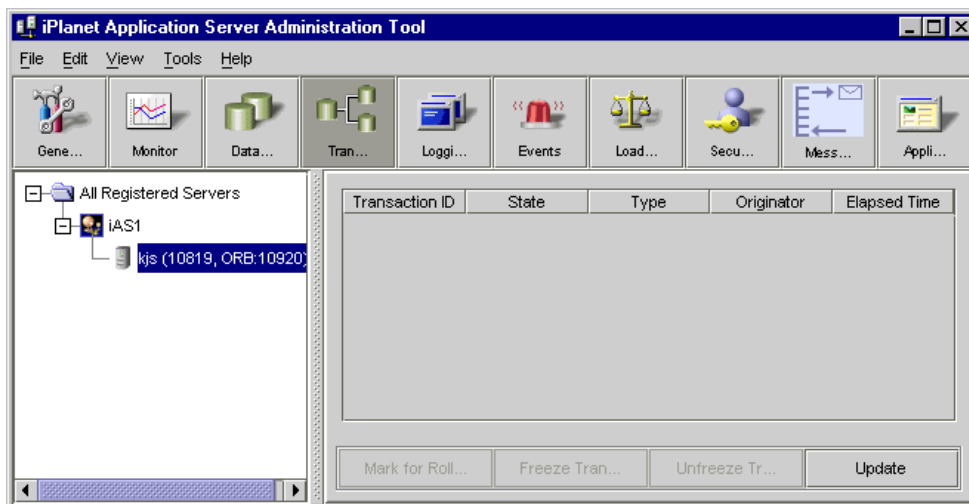
プロセスごとにトランザクションを管理する

目的の iPlanet Application Server インスタンスの下で稼動している特定の KJS エンジンに接続されているトランザクションプロセスを個別に管理できます。このオプションを使うと、iPlanet Application Server で稼動するトランザクションを細かく制御できます。

選択したプロセス上のトランザクションを表示する

特定の KJS エンジンに関するトランザクションの状態を表示するには、左側のペインで目的のエンジンを選択します。そのエンジンで動作するトランザクションが右側のペインに表示されます。

図 9-3 特定の KJS エンジンのトランザクションプロパティの表示



詳細表示の項目は次のとおりです。

- トランザクション ID
トランザクション ID は、コネクションプールマネージャによってトランザクションに割り当てられます。
- 状態
「prepared」、「running」、「rollbacked」、「committing」の 4 つの状態が表示されます。
- タイプ
トランザクションがローカルかグローバルかを示します。
- 作成者
「作成者」は、トランザクションを生成した Servlet または Bean を示します。
- 経過時間
トランザクションが開始されてからの時間を示します。

ヒント 表の見出しをクリックすると、そのデータをソートできます。

プロセスごとにトランザクションを管理する

左側のペインでアプリケーションサーバインスタンスを選択し、メニューバーの「トランザクション」アイコンをクリックします。220 ページの図 9-3 のようなダイアログボックスが表示されます。

一度に複数のエントリを選択してアクティブなトランザクションを制御できます。進行中のトランザクションを制御するためのオプションは、ダイアログボックスのいちばん下にあります。

- ロールバックのマーク

「ロールバックのマーク」オプションを使うと、アクティブなトランザクションがロールバックできます。目的のトランザクションを選択し、「ロールバックのマーク」をクリックします。

複数のトランザクションを選択して「ロールバックのマーク」をクリックすることにより、複数のトランザクションを一度にロールバックすることもできます。

- トランザクションを凍結

「トランザクションを凍結」オプションを使うと、すべてのアクティブトランザクションが一時的に停止します。すべてのトランザクションを停止するかどうかを確認するプロンプトが表示されます。

停止する場合は「OK」をクリックし、凍結オプションを中止する場合は「キャンセル」をクリックします。

「トランザクションを凍結」オプションを使用すると、新しいトランザクションは開始されず、既存のトランザクションは次の論理的な段階で凍結されます。

- トランザクションを凍結解除

「トランザクションを凍結解除」オプションを使うと、凍結コマンドによって停止されていたすべてのトランザクションが再開されます。

確認のためのプロンプトが表示されます。既存のすべてのトランザクションを凍結解除する場合は「OK」をクリックします。

- 更新

「更新」ボタンを定期的をクリックして、期限切れのトランザクションをビューから削除し、現在実行されているトランザクションをウィンドウに表示してください。

トランザクションの監視

フレームワークの監視によって、モジュールに関する各種のプロパティを表示できます。これによりシステムの内部情報を表示することで、アプリケーションパラメータを微調整したり、ボトルネックを検出したり、アプリケーションの健全な動作を確認したりできます。

JTS は、より高度なトランザクション監視機能を提供します。Administration Tool によってすべての結果が表示されます。JTS が提供する機能は次のとおりです。

- すべてのアクティブトランザクションを次の情報とともに表示
 - トランザクション ID
 - トランザクションを起動したコンポーネント
 - 状態 (prepared、running、rolledback、committing)
 - トランザクションタイプ (グローバル / ローカル)
 - 経過時間
- ローカルトランザクションとグローバルトランザクションの TPM を表示
 サンプルタイムの値が表示され、ユーザはこれを編集できます。このサンプルタイムは TPM の計算に使用され、その TPM が表示されます。総数のほかに、ローカルトランザクションの値とグローバルトランザクションの値もあります。
- 各トランザクション (グローバルとトランザクションの両方) にかかる平均時間を表示。サンプルタイムでは、トランザクション、ローカルトランザクション、およびグローバルトランザクションの平均時間を表示

設定パラメータ

トランザクションマネージャ用の設定パラメータは次のとおりです。

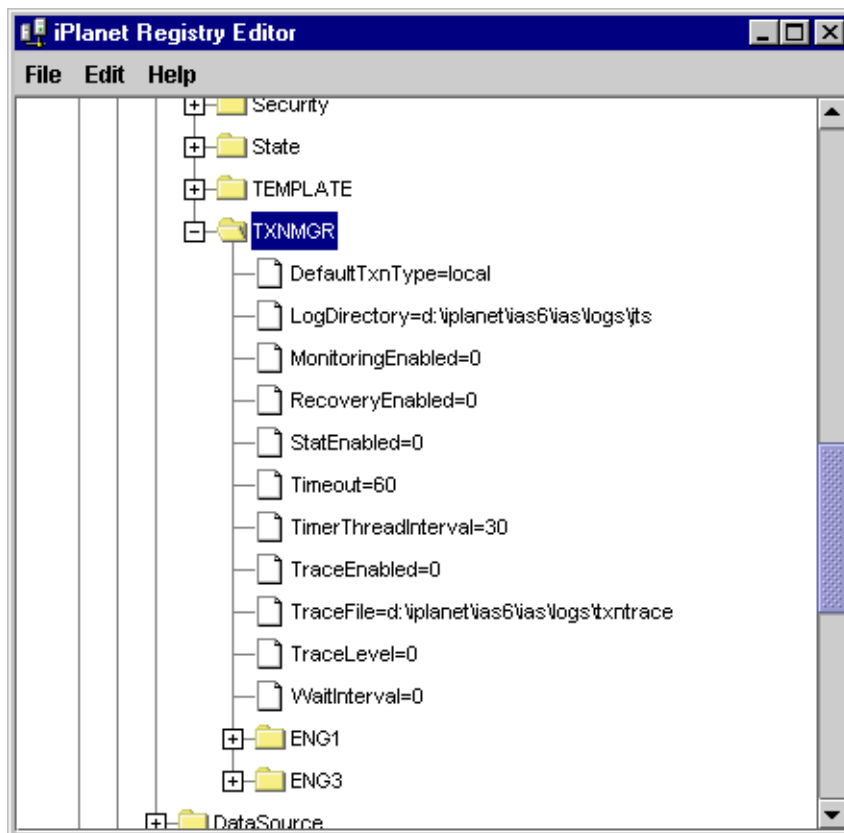
表 9-1 トランザクションマネージャの設定パラメータ

設定パラメータ	説明	デフォルト値
DefaultTxnType	このキーは、トランザクションタイプ (ローカル / グローバル) を特定するために使用される	local
LogDirectory	このディレクトリは JTS ログの保存に使用される	\$IAS_INSTALLDIR/ias/logs/jts

表 9-1 トランザクションマネージャの設定パラメータ (続き)

設定パラメータ	説明	デフォルト値
RecoveryEnabled	リカバリを有効にすると、KJS が起動したときに、保留中のトランザクションのリカバリが試行される (1 または 0、有効は 1、無効は 0)	0
Timeout	デフォルトのトランザクションタイムアウト (秒単位)	60
TimerThreadInterval	タイムアウトになったトランザクションを削除するためにタイムアウトスレッドを起動する間隔 (秒単位)	30
TraceEnabled	追跡を有効または無効にするために使用される (1 または 0 の値があり、有効は 1、無効は 0)	0
TraceFile	追跡データはこのファイルに書き込まれる	\$IAS_INSTALLDIR/ias/logs/txntrace
TraceLevel	追跡レベルの指定に使用される (1 ~ 6、1 が低レベルで 6 が高レベル)	0

これらのキーはすべて、レジストリの SOFTWARE/iPlanet/ApplicationServer/6.5/CCS0/TXNMGR の下にあります。



インダウトトランザクションの解決

Java サーバ (KJS) プロセスが突然終了したときなどには、ハングしたりインダウトトランザクションが発生したりすることがあります。Microsoft SQL Server の場合、インダウトトランザクションを手動で確定またはロールバックするために、DTC 管理者制御を使います。DTC 管理者制御は DAC とも呼ばれます。dac.exe は WINNT¥SYSTEM32¥ ディレクトリにあり、DTC とともにインストールされます。

DAC を起動したら、次の手順を実行してインダウトトランザクションを手動で確定またはロールバックします。

1. iASAT ツールバーで「トランザクション」をクリックして、「トランザクション」ウィンドウを開きます。

2. 「トランザクションマネージャ」タブをクリックします。
3. 強制的に確定またはロールバックするトランザクションを選択し、「詳細」をクリックします。
4. トランザクションを強制的にロールバックするには、「Resolve/Abort」をクリックします。

Oracle リソースマネージャの場合、「lock held by distributed transaction」というエラーが発生したら、データベースに接続し、グローバルトランザクションを明示的にロールバックする必要があります。グローバルトランザクションを明示的にロールバックするには、次の手順を実行します。

1. `dba_2pc_pending` を参照して、そのトランザクションに対応するローカルトランザクション ID を調べます。`dba_2pc_pending` には、保留中のグローバルトランザクションに関する詳細がすべて記録されています。

たとえば、SQLPLUS プロンプトに次の文字列を入力します。

```
SQLPLUS>select * from dba_2pc_pending
```

2. コマンドラインに次の文字列を入力してトランザクションをロールバックします。

```
rollback force transaction_id
```

Sybase リソースマネージャの場合、「lock held by distributed transaction」というエラーが発生したら、データベースに接続し、グローバルトランザクションを明示的にロールバックする必要があります。グローバルトランザクションを明示的にロールバックするには、次の手順を実行します。

1. `sp_xa_scan_xact` を実行してトランザクション ID リストを取得し、そのトランザクションに対応するローカルトランザクション ID を調べます。
2. トランザクション ID と `stat` (`commit` または `rollback`) を指定して `sp_finish_xact` を実行し、ブランチを強制終了させます。

CORBA ベースクライアントサポートの有効化

以前のバージョンの iPlanet Application Server では、RMI/IIOP プロトコルを介しての EJB へのアクセスには、組み込みの Javasoft ORB のみがサポートされていました。現在のバージョンの iPlanet Application Server では、Javasoft ORB だけでなく、IONA Technologies 製のサードパーティ ORB である ORBIX 2000 もサポートされます。つまり、CORBA 要素をサーバ側に提供するときに、組み込み Javasoft ORB または ORBIX 2000 を選択できるようになりました。

この節では、次のどちらかの ORB を使って EJB にアクセスする CORBA ベースクライアントのための RMI/IIOP サポートを設定する方法について説明します。

- iPlanet Application Server の組み込み Javasoft ORB
- ORBIX2000

この節では次のトピックについて説明します。

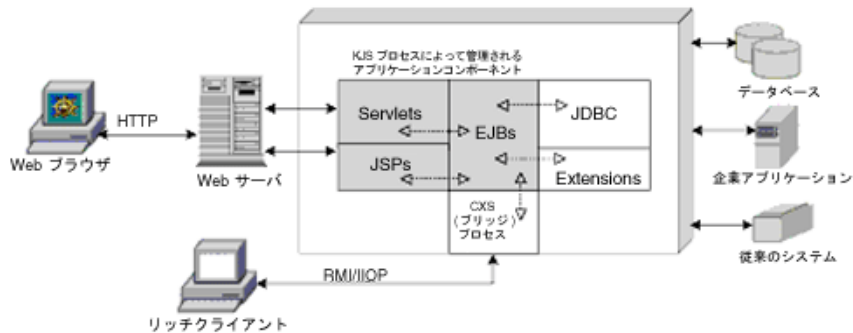
- Javasoft ORB サポートの設定
- ORBIX2000 サポートの設定

Javasoft ORB サポートの設定

CORBA ベースクライアントが Internet Inter-ORB Protocol (IIOP) を経由して、KJS プロセスによって管理される EJB と直接通信するには、iPlanet Application Server に CXS (Bridge) プロセスを追加する必要があります。通常、リクエストは、そのリクエストを出した Web ブラウザの Web パスを通じて行われ、JSP および Servlet によって処理され、さらに EJB にアクセスします。この Web パスでは HTTP プロトコルが使われます。CORBA ベースクライアントの場合、リクエストは、JavaSoft ORB を介し、CORBA Execute Server (CXS) を使って EJB に直接転送されます。CXS は、iPlanet Application Server 内で CORBA ベースクライアントと EJB との間でブリッジとして機能する Java エンジンです。CORBA ベースクライアントの開発および配置については、『開発者ガイド』を参照してください。

次の図は、iPlanet Application Server 内の通信アーキテクチャ、および CXS ブリッジプロセスを使った CORBA ベースクライアントと EJB サーバの通信を示しています。

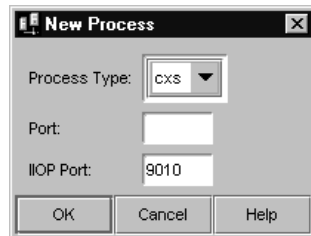
図 10-1 iPlanet Application Server 通信アーキテクチャ



CXS (ブリッジ) プロセスを追加するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、CXS プロセスを追加する iPlanet Application Server インスタンスを選択します。
3. 「ファイル」メニューから「新規」>「プロセス」をクリックします。

「新規プロセス」ダイアログボックスが表示されます。



4. 「処理タイプ」ドロップダウンボックスで「CXS」を選択します。
5. 「ポート」テキストボックスで、追加のプロセスを実行する未使用のポート番号を指定します。これは、内部 iPlanet Application Server エンジンポートです。

6. 「IIOP ポート」テキストボックスで、CORBA ベースクライアントが CXS との通信に使うポート番号を指定します。これは、CXS が CORBA ベースクライアントを待機するポートです。
7. 「OK」をクリックして、新しいプロセスを追加します。
8. このプロセスをシングルスレッド環境で使う場合は、次の手順を実行します。
 - a. 「一般」ウィンドウの左側のペインで必要なプロセスを選択します。
 - b. ウィンドウの右側のペインで、「最小スレッド」と「最大スレッド数」を1に設定します。
9. 「変更の適用」をクリックして変更を保存します。

ORBIX2000 サポートの設定

この節では、ORBIX 2000 サポートを利用するために iPlanet Application Server を設定する方法について説明します。この節では次のトピックについて説明します。

Solaris 上で ORBIX サポートを設定するには

Windows NT で ORBIX サポートを設定するには

Solaris 上で ORBIX サポートを設定するには

Solaris 上で ORBIX 2000 サポートを設定するには、次の手順を実行します。

1. ORBIX 2000 バージョン 1.2.1 をインストールします。
2. ライセンスファイルをマシンにコピーし、ライセンスをインストールします。ライセンスのインストールの詳細については、<http://www.ionas.com>にある ORBIX のマニュアルを参照してください。
3. ORBIX の設定ドメインを作成します。

ORBIX のデフォルトのドメイン設定は、localhost です。このドメイン設定は、iPlanet Application Server および CORBA 要素が同じマシン上で動作しているときに使うことができます。iPlanet Application Server と CORBA 要素が別々のマシン上で動作している場合は、新しい設定ドメインをファイルベースまたはリポジトリベースで作成する必要があります。ファイルベースでドメインを作成した場合は、サービスに関連する設定の詳細はすべて構成ファイルに保存されます。このファイルは、ドメインの作成に必要な手順を実行した後に作成されます。

詳細は、『ORBIX 管理者ガイド』を参照してください。

4. 次の手順で、セキュリティサービスのインターセプタをコンフィグレーションファイルに追加する必要があります。
 - a. 作成したコンフィグレーションファイルの `server_binding_list` に、サーバのインターセプタ名 `iASServerInterceptor` をグローバル・スコープで追加します。

たとえば、サーバのインターセプタ名を追加する前は、コンフィグレーションファイルの関連するエントリは次のようになっています。

```
binding:server_binding_list=[OTS]
```

`iASServerInterceptor` をサーバのインターセプタ名として追加すると、コンフィグレーションファイルのエントリは次のようになります。

```
binding:server_binding_list=[iASServerInterceptor, OTS]
```
 - b. インターセプタ `portable_interceptor` をコンフィグレーションファイルの `orb_plugins` エントリに追加します。

たとえば、インターセプタ `portable_interceptor` を `orb-plugins` エントリに追加する前は、エントリは次のようになっています。

```
orb_plugins=[iiop_profile,giop,iiop,ots]
```

`portable_interceptor` を移植可能なインターセプタ名として `orb-plugins` に追加すると、エントリは次のようになります。

```
orb_plugins=[iiop_profile,giop,iiop,ots,portable_interceptor]
```
5. 共通環境変数ファイル `iasenv.ksh` を変更して、マシン上の ORBIX 設定の場所を `iPlanet Application Server` に認識させます。
 - a. `iasenv.ksh` を開きます。ORBIX Configuration Settings のコメントの下にある ORBIX 設定のセクションに移動します。
 - b. 次の変数に、ORBIX のインストール先ディレクトリの名前を指定します。


```
IT_PRODUCT_DIR = <ORBIX product install directory>
```
 - c. 次の変数に、ORBIX のライセンスファイル `licenses.txt` をコピーしたディレクトリの名前を指定します。


```
IT_LICENSE_FILE_DIR= <path to ORBIX license file>
```
 - d. 次の変数に、ORBIX 設定ドメイン名を指定します。


```
ORBIX_DOMAIN_NAME=<configuration domain name>
```
 - e. 次の変数に、ORBIX 設定ドメインファイルへのパスを指定します。


```
ORBIX_CONFIG_DOMAINS_DIR = <configuration file path>
```
 - f. 次の変数に、CXS (Corba Executive Server) 間でロードバランサを有効にするかどうかを指定します。このロードバランサは、サーバサイドの RMI/IIOP プロセスです。

ORBIX_LOADBALANCING= true または false

CXS 間のロードバランサは、デフォルトで有効になっています。フラグを false に設定すると、この機能はオフになります。

- g. 次に示すように、`iasenv.ksh` の `JAVA_ARGS` 変数は、JavaSoft ORB が使われるように設定されています。

```
JAVA_ARGS=-Xss512k -Xms64m -Xmx64m
```

ORBIX を利用するには、次のように、この変数のコメントを解除する必要があります。

```
##JAVA_ARGS=(-Xss512k -Xms64m -Xmx64m)
-DBridge_ORB_CLASS=com.iona.corba.art.artimpl.ORBImpl
-DORBIXLoadBalancing=$ORBIX_LOADBALANCING
-Xbootclasspath/a:$IT_PRODUCT_DIR/orbix_art/1.2/classes/omg.jar
-DORBdomain_name=$ORBIX_DOMAIN_NAME
-DORBconfig_domains_dir=$ORBIX_CONFIG_DOMAINS_DIR
-Dorg.omg.PortableInterceptor.ORBInitializerClass.com.iplanet.ias.io
na.orbinitializers.IONAServerORBInitializerImpl=
```

ふたたび JavaSoft を利用する場合は、この変数をコメントに戻します。

6. `orb.properties` ファイル内の次のプロパティを変更します。このファイルは `iASInstallDir/ias/usr/java/jre/lib` にあります。

```
org.omg.CORBA.ORBSingletonClass=com.iona.corba.art.artimpl.ORBSingleton
```

7. ORBIX 環境変数を設定するために、コマンドラインから次のコマンドを実行します。各コマンドは、ピリオド (.) とそれに続くスペースで始めます。

```
. <orbix installation directory>/bin/<domain name>_env
. <orbix installation directory>/bin/<domain name>_java_env
```

たとえば、ドメイン名が「Marigold」の場合は、次のように指定します。

```
. <orbix installation directory>/bin/marigold_env
. <orbix installation directory>/bin/marigold_java_env
```

8. 次のコマンドを実行して、ORBIX サービスを開始します。

```
<orbix installation directory>/bin/start_<domain name>_services
```

たとえば、ドメイン名が「Marigold」の場合は、次のように指定します。

```
/space/iona/bin/start_marigold_services
```

9. 必要なアプリケーションを配置します。iPlanet Application Server を配置して再起動したときにスタブとタイが再生成されることを確認します。

CXS エンジンとは、登録が完了すると、クライアントサイドオブジェクトグループのオブジェクトの処理を開始します。RMI/IIOP ORB は、リクエストを CXS エンジンに送信します。CXS エンジンが停止している場合でも、リクエストを送信し続けます。この現象を回避するには、登録した CXS エンジンがすべて動作してなければなりません。通常、KAS (Administrative Server) は、障害が発生したサーバエンジンプロセスを再起動します。ただし、CXS エンジンが無効になっているか、削除されているか、起動していない (KAS が起動していない場合に発生する) 場合は、CXS エンジンのオブジェクトをすべてのオブジェクトグループから削除することをお勧めします。CXS エンジンのオブジェクトを削除するには、以下のコマンドを使います。

CXS エンジンによって処理されている現在のオブジェクトグループを一覧表示するには

```
itadmin nsog list
```

現在のメンバーを一覧表示するには

```
itadmin nsog list_members <Group Name>
```

特定のメンバーを削除するには (メンバー名はエンジン番号と同じ)

```
itadmin nsog remove <GroupName>
```

または

```
itadmin nsog remove_member -og_name <GroupName> <engine no>
```

削除を確認するには

```
itadmin nsog list_members <Group Name>
```

<group name> は ejb-name + Group です。

Windows NT で ORBIX サポートを設定するには

Windows NT で ORBIX サポートを設定するには、次の手順を実行します。

1. ORBIX 2000 バージョン 1.2.1 をインストールします。
2. ライセンスファイルをマシンにコピーし、ライセンスをインストールします。詳細は、<http://www.ionapl.com>にある ORBIX のマニュアルを参照してください。

3. iPlanet レジストリキーを次のように変更します。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥Java¥ClassPath=<path to
ORBIX license file >;<path to ORBIX domain configuration
file>;<ClassPath>;<orbix_inst_dir>¥orbix_art¥1.2¥classes¥orbix200
0.jar;<orbix_inst_dir>¥orbix_art¥1.2¥localhost
```

-
- 注
- <orbix_inst_dir> は、ORBIX のインストールディレクトリです。
 - <orbix_inst_dir>¥orbix_art¥1.2¥classes¥orbix2000.jar は、ORBIX の実装 JAR ファイル orbix2000.jar のパスです。
 - <ClassPath> は、iPlanet レジストリ内の既存の指定です。
 - iPlanet レジストリのキーを変更する方法については、32 ページの「iPlanet レジストリエディタについて」を参照してください。
-

4. iPlanet レジストリを次のように変更します。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥Java¥JavaArgs =
-Xbootclasspath/a:<orb_inst_dir>¥orbix_art¥1.2¥classes¥omg.jar
-DIASORBCLASS=com.ion.corba.art.artimpl.ORBImpl
-DORBdomain_name=domain name
-DORBconfig_domains_dir=<path to ORBIX domain configuration file>
-Dorg.omg.PortableInterceptor.ORBInitializerClass.com.ipplanet.ias.ion.orbinitializers.IONAServerORBInitializerImpl=
-DORBIXLoadBalancing=true
```

-
- 注
- IBride_ORB_CLASS は、ロードされた ORB を参照します。この場合は、ORBIX ORB の com.ion.corba.art.artimpl.ORBImpl を参照します。
 - iPlanet Application Server と CORBA クライアントが同じマシン上で動作している場合は、ORBdomain_name および ORBconfig_domains_dir を iPlanet レジストリの JavaArgs キーに追加する必要はありません。これらの要素は、ORBIX と共に出荷されるローカルホスト構成ファイルに設定されます。
-

5. iPlanet Application Server と CORBA クライアントが別々のマシン上で動作している場合は、ORBIX サービスが動作しているマシンに新しい設定ドメインを作成する必要があります。ドメイン設定は、ファイルベースまたはリポジトリベースで生成できます。詳細は、『ORBIX 管理者ガイド』を参照してください。

ファイルベースでドメインを作成した場合は、ORBIX サービスに関連する設定の詳細は構成ファイルに保存されます。このファイルは、特定のパスに設定ドメインを作成するときに作成されます。

新しい設定ドメイン名と設定ドメインファイルのパスで、次の iPlanet レジストリを更新する必要があります。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥Java¥JavaArgs =
-DORBdomain_name=domain_name -DORBconfig_domains_dir=domain_path
```

たとえば、設定ドメイン名が「Marigold」で、ドメイン設定ファイルがパス D:¥iona¥domains に保存されている場合は、iPlanet レジストリキーを次のように変更する必要があります。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥Java¥JavaArgs =
-DORBdomain_name=marigold
-DORBconfig_domains_dir=D:¥iona¥domains
```

