

IBM Host On-Demand バージョン 13.0



Host On-Demand 計画、インストール、および構成

IBM Host On-Demand バージョン 13.0



Host On-Demand 計画、インストール、および構成

— お願い —

本書および本書で紹介する製品をご使用になる前に、 177 ページの『付録 E. 特記事項』に記載されている情報をお読みください。

本書は、IBM® Host On-Demand (プログラム番号 5724-I20) のバージョン 12、および新版において特に断りのない限り、それ以降のすべてのリリースおよび修正に適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC14-7266-02

IBM Host On-Demand Version 13.0

Planning, Installing, and Configuring Host On-Demand

Ninth Edition (February 2016)

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 1997, 2016.

目次

本書について	vii
Host On-Demand に関するその他の文書	vii
本書で使用する規則	ix
用語	ix
Java に関連する用語	xi

第 1 部 Host On-Demand の計画 . . . 1

第 1 章 IBM Host On-Demand の紹介 . . . 3

Host On-Demand について	3
Host On-Demand の機能の仕組み	3
Host On-Demand の使用目的	5
コスト効率の高い接続方法	5
構成データの集中管理	5
Telnet サーバーへの直接接続	5
ブラウザ・ベースのユーザー・インターフェース	6
多様なプラットフォームおよびネットワーク環境のサポート	6
Java のサポート	6
インターネット・プロトコルバージョン 6 のサポート	6
多くの各国語のサポート	7
セキュア接続	7
カスタム HTML ファイル	7
新規 e-business アプリケーションを作成するためのツールキット	7
Programmable Host On-Demand	8
Host On-Demand Session Manager API	8
WebSphere Portal のサポート	9
IBM System i サーバー上の DB2 データベースへの接続	9
新機能	9
Host On-Demand に関する最新情報の入手	9
Host On-Demand バージョン 13 の新機能	9

第 2 章 配置のための計画 11

HTML ベース・モデルの理解	11
構成サーバー・ベース・モデルの理解	12
結合モデルの理解	13
クライアント配置の考慮事項	14

第 3 章 クライアントでの Java の計画 15

Java 用のキャッシュ・クライアントの改善	15
サポートの制限	15
Java を使用したクライアントのダウンロード	16
プリロード・リストにないコンポーネントをダウンロードできない	16
HTML ファイルにいくつかのコンポーネントが含まれていない	16
Java のある Mac OS X	17

Mac OS X の制限	17
Java クライアントを使用すると起動時間が少し遅くなる	17
Java プラグイン固有の制限	17
カスタマー提供のアプレットおよび Java に関する制限	18
制限付きユーザーおよび Java に関する制限	18
ブラウザと Java プラグイン	18
Java 対応ブラウザ	18
Host On-Demand クライアントでサポートされるブラウザおよびプラグイン	18
Java プラグインを使用した Microsoft Internet Explorer	19
Java プラグインを使用した Firefox	19

第 4 章 セキュリティーの計画 21

Transport Layer Security (TLS)	21
TLS セキュリティーの仕組み	21
Host On-Demand 用の TLS	23
Web サーバー・セキュリティ	29
セキュリティの構成	29
リダイレクター	29
リダイレクターの使用目的	29
リダイレクターの動作	30
リダイレクターのロード容量	31
リダイレクターがサポートするオペレーティング・システム	31
ファイアウォールでの Host On-Demand の使用	33
ファイアウォール・ポートの構成	34
プロキシ・サーバーを介したホスト・システムへの接続	37
ユーザー ID セキュリティー	38
Web 高速ログオン (Web Express Logon)	38
ネイティブ認証	39
Windows ドメインのログオン	39
FIPS 環境	39

第 5 章 各国語サポートの計画 41

サポートされる言語	41
サポートされるホスト・コード・ページ	42
3270 および 5250 コード・ページ	42
VT コード・ページ	45
CICS ゲートウェイ・コード・ページ	46
日本語 JIS2004 Unicode サポート	46
ユーザー定義文字マッピング	47
i/OS および OS/400 でのユニコード・サポート	47

第 2 部 Host On-Demand のインストール、アップグレード、およびアンインストール 49

第 6 章 Host On-Demand サーバーおよび関連ソフトウェアのインストール . . . 51

Installation Manager を使用した Host On-Demand のインストール	51
重要なリンク	51
HOD をインストールする前に	51
インストールの準備	51
旧バージョンの Host On-Demand からのアップグレード	52
Host On-Demand のインストール	53
インストールに Installation Manager GUI を使用	53
デプロイメント・ウィザード	56
旧バージョンのデプロイメント・ウィザードからのアップグレード	56
デプロイメント・ウィザードのインストール	56
デプロイメント・ウィザード・インストール・イメージを Host On-Demand サーバーからダウンロード	57
Host Access Toolkit	58
Host Access Toolkit のインストール	58
コンソール・モードでのインストール	59
コンソール・モードでのインストールについて	60
HOD を iSeries にインストールする前に	61
インストール手順	61
コンソール・モードでのデプロイメント・ウィザードのインストール	65
コンソール・モードでの Host Access Toolkit のインストール	65
サイレント・モードでのインストール	65
インストール手順	65
構成サブプレットのインストール	67
サブプレットを WebSphere Application Server にデプロイ	67

第 7 章 Host On-Demand サーバーのアンインストール 69

Installation Manager コンソール・モードを使用した Host On-Demand のアンインストール	69
--	----

第 3 部 Host On-Demand の構成 71

第 8 章 Host On-Demand エミュレーター・クライアントの構成 73

Host On-Demand HTML ファイルの作成	73
Host On-Demand セッションの構成	75
デプロイメント・ウィザードの使用	75
Host On-Demand サーバーへのデプロイメント・ウィザード出力の配布	76

第 9 章 Host On-Demand の管理クライアントおよび新規ユーザー・クライアントの使用 77

管理クライアントおよび新規ユーザー・クライアントのロード	77
--	----

管理クライアント	77
ディレクトリー・ユーティリティ	79
新規のユーザー・クライアント	79

第 10 章 Host On-Demand エミュレーター・クライアントの使用 81

エミュレーター・クライアントのロード	81
適切なクライアントの選択	82
キャッシュ・クライアント	83
キャッシュ・クライアントのインストール	84
キャッシュ・クライアントの除去	88
複数の Host On-Demand サーバーにアクセスする際のキャッシュ・クライアントのサポートの問題	90
Windows のキャッシュ・クライアント・サポート	90
Mac OS X (Java クライアントのみ) の場合のキャッシュ・クライアントのサポート	91
キャッシュ・クライアントのトラブルシューティング	92
Web Start クライアント	93
Web Start クライアントのインストール	94
Web Start のための Web サーバーの構成	96
Web Start クライアントのアップグレード	96
初期インストール後の Web Start コンポーネントの追加	97
Web Start と Windows 制限ユーザー	97
Web Start でのブックマーク・セッション	97
Web Start の HTTPS での使用	97
Web Start クライアントの除去	97
ダウンロード・クライアント	98
ダウンロード・クライアントの立ち上げ	98
キャッシュ・クライアントまたは Web Start クライアントのインストール後のダウンロード・クライアントの起動	98
事前定義エミュレーター・クライアント	98
クライアントのダウンロード・サイズの削減	99
カスタマー提供の Java アーカイブおよび Java クラスのデプロイ	100
AdditionalArchives HTML パラメーターの使用	101
パブリッシュ・ディレクトリーからのデプロイ	102
アーカイブ・ファイルについてのヒント	102

第 11 章 Database On-Demand クライアントの使用 103

ディスプレイ・エミュレーション・クライアントおよびマクロにおけるデータベース機能	104
Database On-Demand クライアントの開始	104
Database On-Demand の定義済みクライアント	105
ユーザー用の Database On-Demand の構成	106
JDBC ドライバーの入手とインストール	106
データベース・アクセス用のファイル・フォーマット	107
Database On-Demand での複数コード・ページの使用	107

サポートされる Database On-Demand コード・ページ	107
第 12 章 サーバー・マクロ・ライブラリーの作成およびデプロイ	109
サーバー・マクロ・ライブラリーの Web サーバーへのデプロイ	110
サーバー・マクロ・ライブラリーの共有ドライブへのデプロイ	110
第 13 章 セッション・プロパティーの動的変更	113
初期 HTML ファイルのセットアップ	113
コード・ベースの設定	114
ConfigBase パラメーターの追加	114
HTML パラメーターの指定変更	114
指定変更可能な特定のセッション・プロパティー	115
例 1: クライアントの IP アドレスに基づいて LU 名を指定変更する	120
例 2: HTML のフォームを使用してユーザーが接続先ホストを指定できるようにする	123
第 14 章 zSeries での Host On-Demand の構成	125
別個の読み取り/書き込みプライベートおよびパブリッシュ・ディレクトリーのセットアップ	125
Host On-Demand プライベート・ディレクトリー用に別個のファイル・システムをセットアップ	125
別個のユーザー・パブリッシュ・ディレクトリーのセットアップ	125
z/OS でのマイグレーション考慮事項	126
プライベート・ディレクトリーのバックアップ	126
デプロイメント・ウィザードを z/OS サーバーからインストール	127
第 15 章 IBM System i での Host On-Demand の構成	129
IBM System i 上での Host On-Demand サービス・マネージャーの構成、開始および停止	129
構成	129
開始	130
停止	130
HOD サーバー状況での作業	131
証明書管理	132
情報バンドラーの開始	132
HOD プリンター定義テーブルの作成	132
IBM System i でのデプロイメント・ウィザードの使用	132
セキュア接続のための IBM System i サーバーの構成	133
i/OS および OS/400 での TLS を使用する Host On-Demand のインストールと構成	134
セキュア接続のための Telnet サーバーの構成	134
Host On-Demand の CustomizedCAs 鍵リングの構成	134

クライアント認証	135
セキュア接続用の Host On-Demand OS/400 プロキシの構成	136
セキュア Web サービス	136
i/OS および OS/400 でのユニコード・サポート	137
一般情報	137
ホスト・プログラミング情報	138

第 16 章 WebSphere Portal での Host On-Demand のデプロイ

Host On-Demand による Portal Server の処理方法	139
Portal Server での Host On-Demand クライアントの使用	140
ポートレットによる Host On-Demand へのアクセスに関する制限	140
Host On-Demand ポートレット使用時の特殊な考慮事項	141
Host On-Demand ポートレットの拡張	144

第 17 章 Eclipse-Plugin サポート

Host On-Demand プラグインの作成	145
セッション・プロパティーの動的設定	147
別個のユーザー・パブリッシュ・ディレクトリーを使用	148
Host On-Demand プラグインで使用されるビュー ID	149
Eclipse-Plugin 環境での Host On-Demand の使用に関する制限	149

第 18 章 LDAP を使用する Host On-Demand サーバーの構成

LDAP サポートの設定	151
スキーマ拡張機能のインストール	152
LDAP をデータ・ストアとして使用する場合の Host On-Demand サーバーの構成	153

付録 A. ローカル・インストール・クライアントの使用

ローカル・インストール・クライアントをサポートするオペレーティング・システム	155
ローカル・クライアントのインストール	155
ローカル・クライアントの開始	155
ローカル・クライアントの削除	156

付録 B. IKEYCMD コマンド行インターフェースの使用

IKEYCMD コマンド行インターフェースの環境のセットアップ	157
IKEYCMD コマンド行構文	158
Host On-Demand の IKEYCMD タスクのリスト	159
新規鍵データベースの作成	159
データベース・パスワードの設定	160
データベース・パスワードの変更	160
CA のリスト表示	161

新規鍵ペアと証明書要求の作成	161
サーバー証明書の保管	162
CA 署名証明書の受信	162
CA 証明書の保管	163
自己署名証明書の作成	164
サーバー証明書をクライアントに使用可能にする	164
CustomizedCAs.p12 への不明な CA のルート の追加.	165
鍵のエクスポート	166
鍵のインポート.	166
鍵データベースのデフォルト鍵の表示	167
暗号化されたデータベース・パスワードの stash フ ァイルへの保管.	167
IKEYCMD コマンド行パラメーターの概要	167
IKEYCMD コマンド行オプションの概要	169
コマンド行呼び出し	170

ユーザー・プロパティ・ファイル	171
---------------------------	-----

付録 C. P12 鍵リング・ユーティリティ	173
一	173
使用法.	173
オプション	173
例	174

付録 D. ネイティブ・プラットフォーム・ランチャーのコマンド行オプション	175
--	------------

付録 E. 特記事項	177
-------------------	------------

付録 F. 商標	179
-----------------	------------

本書について

「*Host On-Demand* 計画、インストール、および構成」は、*Host On-Demand* プログラムの計画、インストール、および構成を支援するための資料です。本書は管理者を対象としています。本書は、以下の 3 つの部分に分かれています。

1 ページの『第 1 部 *Host On-Demand* の計画』では、*Host On-Demand* をインストールおよび配置する前に考慮すべき情報を提供します。例えば、どのようなサーバー・プラットフォームを使用するのか、どの配置モデルを使用するのか、セキュリティをどのように扱うのか、などについて説明します。

49 ページの『第 2 部 *Host On-Demand* のインストール、アップグレード、およびアンインストール』では、各オペレーティング・システムに対応した段階的な手順を示します。

71 ページの『第 3 部 *Host On-Demand* の構成』では、セッション構成情報を定義および管理する方法、セッション構成情報を動的に変更する方法、新規クライアントをカスタマイズする方法、および *Host On-Demand* をユーザーに配置する方法を指定するための、さまざまな構成モデルについて説明します。

Host On-Demand のインストールと構成を行った後は、オンライン・ヘルプを使用して、セッションの定義方法やその他の管理用タスクの実行方法を学習してください。

「*Host On-Demand* 計画、インストール、および構成」は *Host On-Demand Knowledge Center* でも入手できます。

Host On-Demand に関するその他の文書

Host On-Demand については、「*Host On-Demand* 計画、インストール、および構成」のほかにも、製品の使用に役立つ情報源が提供されています。ここに記載する資料を参照するには、*Host On-Demand Knowledge Center* にアクセスしてください。多くの文書は *Host On-Demand* 製品や *ToolKit* にも組み込まれています。



MySupport 機能を利用すると、サポート・ビューを個別設定できます。また、IBM 製品に対する新しい修正パッケージ、ダウンロード、および最新の技術サポートに関する情報を掲載した電子メール通知を毎週受け取るための登録を行うことができます。サポート通知に登録するには、「About my notifications」を参照してください。

- オンライン・ヘルプ。オンライン・ヘルプは、*Host On-Demand* のインストールが完了した後で、管理者およびユーザーにとっての基本的な情報源となります。これには、*Host On-Demand* の作業を行うための詳細なステップが示されています。目次と索引を利用することにより、タスク指向のヘルプ・パネルおよび概念的なヘルプ・パネルを簡単に見つけることができます。 *Host*

On-Demand のグラフィカル・ユーザー・インターフェース (GUI) の使用中にヘルプ・ボタンを利用すると、その GUI に関するパネル・レベルのヘルプ・パネルが表示されます。

- **README ファイル。** `readme.html` ファイルには、製品文書に載せることのできなかった、製品に関する最新の情報が含まれています。
- **「Web Express Logon 解説書」。** この資料は、Web Express Logon を理解し、実装し、トラブルシューティングするためのステップバイステップのアプローチを提供しています。Web Express Logon の概要を示し、ステップバイステップの例をいくつか示して、Web Express Logon の計画とユーザー環境へのデプロイメントを援助します。さらに、カスタマイズされたマクロおよびプラグインを作成するためのいくつかの API が記載されています。
- **「マクロ・プログラミング・ガイド」。** この資料では、ホスト・アプリケーションとのユーザー対話を自動化したり、ホスト・アプリケーションとネイティブ・アプリケーションとの間でデータを受け渡したりするための、Host On-Demand マクロの作成方法が説明されています。この資料はマクロ開発のすべての面についての詳細情報を提供し、「Host Access Beans for Java Reference」で以前に発行されたマクロ言語に関する改訂された情報が含まれています。
- **「Host Printing Reference」。** ホスト・セッションを構成した後で、「Host Printing Reference」を参照して、ユーザーがホスト・セッション情報をローカルまたは LAN 接続されたプリンターまたはファイルに出力可能です。
- **「Session Manager API 解説書」。** この資料には、ホスト・セッションおよびホスト・セッションとのテキスト・ベースの対話を管理するための JavaScript API が提供されています。
- **「Programmable Host On-Demand」。** この資料には、Host On-Demand クライアント・コードの各種の部分 (ターミナル、メニュー、ツールバーなど) を開発者が自分のカスタム Java アプリケーションおよびアプレットに組み込めるようにする一式の Java API が記載されています。
- **「Host Access Toolkit」。** この資料では、Host On-Demand Toolkit (Host Access Client Package に同梱されている) をインストールおよび構成する方法が説明されています。Host On-Demand Toolkit は、Host On-Demand 基本製品を補完するもので、お客様の環境で Host On-Demand を最大限に活用できるようにするための、Java Bean その他のコンポーネントを提供します。
- **「Host Access Beans for Java」。** 本書は Host On-Demand Toolkit の一部です。Java Bean を使用して Host On-Demand 環境をカスタマイズし、エミュレーター・セッションでのステップを自動化するマクロを作成したいとお考えのプログラマーのための参考になります。
- **「Host Access Class Library for Java」。** 本書は Host On-Demand Toolkit の一部です。データ・ストリーム・レベルでホスト情報にアクセスすることのできる Java アプレットおよびアプリケーションを作成したいとお考えのプログラマーのための参考になります。
- **「Host On-Demand J2EE コネクター」。** 本書は Host On-Demand Toolkit の一部です。Java Enterprise Edition (J2EE) と互換性のあるアプリケーションにアクセスするアプレットおよびサーブレットを作成したいとお考えのプログラマーのための参考になります。

本書で使用する規則

「*Host On-Demand* 計画、インストール、および構成」では、以下の表記規則が使用されています。

表 1. 本書で使用する規則

規則	意味
モノスペース	コマンド・プロンプトで入力する必要のあるテキスト、およびコマンド、関数、ならびにリソース定義属性やそれらの値など、示されているとおりに使用しなければならない値を示します。モノスペースは、画面テキストおよびコード例を示すためにも使用されます。
イタリック	指定する必要がある可変値を示します (例: <i>file_name</i> にファイルの名前を指定します)。イタリックは、強調表示および書名の表示にも使用されます。
Return	Return、Enter または左矢印がラベルに示されているキーを指します。
>	メニューの記述に使用している場合は、一連のメニュー選択を示します。例えば、「「ファイル」>「新規」とクリックします」と書かれている場合、「ファイル・メニューから新規コマンドをクリックする」ことを意味します。 ツリー表示の記述に使用している場合は、一連のフォルダーまたはオブジェクトの展開を示します。例えば、「「HODConfig サブレット」>「シスプレックス」>「Plex1」>「J2EE サーバー」>「BBOARS2」と展開します」と書かれている場合、次のような操作を意味します。 1. 「HODConfig サブレット」フォルダーを展開する 2. 「シスプレックス」フォルダーを展開する 3. 「Plex1」フォルダーを展開する 4. 「J2EE サーバー」フォルダーを展開する 5. 「BBOARS2」フォルダーを展開する



このグラフィックは、読者に対して注意点を強調するために使用されます。



読者のためのヒントはこのグラフィックで示しています。

用語

この節では、本書で使用する用語について説明します。

アプレット

Java で記述されたプログラムであり、HTML ファイルにおいて参照されます。アプレットは、Web ブラウザーで実行されている Java 仮想マシン (JVM) によって起動されます。

アプリケーション

タスクまたは特定の機能を実行する 1 つのプログラムまたは一組のプログラム。

キャッシュ・クライアント

Host On-Demand キャッシュ・クライアントは、コンポーネントがユーザーのワークステーションのハード・ディスクにキャッシュされた (高速アクセスできるように保管された) 任意の Host On-Demand クライアントです。

デフォルト・パブリッシュ・ディレクトリー

デフォルト・パブリッシュ・ディレクトリーは、Host On-Demand サーバーのインストール・ディレクトリーにあるサブディレクトリー HOD です。例えば、c:\Program Files\IBM\HostOnDemand\HOD¥ (Windows オペレーティング・システムの場合)、/opt/IBM/HostOnDemand/HOD (AIX、Linux、Solaris の場合)、/QIBM/Programs/IBM/HostOnDemand/HOD (i as/400) オペレーティング・システムの場合)、/usr/lpp/HOD/hostondemand/HOD (z/OS オペレーティング・システムの場合) です。

ダウンロード・クライアント

ダウンロード・クライアントは、ユーザーが HTML ファイルにアクセスするごとに必要なアプレット・ファイルをダウンロードします。ダウンロード・クライアントは一般的に、高速なネットワーク接続を利用し、Web サーバーからのダウンロードにかかる時間が削減できるような LAN 接続環境で使用されます。

エミュレーター・クライアント

エミュレーター・クライアントは、端末エミュレーター・セッションを起動する Host On-Demand クライアントです。Host On-Demand には、以下のエミュレーター・クライアントが組み込まれています。すなわち、キャッシュ・クライアント、Web Start クライアント、およびダウンロード・クライアントです。

別個のユーザー・パブリッシュ・ディレクトリー

カスタム HTML ファイルを配置するための別個の書き込み可能ロケーションを提供し、Host On-Demand が提供するファイルからそれらを隔離します。これによって Host On-Demand パブリッシュ・ディレクトリーは読み取り専用に保たれ、今後の Host On-Demand アップグレードの適用が容易になります。

注: ユーザーが変更した他のファイル (カスタム・アプレットや HACL プログラムなど) は、依然として Host On-Demand パブリッシュ・ディレクトリーから実行する必要があります。

Web アプリケーション・サーバー

動的な Web アプリケーションのためのランタイム。Web アプリケーション・サーバーには、Java サーブレット、JavaServer Pages (JSP)、および他のエンタープライズ Java アプリケーション・プログラミング・インターフェース (API) のサポートが組み込まれています。Web アプリケーション・サーバーは、通信、リソース管理、セキュリティ、トランザクション管理、および持続性の能力を Web アプリケーションに提供します。また、通常は、サーバーおよび配置されるアプリケーションを管理するための管理インターフェースを提供します。

Web サーバー

HTTP 文書のための要求に応える Web 上のサーバー。Web サーバーは、ブラウザーとの間のトランザクションの流れを制御します。カスタマー・トランザクションの機密性を保護して、ユーザーの識別情報が安全にサーバーへ伝送されるようにします。

Web Start クライアント

Web Start クライアントにより、ブラウザーがなくてもユーザーは Host

On-Demand セッションを実行できるようになります。ユーザーは、Host On-Demand セッションを Java Web Start Application Manager から開始します。

Java に関連する用語

以下の用語と本書におけるそれらの使用法に留意してください。

Java HOD サーバーでも HOD クライアントでも、Java Runtime Environment (JRE) を指します。

Java 対応ブラウザ

インストール済みの Java プラグインの Java JVM 上で Java アプレットを実行する Web ブラウザーで、例として、Java プラグインがインストール済みの Firefox や Internet Explorer があります。詳しくは、18 ページの『ブラウザと Java プラグイン』を参照してください。

Java エミュレーター・クライアント、**Java** キャッシュ・クライアント、**Java** ダウンロード・クライアント

Host On-Demand クライアントの 1 つのバージョン。Java バージョンは、Java コンパイラーでコンパイルされた Host On-Demand クライアント・コンポーネントの完全セットで構成されています。

第 1 部 Host On-Demand の計画

第 1 章 IBM Host On-Demand の紹介

Host On-Demand について

IBM Host On-Demand は、イントラネット・ベースおよびエクストラネット・ベースの環境下のユーザーのために、コスト効果が高く、セキュアなブラウザー・ベースおよび非ブラウザー・ベースのホスト・アクセス手段を提供します。Host On-Demand は Web サーバーにインストールされるため、管理およびデプロイメントを簡単に行うことができ、Host On-Demand アプレットまたはアプリケーションはクライアント・ブラウザーまたはワークステーションにダウンロードされて、ユーザーが重要なホスト・アプリケーションおよびデータに接続できるようになります。

Host On-Demand は、以下のものを含む一般的な端末タイプ、通信プロトコル、通信ゲートウェイ、およびプリンターのエミュレーションをサポートします。

- TN3270 および TN3270E 端末
- TN5250 端末
- VT52、VT100、VT220、VT320、および VT420 端末
- Secure Shell (SSH)
- Transport Layer Security (TLS)
- ファイル転送プロトコル (FTP)
- CICS (顧客情報管理システム) トランザクション・ゲートウェイ
- TN3270E および TN5250 プリンター

Java コンポーネント・ベースの Host Access Toolkit を使用して、カスタマイズされた e-business アプリケーションを作成することができます。この Toolkit には、豊富な Java ライブラリーとアプリケーション・プログラミング・インターフェースのセットが含まれます。これには、ホスト・アクセス・クラス・ライブラリー (HACL)、Host Access Beans for Java、および Java Enterprise Edition (J2EE) コネクターが含まれます。Host On-Demand には、Database On-Demand も含まれています。これは、IBM System i7 システムがホスティングする IBM DB2 データベースに構造化照会言語 (SQL) 照会を送るためのインターフェースを提供します。

Host On-Demand の機能の仕組み

次の図と解説は、Host On-Demand システムが機能する仕組みを示しています。Host On-Demand はクライアント/サーバー・システムです。Host On-Demand のクライアントは、Web サーバーからリモート・コンピューター上にある Web ブラウザーにダウンロードされる Java アプレットです。

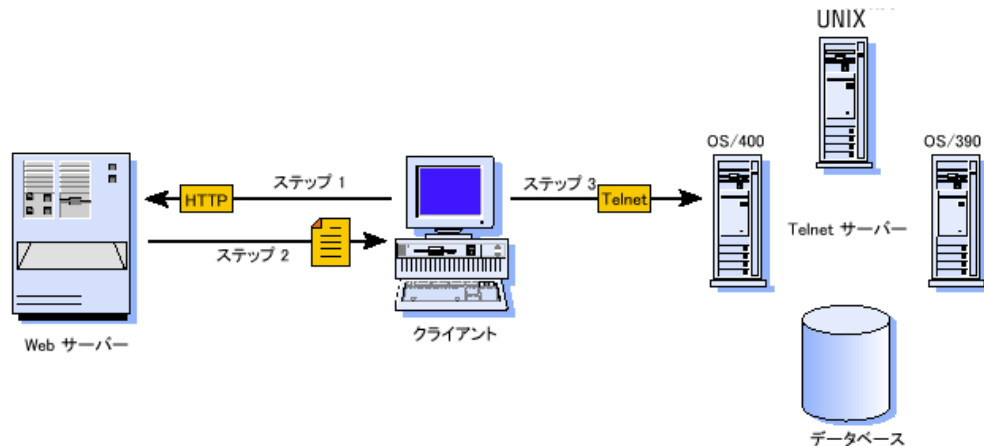


図 1. Host On-Demand の機能の仕組み

ステップ 1. ユーザーがブラウザを開いてハイパーリンクをクリックします。

ステップ 2. IBM Host On-Demand アプレットがクライアント・ワークステーションにダウンロードされます。

ステップ 3. アプレットがダウンロードされると、IBM Host On-Demand は、ホスト・アプリケーションにアクセスするために、任意の Telnet サーバーに直接接続します。

セッション情報は HTML ファイルまたは Host On-Demand 構成サーバー内に構成されます。構成サーバーの詳細については、11 ページの『第 2 章 配置のための計画』を参照してください。

Host On-Demand クライアント・アプレットは、ダウンロード・クライアント、Web Start クライアント、またはキャッシュ・クライアントとして実行することができます。ダウンロード・クライアントは、使用されるたびに Web サーバーからダウンロードされます。キャッシュ・クライアントと Web Start クライアントは、Web サーバーからダウンロードされ、クライアント・コンピューターに保管されます。最初にダウンロードされた後、キャッシュ・クライアントはローカル・マシンからロードされます。キャッシュ・クライアントは、Host On-Demand サーバーにクライアントの新規バージョンがあるかどうかを調べ、自動的に更新済みバージョンをダウンロードします。