6. 次の手順で ORBIX の構成ファイルを変更し、セキュリティサービスを有効にする必要があります。

- a. 構成ファイルの `server_binding_list` に、サーバのインターセプタ名 `iASServerInterceptor` をグローバル・スコープで追加します。

たとえば、サーバのインターセプタ名を追加する前は、コンフィグレーションファイルの関連するエントリは次のようになっています。

```
binding:server_binding_list=[OTS]
```

`iASServerInterceptor` をサーバのインターセプタ名として追加すると、コンフィグレーションファイルのエントリは次のようになります。

```
binding:server_binding_list=[iASServerInterceptor,OTS]
```

- b. インターセプタ `portable_interceptor` をコンフィグレーションファイルの `orb_plugins` エントリに追加します。

たとえば、インターセプタ `portable_interceptor` を `orb-plugins` エントリに追加する前は、エントリは次のようになっています。

```
orb_plugins=[iiop_profile,giop,iiop,ots]
```

`portable_interceptor` を移植可能なインターセプタ名として `orb-plugins` に追加すると、エントリは次のようになります。

```
orb_plugins=[iiop_profile,giop,iiop,ots,?portable_interceptor?]
```

7. CXS 間のロードバランサは、デフォルトで有効になっています。ロードバランサを無効にするには、iPlanet レジストリ内の次の JavaArgs 引数を False に設定します。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥Java¥JavaArgs
=ORBIXLoadBalancing = false
```

注 次のコマンドで、すべてのオブジェクトグループが表示されます。

```
<orbix home>/bin/itadmin nsog list
```

次のコマンドで、グループにバインドされたすべてのオブジェクトが表示されます。

```
<orbix home>/bin/itadmin nsog list_members <Group Name>
```

次のコマンドを使って、ロードバランサ機能のオンとオフを切り替えることができます。

```
<orbix home>/bin/itadmin nsog modify -type
[rr|rand|active] <GroupName>
```

「group name」は、ejb name + Group です。

CXS エンジン、登録が完了すると、クライアントサイドオブジェクトグループのオブジェクトの処理を開始します。RMI/IIOP ORB は、リクエストを CXS エンジンに送信します。CXS エンジンが停止している場合でも、リクエストを送信し続けます。この現象を回避するには、登録した CXS エンジンがすべて動作していなければなりません。通常、KAS (Administrative Server) は、障害が発生したサーバエンジンプロセスを再起動します。ただし、CXS エンジンが無効になっているか、削除されているか、起動していない (KAS が起動していない場合に発生する) 場合は、CXS エンジンのオブジェクトをすべてのオブジェクトグループから削除することをお勧めします。CXS エンジンのオブジェクトを削除するには、以下のコマンドを使います。

CXS エンジンによって処理されている現在のオブジェクトグループを一覧表示するには

```
itadmin nsog list
```

現在のメンバーを一覧表示するには

```
itadmin nsog list_members <Group Name>
```

特定のメンバーを削除するには (メンバー名はエンジン番号と同じ)

```
itadmin nsog remove <GroupName>
```

または

```
itadmin nsog remove_member -og_name <GroupName> <engine no>
```

削除を確認するには

```
itadmin nsog list_members <Group Name>
```

<group name> は ejb-name + Group です。

8. **orb.properties** ファイル内の次のプロパティを変更します。このファイルは *iASInstallDir/ias/usr/java/jre/lib* にあります。

```
org.omg.CORBA.ORBSingletonClass=com.ionacorba.art.artimpl.ORBSingleton
```

9. 必要なアプリケーションを配置します。アプリケーションを配置するときにスタブとタイが再生成されることを確認します。
10. コマンドプロンプトから次のコマンドを実行して、ORBIX 環境変数を設定します。

```
<orbix installation directory>/bin/<domain name>_env.bat
```

```
<orbix installation directory>/bin/<domain name>_java_env.bat
```

たとえば、ドメイン名が「Marigold」の場合は、次のように指定します。

```
<orbix installation directory>/bin/marigold_env.bat
```

```
<orbix installation directory>/bin/marigold_java_env.bat
```

11. コマンドプロンプトで次のコマンドを実行して、ORBIX サービスを起動します。

```
<orbix installation directory>/bin/start_<domain name>_services.bat
```

12. iPlanet Application Server を起動します。

クライアントサイドの CORBA プログラミングについては、『開発者ガイド』を参照してください。

メッセージ駆動 Bean の管理

iPlanet Application Server 上でメッセージ駆動 Bean を導入して実行するには、まず、クライアントがメッセージ駆動 Bean に接続できるように Java Message Service (JMS) プロバイダを設定する必要があります。

メッセージ駆動 Bean は JMS メッセージリスナとして機能するので、メッセージ駆動 Bean を導入できるようにメッセージングミドルウェアを設定する必要があります。現在、このバージョンの iPlanet Application Server では Java Message Queue のみがサポートされています。

この章には次のトピックがあります。

- JMS プロバイダの設定
- メッセージ駆動 Bean の管理

注 iPlanet Application Server でのメッセージ駆動 Bean の機能は、開発者の使用のみを目的としています。運用環境でのテストや動作確認は行われていません。

メッセージ駆動 Bean の導入の詳細は、『iPlanet Application Server 開発者ガイド』および Deployment Tool のオンラインヘルプを参照してください。

JMS プロバイダの設定

JMS プロバイダを設定するには、まず、メッセージングミドルウェアをインストールする必要があります。デフォルトでは、iPlanet Application Server は iPlanet Message Queue (iMQ) for Java 2.0 SP1 を使用します。iMQ for Java 2.0 SP1 は、iPlanet Application Server のインストール CD にバンドルされています。インストール CD が
ない場合は、
http://www.ipplanet.com/products/ipplanet_message_queue/home_message_queue.html からダウンロードできます。

iPlanet Application Server に付属の iMQ for Java 2.0 SP1 のほかに、サードパーティ製品を使うこともできます。

この節には次のトピックがあります。

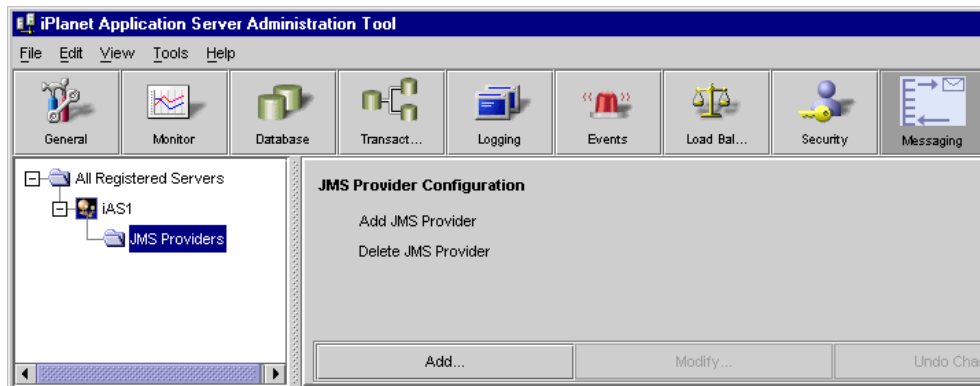
- iMQ for Java を設定するには
- サードパーティ JMQ を設定するには

iMQ for Java を設定するには

iMQ for Java 2.0 SP1 の設定を開始する前に、iPlanet Message Queue for Java 2.0 SP1 がマシンにインストールされていることを確認してください。

次の手順で JMS プロバイダとメッセージキューの設定を行い、メッセージ駆動 Bean をサポートできるようにします。

1. iASAT を起動します。
2. メニューバーの「Messaging」アイコンをクリックします。



3. 左側のペインで「JMS Providers」をクリックします。
4. 右側のペインで「追加」をクリックします。
5. 環境に関する項目を入力します。

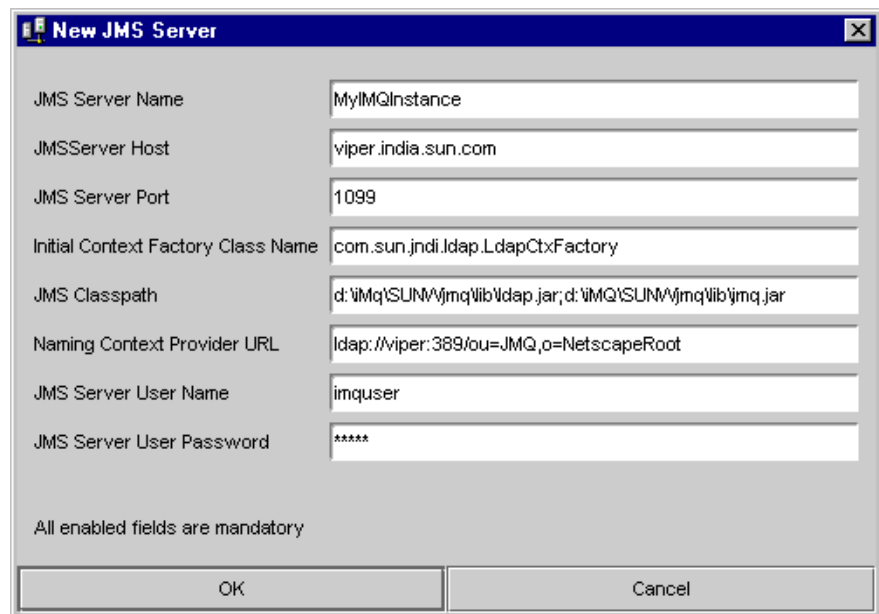
すべてのフィールドは必須です。以下のフィールドを指定する必要があります。

- 「JMS Server Name」
- 「JMS Server Host」
- 「JMS Port Number」
- 「Initial Context Factory Class Name」

これは、LDAP URL、ファイルシステムパスなどの観点で指定できます。

- 「JMS ClassPath」
- 「Naming Context Provider URL」
- 「JMS Server User Name」
- 「JMS Server User Password」

次の図に設定例を示します。



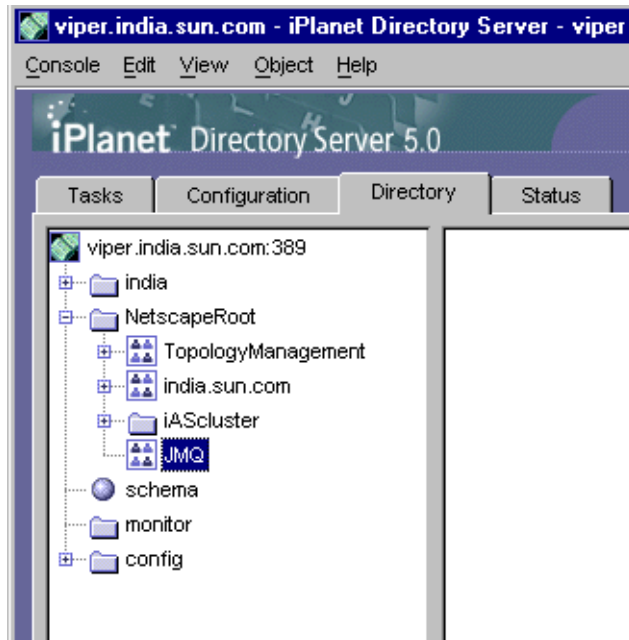
The screenshot shows a dialog box titled "New JMS Server" with the following fields and values:

Field	Value
JMS Server Name	MyJMQInstance
JMS Server Host	viper.india.sun.com
JMS Server Port	1099
Initial Context Factory Class Name	com.sun.jndi.ldap.LdapCtxFactory
JMS Classpath	d:\jmq\SUN\jmq\lib\ldap.jar;d:\jmq\SUN\jmq\lib\jmq.jar
Naming Context Provider URL	ldap://viper:389/ou=JMQ,o=NetscapeRoot
JMS Server User Name	imquser
JMS Server User Password	*****

At the bottom of the dialog, there is a note: "All enabled fields are mandatory". Below the note are "OK" and "Cancel" buttons.

6. 「OK」をクリックして変更を確定します。
7. ディレクトリサーバにキューのエントリを作成します。

ファイルシステム・コンテキストファクトリ名を使用している場合は、そのファイルのパスを指定します。



組織単位の作成については、iPlanet Directory Server 5.0のマニュアルを参照してください。

8. iMQ for Java 2.0 で使用可能なツールを使って、ディレクトリサーバの `ou=JMQ` の下に、キューコネクション・ファクトリオブジェクト `QCF` を作成します。

次にファイルの例を示します。

```
cmdtype=add
obj.type=qf
obj.lookupName=cn=QCF
objstore.attrs.java.naming.factory.initial=%com.sun.jndi.ldap.Ld
apCtxFactory
objstore.attrs.java.naming.provider.url=%ldap://border.india.sun
.com:389/ou=JMQ,o=NetscapeRoot
objstore.attrs.java.naming.security.principal=cn=Directory
Manager
objstore.attrs.java.naming.security.credentials=netscape
objstore.attrs.java.naming.security.authentication=simple
```


9. iPlanet Console を使って、ディレクトリの cn=QCF オブジェクトが読み取り、比較、および検索のパーミッションを持つことを確認します。

これで JMS プロバイダが設定されたので、アプリケーションでメッセージ駆動 Bean の使用を開始できます。

サードパーティ JMQ を設定するには

サードパーティの Java メッセージングミドルウェアを設定するには、次の手順を実行します。使用するメッセージキューが JMS 仕様書の第 8 章「JMS Application Server Facilities」に準拠していることを確認してください。

サードパーティの JMS プロバイダを設定するには、次の手順を実行します。

1. iPlanet Directory Server の ou=JMQ, o=NetscapeRoot の下に、Factory Context と設定パラメータを手動で作成します。
2. iPlanet Message Queue for Java 2.0 SP1 をすでに設定している場合は、iMQ 固有の次のエントリを iPlanet レジストリから削除します。
 - ServerHost
 - ServerPort
 - Username
 - Password

これらの値は、iPlanet Message Queue でのみ有効です。JMQ 用に指定した Factory の設定値を使用してください。

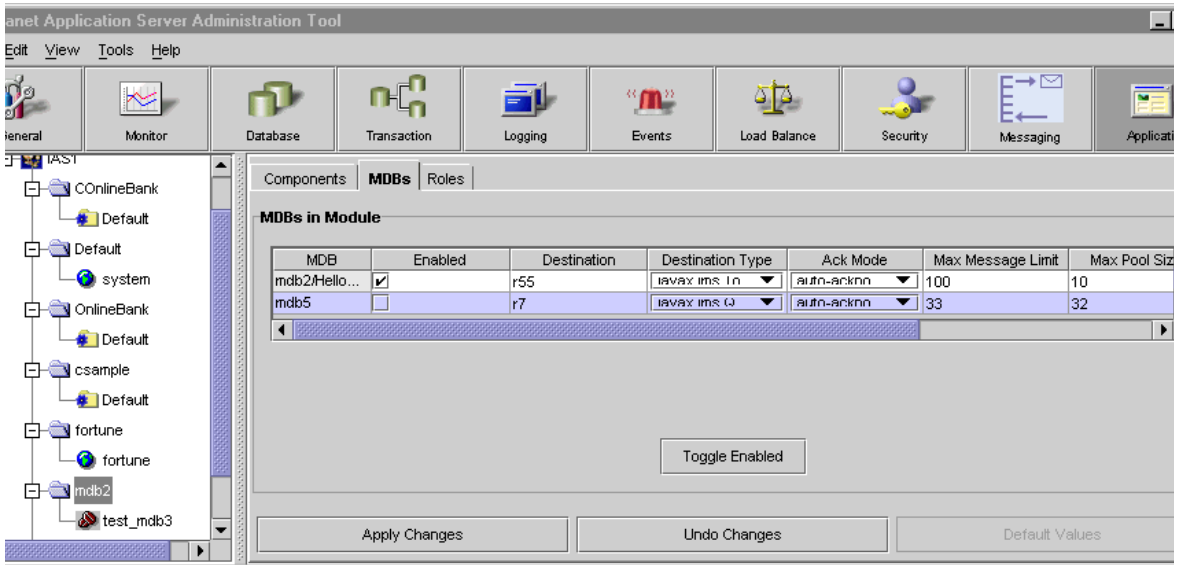
メッセージ駆動 Bean の管理

メッセージ駆動 Bean は、iPlanet Application Server Administration Tool を使って管理できます。モジュール内の各メッセージ駆動 Bean のプロパティを個別に変更できます。

メッセージ駆動 Bean を管理するには

導入済みのメッセージ駆動 Bean のプロパティを変更するには、次の手順を実行します。

1. iASAT を起動します。
2. メニューバーの「アプリケーション」をクリックします。
左側のペインに、配置済みのアプリケーションがすべて表示されます。
3. 変更するメッセージ駆動 Bean を選択します。
4. 右側のペインで、「MDBs」タブを選択します。
次のダイアログボックスが表示されます。



変更できるフィールドは次のとおりです。

- 「MDB Name」

- 「Enabled Flag」
フラグが無効の場合、KJS は起動時にメッセージ駆動 Bean を読み込みません。
 - 「デスティネーション名」
 - 「デスティネーションタイプ」
「Queue」または「Topic」を指定できます。
 - 「通知モード」
 - 「最大メッセージ制限」
 - 「最大プールサイズ」
 - 「最小プールサイズ」
 - 「永続名」
このプロパティは、トピックサブスクリプションの場合のみ設定できます。
5. 「変更の適用」をクリックして変更を確定します。
 6. iPlanet Application Server を再起動して変更を有効にします。

複数の iPlanet Application Server の管理

第 12 章 「複数のサーバの設定」

第 13 章 「複数サーバアプリケーションの管理」

第 14 章 「ユーザリクエストのロードバランス」

第 15 章 「分散のデータ同期の管理」

第 16 章 「開発用 iPlanet Application Server の設定」

付録 A 「トラブルシューティング」

複数のサーバの設定

この章では、iPlanet Application Server Administration Tool を使って複数の iPlanet Application Server マシンを設定する方法について説明します。

この章には次のトピックがあります。

- 複数サーバ企業での Web コネクタプラグイン
- 分散データ同期とロードバランス
- マルチキャスト通信

複数サーバ企業での Web コネクタプラグイン

Web コネクタプラグインは、使用している iPlanet Application Server マシン上のアプリケーションにユーザのリクエストを転送します。複数サーバ企業では、アプリケーションサーバを指定して、そこに、Web コネクタプラグインを接続して、Web サーバのリクエストを記録することができます。指定したアプリケーションサーバは、Web コネクタがリクエストおよび他のアプリケーション情報を交換するデフォルトのサーバとなります。iPlanet Application Server のロードバランサブラグインに対して、アプリケーションリクエストが転送される代替アプリケーションサーバが指定されていない場合、アプリケーションリクエストはこのデフォルトサーバだけに送信されます。

また、指定したアプリケーションサーバに対して、Web コネクタプラグインからアプリケーションリクエスト情報を送信して記録することもできます。

この節には次のトピックがあります。

- 複数サーバ向けの Web コネクタプラグインの設定
- リクエストの送信先となるアプリケーションサーバの指定
- ログの記録先となるアプリケーションサーバの指定

複数サーバ向けの Web コネクタプラグインの設定

複数の iPlanet Application Server マシンを使って企業アプリケーションをサポートする場合は、iPlanet Application Server にリクエストを転送するために Web サーバの設定方法を選択する必要があります。これらの設定オプションは、Web コネクタプラグインに用意されています。次の表に記載されている設定シナリオを参考にして、企業に最適な Web コネクタプラグインの設定方法を決定してください。

表 12-1 複数のサーバ向けの Web コネクタの設定

設定シナリオ	設定方法
ロードバランスが無効になっている複数の iPlanet Application Server マシンを 1 台の Web サーバでサポートする	アプリケーションがパーティション分割されていることを前提とします。リクエストを転送するアプリケーションサーバは、Web ブラウザからの初期リクエストを処理するアプリケーション オブジェクトが収容されているサーバに設定します。他の iPlanet Application Server マシンでは、一方のサーバ上のオブジェクトが起動したアプリケーション コンポーネントを管理します。
ロードバランスが無効になっている複数の iPlanet Application Server マシンを、複数の Web サーバでサポートする	アプリケーションがパーティションされていない場合は、適切な iPlanet Application Server マシンにリクエストを転送するように各プラグインを設定します。 アプリケーションがパーティションに分割されている場合は、リクエストを転送する iPlanet Application Server マシンは、Web ブラウザからの初期リクエストを処理するコンポーネントが収容されているサーバに設定します。1 台の iPlanet Application Server マシンに複数のプラグインを接続できます。
ロードバランスが有効になっている複数の iPlanet Application Server マシンを、1 台の Web サーバでサポートする	ロードバランサプラグインによって、適切な iPlanet Application Server マシンにアプリケーションリクエストを転送します。 リクエストを転送する iPlanet Application Server マシンはデフォルトとして、パフォーマンスが最も高いマシンか、Web ブラウザ初期リクエストを処理するアプリケーション コンポーネントが収容されているマシンに設定します。

表 12-1 複数のサーバ向けの Web コネクタの設定 (続き)

設定シナリオ	設定方法
ロードバランスが有効になっている複数の iPlanet Application Server マシンを、複数の Web サーバでサポートする	ロードバランサプラグインによって、適切な iPlanet Application Server マシンにアプリケーションリクエストを転送します。 デフォルトとして iPlanet Application Server マシンにリクエストを転送するか、Web ブラウザ初期リクエストを処理するアプリケーション コンポーネントが収容されているマシンに転送します。

アプリケーションの負荷を均衡すると、Web コネクタプラグインはロードバランサプラグインと連動して、複数の iPlanet Application Server マシン全体にリクエストを自動的に分散します。すべてのリクエストが 1 台の iPlanet Application Server マシンに転送されることがなくなります。

アプリケーションの負荷を均衡しない場合は、Web サーバのアプリケーションリクエストの転送先をユーザが指定する必要があります。

リクエストの送信先となるアプリケーションサーバの指定

複数のアプリケーションサーバを実装した企業では、Web コネクタがアプリケーションリクエストをどこに送信するかを指定できます。

ロードバランスを有効にしている場合は、ロードバランサプラグインが、最初にリクエストの転送先を判断します。ただし、リクエストの送信先のサーバをロードバランサのプラグインが判断するように設定されない場合、Web コネクタはユーザによって指定された iPlanet Application Server マシンにリクエストを転送します。

Web サーバが接続する iPlanet Application Server マシンを指定するには、次の手順を実行します。

1. iPlanet レジストリエディタを開きます。
エディタが開き、iPlanet Application Server インスタンスに適用するキーと値が表示されます。
2. 次のキーを開きます。
`SOFTWARE¥iPlanet¥Application Server¥6.5¥CCSO¥HTTPAPI¥`
3. GXIP 文字列値をダブルクリックします。
「値を変更」ダイアログボックスが表示されます。

4. 値データとして、デフォルトの iPlanet Application Server マシンのホスト IP アドレスを入力します。
5. GXportNum DWORD 値をダブルクリックします。
6. 値のデータとして、同じアプリケーションサーバの Executive Server プロセスのポート番号を入力し、「OK」をクリックします。

ログの記録先となるアプリケーションサーバの指定

複数サーバ企業では、Web サーバのログの記録に使うアプリケーションサーバを指定できます。

単一のサーバ企業では、そのサーバが、Web コネクタがデフォルトでアプリケーションリクエストを転送する iPlanet Application Server マシンとなります。単一のサーバ企業では、この値を変更しないでください。

複数サーバを備えた企業では、ログを記録するアプリケーションサーバは、Web コネクタがアプリケーションリクエストをデフォルトで送信するサーバと同じです。

ログの記録を実行する iPlanet Application Server マシンを指定するには、次の手順を実行します。

1. iPlanet レジストリエディタを開きます。
エディタには、アプリケーションサーバに適用するキーと値が表示されます。
2. 次のキーを開きます。
`SOFTWARE¥iPlanet¥Application Server¥6.5¥CCSO¥HTTPLOG¥`
3. Host 文字列値をダブルクリックします。
「値を変更」ダイアログボックスが表示されます。
4. 値のデータとして、Web サーバのログを記録するアプリケーションサーバのホスト IP アドレスを入力し、「OK」をクリックします。
5. Port DWORD 値をダブルクリックします。
6. 値のデータとして、同じアプリケーションサーバの Executive Server プロセスのポート番号を入力し、「OK」をクリックします。
7. iPlanet レジストリエディタを閉じます。

分散データ同期とロードバランス

複数のアプリケーションサーバを実装するときは、サーバ全体でロードバランスを有効にするかどうかを判断する必要があります。ロードバランスのために分散されたアプリケーションは、状態およびセッションを管理する場合、アプリケーションサーバの分散同期サービスに依存します。

分散データ同期は、iPlanet Application Server をインストールするときに設定します。インストレーションスクリプトによって、サーバを分散データ同期に含めるかどうかおよび、プライマリサーバのホスト名とポート番号を指定するように要求されます。分散データ同期については、287 ページの「分散のデータ同期について」を参照してください。

分散データ同期環境の設定

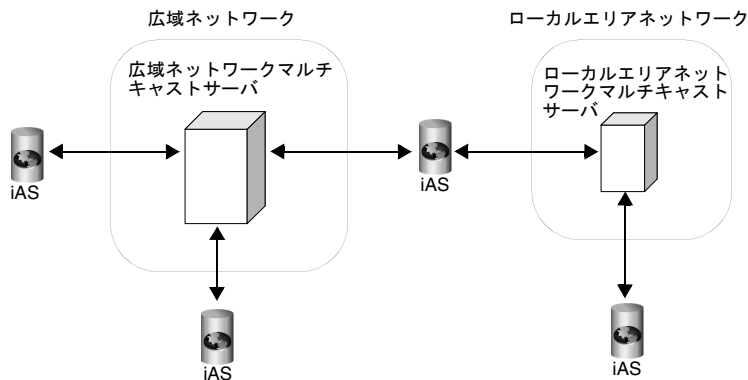
iPlanet Application Server を複数のマシンにインストールした場合は、分散データ同期に加わるサーバのクラスタキーを更新する必要があります。更新は、iPlanet レジストリエディタを使って行います。

クラスタ内のサーバのキーを更新すると、各サーバはプライマリサーバ、即時バックアップ、およびサーバの障害発生時に他のサーバがプライマリサーバとなるときの優先順位に関する情報を共有できます。

分散データ同期環境を設定するには、第 15 章「分散のデータ同期の管理」を参照してください。

マルチキャスト通信

複数サーバ企業では、ロードバランスおよび管理を目的として、マルチキャスト広域ネットワーク (WAN) サービスを利用した、アプリケーションサーバ間通信が行われます。マルチキャストサービスでは、仮想サーバを介して、すべてのメッセージの送信および分散が行われます。アプリケーションサーバでは、N 方向のマルチキャスト構成を使ってブロードキャスト情報を送受信します。次の図は、このネットワークの概要を示しています。



ローカルエリアネットワーク (LAN) 内のマルチキャスト通信サービスは、ネットワークハードウェアによって処理されます。つまり、デフォルトのマルチキャストアドレスを登録または変更する必要はありません。広域ネットワークでサーバを実装する場合、iPlanet Application Server マシン間で通信するには、登録済みのマルチキャストパブリックアドレスを使う必要があります。

ロードバランスでのマルチキャストの適用方法

ロードバランスを行う場合は、すべてのサーバ間で通信したり、サーバのグループを作成し、そのグループ間でのみアプリケーション負荷を均衡させることができます。たとえば、New York のアプリケーションは Los Angeles の同じアプリケーションと負荷を分散する必要はありません。一方、Cupertino、Sunnyvale、および Santa Clara のアプリケーションは、San Jose 地域のすべてのユーザと負荷を共有する必要があります。

ロードバランスを行う場合、マルチキャスト通信は、Executive Server のマルチキャストアドレスによって制御されます。

複数サーバアプリケーションの管理

この章では、iPlanet Application Server Administration Tool (iASAT) を使って複数の iPlanet Application Server マシン上のアプリケーションを管理する方法について説明します。

この章には次のトピックがあります。

- iASAT について
- 複数サーバ上でアプリケーションをローカルで管理する
- 複数サーバ上でパーティション分割したアプリケーションを管理する
- ロードバランスを行うためにアプリケーションを管理および配置する
- 分散アプリケーションコンポーネントの属性を変更する

iASAT について

iASAT を使うと、複数のサーバに保存されているアプリケーションを同時に管理できます。Enterprise Java Beans (EJB) などのアプリケーションコンポーネントが複数のアプリケーションサーバに分散されている場合、アプリケーションコンポーネントに行った設定はサーバ全体で自動的に更新されます。また、特定の iPlanet Application Server マシンの設定を、グループまたは企業全体の他の iPlanet Application Server マシンにコピーおよび適用できます。

管理ツールを使って、企業内の各 iPlanet Application Server マシンを表示し、1 台または複数のサーバを同時に変更できます。

複数の iPlanet Application Server マシンでアプリケーションを管理するには、次のどちらかの手順を実行します。

- 複数のサーバにアプリケーションまたはその一部を分散し、それぞれに特定のリクエストおよびアプリケーション処理を割り当てます。

- 複数のサーバにアプリケーションコンポーネントを複製し、負荷を均衡させて、アプリケーションのパフォーマンスを向上させます。

操作するサーバ数が多いほど、アプリケーションの管理構成の選択肢が増えます。

次の表では、複数の iPlanet Application Server マシン上でアプリケーションを管理するための一般的な 3 つの方法について説明します。

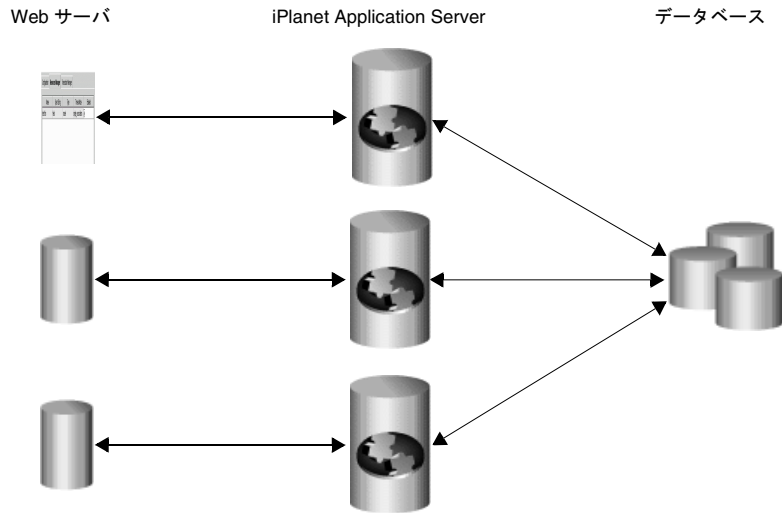
表 13-1 複数の iPlanet Application Server マシン上でアプリケーションを管理する方法

管理構成	説明
ローカル	アプリケーションを各 iPlanet Application Server マシンにインストールし、複数の Web サーバを使ってリクエストを各サーバに送信する。iPlanet Application Server マシン間の通信は行わない
パーティション	アプリケーションを分散し、別々の iPlanet Application Server マシン上で管理する。各サーバは、アプリケーションコンポーネントが管理されているサーバを認識し、適切なサーバにリクエストを転送する
分散してロードバランス	アプリケーションの一部または全部を複数の iPlanet Application Server マシンに複製する。アプリケーションリクエストの負荷が均衡するようにサーバを設定する

複数サーバ上でアプリケーションをローカルで管理する

アプリケーションを複数サーバ上でローカルに管理する方法が、3 つの一般的なサーバ構成の中でもっとも簡単です。この構成では、各 iPlanet Application Server マシン上にすべてのアプリケーションを配置します。アプリケーションが iPlanet Application Server マシン上にすでにインストールされている場合は、Deployment Tool を使って他のサーバにアプリケーションを配置することができます。

この構成では、リクエストが適切な iPlanet Application Server マシンに転送されるように、各 Web コネクタプラグインを設定する必要があります。



また、共通の Web サーバおよび LDAP サーバを共有しながら、複数の iPlanet Application Server インスタンスにローカルアプリケーションを配置することもできます。この構成機能は、Web サーバが 1 つである点を除いて、最初の例に類似しています。すべての iPlanet Application Server インスタンスが、同じ LDAP サーバを介して共通の構成を共有します。前の例では、クライアントが複数の異なる Web サーバにアクセスする必要がありましたが、この構成は、複数の iPlanet Application Server インスタンス間で負荷を均衡させることができます。この構成は、アプリケーションの特性に応じて選択してください。

複数サーバ上でパーティション分割したアプリケーションを管理する

アプリケーションをパーティション分割するには、アプリケーションを構成するアプリケーションコンポーネントを分割する必要があります。分割したアプリケーションコンポーネントは、別個の iPlanet Application Server マシンで管理されます。アプリケーションをパーティション分割すると、iPlanet Application Server マシンごとに固有の処理を実行することができます。

たとえば、主にデータアクセスを処理する Servlets は I/O の使用量が多く、計算を処理する Servlets は CPU とアクティブメモリの使用量が多くなります。アプリケーション全体のパフォーマンスを向上させるには、アプリケーションをパーティション分割して、さまざまなタイプの Servlets を個別の iPlanet Application Server マシン上で管理します。

この節には次のトピックがあります。

- パーティション分割されたアプリケーションの設定
- アプリケーションコンポーネントの有効 / 無効を切り替える

パーティション分割されたアプリケーションの設定

パーティション分割されたアプリケーションを設定するには、次の手順を実行します。

1. iPlanet Application Server Deployment Tool を使って、関係するすべての iPlanet Application Server マシンに完全なアプリケーションを配置します。

各登録済み iPlanet Application Server に配置したアプリケーションと関連モジュールは、「アプリケーション」ウィンドウの左側のペインで確認できます。サーバを展開して配置済みアプリケーションを確認し、アプリケーションフォルダを展開してアプリケーションのモジュールを確認します。

アプリケーションの配置の詳細は、Deployment Tool のオンラインヘルプを参照してください。

2. ロードバランスを有効にすると、他のサーバでホストされているアプリケーションコンポーネントを各サーバが見つけることができます。

ロードバランスの詳細は、第 14 章「ユーザリクエストのロードバランス」を参照してください。

3. サーバごとに、特定のアプリケーションコンポーネントを無効にします。

257 ページの「アプリケーションコンポーネントの有効 / 無効を切り替える」を参照してください。

アプリケーションコンポーネントをパーティション分割しているときに、アプリケーションコンポーネントがインストールされているサーバを表示する場合は、次の手順を実行します。

1. iASAT の「アプリケーション」 ウィンドウを開きます。
2. 「アプリケーション」 ウィンドウの左側のペインで、パーティション分割するアプリケーションコンポーネントが配置されているサーバを展開します。
3. アプリケーションフォルダを開きます。J2EE アプリケーションの場合は、「Servlet」 アイコンを強調表示します。C++ アプリケーションの場合は、「AppLogic」 アイコンを強調表示します。

「アプリケーション」 ウィンドウの右側のペインに、配置したアプリケーションコンポーネントが表示されます。

- a. 「アプリケーション」 ウィンドウの右側のペインでアプリケーションコンポーネントを選択します。
- b. 「Servlet コンポーネントのプロパティ」 (または「アプリケーションコンポーネントプロパティ」) をクリックします。

ダイアログボックスが表示され、コンポーネントがインストールされているアプリケーションサーバが表示されます。選択した iPlanet Application Server マシンが表示されていない場合は、必要なアプリケーションコンポーネントが含まれる .ear ファイルをそのマシンに配置する必要があります。

- c. 「OK」 をクリックしてダイアログボックスを閉じます。

アプリケーションコンポーネントの有効 / 無効を切り替える

アプリケーションのコンポーネント (Servlet など) を無効にすると、ユーザはそのコンポーネントにアクセスできなくなります。コンポーネントを無効にすると、現在のリクエストは処理されますが、コンポーネントを再度有効にするまで新しいリクエストは受け入れられません。

アプリケーションコンポーネントを無効にするには、次の手順を実行します。

1. iASAT ツールバーの「アプリケーション」 をクリックして「アプリケーション」 ウィンドウを開きます。
2. 「アプリケーション」 ウィンドウの左側のペインで、無効にするアプリケーションコンポーネントが配置されているサーバをダブルクリックします。
3. 無効にするアプリケーションコンポーネントが含まれるフォルダを開きます。

4. アプリケーションフォルダを開いて、アプリケーションモジュールを表示します。
5. 無効にするアプリケーションコンポーネントが含まれるモジュールを選択します。
「アプリケーション」ウィンドウの右側のペインには、モジュール内のアプリケーションコンポーネントが表示されます。
6. 「アプリケーション」ウィンドウの右側のペインで、無効にするコンポーネントを検索します。
7. 無効にするコンポーネントを選択し、「有効」チェックボックスをクリックして、チェックボックスをクリアします。

Servlets	Enabled	Mode	Sticky LB
System_FormAuthServlet	<input checked="" type="checkbox"/>	Local	<input type="checkbox"/>
System_CertAuthServlet	<input checked="" type="checkbox"/>	Local	<input type="checkbox"/>
System_JSPRunnerSticky	<input checked="" type="checkbox"/>	Local	<input type="checkbox"/>
System_StaticServlet	<input checked="" type="checkbox"/>	Distributed	<input checked="" type="checkbox"/>
System_JSPRunner	<input checked="" type="checkbox"/>	Local	<input type="checkbox"/>
System_BasicAuthServlet	<input checked="" type="checkbox"/>	Global	<input checked="" type="checkbox"/>

8. グループ内のすべてのアプリケーションコンポーネントを有効または無効にする場合は、「有効に切り替え」をクリックします。
アプリケーションコンポーネントを有効にするには、対応する「有効」チェックボックスをオンにします。

警告

1つの Directory Server で設定されたクラスタ環境でアプリケーションが稼動している場合、1つのサーバ上でアプリケーションコンポーネントを無効にすると、ほかのすべてのサーバ上のアプリケーションコンポーネントも無効になります。

クラスタ内の複数のサーバ上で稼動するアプリケーションコンポーネントを無効にする場合は、次の手順を実行します。

1. Administration Tool のメインメニューの「アプリケーション」をクリックします。
2. 左側のペインからアプリケーションモジュールを選択します。

そのアプリケーションの Servlets が右側のペインに表示されます。

3. 目的の Servlet を選択します。
4. 「Servlet コンポーネントのプロパティ」をクリックします。

「アプリケーションコンポーネントプロパティ」ウィンドウが開き、そのアプリケーションが登録されているサーバのリストが表示されます。