Host On-Demand には、以下の管理コンポーネントが含まれています。

- デプロイメント・ウィザード。これは、エミュレーター・クライアントの HTML ファイルを作成するためのツールです。管理者はデプロイメント・ウィザードを使用することにより、ある組織の必要に合わせてカスタマイズされた Host On-Demand HTML ファイルを、迅速かつ簡単に作成することができます。
- システム管理者が Host On-Demand サーバーで共通セッションの定義、ユーザーおよびグループの作成、およびその他の管理用タスクの実行を行うために使用することのできる、管理クライアント。

さらに、ユーザーや管理者に Host On-Demand のクライアント機能を示すために、多くの定義済みクライアント (例えば、エミュレーション、Database On-Demand、キャッシュ・クライアントの除去、および問題判別ユーティリティ) が Host On-Demand とともに提供されています。

Host On-Demand の使用目的

コスト効率の高い接続方法

Web サーバー上に Host On-Demand をインストールすることにより、ユーザー・デスクトップを個別に管理する必要がなくなるため、保守のコストを削減し、投資効果を向上させることができます。

アプレットはサーバー上に常駐し、必要に応じて Web ブラウザーにダウンロードされるため、保守およびアップグレードのスケジュールを立てる必要はありません。サーバー上でソフトウェアをアップグレードしておくと、ユーザーは次回にクライアント・アプレットにアクセスするときに、そのアップグレードを受け取ることができます。

構成データの集中管理

管理者は、接続オプション、セキュリティ・フィーチャー、マクロ定義、キーボード仕様、およびカラー・マッピングなど、ユーザーが使用できるすべてのセッション構成情報を、中央で定義および制御することができます。さらに、管理者は、どのフィールドをユーザーが変更できるようにし、どのフィールドを変更できないようにするかを完全に制御でき、また、ユーザーが行った更新内容をどこに保管するかを選択できます。

Windows プラットフォームでは、デフォルト Host On-Demand グラフィカル・ユーザー・インターフェースは、Java 1.6 以降で提供される Nimbus Look and Feel に基づいています。管理グラフィカル・ユーザー・インターフェース用の Nimbus Look and Feel は、**SETHODNIMBUSGUI** 環境変数値を `false` に設定することで無効にすることができます。

Telnet サーバーへの直接接続

Host On-Demand のクライアント・アプレットには、エミュレーション機能が含まれます。クライアント上に常駐するエミュレーターを使用すると、IBM Communications Server やサード・パーティーの SNA サーバーなどの中間層サーバーを不要にすることができます。この中間部分によって生じるパフォーマンスやセキュリティの問題もすべて取り除かれます。アプレットがクライアントで使用可能な状態になると、必要なデータに最も効率よくアクセスできる標準 Telnet サーバーに、容易に直接接続できるようになります。Telnet 接続を使用すると、多くのホスト・セッションに並行してアクセスすることができます。中間層サーバーを不要にすることにより、Host On-Demand は容量に対して加えられる制約を最小化します。機能の仕組みについては、4 ページの図 1 を参照してください。

ブラウザー・ベースのユーザー・インターフェース

Host On-Demand のブラウザー・ベースのアクセスにより、重要なホスト・アプリケーションおよびデータの集中管理とデプロイを簡単に行うことができます。Host On-Demand は、Java テクノロジーを活用することによって、ご使用のブラウザーから、必要なときに、どこからでもホスト・システムにアクセスできます。ハイパーリンクをクリックするだけで、Host On-Demand の Java アプレットを起動することができます。このような Web とホスト間の接続ソリューションでは、Java ベースのエミュレーションにより、Web ブラウザー経由でホスト・アプリケーションおよびシステム・データへのセキュア・アクセスを可能にしています。このため、プログラミングを行わずに、既存のホスト・アプリケーションを Web で使用することができます。Host On-Demand は Java をベースとしていることから、そのインターフェースのルック・アンド・フィールは、各種の操作環境間で共通になります。

Windows プラットフォームでは、デフォルト Host On-Demand クライアント・グラフィカル・ユーザー・インターフェースは、Java 1.6 以降で提供される Nimbus Look and Feel に基づいています。Nimbus Look and Feel は、**setHODNimbusGUI** HTML パラメーターまたは **SETHODNIMBUSGUI** 環境変数を使用して無効にすることができます。

注: Host On-Demand ポートレットは、そのポータル・サーバーの外観を継承します。

多様なプラットフォームおよびネットワーク環境のサポート

Host On-Demand のサーバーおよびクライアントは広範な種類のプラットフォームで使用でき、どのような TCP/IP ネットワーク経由でも使用できます。これにより、システムの設定をきわめて柔軟に行えるようになり、新たにハードウェアを購入しなくても、現在使用中のコンピューター環境に Host On-Demand をデプロイできるようになります。

Java のサポート

Host On-Demand は、Java 標準をサポートするブラウザーと互換性があります。さらに、Host On-Demand のいくつかの新規フィーチャーは、Java で提供される機能のみを利用するようになっています。

インターネット・プロトコルバージョン 6 のサポート

インターネット・プロトコル・バージョン 6 をサポートするためには、Java 1.4 以降が必要です。ただし、Host On-Demand バージョン 13 では Java 1.6 以降がサポートされています。

インターネット・プロトコル (IP) は、インターネット環境を通して送信元から送信先にデータを送付するために使用されるプロトコルです。IP は、上位のプロトコル層と物理ネットワークの中間に位置します。

インターネット・プロトコル・バージョン 6 は、インターネット・プロトコル・バージョン 4 を置き換えるものです。インターネット・プロトコル・バージョン 6 では、使用可能な IP アドレスの数が増え、ルーティングおよびネットワーク構成

に関して改善されています。インターネット・プロトコル・バージョン 6 とインターネット・プロトコル・バージョン 4 は、どちらも、Internet Engineering Task Force (IETF) によって設計されました。

現在、ほとんどのインターネットはインターネット・プロトコル・バージョン 4 を使用しています。今後何年かの間には、インターネット・プロトコル・バージョン 4 はインターネット・プロトコル・バージョン 6 に置き換えられるものと予想されます。



Host On-Demand サーバーは、リダイレクターに対してインターネット・プロトコル・バージョン 6 もサポートしています。詳しくは、32 ページの『IPv6 に対するリダイレクターのサポート』を参照してください。

多くの各国語のサポート

Host On-Demand は、2 バイト文字セット (DBCS) 言語を含む複数の言語で 사용할ことができます。欧州通貨記号のサポートや、アラビア語、ヘブライ語、およびタイ語などの多くの言語のキーボードおよびコード・ページのサポートも提供されます。同一のメディア上ですべての言語バージョンを使用でき、複数の言語バージョンに同時にアクセスすることができます。

セキュア接続

Host On-Demand は Transport Layer Security (TLS) を使用して、セキュア・ホスト・データ・アクセスをイントラネット、エクストラネット、およびインターネットにわたって拡張します。モバイル・ワーカーは、セキュア Web サイトにアクセスして認証を受け、セキュアなエンタープライズ・ホストとの通信を確立します。クライアント/サーバー証明書サポートにより、Host On-Demand は認証のためにデジタル証明書を Telnet サーバー (IBM Communications Server for z/OS など) に提示できます。

また、Host On-Demand は、ファイアウォールを含む環境で使用するように構成することもできます。ファイアウォール・ポートは、ユーザーの Host On-Demand セッション定義で定義された機能に合わせてオープンする必要があります。詳しくは、33 ページの『ファイアウォールでの Host On-Demand の使用』を参照してください。

カスタム HTML ファイル

Host On-Demand には、カスタム HTML ファイルの作成に使用できるデプロイメント・ウィザードが組み込まれています。これらのファイルを使用して、特定グループのユーザーのニーズを満たすために必要なクライアントおよび機能の内容を調整することができます。デプロイメント・ウィザードについての詳細は、73 ページの『第 8 章 Host On-Demand エミュレーター・クライアントの構成』を参照してください。

新規 e-business アプリケーションを作成するためのツールキット

Host On-Demand には、カスタマイズされた e-business アプリケーションを作成するための、Java コンポーネント・ベースの Host Access Toolkit が含まれていま

す。この Toolkit には、ホスト・アクセス・クラス・ライブラリー (HACL)、Host Access Beans for Java、および Java Enterprise Edition (J2EE) コネクターなどの、豊富な Java ライブラリーとアプリケーション・プログラミング・インターフェースのセットが含まれます。

HACL は、元来は人間との対話用に設計されたアプリケーションを実行するバックエンド・ホスト・マシンと対話するための、非ビジュアル API を備えています。ホスト・アプリケーションには、人間が読むことのできる文字表示、フォーマット済みフィールド、カラー・コーディング、およびキーボード応答が必要です。HACL は、従来のホスト画面表示 (緑色の画面) との対話を模倣するために必要な、特別な機能のクラスを備えています。HACL には GUI (可視コンポーネント) クラスが含まれていません。例えば、メインフレーム上で 2 次側アプリケーションとして実行される Java プログラムがあるとしします。2 次側アプリケーション・プログラムはまず、CICS データ・アプリケーションを実行する別のメインフレームと対話し、次いで、動的に生成される HTML ページを介してクライアント・ブラウザと対話します。2 次側アプリケーションはクライアント入力を解釈してシミュレートされた端末アクションに入れ、それは HACL API を使って CICS マシンに送られます。CICS マシンからの応答画面は HACL API を使ってキャプチャーされ、動的 HTML ページに変換され、それからクライアントに送り返されます。

Host On-Demand の J2EE コネクターは、3270、5250、CICS、および VT ホストと通信するリソース・アダプターのセットを備えています。これらのリソース・アダプターは、IBM Application Server などの、適切なアプリケーション・サーバーにデプロイされます。ユーザーは、WebSphere Studio Application Developer Integration Edition を介して Host On-Demand の J2EE コネクターで提供される API を使用して、Web アプリケーションを作成することができます。

Programmable Host On-Demand

Programmable Host On-Demand は一式の Java API であり、Host On-Demand クライアント・コードの各種の部分 (ターミナル、メニュー、ツールバーなど) を開発者が自分のカスタム Java アプリケーションおよびアプレットに組み込めるようにします。この API を使えば、Toolkit に含まれている Host Access Java Beans から始めなくても、開発者は Host On-Demand デスクトップを (ユーザーが見るとおりに) 完全に制御することができます。基礎となっている Host On-Demand コードが各種コンポーネントのすべての「配線」を処理します。これには、マクロ、キーボード再マップ、およびカラー再マップなどのユーザー・プリファレンスを今後の使用のためにローカル・ファイル・システムに保管することが含まれています。開発者が決める必要のあるのは Host On-Demand デスクトップのレイアウトのみです。詳しくは、「Programmable Host On-Demand 解説書」を参照してください。

Host On-Demand Session Manager API

Host Access Toolkit に付属のアプリケーション・プログラミング・インターフェース (API) に加えて、Host On-Demand は専門化された公用 API を提供しています。これは、JavaScript を使用して Web ページ内にホスト・セッションを組み込むためのサポートを提供します。これらの JavaScript ベースの API は、アプリケーション開発者がホスト・セッションを管理したり、ホスト・セッションとのテキ

スト・ベースの対話を管理したりするのを助けるもので、Host On-Demand Session Manager を通して使用可能です。詳しくは、「Session Manager API 解説書」を参照してください。

WebSphere Portal のサポート

Host On-Demand は、WebSphere Portal のコンポーネントである Portal Server で、ポートレットとして実行することができます。Portal Server の洗練されたデスクトップ管理およびセキュリティの機能を使用することで、管理者はユーザーのアクセス権限をより柔軟に制御することができ、ユーザーはポータル・デスクトップの外観および配置を制御することができます。

管理者は、デプロイメント・ウィザードを使用して、カスタマイズされた Host On-Demand ポートレットを簡単かつ迅速に作成して、Portal Server に直接ロードすることができます。

注: Portal Server は別個の製品であり、個別にインストールする必要があります。

IBM System i サーバー上の DB2 データベースへの接続

Host On-Demand に組み込まれた Database On-Demand により、Java データベース・コネクティビティ (JDBC) ドライバーを使用して、IBM System i5 サーバーに保管されている DB2 情報にアクセスできます。Database On-Demand は、JDBC ドライバーを介して IBM System i5 データベースへの構造化照会言語 (SQL) 要求を行うための Java アプレットです。Database On-Demand は、Host On-Demand アプレットとは別のアプレットであり、独立した HTML ファイルによって開始されます。また、端末エミュレーションと SQL 照会のサポートの両方が必要な場合は、エミュレーター・セッションの中からデータ転送サポートを使用して SQL 要求を実行できます。

新機能

Host On-Demand に関する最新情報の入手

Host On-Demand バージョン 13 に関する最新情報については、製品の readme ファイルを参照してください。

最新の製品情報については、Host On-Demand Web サイトにアクセスしてください。

Host On-Demand の最新技術情報については、Host On-Demand サポート・ポータルにアクセスしてください。

一般的なソフトウェア・サポート情報については、「Software Support Handbook」を参照してください

Host On-Demand バージョン 13 の新機能

Host On-Demand バージョン 13 には、以下の機能と機能強化が追加されました。

- 1. 新規デスクトップ検索機能により、HOD デスクトップ上でセッションの検索が容易になりました。これにより、セッション定義が多い場合のユーザビリティが向上しました。
- 2. **100% TLS 準拠**
- a. JSSE がセキュア接続 (HACL アプリケーションやリダイレクターなど) 用のデフォルトになりました。
- b. JSSE でプロキシ・サーバーがサポートされるようになりました。
- c. FTP セキュア接続でも TLS 接続がサポートされるようになりました。
- d. FTP クライアントに関して、鍵の使用や拡張鍵の使用に基づくクライアント証明書のフィルタリングがサポートされるようになりました。
- 3. フルスクリーン - ユーザーは、使い勝手を良くするためにデフォルト画面サイズとフルスクリーン・サイズを切り替えることができました。
- 4. 画面サイズ変更時に、表示域全体を占有するようにフォントが拡大されるようになりました。これにより、余分な空間がなくなります。
- 5. 新規ルック・アンド・フィールにより、ツールバーやメニュー・バーのアイコンが見やすくなりました。また、メニュー項目の配置が変更され、ユーザビリティが向上しました。
- 6. **HACP 拡張エディション** - HOD 13 で新規 HACP EE コンポーネントが追加されました。HACP EE はプラグインなしの Web ベースのエミュレーターです。
- 7. GSKit は Windows オペレーティング・システム用の Host On-Demand には同梱されなくなりました。リダイレクター接続ではセキュア接続に JSSE が使用されます。Windows 以外のオペレーティング・システムでは、現在のところ、引き続き GSKit が同梱されます。
- 8. マイグレーション・ユーティリティ - v12 から v13 へのマイグレーション、および v11 から v13 へのマイグレーションがシームレスに行われるように、新規コマンド行マイグレーション・ユーティリティが使用可能になりました。

第 2 章 配置のための計画

Host On-Demand は Web ブラウザーからホスト・アプリケーションへのアクセスを提供します。ブラウザーは Web サーバーから Host On-Demand Java アプレットをダウンロードしたうえで、ホスト・アプリケーションにアクセスするために任意の Telnet サーバーに接続します。Host On-Demand アプレットには、接続先ホストなどのホストのセッション・プロパティを判別するための構成情報が必要です。この構成情報は、Host On-Demand を起動するために使用される HTML ファイルから、または Host On-Demand 構成サーバーを使用して、Host On-Demand アプレットに提供できます。構成サーバーは Host On-Demand の一部で、ユーザー ID およびグループ ID ごとにセッション構成情報とユーザー設定を中央に保管します。これによりユーザーは、構成サーバーに接続してセッション情報とユーザー設定を取得します。構成サーバーは、管理クライアントを介して管理されます。Host On-Demand 構成サーバーの構成については、オンライン・ヘルプを参照してください。

Web ブラウザーで Java プラグインのサポートが廃止され始めると、Host On-Demand クライアントは、「IBM Host On-Demand browser support」の説明にあるように、いくつかの制限付きで動作し続けることになります。

デプロイメント・ウィザードを使用してカスタム・クライアント HTML ファイルを作成することができます。これらの HTML ファイルを作成するときには、3 種類の構成モデルから 1 つを選択して、セッション構成情報とユーザー設定を定義および管理する方法を指定できます。3 種類の構成モデルとは、HTML ベース・モデル、構成サーバー・ベース・モデル、および結合モデルです。

これらのモデルについて、以下で説明します。各モデルについての詳細、およびそれぞれを使用する場合の利点と制約については、オンライン・ヘルプを参照してください。

HTML ベース・モデルの理解

HTML ベース・モデルを選択した場合、その HTML ファイル自体にすべてのホスト・セッション構成情報が含まれるため、ホスト・セッションを定義するためには、ほかに必要な情報はありません。したがって、セッションを指定するために構成サーバーを使用する必要はありません。つまり、ファイアウォールにあるポートを開く必要はありません。ホスト・セッション構成情報についての変更をユーザーが保管できるようにした場合は、ユーザーの変更はブラウザーを実行しているローカル・ファイル・システムに保管されます。

HTML ベース・モデルを使用して HOD サーバーを始動する必要はないため、ポート 8999 は使用しないことをお勧めします。これにより、サーバー・リソースが節約されます。

HTML ファイルで構成情報を定義する際のこのオプションは、デプロイメント・ウィザードを使用して作成されたクライアントだけで使用可能です。

HTML ベース・モデル

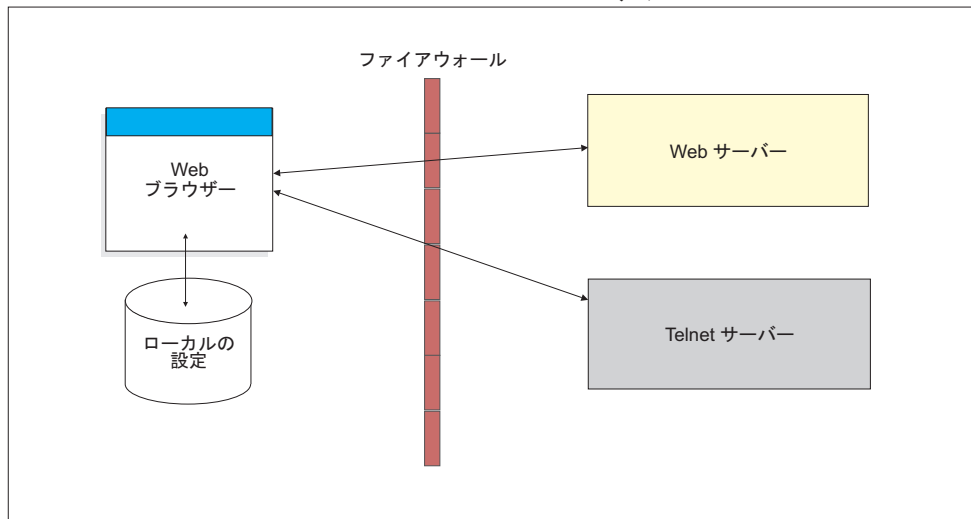


図 2. HTML ベース・モデル

構成サーバー・ベース・モデルの理解

構成サーバー・ベース・モデルでは、ホスト・セッション情報は管理クライアントを使用して構成サーバーで維持され、情報はユーザーおよびグループ構造を使用して定義されます。デフォルトでは、構成サーバーはデータを Host On-Demand サーバー・マシンに直接保管しますが、代わりに LDAP を使用するように構成することもできます。ユーザーはその構成を、デプロイメント・ウィザードに作成されたカスタム HTML ファイル、または Host On-Demand の一部として提供されているいくつかの HTML ファイルの 1 つを使用して利用できます。ユーザー ID は構成サーバーで定義され、ユーザーはほとんどの場合、セッションを表示する前に Host On-Demand サーバーにログオンする必要があります。管理者がユーザーに変更の保管を許可している場合は、ユーザーの設定はユーザー ID ごとに構成サーバーに保管されます。カスタマイズの内容は構成サーバーに保管されるため、複数のマシンからセッションにアクセスする必要がある場合には、このモデルが最適と思われます。

デフォルトでは、Web ブラウザーは構成サーバーと直接に通信します。ファイアウォールを介して通信する場合には、ファイアウォールにある構成サーバーのポートを開く必要があります。代わりに、構成サーブレットを使用することで、ファイアウォール上に構成サーバーのポートを開かなくても済むようにすることができます。Web ブラウザーは、HTTP 接続または HTTPS 接続を通して構成サーブレットに接続し、構成サーブレットは構成サーバーと対話します。構成サーブレットの使用についての詳細は、『構成サーブレットの構成 (Configuring the configuration servlet)』を参照してください。

構成サーバー・ベース・モデルおよび結合モデル

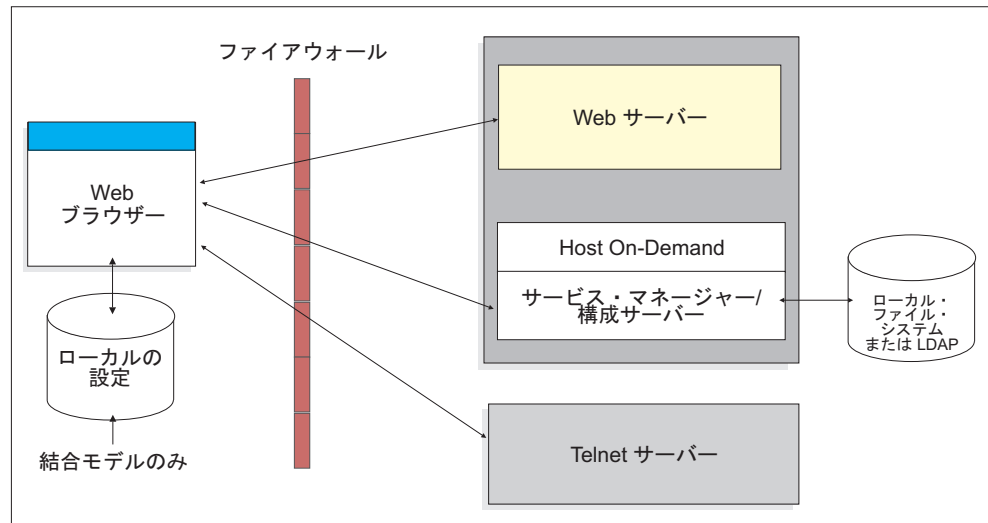


図 3. 構成サーバー・ベース・モデルおよび結合モデル

構成サーブレットを使用した 構成サーバー・ベース・モデルおよび結合モデル

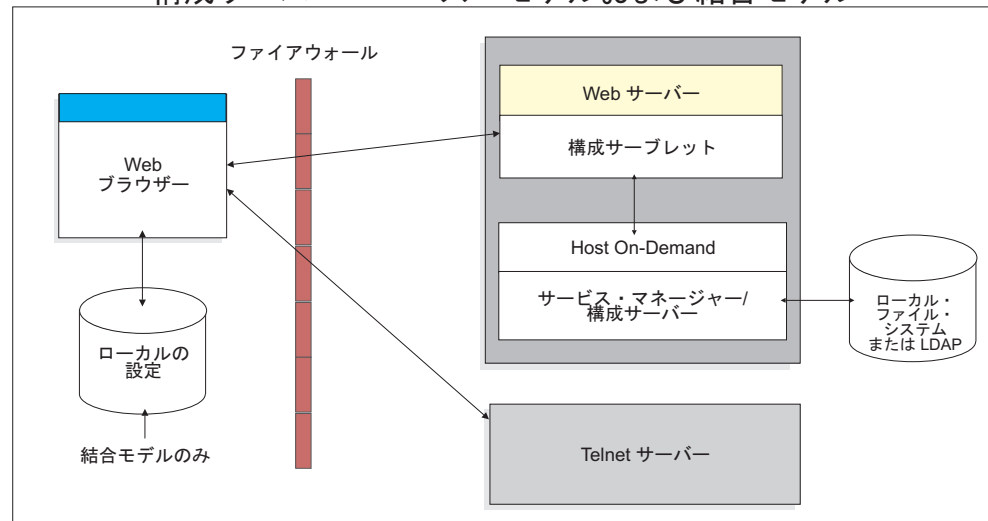


図 4. 構成サーブレットを使用した構成サーバー・ベース・モデルおよび結合モデル

結合モデルの理解

Host On-Demand は結合モデルをサポートします。このモデルでは、ホスト・セッション情報は (構成サーバー・ベース・モデルと同様に) 構成サーバーで定義され、ユーザーの更新内容は (HTML ベース・モデルと同様に) ユーザーのマシンに保管されます。また、HTML ベース・モデルの場合と同様に、結合モデルのユーザーは、セッションを表示するために Host On-Demand サーバーにログオンする必要がありません。

クライアント配置の考慮事項

また、クライアント配置に関する考慮事項として、キャッシュ・クライアント、ダウンロード・クライアント、または Web Start クライアントのいずれを使用するか (81 ページの『第 10 章 Host On-Demand エミュレーター・クライアントの使用』を参照)、また、どのバージョンの Java を使用するのか (15 ページの『第 3 章 クライアントでの Java の計画』を参照) を決める必要があります。

第 3 章 クライアントでの Java の計画

この章では、Java 対応のブラウザで Host On-Demand クライアントを実行することに関連した詳細情報が提供されます。

- 『Java 用のキャッシュ・クライアントの改善』では、Host On-Demand Java キャッシュ・クライアントの機能について説明します。
- 16 ページの『Java を使用したクライアントのダウンロード』では、Java 対応のブラウザでのみ使用可能な Host On-Demand クライアントの拡張フィーチャーについて説明します。
- 17 ページの『Java のある Mac OS X』では、Apple Mac OS X を Java で Host On-Demand クライアントとして使用することに関する問題を扱います。
- 18 ページの『ブラウザと Java プラグイン』では、Java 対応ブラウザおよび Java プラグインの使用に関する問題を扱います。

Java 用のキャッシュ・クライアントの改善

Java キャッシュ・クライアントを使用すると、以下を行うことができます。

- Java キャッシュ・クライアントを LAN ドライブまたは DVD ドライブからインストールします。詳しくは、85 ページの『キャッシュ・クライアントを LAN または DVD からインストール』を参照してください。
- Java キャッシュ・クライアントを、Windows 上の複数のユーザー間で共有します。詳しくは、90 ページの『Windows のキャッシュ・クライアント・サポート』を参照してください。
- Java プラグインのキャッシュを消去することなく、Java キャッシュ・クライアントを 1 つの操作で除去します。詳しくは、88 ページの『キャッシュ・クライアントの除去』を参照してください。
- Java キャッシュ・クライアントをバックグラウンドでアップグレードします。

注: 以下の制約事項が適用されます。

いくつかの Java キャッシュ・クライアント・タイプは、バックグラウンドでアップグレードすることができません。詳しくは、『サポートの制限』を参照してください。

Host On-Demand Java キャッシュ・クライアントのほとんどすべてがこれらの改善点をサポートします。Java Web Start クライアントもこれらの改善点をサポートします。

サポートの制限

以下のタイプの Java キャッシュ・クライアントは Java キャッシュ・クライアントの改善点をサポートしません。

- 印刷画面収集のための「プロセス収集 (Process Collection)」ウィンドウ
- VT ディスプレイ・セッションおよびセキュア・ファイル転送プロトコル (sftp) セッション用のセキュア・シェル (SSH) のサポート

- 自動 IME/即時変換
- 印刷画面の機能強化
- インターネット・プロトコル・バージョン 6 (IPv6)
- アクセシビリティ機能
- 重複キー・サポート
- カスタマイズ可能なポップアップ・キーパッド
- マウス・ホイールのサポート
- 双方向言語の場合、Unicode 文字を表示するための OS/400 コード化文字セット ID (CCSID) のサポートが提供されるようになりました。

Java を使用したクライアントのダウンロード

以下のセクションでは、Java を使用してクライアントをダウンロードする際の制限について説明します。

プリロード・リストにないコンポーネントをダウンロードできない

Java ダウンロード・クライアントを使用する際、ユーザーは、オリジナルのプリロード・リストにない Host On-Demand クライアント・コンポーネントをダウンロードできません。したがって、ユーザーが必要とする可能性のあるすべてのコンポーネントをプリロード・リストで指定する必要があります。