5. コンポーネントを無効にするサーバを選択します。
6. 「削除」をクリックします。

これで、選択した iPlanet Application Server からアプリケーションコンポーネントが削除されます。

9. 「変更の適用」をクリックし、iPlanet Application Server マシンに変更を保存します。

ロードバランスを行うためにアプリケーションを管理および配置する

アプリケーションリクエストの負荷を均衡すること(ロードバランス)は、アプリケーションのパーティション分割とは異なります。ロードバランスを行うには、1つまたは複数のアプリケーションコンポーネントのコピーを複数の iPlanet Application Server マシン上に置く必要があります。アプリケーションコンポーネントを複数のサーバに単純に分割する(アプリケーションをパーティション分割する)わけではありません。次に、各サーバを設定し、アプリケーションコンポーネントが他のサーバでも実行されるようにします。

アプリケーションを配置するとき、アプリケーションのロードバランスを設定するかどうか、設定する場合はその方法を決定する必要があります。次のロードバランス設定方法から選択してください。

- 運用環境のサーバ間だけで負荷を均衡させる。複数の iPlanet Application Server マシンに配置した場合

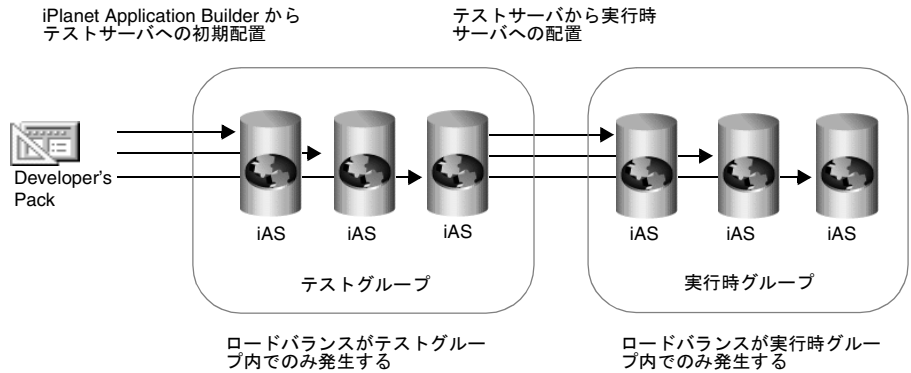
例 1: 3 台の iPlanet Application Server マシンを使ってアプリケーションをテストすると想定します。ユーザのリクエストが実際に処理される運用環境も、3 台の iPlanet Application Server マシンから構成されます。2 つのサーバグループ間でアプリケーションコンポーネントが異なる可能性があるため、ロードバランスは有効にしません。したがって、テストサーバのアプリケーションを運用環境のサーバに配置するときは、配置先のサーバ間だけで負荷を均衡する必要があります。

- 既存の運用サーバと企業環境に追加した新しいサーバとの間でアプリケーション負荷を均衡させる

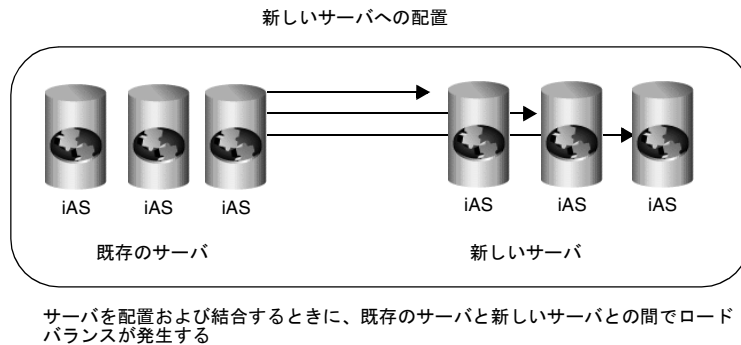
例 2: 企業環境を拡張して、運用グループに 3 台の iPlanet Application Server マシンを追加すると想定します。既存の運用サーバのアプリケーションを新しいサーバに配置するときに、運用グループ内のすべてのサーバを結合します。アプリケーションの負荷は、既存のサーバと新しいサーバ間で均衡させます(シナリオ 2)。

- アプリケーションをサーバにローカル構成で配置します(ロードバランスは行われません)。

次の図は、例 1 で説明したロードバランスを示しています。



次の図は、例 2 で説明したサーバの結合を示しています、新しいサーバをグループに追加し、結合オプションを使ってアプリケーションをグループ内のサーバに配置します。



アプリケーションを配置するときにローカル構成で分散した場合は、サーバ間でアプリケーションリクエストのロードバランスは発生しません。

分散アプリケーションコンポーネントの属性を変更する

複数のサーバにわたって分散されたアプリケーションコンポーネントの固定 (ステイッキー) ロードバランスを有効にする属性などを変更すると、その変更はコンポーネントが管理されるサーバ上にレプリケートされます。ローカル構成でインストールしたアプリケーションを分散して、ロードバランスを行う場合は、インストール済みアプリケーションコンポーネントの分散レベル (ローカル、分散、およびグローバル) を変更します。ロードバランスを無効にするには、指定したサーバ上のアプリケーションの構成を分散からローカルに変更します。

特定のサーバ上に配置されているコンポーネントの状態を分散またはグローバルからローカルに変更すると、そのコンポーネントを管理する各サーバは、そのサーバとのロードバランスを停止します。

たとえば、ShopCart というアプリケーションコンポーネントが、サーバ A、B、および C にわたって分散されているとします。ShopCart をサーバ A 上ではローカル状態で実行し、サーバ B および C では引き続き分散状態で実行する場合、各サーバ (A、B、および C) が自動的に更新され、ShopCart のリクエストがサーバ B および C からサーバ A に渡されなくなります。サーバ B または C に送信された ShopCart のリクエストは、2 つのサーバ間だけで交換されます。サーバ A に送信された ShopCart へのリクエストは、サーバ A だけで処理されます。

アプリケーションコンポーネントの分散レベルを変更するには、次の手順を実行します。

1. iASAT の「アプリケーション」ウィンドウを開きます。
2. 「アプリケーション」ウィンドウの左側のペインで、アプリケーションの設定を変更するサーバを展開します。
3. アプリケーションフォルダを展開して、変更するアプリケーション コンポーネントが含まれる「Servlet」または「AppLogic」アイコンを選択します。
4. 「アプリケーション」ウィンドウの右側のペインで、分散レベルを変更する各アプリケーション コンポーネントを選択します。
 - a. ローカル : Servlet または AppLogic は、特定の iPlanet Application Server マシン上だけで動作します。
 - b. 分散 : Servlet または AppLogic は、指定した iPlanet Application Server マシン上だけで動作します。
 - c. グローバル : Servlet または AppLogic は、企業内の任意の場所で動作します。

Components		Roles	
Servlets in Module System			
Servlets	Enabled	Mode	Sticky LB
System_FormAuthServlet	<input checked="" type="checkbox"/>	Local	<input type="checkbox"/>
System_CertAuthServlet	<input checked="" type="checkbox"/>	Distributed	<input type="checkbox"/>
System_JSPRunnerSticky	<input checked="" type="checkbox"/>	Distributed	<input checked="" type="checkbox"/>
System_StaticServlet	<input checked="" type="checkbox"/>	Distributed	<input type="checkbox"/>
System_JSPRunner	<input checked="" type="checkbox"/>	Distributed	<input type="checkbox"/>
System_BasicAuthServlet	<input checked="" type="checkbox"/>	Distributed	<input type="checkbox"/>

5. 「モード」カラムで分散レベルを変更します。

- 選択したグループ内のすべてのコンポーネントの分散レベルを変更する場合は、「有効に切り替え」をクリックします。すべてのアプリケーションコンポーネントが同時に更新されます。
- アプリケーション「モード」を「ローカル」から「分散」または「グローバル」に変更する場合は、ロードバランスを適用する iPlanet Application Server マシンを指定する必要があります。
- 「モード」を「分散」または「グローバル」から「ローカル」に変更する場合は、他の設定を変更する必要はありません。

アプリケーションコンポーネントの「モード」を「分散」に変更すると、「アプリケーション」ウィンドウの左側のペインに表示されているすべての登録済みのサーバが、アプリケーションコンポーネントのサーバリストに追加されます。「アプリケーションコンポーネントプロパティ」ボタンをクリックすると、サーバリストにアクセスできます。

6. 左側のペインの「全ての登録済みサーバ」の下で、選択したアプリケーションコンポーネントのロードバランスに加わる iPlanet Application Server マシンを選択します。ロードバランスに含める各 iPlanet Application Server マシンには、アプリケーションコンポーネントをインストールする必要があります。
7. アプリケーションサーバを追加登録する場合は、「ファイル」メニューの、「新規」を選択し、「サーバ」を選択します。
8. 各アプリケーションコンポーネントに対して、これらの手順を繰り返します。
9. 「変更の適用」をクリックし、iPlanet Application Server マシンに変更を保存します。

分散アプリケーションコンポーネントの属性を変更する

ユーザリクエストのロードバランス

この章では、ロードバランスについて説明します。ロードバランスは、ユーザリクエストを処理する各 iPlanet Application Server のパフォーマンスを最適化するために、複数の iPlanet Application Server マシン間でリクエストの負荷を均衡させる機能です。

この章には次のトピックがあります。

- ロードバランスの動作
- ロードバランスの必要条件
- 固定 (スティッキー) ロードバランスとは
- ロードバランス方式の選択
- コンポーネントの応答時間によるロードバランス
- サーバの応答時間によるロードバランス
- ラウンドロビンロードバランス
- ユーザ定義条件ロードバランス

ロードバランスの動作

ロードバランスの目的は、複数の iPlanet Application Server マシン間に負荷を均一に分散させることです。iASAT を使ってロードバランスを設定すると、ユーザのリクエストを適切に分散させることができます。

たとえば、特定の時間帯に多数のユーザが集中してアプリケーションにアクセスする場合は、複数の iPlanet Application Server マシン上で AppLogics や Servlets などのアプリケーションコンポーネントを複製し、ロードバランスを有効にします。特定の iPlanet Application Server マシンの処理能力が限界に達すると、後続のリクエストはアプリケーションコンポーネントが複製されている他の iPlanet Application Server マシンに転送されます。複数のサーバ間にリクエストを均一に分散すると、応答時間を短縮することができます。

iPlanet Application Server マシンでは、ロードバランス方式を指定することができます。ロードバランス方式には、Web コネクタプラグイン駆動または iPlanet Application Server 駆動があります。

- Web コネクタプラグイン駆動 : Web コネクタプラグインが、リクエストを送信する iPlanet Application Server インスタンスを選択します。
- iPlanet Application Server 駆動 : iPlanet Application Server にロードバランスがロードバランスを決定します。サーバおよびリクエスト統計が収集され、マルチキャストを介してクラスタ内の iPlanet Application Server マシン間で交換されます。マルチキャストの詳細については、第 12 章「複数のサーバの設定」を参照してください。

ロードバランスの必要条件

アプリケーション負荷を均衡させる前に、次の必要条件を満たす必要があります。

- アプリケーションコンポーネントは、複数の iPlanet Application Server マシン上、またはロードバランスに含めるすべての iPlanet Application Server マシン上に複製する必要があります。
- アプリケーションコンポーネントは、特定の iPlanet Application Server マシン、または企業内のすべての iPlanet Application Server マシンに分散する必要があります。

ロードバランスを有効にする方法については、260 ページの「ロードバランスを行うためにアプリケーションを管理および配置する」を参照してください。

固定 (スティッキー) ロードバランスとは

同じセッション内のリクエストが複数の iPlanet Application Server マシンまたはプロセスによって処理された場合、分散に対応していないセッション情報は失われます。したがって、特定のアプリケーションコンポーネントは、セッション (固有ロードバランス) に設定され、同じサーバ上で処理することによって、セッション情報の喪失を回避しています。

アプリケーションコンポーネントが固定ロードバランスに設定されている場合、同一 iPlanet Application Server マシン、つまり最初に起動されたマシン上で処理されます。たとえば、ShopCart というアプリケーションコンポーネントがアプリケーションサーバ Server A と Server B に複製され、ロードバランスが行われるとします。ShopCart が Server B 上の Client 1 によって起動されると、Client 1 から送信された ShopCart への固定リクエストはすべて Server B だけで処理されます。つまり、ShopCart は Client 1 のセッション期間中、Server B に固定されます。ただし、Client 1 のセッション期間中、Client 2 も Server A 上の ShopCart にアクセスできます。けれども Client 1 は Server B 上の ShopCart の使用に影響はありません。これによって、セッション情報を分散しないアプリケーションコンポーネントに関するステートおよびセッション情報の整合性が保持されます。

この節では、次のトピックについて説明します。

- 固定ロードバランスの適用
- 固定ロードバランスの有効化

固定ロードバランスの適用

固定ロードバランスは、依存関係を持つが、分散環境で実行されるアプリケーションコンポーネントを実行するときに必要になります。このようなアプリケーションコンポーネントには通常、次の特性があります。

- 特定のマシン上で動作するように作成されている
- 正しく動作するために、セッション情報を必要とする
- iPlanet Application Server 環境で実行するときは、再作成せずにラップする

たとえば、使用頻度の高い既存のアプリケーションは、iPlanet Application Server に移植して実行します。使用頻度の高い既存のアプリケーションは、複数の iPlanet Application Server マシンに分散してより使いやすくします。ユーザが固定アプリケーションコンポーネントを起動するリクエストを送信すると、ロードバランスサービスはそのリクエストを処理する iPlanet Application Server マシンを決定します。iPlanet Application Server サーバが選択されると、固定アプリケーションコンポーネントを使う後続のリクエストは、すべてその iPlanet Application Server サーバによっ

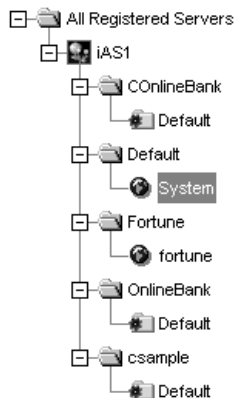
て処理されます。このサーバに多数のユーザリクエストが集中した場合、ロードバランスは新しいリクエストを他の iPlanet Application Server マシンに転送し、そのサーバが新しいセッションリクエストをすべて処理します。これにより、ロードバランスの効率が保持されます。

固定ロードバランスの有効化

複数の iPlanet Application Server マシンを配置している場合、特定のアプリケーションコンポーネントのセッションおよびステート情報を分散できないときは、固定ロードバランス機能を有効にします。

固定ロードバランスを有効にするには、次の手順を実行します。

1. iASAT ツールバーの「アプリケーション」をクリックして「アプリケーション」ウィンドウを開きます。
2. 「アプリケーション」ウィンドウの左側のペインで、固定ロードバランを有効にするサーバを選択します。
3. 固定ロードバランスを有効にするアプリケーションコンポーネントが含まれているアプリケーショングループを開きます。



4. 「アプリケーション」ウィンドウの右側のペインで、固定ロードバランスを有効にするアプリケーションコンポーネントを選択します。
5. 「Sticky LB」カラムで、選択したアプリケーションコンポーネントのチェックボックスをオンにします。

Components		Roles	
Servlets in Module System			
Servlets	Enabled	Mode	Sticky LB
System_FormAuthServlet	<input checked="" type="checkbox"/>	Local	<input type="checkbox"/>
System_CertAuthServlet	<input checked="" type="checkbox"/>	Local	<input type="checkbox"/>
System_JSPRunnerSticky	<input checked="" type="checkbox"/>	Local	<input type="checkbox"/>
System_StaticServlet	<input checked="" type="checkbox"/>	Distributed	<input checked="" type="checkbox"/>
System_JSPRunner	<input checked="" type="checkbox"/>	Local	<input type="checkbox"/>
System_BasicAuthServlet	<input checked="" type="checkbox"/>	Global	<input checked="" type="checkbox"/>

選択したコンポーネントの固定ロードバランスが有効になります。

6. 固定ロードバランスを有効にするアプリケーションコンポーネントごとに、手順 4 ~ 5 を繰り返します。
7. 「Toggle Sticky LB」をクリックすると、すべての「Sticky LB」チェックボックスが選択 または 選択解除されます。

固定ロードバランスと外部 Sticky cookie

通常、Web コネクタは、iPlanet レジストリ内で専用に配置されている iPlanet Application Server インスタンスへのリクエストのディスパッチのみを行うことができます。たとえば、LOCAL に設定されたアプリケーションコンポーネントのリクエストは、次の iPlanet レジストリキーに設定されたインスタンスにディスパッチされます。

Software/iPlanet/Application Server/6.5/CCSO/HTTPAPI/GXIP

DISTRIBUTED に設定されたアプリケーションコンポーネントのリクエストは、次の iPlanet レジストリキーでターゲットコンポーネント用に指定されている iPlanet Application Server インスタンスにディスパッチできます。

Software/iPlanet/Application Server/6.5/GDS/subtreemaps

つまり、ディスパッチ可能なインスタンスは Web コネクタのレジストリから解釈される必要があります。しかし、固定 cookie セットとともにリクエストが届いたときの Web コネクタの選択はあまり明確ではありません。Web コネクタが固定 cookie を持つリクエストを処理するときは、固定ターゲット指定値に基づいてリクエストをディスパッチします。正しく計画的に設定されている場合は、Web コネクタは未知の固定ターゲットのリクエストを正しくディスパッチできます。たとえば、アプリケーションサーバの上位にロードバランス層が存在する場合、Web コネクタは、以前に接続し

たことがないターゲットアプリケーションサーバの固定リクエストを受け取ることができます。また、Web コネクタが認識していないインスタンスの固定ターゲット指定を無視することもできます。Web コネクタが固定ターゲット指定を許可するインスタンスのセットは、同じマルチキャストグループの構成要素のインスタンスによって決まります。

ロードバランス方式の選択

サーバのロードバランスを有効にするときは、ロードバランス方式を選択する必要があります。ロードバランスの制御方法は、「だれ」がロードバランスを制御するかによって異なります。つまり、サーバ自体がロードバランスを制御する方式と、Web コネクタプラグインが制御する方式があります。

この節では、次のトピックについて説明します。

- Web コネクタプラグインによるロードバランス
- iPlanet Application Server によるロードバランス

Web コネクタプラグインによるロードバランス

Web コネクタプラグインがロードバランスを制御する場合、次の方法を選択できます。

- コンポーネントの応答時間 (デフォルト)

Web コネクタプラグインは、アプリケーションコンポーネントの応答時間を測定して、アプリケーションリクエストの転送先を決定します。転送先の決定はデフォルトで選択されます。

- サーバあたりの応答時間

Web コネクタプラグインは、サーバの応答時間を測定してアプリケーションの転送先を決定します。コンポーネントの応答時間を使用した場合よりもオーバーヘッドが減少します。

- ラウンドロビン

リクエストは、ユーザが指定したウェイト方式に基づいてサーバ全体に分散されます。

プラグインは、ユーザが指定したウェイトに基づいて、iPlanet Application Server マシンにリクエストを分散します。このロードバランスオプションは、ユーザが指定するウェイトだけにに基づき、コンポーネントまたはサーバ応答時間に関するデータの収集を必要としないので、オーバーヘッドが発生しません。

iPlanet Application Server によるロードバランス

iPlanet Application Server がロードバランスを制御する場合、このアプリケーションサーバは、CPU 負荷やディスク I/O などのハードウェアリソースプロファイルと、結果キャッシュや Servlet 実行速度などのリクエスト実行プロファイルを使って、リクエストの負荷を均衡させます。サーバおよびリクエストの統計は、マルチキャストを介してクラスタ内の別の iPlanet Application Server マシンと交換されます。マルチキャストは、管理者が詳細に制御できるため、複雑なシナリオに適しています。

注 このロードバランス方式は、設定が最も難しく、パフォーマンスが向上するとは限りません。まず、Web コネクタプラグイン方式を試すことをお勧めします。

コンポーネントの応答時間によるロードバランス

コンポーネントの応答時間による方式は、特定のアプリケーションに対する iPlanet Application Server マシンの平均応答時間の測定に基づいて行われます。

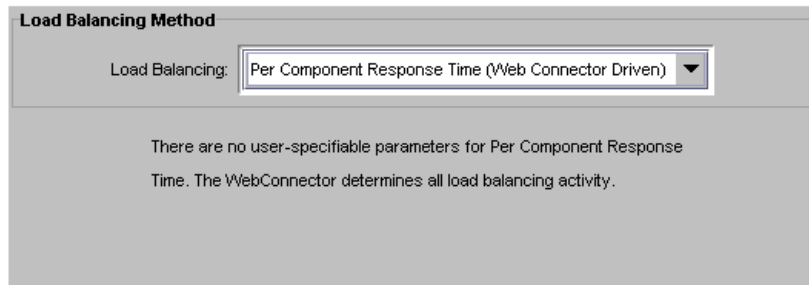
コンポーネント方式では、Web コネクタプラグインがロードバランスをより豊かかつ詳細に制御することができます。ただし、コンポーネント方式では、サーバ方式よりオーバーヘッドが若干増加します。コンポーネント方式は、アプリケーションコンポーネントの応答時間がパフォーマンス特性によりサーバごとに大きく異なる場合に適しています。

コンポーネントの応答時間によるロードバランスを有効にするには、次の手順を実行します。

1. iASAT ツールバーで「ロードバランス」をクリックして、「ロードバランス」ウィンドウを開きます。
2. 左側のペインで、ロードバランス方式を指定するサーバを選択します。



3. 「ロードバランス」ドロップダウンボックスで「コンポーネントあたりのレスポンス時間 (Web コネクタ駆動)」を選択し、Web コネクタプラグインがコンポーネント応答時間の統計に基づいてロードバランスを制御するように指定します。これはデフォルトです。



4. 「変更の適用」をクリックして設定を保存します。

サーバの応答時間によるロードバランス

サーバの応答時間による方式は、iPlanet Application Server マシンが処理するアプリケーションコンポーネント全体の平均応答時間の測定に基づいて行われます。

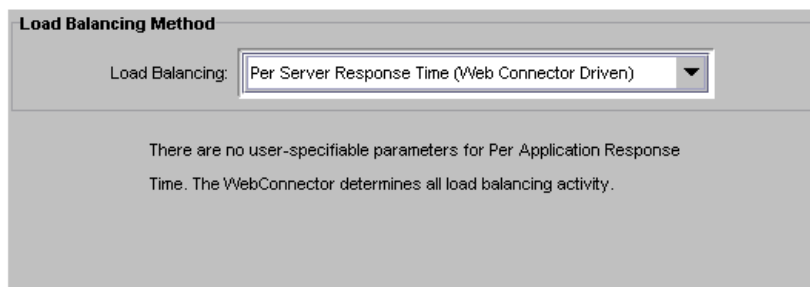
サーバ方式は、アプリケーションコンポーネントの応答時間がサーバごとに類似している場合に適しています。

サーバの応答時間によるロードバランスを有効にするには、次の手順を実行します。

1. iASAT ツールバーで「ロードバランス」をクリックして、「ロードバランス」ウィンドウを開きます。
2. 左側のペインで、ロードバランス方式を指定するサーバを選択します。



3. 「ロードバランス」ドロップダウンボックスで「サーバあたりのレスポンス時間 (Web コネクタ駆動)」を選択し、Web コネクタプラグインがサーバ応答時間の統計に基づいてロードバランスを制御するように指定します。



4. 「変更の適用」をクリックして設定を保存します。

ラウンドロビンロードバランス

ラウンドロビンロードバランス方式を選択するときは、ラウンドロビンロードバランスに加わる各 iPlanet Application Server マシンのウェイトを指定する必要があります。ウェイトのデフォルト値は、変更しないかぎり、1 です。すべてのサーバのウェイトが等しい場合、ラウンドロビンロードバランスは、同数のリクエストを各サーバに送信します。サーバのパフォーマンスが異なる場合は、ウェイトシステムを使用する必要があります。パフォーマンス特性が異なる 4 台のマシンをラウンドロビンに含めた場合、パフォーマンスの高いマシンにより多くのリクエストを割り当てる必要があります。これを実行するには、各 iPlanet Application Server マシンにウェイトを割り当てます。4 台の iPlanet Application Server マシンに対して、8 リクエストごとに次のウェイトを割り当てたと想定します。

マシン 1 = 4

マシン 2 = 2

マシン 3 = 1

マシン 4 = 1

この場合、4 リクエストがマシン 1 に、2 リクエストがマシン 2 に、以下同様に転送されます。リクエスト数を詳細に制御するには、1,000 個のうち何個のリクエストがサーバに割り当てられるかというように想定してください。たとえば、ウェイトを 135、270、および 595 に指定すると、サーバに送信されるリクエスト数の精度が増します。

ラウンドロビンロードバランスを設定するには、次の手順を実行します。

1. iPlanet レジストリエディタを開きます。

エディタが開き、iPlanet Application Server に適用されるキーと値が表示されます。

2. 次のキーを 1 に設定します。

SOFTWARE¥iPlanet¥Application Server¥6.5¥CCSO¥Loadb¥RoundRobin



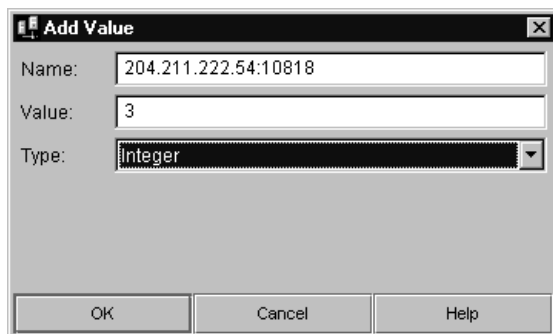
3. 次のキーを強調表示します。

SOFTWARE¥iPlanet¥Application Server¥6.5¥CCSO¥Loadb¥ServerWeights

4. 「編集」 から 「値を追加」 を選択します。

「値を追加」 ダイアログボックスが開きます。

5. ラウンドロビンロードバランスに含める各 iPlanet Application Server マシンの「Name」 (IP アドレスとポート番号) および 「Value」 (ウェイト) を入力し、「Type」 を 「Integer」 に設定します。

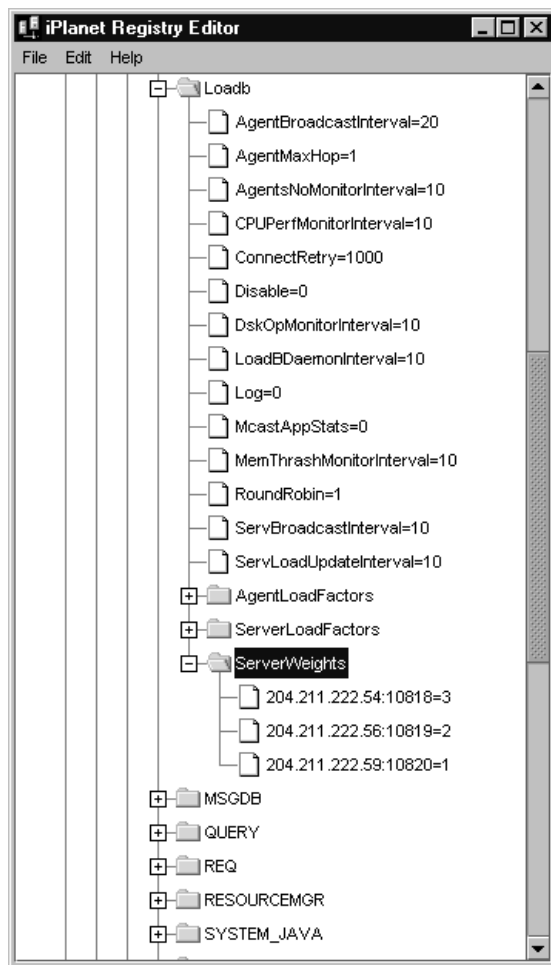


たとえば、次の IP アドレスを持つ 3 台の iPlanet Application Server マシン (KXS) を想定します。

- a. 204.211.222.54:10818
- b. 204.211.222.56:10819
- c. 204.211.222.59:10820

次の値を割り当てます。

```
SOFTWARE%iPlanet%Application Server%6.5%CCSO%LoadB%ServerWeights  
204.211.222.54:10818=3  
204.211.222.56:10819=2  
204.211.222.59:10820=1
```



このウェイト方式では、Web コネクタプラグインは、6 リクエストごとに3 リクエストをポート 10818 に、2 リクエストをポート 10819 に、1 リクエストをポート 10820 に転送します。

6. 保存してエディタを閉じます。

ユーザ定義条件ロードバランス

Web コネクタプラグインではなく、iPlanet Application Server が企業のロードバランスを制御する場合、ロードバランスサービスは、「サーバ負荷」または「アプリケーションコンポーネントクライテリア」に指定されたウェイト要因に基づいて、リクエストを処理する iPlanet Application Server マシンを決定します。これらの要因は、iASAT の「ロードバランス」ウィンドウを使って設定します。ウェイト要因を決定するときは、アプリケーションを適切に実行するために重要な要因を判断する必要があります。

iASAT のウェイト要因は、iPlanet Application Server マシン上で動作する標準的なアプリケーションを基にしたデフォルト値に設定されています。特定のアプリケーションを最適化するには、「サーバ負荷クライテリア」または「アプリケーションコンポーネントクライテリア」のウェイト要因を調整します。

この節では、次のトピックについて説明します。

- 「サーバ負荷クライテリア」のウェイト要因の調整
- アプリケーションコンポーネントパフォーマンスクライテリアのウェイト要因の調整
- 更新およびブロードキャスト間隔の調整
- ロードバランスのマルチキャストホストアドレスの変更

「サーバ負荷クライテリア」のウェイト要因の調整

「サーバ負荷」値は、iPlanet Application Server がユーザリクエストを処理しているときに、マシンにかかる負荷を定量化したものです。「サーバ負荷」値は、各 iPlanet Application Server 内のロードバランスサービスによって、マシンごとに計算されます。「サーバ負荷クライテリア」のウェイト要因を調整すると、複数の iPlanet Application Server マシン間のアプリケーションリクエストの分散をシステムリソースに基づいて最適化することができます。

「サーバ負荷」値は、アプリケーションコンポーネントパフォーマンス値を計算するときの条件の 1 つとして使われます。「サーバ負荷クライテリア」について次の表で説明します。

表 14-1 サーバ負荷クライテリア

サーバ負荷クライテリア	説明
CPU 負荷	コンピュータ内のすべてのプロセッサが使われる時間の平均パーセンテージ

表 14-1 サーバ負荷クライテリア (続き)

サーバ負荷クライテリア	説明
ディスク I/O	システムが読み書き操作をハードディスクドライブに発行する速度
メモリスラッシュ	参照時にメモリになかったページへのメモリ参照を解決するために、ハードディスクドライブから読み込まれたまたはハードディスクドライブに書き込まれたページ数
キューされるリクエストの数	サーバが現在処理しているユーザリクエストとアプリケーションリクエストの数
サーバの応答時間	すべてのアプリケーションコンポーネントに対する特定のサーバの平均応答時間

各「サーバ負荷クライテリア」は、設定したウェイト要因の積が計算されます。その値平均が最終的な「サーバ負荷」値となります。「サーバ負荷」値は、アプリケーションコンポーネントパフォーマンス条件の1つとして使われます。

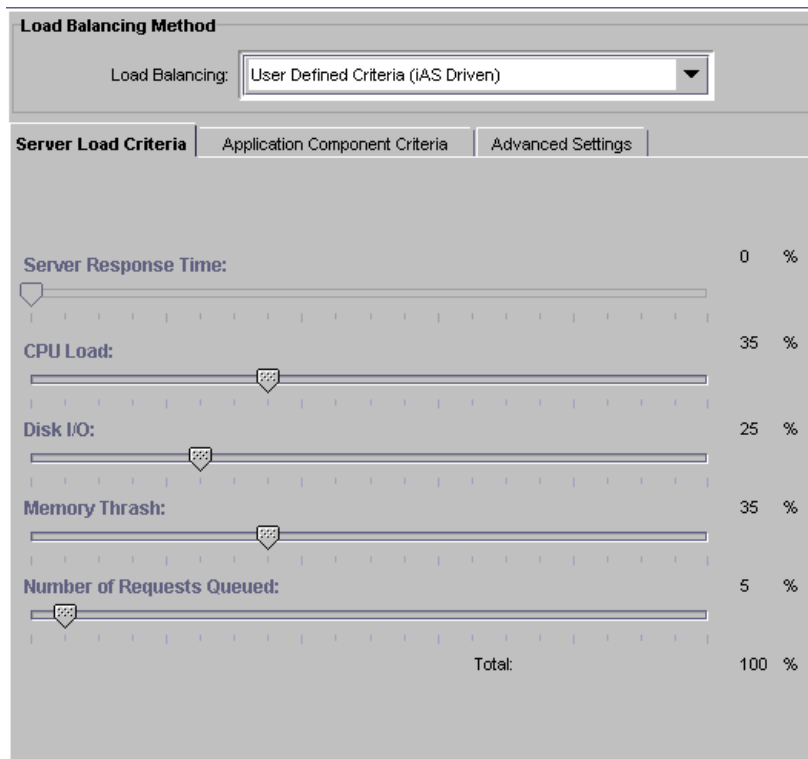
「サーバ負荷クライテリア」のウェイト要因を調整するには、次の手順を実行します。

1. iASAT ツールバーで「ロードバランス」をクリックして、「ロードバランス」ウィンドウを開きます。
2. 左側のペインで、ウェイト要因を調整するサーバを選択します。



3. 「ロードバランス」ドロップダウンボックスから「ユーザ定義クライテリア (iAS 駆動)」を選択し、サーバがロードバランスを制御するように指定します。

企業の要件に応じて、ウェイト要因を調整します。「サーバ負荷クライテリア」タブをクリックすると、次のウィンドウが表示されます。



- 「ロードバランス」ウィンドウの右側のペインで、スケールマーカーをスライドして各条件のウェイト要因を調整します。条件の詳細については、277 ページの「サーバ負荷クライテリア」のウェイト要因の調整」を参照してください。

すべてのウェイト要因の合計は 100 になる必要があります。

- 以上の手順を終了したら、「変更の適用」をクリックして設定値を保存します。

アプリケーションコンポーネントパフォーマンスクライテリアのウェイト要因の調整

アプリケーションコンポーネントパフォーマンス値は、iPlanet Application Server マシン上で実行されるアプリケーションコンポーネントのパフォーマンスを示します。アプリケーションコンポーネントパフォーマンス値は、ロードバランスに含めるアプリケーションコンポーネントごとに計算されます。ロードバランスは、アプリケーションコンポーネント単位に行われ、分散を最適化します。

アプリケーションコンポーネントパフォーマンス値には、5つのアプリケーション条件があります。ロードバランスサービスは、各アプリケーション条件に割り当てたウェイト要因に基づいて iPlanet Application Server マシンを比較します。ウェイト要因の合計値が最大となるサーバで、そのアプリケーションコンポーネントへのリクエストが処理されます。条件について、次の表で説明します。

表 14-2 「アプリケーションコンポーネントロードバランスクライテリア」

アプリケーションコンポーネントパフォーマンスクライテリア	説明
サーバ負荷	すべての「サーバ負荷クライテリア」に対して計算された値
利用可能なキャッシュ結果	アプリケーションコンポーネントの結果をキャッシュすかどうかを示すフラグ。アプリケーションコンポーネントの結果をキャッシュすると、通常は、ユーザのリクエストの処理速度が向上します。
最低平均実行時間	アプリケーションコンポーネントを各 iPlanet Application Server マシン上で実行する時間
最も最近の実行	アプリケーションコンポーネントを最後に実行したサーバ。サーバが動作しているシステムには、アプリケーションデータがキャッシュされているため、そのコンポーネントをすぐに再実行すると実行時間は短縮されます。
最小限の実行	アプリケーションコンポーネントが iPlanet Application Server マシン上で実行された回数。ロードバランスの目的は、企業内のすべてのサーバにリクエストを均一に分散させることです。したがって、アプリケーションコンポーネントの実行回数をもっとも少なかったサーバに対して、優先的にリクエストを割り当てます。
アプリケーションコンポーネントのレスポンス時間	特定のアプリケーションコンポーネントに対する特定のサーバからの平均レスポンス時間

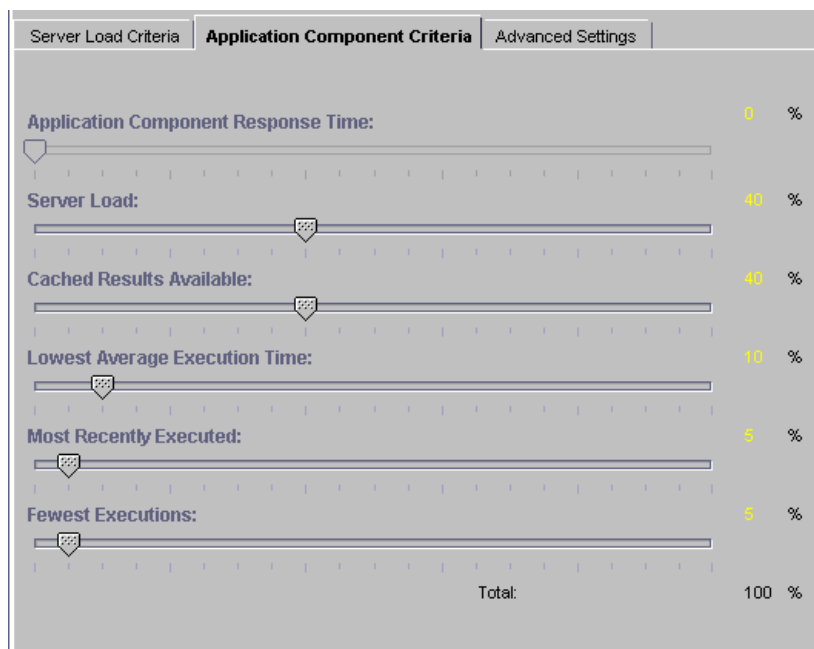
各アプリケーション条件は、ユーザが設定したウェイト要因の積が計算されます。その値の平均値が最終的なアプリケーションコンポーネントパフォーマンス値となります。ロードバランスサービスは、この値を使って、新しいユーザリクエストの処理に適した iPlanet Application Server マシンを決定します。

アプリケーションコンポーネントパフォーマンスクライテリアのウェイト要因を調整するには、次の手順を実行します。

1. iASAT ツールバーで「ロードバランス」をクリックして、「ロードバランス」ウィンドウを開きます。
2. 左側のペインで、ウェイト要因を調整するサーバを選択します。



3. 「ロードバランス」ドロップダウンボックスから「ユーザ定義クライテリア (iAS 駆動)」を選択し、サーバがロードバランスを制御するように指定します。
企業の要件に応じて、ウェイト要因を調整します。
4. 「アプリケーションコンポーネントクライテリア」タブをクリックします。
次のウィンドウが表示されます。



5. 「ロードバランス」ウィンドウの右側のペインで、スケールマーカーをスライドして各条件のウェイト要因を調整します。
すべてのウェイト要因の合計は 100 になる必要があります。
6. 以上の手順を終了したら、「変更の適用」をクリックして設定値を保存します。

更新およびブロードキャスト間隔の調整

iPlanet Application Server マシンが「サーバ負荷」およびアプリケーションコンポーネントパフォーマンス条件を更新する間隔を設定することができます。これらの値が頻繁かつ大幅に変化する場合は、間隔を調整して更新頻度を高くします。ただし、値を頻繁に更新すると、iPlanet Application Server マシンの負荷が増加します。条件値の更新頻繁が高くない場合は、更新期間を大きくするとサーバリソースを節約できます。

この理論はブロードキャスト間隔の設定にも当てはまります。条件値が頻繁かつ大幅に変化する場合は、ブロードキャスト間隔を短くして、サーバを頻繁に更新します。サーバを頻繁に更新するとネットワークトラフィック負荷が増すため、最適な値に調整することが重要です。

ブロードキャスト間隔および更新間隔は、「ベースブロードキャスト / 更新間隔」に関連付けられています。「ベースブロードキャスト / 更新間隔」は、ロードバランスサービスが「起動」し、更新を行い、更新を受けとったかどうかをチェックし、新しい値をブロードキャストするときの間隔です。

ブロードキャスト間隔および更新間隔は、ベース間隔の偶数倍になっており、ロードバランスサービスが「起動」すると呼び出されます。つまり、ベース間隔が 300 秒で、「サーバ負荷」および「アプリケーションコンポーネントクライテリア」のブロードキャスト間隔がそれぞれ 900 秒の場合、これらの値は、ロードバランスサービスが 3 回起動するたびに 1 回ブロードキャストされます。残り 2 回のロードバランスサービスの起動時には、ロードバランスサービスは、各ブロードキャストの間に、他の iPlanet Application Server マシンから更新を受信したかどうかに基づいて分散順序を再評価します。

次の表では、各エンティティの更新間隔およびブロードキャスト間隔の設定方法について説明します。

表 14-3 「ブロードキャスト間隔」

間隔の設定項目	説明
ベースブロードキャスト / 更新間隔	ロードバランスサービスが「起動」する間隔
アプリケーションコンポーネントクライテリア	ロードバランスサービスが「アプリケーションコンポーネントパフォーマンス」値をブロードキャストする間隔
サーバ負荷クライテリア	ロードバランスサービスが「サーバ負荷クライテリア」値をブロードキャストする間隔
サーバ負荷	ロードバランスサービスが「サーバ負荷」値を更新する間隔
CPU 負荷	ロードバランスサービスが「CPU 負荷」値を更新する間隔

表 14-3 「ブロードキャスト間隔」(続き)

間隔の設定項目	説明
ディスク I/O	ロードバランスサービスが「ディスク I/O」値を更新する間隔
メモリスラッシュ	ロードバランスサービスが「メモリスラッシュ」値を更新する間隔
キューされるリクエストの数	ロードバランスサービスが「キューされるリクエストの数」を更新する間隔
最大ホップ数	リクエストがサーバ間で交換される最大回数

間隔を更新およびブロードキャストするには、次の手順を実行します。

1. iASAT ツールバーで「ロードバランス」をクリックして、「ロードバランス」ウィンドウを開きます。
2. 「ロードバランス」ウィンドウの左側のペインで、高度な設定を調整するサーバを選択します。



3. 「ロードバランス」ドロップダウンボックスから「ユーザー定義クriteria (iAS 駆動)」を選択し、サーバがロードバランスを制御するように指定します。
4. 「高度な設定」タブをクリックします。
次のウィンドウが表示されます。

Load Balancing Method

Load Balancing: User Defined Criteria (IAS Driven) ▼

Server Load Criteria Application Component Criteria **Advanced Settings**

Base Broadcast/Update Interval: seconds

Broadcast Intervals

Server Load: seconds

Application Component Criteria: seconds

Update Intervals

Server Load: seconds

CPU Load: seconds

Disk I/O: seconds

Memory Thrash: seconds

Number of Requests Queued: seconds

Maximum Hops:

5. 「ロードバランス」ウィンドウの右側のペインで、各間隔パラメータの時間を基準時間の倍数で設定します。
6. 「最大ホップ数」テキスト領域に、アプリケーションコンポーネントがサーバ間で交換される最大回数を指定します。
7. 以上の手順が終了したら、「変更の適用」をクリックして変更を保存します。

ロードバランスのマルチキャストホストアドレスの変更

都市間ネットワークなど、ネットワーク間のアプリケーション負荷を均衡させるには、マルチキャストサーバのホストアドレスおよびポート番号を変更します。ネットワーク内では、衝突が発生しないかぎり、デフォルトアドレスを変更する必要はありません。

マルチキャストホストアドレスを変更するには、次の手順を実行します。

1. iPlanet レジストリエディタを起動します。

2. 次のキーを開きます。
SOFTWARE¥iPlanet¥Application Server¥6.5¥GMS¥KES
3. MCastHost 文字列値をダブルクリックします。
「文字列値」編集ダイアログボックスが表示されます。
4. 値データとして、新しいホストの IP アドレスを指定し、「OK」をクリックします。
5. MCastPort DWORD 値をダブルクリックします。
「DWORD」編集ダイアログボックスが表示されます。
6. 値データとして、新しいホストのポート番号を指定し、「OK」をクリックします。
7. エディタを閉じます。
新しいマルチキャストアドレスが有効になります。

分散のデータ同期の管理

この章では、iPlanet Application Server インスタンスをデータの同期化に参加するクラスタにグループ化する方法について説明します。

この章には次のトピックがあります。

- 分散のデータ同期について
- フェールオーバーがデータアクセスを維持する方法
- クラスタについて
- クラスタの通信方法
- クラスタの設定
- 優先度を設定した Sync Server の調整

分散のデータ同期について

分散データを同期化することによって、複数の iPlanet Application Server インスタンス間で共有されるステートおよびセッション情報の完全性を保持します。これは、複数の iPlanet Application Server インスタンスで管理されるパーティション分割された分散アプリケーションにとって重要です。

ほとんどの企業では、複数の iPlanet Application Server インスタンスで、1つ以上の分散アプリケーションをサポートしています。分散アプリケーションが機能するには、各サーバがステートやセッション情報などのアプリケーション関連情報にアクセスする必要があります。

これらの情報は、iPlanet Application Server に組み込まれているシステムレベルの分散データ同期サービスを介して分散されます。

フェールオーバーがデータアクセスを維持する方法

分散データシンクロナイザはシステムレベルのサービス 1 つで、アプリケーションのセッション情報など、複数の iPlanet Application Server インスタンス間の分散データの完全性と耐障害性を制御します。

各 iPlanet Application Server インスタンスは、次の 4 つのエンジンから構成されています。

- **Administrative Server(KAS) - Administrative Server** が他のエンジンを起動および監視し、障害が発生したエンジンが確実に再起動されるようにします。
- **Executive Server(KXS) - 1 つの Executive Server** だけが iPlanet Application Server クラスタ内のプライマリ同期エンジン (シンクロナイザ) になります。

iPlanet Application Server インスタンスのクラスタ内では、いずれかの Executive Server が分散 (同期) 情報を管理し、クラスタに登録されている他のすべてのサーバのサーバロールを設定します。クラスタ内のすべてのエンジンは、このプライマリエンジンおよびその情報にアクセスできます。

- **Java サーバ (KJS) - ないこともあります。**
- **C++ サーバ (KCS) - ないこともあります。**

KJS または KCS に障害が発生すると、KAS はそのエンジンを再起動します。KXS に障害が発生した場合は、次の対応策を適用します。

- KXS のバックアップに現在の適切なロールを割り当てます。このロールは、クラスタ内にある KXS エンジンを考慮して決定されるため、以前のロールと異なることがあります。
- KJS および KCS エンジンを停止します。
- KJS および KCS エンジンのバックアップを作成します。

クラスタについて

クラスタは、iPlanet Application Server インスタンスのグループで、各ユーザセッションのステートに関連する情報を共有しています。クラスタを使用すると、信頼性が向上し、プロセスおよびハードウェアに障害が発生した場合でもユーザセッションを継続することができます。また、すべての利用可能なリソースに対して、より効率的に負荷を分散することができます。

クラスタ内のサーバは、同一ネットワーク内だけでなく、異なるネットワーク内、または同一ネットワーク内の異なるサブネットに配置することができます。共通のサブネット上のクラスタ内に各サーバを配置すれば、パフォーマンスが向上します。クラスタ内のメンバーは、TCP/IP を介して通信します。つまり、クラスタ内のサーバの IP アドレスおよびポート番号を使用して、各インスタンスの KAS および KXS エンジンにアクセスします。この通信が有効なときにのみ、クラスタを操作することができます。この通信方法の詳細は、116 ページの「ファイアウォールを使ったセキュリティ」を参照してください。

クラスタ内の、ステートおよびセッション情報は、Primary Sync Server の KXS プロセスのメモリに保存されます。クラスタの設定データは、Directory Server に保存されます。クラスタ内のすべての iPlanet Application Server インスタンスは、1 つの Directory Server を共有することができます。クラスタ内の iPlanet Application Server インスタンスが 1 つの Directory Server を共有しない場合は、各サーバが同一のクラスタ情報にアクセスできるように、クラスタ設定を Directory Server 間からコピーする必要があります。

この節では、次のトピックについて説明します。

- データ同期の設定
- クラスタ内でのサーバロール

データ同期の設定

サーバ間のデータ同期を設定するには、クラスタ内でサーバが実行するロールを決定する必要があります。次に、各クラスタエントリを編集して、サーバロールを設定し、クラスタをシンクロナイザサービス登録します。最後に、サーバロールに指定した順序で各 iPlanet Application Server を起動します。

クラスタ内でのサーバロール

クラスタ内起動のサーバロールを、Sync Alternate または Sync Local に設定します。最初に起動するサーバは Primary Sync Server になります。次に起動したときに変わります。2つのバックアップサーバを定義した場合は、Primary Sync Server の後に起動した2つのサーバに対して、Backup Sync Server のロールが割り当てられます。

定義済みのバックアップサーバをすべて起動した後で、次のサーバを起動すると Sync Alternate が割り当てられます。Primary および Backup に障害が発生すると、このサーバにロールが引き継がれます。

クラスタ内のサーバが、割り当てられたロールを実行しながら、相互に調整する方法の詳細については、「例:7台のサーバで構成されるクラスタ内の調整」を参照してください。

クラスタ内のサーバのロールについて、次の表で説明します。

表 15-1 クラスタ内のデータ同期のロール

サーバロール	説明
Sync Server	<p>任意の iPlanet Application Server インスタンスを、Sync Primary として定義できる。Sync Server は Sync Primary、Sync Backup、および Sync Alternate で構成される。</p> <p>すべての Sync Server は、iPlanet レジストリの SyncServers キーに記述される。</p>
Sync Primary	<p>プライマリデータストアが格納されるサーバ。クラスタ内の他のすべてのメンバーは、最新の分散データ情報を取得するためにこのサーバと通信する。</p> <p>クラスタ内で最初に起動する iPlanet Application Server が Sync Server となる。また、最初に起動した Sync Server が、Sync Primary となる。</p>
Sync Backup	<p>Sync Server の数。最大 MaxBackups 個まで。Sync Primary の情報をミラー化する。Sync Backup を増やすたびにクラスタの負荷が増加するので、割り当てるバックアップ数を決めるときは、安全性とパフォーマンスのバランスを考慮する必要がある。</p> <p>Sync Primary にアクセスできなくなった場合は、他の Sync Backup と比較して優先度のもっとも高い (最小整数値) Sync Backup が次の Sync Primary になる。</p>

表 15-1 クラスタ内のデータ同期のロール (続き)

サーバロール	説明
Sync Alternate	<p>iPlanet レジストリの SyncServers キーに指定されているサーバ。Sync Backup になることができる。Sync Backup の数が、設定した最大数を下回ると、他の Sync Alternate と比較して優先度のもっとも高い Sync Alternate が Sync Backup に昇格する。</p> <p>各 Sync Alternate は Sync Backup に昇格するまで、Sync Local と同様に動作する。</p>
Sync Local	<p>データ同期サービスを使用するサーバ。ただし、Sync Primary、Sync Backup、または Sync Alternate になることはできない。Sync Local はすべての分散データの使用、作成、および破棄できるが、そのデータを保持する機能をもたない</p> <p>Sync Local は iPlanet レジストリの SyncServers キーには指定されていない。ただし、クラスタ内にあるすべてのレジストリの SyncServers リストには、クラスタ内の各 Sync Server の識別および優先度の情報が記述されている</p> <p>各 Sync Local は、iPlanet レジストリの SyncServers キー内ですべてのサーバを照会し Sync Primary を検出すると、クラスタ内で有効になる。Sync Local が iPlanet レジストリの SyncServers キーから Sync Primary を検出できなかった場合はクラスタが停止していると見なし、ローカルサーバとして動作する</p> <p>Sync Local は Sync Primary とだけ通信し、クラスタ内の他のサーバからは Sync Local を認識されない</p>

クラスタの通信方法

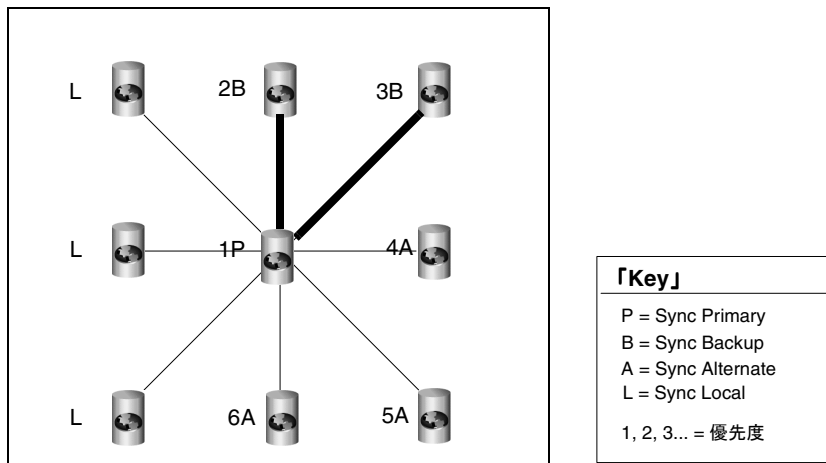
クラスタ内のサーバは、相互に通信する必要があります。この通信を有効にするには、各サーバがそれぞれ属するクラスタを識別する必要があります。iPlanet Application Server インスタンスのシンクロナイザをクラスタに割り当てると、iPlanet Application Server インスタンスはクラスタの一部として有効になります。この手順については、309 ページの「クラスタへのシンクロナイザの割り当て」で説明します。

アプリケーションコンポーネントが分散データソースに書き込みアクセスを要求すると、書き込みは最初に Sync Primary 上で発生します。Sync Primary 上のデータが変更されると、Sync Primary はただちに Sync Backup を更新します。

各 iPlanet Application Server インスタンスに対するシンクロナイザの割り当ては、1 つのクラスタに対して 1 度だけ行うことができます。

クラスタ内の情報の流れ

Sync Backup、Sync Alternate、Sync Local、および Sync Primary の通信は、次の図のように星形構成になっています。



この図では、すべてのサーバが Sync Primary と通信しています。Sync Local には、優先度番号が割り当てられていません。

この図は、起動したばかりのクラスタを表しています。フェールオーバーが発生していないうえ、優先度番号は現在割り当てられているロールと一致しています。

クラスタの設定

クラスタの設定および管理は、iPlanet Application Server Administration Tool (iASAT) を使って行います。iASAT は、スタンドアロンのグラフィカルユーザインタフェースツールです。このツールを使うことにより、iPlanet Application Server のあらゆる機能を管理できます。iASAT を使ってクラスタを設定すると、関連するバックエンドエントリが iPlanet レジストリに作成されます。これらのエントリは、確認および編集することができます。

クラスタの設定は、iPlanet レジストリエディタを使って、iPlanet レジストリ内で直接行うこともできます。ただし、クラスタの作成と管理は、「iASAT」を使うことをお勧めします。

次の節では、iASAT および iPlanet レジストリエディタを使って、クラスタを作成および管理する方法について説明します。

- iASAT を使ったクラスタの設定
- iPlanet レジストリエディタを使ったクラスタの設定
- Sync Server の優先度の決定
- クラスタパラメータの設定
- クラスタへのシンクロナイザの割り当て

iASAT を使ったクラスタの設定

iPlanet Application Server をインスタンスにインストールすると、ホストサーバは、1つのノードを持つデフォルトのクラスタとして自動的に認識されます。他の登録済み iPlanet Application Server インスタンスを既存のクラスタ (ホストサーバ) に追加したり、新しいクラスタを作成したりすることができます。

この節では、次のトピックについて説明します。

- 新しいクラスタを作成するには
- クラスタにサーバを追加するには
- クラスタからサーバを削除するには

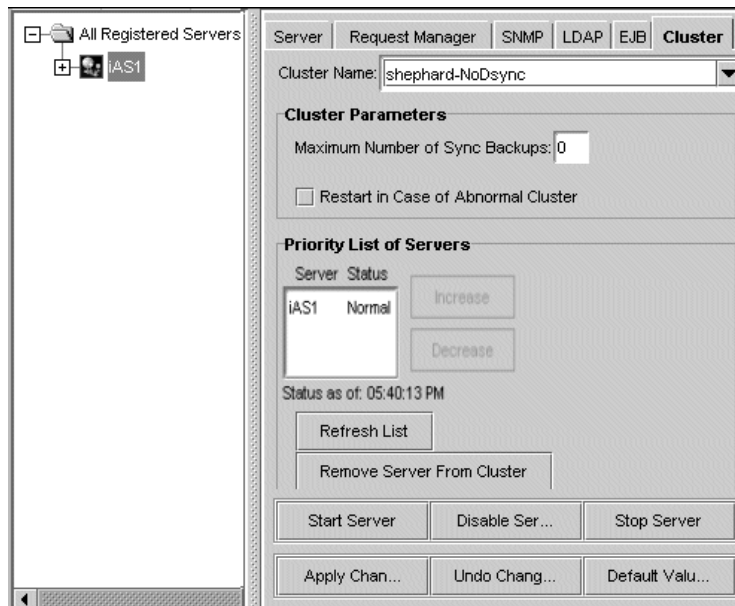
注 iASAT では、特定のサーバを選択して停止すると、ほかのすべてのサーバについても「サーバの停止」ボタンがグレー表示になります。ただし、これはほかのサーバも停止したという意味ではなく、ほかのサーバは動作を続けます。

クラスタ内のほかのサーバのどれかを停止するには、「一般」タブに移動し、左側のペインでサーバを選択して、「サーバの停止」をクリックします。

新しいクラスタを作成するには

iASAT を使って新しいクラスタを作成するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックして「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、新しいクラスタを作成する iPlanet Application Server インスタンスを選択します。
3. 「一般」ウィンドウの右側のペインで、次の図に示すように、「クラスタ」タブをクリックします。



4. 「クラスタ名」ドロップダウンリストで、Backspace キーまたは Delete キーを使用して、デフォルトのクラスタ名を削除します。

5. 新しいクラスタの名前を入力し、Enter キーを押します。新しいクラスタに対して任意の一意名を選択します。
6. 「変更の適用」をクリックします。新しいクラスタ設定を有効にするには、サーバを再起動する必要があります。

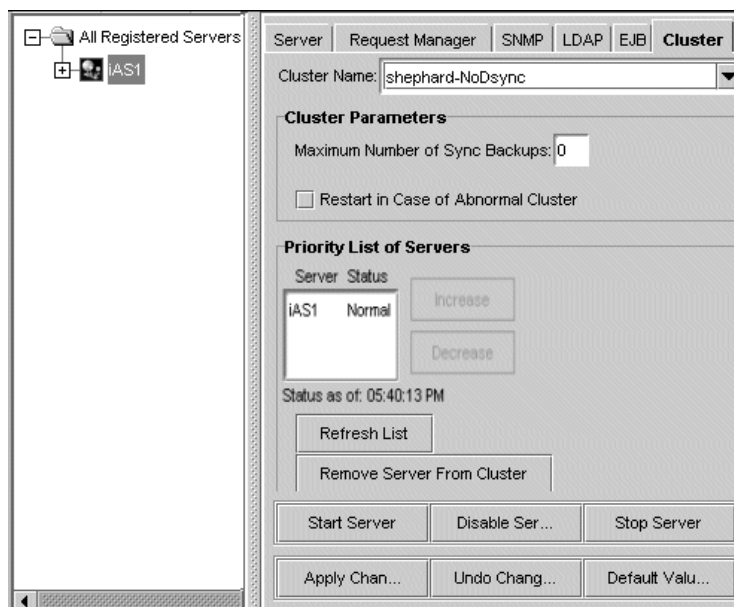
サーバを再起動した後で、次の節で説明するように、新しいクラスタに対して他の iPlanet Application Server インスタンスを追加できます。

クラスタにサーバを追加するには

iASAT を使って、割り当てられていないサーバをクラスタに追加したり、特定のサーバを別のクラスタに割り当て直すには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックし、「一般」ウィンドウを開きます。
2. 「クラスタ」タブをクリックします。

次のウィンドウが表示されます。



すべての登録済みサーバが「一般」ウィンドウの左側のペインに表示されます。既存のクラスタに属しているサーバは、「一般」ウィンドウの右側のペインにある「サーバの優先度リスト」ボックスに表示されます。表示されたサーバは、クラスタ内の優先度順にソートされています。

「サーバの優先度リスト」ボックスには、サーバのクラスタの状態も表示されます。サーバの状態には、「Normal」、「Dual Primary」、または「No Primary」があります。「リストを更新」ボタンをクリックすると、「サーバの優先度リスト」がただちに更新されます。デフォルトでは、このボックスは 15 秒ごとに更新されます。

3. 「一般」ウィンドウの左側のペインで、クラスタに追加するサーバの名前をクリックします。
4. 「クラスタ名」ドロップダウンボックスから、サーバを追加するクラスタの名前を選択します。
「クラスタ名」ドロップダウンリストには、登録済みサーバが属するクラスタ名がすべて表示されます。
5. 「変更の適用」をクリックします。
6. 追加したサーバを含め、クラスタ内のすべてのサーバを停止して再起動します。

注 クラスタからサーバを削除して別のクラスタに追加した場合は、変更を有効にするために、両方のクラスタにあるすべてのサーバを再起動する必要があります。

クラスタからサーバを削除するには

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. iASAT ツールバーの「一般」をクリックし、「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、クラスタのメンバーから削除する iPlanet Application Server を選択します。
3. 「一般」ウィンドウの右側のペインにある「クラスタ」タブをクリックします。
すべての登録済みサーバが「一般」ウィンドウの左側のペインに表示されます。既存のクラスタに属しているサーバは、「一般」ウィンドウの右側のペインにある「サーバの優先度リスト」ボックスに表示されます。
4. 「クラスタ名」ドロップダウンリストで、サーバを削除するクラスタを選択します。「サーバの優先度リスト」ボックスに、選択されたクラスタに属するすべてのサーバが表示されます。
5. 「サーバの優先度リスト」ボックスで削除するサーバを選択し、「Remove Server from Cluster」をクリックします。
6. 「変更の適用」をクリックします。
7. 削除したサーバを含め、クラスタ内のすべてのサーバを停止して再起動します。

-
- 注
- クラスタのメンバーでないサーバは、データ同期に含まれないため、クラスタリストの「<hostname>-NoDsync」の下に表示されます。<hostname>-NoDsync リストのサーバは削除できません。
 - クラスタ名を変更するには、iASAT の「一般」ウィンドウで、「クラスタ」タブのクラスタ名を削除します。次に、クラスタの新しい名前を入力し、「変更の適用」をクリックします。
-

iPlanet レジストリエディタを使ったクラスタの設定

iASAT を使ってクラスタを設定すると、必要なバックエンドエントリが iPlanet レジストリに作成されます。これらのエントリは、iASAT を使わなくても、iPlanet レジストリに作成してクラスタを設定できます。クラスタを設定および管理するときは、この機能を使うこともできますが、iASAT を使うことを強くお勧めします。

この節では、次のトピックについて説明します。

- 新しいクラスタを作成するには
- クラスタにサーバを追加するには
- クラスタからサーバを削除するには

新しいクラスタを作成するには

iPlanet Application Server をインストールすると、デフォルトのクラスタが自動的にホストサーバを使って作成されます。クラスタを設定するときは、このデフォルトクラスタの iPlanet レジストリに作成されたバックエンドエントリを変更するのがもっとも簡単です。

デフォルトクラスタの名前は <hostname>-NoDsync です。hostname はローカルマシンの名前です。たとえば、iPlanet Application Server を *Bozo* というマシンにインストールすると、デフォルトクラスタのエントリは *Bozo-NoDsync* という名前になります。このクラスタのデフォルト値は、iPlanet Application Server のインストール時に設定されます。既存のデフォルトクラスタを設定する以外に、次の節で説明するように、新しいクラスタを設定することもできます。

新しいクラスタを作成するには、次の手順を実行します。

1. iPlanet レジストリを開始します。

注 UNIX システムでは、複数の iPlanet Application Server インスタンスを生成して、個別に iPlanet レジストリを割り当てることができます。必ず、シンクロナイザを割り当てるときは、サーバインスタンスに属する iPlanet レジストリを編集してください。

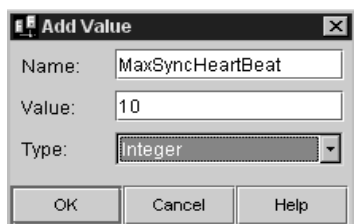
2. 次のキーを開きます。

SOFTWARE¥iPlanet¥Application Server¥Clusters¥

3. 「編集」メニューから「キーを追加」を選択します。次の図のように、「キーを追加」ダイアログボックスが表示されます。



4. 「キー」フィールドに新しいクラスタの名前を入力して、「OK」をクリックします。新しいクラスタが作成されます。次に、次の手順の説明に従ってクラスタを設定する必要があります。
5. 作成した新しいクラスタを選択し、「編集」メニューから「値を追加」を選択します。次の図のように、「値を追加」ダイアログボックスが表示されます。



クラスタに必要な値を追加するときは、各値のタイプを「Integer」に設定してください。「値を追加」ダイアログボックスの「入力」ドロップダウンリストで「Integer」を選択します。

値の追加が終了したら、「OK」をクリックして確認します。別の値を追加するには、作成した新しいクラスタを選択し、「編集」メニューから「値を追加」を選択します。

6. 新しいクラスタには、次の値を追加する必要があります。

- MaxBackups

この値は、バックアップサーバに割り当てることができる最大サーバ数を表します。このキーのデフォルト値は0です。バックアップ数を指定しない場合、セッションデータはバックアップされません。バックアップはパフォーマンスを低下させますが、信頼性との兼ね合いがあります。少なくとも1つのバックアップサーバを指定することをお勧めします。

指定できる最大バックアップサーバ数は、クラスタ内の総サーバ数-1です。

注 iASAT を使って新しいクラスタを作成すると、MaxHops という名前のキーが、MaxBackups キー の下に自動的に設定されます。このキーには、有効な機能は関連付けられていません。iPlanet レジストリエディタを使ってクラスタを作成した場合には、このキーのエントリを作成する必要はありません。

- MaxSyncHeartBeat

この値には、別のエンジンに送信する最大ハートビートメッセージ数を指定します。ハートビートメカニズムは、異常なクラスタ状態を検出するときに使われます。このキーのデフォルト値は10です。

- SyncHeartBeatInterval

この値には、サーバ間でハートビートメッセージが送信される間隔 (秒単位) を指定します。デフォルト値は30です。

- SyncTimerInterval

このキーには、同期サービスを呼び出して期限切れのデータがないかどうかを確認する間隔を、秒単位で指定します。タイマースレッドがノードリストを調べて期限切れのノードをすべて削除する頻度を指定します。デフォルト値は30です。

これらの値の最適は設定方法については、クラスタパラメータの設定を参照してください。

7. これらの値の追加が終了したら、新しいクラスタキーを選択して、「編集」メニューから「キーを追加」を選択します。
8. 「キーを追加」ダイアログボックスが表示されます。「SyncServers」と入力します。

作成した新しいクラスタキーの下に、SyncServers というフォルダが作成されます。クラスタにサーバを追加するときは、このフォルダをこのキーの下に追加する必要があります。

これで、新しいクラスタが作成および設定されました。次に、次の節の説明に従って、サーバをクラスタに追加します。

クラスタへのサーバの追加

サーバをクラスタに追加するには、次の手順を実行します。

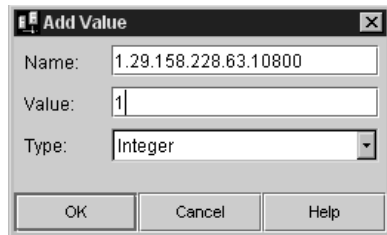
1. iPlanet レジストリを開始します。

注 UNIX システムでは、複数の iPlanet Application Server インスタンスを生成して、個別に iPlanet レジストリを割り当てることができます。シンクロナイザを割り当てるときは、サーバインスタンスに属している iPlanet レジストリを編集してください。

2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application
Server¥Clusters¥<clustername>¥SyncServers
```