この制限は、プリロード・リストにないコンポーネントをダウンロードするダウンロード・クライアントが使用するメソッドと、Java プラグインによって課せられるセキュリティ制限との間の競合が原因です。

HTML ファイルにいくつかのコンポーネントが含まれていない

Java を使用すると、デフォルトのダウンロード・クライアント HTML ファイル (HOD_xx.html、xx はご使用の言語の 2 文字の接尾部) には、以下のクライアント・コンポーネントが含まれません。

- データ転送
- 5250 ファイル転送
- 5250 ホスト印刷サポート
- インポート/エクスポート
- SLP
- タイ語セッション
- FTP コード・ページ・コンバーター
- 双方向セッション
- 5250 ヒンディ語セッション
- ユーザー定義の文字設定値を使用する DBCS セッション
- DBCS セッションの ZipPrint

ダウンロード時間を短縮するために、IBM は、これらのあまり頻繁に使用されないコンポーネントを Java デフォルトのダウンロード HTML ファイルのプリロー

ド・リストから除去しました。ただし、Java ダウンロード・クライアントを使用する場合、プリロード・リストにないコンポーネントは後になってもダウンロードできません。

これらのコンポーネントの一部またはすべてをプリロード・リストに入れた場合、以下のアクションのいずれかを実行します。

- デプロイメント・ウィザードを使用して、必要なコンポーネントが含まれているダウンロード・クライアント/キャッシュ・クライアント Java HTML ファイルを作成します。
- ダウンロード・クライアント用のデフォルトの HTML ファイルの代わりに、キャッシュ・クライアント用のデフォルトの HTML ファイル (HODCached_xx.html、xx は言語の 2 文字の接尾部) を使用します。
- デフォルトのダウンロード・クライアントのデバッグ・バージョン (HODDebug_xx.html、xx は 2 文字の言語接尾部) を使用します。デバッグ・バージョンにはすべてのコンポーネントが含まれます。ただし、デフォルトのダウンロード・クライアントのデバッグ・バージョンは、非デバッグ・バージョンより大きくなります。

Java のある Mac OS X

Host On-Demand Mac OS X エミュレーターおよびデータベース・クライアントは、Safari、Firefox、および Mac バージョンの Internet Explorer をサポートします。Host On-Demand は、Mac OS X 上の管理クライアントをサポートしません。Host On-Demand バージョン 12.0 は、Java 1.6 以上をサポートします。

重複キー・サポート機能を利用するには、Java プラグイン 1.4.2 以上が Macintosh クライアントに存在する必要があります。ただし、Host On-Demand バージョン 12 では Java 1.6 以降がサポートされています。

Mac OS X の制限

Mac OS X は、15 ページの『Java 用のキャッシュ・クライアントの改善』で説明されている Java キャッシュ・クライアントの改善点をサポートしません。詳しくは、91 ページの『Mac OS X (Java クライアントのみ) の場合のキャッシュ・クライアントのサポート』を参照してください。

Java クライアントを使用すると起動時間が少し遅くなる

Java 対応のブラウザを使用すると、Host On-Demand クライアントは少し遅く (ワークステーション・タイプに応じて、5 から 15 秒) 始動します。この遅延は、システムが Java プラグインをロードするために生じます。

また、Java 対応ブラウザを使用すると、Host On-Demand クライアント・デスクトップ上のホスト・セッションは、開始に少し余分に時間がかかります。

Java プラグイン固有の制限

Oracle Java プラグインを使用している場合に、ヒンディ語文字が正しく表示されないときは、Oracle JRE レベルが最新になっていることを確認してください。

カスタマー提供のアプレットおよび Java に関する制限

ユーザーがカスタマー提供のアプレット (社内またはサード・パーティーで作成されたアプレット) を Java Host On-Demand クライアントから起動されたセッション (3270 ディスプレイなど) で実行し、このアプレットが何らかの Java 許可を必要とする場合は、Java のセキュリティ要件を満たすために、以下のいずれかのアクションを行ってください。

- 署名された Java .JAR ファイルにこのアプレットをアーカイブする必要があります。
- Java プラグインに付属する Java Policy Tool を使用して、ワークステーション上で許可を前もって付与されている必要があります。

Java のセキュリティ要件を満たしていない場合、アプレットは暗黙的に失敗します。

制限付きユーザーおよび Java に関する制限

アクセス制限されたユーザーには、Java プラグインをインストールする権限がありません。管理権限を持つユーザーが Java プラグインをインストールする必要があります。

ブラウザーと Java プラグイン

このセクションでは、Java 対応ブラウザーおよび Java プラグインの使用に関する問題について説明します。

Java 対応ブラウザー

Java 対応ブラウザーには JVM が組み込まれていません。これは HTML ファイルを表示できますが、別個の Java プラグインをインストールして、Host On-Demand クライアントといった Java アプレットを起動する必要があります。Java 対応のブラウザーの例は、Java プラグインがインストール済みの Firefox および Microsoft Internet Explorer です。

Host On-Demand クライアントでサポートされるブラウザーおよびプラグイン

Windows が稼働するクライアント・ワークステーションを使用しているユーザーは、会社が HOD にアクセスする資格を所有している限りは Fix Central の Host On-Demand 製品の下から IBM Java プラグインをダウンロードできます。

Oracle や IBM などの Java プラグイン・ベンダーが自分たちの Java プラグインの新規バージョンを公開するときや、IBM がその新規バージョンをサポートするように Host On-Demand を拡張するときは、IBM は新規バージョンの JRE をサポートするために「Software Product Compatibility Reports」を更新します。

Java プラグインを使用した Microsoft Internet Explorer

Windows クライアント・ワークステーション上で Java プラグインが正しくインストールおよび構成されている場合、Microsoft Internet Explorer は、Host On-Demand が選択するクライアントの起動方法に応じて、Java 対応ブラウザとして機能します。

Java プラグインを使用した Firefox

Firefox 上で Java アプレットを実行するには、Java プラグインをインストールする必要があります。

したがって、Host On-Demand は、Java プラグインが Firefox 用のデフォルトの Java ランタイムである ように構成されることを想定しています。

注: Windows 上でキャッシュ・クライアントを共有する制限付きユーザーや、Linux または AIX ワークステーション上の制限付きユーザーなど、制限付きユーザーは Java プラグインをインストールできません。

第 4 章 セキュリティーの計画

Host On-Demand を完全に企業ネットワーク内でのみインプリメントする場合でも、または Host On-Demand を使用してインターネット経由でホスト・システムへのアクセスを提供する場合でも、セキュリティは重要です。この章では、Host On-Demand のセキュリティの概要について述べます。

- Transport Layer Security (TLS). 確立された Telnet または FTP 接続を介して、暗号化、証明書ベースの認証、およびセキュリティ・ネゴシエーションを行います。詳しくは 23 ページの『Host On-Demand 用の TLS』を参照してください。
- リダイレクター。Host On-Demand クライアントと Host On-Demand サーバー間で TLS をサポートします。詳しくは 29 ページの『リダイレクター』を参照してください。
- ファイアウォール。ファイアウォールを経由するように Host On-Demand を構成することができます。詳しくは 33 ページの『ファイアウォールでの Host On-Demand の使用』を参照してください。
- ユーザー ID セキュリティー。これには、Web Express Logon、ネイティブ認証、Windows ドメイン・ログオンが含まれています。詳しくは 38 ページの『ユーザー ID セキュリティー』を参照してください。
- 連邦情報処理標準 (FIPS) 環境。お客様の環境でセキュリティ・コンポーネントが FIPS 認定のコンポーネント/モジュールを使用する必要がある場合は、39 ページの『FIPS 環境』を参照してください。

Transport Layer Security (TLS)

TLS セキュリティーの仕組み

TLS は SSL プロトコルをベースとしています。TLS は、クライアント/サーバー認証および暗号化の確立に初期ハンドシェイク・プロトコルを使用します。TLS について詳しくは、*TLS Protocol* バージョン 1.0 の説明を参照してください。

TLS プロトコルは、公開鍵および対称鍵の暗号テクノロジーを使用します。公開鍵暗号化では、1 対の公開鍵と秘密鍵を使用します。一方の鍵で暗号化された情報は、他方の鍵でのみ暗号化解除することができます。例えば、公開鍵で暗号化された情報は、秘密鍵でのみ暗号化解除することができます。各サーバーの公開鍵は公表されていますが、秘密鍵は秘密にされています。セキュア・メッセージをサーバーに送信するために、クライアントはメッセージをサーバーの公開鍵を使って暗号化します。サーバーは、サーバーの秘密鍵を使用して、受信したメッセージを復号します。

対称鍵暗号化では、同じ鍵を使用してメッセージを暗号化および暗号化解除します。クライアントは、すべてのセッション・データを暗号化するのに使用する対称鍵をランダムに生成します。これらの鍵はサーバーの公開鍵によって暗号化され、サーバーまで送信されます。

TLS は、以下の 3 つの基本セキュリティー・サービスを提供します。

メッセージ・プライバシー

公開鍵と対称鍵の暗号化を組み合わせることにより実現します。クライアントとサーバーとの間のすべてのトラフィックは、セッション・セットアップ時に折衝された鍵および暗号化アルゴリズムを使って暗号化されます。

メッセージ保全性

セッション・トラフィックがその最終宛先までの経路上で変更されないようにします。TLS は、公開/秘密鍵の組み合わせとハッシュ関数を使用してメッセージ保全性を確保します。

相互認証

公開鍵証明書を紹介して ID を交換します。クライアントとサーバーの身元は、公開鍵証明書でエンコードされています。公開鍵証明書には、以下のコンポーネントが含まれます。

- 対象の識別名
- 発行元の識別名
- 対象の公開鍵
- 発行元の署名
- 有効期間
- シリアル番号

表 2. ヒント



セキュア HTTP (HTTPS) を使用すると、クライアントのセキュリティー情報はサーバーからダウンロードされるため、確実に漏えいしないようにすることができます。

証明書

セキュリティーは、電子的な ID カードの役割を果たすデジタル証明書によって制御されます。プログラムまたはユーザーに対して、求められている接続を許可しても安全であるということ、暗号化が関係している場合は必要な暗号化鍵および暗号化解除鍵を付与しても安全であることを保証することが証明書の目的です。これらは通常、認証局 (CA) によって発行されます。認証局とは、インターネットの証明書の発行業務を行う、業界全体により承認されている組織です。CA 証明書 (ルート証明書とも言われる) には、CA の署名と有効期間が必ず含まれています。

暗号化と認証は、公開鍵と秘密鍵を 1 つずつ組み合わせることによって実行されます。公開鍵はサイトまたはサーバー証明書として知られている証明書に組み込まれています。この証明書には、証明書を発行している認証局 (CA) の名前、サーバーまたはクライアントの公開鍵の名前、CA の署名、および証明書の日付とシリアル番号などの、いくつかの情報項目があります。秘密鍵は、自己署名証明書の作成時や、CA 証明書要求が行われたときに作成され、クライアントからのメッセージを暗号化解除するために使用されます。

TLS セッションは、次の順序で確立されます。

1. クライアントとサーバーはハロー・メッセージを交換して、セッションで使用する (メッセージ保全性のための) 暗号化アルゴリズムとハッシュ関数とのネゴシエーションを行います。

2. クライアントは身元を証明する X.509 証明書をサーバーに要求します。オプションで、サーバーがクライアントに証明書を要求することもできます。証明書は、その書式と有効期間が正しいかどうか、および承認された認証局の署名が入っているかどうか (あるいは自己署名であるか) について調べられます。
3. クライアントは、暗号化に使用する鍵のセットをランダムに生成します。これらの鍵はサーバーの公開鍵によって暗号化され、保護されてサーバーまで送信されます。

Host On-Demand 用の TLS

Host On-Demand のセキュリティの構成は、セッション・セキュリティ、Web サーバー・セキュリティ、および構成セキュリティの、3 種類のセキュリティによって行うことができます。

セッション・セキュリティ

Host On-Demand バージョン 12.0 では、エミュレーターおよび FTP セッションのためのセキュリティを提供するために TLS プロトコルが使用されます。

TLS プロトコルは、TCP/IP ネットワークにおける通信プライバシーを提供します。TLS は、盗聴、メッセージの改ざん、またはメッセージの偽造を防ぐように設計されています。TLS には、新規暗号アルゴリズムを簡単に取り込めるようなフレームワークも備わっています。Host On-Demand では、*TLS Protocol* バージョン 1.0 に従って、エミュレーションおよび FTP セッションの暗号化とサーバー/クライアント認証がサポートされます。

以下のものがサポートされています。

- Host On-Demand クライアントと、TLS バージョン 1.0、1.1、1.2 をサポートする Telnet または FTP サーバーとの間の接続に関する、RSA タイプ 4 のデータ暗号化
- X.509 証明書
- 最大長 168 ビットのキーを使用したバルク暗号化アルゴリズム
- 最大長 2048 ビットのキーを使用した認証アルゴリズム
- サーバーおよびクライアントの認証
- クライアント・システム上のクライアント証明書の保管と使用のサポート
- サーバーからクライアント証明書を要求されたときにユーザーに出すプロンプト (オプション)
- セキュア・セッション標識。セッションが安全であることをユーザーに示すロック・アイコンが、セッション・ステータス・バーに表示されます。マウスをロック・アイコン上に移動すると、暗号化強度 (例えば 64、128、または 256) もロック・アイコンの横に表示されます。

Host On-Demand では、CA 証明書を使用することができますが、オンライン・ヘルプの『自己署名証明書の使用 (Using a self-signed certificate)』トピックに記載されているように、独自の自己署名証明書を作成することもできます。

以下のことを行うために、グラフィカルな証明書管理ユーティリティ (Windows および AIX プラットフォームで使用可能) が提供されています。

- 認証要求の作成

- 証明書の受信と保管
- 自己署名証明書の作成

IKEYCMD は、証明書管理ユーティリティーに加えて提供されている、鍵、証明書、および認証要求の管理に使用できるツールです。IKEYCMD は、機能的には証明書管理ユーティリティーに類似したものです。グラフィカル・インターフェースを使用しないでコマンド行から実行するようになっています。詳しくは、157 ページの『付録 B. IKEYCMD コマンド行インターフェースの使用』を参照してください。

TLS サービスをサポートするために、Host On-Demand は以下の 6 つのデータベースを使用します。

HODServerKeyDb.kdb

HODServerKeyDb.kdb は、Host On-Demand リダイレクター用に TLS を最初に構成するときに作成します。このデータベースには、サーバーの秘密鍵と証明書、および CA (または署名者) 証明書のリストが含まれています。これらの CA は既知のものと見なされ、Host On-Demand サーバーによって信頼されます。このデータベースに対して、他の CA (未知の CA) から証明書を追加したり、自分で作成して署名した証明書 (自己署名証明書) を追加したりすることができます。詳しくは、29 ページの『リダイレクター』を参照してください。

HODServerKeyStore.jks

GSKit ではなく Java Secure Socket Extension (JSSE) を使用するようにリダイレクターを構成できます。JSSE による構成を行うと、リダイレクターは秘密鍵と証明書を HODServerKeyStore.jks から読み取ります。詳しくは、リダイレクターを参照してください。

CustomizedCAs.p12

CustomizedCAs.p12 は、WellKnownTrusted リストに存在しない、未知の CA のルート証明書と自己署名証明書を含む、PKCS#12 フォーマット・ファイルです。CustomizedCAs.p12 ファイルは SSLite とともに使用されます。CustomizedCAs.jks は JSSE サポートとともに使用されます。自己署名証明書または未知の認証局 (CA) によって作成された証明書を使用する場合には、CustomizedCAs.p12 ファイルを作成または更新しなければなりません。Host On-Demand は、デフォルトでは CustomizedCAs.p12 ファイルのインストールを行いません。CustomizedCAs.p12 は、クライアントから証明書にアクセスできるようにする機能です。この機能は、クライアントとホストの間で行われる TLS ハンドシェイク・プロセスで使用されます。

CustomizedCAs.p12 ファイルは CustomizedCAs.class ファイルの推奨版です。このファイルは、旧リリースの Host On-Demand で作成された可能性があります。CustomizedCAs.class ファイルは Host On-Demand バージョン 7 以前のクライアントをサポートし、デフォルトではパブリッシュ・ディレクトリーにあります。Windows または AIX を実行している場合は、バージョン 12 にアップグレードすると、Host On-Demand のインストールが CustomizedCAs.class ファイルを自動的に検出し、新しい CustomizedCAs.p12 ファイルを作成して、パブリッシュ・ディレクトリーに置きます。この 2 つのファイルはパブリッシュ・ディレクトリーに保持され、さまざまなバージョンのクライアントで使用可能です。デフォルトの

パブリッシュ・ディレクトリーではない別のユーザー・パブリッシュ・ディレクトリーがある場合、Host On-Demand インストールは CustomizedCAs.class ファイルを検出せず、コマンド行で手動でマイグレーション・ツールを実行する必要があります。

Host On-Demand 証明書管理ユーティリティ (IKEYMAN) を使用して、初めて CustomizedCAs.p12 ファイルを作成する場合は、古い CustomizedCAs.class ファイルをパブリッシュ・ディレクトリーに保持し、以前のクライアントが新しいサーバーでも動作できるようにする必要があります。また、後で CustomizedCAs.p12 ファイルを更新する際には、変更が CustomizedCAs.class ファイルに反映されるようにする必要があります。Windows オペレーティング・システムでは、これらのファイルがデフォルトのパブリッシュ・ディレクトリー c:\Program Files\IBM\HostOnDemand\HOD にある場合、IKEYMAN を開いて CustomizedCAs.p12 ファイルを更新した後、IKEYMAN を閉じるたびに、CustomizedCAs.class ファイルは CustomizedCAs.p12 ファイルと共に自動的に更新されます。ファイルがデフォルトのパブリッシュ・ディレクトリーにない場合には、次のコマンドを使用して、パブリッシュ・ディレクトリーから逆マイグレーション・ツールを実行する必要があります。コマンドは表示上は 3 行になっていますが、1 行で入力してください。

```
..\hod_jre\jre\bin\java -cp ../lib\sm.zip;  
com.ibm.eNetwork.HOD.convert.CVT2SSLIGHT  
CustomizedCAs.p12 hod CustomizedCAs.class
```

AIX では、CustomizedCAs.p12 ファイルに対する変更を CustomizedCAs.class ファイルに反映するには、次のコマンドを使用して、この逆マイグレーション・ツールをパブリッシュ・ディレクトリーから手動で実行する必要があります。コマンドは表示上は 3 行になっていますが、1 行で入力してください。

```
../hod_jre/jre/bin/java -cp ../lib/sm.zip  
com.ibm.eNetwork.HOD.convert.CVT2SSLIGHT  
CustomizedCAs.p12 hod CustomizedCAs.class
```

CustomizedCAs.class

CustomizedCAs.class は、WellKnownTrusted リストに存在しない、未知の CA の証明書と自己署名証明書を含む、Java クラス・ファイルです。自己署名証明書または未知の認証局 (CA) によって作成された証明書を使用する場合には、CustomizedCAs.class ファイルを更新しなければなりません。しかし、Windows または AIX プラットフォームでは、証明書管理ユーティリティを使用して CustomizedCAs.class ファイルを作成または更新することはもはやできません。Host On-Demand バージョン 9 以降では、CustomizedCAs.p12 という名前の新しいバージョンのファイルのみを作成できます。ただし、どのクライアントもまだ古いフォーマットをサポートしています。詳しくは、上述の CustomizedCAs.p12 の説明を参照してください。

WellKnownTrustedCAs.class、WellKnownTrustedCAs.p12、および WellKnownTrustedCAs.jks

WellKnownTrustedCAs.class、WellKnownTrustedCAs.p12、および WellKnownTrustedCAs.jks は、Host On-Demand によって提供されるフ

ファイルであり、Host On-Demand が信頼するすべての CA の共通証明書が含まれています。これらのファイルは変更しないでください。

WellKnownTrustedCAs.class/

WellKnownTrustedCAs.p12、WellKnownTrustedCAs.jks および

CustomizedCAs.jks、さらに CustomizedCAs.p12 または CustomizedCAs.class のいずれかもしくは両方が、Host On-Demand パブリッシュ・ディレクトリーに存在していなければなりません。Host On-Demand クライアントは、TLS ハンドシェーク時に、サーバーの証明書の信頼性を確認するために、こうしたファイルを使用します。

CustomizedCAs.jks

CustomizedCAs.jks ファイルと CustomizedCAs.p12 ファイルは別々のファイルですが、どちらのファイルも機能は同じです。CustomizedCAs.jks ファイルは、既存の CustomizedCAs.p12 を JKS フォーマットに変換したり、このフォーマットで新しいファイルを作成したりすることで作成できます。Host On-Demand または keytool.exe コマンド行ツールによってインストールされた証明書管理ユーティリティを使用できます。このユーティリティは、この用途のために JRE で使用できる Java 鍵および証明書管理ツールです。

Host On-Demand クライアントのための基本 TLS の使用可能化

Host On-Demand クライアントのための TLS プロトコルを選択すると、基本的な TLS セッションが確立されます。TLS ネゴシエーション処理中に、サーバーはその証明書をクライアントに提示します。基本 TLS を使用可能にする場合、証明書は、クライアントが信頼している認証局によって署名されていなければなりません。クライアントは、最初に WellKnownTrustedCAs.class/WellKnownTrustedCAs.p12 を検査し、その後で CustomizedCAs.p12 または CustomizedCAs.class を検査します。TLS の有効化に JSSE を使用するように Host On-Demand が構成されている場合は、WellKnownTrustedCAs.jks ファイルおよび CustomizedCAs.jks ファイルが使用されます。これらのファイルから署名者が見つからなかった場合には、クライアントはセッションを拒否します。クライアントがこれらのファイルから署名者を検出した場合、そのセッションは確立されます。これが基本サーバー認証です。Host On-Demand のクライアント構成を使用すると、より拡張された形式のサーバー認証を構成することができます。詳しくは、以下のセクションを参照してください。

サーバー認証

クライアントとサーバー間のデータ交換を暗号化しても、クライアントが適切なサーバーと通信を行うことが保証されるわけではありません。このような危険を避けるために、サーバー認証を使用可能にし、サーバーの証明書が信頼できることを確認してから、クライアントが証明書にあるインターネット名とサーバーのインターネット名が一致することを確認できるようにします。両者が一致する場合、TLS のネゴシエーションが続行されます。一致しなければ、接続は即時に終了します。詳細については、オンライン・ヘルプの『サーバー認証 (Server Authentication)』を参照してください。

クライアント認証

クライアント認証は、サーバー認証と似ていますが、サーバー認証と異なる点は、クライアントが本人であることを確認するため、Telnet サーバーが

クライアントからの証明書を要求するという点です。Host On-Demand リダイレクターほか、すべてのサーバーがクライアント認証をサポートするわけではありません。クライアント認証を構成するには、次の事柄を実行しなければなりません。

- クライアントの証明書を入手する
- クライアントに証明書を送信する
- クライアント認証を使用するようにクライアントを構成する

詳細については、オンライン・ヘルプの『クライアント認証を使用するためのクライアント構成 (configuring clients to use client authentication)』を参照してください。

Express Logon

Express Logon の 2 つのタイプがあります。

- Web Express Logon: 「Web Express Logon」を使用すると、ユーザーはホスト・システムおよびホスト・アプリケーションにユーザー ID とパスワードを提供しなくてもログオンできます。この機能はネットワーク・セキュリティー・アプリケーションと連動して働き、ユーザーのネットワーク信任状を取得して、それをホスト信任状にマップし、何度もログオンする必要を無くします。ご使用のホストに応じて、この自動ログオン・プロセスはマクロ・ベースか、接続ベースで行えます。詳しくは、「Web Express Logon 解説書」を参照してください。
- 証明書高速ログオン (Certificate Express Logon): 「証明書高速ログオン」はマクロ・ベースで、これによりユーザーはユーザー ID とパスワードを入力することなくログオンできます。TLS のセッションおよびクライアント認証を構成し、Communications Server が高速ログオンをサポートするよう構成されていることが必要ですが、機能的には「Web Express Logon」と似ています。詳しくは、オンライン・ヘルプの『高速ログオン (Express logon)』を参照してください。

表 3. ヒント



Host On-Demand V9 以降の Web Express Logon は、クライアント・サイド証明書を使用するタイプのログオン自動化を提供します。このモデルは証明書ベースの Web Express Logon と呼ばれ、証明書高速ログオンとは大きく異なります。証明書 Express Logon では、ログイン・プロセスを自動化するように構成されている高速ログオン対応の TN3270 サーバーに対してユーザーを認証するために、クライアント証明書が使用されます。一方、証明書ベースの Web Express Logon では、クライアント証明書は、Web サーバーまたはネットワーク・セキュリティー・アプリケーションに対してユーザーを認証するために使用され、ログイン・プロセスの自動化はプラグインとマクロによって行われます。詳しくは、『Web Express Logon 解説書』を参照してください。

TLS ベースの Telnet セキュリティー

Telnet 折衝セキュリティーでは、確立した Telnet 接続において、クライアントと Telnet サーバー間でのセキュリティー・ネゴシエーションを行うことができます。Host On-Demand 3270 ディスプレイおよびプリンター・セッションに対して Telnet 折衝セキュリティーを構成することができます。

Host On-Demand クライアントで Telnet 折衝セキュリティを使用するには、Telnet サーバーが TLS ベースの Telnet セキュリティ (IETF インターネット・ドラフト 『*TLS-based Telnet Security*』 に説明があります) をサポートしていなければなりません。Communications Server for z/OS は、TLS ベースの Telnet セキュリティをサポートします。

Telnet 折衝セキュリティの詳細については、オンライン・ヘルプの『Telnet 折衝セキュリティの概説』を参照してください。Telnet サーバーでの TLS の構成の詳細については、Telnet サーバーの資料を、またセキュア Telnet サーバーへのクライアント接続の構成の詳細については、オンライン・ヘルプの『セキュリティ (Security)』のトピックを参照してください。

TLS ベースの FTP セキュリティ

Host On-Demand は、FTP セッションに対して、TLS ベースのセキュア・ファイル転送を提供します。FTP セッションは、ポート 990/989 に対する暗黙/無条件の TLS のネゴシエーションをサポートしません。そのため、セキュア FTP セッションに対してポート 990 を使用することはできません。FTP セッションは、他のすべてのポートに対しては、明示的/条件付き (AUTH コマンド) の TLS のネゴシエーションのみをサポートします。

FTP セッションのセキュリティ・プロパティは、エミュレーター・セッションのセキュリティ・プロパティから独立しています。統合された FTP セッションについては、FTP セッション・プロパティの新しい「セキュリティ」タブを使用して、FTP セキュリティ情報を構成する必要があります。エミュレーター・セッションをセキュアとして構成し、ファイル転送タイプを FTP に設定した場合、FTP セッションは自動的に保護されません。このような場合は、「OK」ボタンをクリックすると、「セキュア・ファイル転送セッションが必要な場合には、ファイル転送のデフォルトにセキュリティ情報を構成してください」というメッセージが表示されます。

TLS ベースのセキュア FTP 機能は、z/OS V1R2 以降でサポートされています。

セッション・セキュリティを使用する場合の例

セッション・セキュリティを使用する必要のある状況として、次の例を参照してください。

- インターネットを介して顧客から製品の注文を受け付ける場合。この状況では、顧客の情報 (クレジット・カード番号など) は必ず暗号化して、盗まれないようにする必要があります。また、顧客に送信する情報も必ず保護する必要があります。
- ご使用のホスト・コンピューター上で、供給業者またはビジネス・パートナーに情報をアクセスさせる場合。このデータには、ほかの誰にもアクセスさせたくありません。
- 社員がホスト・コンピューターの情報にリモート・サイトから、または出張先からアクセスできるようにする場合。
- 医師がどこからでも患者の記録にアクセスでき、認証されていない人々はその記録にアクセスできないようにする場合。

Web サーバー・セキュリティ

TLS を使用するように Web サーバーを構成して、Web サーバーからブラウザへのデータ・ストリームを暗号化することができます。Web サーバーで TLS を使用するための構成の詳細については、Web サーバーの資料を参照してください。ただし、クライアントがブラウザにロードされると、クライアントはホストと直接に通信を行います。ホスト・セッションに TLS セキュリティーを適用するように Host On-Demand を構成することができます。詳しくは、オンライン・ヘルプの『TLS の構成 (Configuring TLS)』を参照してください。