3. 「編集」メニューから「値を追加」を選択します。次の図のように、「値を追加」ダイアログボックスが表示されます。



4. 「名前」テキストフィールドに、クラスタに追加するインスタンスの IP アドレスを入力します。次の例に示すように、IP アドレスの後にコロン (:) を入力し、KXS ポート番号を入力します。

ホスト IP アドレス :KXS ポート番号

エントリは、次のようになります。

```
129.158.228.63:10800
```

5. 「値」テキストフィールドに、クラスタ内でのサーバの優先度を示す値を入力します。たとえば、0 はもっとも高い優先度を示し、1 はその次のレベルの優先度を示します。もっとも低い優先度は 65,535 です。
6. 「入力」ドロップダウンリストから「Integer」を選択し、「OK」をクリックして、エントリを iPlanet レジストリに登録します。

他のサーバをクラスタに追加することもできます。サーバをクラスタに追加し終わったら、サーバが属するクラスタに、サーバを割り当てる必要があります。「クラスタへのシンクロナイザの割り当て」を参照してください。

クラスタからサーバを削除するには

追加したサーバは、SyncServers キーからサーバの IP アドレスのエントリを削除することによって、クラスタから削除できます。

iPlanet レジストリエディタを使ってサーバをクラスタから削除するには、次の手順を実行します。

1. iPlanet レジストリを開始します。

注	Solaris システムでは、複数の iPlanet Application Server を生成して、個別に iPlanet レジストリを割り当てることができます。必ず、シンクロナイザを割り当てるときは、サーバインスタンスに属する iPlanet レジストリを編集してください。
----------	--

2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application
Server¥Clusters¥<clustername>¥SyncServers
```

クラスタに追加したサーバの IP アドレスおよびポート番号のエントリが表示されます。

3. サーバを削除するには、目的の IP アドレスのエントリを選択し、「編集」メニューから「削除」を選択します。代わりに、目的のエントリを選択して、Delete キーを押すこともできます。
4. 操作を確認するように要求されます。「はい」をクリックして確認します。サーバがクラスタから削除されます。

注	<ul style="list-style-type: none"> • 削除したサーバは、同じクラスタに再度追加したり、別のクラスタに追加できます。詳細については、「クラスタにサーバを追加するには」を参照してください。 • iPlanet レジストリでは、クラスタの名前を変更できません。クラスタ名エントリを削除して、新しいクラスタ設定を作成する必要があります。詳細については、「新しいクラスタを作成するには」を参照してください。
----------	--

Sync Server の優先度の決定

クラスタ内の Sync Server に優先度値を設定できます。優先度を使用して、同じステータスの Sync Server (Sync Backup グループ内または Sync Alternate グループ内) の順序を選択します。優先度にかかわらず、iPlanet Application Server インスタンスが起動する順番で Sync Primary になるサーバ、Sync Backup、または Sync Alternate となる Sync Server が決まります。

Sync Server 優先度は、iASAT または iPlanet レジストリエディタを使って設定できます。

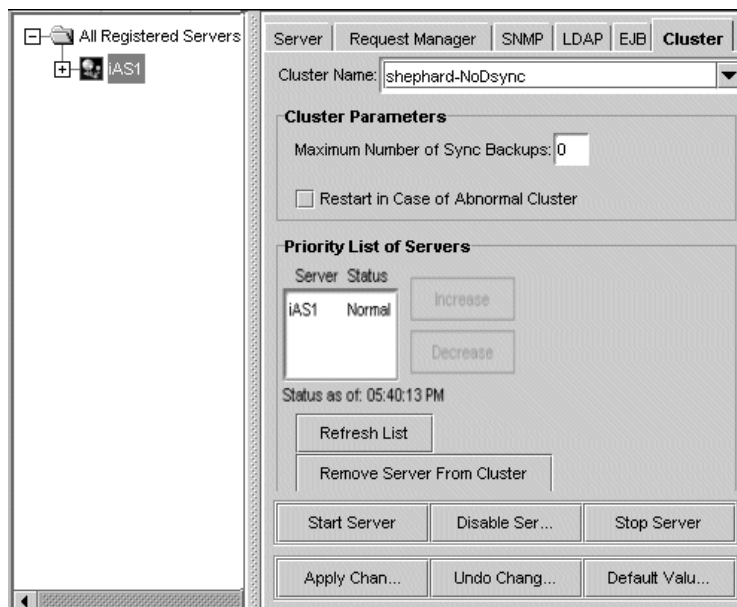
次の節では、Sync Server 優先度をサーバに割り当てる方法と、それらを変更する方法について説明します。

- iASAT を使って Sync Server の優先度を変更するには
- iPlanet レジストリエディタを使って Sync Server の優先度を変更するには

iASAT を使って Sync Server の優先度を変更するには

新しい Sync Server 優先度をクラスタ内のサーバに割り当てるには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックし、「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、優先度を変更する Sync Server が属しているクラスタ内のサーバをクリックします。
3. 「一般」ウィンドウの右側のペインで、「クラスタ」タブをクリックします。次のウィンドウが表示されます。



登録済みのサーバが「一般」ウィンドウの左側のペインに表示されます。値のサーバは、クラスタ内の優先度でソートされ、右側のペインに表示されます。

このリストは、クラスタ内の各サーバのステータスも表示されます。ステータスは通常、「Normal」です。クラスタの状態が異常な場合、ステータスは「Dual Primary」または「No Primary」になります。状態を修正する方法については、304 ページの「クラスタパラメータの設定」を参照してください。ステータスをチェックするには、「リストを更新」ボタンをクリックします。デフォルトでは、サーバのステータスは 15 秒ごとに更新されます。

4. 「サーバの優先度リスト」ボックスで、Sync Server 優先度を変更するサーバの名前をクリックします。
5. サーバの Sync Server 優先度を変更するには、「サーバの優先度リスト」ボックスの横にある次のボタンをクリックします。
 - 「増加」優先度を高くします。
 - 「減少」優先度を低くします。
6. たとえば、Sync Primary の Sync Server 優先度が 3 である場合は、「増加」を 1 回クリックすると優先度は 3 から 2 に変わります。数値が小さいほど、優先度が高いことを示します。たとえば、0 はもっとも高い優先度を示し、1 はその次のレベルの優先度を示します。もっとも低い優先度は 65,535 です。
7. サーバの優先度の再割り当てが完了したら、「変更の適用」をクリックします。

8. Sync Server の優先度の変更をクラスタ全体に適用するには、すべてのサーバを再起動し、すべてのサーバに相互に認識させる必要があります。

iASAT で行った変更は、iPlanet レジストリに反映されます。Sync Server の優先度は、iPlanet レジストリでも変更できます。次の節では、この方法について説明します。

iPlanet レジストリエディタを使って Sync Server の優先度を変更するには

iPlanet レジストリエディタを使って、iPlanet レジストリ内の Sync Server 優先度を直接変更するには、次の手順を実行します。

1. iPlanet レジストリを開始します。

注 UNIX システムでは、複数の iPlanet Application Server インスタンスを生成して、個別に iPlanet レジストリを割り当てることができます。必ず、シンクロナイザを割り当てるときは、サーバインスタンスに属する iPlanet レジストリを編集してください。

2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application
Server¥Clusters¥<clustername>¥SyncServers
```
3. クラスタに追加したサーバの IP アドレスおよびポート番号が表示されます。優先度を変更する Sync Server を選択します。
4. エントリをダブルクリックして開くか、「編集」メニューの「値を変更」を選択します。「値を変更」ダイアログボックスが表示されます。
5. 必要に応じて値を変更します。値が小さいほど、高い優先度を示します。たとえば、0 はもっとも高い優先度を示し、1 はその次のレベルの優先度を示します。もっとも低い優先度は 65,535 です。
6. 優先度を変更したサーバを含め、クラスタ内のすべてのサーバを再起動します。Sync Server の優先度の変更をクラスタ全体に適用するには、すべてのサーバを再起動し、すべてのサーバに相互に認識させる必要があります。

クラスタパラメータの設定

クラスタが効率的に動作するには、パフォーマンスに関連するパラメータを設定する必要があります。これらのパラメータは、iASAT または iPlanet レジストリエディタを使って設定できます。次の節では、各ツールを使ってクラスタパラメータを設定する方法について説明します。

- iASAT を使ってクラスタパラメータを設定するには
- iPlanet レジストリエディタを使ってクラスタパラメータを設定するには

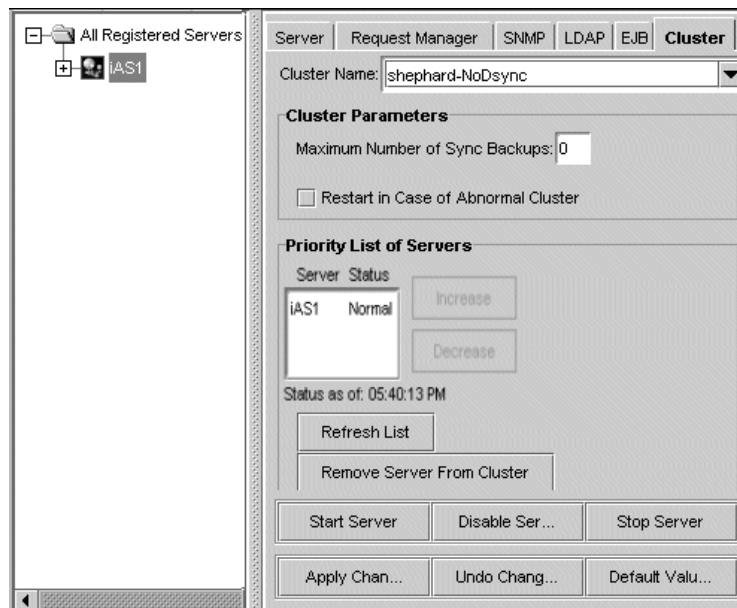
iASAT を使ってクラスタパラメータを設定するには

iASAT を使うと、次のパラメータを各クラスタに設定できます。

- 「Maximum Number of Sync Backups」
Sync Primary が使用する Sync Backup の最大数を指定できます。多数のサーバが含まれるクラスタでは、Sync Backup の最大数を指定することにより、バックアップとして使うサーバの数を制御できます。
- 「Restart」。クラスタが異常な場合。
クラスタの異常が検出された場合は、プロセスを再起動することもできます。クラスタの異常とは、Sync Primary ロールを持つ iPlanet Application Server インスタンスが複数存在する状態 (デュアルプライマリ) か、Sync Primary ロールを持つ iPlanet Application Server インスタンスが存在しない状態のことです。

iASAT を使ってクラスタパラメータを設定するには、次の手順を実行します。

1. iASAT ツールバーの「一般」をクリックし、「一般」ウィンドウを開きます。
2. 「一般」ウィンドウの左側のペインで、変更するクラスタのメンバーである iPlanet Application Server を選択します。
3. 「一般」ウィンドウの右側のペインで、「クラスタ」タブをクリックします。次のウィンドウが表示されます。



4. 「一般」ウィンドウの右側のペインにある「最大同期バックアップ数」テキストフィールドに、1つのクラスタセッションに許可する Sync Backup の最大数を入力します。
5. 異常なクラスタが検出されたときに、そのクラスタを修正する場合は、「異常クラスタの場合は再起動」チェックボックスをクリックします。

クラスタ内のすべてのサーバを再起動します。クラスタ全体に変更を適用するには、クラスタ内の各インスタンスを再起動し、すべてのインスタンス変更を認識させる必要があります。

注 サーバをクラスタに追加するには、その前に iASAT でサーバを登録する必要があります。

iPlanet レジストリエディタを使ってクラスタパラメータを設定するには

iPlanet レジストリエディタを使ってクラスタパラメータを設定するには、次の手順を実行します。

1. iPlanet レジストリを編集するサーバを停止します。

注 サーバの動作中にサーバレジストリを編集すると、重大な問題が発生する可能性があります。また、一部の変更は、エンジンを再起動しないと有効になりません。クラスタの設定を編集する前に、そのクラスタに属しているすべてのアプリケーションサーバを停止することを強くお勧めします。

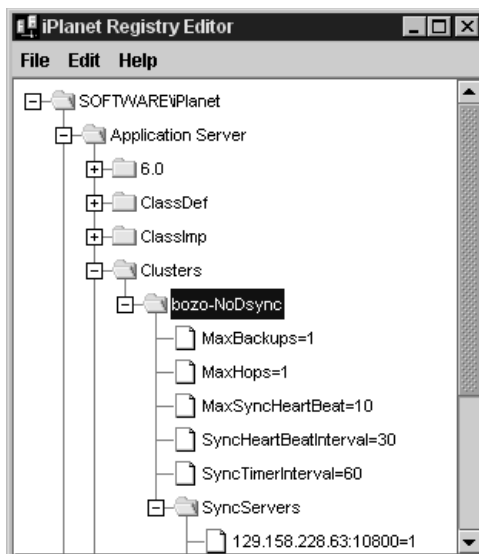
2. iPlanet レジストリを開始します。

注 UNIX システムでは、複数の iPlanet Application Server インスタンスを生成して、個別に iPlanet レジストリを割り当てることができます。必ず、シンクロナイザを割り当てるときは、サーバインスタンスに属する iPlanet レジストリを編集してください。

3. 次のフォルダを開きます。

SOFTWARE\iPlanet\Application Server\Clusters\\

この例では、次の図に示すように、デフォルトクラスタの名前は bozo-NoDsync で、優先度が 1 の Sync Server は 1 つです。



4. <clustername> キーの下にある次の値を、必要に応じて変更します。

ヒント iPlanet レジストリで値を変更するには、次の手順を実行します。

- a. キー名をダブルクリックするか、あるいはキーを選択し、「編集」メニューから「値を変更」を選択すると、「値を変更」ダイアログボックスが表示されます。
 - b. ダイアログボックスに新しい値を入力します。
 - c. 「OK」をクリックして iPlanet レジストリに変更を登録します。
-

- **MaxBackups**

バックアップデータ同期サーバ (Sync Server) の最大数により、Sync Primary のデータによって同時に更新される Sync Backup の数が決まります。バックアップデータ Sync Server の詳細は、289 ページの「クラスタについて」を参照してください。

すべての Sync Backup が同時に更新されるため、MaxBackup の値を増加するたびに、追加したバックアップサーバの負荷が増加します。バックアップサーバの数を設定するときは、パフォーマンスに与える影響を考慮してください。安全性を確保しながら、パフォーマンスを低下させない数を選択してください。通常は、デフォルト値の 1 かまいません。

注 次のエントリは、MaxHops キーです。このキーには、有効な機能は関連付けられていません。このキーの値を変更する必要はありません。

- **MaxSyncHeartBeat**

必要に応じて、MaxSyncHeartBeat の値をチェックおよび変更します。

この値には、別のエンジンに送信する最大ハートビートメッセージ数を指定します。ハートビートメカニズムは、異常なクラスタ状態を検出するときに使われません。デフォルト値は 10 です。

各ハートビートメッセージは次の要素で構成されています。

- メッセージを送信先エンジンのホスト ID およびポート
- クラスタ内の送信者のロール

ハートビートメッセージを受信すると、iPlanet Application Server エンジンは、そのクラスタ内のロールを識別して応答を返信します。

ハートビートは、Sync Backup サーバが Sync Primary に昇格したときに開始されます。新しい Sync Primary は、元の Sync Primary エンジンへのハートビートメッセージの送信を開始します。ネットワークエラーに一時的な障害が発生した場合は、2つのエンジンが Sync Primary になり、ダブルプライマリ (分割プライマリ) という異常な状態になります。この状態は自動的に修正されます。

- SyncHeartBeatInterval

必要に応じて、SyncHeartBeatInterval の値をチェックおよび変更します。

この値には、サーバ間でハートビートメッセージが送信される間隔 (秒単位) を指定します。デフォルト値は 30 秒です。

- SyncTimerInterval

必要に応じて、SyncTimerInterval の値をチェックおよび変更します。

このキーには、同期サービスを呼び出して期限切れのデータがないかどうかを確認する間隔を、秒単位で指定します。タイマースレッドがノードリストを調べて期限切れのノードをすべて削除する頻度を指定します。

この値が大きすぎると、期限切れのデータが残ることになります。小さすぎると、同期サービスを呼び出して確認する頻度が高くなり、システムパフォーマンスが低下します。ほとんどのクラスタでは、デフォルト値の 60 秒が適しています。

5. 必要な値の設定が終了したら、iPlanet レジストリエディタを閉じます。
6. この変更の影響を受けるすべてのサーバを再起動します。

注 クラスタパラメータは、iASAT を使って設定することもできます。詳細については、305 ページの「iASAT を使ってクラスタパラメータを設定するには」を参照してください。

上記の手順を正しく終了すると、デフォルトクラスタが再定義されます。次に、「クラスタへのシンクロナイザの割り当て」の手順に従って、クラスタ内のサーバ間の通信を有効にします。

クラスタへのシンクロナイザの割り当て

各 iPlanet Application Server には、データシンクロナイザが組み込まれています。シンクロナイザは、サーバ内のデータ、およびサーバ間の通信を同期化するツールです。

クラスタが通信するには、各サーバ内のシンクロナイザは、サーバが所属しているクラスタを認識する必要があります。iASAT を使ってクラスタを作成する場合、サーバシンクロナイザは自動的に割り当てられます。iPlanet レジストリエディタを使って手動でクラスタを作成する場合は、次の手順を実行して、シンクロナイザをクラスタに割り当てる必要があります。

1. iPlanet レジストリを編集するサーバを停止します。

注 サーバの動作中にサーバレジストリを編集すると、重大な問題が発生する可能性があります。また、一部の変更は、エンジンを再起動しないと有効になりません。クラスタの設定を編集する前に、そのクラスタに属しているすべてのアプリケーションサーバを停止することを強くお勧めします。

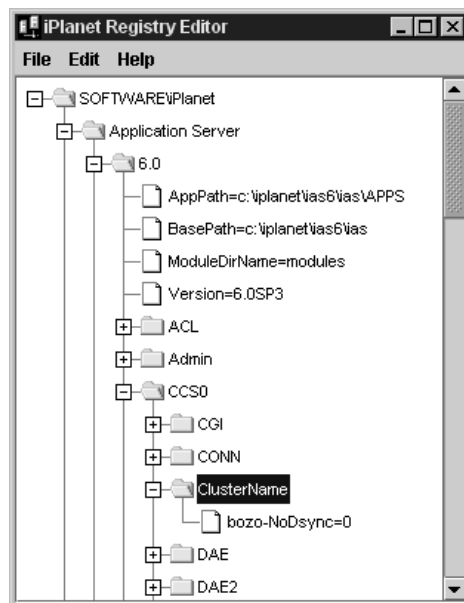
2. iPlanet レジストリを開始します。

注 UNIX システムでは、複数の iPlanet Application Server インスタンスを生成して、個別に iPlanet レジストリを割り当てることができます。必ず、シンクロナイザを割り当てるときは、サーバインスタンスに属する iPlanet レジストリを編集してください。

3. 次のキーを開きます。

`SOFTWARE\iPlanet\Application Server\6.5\CCS0\ClusterName¥`

次のウィンドウが表示されます。



4. デフォルトのクラスタ名が一覧表示されます。デフォルトのクラスタ名を削除して、新しいクラスタ名を指定することができます。
 5. iPlanet レジストリを閉じます。
 6. シンクロナイザを割り当てたサーバを開始します。
- シンクロナイザがクラスタに割り当てられました。

優先度を設定した Sync Server の調整

この節では、一般的な優先度について説明し、クラスタ協調の広範な例を示します。

優先度値は同じステータスの Sync Server (Sync Backup グループ内または Sync Alternate グループ内) 同士を選択することによりのみ使用されます。優先度にかかわらず、iPlanet Application Server インスタンスが起動する順番で Sync Primary になるサーバ、Sync Backup、または Sync Alternate となる Sync Server が決まります。

Sync Local には Sync Server になる資格がないため、優先度は割り当てられません。このため、Sync Local は、Sync Primary、Sync Backup、または Sync Alternate になることはできません。

クラスタ内で最初に起動した Sync Server が Sync Primary になります。次に起動した Sync Server は、Sync Backup になります。Sync Backup の最大数は、MaxBackups です。Sync Primary に障害が発生すると、もっとも高い優先度 (もっとも小さい値) を持つ Sync Backup が新しい Sync Primary になります。

Sync Backup が Sync Primary になると、Sync Backup の数が MaxBackups の値よりも少なくなります。Sync Backup の数を回復するために、もっとも優先度を持つ Sync Alternate が Sync Backup になります。

例：7 台のサーバで構成されるクラスタ内の調整

この例では、クラスタ内でのサーバロールの調整、サーバロールを決定するときの優先度の役割を示します。この例では、ロールの変化を説明するために、サーバの障害が頻繁に発生した場合を想定します。

必要に応じて、Sync Primary として起動する iPlanet Application Server インスタンスにもっとも高い優先度を割り当て、次に高い優先度 (降順) を Sync Backup に割り当てると、クラスタの保持が容易になります。この例のクラスタには、このような優先度は割り当てていません。また、このクラスタでは、優先度順にサーバを起動していません。

iPlanet Application Server インスタンスが実装された 7 台のサーバでクラスタを想定します。各サーバには、0～6 の番号がついています。0～4 のサーバは、サーバ番号と同じ優先度が割り当てられた Sync Server です。たとえば、サーバ 0 の優先度は 0 です。サーバ 5 および 6 は Sync Local です。クラスタの MaxBackups は 2 に設定します。

- サーバ 3 が最初に起動され、Sync Primary になります。
- サーバ 4 が次に起動され、Sync Backup になります。
- サーバ 6 が次に起動され、Sync Local になります。
- サーバ 1 が次に起動され、Sync Backup になります。
- サーバ 2 が次に起動され、Sync Alternate になります。
- サーバ 5 が次に起動され、Sync Local になります。
- サーバ 0 が次に起動され、Sync Alternate になります。

サーバ 3 は、障害が発生したため、停止します。サーバ 4 とサーバ 1 の 2 つの Sync Backup のうち、サーバ 1 の優先度が高い (値が小さい) ので、サーバ 1 が新しい Sync Primary になります。Sync Backup はサーバ 4 だけになります。

MaxBackups を 2 に設定したので、Sync Alternate のどちらかが Sync Backup に変換されます。サーバの優先度が他の Sync Alternate (サーバ 2) よりも高いので、サーバ 0 が新しい Sync Backup になります。現在の状態は、次のとおりです。

- サーバ 1 は Sync Primary です。
- サーバ 0 および 4 は Sync Backup です。
- サーバ 2 は Sync Alternate です。
- サーバ 5 および 6 は Sync Local です。
- サーバ 3 はオフラインです。

サーバ 3 がオンラインに戻り、Sync Alternate になります。サーバ 3 は元は Sync Primary でしたが、シンクロナイザからは現在、Sync Server と見なされています。このため、Sync Primary ロールには戻りません。現在の状態は次のとおりです。

- サーバ 1 は Sync Primary です。
- サーバ 0 および 4 は Sync Backup です。
- サーバ 2 および 3 は Sync Alternate です。
- サーバ 5 および 6 は Sync Local です。

サーバ 0 に障害が発生します。Sync Alternate のうち、サーバ 2 の優先度が高い (値が小さい) ので、サーバ 2 が Sync Backup になります。現在の状態は次のとおりです。

- サーバ 1 は Sync Primary です。
- サーバ 2 および 4 は Sync Backup です。
- サーバ 3 は Sync Alternate です。
- サーバ 5 および 6 は Sync Local です。
- サーバ 0 はオフラインです。

サーバ 0 がオンラインに戻り、Sync Alternate になります。Sync Primary であるサーバ 1 に障害が発生します。Sync Backup のうち、サーバ 4 よりもサーバ 2 の優先度が高いので、サーバ 2 が新しい Sync Primary になります。サーバ 0 は Sync Backup になります。現在の状態は次のとおりです。

- サーバ 2 は Sync Primary です。
- サーバ 0 および 4 は Sync Backup です。
- サーバ 3 は Sync Alternate です。
- サーバ 5 および 6 は Sync Local です。
- サーバ 1 はオフラインです。

サーバ 2 に障害が発生します。サーバ 0 は Sync Primary になり、サーバ 3 は Sync Backup になります。現在の状態は次のとおりです。

- サーバ 0 は Sync Primary です。
- サーバ 3 および 4 は Sync Backup です。
- サーバ 5 および 6 は Sync Local です。
- サーバ 1 および 2 はオフラインです。

サーバ 3 に障害が発生します。Sync Backup は 1 つだけですが、サーバ 5 および 6 は Sync Server ではないため、Sync Backup にはなれません。現在の状態は次のとおりです。

- サーバ 0 は Sync Primary です。
- サーバ 4 は Sync Backup です。
- サーバ 5 および 6 は Sync Local です。
- サーバ 1、2、および 3 はオフラインです。

優先度を設定した Sync Server の調整

開発用 iPlanet Application Server の設定

この章では、開発目的の iPlanet Application Server を設定するために必要な作業について説明します。

この章には次のトピックがあります。

- クラスパスの設定
- Java エンジンのヒープサイズの変更
- デフォルトのアプリケーションパスの変更
- ダイナミッククラス再読み込みの有効化
- J2EE コンポーネントのセッションタイムアウト値とキャッシュタイムアウト値の指定
- RMI/IIOP サポートの有効化
- データベースの環境変数の設定

クラスパスの設定

次の操作を実行するときは、iPlanet Application Server クラスパスを手動でセットアップする必要があります。

- EJB への RMI/IIOP アクセスを有効にするとき
- サーバに配置されているすべてのアプリケーションで使用できるヘルパーおよびフレームワーククラスのセットを作成するとき

iPlanet Application Server クラスパスを変更するには、次の手順を実行します。

Solaris の場合

1. <iasinstall>/ias/env パスに移動し、Common Environment スクリプト、iasenv.ksh を編集します。ファイルを編集するにはテキストエディタを使います。
2. ClassPath エントリを検索します。デフォルトのクラスパスを変更するか、または新しいクラスパスを追加することができます。
3. 変更を有効にするには、サーバを再起動します。

Windows の場合

1. iPlanet レジストリエディタを開きます。
2. 次のキーを開きます。
`SOFTWARE¥iPlanet¥Application Server¥6.5¥Java¥`
3. classPath を変更したら、変更を有効にするためにサーバを再起動します。

Java エンジンのヒープサイズの変更

KJS のデフォルトのヒープサイズは KJS スクリプトで指定されています。KJS により多くのオブジェクトを保存できるようにする場合は、手動でヒープサイズを変更します。Java ヒープのチューニングの詳細については、『iPlanet Application Server パフォーマンスおよびチューニングガイド』を参照してください。

Solaris の場合

1. <iasinstall>/ias/env に移動し、KJS (Java エンジン) 起動スクリプト、iasenv.ksh を編集します。

JAVA_ARGS エントリを検索し、必要に応じて最小ヒープサイズ (-Xms) と最大ヒープサイズ (-Xmx) の仕様を変更します。
2. ヒープサイズの変更を有効にするには、KJS プロセスをすべて閉じてから再起動します。

Windows の場合

1. iPlanet レジストリエディタを起動します。

2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥Java¥
```

3. `JavaArgs` 値を変更して、Java エンジンの最小ヒープサイズ (`-Xms <number>m`) と最大ヒープサイズ (`-Xmx<number>m`) を設定します。

ヒープサイズの仕様は次のようになります。

```
JavaArgs=-Xms16m -Xmx32m
```

4. 変更を有効にするには、すべての `kjs` プロセスを再起動します。

Windows では、Java エンジンのデフォルトのヒープサイズは設定されていません。

デフォルトのアプリケーションパスの変更

アプリケーションはデフォルトのアプリケーションパスに保存されています。これらのアプリケーションを配置するには、`DeployTool` を使ってその場所からアプリケーションを選択します。たとえば、サーバのホームディレクトリにアプリケーションを保存する場合は、このパスを変更する必要があります。`kregedit` を介してデフォルトのアプリケーションクラスパスを変更できます。

デフォルトのアプリケーションパスを変更する前に、次の手順を実行します。

- マシンまたはネットワークで新しいパスを作成します。
- この場所にアプリケーションのクラスファイルを手動でコピーします。

パスを作成したら、`iplanet` レジストリでデフォルトのクラスパスを変更します。`iplanet` レジストリでデフォルトのクラスパスを変更するには、次の手順を実行します。

1. `iPlanet` レジストリエディタを開きます。
2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥
```

3. `AppPath` 値を編集して、デフォルトのアプリケーションクラスパスを変更します。315 ページの「クラスパスの設定」を参照してください。
4. 変更を有効にするには、サーバを再起動します。

ダイナミッククラス再読み込みの有効化

デフォルトでは、ダイナミッククラスの再読み込みは iPlanet Application Server で無効になっています。ダイナミッククラスの再読み込みを有効にするには 2 つの方法があります。

ダイナミッククラスの再読み込みを有効にするには、次の手順のどちらかを実行します。

- iASAT を使う
- レジストリエディタを使う

iASAT を使う

1. iASAT を起動します。
2. アプリケーションサーバインスタンスを選択します。
3. 右側のペインで、「サーバ」の下の「ダイナミッククラスの再読み込みを有効にする」を選択します。

レジストリエディタを使う

1. iPlanet レジストリエディタを起動します。
2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application
Server¥6.5¥CCSO¥SYSTEM_JAVA¥Versioning
```
3. Disable=0 に設定します。デフォルト値は 1 です。この値は Servlet のダイナミック再読み込みが無効であることを示しています。
4. 変更を有効にするには、iPlanet Application Server を再起動します。

これにより Servlet クラスと登録済み JSP、および EJB 実装クラスのダイナミック再読み込みが有効になります。登録済み JSP は GUID が割り当てられていて、web.xml ファイルに Servlets として一覧表示されています。

J2EE コンポーネントのセッションタイムアウト値とキャッシュタイムアウト値の指定

アプリケーションの配置時に、JSP、Servlets、EJB などの J2EE コンポーネントのセッションタイムアウト値が設定されます。これらの値は Web サーバおよびアプリケーションサーバで実行しているすべての J2EE コンポーネントに適用されます。

必要に応じて、それぞれのアプリケーションに対して個々の J2EE コンポーネントのセッションタイムアウト値を指定できます。個々の J2EE コンポーネントに指定した値は配置時に設定されたデフォルトの値よりも優先されます。

Servlets および JSP にはキャッシュタイムアウト値を指定できます。アプリケーション配置時の Servlets および JSP のデフォルトのキャッシュタイムアウト値は 0 に設定されています。

Deployment Tool を使ってこれらの値を設定することもできます。詳細は Deployment Tool に付属のオンラインヘルプを参照してください。

この節では次のトピックについて説明します。

- JSP と Servlets のセッションタイムアウト値を設定するには
- EJB のセッションタイムアウト値を設定するには
- JSP と Servlets のキャッシュタイムアウト値を設定するには

JSP と Servlets のセッションタイムアウト値を設定するには

JSP と Servlets では、それぞれのアプリケーションに対してセッションタイムアウト値を設定できます。これらの値は配置時に設定されます。セッションタイムアウト値を変更するには、次の手順を実行します。

1. iPlanet レジストリエディタを起動します。
2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥J2EE-Module¥<module name>¥
```

セッションタイムアウト値がすでに指定されている場合は、`session-timeout=()` というキーが表示されます。セッションタイムアウト値が指定されていない場合は、この値をモジュールに追加できます。

3. セッションタイムアウト値を指定するモジュールを選択します。
4. 「編集」 > 「値を追加」をクリックします。

5. 「名前」フィールドで名前を指定します (例: Session-timeout)。
6. 「値」フィールドで値を指定します (例: 10)。タイムアウト値は分単位で指定されます。
7. 値の種類を Integer に設定します。
8. 「OK」をクリックして変更を保存します。

セッションタイムアウト値はアプリケーションに属している Servlets と JSP に設定されます。

EJB のセッションタイムアウト値を設定するには

EJB では、セッションタイムアウト値は iASAT を使ってグローバルに設定されます。詳細については、137 ページの「実行時に EJB コンテナパラメータを指定する」を参照してください。iASAT を介して設定する値は配置されるすべての EJB に適用されます。

ただし、iPlanet レジストリの個々の EJB に対してセッションタイムアウト値をカスタマイズできます。1 つの EJB のセッションタイムアウト値を指定するには、次の手順を実行します。

1. iPlanet レジストリエディタを起動します。
2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥J2EE-Module¥<module name>¥ejbs
```

次のような文字列が表示されます。

```
<Beanhomename>=(18c06cf0-21dd-11d2094b6-0060083a5082)
```

角かっこで囲まれた文字列は、選択したアプリケーションモジュール内で使われる EJB を識別する GUID (Global Unique Identifier) です。この GUID を書き留めてください。

3. SOFTWARE¥iPlanet¥Application Server¥ClassDef¥ に移動します。
このフォルダでは、アプリケーション配置時のすべての GUID を一覧表示します。希望の GUID を検索します。
4. GUID のキーをツリー状の階層構造に展開します。
5. SessionDescriptor キーを開きます。このキーには、EJB に設定されているセッションタイムアウト値や不活性化タイムアウト値などの値が含まれています。
6. セッションタイムアウト値を変更したら、変更を有効にするためにサーバを再起動します。

JSP と Servlets のキャッシュタイムアウト値を設定するには

デフォルトのキャッシュタイムアウト値とは、Web サーバによって削除されるまでに Servlets と JSP がキャッシュ内に保持されている期間です。デフォルトのキャッシュタイム値はアプリケーションの配置時に設定されます。

JSP と Servlets のデフォルトのキャッシュタイムアウト設定値は GUID ごとに変更できます。キャッシュタイムアウト値を変更するには、次の手順を実行します。

1. iPlanet レジストリエディタを起動します。
2. 次のキーを開きます。

```
SOFTWARE¥iPlanet¥Application Server¥6.5¥J2EE-Module¥<module name>¥servlets¥
```

次のような GUID が表示されます。

```
<module name>servlet=(1a488137-7510-1941-bae5-080020b90148)
```

この GUID を書き留めてください。

3. SOFTWARE¥iPlanet¥Application Server¥ClassDef¥ に移動します。
このフォルダでは、アプリケーション配置時のすべての GUID を一覧表示します。希望の GUID を検索します。
4. GUID を選択して、「編集」>「キーを追加」をクリックします。
5. 「名前」のフィールドに、「Caching」と入力します。
6. このキーの下に次の値を入力します。
 - キャッシュサイズ (K バイト単位の値。Integer タイプ)
 - キャッシュタイムアウト (秒単位の値。Integer タイプ)
 - キャッシュオプション (値は Timeout-Create または Timeout-Lastaccess として設定できます。String タイプ)
 - キャッシュ基準 (適切なキャッシュ基準を指定する値。String タイプ) キャッシュの基準の詳細については、『Java デベロッパガイド』を参照してください。
7. 変更を有効にするには、サーバを再起動します。

アプリケーションで JSP を使っている場合は、手順 2 で説明したパスで該当する GUID が検出されます。

RMI/IIOP サポートの有効化

iPlanet Application Server では、インストール時に EJB への RMI/IIOP アクセスのサポートを有効にしません。ただし、iASAT を介して RMI/IIOP ブリッジプロセスを iPlanet Application Server 環境に追加できます。

iASAT を介して iPlanet Application Server 環境を RMI/IIOP ブリッジプロセスに追加するには、次の手順を実行します。

1. iASAT を起動します。
 - Solaris の場合は、<iASinstall>/ias/bin/ に移動し、「ksvradmin」と入力します。
 - Windows の場合は、「スタート」メニューで「プログラム」>「iPlanet Application Server」を選択し、「iAS Administration Tool」を選択します。
2. 新しい RMI/IIOP プロセスを追加するサーバを選択します。サーバを展開してサーバの下にプロセスを表示します。

kjs プロセスと kxs プロセスが少なくとも1つずつ表示されます。cxs プロセスを追加して、RMI/IIOP プロセスを有効にします。
3. 「ファイル」>「新規」をクリックします。「プロセス」を選択します。
4. 「プロセス」ドロップダウンリストで「cxs」を選択します。
5. 「Port」テキストフィールドで、他のプロセスで使われていないポート番号を入力します。
6. 「IIOP Port」テキストフィールドで、IIOP ブリッジプロセスのポート番号を指定します。デフォルトの IIOP ポート番号は、9010 です。システム環境にすでに存在するポート番号と競合しない場合は、このポート番号を使うことができます。
7. 「OK」をクリックして cxs プロセスを登録します。iPlanet Application Server マシンに cxs プロセスが追加されます。
8. サーバを再起動して、cxs プロセスを起動します。

Solaris では、コマンドラインからも IIOP ブリッジプロセスの有無を確認できます。次のようにします。

```
ps - ef | grep iiop  
root 1153 1 0 17:00:15 ?0:00 /bin/sh/usr/iPlanet/iAS6/bin/kjs -cset  
CCSO - eng 3 - iiop - DORBinsPort=9010
```

データベースの環境変数の設定

データベースのシステム環境の変数はインストール時に設定されますが、それらの変数を変更されていないことを確認します。

Solaris では、プロンプトに「env」と入力すると、環境変数を確認できます。環境変数のリストを確認してください。システム環境の変数がここで示したガイドラインどおりに設定されていない場合は、適切な設定値に変更します。

Sybase

- Bourne シェルの場合

```
DSQUERY=<sybase servername>; export DSQUERY
```

- C シェルの場合

```
setenv DSQUERY <sybase servername>
```

sybase servername をユーザの Sybase サーバ名に置き換えます。

Oracle

- Bourne シェルの場合

```
ORACLE_HOME=<oracle install directory>; export ORACLE_HOME
```

- C シェルの場合

```
setenv ORACLE_SID <oracle SID>
```

```
setenv ORACLE_HOME <oracle install directory>
```

DB2

- Bourne シェルの場合

```
DB2INSTANCE=<db2instance>; export DB2INSTANCE
```

- C シェルの場合

```
setenv DB2INSTANCE <db2instance>
```

INFORMIX

- Bourne シェルの場合

```
INFORMIXSERVER=<informixserver>; export INFORMIXSERVER
```

- C シェルの場合

```
setenv INFORMIXSERVER <informixserver>
```

環境サイズの調整

インストール後に Administrative Server プロセスがシステムリソースを 100% 消費している場合は、NT マシンで環境領域のサイズを増やします。システムルート (通常は、C:\Winnt または C:\Winnt4) の System32 ディレクトリにある Config.nt ファイルを編集して次の行を追加し、マシンを再起動します。