セキュリティの構成

HTML モデルを使用する場合、HTTPS を使用するとセッション構成情報が暗号化されます。それ以外のすべてのモデルでは、セッション構成情報を暗号化するには、構成サーバーと直接に通信するのではなく、(Web アプリケーション・サーバーを構成した後で) HTTPS を介して構成サーブレットを使用するように Host On-Demand を構成する必要があります。構成サーブレットのインストールの詳細については、本書の 67 ページの『構成サーブレットのインストール』を、また、構成サーブレットを使用するためのクライアントの構成の詳細については、オンライン・ヘルプの『構成サーブレットの構成 (configuring the configuration servlet)』を参照してください。

リダイレクター

リダイレクターは、Host On-Demand サーバーで稼働するサービスです。リダイレクターを使用すると、Host On-Demand クライアントは、Host On-Demand サーバーのリダイレクター・ポートに接続することで、Telnet サーバーと通信できます。

通常、Host On-Demand クライアントは次のように動作します。

- Host On-Demand サーバーに直接接続し、クライアント・コードをダウンロードしたり、共通 HTML ファイルにアクセスしたりします。
- また、3270、5250、VT、または CICS の各ホスト上で稼働する、またはこれらのホストに接続されている Telnet サーバーに、直接接続します。

ただし、リダイレクターを使用すると、リダイレクターはクライアントと Telnet サーバーの間の仲介として機能します。クライアントは、Telnet サーバーに直接接続する代わりに、Host On-Demand サーバーのリダイレクター・ポートに接続します。リダイレクターは、クライアントから受け取ったデータを、Telnet サーバーに送信します。Telnet サーバーが応答すると、リダイレクターは、Telnet サーバーから受信したデータを、クライアントに送信します。セッションが終了するまで、この処理が続けられます。

リダイレクターの使用目的

Telnet サーバーが TLS をサポートしておらず、リダイレクターがセキュア・セッションをサポートしているオペレーティング・システム (31 ページの『リダイレクターがサポートするオペレーティング・システム』を参照) のいずれかで Host On-Demand を実行している場合は、TLS をサポートするように Host On-Demand リダイレクターを構成できます。

表 4. ヒント



多くの Telnet サーバーは TLS をサポートします (例えば、zSeries、IBM System i、AIX、または NT 上の IBM Communications Server)。使用している Telnet サーバーが TLS をサポートする場合には、Telnet サーバーを使用することをぜひお勧めします。使用している Telnet サーバーが TLS をサポートしない場合には、Communications Server for AIX のリダイレクターのほうが Host On-Demand のリダイレクターよりも拡張性が優れています。

リダイレクターは透過的な Telnet プロキシとして動作し、ポートを再マップして Host On-Demand サーバーを他の Telnet サーバーに接続します。それぞれの定義済みサーバーで、ローカル・ポート番号のセットを構成できます。クライアントは、ターゲットの Telnet サーバーに直接接続する代わりに、Host On-Demand サーバーとポート番号に接続します。リダイレクターはローカル・ポート番号をターゲットのホスト・ポート番号にマップして接続を作成します。

表 5. 推奨



Telnet プロキシのための推奨されるソリューションは、Host On-Demand のリダイレクターの代わりに、WebSphere Application Server の Edge Component のフィーチャーであるロード・バランサー、または全体的なファイアウォール・ソリューションの一部としてアドレス変換を行う類似の製品を使用することです。

リダイレクターの動作

図 5 は、リダイレクターがクライアント・データを Telnet サーバーに送信し、Telnet サーバーからの応答データをクライアントに送信する仕組みを示しています。

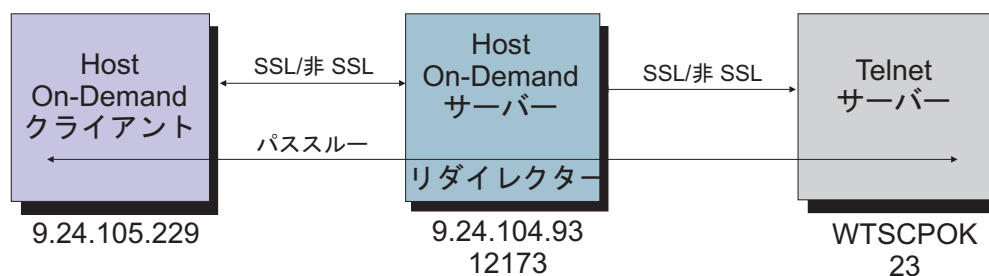


図 5. リダイレクターの動作

リダイレクターは、次の 4 つのモードのいずれかで構成できます。

- パススルー
 - リダイレクターは、データの内容を変更せずに、Telnet サーバーおよびクライアントと通信します。
- クライアント・サイド
 - クライアントとリダイレクターは、TLS を使用するセキュア・セッションで通信します (内容は暗号化/復号化されます)。
 - リダイレクターと Telnet サーバーは、非セキュア・セッションで通信します。
- ホスト・サイド

- クライアントとリダイレクターは、非セキュア・セッションで通信します。
 - リダイレクターと Telnet サーバーは、TLS を使用するセキュア・セッションで通信します (内容は暗号化/復号化されます)。
- 両方
 - クライアントとリダイレクターは、TLS を使用するセキュア・セッションで通信します (内容は暗号化/復号化されます)。
 - リダイレクターと Telnet サーバーは、TLS を使用するセキュア・セッションで通信します (内容は暗号化/復号化されます)。

クライアント・サイド、サーバー・サイド、またはその両方のモードを使用する前に、リダイレクター用に HODServerKeyDb.kdb または HODServerKeyStore.jks (JSSE を使用するように構成されている場合) を作成する必要があります。

データ・ストリームを暗号化する必要がないために、またはクライアントと Telnet サーバー間でデータ・ストリームが既に暗号化されているために、リダイレクターによる暗号化が必要ない場合には、パススルー・モードを使用できます。Host On-Demand クライアントが、クライアント認証または Express Logon を必要とするホストにリダイレクター経由で接続している場合は、パススルー・モードを使用する必要があります。

詳しくは、オンライン・ヘルプの『リダイレクターへのホストの追加 (Adding a host to the Redirector)』を参照してください。

リダイレクターのロード容量

リダイレクターの推奨ロード容量については、README ファイルを参照してください。

リダイレクターがサポートするオペレーティング・システム

リダイレクターは、現在、以下のオペレーティング・システムをサポートしています。

- Host On-Demand サーバーによってサポートされ、かつ、インターネット・プロトコル・バージョン 4 (IPv4) もサポートするすべてのオペレーティング・システム。
- Host On-Demand サーバーによってサポートされ、かつ、インターネット・プロトコル・バージョン 6 (IPv6) もサポートする一部のオペレーティング・システム。

すべてのオペレーティング・システムが、すべてのリダイレクター・モードをサポートしているわけではありません。次の 2 つの節では、リダイレクターのサポートについてさらに詳細に説明します。IPv4 および IPv6 について詳しくは、6 ページの『インターネット・プロトコルバージョン 6 のサポート』を参照してください。

IPv4 をサポートするオペレーティング・システム

IPv4 をサポートするオペレーティング・システムについては、リダイレクターは以下をサポートします。

- パススルー・モードは、Host On-Demand サーバーがサポートするすべてのオペレーティング・システムでサポートされます。
- 他のモード (クライアント・サイド、サーバー・サイド、および両方) は、Host On-Demand サーバーがサポートするオペレーティング・システムの一部でのみサポートされます。

注: z/OS および iSeries では、これらのモードはサポートされません。

表 6 と 表 7 にこの情報が示されています。

表 6. リダイレクターが GSKit を使用して IPv4 をサポートする 32 ビット・オペレーティング・システムとリダイレクター・モード

オペレーティング・システム:	パススルー:	クライアント・サイド:	ホスト・サイド:	両方:
Windows	はい	はい	はい	はい
AIX	はい	はい	はい	はい
Linux	はい	はい	はい	はい
他のすべてのオペレーティング・システム	はい	いいえ	いいえ	いいえ

表 7. リダイレクターが JSEE を使用して IPv4 をサポートする 64 ビット・オペレーティング・システムとリダイレクター・モード

オペレーティング・システム	パススルー:	クライアント・サイド:	ホスト・サイド:	両方:
Windows	はい	はい	はい	はい
AIX	はい	はい	はい	はい
Linux	はい	はい	はい	はい
他のすべてのオペレーティング・システム	はい	いいえ	いいえ	いいえ

IPv6 に対するリダイレクターのサポート

表 8 と 33 ページの表 9 に、リダイレクターがインターネット・プロトコル・バージョン 6 (IPv6) をサポートするオペレーティング・システムとリダイレクター・モードが示されています。

表 8. リダイレクターが GSKit を使用して IPv6 をサポートする 32 ビット・オペレーティング・システムとリダイレクター・モード

オペレーティング・システム	パススルー:	クライアント・サイド:	ホスト・サイド:	両方:
Windows	はい	はい	はい	はい
Linux	はい	はい	はい	はい
AIX	はい	はい	はい	はい

表 9. リダイレクターが JSEE を使用して IPv6 をサポートする 64 ビット・オペレーティング・システムとリダイレクター・モード

オペレーティング・システム:	パススルー:	クライアント・サイド:	ホスト・サイド:	両方:
Windows	はい	はい	はい	はい
Linux	はい	はい	はい	はい
AIX	はい	はい	はい	はい

ファイアウォールでの Host On-Demand の使用

ファイアウォールを介するよう Host On-Demand を構成する場合は、ファイアウォール管理者は、クライアントが機能するために必要となるポートのみを開くようにしてください。Telnet ポートを使用すると、TLS で暗号化されたセッションのトラフィックが許可されます。

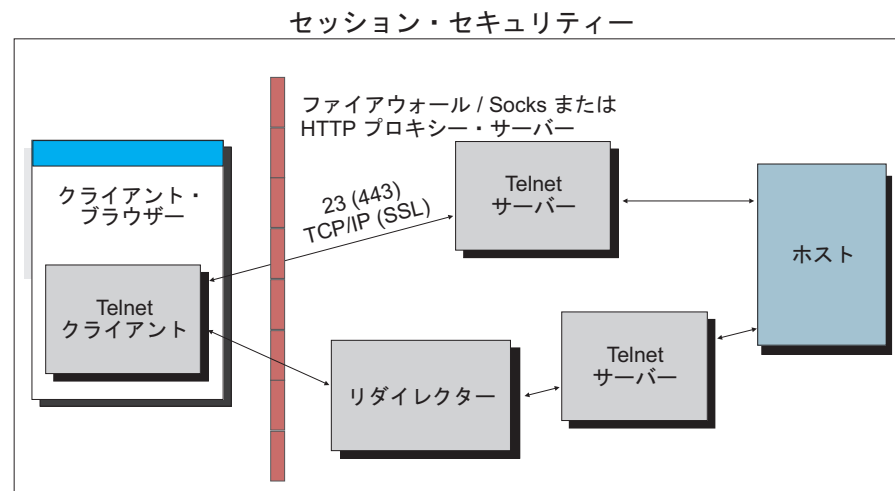


図 6. ファイアウォールまたはプロキシ・サーバーを介したセッション・セキュリティ

構成サーバー・ベース・モデルまたは結合モデルを使用している場合には、Host On-Demand 構成サブレットを使用することで、Host On-Demand クライアントは HTTP または HTTPS を介して構成サーバーと通信できます。

構成セキュリティ

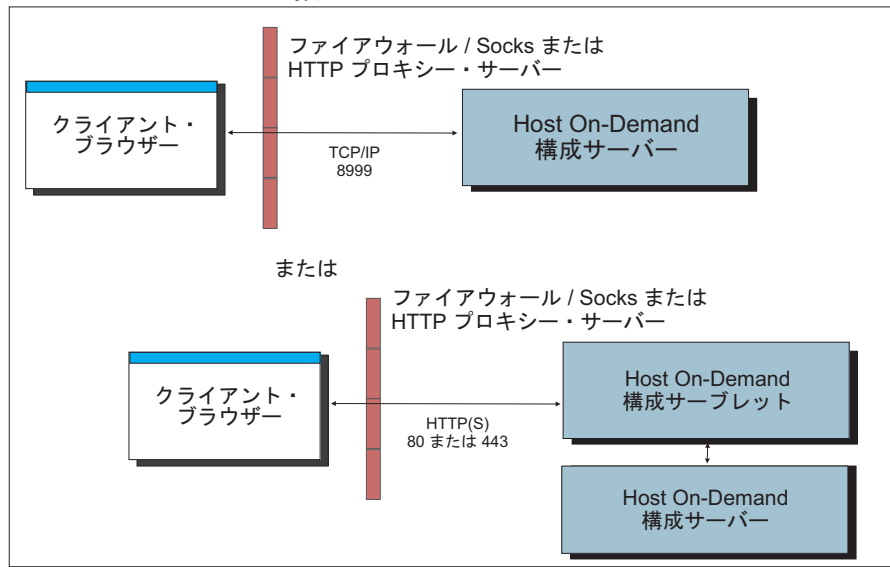


図 7. ファイアウォールまたはプロキシ・サーバーを介した、構成サーブレットを使用する構成セキュリティと、構成サーブレットを使用しない構成セキュリティ

ファイアウォールのオープン・ポート経由でホスト・システムに接続している Host On-Demand クライアントの場合、詳しくは『ファイアウォール・ポートの構成』を参照してください。Socks または HTTP プロキシ・サーバー経由でホスト・システムに接続している Host On-Demand クライアントの場合、詳しくは 37 ページの『プロキシ・サーバーを介したホスト・システムへの接続』を参照してください。

ファイアウォール・ポートの構成

構成サーバー・ベース・モデルまたは結合モデルを使用している場合、Host On-Demand クライアントは構成サーバーと通信する必要があります。ファイアウォールを介してこれを行えるようにするためには、Host On-Demand のサービス・マネージャー・ポートを開くか、あるいは Host On-Demand 構成サーブレットを使用する必要があります。サービス・マネージャーは、デフォルトではポート 8999 で listen します。このデフォルトを変更して、使用可能な別のポート番号を使用することもできます。詳しくは、オンライン・ヘルプの『サービス・マネージャー・ポートの変更 (Changing the Service Manager port)』を参照してください。Host On-Demand 構成サーブレットは、Host On-Demand クライアントが HTTP または HTTPS を介して構成サーバーと通信できるようにします。したがって、サービス・マネージャー・ポートをファイアウォールで開く必要はありません。(13 ページの図 4 を参照してください。) 構成サーブレットの詳しい使用法については、67 ページの『構成サーブレットのインストール』およびオンライン・ヘルプの『構成サーブレットの構成 (Configuring the configuration servlet)』を参照してください。

HTML ベース・モデルを使用している場合には、Host On-Demand クライアントが構成サーバーにアクセスするために特に要件が必要となることはなく、また、サービス・マネージャー・ポートをファイアウォールで開く必要もありません。この

場合にも、クライアントは、ライセンスをカウントするために構成サーバーに接続しようと試みますが、サービス・マネージャー・ポートが開いていない場合には、何も行わずに失敗します。

ファイアウォール管理者は、サービス・マネージャー・ポートに加え、クライアントが使用する機能によって使用されるポートを開くようにしてください。例えば、ポート 5000 上のリダイレクターとの TLS セッションを行う場合は、ポート 5000 を開いて Telnet トラフィックを確保する必要があります。以下の表に、Host On-Demand で使用可能なポートをまとめます。

表 10. Host On-Demand の機能と、それによって使用されるポート

Host On-Demand の機能	使用されるポート
ディスプレイ・エミュレーション (3270 および VT) および 3270 プリンター・エミュレーション	23 (Telnet)、80 (HTTP)、または 443 (TLS) および 8999 (構成サーバー) ³
5250 ディスプレイおよびプリンター・エミュレーション	23 (Telnet) または 992 ¹ (TLS) または 80 (HTTP) または 443 (TLS) および 8999 (構成サーバー) ³
3270 ファイル転送	23 (Telnet)、80 (HTTP)、または 443 (TLS) および 8999 (構成サーバー) ³
5250 ファイル転送 - savfile	80 (HTTP)、8999 (構成サーバー) ³ 、21 (FTP) ⁴ 、>1024 (FTP) ⁴ 、446 (drda) ⁴ 、449 (as-svrmap) ⁴ 、8470 (as-central) ^{1 2 4} 、8473 (as-file) ^{1 4} 、8475 (as-rmtcmd) ^{1 4} 、および 8476 (as-signon) ^{1 4}
5250 ファイル転送 - データベース	80 (HTTP)、8999 (構成サーバー) ³ 、446 (drda) ⁴ 、449 (as-svrmap) ⁴ 、8470 (as-central) ^{1 2 4} 、8473 (as-file) ^{1 4} 、8475 (as-rmtcmd) ^{1 4} 、および 8476 (as-signon) ^{1 4}
5250 ファイル転送 - ストリーム・ファイル	80 (HTTP)、8999 (構成サーバー) ^{1 2 4} 、449 (as-svrmap) ⁴ 、8470 (as-central) ^{1 2 4} 、8473 (as-file) ^{1 4} 、および 8476 (as-signon) ^{1 4}
FTP	21 (FTP)、80 (HTTP)、8999 (構成サーバー) ^{1 2 4} 、および >1024 (FTP) ⁵
CICS	2006
Database On-Demand	80 (HTTP)、8999 (構成サーバー) ³ 、449 (as-svrmap) ⁴ 、8470 (as-central) ^{1 2 4} 、8471 (as-database) ^{1 4} 、および 8476 (as-signon) ^{1 4}
Host On-Demand クライアント	23 (Telnet)、80 (HTTP)、および 8999 (構成サーバー) ³
管理クライアント	80 (HTTP) および 8999 (構成サーバー) ³
SSH (Secure Shell)	22

表 11. 注

注:

- 1 ポート番号は、コマンド WRKSRVTBLE で変更できます。リストされたポート番号は、デフォルト値です。

表 11. 注 (続き)

- 2 as-central 用のポートは、コード・ページ変換テーブルを (EBCDIC と Unicode 間で) 動的に作成する必要がある場合にのみ使用します。これは、JVM およびクライアントのロケールに依存します。
- 3 構成サーバー・ポートは変更可能です。ポート 8999 がデフォルトです。
- 4 IBM System i プロキシ・サーバー・サポートを使用している場合、ファイアウォール上でこれらのポートを開く必要はありません。デフォルトのプロキシ・サーバー・ポート 3470 を開く必要があります。このポートは変更可能です。
- 5 受動 (PASV) モードの場合には、FTP クライアントがサーバーへの両方の接続を開始し、サーバーからクライアントへの着信データ・ポート接続がファイアウォールによってフィルター操作されるという問題を解決します。FTP 接続を開くときに、クライアントはローカルで 2 つのランダムな非特権ポート ($N > 1024$ および $N+1$) を開きます。最初のポートはポート 21 のサーバーと接続しますが、そのときに PORT コマンドを出してサーバーがそのデータ・ポートに接続できるようにするのではなく、PASV コマンドを発行します。その結果、サーバーはランダムな非特権ポート ($P > 1024$) を開き、PORT P コマンドをクライアントに送ります。クライアントは、データを転送するために、ポート $N+1$ からサーバー上のポート P への接続を開始します。

サーバー側ファイアウォールの立場からは、受動モードの FTP をサポートするためには、以下の通信ポートを開かなければなりません。

- 任意の場所から FTP サーバーのポート 21 へ (クライアントが接続を開始)
- FTP のポート 21 からリモート・ポート > 1024 へ (サーバーがクライアントの制御ポートに応答)
- 任意の場所から FTP サーバーのポート > 1024 へ (クライアントがサーバーで指定されたランダム・ポートへのデータ接続を開始)
- FTP サーバーのポート > 1024 からリモート・ポート > 1024 へ (サーバーが ACK (およびデータ) をクライアントのデータ・ポートに送信)

ファイアウォールでポート 8999 を開きたくない場合は、まだユーザーに Host On-Demand をアクセスさせることができます。2 つのオプションがあります。

- すべての構成情報を含む HTML ファイルを作成するには、デプロイメント・ウィザードを使用してください。そうすることにより構成サーバーにアクセスする必要がなくなります。HTML ファイルを作成するときは、デプロイメント・ウィザードの「構成モデル」ページから「HTML ベースのモデル」を選択します。
- 構成サーバーを使用したい場合は、構成サーブレットを使用するためにクライアントを構成することができます。Host On-Demand のオンライン・ヘルプの『構成サーブレットの構成 (Configuring the configuration servlet)』を参照してください。このオプションは、Web アプリケーション・サーバーがサーブレットをサポートしている場合にのみ使用できます。

ファイアウォールによってユーザーの Web ブラウザーから分離された構成サーバーを使用する場合は、ファイアウォールの構成サーバー・ポートを開くか、または Host On-Demand 構成サーブレットを実行する必要があります。構成サーブレットは HTTP または HTTPS などの標準 Web プロトコルを使用してブラウザーが構成サーバーと通信できるようにします。(13 ページの図 4 を参照してください。)

プロキシ・サーバーを介したホスト・システムへの接続

Host On-Demand のクライアントは、プロキシ・サーバーを使用して、ファイアウォールで隔てられたホスト・システムに透過的にアクセスすることができます。次の 2 つのタイプのプロキシ・サーバーがサポートされます。

- 『Socks プロキシ・サーバーを介した接続』で説明されている Socks プロキシ・サーバー。バージョン 4 とバージョン 5 の両方の Socks がサポートされます。
- 38 ページの『HTTP プロキシ・サーバーを介した接続』で説明されている HTTP プロキシ・サーバー。

プロキシ・サーバーを介してホスト・システムに接続するためには、その前に、そのプロキシ・サーバーがどのプロトコルをサポートするのかを判別しなければなりません。Web ブラウザーを介してプロキシ・サーバーの設定値を指定するか、そのセッションで使用するプロキシ・サーバーを明示的に識別するのかを決めてください。プロキシ・サーバーを明示的に識別する場合には、そのプロキシ・サーバーが使用するプロトコル、プロキシ・サーバーの名前とポート番号、その他の情報を指定しなければなりません。

一般に、Socks プロキシ・サーバーが使用可能になっている場合には、それを使用するように Host On-Demand セッションを構成してください。ユーザーのサイトでサポートされるプロキシ・サーバーのタイプが HTTP プロキシ・サーバーだけである場合には、HTTP プロキシ・サーバーを使用するようにセッションを構成してください。

Socks プロキシ・サーバーを介した接続

多くの組織では、コンピューター・リソースをファイアウォールで保護するために、Socks プロキシ・サーバーを使用しています。Socks は、TCP/IP ベースのネットワーク・プロキシ向けのプロトコルです。これを使用すると、Socks プロキシ・サーバーの一方の側にあるアプリケーションが、Socks プロキシ・サーバーの他方の側にあるホストに、直接に接続することなく、完全にアクセスすることができます。プロキシ・サーバーは一般に、ファイアウォールと一緒に使用されます。Socks プロトコルでは、ファイアウォールを超えてホスト・システムに接続しようとするクライアントは、実際には、Socks プロキシ・サーバーと接続することになります。Socks プロキシ・サーバーは、クライアントとホスト・システムとの間の仲介機能としての役割を果たします。このサーバーは通信要求を許可し、クライアントに代わってホストに接続し、2 つのシステム間でデータを中継します。

Host On-Demand では、バージョン 4 とバージョン 5 の両方の Socks プロトコルがサポートされます。

- Socks バージョン 4 は、TCP ベースのアプリケーションのユーザーがファイアウォールを超えてアクセスできるようにするための、メッセージ・フォーマットと規則を指定しています。そして、IP アドレスやソースおよび宛先のポート番号などの TCP ヘッダー情報に基づいて、アクセス制御を行います。
- Socks バージョン 5 (認証されたファイアウォール横断 (authenticated firewall traversal; AFT) とも呼ばれています) は、ネットワーク・プロキシのためのオープンなインターネット標準です。バージョン 5 では、バージョン 4 に対して、認証、ドメイン・ネーム解決のためのサポートの改善、IPv6 アドレス、お

よびその他のフィーチャーが追加されています。これらのフィーチャーは、ファイアウォールの外側にあるクライアントにとって、非常に便利なものです。オプションとして、Host On-Demand クライアントとプロキシ・サーバーとの間の接続を介して、プロキシ・サーバーのための Socks のユーザー ID とパスワードを送信することができます。このユーザー ID とパスワードは暗号化されていません。バージョン 5 について詳しくは、『Socks Protocol バージョン 5』を参照してください。

ほとんどの Web ブラウザーで使用されている Java 仮想マシン (JVM) は Socks バージョン 4 をサポートします。セッションは、Socks バージョン 4 またはバージョン 5 のプロキシ・サーバーにアクセスして、Web ブラウザーのプロキシ・サーバー設定値をバイパスすることができます。プロキシ・サーバーが Socks バージョン 5 をサポートしない場合には、セッションに Socks バージョン 4 接続とネゴシエーションさせることができます。Socks プロキシ・サーバー設定値の詳細については、オンライン・ヘルプの『プロキシ・サーバー (Proxy Server)』を参照してください。

HTTP プロキシ・サーバーを介した接続

HTTP プロキシ・サーバーは、ファイアウォールを介して HTTP 要求を処理します。このプロキシ・サーバーは、専用ローカル・ネットワークとインターネットの間の仲介機能として働きます。HTTP プロキシ・サーバーは、ローカル・ネットワークとインターネットの両方に接続されます。ローカル・ユーザーは、HTTP プロキシ・サーバーの IP アドレスと TCP ポート番号を指定することにより、HTTP 要求が HTTP プロキシ・サーバーを経由するようにブラウザーを構成します。HTTP プロキシ・サーバーはこれらの HTTP 要求を受け入れ、ブラウザーから入力された URL で指定された、実際の Web サーバーにその要求を転送します。

Host On-Demand クライアントについては、HTTP プロキシ・サーバーはホスト・システムへの接続のための転送エージェントとして機能します。HTTP プロキシ・サーバーはホスト・システムへの接続をオープンし、ホスト・システムとクライアントの間でデータをやり取りします。HTTP プロキシ・サーバーは、通常は、HTTP 要求を処理した後で接続をクローズしますが、Host On-Demand は、HTTP 接続メソッドを使用して (このメソッドがプロキシ・サーバーで使用できるようになっている場合)、ホスト・トラフィックのために接続をオープンしたままにします。

セッションに HTTP プロキシ・サーバーを使用させるためには、プロキシ・タイプとして HTTP プロキシを選択し、プロキシ・サーバーの名前とポート番号を指定する必要があります。HTTP プロキシ・サーバー設定値の詳細については、オンライン・ヘルプの『プロキシ・サーバー (Proxy Server)』を参照してください。

ユーザー ID セキュリティー

Web 高速ログオン (Web Express Logon)

ネットワーク・セキュリティ・アプリケーションが適所にあり、構成サーバー・ベース・モデルを使用している場合には、デプロイメント・ウィザードで「Web

Express Logon」を選択すると、追加のユーザー ID およびパスワードを提供しなくてもホストとホスト・ベース・アプリケーションにユーザーがアクセスできます。 信任状マッパー・サーバーの絶対 URL を入力すると、信任状マッパー・サーバーの位置を Host On-Demand に知らせることになります。この信任状マッパー・サーバーはユーザーからの HTTPS 要求を処理し、ルックアップを実行し、ユーザーの信任状を戻します。 その後、信任状はセキュアな自動 Host On-Demand ログインを実行するのに使用されます。

ネイティブ認証

構成サーバー・ベース・モデルを使用する場合には、Host On-Demand ユーザーを、ネイティブで認証されるように構成することができます。このオプションを使用すると、ユーザーは、Host On-Demand がアクティブになっているオペレーティング・システム (AIX または z/OS) にログオンするときと同じパスワードを使用して Host On-Demand にログオンすることができます。ユーザーが Host On-Demand にログオンするときに、そのパスワードは、Host On-Demand 固有のパスワードではなく、オペレーティング・システムのパスワードと突き合わせて妥当性検査されます。これにより、管理者は、パスワードを一元的に管理することができ、ユーザーは、1 つのパスワードを覚えておくだけで済むようになります。