```
SHELL=%systemroot%\system32\command.com /e:2048
```

マシンを再起動しても問題が解消されない場合は、上記の行で指定した /e:2048 の代わりに /e:4096 を指定してさらに環境サイズを増やします。

トラブルシューティング

この付録には、iPlanet Application Server のトラブルシューティングに関する次の情報が記載されています。

- クラスパスの設定
- サーバサイドコネクション数の設定

クラスパスの設定

アプリケーションの実行時に、iPlanet Application Server クラスローダがレジストリの `SYSTEM_JAVA` パラメータを使って AppLogic クラスファイルを見つけることができなかった場合は、iPlanet Application Server は Java クラスローダにリクエストを引き渡し、入れ替わりに、Java クラスローダはクラスファイルを見つけるために `CLASSPATH` 環境変数を読み込みます。`SYSTEM_JAVA` パラメータは、`CLASSPATH` および `GX_CLASSPATH` 設定の両方を含むレジストリパラメータです。ユーザのクラスパスが指定されていなくても、AppLogics および Servlets はこの操作を実行します。

サーバサイドコネクション数の設定

あるスレッドに対してグローバルトランザクションを起動すると、そのスレッドにコネクションが確立されます。したがって、サーバサイドコネクション数を設定するには、企業内の Java サーバ (KJS) スレッドの総数を使います。

たとえば、Oracle データベースの場合は、Oracle サーバインストールの `pfile` ディレクトリにある `initinstancename.ora` ファイルの `max_number_processes` の値を変更します。

用語集

Administration Server すべての iPlanet Web Server を設定するために使うフォームを含む、Web ベースのサーバ。

admpw Enterprise Administrator Server スーパーユーザのユーザ名およびパスワードファイル。

CGI Common Gateway Interface の略。外部プログラムが HTTP サーバと通信するときを使うインターフェイス。CGI を使うように作成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や解析を行わないフォームを処理したり解析したりする。

chroot サーバを特定のディレクトリに制限するために作成する追加ルートディレクトリ。この機能は、保護されていないサーバを保護するために使う。

ciphertext 暗号化によって隠された情報。意図した受信者だけが復号化できる。

client auth クライアント認証。

DHCP Dynamic Host Configuration Protocol の略。Internet Proposed Standard Protocol の一つで、これによりシステムは IP アドレスをネットワーク上の個々のコンピュータに対してダイナミックに割り当てることができる。

DNS Domain Name System の略。ネットワーク上のマシンが標準 IP アドレス (198.93.93.10 など) をホスト名 (www.netscape.com など) に割り当てるために使うシステム。通常、マシンはこの変換された情報を DNS サーバから取得するか、システム上に維持されるテーブル上で探す。

DNS 別名 DNS サーバが認識するホスト名で別のホストを指している、特に DNS CNAME レコードのこと。マシンには必ず、実名があるが、1 つ以上のエイリアスを持つことがある。たとえば、www.yourdomain.domain というエイリアスは、realthing.yourdomain.domain という名前の、現在サーバ存在する実際のマシンを指す。

FORTEZZA 重要ではあるが機密指定を受けていない情報を保護するために、米国政府機関で使われる暗号化システム。

FTP File Transfer Protocol の略。ファイルのあるコンピュータから別のコンピュータにネットワークを介して転送できるインターネットプロトコル。

GIF Graphics Interchange Format の略。CompuServe によって作成された、クロスプラットフォーム対応のイメージ形式。GIF ファイルのサイズは通常、ほかのグラフィックファイルタイプ (BMP や TIFF) よりもはるかに小さくなる。GIF は、もっとも一般的な交換形式である。GIF イメージは、UNIX、Microsoft Windows、および Apple Macintosh システムで容易に表示できる。

HTML Hypertext Markup Language の略。WWW (World Wide Web) 上のドキュメントで使われるフォーマット化言語。HTML ファイルは、Netscape Navigator のようなブラウザに対して、テキストの表示方法、グラフィックやフォームアイテムの配置方法、およびほかのページへのリンクの表示方法を指示するフォーマット化コードを含む、プレーンテキストファイルである。

HTTP HyperText Transfer Protocol の略。HTTP サービスとクライアントとの間で情報を交換する方法。

HTTP-NG 次世代のハイパーテキスト転送プロトコル。

HTTPD HTTP デーモンまたはサービスの略。HTTP プロトコルを使って情報を提供するプログラム。Netscape Enterprise Server は HTTPD と呼ばれる。

HTTPS 安全なバージョンの HTTP。SSL (Secure Sockets Layer) を使って実装される。

imagemap イメージの領域をアクティブにするプロセス。これにより、ユーザはイメージ上のさまざまな領域をマウスでクリックして情報にアクセスしたり、情報を取得したりできる。imagemap は「imagemap」という CGI プログラムを意味することもある。これは、ほかの HTTPD 実装で imagemap 機能进行处理するために使われる。

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 UNIX ファイルのリスト。これにより、確実にプログラムが連続して実行される。存在する場所の名前から、/etc/inittab と呼ばれることもある。このファイルは、すべての UNIX システムで使用できるとは限らない。

IP アドレス IP (Internet Protocol) アドレス。インターネット上のマシンの実際の場所を指定する、ドットで区切られた数字の組み合わせ (198.93.93.10 など)。

iPlanet Console すべての iPlanet サーバをエンタープライズネットワーク上で集中管理するためのグラフィックインターフェイスをサーバ管理者に提供する Java アプリケーション。iPlanet Console のインストールされたどのインスタンスからでも、アクセス権を与えられた企業内ネットワーク上のすべての Netscape サーバを表示し、アクセスできる。

ISDN Integrated Services Digital Network の略。

ISINDEX クライアントでの検索を有効にする HTML タグ。ドキュメントはネットワークナビゲータの機能を使い、検索文字列を受け取ってこれをサーバに送信し、フォームを使わずに検索可能な索引にアクセスできる。<ISINDEX> を使うには、クエリハンドラを作成する必要がある。

ISMAP ISMAP は HTML ドキュメントの中で使われる IMG SRC タグの拡張機能で、指定したイメージが **imagemap** になるようサーバに指示する。

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プログラミング言語で、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を作成するために使われる。

Java Servlet インスタンス化、初期化、破棄、ほかのコンポーネントからのアクセス、設定管理など、すべての **Java Servlet** メタファンクションを可能にする拡張機能。Java Servlet 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。

JavaScript クライアントインターネットアプリケーションおよびサーバインターネットアプリケーションを開発するための、コンパクトでオブジェクト指向のスクリプト言語。

JavaServer Pages インスタンス化、初期化、破棄、ほかのコンポーネントからのアクセス、設定管理など、すべての **JavaServer** ページメタファンクションを可能にする拡張機能。JavaServer 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。

magnus.conf 主要な Enterprise Server コンフィグレーションファイル。このファイルには、全体的なサーバ設定情報 (ポート、セキュリティなど) が含まれている。このファイルには、初期化中にサーバを設定する変数の値を設定する。Enterprise Server はこのファイルを読み取り、起動時に変数の設定を実行する。再起動されるまでサーバがこのファイルを再び読み取ることはないため、このファイルに変更を追加するたびにサーバを再起動する必要がある。

MD5 RSA Data Security によるメッセージダイジェストアルゴリズム。MD5 を使うと、高い確率で固有な、データの短いダイジェストを作成できる。同じダイジェスト電子メールを作成するデータを作成することは、数学的にきわめて困難。

MD5 シグネチャ MD5 アルゴリズムによって作成されたメッセージダイジェスト。

MIB (マネージメントインフォメーションベース) Management Information Base の略。

MIME Multi-Purpose Internet Mail Extensions の略。マルチメディア電子メールおよびメッセージング向けに、最近登場した標準。

mime.types MIME (Multi-purpose Internet Mail Extension) タイプのコンフィグレーションファイル。このファイルは、ファイル拡張子を MIME タイプにマッピングし、要求されている内容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リクエストは、クライアントが HTML ファイルを要求していることを意味し、.gif 拡張子を持つリソースのリクエストは、クライアントが GIF フォーマットのイメージファイルを要求していることを意味する。

MTA メッセージ転送エージェント (MTA) サーバでエージェントサービスを使うには、サーバの MTA ホストを定義する必要がある。

NIS (Unix) Network Information Service の略。UNIX マシンが、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特定の情報を、コンピュータネットワーク全体で収集、照合、および共有するために使うプログラムおよびデータファイルのシステム。

NNTP ニュースグループのための NNTP (Network News Transfer Protocol)。サーバでエージェントサービスを使うには、ニュースサーバホストを定義する必要がある。

NSAPI 「サーバプラグイン API」を参照してください。

obj.conf サーバのオブジェクトコンフィグレーションファイル。このファイルには、補足的な初期化情報、サーバのカスタマイズのための設定、およびサーバがクライアント (ブラウザなど) からのリクエストを処理するために使う命令が含まれている。Enterprise Server は、クライアントの要求を処理するたびに、このファイルを読み取る。

protocol ネットワーク上のデバイスが情報を交換する方法を示す規則。

RAM ランダムアクセスメモリ。コンピュータ内部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されるプログラムについて記述した、UNIX マシンにあるファイル。このファイルは、その格納場所から /etc/rc.2.d と呼ばれる。

resource サーバがアクセスし、要求した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、またはプログラム。

RFC Request For Comments の略。通常、インターネットコミュニティに提出されるプロシージャまたは標準ドキュメント。技術が標準として受け入れられる前に、誰もがその技術に関するコメントを送信できる。

root (UNIX) UNIX マシンでもっとも高レベルの権限を持つユーザ。root ユーザは、そのマシンにあるすべてのファイルに関する完全なアクセス権限を持つ。

SNMP Simple Network Management Protocol の略。

SOCKS ファイアウォールソフトウェアまたはハードウェア (ルータ設定など) によって、直接接続が妨げられている場合に、ファイアウォール内部から接続を確立するファイアウォールソフトウェア。

SSL Secure Sokets Layer の略。HTTP の保護バージョンである HTTPS を実装するために使われる、二者 (クライアントとサーバ) 間に保護された接続を確立するソフトウェアライブラリ。

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際にサーバによって使われる。strftime はサーバがファイルの最終変更日を示すためにトレーラで使う特殊なフォーマット化言語を持つ。

Sym-link (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われるリダイレクションのタイプ。Sym-link により、ユーザのファイルシステムのある部分から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に対する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エンタープライズ (企業内) ネットワークのため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 台のマシンを相互に接続し、リモートログインのための端末エミュレーションをサポートするプロトコル。

TLS Secure Sokets Layer の略。HTTP の保護バージョンである HTTPS を実装するために使われる、二者 (クライアントとサーバ) 間に保護された接続を確立するソフトウェアライブラリ。

top (UNIX) 一部の UNIX プログラムにある、システムリソースの使用に関する現在の状態を示すプログラム。

uid (UNIX) UNIX システムで各ユーザに関連付けられている固有の番号。

URI Uniform Resource Identifier の略。省略された URL を使ってセキュリティの層を追加するファイル識別子。URL の最初の部分は、ファイルの完全な物理パス名をユーザから隠す URL マッピングに置き換えられる。「URL マッピング」も参照。

URL Uniform Resource Locator の略。サーバおよびクライアントが、要求されたドキュメントに対して使うアドレス指定方式。URL は「場所」とも呼ばれる。URL の形式は protocol://machine:port/document である。

たとえば、http://www.netscape.com/index.html と記述する。

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によるエイリアスにマッピングするプロセス。これにより、ディレクトリ内のファイルはファイルの完全な物理パス名を参照する代わりに、ディレクトリのエイリアスだけを参照すればよくなる。したがって、ファイルは、usr/Netscape/SuiteSpot/docs/index.html ではなく、/myDocs/index.html のように識別され、ユーザがサーバファイルの物理的な場所を知る必要がなくなるため、さらにセキュリティが強化される。

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、またはファイルシステムが満杯になったことにより損傷を受けた URL データベースを修復し、更新するプロセス。

Web アプリケーション Servlet、JavaServer Pages、HTML ドキュメント、およびイメージファイル、圧縮アーカイブ、などのデータを含むその他の Web リソースのコレクション。Web アプリケーションは、アーカイブ (WAR ファイル) にパッケージされている場合もあるが、オープンディレクトリ構造に存在する場合もある。

Web アプリケーションアーカイブ (WAR) 完全な Web アプリケーションが圧縮形式で含まれているアーカイブファイル。iPlanet Web Server は、WAR ファイルにあるアプリケーションにはアクセスできない。iPlanet Web Server がサービスを提供できるようにするには、その前に Web アプリケーションを圧縮解除する (wdeploy ユーティリティを使って配置する) 必要がある。

Web パブリッシング サーバクライアントがサーバファイルにアクセスして操作し、ドキュメントをリモートで編集および発行できる機能。Web パブリッシングにより、ドキュメントのバージョン管理、リンク管理、検索、アクセス制御、およびエージェントサービスが、サーバユーザに提供される。

Windows CGI (Windows NT) Visual Basic など、Windows ベースのプログラミング言語で作成された CGI プログラム。

アクセス制御エントリ (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う規則の階層。

アクセス制御リスト (ACL) ACE のコレクション。ACL は、サーバにアクセスできるユーザを定義するためのメカニズム。特定のファイルまたはディレクトリに固有の ACL 規則を定義して、1 人以上のユーザまたは 1 つ以上のグループにアクセスを許可または拒否できる。

暗号化 意図した受信者以外によって復号化されたり、読み取られたりしないように情報を変形するプロセス。

暗号化方式 暗号化方式とは、暗号化または復号化に使われる、暗号化のためのアルゴリズム (数学関数) のこと。

インテリジェントエージェント ユーザのためにさまざまな要求 (HTTP、NNTP、SMTP、および FTP 要求) を実行するサーバ内にあるオブジェクト。ある意味で、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として機能し、サーバが実現する要求を行う。

エージェント ルータ、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 顧客、供給業者、および外勤の社員がデータをアクセスできるようにするための、企業のイントラネットからインターネットへの拡張部分。

危殆化キーリスト (CKL) 妥協キーを持つユーザに関するキー情報のリスト。証明機関もこのリストを提供している。

キャッシュ ローカルに格納されている元のデータのコピー。キャッシュされたデータは、要求があった場合でもリモートサーバから再び取得する必要はない。

共通ログファイル形式 アクセスログに情報を入力するために、サーバで使われる形式。この形式は、iPlanet FastTrack や Enterprise サーバなど、すべての主なサーバで共通。

クライアント Netscape Navigator など、WWW (World Wide Web) の素材を要求し、表示するのに使うソフトウェア。ブラウザプログラムとも呼ばれる。

検索から除外する単語 「ストップワード」を参照。

公開キー 公開キー暗号化方式で使われる暗号化キー。

公開情報ディレクトリ (UNIX) UNIX ユーザのホームディレクトリにあるドキュメントルートにないディレクトリ、またはユーザが制御しているディレクトリ。

コネクショングループ

コレクション 単語リストやファイルプロパティなど、ドキュメントに関する情報が含まれたデータベース。コレクションは、指定の検索条件に一致するドキュメントを検索するために検索関数で使われる。

サーバデーモン いったん実行を開始すると、クライアントからの要求を待機し、受け取るプロセス。

サーバルート サーバプログラムファイル、コンフィグレーションファイル、メンテナンスファイル、および情報ファイルの保持専用の、サーバマシン上にあるディレクトリ。

サーバプラグイン API iPlanet サーバの中心的な機能の拡張またはカスタマイズ、あるいはその両方ができるようになり、HTTP サーバとバックエンドアプリケーションの間でインターフェイスを構築するために有効なスケーラブルで効果的なメカニズムを提供する拡張機能。NSAPI と呼ばれる。

最終変更ヘッダー サーバからの HTTP 応答で返された、ドキュメントファイルの最終変更時刻。

証明機関 (CA) 暗号化されたトランザクションに使うデジタルファイルを発行するサードパーティの組織。

証明書 通信者双方によって信頼されたサードパーティが発行する、譲渡や偽造ができないデジタルファイル。

証明書破棄リスト (CRL) 証明機関 (CA) が提供する、取り消されたすべての証明書のリスト。

シンプルインデックス ファンシーインデックスの逆。このタイプのディレクトリの表示では、ファイルの名前だけがグラフィック要素なしで表示される。

スーパーユーザ (UNIX) UNIX マシンでもっとも高レベルの権限を持つユーザ (「root」とも呼ばれる)。スーパーユーザは、そのマシンにあるすべてのファイルに関する完全なアクセス権限を持つ。

ストップワード 検索対象でない単語として検索関数に識別される単語。これに含まれる代表的な単語には、*the*、*a*、*an*、*and* がある。これは、検索から除外する単語とも呼ばれる。

ソフトリスタート サーバを内部的に再起動させるように、コンフィグレーションファイルを再読み込みしてサーバを再起動させる方法。ソフトリスタートにより、プロセスに HUP シグナル (シグナル 1) が送信される。プロセス自体は、ハードリスタートの場合のように終了することはない。

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終了させようとするのが中断するまでの指定経過時間。

デーモン (UNIX) 特定のシステムタスクを実行するバックグラウンドプロセス。

ドキュメントルート サーバマシン上にあり、サーバにアクセスするユーザに提示するファイル、イメージ、およびデータを含むディレクトリ。

トップレベルドメイン機関 ホスト名分類のもっとも高いカテゴリ。通常、ドメインが所属する組織のタイプ (com は企業、.edu は教育機関など) またはその所属する国家 (.us はアメリカ、.jp は日本、.au はオーストラリア、.fi はフィンランドなど) を表す。

認証 クライアントが SSL 対応のサーバに接続されていることを確認できるようにして、別のコンピュータがこのサーバになりすましたり、SSL 対応であるように見せかけたりすることがないようにすること。

認定 サーバ全体またはサーバ上の特定のファイルやディレクトリにアクセスを許可すること。認定は、ホスト名および IP アドレスなどの条件によって制御できる。

ネットワーク マネージメントステーション (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、iPlanet サーバなどの SNMP を実行するデバイスが管理される。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ある。

ハードリスタート プロセスまたはサービスの終了およびそのあとの再起動。「ソフトリスタート」も参照。

パスワードファイル (UNIX) UNIX マシンにあり、UNIX ユーザのログイン名、パスワード、およびユーザ ID 番号が格納されているファイル。このファイルは、格納場所から /etc/passwd と呼ばれる。

秘密キー 公開キー暗号化方式で使われる復号化キー。

品質フィードバックエージェント ユーザがエラー情報 (スタックおよびレジスタダンプ) を自動的に iPlanet へ送信できるようにするエラー処理メカニズム。

ファイアウォール ネットワークに接続された組織内のコンピュータを外部のアクセスから保護するネットワーク設定で、通常、ハードウェアとソフトウェアの両方が含まれる。ファイアウォールは一般的に、物理的な建造物や組織の敷地内に置かれてい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われる。

ファイル拡張子 通常はファイルのタイプを定義する、ファイル名の最後の部分。たとえば、index.html というファイル名では、ファイル拡張子は html となる。

ファイルの種類 特定のファイルの形式。たとえば、グラフィックファイルのファイルタイプは、テキストファイルとは異なる。ファイルタイプは常に、ファイル拡張子 (.gif、.html など) によって識別される。

ファンシーインデックス シンプルインデックスより多くの情報を提供する、インデックスのメソッド。ファンシーインデックスでは、内容のリストが名前順に、ファイルサイズ、最終変更日、およびファイルタイプを示すアイコンとともに表示される。このため、ファンシーインデックスは、クライアントがロードするための時間がシンプルインデックスよりも長くかかることがある。

プライマリドキュメントディレクトリ 「ドキュメントルート」を参照してください。

ブラウザ 「クライアント」を参照してください。

フレキシブルログ形式 アクセスログに情報を入力するためにサーバで使われる形式。

ホームページ サーバ上にあり、サーバのコンテンツのカatalogまたはエントリポイントとして機能するドキュメント。このドキュメントの場所は、サーバのコンフィギュレーションファイル内で定義される。

ホスト名 *machine.domain.dom* の形式を取るマシンの名前で、IP アドレスに変換される。たとえば、*www.iplanet.com* は、サブドメイン *iplanet* およびドメイン *com* 内にあるマシン *www* である。

有効期限ヘッダー リモートサーバにより指定された、返信ドキュメントの有効期限。

リダイレクション 特定の URL へアクセスするクライアントを同じサーバまたは異なるサーバの別の場所に転送するシステム。このシステムは、リソースが移動した場合に、透過的にクライアントが新しい場所を使えるようにするときに便利である。さらにこのシステムは、末尾のスラッシュなしでディレクトリがアクセスされた場合に、相対リンクの整合性を維持するためにも使う。

索引

A

- ACL、「アクセス制御リスト」を参照, 109
- Administration Server
 - フェールオーバーの, 288
- AgentToken, 174
- AppLogics
 - HTTP 入力変数, 162

C

- C++ サーバ
 - 追加と調整, 130
 - フェールオーバーの, 288
 - プロセス属性, 53
- CGI
 - 有効にする, 163
- CGI フラグ
 - 設定, 174
 - 説明, 162
- CONFIG ファイル
 - 説明, 68
 - 編集, 69
 - 例, 69
- CONTENT_LENGTH, 89
- CONTENT_TYPE, 89
- cookie

- 設定, 173
- 無効にする, 173
- 有効にする, 173

- CPU 負荷, 52, 277, 282
- CXS
 - プロセス属性, 53
- CXS プロセス
 - 追加する, 227

D

- Deployment Tool, 254, 256
- Directory Server
 - クラスタ, 143
 - 説明, 98
 - フェールオーバーの設定, 143
 - マニュアル, 98, 155
- Directory Server エントリ
 - iPlanet Console を使って作成する, 104
- DN (識別名), 101
- DNS (Domain Name System)
 - エイリアス、定義, 327
 - 定義, 327
- DSync ノードのデータ
 - ダンプ, 92
 - ダンプファイルの書式設定, 93
- DSync メッセージ

記録する, 95

E

EJB

コンテナ, 137

EJB コンテナ宣言パラメータエディタ
アクセスする, 137

evtcategory フィールド, 83

evtstring フィールド, 83

evttime フィールド, 83

evttype フィールド, 83

G

GIF、定義, 328

H

HTML

定義, 328

HTTP

定義, 328

HTTP_ACCEPT, 90

HTTP_CONNECTION, 90

HTTPD, 328

HTTP_HOST, 90

HTTP_REFERER, 90

HTTPS

定義, 328

HTTP_USER_AGENT, 90

HTTP 変数, 175

Web サーバリクエストの, 171

アプリケーションコンポーネントに入力, 162,
176

作成, 177

追加する, 172

データフィールドに割り当てる, 171

ログで使う, 89

割り当てる, 171

HTTP 変数を割り当てる, 171

HTTP ログ、スクリプトを使ったセットアップ, 83

I

I18N

サポートする, 140

iAS Administration、「iPlanet Application Server
Administration Tool」を参照

inittab

定義, 328

INIT ファイル, 70

iPlanet Application Server

登録する, 24

複数のサーバ環境, 247, 253

リソースの増強, 129

iPlanet Application Server Administration Tool

説明, 22

起動, 23

iPlanet Console

説明, 99

マニュアル, 98

ユーザおよびグループの作成に使う, 101, 102

iPlanet レジストリエディタ, 32

IP アドレス

定義, 328

ISAPI, 90

J

Java クラスローダ, 325

Java サーバ (KJS)

追加する, 130

追加と調整, 130

フェールオーバーの, 288

プロセス属性, 53

K

Key, 38

kregedit

CGI リクエストの設定, 174

HTTP 変数を設定する, 172, 176

Web コネクタポートを設定する, 175
について, 32

L

LDAP

説明, 98

ldapmodify

エントリーを変更する, 105

説明, 105

LDIF

エントリー、書式化, 105

エントリー、説明, 105

説明, 104

ListenPort, 175

log(), 80

Log_db2.sql, 83

Log_ifmx.sql, 83

Log_mssql.sql, 83

Log_ora.sql, 83

Log_syb.sql, 83

logtime フィールド, 90

M

Management Information Base (MIB)

エントリーを書式設定する, 66

説明, 66

有効化, 67

MaxBackups

説明, 290

MaxHops, 299, 308

MCastHost, 285

MCastPort, 285

MD5、定義, 329

MIB (マネージメントインフォメーションベース), 66

MIME、定義, 329

MTA

定義, 330

N

NIS、定義, 330

NNTP

定義, 330

NoCookie, 173

NSAPI, 90

O

Oracle リソースマネージャ, 225

P

PATH_INFO, 90

Primary Sync Server、「Sync Primary」を参照

R

RAM

定義, 330

rc.2.d, 330

REMOTE_ADDR, 90

REQUEST_METHOD, 90

root

定義, 330

S

SERVER_PROTOCOL, 90

SNMP

監視, 63

設定の確認, 67, 71

説明, 63

統計収集を有効にする, 65

SOCKS、定義, 331

SSL

定義, 331

Sybase リソースマネージャ, 225

Sync Alternate

Sync Backup に昇格, 291

説明, 291

Sync Backup

Sync Primary に昇格, 290

説明, 290

Sync Local

説明, 291

Sync Primary

説明, 290

Sync Server

説明, 290

SyncServers レジストリキー

Sync Server を定義するには, 290

SyncTimerInterval, 309

マニュアルの形式, 16, 17

W

wdeploy ユーティリティ, 332

Web アプリケーション

定義, 332

Web アプリケーションアーカイブ (WAR)

定義, 332

Web コネクタ

設定, 161, 249

説明, 89

複数の iAS 環境での, 247

ポート番号、設定, 175

Web コネクタのポート番号

設定, 175

説明, 162

Web コネクタプラグイン, 247, 248

Web サーバ

手動で設定する, 162

Web サーバのリクエスト

記録する, 162

Web パブリッシング

定義, 332

T

telnet, 331

U

uid

定義, 331

UNIX, 83

URI、定義, 331

URL

定義, 331

マッピング、定義, 332

あ

アクセス制御リスト (ACL)

作成, 109

説明, 109

変更する, 112

アプリケーション

セキュリティ, 97

パーティションする, 254, 256, 287

複数のサーバ上で管理する, 253

ローカルに管理する, 254

ロードバランスの分散, 254

ロードバランスを実行するように管理して配置する, 260

アプリケーションエラー, 80

アプリケーションコンポーネント

- HTTP 変数からの入力, 162
- 管理, 22
- 説明, 22
- パーティションする, 256
- パフォーマンス条件, 280
- パフォーマンスを計算する, 279
- 分散する, 266
- 分散レベルを変更する, 262
- 無効にする, 257
- 有効にする, 257
- アプリケーションコンポーネントクライテリア, 277, 280, 281, 282
- アプリケーションコンポーネントパフォーマンス値, 279
- アプリケーションコンポーネントを無効にする, 257
- アプリケーションログ, 81
- アプリケーションをパーティションする, 254, 256

い

- イベント
 - ポーリング, 61
- イベント通知, 59
 - スクリプト, 62
- イベントのポーリング, 61
- イベントログ, 83
 - コンソールに, 81
 - スクリプトを使ってセットアップする, 83
 - 説明, 73
 - データベースに, 81
 - ファイルに, 81
 - メッセージの書式設定, 75
 - メッセージの送信先を選択する, 81
- インストールキーを更新する, 37
- インダクトトランザクション
 - 解決, 224
- インテリジェントエージェント、「エージェント」を参照

う

- ウェイト要因
 - 調整する, 279, 280
 - ロードバランスについての調整, 277

え

- エージェント
 - 定義, 333
- エクストラネット、定義, 333
- エラー、アプリケーション, 80
- エラーメッセージ
 - アプリケーションから, 80
 - サービスから, 74

お

- 応答時間
 - 短縮する, 266
- オプションの再起動
 - 調整する, 139

か

- 隠しフィールド
 - 設定, 173
 - 無効にする, 173
 - 有効にする, 173
- 間隔の更新, 282
- 監視
 - SNMP を使う, 63
- 監視する, 51
 - クエリ, 54
 - サービス, 51
 - 受動, 59
 - プロセス属性, 52

き

期限切れノードの削除, 299, 309

起動

iPlanet Application Server Administration
Tool, 23

キャッシュ

サイズ、設定する, 138

サイズ、説明する, 138

キャッシュ、定義, 333

共通ログファイル形式, 333

記録する

iAS を指定する, 250

Web サーバのリクエスト, 162

サーバメッセージ, 73

説明, 73

ファイルに, 57

プロセスコンソールに, 81

プロセスデータをファイルに, 57

く

クエリ、監視する, 54

クラスタ

Directory Server の, 143

からサーバを削除する, 296

作成, 294

説明, 289

内の通信, 292

にサーバを追加する, 295

の優先度, 311

例, 311

クラスタデータ

ダンプファイルの書式設定, 92

クラスタのデータをダンプする, 92

クラスパス

設定, 325

グローバルトランザクション, 213

け

警告メッセージ, 74

検索から除外する単語, 333

こ

広域ネットワーク (WAN), 252

固定ロードバランス, 262, 267

有効にする, 268

コレクション

定義, 333

コンポーネントの応答時間

セットアップ, 271

さ

サーバ

優先度の変更, 302

サーバデーモン、定義, 333

サーバの応答時間

セットアップ, 272

サーバの削除, 25

サーバの登録の削除, 25

サーバのレスポンス時間, 278

サーバ負荷クライテリア, 277, 282

設定, 278

サーバルート、定義, 333

最大エンジンシャットダウン時間

設定, 141

最大サーバシャットダウン時間

設定, 141

サブエージェント, 64

し

識別名 (DN), 101

システムレベルのサービス, 52

- 実行サーバ, 288
 - プロセス属性, 52
- 実行時
 - の EJB コンテナ宣言パラメータの設定, 137
- 実行時間、平均, 54
- 受信バイト
 - 監視する, 54
- 受動監視, 59
- 昇格
 - Sync Backup に昇格する Sync Alternate, 291
 - Sync Primary に昇格する Sync Backup, 290
- 章テンプレート, 325
- 情報メッセージ, 74

す

- スーパーユーザ、定義, 334
- ストップワード, 334
- スレッド
 - 監視する, 53, 54
 - 現在のリクエスト, 54
 - 最小数と最大数を指定する, 131
 - シングルスレッド環境, 142
 - の数の調整, 132, 133
 - パフォーマンスインパクト, 129
 - ユーザリクエストの数を調整, 131
 - リクエスト待機, 54
 - 利用度の設定, 131
- スレッドセーフ, 142
- スレッドプール, 131

せ

- セキュリティ
 - cookie, 173
 - アクセスコントロールリスト (C++ アプリケーション), 97
 - 隠しフィールド, 173
 - 実装, 100
 - 説明, 100
 - 認証, 100

- ロールベース, 97
- セキュリティ、アプリケーション, 97
- セッションタイムアウト
 - 設定, 138
 - 説明, 138
- 設定
 - Web コネクタ, 161, 248, 249
 - 手動による Web サーバ, 162
- 宣言パラメータ、実行時の設定, 137

そ

- 送受信バイト、監視する, 53
- 送信バイト、監視する, 54

た

- タイマー間隔
 - 設定, 138
 - 説明, 138
- タイムアウト
 - DSync ロギングオプション, 92

て

- ディスク
 - 入出力, 52
- データベース
 - Web サーバメッセージログ, 89
 - 記録する, 81
 - メッセージログ, 83
- データベースコネクション
 - 監視する, 54
 - キャッシュする, 54
- データベースドライバ
 - 説明, 179

と

- 登録する
 - iPlanet Application Server, 24
- トークン
 - DSync ログインオプション, 92
- トップレベルドメイン機関, 334
- トランザクション
 - 監視する, 54
 - サーバごとに設定, 216
 - 「トランザクション」ウィンドウの管理, 215
 - プロセスごとの設定, 221
- トランザクションマネージャ, 213
- トランザクションログの失敗
 - 復旧, 225
- トランスポートマッピング
 - 説明, 71
 - 例, 71

に

- 認証
 - C++ アプリケーションのアクセスコントロールリスト, 101
 - アクセスコントロールリストの定義, 109
 - ロールベースの J2EE アプリケーション, 101
 - ロールベースの設定, 106

は

- ハートビート
 - 間隔の設定, 299, 309
 - 最大メッセージ数の設定, 299, 308
- パスワードファイル, 335
- バックアップ
 - クラスタの最大数, 305, 308
- パフォーマンス, 129
 - 監視する, 51
 - 記録する, 51
 - チャートにする, 51

ひ

- 表記規則、マニュアル, 17

ふ

- ファイル拡張子、定義, 335
- ファイルタイプ
 - 定義, 335
- フェールオーバー, 288
 - Administration Server, 288
 - Directory Server, 143
 - DSync ログインオプション, 92
- フェールオーバー保存間隔
 - 設定, 138
- フェールオーバー保存間隔、説明, 138
- 不活性化タイムアウト
 - 設定, 138
 - 説明, 138
- 符号化方式, 332
- ブリッジサーバ
 - プロセス属性, 53
- ブリッジプロセス
 - 説明, 130
 - 追加する, 227
- ブロードキャスト間隔, 282
 - 説明, 282
 - 調整する, 283
- ブロードキャスト情報, 252
- プロセス
 - 追加する, 130
- プロセスコンソール
 - 記録する, 81
- プロセス属性
 - 監視する, 55
- プロセスデータプロット
 - 削除, 58
 - 変更する, 57
- プロット
 - 追加する, 56
- 分散データ同期
 - サーバ間での設定, 289
 - 設定, 251

説明, 287
分散データ同期 (DSync)
オプションを記録する, 92

へ

ベース間隔, 282

ほ

ホスト, 250
ホスト名
定義, 336

ま

マスターエージェント, 64
起動, 70
設定, 68
非標準ポートで起動する, 71
マニュアル
表記規則, 17
マルチキャスト, 271
マルチキャストサーバのホストアドレス, 284
マルチキャスト通信, 252
設定, 284
マルチスレッド化, 142

め

メッセージ
イベントログ, 73
エラー, 74
記録するタイプを選択する, 74
警告, 74
コンソール, 81
種類, 74

情報, 74
書式設定, 75
メッセージロギングサービス, 73
設定, 81
メッセージャ
DSync ロギングオプション, 92
メモリスラッシュ, 52, 278, 283

も

モードの切り替え, 263
モジュール
DSync ロギングオプション, 92

ゆ

有効期限ヘッダー、定義, 336
ユーザ
変更する, 112
ユーザおよびグループ
iPlanet Console で作成する, 101, 102
管理, 99, 100
保存, 100
ユーザおよびグループの保存, 100
ユーザグループ, 109
ユーザ定義基準
セットアップ, 277
優先度
Sync Local に割り当てられていない, 292
クラスタ, 311
サーバの変更, 302
データ同期、説明, 290

ら

ラウンドロビン
セットアップ, 273

り

リクエスト

- CGI、設定する, 174
- 間隔ごと, 54
- 現在, 54
- 合計, 52, 54
- 受信した数, 52
- 準備完了, 54
- 待機, 54

リクエスト実行プロファイル, 271

リクエストスレッド, 131

リソース

- 定義, 330

リダイレクション, 336

リッチクライアント, 53, 227

ユーザの要求への影響, 249

ラウンドロビン, 270

ロードバランスサービス, 277, 280, 282

ロール

- 管理, 107
- について, 97

ロギングサービス

- 設定, 81

ログの失敗

- からの復旧, 225

れ

レジストリ, 171

ろ

ロードバランス, 248, 251, 252, 256, 260, 270

ias、ユーザ定義基準, 271

Web コネクタプラグイン, 270

ウェイト要因を調整する, 277, 278, 280

間隔の更新, 282

固定, 267

固定、有効にする, 268

コンポーネントの応答時間, 270

サーバの応答時間, 270

説明, 265

前提条件, 266

とマルチキャスト通信, 252

のためにアプリケーションを管理する, 260

のためにアプリケーションを分散する, 254

負荷の計算, 277

プラグイン, 247, 248

ブロードキャスト間隔, 282

方式を選択する, 270

無効にする, 262