このオプションを使用可能にするための詳細については、オンライン・ヘルプの『ネイティブ認証 (Native Authentication)』を参照してください。

Windows ドメインのログオン

このオプション (デプロイメント・ウィザードの構成サーバー・ベース・モデルで選択可能です) を使用すると、ユーザーは、Windows ドメインにログオンしたときに、その Windows ユーザー名を使用して自動的に Host On-Demand にログオンされます。 Host On-Demand のログオン・ウィンドウは表示されず、Windows のユーザー名が Host On-Demand のユーザー ID として使用されます。(Windows ユーザー名と一致する) Host On-Demand のユーザー ID がまだ存在しない場合、指定された Host On-Demand グループ内にユーザー ID が自動的に作成されるようにすることもできます。

ユーザーが Host On-Demand 構成サーバーにアクセスする方法の選択の詳細については、オンライン・ヘルプの『ログオン・タイプ (Logon Type)』を参照してください。

FIPS 環境

セキュリティー・コンポーネントが連邦情報処理標準 (FIPS) 認証済みのコンポーネント/モジュールを使用することが要求される環境では、以下のことを考慮する必要があります。 Telnet および FTP のセキュアな接続に対しては、Host On-Demand はデフォルトで FIPS 準拠モジュールを使用します。ご使用の環境においてファイル転送やデータ転送のために IBM System i ホストに接続しなければならない場合は、ご使用のシステムが以下の要件を満たしていることを確認してください。

- FIPS 認証済みの Java JRE (IBM 1.6.0 サービス・リリース 5 など) を使用している必要があります。
- デプロイメント・ウィザードの「拡張オプション」ウィンドウで、HTML パラメーターの UseJSSEforSeries を true に設定する必要があります。

- IBM System i ホストからの証明書を、Java JRE に対する Java Secure Socket Extension (JSSE) クライアント・トラスト・ストアに追加する必要があります。構成の詳細については、Java JRE のプロバイダーに問い合わせてください。

IBM System i ホストに対してセキュア接続している場合、ファイル転送機能にアクセスすると、JSSE トラスト・ストアに対するパスとパスワードの入力が求められます。IBM System i ホストに対してデータ転送を行う場合にも、JSSE トラスト・ストアに対するパスとパスワードを入力するための追加フィールドが表示されます。

パスとパスワードを入力するもう 1 つの方法として、Host On-Demand で提供されている「アプレットの実行」を使用できます。そのためには、以下のステップを実行します。

1. ディスプレイ・セッションのメニューから、「アクション (Actions)」>「アプレットの実行」を選択する。
2. クラス名のフィールドに「com.ibm.eNetwork.HOD.util.jsse.JSSESetup」と入力する。
3. 「OK」をクリックする。

JSSE トラスト・ストアを構成しなければならないのは 1 回のみです。これはグローバル設定であり、すべてのセッションに適用されます。入力した値は、ブラウザーが再始動されるまで保持されます。

旧バージョンの Host On-Demand では、HTML パラメーターを使用して FIPS モード認証を有効にすることができました。現行バージョンの Host On-Demand には、各セッションごとに FIPS モードを使用可能または使用不可に設定するメニュー・オプションがあります。デフォルトでは、すべてのセッションで FIPS モードは使用可能に設定されます。

第 5 章 各国語サポートの計画

Host On-Demand は複数の言語で提供されています。セッション・ウィンドウ、構成パネル、ヘルプ・ファイル、および資料が翻訳されています。さらに、ディスプレイ、キーボード、および処理のサポートがアラビア語、ヘブライ語、タイ語、およびヒンディ語で提供されています。このサポートについては「オンライン・ヘルプ」で詳しく説明します。

翻訳版はすべてダウンロードで提供されます。グラフィカル・インストール・プログラムを使用して Host On-Demand を i/OS、OS/400、Windows、AIX、Linux、および Solaris 上にインストールする場合は、インストールする言語を選択することができます。z/OS では、コンソール・モードから言語を選択できます。



各国語サポートはオペレーティング・システムに依存しているため、使用する言語の適切なフォントおよびキーボード・サポートをオペレーティング・システムにインストールしなければなりません。例えば、ホスト・セッション言語として韓国語を使用するものの、韓国語のフォントとキーボード・サポートをインストールしていない場合は、文字を正しく表示することができない場合があります。



DBCS を HTML ファイル名として使用することはできません。

サポートされる言語

以下に、Host On-Demand が翻訳されている言語と、Host On-Demand クライアントの翻訳版をロードするために使用できる言語接尾部をリストします。例えば、IBM 提供の HTML ページには、異なる言語で行われたインストールと異なる言語で定義済みの HTML ファイル (英語の場合は HOD_en.html など) を識別するために、言語拡張機能が備わっています。

言語	言語接尾部
中国語 (簡体字)	zh
中国語 (繁体字)	zh_TW
チェコ語	cs
デンマーク語	da
オランダ語	nl
英語	en
フィンランド語	fi
フランス語	fr
ドイツ語	de
ギリシャ語	el
ハンガリー語	hu

イタリア語	it
日本語	ja
韓国語	ko
ノルウェー語	no
ポーランド語	pl
ブラジル・ポルトガル語	pt
ポルトガル語	pt_PT
ロシア語	ru
スロベニア語	sl
スペイン語	es
スウェーデン語	sv
トルコ語	tr
カタロニア語	Ca

サポートされるホスト・コード・ページ

Host On-Demand は、複数のコード・ページをサポートします。セッションごとに、これらのコード・ページを指定することができます。

3270 および 5250 コード・ページ

3270 および 5250 エミュレーターによってサポートされるコード・ページは、以下のとおりです。これらは、セッション構成ウィンドウで選択することができます。

国または地域	コード・ページ	注記
アラビア語	420	
オーストリア	273	
オーストリア (欧州文字)	1141	
ベラルーシ	1025	
ベラルーシ (欧州文字)	1154	
ベルギー	037	
ベルギー (欧州文字)	1140	
ベルギー (旧コード)	274	
ボスニア・ヘルツェゴビナ	870	
ボスニア・ヘルツェゴビナ (欧州文字)	1153	
ブラジル	037	
ブラジル (欧州文字)	1140	
ブラジル (旧)	275	
ブルガリア	1025	
ブルガリア (欧州文字)	1154	
カナダ	037	
カナダ (欧州文字)	1140	
中国 (拡張中国語 (簡体字))	1388	

クロアチア	870	
クロアチア (欧州文字)	1153	
チェコ共和国	870	
チェコ共和国 (欧州文字)	1153	
デンマーク	277	
デンマーク (欧州文字)	1142	
エストニア	1122	
エストニア (欧州文字)	1157	
フィンランド	278	
フィンランド (欧州文字)	1143	
フランス	297	
フランス (欧州文字)	1147	
マケドニア	1025	
FYR マケドニア (欧州文字)	1154	
ドイツ	273	
ドイツ (欧州文字)	1141	
ギリシャ	875	
ヘブライ語 (新コード)	424	
ヘブライ語 (旧コード)	803	
ヒンディ語	1137	5250 ディスプレイのみ
ハンガリー	870	
ハンガリー (欧州文字)	1153	
アイスランド	871	
アイスランド (欧州文字)	1149	
イタリア	280	
イタリア (欧州文字)	1144	
日本 (カタカナ)	930	
日本 (拡張カタカナ)	930	
日本語 (カタカナ Unicode 拡張、JIS2004)	1390	3270 のみ
日本 (拡張ローマ字)	939	
1399 日本語 (ローマ字 Unicode 拡張、JIS2004)	1399	
カザフスタン (欧州文字)	1166	
韓国 (欧州文字)	1364	3270 のみ
韓国 (拡張)	933	
ラテンアメリカ	284	
ラテンアメリカ (欧州文字)	1145	
ラトビア	1112	
ラトビア (欧州文字)	1156	
リトアニア	1112	
リトアニア (欧州文字)	1156	

マルチリンガル	500	
マルチリンガル ISO (欧州文字)	924	
マルチリンガル (欧州文字)	1148	
オランダ	037	
オランダ (欧州文字)	1140	
ノルウェー	277	
ノルウェー (欧州文字)	1142	
オープン・エディション	1047	
ポーランド	870	
ポーランド (欧州文字)	1153	
ポルトガル	037	
ポルトガル (欧州文字)	1140	
ルーマニア	870	
ルーマニア (欧州文字)	1153	
ロシア	1025	
ロシア (欧州文字)	1154	
セルビア・モンテネグロ (キリル文字)	1025	
セルビア・モンテネグロ (キリル文字; 欧州文字)	1154	
スロバキア	870	
スロバキア (欧州文字)	1153	
スロベニア	870	
スロベニア (欧州文字)	1153	
スペイン	284	
スペイン (欧州文字)	1145	
スウェーデン	278	
スウェーデン (欧州文字)	1143	
台湾 (拡張中国語 (繁体字))	937	
台湾 (中国語 (繁体字) 拡張; 欧州文字)	1371	
タイ語	838	
タイ (欧州文字)	1160	
トルコ	1026	
トルコ (欧州文字)	1155	
ウクライナ	1123	
ウクライナ (欧州文字)	1158	
英国	285	
英国 (欧州文字)	1146	
米国	037	
米国 (欧州文字)	1140	

注:

- プリンター定義テーブル (PDT) を使用した 3270 ホスト・プリントは Latin-1、DBCS、双方向、およびタイ語コード・ページのみをサポートします。その他のコード・ページは、Adobe PDF の印刷で、あるいは PDT を使用しない Windows プラットフォームでサポートされます。
- さらに多くの文字 (中華人民共和国政府によって GB18030 標準に定義されている) を組み込むために、6582 Unicode Extension-A および 1,948 の追加の非 Han 文字 (モンゴル語、ウイグル語、チベット語、およびイ語) が Host On-Demand バージョン 6 用に中国語 (簡体字) コード・ページ 1388 に追加されました。

VT コード・ページ

言語	コード・ページ
アラビア語	ASMO 708 および ASMO 449
英国	1101
DEC ギリシャ語	
DEC ヘブライ語	
DEC 各国間置換文字セット	1100
DEC テクニカル	
オランダ語	1102
フィンランド語	1103
フランス語	1104
カナダ系フランス語	1020
ドイツ語	1011
ヘブライ語 NRCS	
ISO ギリシャ語の補足 (ISO Latin-7)	813
ISO ヘブライ語補足	
ISO Latin 1	819
イタリア語	1012
ノルウェー語/デンマーク語	1105
PC デンマーク語/ノルウェー語	865
PC インターナショナル	437
PC マルチリンガル	850
PC ポルトガル語	860
PRC GBK	936
PC スペイン語	220
スペイン語	1023
スウェーデン語	1106
スイス語	1021
米国	1100

CICS ゲートウェイ・コード・ページ

コード・ページ	文字セット
000	自動検出 (デフォルト)
437	Latin-1
813	ISO ギリシャ語 (8859_7)
819	ISO Latin 1 (8859_1)
850	Latin 1
852	Latin 2
855	キリル文字
856	ヘブライ語
857	Latin 5
864	アラビア語
866	キリル文字
869	ギリシャ語
874	タイ語
912	ISO Latin 2 (8859_2)
915	ISO キリル文字 (8859_5)
920	ISO Latin 5 (8859_9)

日本語 JIS2004 Unicode サポート

既存のホスト・コード・ページ 1390 日本語 (英数カナ拡張 (ユニコード)) および 1399 日本語 (英数小文字拡張 (ユニコード)) を選択することで、JIS2004 サポートを有効にできるようになりました。以下の機能がサポートされます。

- 表示スペース編集
- キー割り当て
- ファイル転送
- 画面印刷
- プリンター・セッション
- GDI
- Adobe PDF
- Host Access Class Library (HACL)

ユニコード形式が原因で組み込まれていない以下の機能は、現在のところ HOD ではサポートされていません。

- マクロ
- プリンター・セッションでのプリンター定義テーブル (PDT) の使用

ユーザー定義文字マッピング

2 バイト文字セット (DBCS) 言語の場合は、ユーザーのセッション (3270、5250、3270 ホスト・プリント) で、デフォルトのマッピングの代わりにカスタマイズされたユーザー定義の文字 (UDC) マッピングを使用できます。ユーザーのセッション用のカスタマイズされたマッピングを保管するために UDC マッピング・エディターを使用して UDC 変換テーブルを作成できます。文字マッピングを変更するために UDC マッピング・エディターを使用する方法については、オンライン・ヘルプの『ユーザー定義文字 (UDC) マッピング・エディターの使い方 (Using the user-defined character (UDC) mapping editor)』を参照してください。

i/OS および OS/400 でのユニコード・サポート

137 ページの『i/OS および OS/400 でのユニコード・サポート』を参照してください。

第 2 部 **Host On-Demand** のインストール、アップグレード、およびアンインストール

第 6 章 Host On-Demand サーバーおよび関連ソフトウェアのインストール

この章では、次の 3 つの Host On-Demand コンポーネントのインストールについて説明します。

- Host On-Demand サーバー。これは、Host On-Demand を使用するために必要なものです。手順については、『Installation Manager を使用した Host On-Demand のインストール』を参照してください。
- Host On-Demand 構成サブリット。これは、Host On-Demand をファイアウォールと一緒に実行している特定のインスタンスでのみ必要です。詳細な説明や手順については、67 ページの『構成サブリットのインストール』を参照してください。
- デプロイメント・ウィザード。これは、Windows 上で実行して、カスタマイズされた Host On-Demand クライアントを生成する、非常に便利なツールです。デプロイメント・ウィザードのインストールは必須ではありませんが、強くお勧めします。手順については、56 ページの『デプロイメント・ウィザード』を参照してください。

Installation Manager を使用した Host On-Demand のインストール

Host On-Demand をインストールするには、IBM Installation Manager が必要です。Host On-Demand をインストールする予定のシステムに、最初に管理者モードで IBM Installation Manager をインストールする必要があります。その後で、Installation Manager を使用して Host On-Demand をインストールできます。

Host On-Demand をインストールするには、IBM Installation Manager バージョン 1.8.3 以上が必要です。

重要なリンク

Installation Manager のインストールについては、『Installation Manager のインストールまたは更新』の手順を参照してください。IBM Installation Manager について詳しくは、IBM Installation Manager Knowledge Center を参照してください。

HOD をインストールする前に

インストールの準備

インストールを行うマシンがすべての前提条件に合致していることを確認します。

Host On-Demand のソフトウェア要件は、「Software Products Compatibility Reports」に記載されています。準備については、以下のリストを確認してください。

- IBM Installation Manager v1.8.3 以上がインストールされていることを確認します。
- 32 ビット・アーキテクチャーで言語が 1 つ場合は、ご使用のマシンにインストール用のディスク・スペース (インストール・スペースおよび一時スペース) が最低でも 1.2GB 必要になります。複数の言語をインストールする場合、この値は言語ごとに 4 MB から 8 MB 増加します。
- マルチプラットフォーム製品リポジトリ (ダウンロードおよび解凍) 用に最小 4.5 GB が必要です。
- ユーザーは管理者の特権を使用してログオンする必要があります。
- サポートされる HTTP サーバーのバージョン (例えば IBM HTTP Server または Apache サーバー) がシステムにインストールされていることを確認します。

旧バージョンの Host On-Demand からのアップグレード

旧バージョンの Host On-Demand (HOD V11.0 など) がある場合、HOD V11 から HOD V13.0 以上への直接のマイグレーション・パスはありません。マイグレーションを行うには、以下の手順に従ってください。

1. 旧 Host On-Demand ディレクトリー (特にプライベート・ディレクトリー) にあるすべてのカスタマイズ済みファイルと、デプロイメント・ウィザードで作成したすべてのクライアント・ページをバックアップする必要があります。これらのファイルは HOD V12.0 以上で再利用できます。
2. 既存の Host On-Demand インストール済み環境をすべてアンインストールします。
3. 初期 Host On-Demand 以降のインストールは、空のパスが使用可能であることを必要とします。そのため、旧バージョンが以前にインストールされていた既存のフォルダーまたはディレクトリーがあれば、名前変更または削除できます。
4. IBM Installation Manager を使用して Host On-Demand をインストールします。インストールの進行中には「キャンセル」をクリックしないことをお勧めします。
5. プライベート・ディレクトリーを Host On-Demand のフォルダーまたはディレクトリーにリストアします。
6. 作成されたすべてのクライアントを Host On-Demand デプロイメント・ウィザードで編集し、HOD サーバーにデプロイします。

V12.0 から HOD V13.0 へのマイグレーションを行うには、以下のようになります。

1. プラットフォームの手順に従って Installation Manager を開始します。
2. 「ファイル」>「設定」を選択します。
3. 左側の「リポジトリ」を選択します。このオプションは、Installation Manager に追加された使用可能なリポジトリを表示します。
4. Host On-Demand がリストされていない場合は、「リポジトリの追加」を選択します。
5. 「参照」をクリックして、抽出した Host On-Demand 更新パッケージのパスの場所にナビゲートし、disk1 フォルダーにある diskTag.inf ファイルを選択します。
6. 「OK」をクリックすると、新しいリポジトリの場所がリストされます。

7. 「接続のテスト」をクリックして、リポジトリ URL が使用可能であることを確認します。
8. Installation Manager の開始ページで「更新」をクリックします。Installation Manager は、使用可能なパッケージを定義済みリポジトリで検索します。
9. 「IBM Host on-Demand」を選択します。
10. 「次へ」ボタンをクリックします。
11. 使用条件を読んで「次へ」ボタンをクリックします。
12. 「更新」ボタンをクリックします。

注: マイグレーションには、Fix Central で入手できる HOD マイグレーション・ユーティリティを使用できます。このユーティリティは、セッション・パラメータを更新したり、カスタム・ファイルと HTML ファイルをすべてパッケージしたり、それらのファイルを HOD V13.0 の要件に従って変更したりできます。

詳しくは、『Migration Utility』を参照してください。

Host On-Demand のインストール

サポートされているすべてのプラットフォームにおいて IBM Installation Manager を使用して Host On-Demand をインストールできます。Installation Manager (IM) を使用すれば、インストールに IM GUI、コマンド・モード、またはコンソール・モードを使用できます。Installation Manager GUI は、多くのオペレーティング・システム (z/OS を除く) でサポートされています。z/OS 上にインストールを行うには、コンソール・モードを使用するか、または BPXBATCH ジョブを実行します。

インストールに Installation Manager GUI を使用

Installation Manager GUI で以下を行います。

1. プラットフォームの手順に従って Installation Manager を開始します。
2. 「ファイル」 > 「設定」を選択します。
3. 左側の「リポジトリ」を選択します。このオプションは、Installation Manager に追加された使用可能なリポジトリを表示します。
4. Host On-Demand がリストされていない場合は、「リポジトリの追加」を選択します。
5. 「参照」をクリックして、抽出した Host On-Demand のパスの場所にナビゲートします。初期インストールには、disk1 フォルダーにある diskTag.inf ファイルを選択します。
6. 「OK」をクリックすると、新しいリポジトリの場所がリストされます。
7. 「接続のテスト」をクリックして、リポジトリ URL が使用可能であることを確認します。
8. Installation Manager の開始ページで「インストール」をクリックします。Installation Manager は、使用可能なパッケージを定義済みリポジトリで検索します。
9. **Host On-Demand** パッケージを選択します。「次へ」をクリックします。

10. ご使用条件を読みます。ご使用条件の条項に同意する場合は、「使用条件の条項に同意します (**I accept the terms of the license agreement**)」をクリックし、「次へ」をクリックして先に進みます。
11. 「新規パッケージ・グループの作成 (**Create a new package group**)」を選択し、「アーキテクチャー (**the Architecture**)」を選択します。
12. オペレーティング・システムが 64 ビットの場合は、「64 ビット」または「32 ビット」を選択し、対応するビット・モードで製品をインストールします。
13. 「次へ」をクリックします。
14. インストールする言語を選択します。デフォルトは英語 です。「次へ」をクリックします。
15. 「**Host On-Demand 13.0**」機能を選択します。「次へ」をクリックします。
16. 「Host On-Demand 13.0」タブですべての情報を確認および指定します。
 - a. 「Host On-Demand 13.0」の下にある「パブリッシュ情報」パネルでパブリッシュ・ディレクトリーを設定し、Web サーバー別名およびサービス・マネージャー・ポート番号 を指定します。「次へ」をクリックします。

このパブリッシュ・ディレクトリーは、クライアントからアクセスできるものでなければなりません。パブリッシュ・ディレクトリーのパスを指定できます。以下の手順を実行してください。

 - 1) ディレクトリーの別名を指定します。デフォルトは *hod* です。
 - 2) Host On-Demand クライアントがサービス・マネージャーと通信するために使用するサービス・マネージャー・ポートを指定します。この通信は、以下のデプロイメント・オプションで必要となります。
 - 11 ページの『第 2 章 配置のための計画』で説明されている、構成サーバー・ベース・モデルおよび結合展開モデル内にあるようなセッション構成情報を保守するための構成サーバーの使用。
 - これらの目的のためには ポート 8999 を指定することをお勧めします。サーバーの資料を参照して、このポートが使用されていないか確認してください。使用中である場合は、インストール中またはインストール後にポートを変更できます。サービス・マネージャー・ポートの変更について詳しくは、オンライン・ヘルプの『サービス・マネージャーの構成ポートの変更 (**Changing the Service Manager's configuration port**)』を参照してください。
 - b. 「Host On-Demand 13.0」にある「Web サーバー」パネルで、自分の要件に適した Web サーバー・オプションを選択します。
 - Web サーバーがユーザーによって手動で構成されている場合は、「**Web サーバーがありません**」を選択します。このオプションは、IPlanet や IBM Domino などの Web サーバー用に推奨されます。ユーザーには、詳細について、Web サーバー管理者に問い合わせるか、Web サーバー資料を参照することをお勧めします。
 - 複数が見つかった場合は、オプション「**検出された Web サーバーのリストから選択**」を選択し、次に Web サーバーをリストから選択します。

- IBM HTTP Server または Apache Web サーバーがインストールされていても検出されない場合は、オプション「特定の **Web** サーバーを手動で選択」を選択します。
- システムにインストールされている Web サーバーのタイプを選択します。
- 「参照」ボタンをクリックし、システムにインストールされている Web サーバーの構成ファイル (httpd.conf) にナビゲートします。代わりに、Web サーバー・インストール・ディレクトリー内の httpd.conf ファイルの完全パスをフィールドに入力することもできます。

- c. インストール・プログラムがご使用のシステム上で IBM WebSphere Application Server を検出した場合は、「Host On-Demand 13」にある「アプリケーション・サーバー」パネルで構成サブレットを構成できます。「アプリケーション・サーバー」タブの次のパネルで、WebSphere Application Server で HOD 構成サブレットを構成するかどうかを尋ねられます。詳しくは、構成サブレットのインストールを参照してください。

構成サブレットを使用する予定のない場合は、チェック・ボックスのチェック・マークを外してください。

構成サブレットを使用する予定の場合は、検出されたサーバーのリストからアプリケーション・サーバーを選択してください。インストール・プログラムは、ユーザーが指定した Web アプリケーション・サーバー上に構成サブレットを自動的にデプロイし、クライアントがサブレットを介してサービス・マネージャーにアクセスするよう構成します。

注:

- WebSphere Application Server がシステム上の同じ IBM Installation Manager プログラムによってインストールされている場合、WebSphere Application Server は検出されます。検出可能なバージョンは WebSphere Application Server V8.0 および V8.5.2 です。
- 管理セキュリティが有効になっているアプリケーション・サーバーは、インストール中のサブレット構成ではサポートされません。

- d. パネルが適切に更新されたら、「次へ」をクリックします。

17. 要約情報を確認し、「インストール」をクリックします。

18. インストールが完了すると、要約ページが表示されます。メッセージを確認します。

- インストールが正常に行われた場合、プログラムはインストールが成功したことを示すメッセージを表示します。プログラムはまた、重要なインストール後の指示も表示することがあります。「終了」をクリックします。
- インストールが正常に行われなかった場合は、「ログ・ファイルの表示」をクリックして、問題をトラブルシューティングします。

19. インストールが正常に完了したことを確認するには、以下の追加アクションを実行します。

- a. Web サーバーを再始動します。
- b. HOD ページがブラウザからアクセス可能であることを確認します。アクセス可能でない場合は、Web サーバー構成を調べて、Host On-Demand パブリッシュ・ディレクトリー内のファイルがアクセス可能であることを確認します。構成について詳しくは、Web サーバーの資料を参照してください。

デプロイメント・ウィザード

デプロイメント・ウィザードは、Windows Host On-Demand サーバー・インストールの一環として自動的にインストールされます。また、Windows Host On-Demand サーバー全体をインストールしたくないお客様は、デプロイメント・ウィザードを別個に使用することもできます。

z/OS および iSeries の場合、デプロイメント・ウィザードのインストール・パッケージは、HOD サーバー上の `<install directory>/HOD/depwiz` ディレクトリーに `DW.zip` という名前で置いてあります。このファイルは Windows ワークステーションにダウンロードして個別のパッケージとしてインストールできます。

旧バージョンのデプロイメント・ウィザードからのアップグレード

旧バージョンのデプロイメント・ウィザード (HOD V11.0 など) がある場合、デプロイメント・ウィザード V11.0 からデプロイメント・ウィザード V12.0 以降のバージョンに直接アップグレードする方法はありません。アップグレードするには、以下の作業を行います。

1. デプロイメント・ウィザード・ディレクトリーからすべてのカスタマイズ済みファイルのバックアップを取得します。いずれの既存ファイルもバージョン 12 デプロイメント・ウィザードで編集できます。
2. 新しいデプロイメント・ウィザードは、空のパスが使用可能であることを必要とします。このため、デプロイメント・ウィザード・インストール済み環境の既存フォルダーを名前変更または削除してください。
3. IBM Installation Manager を使用してデプロイメント・ウィザードをインストールします。
4. カスタマイズ済みファイルをデプロイメント・ウィザードのインストール・フォルダーに再デプロイします。

デプロイメント・ウィザードのインストール

Windows オペレーティング・システムでは、Host On-Demand がインストールされるとデプロイメント・ウィザードが自動的にインストールされます。

デプロイメント・ウィザードをインストールおよび実行するには、以下の作業を行います。

1. Installation Manager を開きます。
2. Host On-Demand リポジトリの場所を Installation Manager に追加します。

- a. Installation Manager の「開始 (Start)」ページで、「ファイル」 > 「設定」をクリックし、次に「リポジトリ」をクリックします。「リポジトリ」ページが開き、使用可能なリポジトリと、それらの場所および接続状況が表示されます。
 - b. 「リポジトリ」ページで、「リポジトリの追加」をクリックします。
 - c. 「リポジトリの追加」ダイアログ・ボックスで、「参照」をクリックします。
 - d. Host On-Demand ディスク 1 の場所にナビゲートし、diskTag.inf ファイルを選択して、「OK」をクリックします。新しいリポジトリの場所がリストされます。
 - e. 「接続のテスト」をクリックして、リポジトリ URL が使用可能であることを確認します。
 - f. 「開始」ページから、「インストール」をクリックします。Installation Manager は、使用可能なパッケージを定義済みリポジトリで検索します。
 - g. 2 番目のディスクについて上記の手順を繰り返します。2 番目のディスクを構成せずに先に進むと、Installation Manager はインストール・プロセス中にこの構成についてユーザーに確認します。
3. Host On-Demand デプロイメント・ウィザード・パッケージを選択します。
 4. その下でバージョン 13.0 も選択されていることを確認します。「次へ」をクリックします。
 5. 「パッケージのインストール」パネルで、「新規パッケージの作成 (Create a new package)」を選択し、パッケージ・グループ名として「**IBM Host On-Demand デプロイメント・ウィザード**」を選択します。
 6. アーキテクチャー選択が 64 ビットに設定されている場合、必要であればこの選択を 32 ビットに変更します。「次へ」をクリックします。
 7. 「パッケージのインストール」パネルで機能「Host On-Demand デプロイメント・ウィザード 13.0 (Host On-Demand Deployment Wizard 13.0)」を選択します。パネルの下部領域にあるディスク情報に、使用可能なディスク・スペースおよび必要なディスク・スペースに関する情報が示されます。「次へ」をクリックします。
 8. 左のタブの見出し「Host On-Demand ウィザード 13.0 (Host On-Demand Wizard 13.0)」の下にある「デプロイメント・ウィザードの Host On-Demand ウィザードの場所 (Deployment Wizard Host On-Demand Wizard Location)」パネルのタブを選択します。Host On-Demand サーバーの場所は、Host On-Demand サーバーへのリンクを示しています。必ず、HOD サーバーへの有効かつ作動しているリンクをフィールドに入力してください。「次へ」をクリックします。
 9. 要約パネルで、選択したパッケージとインストール選択内容を確認します。「インストール」をクリックしてインストールを続行します。

デプロイメント・ウィザード・インストール・イメージを Host On-Demand サーバーからダウンロード

デプロイメント・ウィザード・イメージは、すべての Host On-Demand サーバー・プラットフォームで出荷されているので、サーバーからダウンロードして、Windows マシンにインストールすることができます。

デプロイメント・ウィザードを Host On-Demand サーバーからダウンロードする方法は 2 つあります。1 つは、HODMain_xx.html ページを使用してサーバーにアクセスする方法です (xx は 2 文字の言語接尾部です)。もう 1 つは、サーバーから ftp で直接ダウンロードする方法です。HODMain_xx.html を使用してダウンロードする場合、ダウンロードは Web サーバー経由で行われます。以下に手順を示します。

1. Windows マシンからブラウザを開始して、Host On-Demand サーバー上の HODMain_xx.html ファイルを位置指定します。xx は、ご使用の言語を表す 2 文字の言語接尾部です。
2. 「管理者」タブ 3 をクリックします。
3. 「デプロイメント・ウィザード」リンクをクリックして、デプロイメント・ウィザードのインストール・イメージをご使用の Windows マシンにダウンロードします。
4. Installation Manager を実行してデプロイメント・ウィザードをインストールします。
5. インストールの完了後に、デプロイメント・ウィザードを「スタート」 > 「すべてのプログラム」デスクトップ・メニューから開始できます。

ftp からダウンロードを行うには、以下の手順に従います。

1. Windows マシンから HOD サーバー・マシンに FTP でアクセスします。
2. プロンプトが出されたらサーバーにログインします。
3. bin と入力してモードをバイナリーに設定します。
4. 対象のデプロイメント・ウィザード・ファイルがあるディレクトリーに cd で移動します。この場所は <install-directory>/HOD/depwiz になります。
5. get DW.zip と入力してファイルを取得します。
6. その zip ファイルを Windows マシン上に解凍します。
7. Installation Manager を実行してデプロイメント・ウィザードをインストールします。
8. デプロイメント・ウィザードを「スタート」 > 「すべてのプログラム」デスクトップ・メニューから開始します。

Host Access Toolkit

Host Access Toolkit は、独自の Host On-Demand アプリケーションを作成したいお客様用に個別にインストールされます。

Host Access Toolkit のインストール

以下の基本的な手順を実行して、Host Access Toolkit を Windows システムにインストールします。

1. Installation Manager を開きます。
2. Host On-Demand リポジトリの場所を Installation Manager に追加します。

- a. Installation Manager の「開始 (Start)」ページで、「ファイル」 > 「設定」をクリックし、次に「リポジトリ」をクリックします。「リポジトリ」ページが開き、使用可能なリポジトリと、それらの場所および接続状況が表示されます。
 - b. 「リポジトリ」ページで、「リポジトリの追加」をクリックします。
 - c. 「リポジトリの追加」ダイアログ・ボックスで、「参照」をクリックします。Host On-Demand ディスク 1 の場所にナビゲートし、diskTag.inf ファイルを選択します。続いて、「OK」をクリックします。新しいリポジトリの場所がリストされます。
 - d. 「接続のテスト」をクリックして、リポジトリ URL が使用可能であることを確認します。
 - e. 「開始」ページから、「インストール」をクリックします。Installation Manager により、定義済みのリポジトリが検索され、使用可能なパッケージが検出されます。
 - f. 2 番目のディスクについて上記の手順を繰り返します。2 番目のディスクを構成せずに先に進むと、Installation Manager はインストール・プロセス中にこの構成について確認します。
3. Host Access Toolkit パッケージを選択します。
 4. その下でバージョン 13.0 も選択されていることを確認します。「次へ」をクリックします。
 5. 「パッケージのインストール」パネルで、「新規パッケージの作成 (Create a new package)」を選択し、パッケージ・グループ名として「IBM Host Access Toolkit」を選択します。
 6. アーキテクチャー選択が 64 ビットに設定されている場合、この選択を 32 ビットに変更します (こちらが推奨されています)。「次へ」をクリックします。
 7. インストールする言語を選択します。デフォルトは英語 です。「次へ」をクリックします。
 8. 「パッケージのインストール」パネルで機能「Host Access Toolkit 13.0」を選択します。パネルの下部領域にあるディスク情報に、使用可能なディスク・スペースおよび必要なディスク・スペースに関する情報が示されます。「次へ」をクリックします。
 9. 左のタブの見出し「Host Access Toolkit 13.0」の下にある「Host Access Toolkit 13.0」パネルのタブを選択します。
 10. 要約パネルで、選択したパッケージとインストール選択内容を確認します。「インストール」をクリックしてインストールを続行します。

コンソール・モードでのインストール

この章では、グラフィカル・ユーザー・インターフェースがサポートされていないプラットフォームに Installation Manager コンソール・モードを使用して Host On-Demand をインストールするための手順について説明します。

注: IBM iSeries をインストールする場合は、61 ページの『HOD を iSeries にインストールする前に』を読むことをお勧めします。

コンソール・モードでのインストールについて

グラフィカル・ユーザー・インターフェース (GUI) がサポートされていない Linux、UNIX、および z/OS の各システムでは、管理者は Installation Manager のコンソール・ベースのインターフェースを使用して Host On-Demand をインストールすることができます。

IBM Installation Manager のコンソール・モードを使用して、インストール・パッケージで以下の作業を行うことができます。

- インストール
- アップグレード
- 修正
- ロールバック
- アンインストール

Installation Manager コンソール・モードを開始するには、Installation Manager ツール・ディレクトリーで使用可能な *imcl* ユーティリティーを使用してください。

ここで示すインストール手順は、コンソール モードを使用した標準的なインストール シナリオです。インストール・セッション中、インストールするパッケージに固有のコンソール・モード・プロンプトが表示されます。コンソール画面に表示されたオプションに従ってインストールを進めることができます。

Installation Manager のコンソール モード インターフェースでは、次のような規則が使用されます。

- [X] は、選択したオプションを示します。
- [] は、選択されていないオプションを示します。
- デフォルトのコマンドは、大括弧 [] で囲まれています。
- [N] は、デフォルトのコマンドが N: [次へ] であることを示します。

注: Installation Manager およびコンソール モードについて詳しくは、インストール済みの Installation Manager バージョンに対応する Installation Manager Knowledge Center で参照できます。IBM Installation Manager Knowledge Center を参照してください。

Installation Manager は、Installation Manager をインストールまたは更新するための Installation Manager 資料に記載された情報を使用してインストールできます。

Host On-Demand をインストールするには、Installation Manager が管理者モードでインストールされていなければなりません。Installation Manager のダウンロードについて詳しくは、IBM Installation Manager および Packaging Utility のシステム要件を参照してください。Host On-Demand をインストールするための最小レベルは 1.8.3 です。

Installation Manager の使用法について詳しくは、IBM Installation Manager Knowledge Center を参照してください。

HOD を iSeries にインストールする前に

Host On-Demand の IBM iSeries プラットフォームへのインストールは、Installation Manager のコンソール・モードでサポートされています。GUI モードのインストールは、IBM iSeries では使用できません。

Host On-Demand を IBM iSeries にインストールする前のその他の注意事項は、以下のとおりです。

- IBM Installation Manager V1.8.3 以上がインストールされていることを確認します。管理者モードでインストールされている必要があります。詳細については、IBM Installation Manager の資料に従うことをお勧めします。Installation Manager V1.8.3 に関する情報は、『Installation Manager を IBM i にインストール』に記載されています。
- インストールは、管理者権限または root 権限を持つユーザーが実行します。
- HOD V13.0 では、Installation Manager を使用してリモートから IBM i 上にインストールを行うことはできません。

インストールを始めるには、以下の作業を行う必要があります。

1. Host On-Demand ESD zip ファイルを IBM i に、FTP (File Transfer Protocol) から、または何らかの通常の方法を使用してコピーし、zip ファイルを解凍します。
2. Installation Manager を開き、Host On-Demand ディスクにある diskTag.inf ファイルへの完全パスを指定することによってリポジトリを構成します。
3. コンソール・モード・インストールで提供される残りの手順を進めます。

インストール手順

HOD をコンソール・モードでインストールするには、以下の作業を行います。

1. IBM Installation Manager をコンソール・モードで開始します。管理者権限でコマンド・プロンプトを開き、IBM Installation Manager インストール・ディレクトリー内の tools フォルダに変更します。

2. tools ディレクトリーでコマンド

```
imcl -c
```

を実行します。

このコマンドは、オペレーティング・システムによって以下のようになります。

- AIX® または Linux:

```
/opt/IBM/InstallationManager/eclipse/tools/imcl -c
```

- IBM i:

```
/QIBM/ProdData/InstallationManager/eclipse/tools/imcl -c
```

- Windows:

```
%Program Files%\IBM\Installation Manager\eclipse\tools\imcl.exe -c
```

- z/OS:

```
/InstallationManager/bin/eclipse/tools/imcl -c
```

コンソール・モードでの Installation Manager の開始について詳しくは、『コンソール・モードの開始』を参照してください。

3. コンソール・ウィンドウで、IBM Host On-Demand リポジトリを指定します。
 - a. 設定を編集するには、**P** と入力し、**Enter** を押します。
 - b. リポジトリを指定するには、**1** と入力し、**Enter** を押します。
 - c. リポジトリを追加するには、**D** と入力し、**Enter** を押します。
 - d. IBM Host On-Demand 13.0 のリポジトリ・パスを入力します。例えば、`<path>¥HOD¥disk1¥diskTag.inf` と入力します。
 - e. リポジトリ情報を保存するには、**A** と入力し、**Enter** を押します。
 - f. メインメニューに戻るには、**R** と入力し、**Enter** を押します。
4. メインメニューからインストールするには、**1** を選択します。資格情報を必要とするリポジトリがある場合は、ID とパスワードの入力を要求されます。この要求時に、資格情報を保存することもできます。Installation Manager Knowledge Center のコンソール モードでの資格情報の保存を参照してください。
5. インストールするパッケージを選択するパネルで、該当する番号を入力して Host On-Demand 13.0 パッケージを選択します。
6. 後続のパネルで、該当する番号を入力してインストール対象にバージョン 13.0 を選択し、**Enter** を押します。
7. **N** を入力して先に進みます。
8. 該当する番号を入力してライセンス情報を表示し、ご使用条件に同意します。ご使用条件に同意するには、**A** と入力し、**Enter** をクリックします。 **N** を入力し、**Enter** を押して先に進みます。
9. 「Installation Manager 共有リソース・ディレクトリー (Installation Manager Shared Resources Directory)」を選択します。詳しくは、『パッケージ・グループおよび共有リソース・ディレクトリーの概要』を参照してください。ディレクトリーを変更するには、**M** と入力し、**Enter** を押します。正しいパスを入力し、次に **N** を入力して先に進みます。
10. 「場所」パネルで IBM Host On-Demand 13.0 インストール・ディレクトリーの場所を指定できます。インストール・ディレクトリーの場所を変更するには、**M** と入力します。正しいパスを入力し、**N** を入力して先に進みます。
11. 64 ビットのオペレーティング・システム上にインストールする際、パッケージのアーキテクチャーが表示されます。新しいパッケージ・グループの場合、**T** (ビット・アーキテクチャーに変更) を入力して、ビット・モードを変更できます。例えば、選択したアーキテクチャーが 64 ビットと表示され、32 ビット・アーキテクチャーに変更するための **T** が表示された場合は、**T** を入力して 32 ビット・アーキテクチャーに変更します。
12. デフォルト値を受け入れるか、別の値を入力して続行するには、**N** を入力して先に進みます。
13. 言語パネルで、言語の左側にある番号を入力して、言語のリストでインストールする言語を追加または削除します。一度に 1 つの言語のみを選択できます。または、**S** を入力してすべての言語を選択できます。デフォルトでは英語が選

択されています。英語は必須です。言語の選択は、パッケージ グループにインストールされるすべてのパッケージに適用されます。N を入力して先に進みます。

14. 次のパネルには、Host On-Demand 13 のインストールで必要となる構成詳細について「構成」メニューが表示されます。

通常、Host On-Demand 12 構成メニューには以下の項目があります。

- パブリッシュ情報
- Web サーバー
- a. 「パブリッシュ情報」項目の左側にある該当する番号を入力して、設定を確認します。「パブリッシュ情報」パネルには、以下の情報が表示されます。
 - 「パブリッシュ宛先ディレクトリー」は、ユーザーが Web からアクセスする Host On-Demand ファイルがインストールされている場所です。デフォルト値はパネルに示されています。必要であれば、1 を入力して場所を変更します。
 - 「Host On-Demand パブリッシュ別名」は、Host On-Demand パブリッシュ・ディレクトリーの Web サーバー別名設定です。必要であれば、A を入力して場所を変更します。
 - 「サービス・マネージャー・ポート」は、Host On-Demand サービス・マネージャーが listen するポート番号です。Host On-Demand クライアントがサービス・マネージャーと通信するために使用するサービス・マネージャー・ポートを指定します。この通信は、以下のデプロイメント・オプションで必要となります。
 - (11 ページの『第 2 章 配置のための計画』で説明されている、構成サーバー・ベース・モデルおよび結合展開モデル内にあるような) セッション構成情報を保守するための構成サーバーの使用。

「ポート 8999」は、Host On-Demand のデフォルト・ポートです。このポートが使用されているかどうかをシステム管理者に確認してください。使用中である場合は、インストール中またはインストール後にポートを変更できます。サービス・マネージャー・ポートの変更について詳しくは、オンライン・ヘルプの『サービス・マネージャーの構成ポートの変更 (Changing the Service Manager's configuration port)』を参照してください。

それぞれの設定を変更するには、これらのオプションに関連付けられている番号を入力します。ナビゲートするには、画面上の残りのオプションを参照してください。

- b. Web サーバーの左側にある該当する番号を入力して、設定を確認します。
 - 1) Web サーバー・パネルに以下のオプションが表示されます。お客様の要件に合った Web サーバー・オプションを選択する必要があります。
 - **Web** サーバーがありません: Web サーバーを手動で構成する場合、または Web サーバーがない場合は、これを選択します。このオプションは、IPlanet や IBM Domino などの Web サーバー用

に推奨されます。詳しくは、Web サーバー管理者に問い合わせるか、Web サーバーの資料を参照してください。

注: z/OS インストールには「Web サーバーがありません」を選択してください。これは、Web サーバーを検出できず、手動で構成する必要があるためです。

- 検出された **Web** サーバーのリストから選択: 検出された Web サーバーを選択するには、このオプションを選択し、次に検出された Web サーバーを選択します。
- 特定の **Web** サーバーを手動で選択: IBM HTTP Server 8.5 または Apache Web サーバー 2.2 がインストールされていても検出されない場合に、このオプションを選択します。Web サーバー・インストール・ディレクトリー内の httpd.conf ファイルへの完全パスを入力する必要があります。

2) それぞれの設定を変更するには、必須オプションに関連付けられている番号を入力します。ナビゲートするには、画面上のオプションを参照してください。N を入力して先に進みます。

- c. インストール・プログラムがシステムで IBM WebSphere Application Server を検出した場合、「アプリケーション・サーバー」タブを使用してアクセスする次のパネルで、Host On-Demand 構成サブルーットを WebSphere Application Server で構成するかどうかが尋ねられます。ユーザーがファイアウォール経由で Host On-Demand を実行していれば、クライアントが Host On-Demand サービス・マネージャーと通信するために追加のポートを開く必要はなくなります。詳しくは、67 ページの『構成サブルーットのインストール』を参照してください。
- 質問の左側に表示される番号または英字を入力すると、IBM Installation Manager はアプリケーション・サーバーのバージョンのリスト、それらのプロファイル、および検出されたサーバーを表示し、それらの中から選択するように促すプロンプトがユーザーに出されます。インストール・プログラムは、指定した Web アプリケーション・サーバー上に構成サブルーットを自動的にデプロイし、そのサブルーットを介してサービス・マネージャーにアクセスするようにクライアントを構成します。
 - サブルーットの構成を選択せずに先に進む場合、インストールでは構成サブルーットは構成されません。クライアントは、ポート 8999 (または指定した代替ポート) でサービス・マネージャーに直接アクセスできます。

注:

- WebSphere Application Server がシステム上の同じ IBM Installation Manager プログラムによってインストールされている場合、WebSphere Application Server は検出されます。検出可能なバージョンは WebSphere Application Server V8.0 および Websphere Application Server V8.5 です。
- 管理セキュリティが有効になっているサーバーは、インストール中のサブルーット構成ではサポートされません。

15. 次のパネルは要約パネルです。インストールを続ける前に、選択内容を確認してください。

16. 応答ファイルを生成するには、G: (インストール応答ファイルを生成) を入力します。
17. 応答ファイルの名前を入力します。ファイル拡張子には `.xml` を使用します。応答ファイルは XML ファイルです。
18. 応答ファイル名をファイルを別の場所に保存する際には、ディレクトリー・ロケーションを含めます。
19. `I` を入力してインストールを開始します。
20. インストールが完了したら、`F:` を入力して終了します。
21. `X` を入力して `Installation Manager` を終了します。

コンソール・モードでのデプロイメント・ウィザードのインストール

デプロイメント・ウィザードは、Windows Host On-Demand サーバー・インストールの一環として自動的にインストールされます。また、Windows Host On-Demand サーバー全体をインストールしたくないお客様は、デプロイメント・ウィザードを別個に使用することもできます。ユーザーは、インストール中にデプロイメント・ウィザード・オプションのみを選択できます。

詳しくは、59 ページの『コンソール・モードでのインストール』を参照してください。

コンソール・モードでの **Host Access Toolkit** のインストール

Host Access Toolkit は、Windows Host On-Demand サーバー・インストールの一環として自動的にインストールされます。また、Windows Host On-Demand サーバー全体をインストールしたくないお客様は、デプロイメント・ウィザードを別個に使用することもできます。ユーザーは、インストール中に Host Access Toolkit オプションのみを選択できます。

詳しくは、59 ページの『コンソール・モードでのインストール』を参照してください。

サイレント・モードでのインストール

サイレント・モードでの Host On-Demand のインストールでは、インストールにスクリプトを使用できます。応答ファイルを使用して `Installation Manager` を開始する前に、まず応答ファイルを作成する必要があります。

`Installation Manager` バージョン V1.8.3 を使用したパッケージのサイレント・インストールについては、`Installation Manager` インフォメーション・センターの以下のトピックを参照してください。

- サイレント・インストール・ロードマップ
- サイレント モードの `Installation Manager` コマンド行引数

インストール手順

このセクションでは、サイレント・モードでの HOD のインストール手順について説明します。

HOD をサイレント・モードでインストールするには、以下の作業を行います。

1. 応答ファイルを作成するには、GUI が使用可能なマシン上で、ウィザード・モードの IBM Installation Manager を使用して、`-record` オプションによって応答ファイルを記録します。詳しくは、『Installation Manager による応答ファイルの記録』を参照してください。例えば、Windows では、応答ファイルの記録は次のように表示されます。

```
C:\Program Files (x86)\IBM\Installation Manager\eclipse>IBMIM.exe -record  
e:\recordResponse.xml
```

2. 必要であれば、生成された XML ファイルを開いて、設定を表示および編集してください。ファイルについて詳しくは、『応答ファイル・コマンド』を参照してください。
3. 生成された応答ファイルを使用してサイレント・インストールを実行するには、IBM Installation Manager で提供されている `imcl` コマンド行ユーティリティを使用します。他のオペレーティング・システムでの例は以下のとおりです。

- Windows:

```
imcl.exe input response_file -log log_file
```

- Linux、UNIX、IBM i、IBM z/OS[®]、および OS X

```
./imcl input response_file -log log_file
```

詳しくは、『応答ファイルを使用したパッケージのサイレント・インストール』を参照してください。

注:

- Host On-Demand インストール用に応答ファイルを記録する際には、`-skipInstall` パラメーターの使用は避けることをお勧めします。
- サイレント・インストール中に Host On-Demand によって Web サーバー、アプリケーション・サーバー、またはその両方が構成されている場合は、ユーザー設定およびソフトウェア・パラメーターが適切に応答ファイルに記録されるように、類似のソフトウェアがセットアップされた環境で応答ファイルを記録する必要があります。例えば、HTTP Server V8.5 を構成中である場合は、より良い結果になるように以下のパラメーターを設定することをお勧めします。

- HTTP サーバーのバージョン
- HTTP サーバーのインストール・ロケーション・パス
- HTTP サーバー内の `httpd.conf` ファイル・ロケーション・パス

同様に、Websphere Application Server がある環境でのサイレント・インストールでは、類似の Websphere Application Server セットアップが使用可能なシステムで応答ファイルを記録してください。

Websphere Application Server がインストールされていない環境で応答ファイルが記録された場合、その応答ファイルは Websphere Application Server がインストールされていない環境で使用することをお勧めします。

- 異なるデプロイメント・シナリオには、それぞれ別個に応答ファイルを保持することが役立ち、これをお勧めします。

- Host On-Demand のインストール先と同じオペレーティング・システム・プラットフォームで応答ファイルを記録することが必要です。例えば、Linux でのサイレント・インストールの場合は、Linux で応答ファイルを記録します。異なるオペレーティング・システムには、それぞれ別個に応答ファイルを保持することが役立ちます。
- コンソール・モードまたは GUI モードのインストール (該当する場合) の前提条件は、サイレント・インストール・モードでも同様に関連します。前提条件には以下のようなものがあります。
 - ログイン・ユーザーには管理者権限が必要です。
 - Installation Manager が管理者モードでインストールされていることが必要です。
 - Host On-Demand をインストールするには、Installation Manager V1.8.3 以上がインストールされている必要があります。管理セキュリティが WebSphere Application Server で有効になっている場合、Host On-Demand 構成サブルーットの構成はインストール中にサポートされません。手動で構成する必要があります。

構成サブルーットのインストール

Host On-Demand のインストールでは、IBM Application Server 用に構成サブルーットを i/OS、OS/400、Windows、AIX、Linux、および Solaris にインストールして構成することを選択できます。



すべての Web サーバーおよびサブルーット・エンジンは、構成が異なります。ご使用のオペレーティング・システムにおける構成サブルーットの詳細については、Web サーバーおよびサブルーット・エンジンの資料を参照してください。

構成サブルーットのインストールは、ご使用の Host On-Demand デプロイメントで次の記述がいずれも真の場合にのみ必要となります。

- Host On-Demand の構成を計画していて、クライアントとサービス・マネージャーとの通信が必要である (ライセンス使用カウントが使用可能であるか、リダイレクターを使用する場合の、構成サーバー・ベース・モデルおよび結合デプロイメント・モデルに該当する)。
- ファイアウォールがセッション構成情報の保守を計画しているサーバーを保護し、サービス・マネージャーへのアクセスをクライアント以外にさせないためファイアウォール内のポートを開かない。

デフォルトでは、Host On-Demand クライアントは、ポート 8999 を使用してサービス・マネージャーから構成情報にアクセスします。ファイアウォールの外部にクライアントがある場合は、ファイアウォールの管理者が内部および外部の両方でポート 8999 を開く必要があります。しかし、クライアントが構成サブルーットを使用して構成情報にアクセスするようにカスタマイズすることによって、このポートを開くことを回避できます。

サブルーットを WebSphere Application Server にデプロイ

Windows、AIX、Linux、および Solaris 上で Host On-Demand をインストールする際、インストール・ユーティリティーは WebSphere Application Server のイン

スタンスをシステムで検索します。インスタンスが検出された場合は、インストール・ユーティリティーは、WebSphere Application Server バージョン 5.1、6.0、6.1、および 7.0 上に、構成サーブレットを自動的にインストールして構成することができます。

インストール・プログラムを提供する System z などのプラットフォームでは、構成サーブレットを手動でインストールする必要があります。エンタープライズ・アプリケーションのインストールに関するステップについては、WebSphere Application Server 資料を調べてください。 <http://www.ibm.com/software/webservers/> にアクセスし、「WebSphere Application Server」サポート・ページにアクセスすることもできます。このページには、ご使用のバージョンの資料へのリンクがあります。

Host On-Demand 構成サーブレット EAR ファイル `cfgservlet.ear` は、Host On-Demand インストールの `lib` ディレクトリーにあります。



WebSphere Application Server 5 の場合: デプロイメント設定を管理コンソールに保管した後、Host On-Demand 構成サーブレットを「エンタープライズ・アプリケーション (Enterprise Applications)」ウィンドウで開始する必要があります。その後「環境 (Environment)」ウィンドウに移動して、「Web サーバー・プラグインの更新 (Update Web Server Plug-in)」を選択します。

構成サーブレットをインストール後、サービス・マネージャーに直接アクセスするのではなく、構成サーブレットを使用するようにクライアントを構成することができます。デプロイメント・ウィザードを使用して、カスタマイズした HTML クライアント・ページを作成することができます。入力に応じてウィザードが HTML にアプレットのパラメーターを設定するため、構文や有効なパラメーター値についての知識は必要ありません。デプロイメント・ウィザードを使用して、クライアント HTML 内の `ConfigServerURL` パラメーターを `HODConfig/HODConfig/hod` に設定するよう IBM はお勧めします。

構成サーブレットのパラメーター、構成、および例の詳細については、オンライン・ヘルプの『構成サーブレットの構成 (Configuring the configuration servlet)』を参照してください。

第 7 章 Host On-Demand サーバーのアンインストール

Host On-Demand バージョン 13 は、Installation Manager GUI を使用してアンインストールできます。以下のアンインストール手順に従ってください。

1. Host On-Demand 関連アプリケーション (デプロイメント・ウィザードや IBM Host On-Demand サービス・マネージャーなど) をすべて停止します。
2. Installation Manager を開始します。「アンインストール」をクリックします。
3. 「IBM® Host On-Demand」と該当するバージョンを選択し、「次へ」をクリックします。
4. 要約情報を確認します。「アンインストール」をクリックします。
 - 正常にアンインストールされた場合、プログラムは成功を示すメッセージを表示します。
 - アンインストールが正常に行われなかった場合は、「ログの表示 (View log)」をクリックして、問題をトラブルシューティングします。
5. 「終了」をクリックします。
6. 「ファイル」 > 「終了」をクリックして、Installation Manager を閉じます。

Installation Manager コンソール・モードを使用した Host On-Demand のアンインストール

コンソール・モードを使用してパッケージをアンインストールすることができます。アンインストールするには、ユーザーは管理者でなければならない、管理者特権を使用してログインする必要があります。

Installation Manager コンソール・モードで HOD をアンインストールするには、以下の作業を行います。

1. Host On-Demand インストール済み環境に関連するプログラムをすべて閉じます。例えば、デプロイメント・ウィザードや IBM Host On-Demand サービス・マネージャーなどです。
2. コマンド
:
: imcl -c

を入力し、**Enter** を押します。
3. 5 を入力し、アンインストールを続行します。
4. Host On-Demand 13.0 パッケージ・グループの左側に表示されている番号を入力します。**Enter** を押します。
5. アンインストールする Host On-Demand 13.0 パッケージ・グループの詳細を確認します。N (Next) を入力するか、**Enter** を押します。N がデフォルトの選択です。
6. Host On-Demand 13.0 パッケージの左側に表示されている番号を入力することにより、Host On-Demand パッケージを選択します。**Enter** を押します。N (Next) を入力します。

7. アンインストールするパッケージを確認します。 *U (Uninstall)* を入力して、**Enter** を押します。このパネルには、応答ファイルを作成するオプションもあります。 *G* を入力し、**Enter** を押して、応答ファイルの作成に進みます。これによりアンインストールが開始されます。
8. 次のプロンプトが出されたら、*F* を押して終了 します。

第 3 部 Host On-Demand の構成

第 8 章 Host On-Demand エミュレーター・クライアントの構成

Host On-Demand をインストールした後、ユーザー用に HTML ファイルを作成して Host On-Demand セッションを構成する必要があります。



Host On-Demand では、ダウンロード・クライアント・コンポーネントおよび Java 自動検出コンポーネントであらかじめ構成されている、すぐに使用可能な 3270、5250、VT、および FTP のエミュレーター・セッションのサンプル HTML ファイルが提供されています。これらのセッションは、HTML ベースの構成モデルを使用し、Host On-Demand を迅速に稼働させてホスト・システムにアクセスできるようにするために提供されています。これらのエミュレーター・セッションを使用するには、以下のステップを実行します。

1. `your_publish_directory\samples\html` ディレクトリーで、`hodclients.zip` ファイルを探す。ここで、`your_publish_directory` は、Host On-Demand パブリッシュ・ディレクトリーの名前です。
2. デプロイメント・ウィザードによって作成された `hodclients.zip` ファイルが、ファイルを `unzip` したいディレクトリー (Host On-Demand パブリッシュ・ディレクトリーまたは特殊パブリッシュ・ディレクトリーのいずれか) にあることを確認する。ない場合は、`.zip` ファイルをそのディレクトリーにコピーします。
3. DWunzip ツールを使用して、`hodclients.zip` の内容をパブリッシュ・ディレクトリーに `unzip` する。このツールの使用方法についての詳細は、『DWunzip の使用 (Using DWunzip)』を参照してください。
4. ブラウザーを使用して、Web サーバー上の `hodclients.html` (<http://host/alias/hodclients.html> など) にアクセスする。
5. 適切なセッション・アイコンを右マウス・ボタンでクリックし、「プロパティー」を選択して、セッションのプロパティーを開く。使用しているホスト・システムの正しい宛先アドレス、ポート、および他の接続プロパティーを入力します。「OK」をクリックします。
6. セッション・アイコンをダブルクリックしてセッションを開始する。

デプロイメント・ウィザードを使用して、HTML ファイルをカスタマイズできます。詳しくは、75 ページの『デプロイメント・ウィザードの使用』を参照してください。

Host On-Demand HTML ファイルの作成

Host On-Demand 用の HTML ファイルを作成してセットアップするには、デプロイメント・ウィザードを使用するのが最適な方法です。デプロイメント・ウィザードにより、使用環境に合わせたすべての Host On-Demand 機能を含むカスタム HTML ファイルを容易に作成することができます。以下に、デプロイメント・ウィザードを使用して構成できる多くの機能の一部をリストします。

- 構成モデル。構成モデルは、セッションの定義場所およびユーザー設定の保管場所に関して従うハイレベルのアプローチを定義します。構成モデルについての詳細は、11 ページの『第 2 章 配置のための計画』を参照してください。

- **プリロード。**Host On-Demand はアプレットまたはアプリケーションとして実行され、ユーザーのマシンにコードをダウンロードする必要があります。デフォルトでは、Host On-Demand クライアントはすべてのコンポーネントをダウンロードしますが、不要なコンポーネントを除去することにより、ダウンロード・サイズを削減できます。
- **キャッシュ・クライアント、Web Start クライアント、またはダウンロード・クライアント。**キャッシュ・クライアントはユーザーが HTML ファイルに最初にアクセスするときにコードを保存し、ユーザーのマシンに格納します。Web Start クライアントでは、キャッシュ・クライアントのようなクライアント・コードがキャッシュされますが、さらに、ブラウザを使用せずに Host On-Demand を実行できます。ダウンロード・クライアントは、ユーザーが HTML ファイルにアクセスするごとに必要なアプレット・ファイルをダウンロードします。
- **Web ページの外観 (カスタム HTML テンプレート)。**デプロイメント・ウィザードが HTML ファイルの生成に使用するテンプレートのセットアップを容易に行うことができます。この機能により、ユーザー自身の背景やバナーなどを容易に追加することができます。
- **キャッシュ・クライアント/Web Start オプション。**キャッシュ・クライアントまたは Web Start クライアントを実行する場合は、新しいバージョンのクライアント・コードが使用可能になった際に、古いコードをアップグレードする必要があります。アップグレードを制御するために、数多くのデプロイメント・ウィザード・オプションを使用できます。
- **Host On-Demand インストールのロケーション (コード・ベース)。**通常、デプロイメント・ウィザードのファイルは、Host On-Demand サーバーのパブリッシュ・ディレクトリーに配置されます。ただし、これらのファイルを Host On-Demand サーバーとは独立した場所に配置すると、異なるセキュリティ管理が可能になり、Host On-Demand サーバーのアップグレードが容易になるなど、役立つ場合があります。
- **WebSphere Portal。**WebSphere Portal は、ポートレットと呼ばれるコンテンツ拡張機能を Web サイトに組み込むためのフレームワークを提供します。ポートレットは、各種のソースからコンテンツを編成し、それを単一の HTML ファイルとしてブラウザ・ウィンドウに表示するアプリケーションです。Host On-Demand セッションを立ち上げるための HTML ファイルはポートレットとして配置することができ、ユーザーがポータル・インターフェースを通して Host On-Demand にアクセスできるようになっています。
- **Windows ドメインのログオン。**ユーザーが Windows ドメインにログオンすると、このオプションによりユーザーは、その Windows ユーザー名を使用して自動的に Host On-Demand にログオンします。このオプションは、デプロイメント・ウィザードで構成サーバー・ベース・モデルを使用する場合にのみ有効です。
- **Session Manager API。**Host On-Demand Session Manager には、ホスト・セッションの管理やホスト・セッションとテキスト・ベースで対話するための JavaScript API が提供されています。これらの API は、JavaScript を使用して Web ページにホスト・セッションを組み込むためのものであり、デプロイメント・ウィザードで使用することができます。



Web Start クライアントを使用するには、デプロイメント・ウィザードを使用する必要があります。このクライアント・タイプ用の定義済みファイルは用意されていません。

Host On-Demand セッションの構成

HTML ファイルのセットアップの他に、ユーザー用にセッションを定義する必要があります。HTML ベース・モデルを使用する場合は、デプロイメント・ウィザードで HTML ファイルを作成すると同時に、セッションも構成します。それ以外の場合、構成サーバー・ベース・モデルまたは結合モデルを使用、あるいはいずれかの事前定義クライアントを使用する場合は、いずれかの管理クライアントを使用して構成サーバーでグループ、ユーザー、およびセッションを作成する必要があります。

セッションを構成する際は、デプロイメント・ウィザードまたはいずれかの管理クライアントを使用するかどうかに関係なく、すべてのオプションを使用することができます。

- セッション・プロパティ。接続情報、セキュリティなど、すべてのセッション・プロパティが構成可能です。それぞれのフィールドは、ユーザーが更新できないようにロックすることができます。
- ランタイム・オプション。セッションを構成する場合、セッションを起動して、セッションのサイズと配置、色、ツールバーのカスタマイズ、およびマクロなどの機能を構成することができます。デプロイメント・ウィザードおよびフル管理クライアントで、ランタイム・オプションを構成することができます。
- ユーザー機能の使用不可化。ユーザーが通常 Host On-Demand セッションの一部として利用する機能 (ブックマーク、マクロの作成や実行など) は、ほとんどすべてを使用不可にすることができます。

デプロイメント・ウィザードの使用

デプロイメント・ウィザードは、Windows および Linux プラットフォームで実行されます。デプロイメント・ウィザードを開始するには、次のいずれかの方法を選択します。

- Windows の Host On-Demand サーバーの一部としてデプロイメント・ウィザードを自動的にインストールした場合は、「スタート」 > 「プログラム」 > 「**IBM Host On-Demand**」 > 「管理」 > 「デプロイメント・ウィザード」と選択します。

「デプロイメント・ウィザードによるこそ」ウィンドウが表示されます。

デプロイメント・ウィザードをガイドとして構成の選択を行い、各機能を説明する総合的なヘルプを利用することができます。機能の選択が完了すると、デプロイメント・ウィザードにより HTML およびサポート・ファイルが作成されます。これらのファイルは、Web サーバーが識別する Host On-Demand サーバー上のディレクトリに配置する必要があります。通常、このディレクトリは、Host On-Demand サーバーのパブリッシュ・ディレクトリです。

Host On-Demand サーバーへのデプロイメント・ウィザード出力の配布

Host On-Demand サーバーが Windows または IBM System i プラットフォーム上にある場合、デプロイメント・ウィザードの HTML ファイルおよび構成ファイルを Host On-Demand サーバーのパブリッシュ・ディレクトリーに直接に書き込むことができます。デプロイメント・ウィザードの最後の画面では、生成ファイルの書き込み先を選択できます。デプロイメント・ウィザードが実行されているマシンよりアクセス可能なローカルまたはネットワーク・ドライブを選択することができます。この場合、デプロイメント・ウィザードがファイルを Host On-Demand サーバー上のパブリッシュ・ディレクトリーに出力するよう指定して、出力形式に *HTML* を選択することができます。セッションが定義済みであれば、HTML ページはユーザーからアクセス可能な状態になります。

それ以外のプラットフォームで、デプロイメント・ウィザードが Host On-Demand サーバーに直接に書き込むことができなければ、デプロイメント・ウィザードで出力形式として *zip* ファイルを生成するよう選択する必要があります。これによりデプロイメント・ウィザードは、すべての HTML およびサポート・ファイルを含む単一の *zip* ファイルを作成します。*zip* ファイルを Host On-Demand サーバーに移動し、これを *DWunzip* を使用して任意のパブリッシュ・ディレクトリーに展開する必要があります。セッションが定義済みであれば、HTML ページはユーザーからアクセス可能な状態になります。

第 9 章 Host On-Demand の管理クライアントおよび新規ユーザー・クライアントの使用

Host On-Demand には、Host On-Demand の管理用と新規ユーザー・アカウント作成用にいくつかの事前定義クライアントが用意されています。構成サーバー・ベース・モデル、または結合モデルを使用して展開したエミュレーター・クライアントや Database On-Demand クライアントにアクセスする前に、まず管理クライアントまたはフル管理クライアントのいずれかを使用して、ユーザーを追加し、そのセッションを構成する必要があります。

管理クライアントおよび新規ユーザー・クライアントのロード

管理クライアントまたは新規ユーザー・クライアントをロードするには、次のいずれかを行います。

- ブラウザーに HTML ファイルの全 URL を指定します。

`http://server_name/hod_alias/client_name.html`

ここで、*server_name* は Host On-Demand サーバーのホスト名または IP アドレス、*hod_alias* はパブリッシュ・ディレクトリーの別名 (またはパス)、*client_name* は管理クライアントまたは新規ユーザー・クライアントの HTML ファイル名です。例えば、次のように URL を指定することにより、Web サーバーからキャッシュ・バージョンの管理クライアントをダウンロードすることができます。

`http://host.yourcompany.com/hod/HODAdminCached.html`

初期インストールの後、以下のようにして、最初に管理者としてログオンします。

1. デフォルト・ユーザー ID を入力: admin
 2. デフォルト・パスワードを入力: password
 3. 「ログオン」をクリックします。
- HODMain_xx.html ファイルをブラウザにロードして (xx は 2 文字の言語接尾部)、すべての使用可能な管理クライアントと新規ユーザー・クライアント、および他の事前定義クライアントへのリンクを表示させます。HODMain_xx.html は、パブリッシュ・ディレクトリーにあります。

管理クライアント

管理クライアントを使用することにより、構成サーバーに保管されたデータについて以下の操作を実行できます。

- ユーザー、グループ、およびセッションの管理
- リダイレクター・サービスの構成、管理、およびトレース
- Database On-Demand の構成
- セキュリティーの使用可能化

- トレースおよびメッセージ・ログの表示
- エンド・ユーザーに対する機能の使用不可

管理クライアントは、Macintosh オペレーティング・システムを除くすべての Host On-Demand クライアント・プラットフォーム上で稼働します。構成サーバー・ベース・モデルまたは結合モデルを使用して、デプロイメント・ウィザードで HTML ファイルを作成する場合、管理クライアントを使用して構成サーバー上のセッションを構成する必要があります。Host On-Demand 構成サーバーの構成についての詳細は、オンライン・ヘルプの『基本構成ステップ』を参照してください。

Host On-Demand には、以下の管理クライアントおよびフル管理クライアントが事前定義されています。

管理クライアント (HODAdmin.html)

ダウンロード・バージョンの管理クライアントをロードします。

キャッシュされた管理クライアント (HODAdminCached.html)

キャッシュ・バージョンの管理クライアントをロードします。このクライアントを使用する利点は、キャッシュ・クライアントとともにブラウザーでキャッシュに入れることができる点です。



キャッシュされた管理クライアントにブックマークを付けるには、手動でブックマークを作成する必要があります。このブックマークは、HODAdminCached.html を指している必要があります。指している場合は、Host On-Demand はキャッシュ・バージョンとサーバー・バージョンとを比較することができます。これにより、Host On-Demand は、キャッシュされた管理クライアントの新しいバージョンがサーバーで使用可能であると認識した際に、それをユーザーに通知することができます。

キャッシュされた管理クライアント (問題判別あり)

(HODAdminCachedDebug.html)¹

問題判別 (セッション・ログおよびトレース) を有効にして管理クライアントをキャッシュ環境にロードします。

フル管理クライアント (HODAdminFull.html)²

ダウンロード・バージョンのフル管理クライアントをロードします。フル管理クライアントにより、管理者は実行時プロパティーを構成するためのセッションを開始できるようになります。ただし、フル管理クライアントのダウンロード・サイズは、管理クライアントよりも大きくなります。

キャッシュされたフル管理クライアント (HODAdminCachedFull.html)²

キャッシュ・バージョンのフル管理クライアントをロードします。通常の管理クライアントのキャッシュ・バージョンと同様に、このクライアントもキャッシュ・クライアントとともにブラウザーでキャッシュに入れることができます。

キャッシュされたフル管理クライアント (問題判別あり)

(HODAdminCachedDebugFull.html)^{1, 2}

問題判別 (セッション・ログおよびトレース) を有効にしてキャッシュ・バージョンのフル管理クライアントをロードします。

注:

1. 問題判別クライアントは、Host On-Demand インストール・システムにおいて問題が発生した際に、サポートと一緒にその問題を解決する場合にだけ使用します。
2. フル管理クライアントは、「セッション開始」を有効にした管理クライアントです。
3. Java 対応ブラウザを使用している場合、Java コントロール・パネルを使用して Administration キャッシュ・クライアントを除去する必要があります。手順については、オンライン・ヘルプの『Java プラグインの使用 (Using the Java plug-in)』を参照してください。

ディレクトリー・ユーティリティー

ディレクトリー・ユーティリティーは Java アプリケーションであり、管理者はこれを使用して、ユーザー、グループ、またはセッションの構成情報を管理できます。この情報は、Host On-Demand のデフォルトのデータ・ストアまたは LDAP ディレクトリーに保管されます。このユーティリティーは、構成サーバー・ベース・モデルを使用している環境でのみ有効です。ディレクトリー・ユーティリティーにより、管理クライアントを使用せずにバッチ・モード環境で多数のユーザー、グループ、またはセッションを追加、削除、または更新することができます。ディレクトリー・ユーティリティーは、構成サーバーに定義されたユーザー、グループ、またはセッションに対して実行する以下のアクションが記述された XML ASCII ファイルを読み取ります。

- グループの追加、更新、および削除
- グループのユーザーの追加、更新、および削除
- ユーザーまたはグループのセッションの追加、更新、および削除
- 既存のユーザーおよびグループを、単一の検索の成果として出力ファイルにリスト
- 既存のユーザーおよびグループを、入力として再利用できる出力ファイルにリスト



リスト・アクションで実行される検索は、ユーザー・ベース (ユーザーごとの情報を戻す) かグループ・ベース (グループごとの情報を戻す) のいずれかです。ただし、LDAP 環境では、ユーザー・ベースの検索だけがサポートされます。

詳しくは、オンライン・ヘルプの『ディレクトリー・ユーティリティーの使用 (Using the Directory Utility)』を参照してください。

新規のユーザー・クライアント

管理者が「ユーザー/グループ」ウィンドウの「ユーザーによるアカウントの作成を許可」を有効にしている場合、ユーザーは事前定義された新規ユーザー・クライアントを使用して新規アカウントを作成できます。このクライアントについて詳しくは、オンライン・ヘルプの『ユーザーによるアカウントの作成を可能にする』トピックを参照してください。

Host On-Demand には、以下の新規ユーザー・クライアントが提供されています。

新規ユーザー・クライアント (NewUser.html)

ダウンロード・バージョンの新規ユーザー・クライアントをロードします。

キャッシュされた新規ユーザー・クライアント (NewUserCached.html)

新規ユーザー・クライアントをキャッシュ環境にロードします。

新規ユーザー・クライアント (問題判別あり) (NewUserCachedDebug.html)¹

問題判別 (セッション・ログおよびトレース) を有効にして新規ユーザー・クライアントをキャッシュ環境にロードします。

注: 問題判別クライアントは、Host On-Demand インストール・システムにおいて問題が発生した際に、IBM サポートと一緒にその問題を解決する場合にだけ使用します。

第 10 章 Host On-Demand エミュレーター・クライアントの使用

この章では、Host On-Demand 端末エミュレーター・クライアントを構成して使用する際の留意点について説明します。

- 『エミュレーター・クライアントのロード』では、Host On-Demand エミュレーター・クライアントへのアクセス方法について説明します。
- 82 ページの『適切なクライアントの選択』では、ユーザーのニーズに最適なクライアントの決定方法について説明します。
- 83 ページの『キャッシュ・クライアント』では、キャッシュ・クライアントの使用方法について説明します。これには、それらのインストールおよび除去、インターネット経由のデプロイ、Windows および Mac OS X のサポート、および問題のトラブルシューティングが含まれています。
- 93 ページの『Web Start クライアント』では、Web Start クライアントの使用方法について説明します。これには、そのインストールおよび除去、Web ブラウザーの構成、Windows 制限付きユーザーでの Web Start の使用、およびアップグレードが含まれます。
- 98 ページの『ダウンロード・クライアント』では、ダウンロード・クライアントの使用方法について説明します。これには、それらのインストール、およびキャッシュ・クライアントまたは Web Start クライアントをダウンロードした後のそれらのロードが含まれます。
- 98 ページの『事前定義エミュレーター・クライアント』では、Host On-Demand が提供する定義済みのエミュレーター・クライアントについて説明します。
- 99 ページの『クライアントのダウンロード・サイズの削減』では、クライアントのダウンロード・サイズを減らす方法について説明します。
- 100 ページの『カスタマー提供の Java アーカイブおよび Java クラスのデプロイ』では、Java アーカイブおよびクラス・ファイルをクライアントに展開する方法について説明します。

エミュレーター・クライアントのロード



Host On-Demand では、ダウンロード・クライアント・コンポーネントおよび Java 自動検出コンポーネントであらかじめ構成されている、すぐに使用可能な 3270、5250、VT、および FTP のエミュレーター・セッションのサンプル HTML ファイルが提供されています。これらのセッションは、HTML ベースの構成モデルを使用し、Host On-Demand を迅速に稼働させてホスト・システムにアクセスできるようにするために提供されています。詳しくは、73 ページの『第 8 章 Host On-Demand エミュレーター・クライアントの構成』を参照してください。

Host On-Demand エミュレーター・クライアントをロードするには、Web ブラウザーを開始し、「アドレス」フィールドに Host On-Demand HTML ファイルの URL を入力します。Host On-Demand HTML ファイルは次のいずれかでなければなりません。

- デプロイメント・ウィザードで作成する HTML ファイル。
- Host On-Demand と共に組み込まれている定義済みのいくつかの汎用 HTML ファイルの 1 つ。

IBM では、最初のオプションを推奨します。デプロイメント・ウィザードについての詳細は、オンライン・ヘルプの『デプロイメント・ウィザード (Deployment Wizard)』を参照してください。定義済みの汎用 HTML ファイルについての詳細は、

98 ページの『事前定義エミュレーター・クライアント』を参照してください。



エミュレーター・クライアントの展開を構成サーバー・ベース・モデルまたは結合展開モデルにより行う場合は、エミュレーター・クライアントを使用する前に、管理クライアントを使用してユーザーを追加しセッションを構成する必要があります。

デプロイメント・ウィザードにより生成された HTML ファイルを起動するには、ブラウザーに HTML ファイルの全 URL を指定します。

`http://server_name/hod_alias/client_name.html`

ここで、`server_name` は Host On-Demand サーバーのホスト名または IP アドレス、`hod_alias` はパブリッシュ・ディレクトリーの別名 (またはパス)、`client_name` はクライアントの HTML ファイル名です。例えば、デプロイメント・ウィザードで HTML ファイル `3270sessions.html` を作成した場合、次のように URL を指定することによりこれをロードできます。

`http://host.yourcompany.com/hod/3270sessions.html`

Host On-Demand と共に組み込まれた定義済み HTML ファイルを起動するには、ブラウザーが `HODMain_xx.html` ファイルを指すようにし (`xx` は 2 文字の言語接尾部)、すべての使用可能な事前定義クライアントへのリンクを表示させます。`HODMain_xx.html` は、パブリッシュ・ディレクトリーにあります。

クライアントにアクセスすると、セキュリティに関する警告が表示され、Host On-Demand が **International Business Machines** により作成されたことを示します。ユーザーは、Host On-Demand が正しく動作するように、適切なボタンをクリックして、このセッションまたは将来のセッションの Java セキュリティ特権を付与する必要があります。

注: ポップアップ・ブロッカーにより、Java セキュリティ・ウィンドウやその他の応答ウィンドウが表示されないことがあります。

適切なクライアントの選択

使用する Host On-Demand クライアントのタイプは、コンピューティング環境や個人の好みに応じて選ぶことになります。

キャッシュ・クライアントおよび Web Start クライアントはローカルに保管され、ダウンロード・クライアントよりも高速にロードされます (Web サーバーから更新済みバージョンのクライアントがダウンロードされる場合を除く)。ネットワーク接続でもダイヤルアップ接続でも、同様に使用することができます。キャッシュ・クライアントおよび Web Start クライアントは、ダウンロード・クライアントよりもローカル・ディスク・スペースを必要としますが、ほとんどのマシンではこれは問題になりません。

Web Start クライアントにより、ブラウザがなくてもユーザーは Host On-Demand セッションを実行できるようになります。ユーザーは、Host On-Demand セッションを Java Web Start Application Manager から開始します。ユーザーが Host On-Demand デスクトップをクローズする時アクティブ・セッションが実行中である場合、すべてのセッションをクローズしてよいかを確認するプロンプトが出されます。

ダウンロード・クライアントは一般的に、高速なネットワーク接続を利用し、Web サーバーからのダウンロードにかかる時間が削減できるような LAN 接続環境で使用されます。ダウンロード・クライアントは、使用するたびにダウンロードする必要があり、ダイヤルアップ接続ではより時間を必要とすることから、低速のダイヤルアップ接続での使用はお勧めできません。ダウンロード・クライアントはディスク上の占有スペースが少ないことから、特に NetStation マシンなど、ローカル・ディスク・スペースの多くないクライアント・マシンに十分適しています。

キャッシュ・クライアント、Web Start クライアント、およびダウンロード・クライアントを、同じ Host On-Demand 環境で使用できます。キャッシュ・クライアントの除去に関する説明は、88 ページの『キャッシュ・クライアントの除去』を参照してください。

Web Start クライアントを使用する予定がある場合、デプロイメント・ウィザードを使用して HTML ファイルを生成する必要があります。キャッシュ・クライアントまたはダウンロード・クライアントを使用する場合、いずれかの事前定義クライアントを使用するのではなく、デプロイメント・ウィザードを使用してユーザー独自のクライアントを作成することをお勧めします。詳しくは、99 ページの『クライアントのダウンロード・サイズの削減』を参照してください。

キャッシュ・クライアント

Host On-Demand キャッシュ・クライアントは、コンポーネントがユーザーのワークステーションのハード・ディスクにキャッシュされた (高速アクセスできるように保管された) 任意の Host On-Demand クライアントです。ユーザーが最初にキャッシュ・クライアントを実行する時、Host On-Demand 始動コードは Host On-Demand クライアント・コンポーネントをダウンロードし、それらをユーザーのワークステーションのハード・ディスクに保管します。これは、キャッシュ・クライアントのインストールと呼ばれます。

その後ユーザーがキャッシュ・クライアントを実行する時、Host On-Demand 始動コードは、サーバーから小さい開始アプレットのみをダウンロードします。その後開始アプレットは、ハード・ディスクのキャッシュ・コンポーネントから Host On-Demand クライアントを開始します。

キャッシュ・クライアントを使用することにより、Host On-Demand クライアント・コンポーネントがワークステーションのハード・ディスク上で即時に使用可能になっているため、それらをダウンロードするためにユーザーが待機する必要がなくなります。加えて、キャッシュ・クライアントは、オペレーティング・システムを再始動してブラウザを再ロードしても同じ状態が保持されます。キャッシュ・クライアントは、本来大きなアプレットのダウンロードに時間がかかるダイヤルアップ電話回線などの低速の接続環境のユーザーを意図していましたが、高速回線の場合でも多くのユーザーはキャッシュ・クライアントを選択しています。

すべての Host On-Demand クライアントと同様、キャッシュ・クライアントは、サポートされる Web ブラウザーの「アドレス」フィールドに Host On-Demand HTML ファイルの URL を指定することにより、開始されます (初回もそれ以降も)。IBM は、デプロイメント・ウィザードを使用して独自の HTML ファイルを作成することをお勧めします。しかしながら、Host On-Demand と共に組み込まれている定義済み汎用キャッシュ・クライアント HTML ファイルの 1 つを使用することもできます。

キャッシュ・クライアントを開始するアプレットは、Host On-Demand サーバーのいずれかの Host On-Demand クライアント・コンポーネントのバージョン番号が、対応するダウンロードしたコンポーネントのバージョン番号より新しいかどうかも判別します。ダウンロードしたコンポーネントのバージョン番号より新しい場合には、アプレットは、キャッシュ・クライアントを立ち上げる前にサーバーから新しいコンポーネントをダウンロードしてキャッシュすることにより、キャッシュ・クライアントをアップグレードします。

ユーザーは、同じワークステーションに複数のタイプのキャッシュ・クライアントをインストールできます。例えば、エミュレーター・キャッシュ・クライアント、Database On-Demand キャッシュ・クライアント、および管理キャッシュ・クライアントを、すべて 1 つのワークステーションにインストールできます。さらに、Java バージョンの Host On-Demand では、ユーザーは同じキャッシュ・クライアントの 2 つのバージョン (1 つは問題判別あり、もう 1 つは問題判別なし) をインストールできます。

キャッシュ・クライアントのインストール

キャッシュ・クライアントを、Host On-Demand サーバーから、または LAN ドライブか DVD ドライブからインストールできます。

キャッシュ・クライアント用にインストールされる情報

Java キャッシュ・クライアントがインストールされると、ユーザーのワークステーションには次の 2 種類の情報が保管されます。

- Host On-Demand コンポーネント

これらのコンポーネントは、Java アーカイブ・ファイル (JAR) の形式です。

- 制御情報

この情報には、Host On-Demand サーバーの URL やダウンロードされた各コンポーネントのバージョンなどが含まれます。

Java キャッシュ・クライアント: Java キャッシュ・クライアント始動コードは、ユーザーがキャッシュ・クライアントをダウンロードする元となったサーバーごとに、ワークステーションのハード・ディスクの異なるディレクトリーにキャッシュ・クライアント・コンポーネントを保管するので、Java キャッシュ・クライアントの複数のバージョンがユーザーのワークステーションに存在できます。

Java キャッシュ・クライアントにおいて、同じサーバーからダウンロードされたすべてのクライアント・コンポーネントは、ユーザーのハード・ディスクの同じディレクトリーに保管されます。例えば、ユーザーが Java エミュレーター・クライアントおよび Java Database On-Demand クライアントを同じサーバーからインストールする場合には、両方のタイプのクライアント用のコンポーネント・ファイルは同じディレクトリーに保管されます。

Java キャッシュ・クライアントのいくつかの特殊なタイプでは、クライアント・コンポーネントは Java プラグインの *sticky* キャッシュ に保管されます。これらは、15 ページの『サポートの制限』にリストされているものと同じキャッシュ・クライアント・タイプです。

キャッシュ・クライアントを **Host On-Demand** サーバーからインストール

キャッシュ・クライアントを Host On-Demand サーバーからインストールするには、以下のようにします。

1. ブラウザーで HTML ファイルの完全な URL を指定する (81 ページの『エミュレーター・クライアントのロード』を参照)。
2. 事前定義クライアントを使用する場合は、`http://server_name/hod_alias/HODMain.html` をロードしてからキャッシュ・クライアントのリンクをクリックする (`server_name` は Host On-Demand サーバーのホスト名または IP アドレス、`hod_alias` はパブリッシュ・ディレクトリーの別名 (またはパス))。
3. キャッシュ・クライアントがすぐにインストールを開始する。ウィンドウに、インストールの進行状況が表示されます。このウィンドウの上部の進行バーには、個々のファイルのダウンロード状況が示され、下部の進行バーには、インストール全体の状況が示されます。



インストール進行状況ウィンドウは、Java キャッシュ・クライアントのいくつかのタイプでは表示されません。これらは、15 ページの『サポートの制限』にリストされているものと同じ Java キャッシュ・クライアントです。

4. インストールが完了すると、インストール・コードにより、Java キャッシュ・クライアントはすぐに起動される。ユーザーはブラウザーを再始動する必要はありません。

キャッシュ・クライアントを **LAN** または **DVD** からインストール

この時点で、ユーザーの一部またはすべてに、LAN ドライブまたは DVD からキャッシュ・クライアントを初めてダウンロードしてもらうことができます。キャッシュ・クライアントをインストールするには、ユーザーは LAN ドライブまたは DVD に一度だけアクセスする必要があります。インストール後、ユーザーは通常の方法で Host On-Demand サーバーに接続します。

この方式の利点は、キャッシュ・クライアント・コンポーネントが、Web サーバーからダウンロードしなければならない場合よりも速くユーザーのワークステーションにインストールされるという点と、キャッシュ・クライアント・コンポーネントのセット全体をダウンロードしても Web サーバーに余分な負荷がかからないという点です。

この方式は、ほとんどのクライアント・プラットフォームでサポートされています。しかしながら、いくつかの Java キャッシュ・クライアントはこの機能をサポートしていません。この機能をサポートしない Java キャッシュ・クライアントは、15 ページの『サポートの制限』にリストされています。

制限: HTML ファイルは別個のユーザー・パブリッシュ・ディレクトリーを指定できません。(デプロイメント・ウィザードでコード・ベースを指定した場合、LAN または DVD ドライブからキャッシュ・クライアントをインストールするために HTML ファイルを使用することはできません。) 別個のユーザー・パブリッシュ・ディレクトリーについての詳細は、オンライン・ヘルプを参照してください。

管理者が **DVD** または **LAN** イメージを作成するためのステップ:

1. デプロイメント・ウィザードの「ファイル名と出力形式」ウィンドウを使用して、カスタマイズした *.html ファイル (例、MyHOD.html) を作成します。デプロイメント・ウィザード・ファイルを他のサーバーに配布する必要がある場合は、DWunzip を使用できるようにするために「出力 Zip」を選択できます。詳しくは、オンライン・ヘルプの『DWunzip の使用 (Using DWunzip)』を参照してください。
2. Java キャッシュ・クライアントでは、デプロイメント・ウィザードに追加の HTML パラメーター WebServerHostname を指定することにより、ユーザーがインストール中に Host On-Demand サーバーのホスト名を入力することを避けることができます。詳しくは、オンライン・ヘルプの『HTML パラメーター (HTML parameters)』を参照してください。
3. 作成したデプロイメント・ウィザード・ファイルをサーバーにロードした後は、新しいファイルをテストして、正しく機能することを確認してください。
4. Host On-Demand サーバーのインストール・システムのパブリッシュ・ディレクトリーにある以下のファイルを、ネットワーク・ドライブまたは DVD にコピーするか、あるいは FTP 転送します (必ずご利用の Host On-Demand サーバーにあるものと同じバージョンの Host On-Demand を DVD または LAN ドライブに入れてください)。

- MyHOD.html
- MyHOD.jnlp (存在する場合)
- z_MyHOD.html (存在する場合)
- hoddetect*.html
- hodlogo.gif
- hodbkgnd.gif
- Installer.html
- Installer2.html
- *.jar
- *.properties

- *.js
5. ディレクトリー構造はそのままにしながら以下のファイルとディレクトリーをコピーします。
- msgsrc\cached_*.properties
 - HODData\MyHOD*.*



これらのファイルを z/OS インストールから DVD イメージにコピーする場合は、先に、すべての HTML、PROPERTIES、JS、JNLP、および CSS ファイルから .ascii ファイル拡張子を除去する必要があります。例えば、*.properties.ascii という名前のファイルは、*properties として DVD にコピーする必要があります。



キャッシュ・クライアントのインストールに DVD を使用する場合は、その DVD には暗号化テクノロジーが含まれるため、配布に際してはご使用条件および輸出入規制と同様のガイドラインに従ってください。

ユーザーのステップ: 管理者が LAN ドライブまたは DVD のセットアップを完了した後、キャッシュ・クライアントをインストールするためにユーザーは以下のステップを実行する必要があります。

1. 以下のようにして、インストールのためにクライアント・マシンを準備します。
 - LAN ドライブまたは DVD ドライブにアクセスします。
 - システム管理者が LAN ドライブまたは DVD に置いた HTML ファイルの名前と場所 (f:\myPath\MyHOD.html など) を取得します。(その HTML ファイルは、すべてのユーザーに対して同じ名前と同じ内容を持っています。1 人のユーザーに固有のものではありません。)
 - Java キャッシュ・クライアントの場合のみ、キャッシュ・クライアントのインストール後にユーザーが接続する Host On-Demand サーバーのホスト名を検出します。例えば、ユーザーが `http://myHODServer/hod/MyHOD.html` に接続する場合には、ホスト名は `myHODServer` となります。



Java キャッシュ・クライアントでは、システム管理者は HTML パラメーター `WebServerHostname` をその HTML ファイルに追加することにより、このステップを省略できます。オンライン・ヘルプの『HTML パラメーター (HTML parameters)』を参照してください。

2. 以下のようにして HTML ファイルを実行します。

HTML ファイルのパスと名前をブラウザのアドレス入力フィールドに入力します。例:

`f:/mypath/MyHOD.html`

3. Java キャッシュ・クライアントの場合のみ、インストール・コードによりプロンプトが出された時には、キャッシュ・クライアントのインストール後にユーザーが接続する Host On-Demand サーバーのホスト名を入力します。例えば、ユーザーが `http://myHODServer/hod/MyHOD.html` を起動する場合には、ホスト名は `myHODServer` となります。



Java キャッシュ・クライアントでは、システム管理者は HTML パラメーター `WebServerHostname` をその HTML ファイルに追加することにより、このステップを省略できます。オンライン・ヘルプの『HTML パラメーター (HTML parameters)』を参照してください。

4. LAN ドライブまたは DVD から Host On-Demand キャッシュ・クライアントがインストールされている間待ちます。
5. プロンプトが出されたらブラウザを再始動し、それが Host On-Demand サーバー上で同じ名前の HTML ファイルを指すようにします。例:

`http://myServer/hod/MyHOD.html`

Host On-Demand サーバー上の HTML の名前は、LAN または DVD 上の HTML ファイルの名前と同じです。

これらのステップの完了後、Host On-Demand キャッシュ・クライアントは通常の方法で開始します。

キャッシュ・クライアントの除去

以下のセクションでは、汎用の除去方法について説明します。

始める前に

キャッシュ・クライアントを除去するということは、Java キャッシュ・クライアントがインストールされた時にユーザーのハード・ディスクに保管された情報を消去することを意味します。

Java バージョンのキャッシュ・クライアントを実行するユーザーは、キャッシュ・クライアントをダウンロードしたそれぞれの Host On-Demand サーバーごとに、別個のバージョンのキャッシュ・クライアントを持っています。詳しくは、84 ページの『キャッシュ・クライアント用にインストールされる情報』を参照してください。

Java キャッシュ・クライアントを除去しても、除去を行う時にユーザーがアクセスするサーバーからダウンロードされたバージョンの Java キャッシュ・クライアントしか除去されません。例えば、ユーザーがサーバー `myHODServerA` の `http://myHODServerA/hod/HODRemove.html` にアクセスしてユーザーのワークステーションの Java キャッシュ・クライアントを除去する場合には、`myHODServerA` からダウンロードされた Java キャッシュ・クライアントのみが除去されます。

最後に、Java キャッシュ・クライアントにおいて、キャッシュ・クライアントを除去すると、そのインストールに関連したすべてのタイプのキャッシュ・クライアント (エミュレーション、Database On-Demand、および管理) が除去されます。

サーバー `myHODServerA` に接続中にワークステーションから Java キャッシュ・クライアントを除去すると、`myHODServerA` から以前にダウンロードされた、エミュレーション・キャッシュ・クライアント、Database On-Demand キャッシュ・クライアント、および管理キャッシュ・クライアントが除去されます。しかしながら、除去されるのはそのサーバーからダウンロードされたキャッシュ・クライアント・コ

ンポーネントのみです。他のサーバーからのキャッシュ・クライアント・コンポーネントがある場合には、ユーザーがそのサーバーに接続して除去を実行するまで、それは除去されません。

Java キャッシュ・クライアントの除去

汎用の除去方法で Java キャッシュ・クライアントを除去します。以下のステップに従ってください。

1. ブラウザーを開始します。

Java キャッシュ・クライアントを除去するため、Java 対応ブラウザを開始してください。

2. Host On-Demand サーバー上の HODMain.html に接続します。例えば、次の URL に接続します。

`http://myServer/HOD/HODMain.html`



Java キャッシュ・クライアントを除去する場合、正常にそれを除去するには、Java キャッシュ・クライアントをインストールする元となったサーバーと同じサーバーに接続する必要があります。詳しくは、88 ページの『始める前に』を参照してください。

3. 「ユーティリティ」の下にある次の項目をクリックします。

Remove Cached Client

この汎用の除去を実行するための、さらに直接的な代替方法もあります。以下のステップに従ってください。

1. ブラウザーを開始します。
2. Host On-Demand サーバー上の HODRemove.html に接続します。例えば、次の URL に接続します。

`http://myServer/HOD/HODRemove.html`

これでキャッシュ・クライアントが除去されます。



Java キャッシュ・クライアントを除去する場合、正常にそれを除去するには、Java キャッシュ・クライアントをインストールする元となったサーバーと同じサーバーに接続する必要があります。詳しくは、88 ページの『始める前に』を参照してください。

どちらの汎用の除去方法を使用したとしても、以下の Java キャッシュ・クライアントを除去した場合には、Java プラグインのキャッシュを消去するというプロンプトが出されます。

- 管理キャッシュ・クライアント
- Apple Mac OS X 上のキャッシュ・クライアント
- JavaScript Session Manager API が使用可能になっているエミュレーター・キャッシュ・クライアント (Java Mozilla のみ)

ウィンドウが表示されて、Java プラグインのキャッシュを消去するための通知が出されます。詳しくは、オンライン・ヘルプの『Java プラグインの使用 (Using the Java plug-in)』を参照してください。

複数のユーザーが共有するキャッシュ・クライアントの除去

複数のユーザーが単一のキャッシュ・クライアントを共有しており、そのうちの 1 人のユーザーがキャッシュ・クライアントを除去する場合、キャッシュ・クライアントはすべてのユーザーに対して除去されます。単一キャッシュ・クライアントの共有については、『Windows のキャッシュ・クライアント・サポート』を参照してください。

複数の Host On-Demand サーバーにアクセスする際のキャッシュ・クライアントのサポートの問題

以下のセクションでは、キャッシュ・クライアント・ユーザーが複数の Host On-Demand サーバーにアクセスする時に発生する可能性がある問題を詳しく説明しています。

Java キャッシュ・クライアント

Host On-Demand Java キャッシュ・クライアントは、ユーザーがアクセスする Host On-Demand サーバーごとにキャッシュ・クライアント・コードの別々のコピーをインストールします。それゆえに、様々なサービス・レベルでサーバーにアクセスする上で問題は生じません。一部のプラグイン・バージョンでは、ユーザーは、多数の Host On-Demand サーバーにアクセスする際に、Java のキャッシュ・サイズを増やす必要がある場合があります。

Java キャッシュ・クライアントに関して、以下の問題が発生する場合があります。

ローカルに保管された設定を使用する上での問題: ローカルに保管された設定を使用している場合は、作成したカスタム HTML ファイルは企業で固有な名前であればなりません。これは、別のサイトのローカルに保管された設定間で HTML ファイル名が異ならないなければならないからです。総称名を使用すると、ユーザーの設定競合が起きる場合があります。

さらに情報を得るために **Host On-Demand サポート Web サイト**を参照する: インターネットではキャッシュ・クライアント・デプロイメントを管理する場合に問題があれば、さらに情報を得るために <http://www.ibm.com/software/webservers/hostondemand/support.html> をご覧ください。

Windows のキャッシュ・クライアント・サポート

Windows 7、Windows 8、Windows 10、または Windows 2012 のいずれかのオペレーティング・システムが稼働するマルチユーザー Windows マシンでは、ユーザーは自身の独立したバージョンのキャッシュ・クライアントをダウンロードできます。

- Java プラグインを持つすべてのサポートされているブラウザ

JavaScript API が使用可能な場合、技術的な制限のために、Mozilla Java ブラウザーではキャッシュ・クライアントを共有することはできません。

代わりに、デプロイメント・ウィザードの「拡張オプション」ウィンドウの HTML パラメーターの選択を使用して、以下のパラメーターを追加できます。

- **ShareCachedClient:** キャッシュ・クライアントの単一のインスタンスをユーザーが共有することを許可します。

- **SharedCachedDirectory:** キャッシュ・クライアントがインストールされるディレクトリーの場所を指定することを許可します。

ディレクトリーを指定せずにキャッシュ・クライアントを共有する場合、キャッシュ・クライアントはデフォルトのディレクトリーである `¥Documents and Settings¥All Users¥IBM¥HOD` にインストールされます。ディレクトリーを指定した場合 (例えば、`SharedCachedDirectory=c:¥ibm¥`)、**Host On-Demand** キャッシュ・クライアントは `IBM¥HOD¥HODCC` をこのストリングに付加し、キャッシュ・クライアントはこの新しい場所 (例えば、`c:¥ibm¥IBM¥HOD¥HODCC`) にインストールされます。管理者またはパワー・ユーザーは、インストール・ディレクトリーを手動で作成するか、最初から共有キャッシュ・クライアントとしてインストールを行う必要があります。いずれの場合も、管理者またはパワー・ユーザーは、このディレクトリーのセキュリティ設定を変更して、制限ユーザーに読み取り、修正、および書き込みアクセス権を与える必要があります。管理者は、セキュリティ設定を変更してからキャッシュ・クライアントをディレクトリーにダウンロードしても、または共有キャッシュ・クライアントをディレクトリーにダウンロードしてからセキュリティ設定を変更してもかまいません。セキュリティ設定が更新されていない場合、制限ユーザーが共有キャッシュ・クライアントをインストールしようとすると、そのユーザーは、ファイル・システムに問題がある可能性を示すエラー・メッセージを受け取り、キャッシュ・クライアントを使用または更新できなくなります。

管理者またはパワー・ユーザーがセキュリティ設定を変更すると、制限ユーザーは **Windows** にログオンして、共有キャッシュ・クライアントをインストール、または以前にインストールされたバージョンの共有キャッシュ・クライアントを使用 (または更新) することができます。他の制限ユーザーも **Windows** にログオンして、**Host On-Demand** サーバーから再度ダウンロードすることなく、キャッシュ・クライアントを使用することができます。また、必要に応じて共有キャッシュ・クライアントをアップグレードすることもできます。

制限ユーザーにキャッシュ・クライアントを共有させたくない場合は、各制限ユーザーのユーザー・ディレクトリーにキャッシュ・クライアントの別個のインスタンスをダウンロードするようにします。

管理者またはパワー・ユーザーが以前のバージョンのキャッシュ・クライアントをダウンロードしており、これに制限ユーザーがアクセスできるようにする場合は、管理者またはパワー・ユーザーは `HODRemove.html` を使用して、以前のバージョンのキャッシュ・クライアントを除去し、前記のように、共用キャッシュ・クライアント・ディレクトリーのセキュリティ設定を、制限ユーザーに対して読み取り、修正、および書き込み可能に変更する必要があります。

キャッシュ・クライアントの除去に関する情報は、90 ページの『複数のユーザーが共有するキャッシュ・クライアントの除去』を参照してください。

Mac OS X (Java クライアントのみ) の場合のキャッシュ・クライアントのサポート

キャッシュ・クライアントは、Mac OS X では以下の制限があります。

- **Host On-Demand** 更新の過程は、サーバーごとに管理されます。

- DVD ドライブまたは LAN ドライブからキャッシュ・クライアントをプリロードしても機能しません。ブラウザーが実際の Web サイトに指定変更されると、プラグインはそれを別の Web サーバーであると見なし、そのクライアントが再度キャッシュに入れられます。
- Host On-Demand はアプレットとして実行され、ユーザーのマシンにコードをダウンロードする必要があります。Host On-Demand クライアントはすべてのコンポーネントをダウンロードしますが、不要なコンポーネントを除去することにより、ダウンロード・サイズを削減できます。Mac OS X では、初期のダウンロードの後に追加コンポーネントをインストールすることはできません。
- Java 対応 Web ブラウザー上で Host On-Demand キャッシュ・クライアントを実行する場合に使用される Host On-Demand Java ファイルは、Java Runtime Environment (JRE) キャッシュに保管されます。Mac OS X 上でキャッシュ・クライアントを除去するには、Java コントロール・パネルを使用して JRE キャッシュを消去する必要があります。手順については、オンライン・ヘルプの『Java プラグインの使用 (Using the Java plug-in)』を参照してください。
- キャッシュ・クライアントを実行する場合は、新しいバージョンのクライアント・コードが使用可能になった際に、古いコードをアップグレードする必要があります。デプロイメント・ウィザードには、アップグレードのタイミングを制御するための多数のオプションがあります。これらのオプションは、Mac OS X では選択できません。



Java キャッシュ・クライアントの改善点は、Mac OS X Java キャッシュ・クライアントには適用されません。詳しくは、15 ページの『サポートの制限』を参照してください。

キャッシュ・クライアントのトラブルシューティング

キャッシュ・クライアントをロードできないことが判明した場合は、以下にあるトラブルシューティングの提案に従ってください。

Microsoft Internet Explorer 11.0

ブラウザーを Microsoft Internet Explorer 11.0 にアップグレードした後で、Java コンソールにセキュリティー例外が表示されることがあります。キャッシュ・クライアントをインストールすると、いくつかのファイルがブラウザーのディレクトリー構造に保管されます。Internet Explorer をバージョン 11.0 にアップグレードすると、ブラウザーは Host On-Demand のキャッシュ・コードを含んでいる CAB ファイルを認識できなくなります。ブラウザーは CAB ファイルを検索できないので、サーバーからのクラス・ファイルを直接使用しようとして、セキュリティー例外を起こします。この問題を解決するには、ブラウザーのバージョン・アップグレードをし、HODRemove.html を使用して Host On-Demand を除去し、HODCached.html を使用してその製品を再インストールする必要があります。

Mozilla および Firefox

Mozilla および Firefox ブラウザーでは、キャッシュ・クライアントのインストールを試行するときに何も起こらなかった場合、またはキャッシュ・クライアントのインストールの試行に失敗した場合には、ブラウザーの設定をチェックしてください。「ナビゲーター」ウィンドウの上部または下に表示されるポップアップ・ウィ

ンドウを表示しないように Mozilla および Firefox が設定されていないことを確認します。この設定では、Host On-Demand キャッシュ・クライアントがインストールされません。

この設定を行う場所は、以下のように Mozilla のバージョンに依存しています。

- Mozilla 1.2 では、この設定は、「編集 (Edit)」>「設定 (Preferences)」>「詳細」>「スクリプトとプラグイン」に含まれています。
- Mozilla 1.3 では、この設定は、「編集 (Edit)」>「設定 (Preferences)」>「プライバシーとセキュリティ (Privacy & Security)」>「ポップアップ・ウィンドウ (Popup Windows)」に含まれています。

キャッシュ・クライアントのインストール後、この設定を復元してポップアップ・ウィンドウが表示されないようにできます。しかし、キャッシュ・クライアント全体を再度インストールする必要がある場合や、フォアグラウンドで新しいバージョンに更新する必要がある場合には、ポップアップ・ウィンドウの表示を抑制しないように Mozilla および Firefox を再度設定する必要があります。



ポップアップ・ウィンドウを表示しないように設定しても、初期のダウンロード (プリロード・リスト) に含まれなかった追加コンポーネントのダウンロードを妨害することはありません。

Web Start クライアント

Java Web Start クライアントにより、ブラウザがなくてもユーザーは Host On-Demand を実行できるようになります。デプロイメント・ウィザードを使用して、Web Start クライアント用の HTML ファイルを生成する必要があります。デプロイメント・ウィザードにより生成された HTML ファイルは、Java Network Launch Protocol (JNLP) ファイルを指します。JNLP ファイルは、Java アプリケーションを定義します。これには、アプリケーションに渡されるパラメーターおよびアプリケーションが使用するクラス・ファイルを含むアーカイブが含まれます。JNLP ファイルおよび関連アーカイブは、Web サーバー上に保管されます。

ユーザーが JNLP ファイルを指す時、ブラウザはクライアント・コンピューター上で Web Start アプリケーションを起動します。ブラウザは、関連アーカイブをダウンロードし、必要最小限の JRE (指定されている場合) があることを保証するためのチェックを行い、アーカイブをユーザーのマシンに保管し、アプリケーションを表すアイコンをセットアップし、アプリケーションを起動します。

ユーザーは、Host On-Demand セッションを Java Web Start Application Manager から開始できます。Java Web Start Application Manager を使用することにより、Host On-Demand セッションはブラウザに依存なくなります。そのため、ブラウザをクローズしても Host On-Demand セッションは終了しません。ユーザーが Host On-Demand デスクトップのクローズを試行する時アクティブ・セッションが実行中である場合、すべてのセッションをクローズするかを確認するプロンプトが出されます。クローズする場合、ブラウザで実行中のセッションがあり、ブラウザが突然クローズされる時に発生する問題を避けるために、セッションはクリーンに終了します。

アプリケーションの初期起動の後、JNLP ファイルで Web ブラウザーを指すようにするか、あるいはクライアント・マシンで作成されたアイコンをマウスでクリックすることができます。Web Start が再始動した後、Web サーバーにアーカイブの更新がないかを検査し、すべての更新されたファイルをダウンロードします。

Java Web Start は、JRE (Java ランタイム環境) 1.4.0 以降にバンドルされています。JRE 1.3 を使用している場合、JRE 1.4 にアップグレードする必要があります。Java Web Start に関する詳細は、<http://www.javasoft.com> を参照してください。Host On-Demand バージョン 13 では Java 1.5 以降が推奨されています。

Host On-Demand Web Start クライアントには、以下の要件があります。

- HTTPS を使用して Web サーバーからファイルにアクセスするには、JRE 1.4 以降が必要。
- Web Start で HTTP プロキシを使用するには、JRE 1.4 以降が必要。
- 「ブラウザの設定を使用 (use Browser settings)」というセッション・プロパティ (プロキシ・サーバーまたは TLS) は、Web Start では使用できません。

Web Start クライアントのインストール

Web Start クライアントには 2 種類のインストール方法があります。通常は、ネットワークを介して Host On-Demand サーバーからインストールします。Web ブラウザーは、使用してもしなくてもかまいません。代わりに、ユーザーは LAN ドライブまたは DVD ドライブからインストールすることもできます。ただし、この場合は、ネットワーク経由でのダウンロードが、若干余分に必要になります。Web Start クライアントは、インストール方法に関係なく、いったんインストールされて Java Web Start Application Manager に組み込まれたら、Application Manager の適切なアイコンをクリックすることで開始できます。

Web Start クライアントを Host On-Demand サーバーからインストール

Web Start クライアントは、Host On-Demand サーバーから、ブラウザを使用して、あるいは使用せずにインストールできます。

Web ブラウザーを使用する場合: Web ブラウザーを使用して Web Start クライアントをインストールするには、以下のステップを実行します。

1. ブラウザーで HTML ファイルの完全な URL を指定する (81 ページの『エミュレーター・クライアントのロード』を参照)。

Web Start クライアントのインストールがすぐに開始します。ウィンドウに、インストールの進行状況が表示されます。このウィンドウの上部の進行バーには、個々のファイルのダウンロード状況が示され、下部の進行バーには、インストール全体の状況が示されます。

2. インストールが完了すると、インストール・コードにより直ちに Web Start クライアントが起動される。ブラウザを再始動する必要はありません。

Web ブラウザーを使用しない場合: Windows ユーザーの場合には、デプロイメント・ウィザードによって生成された JNLP ファイル (myhod.jnlp など) を、エンド・ユーザーに配布します。このファイルが配布されると、ユーザーは「start

myhod.jnlp」と入力して Web Start アプリケーションを開始し、Host On-Demand クライアントのインストールを開始できます。「.jnlp」というファイル拡張子が Web Start アプリケーションに対して登録されるので、Web Start アプリケーションが開始し、ファイルを読み込んで、デプロイメント・ウィザードで生成された JNLP ファイルで指定されている Host On-Demand サーバーから適切なアーカイブ・ファイルをすべてダウンロードします。ダウンロードが完了すると、Host On-Demand Web Start クライアントが開始します。

JNLP ファイルを Windows ユーザーに配布していない場合、またはクライアントが Windows 以外のプラットフォームで稼働している場合であっても、Java Web Start Application Manager を直接開始し、Web サーバー上で JNLP ファイルを位置指定することで、Web ブラウザーを使用せずに Web Start クライアントをダウンロードできます。

Windows クライアントの場合は、以下のステップを実行します。

1. javaws.exe ファイルをダブルクリックして、Java Web Start Application Manager を開く。このファイルは、通常、C:\Program Files\Java Web Start ディレクトリーにあります。
2. Web サーバーの <http://HODServer/HODAlias/myhod.jnlp> にある JNLP ファイルを位置指定する。

Linux クライアントの場合は、`/javaws http://HODServer/HODAlias/myhod.jnlp` と入力すると、Host On-Demand をインストールし、セッションを実行できます。Host On-Demand アイコンが Java Web Start Application Manager に表示されます。このアイコンをダブルクリックすると、Host On-Demand を起動できます。

Web Start クライアントを LAN または DVD からインストール

ネットワーク・トラフィックを減らし、ダウンロード時間を最小限にするため、ユーザーが LAN または DVD から Web Start クライアントをインストールすることが望ましい場合があります。Web Start クライアントとキャッシュ・クライアントは同じキャッシュ・アーカイブを共有するので、大部分の Web Start クライアントは、キャッシュ・クライアントと同じインストール手順を使用してインストールできます。ただし、Web Start クライアントには、ネットワーク経由で Host On-Demand サーバーから直接インストールする必要のある追加コンポーネントが必要です。

Web Start クライアントをインストールするには、管理者が 2 つのステップを実行した後で、エンド・ユーザーが 2 つのステップを実行する必要があります。

最初に、管理者が次の 2 つのステップを実行します。

1. 86 ページの『管理者が DVD または LAN イメージを作成するためのステップ』を参考にし、デプロイメント・ウィザードを使用してキャッシュ・クライアントの HTML ファイルを生成する。
2. もう一度デプロイメント・ウィザードを使用して、前のステップで作成した HTML ファイルを編集し、クライアントのタイプをキャッシュ・クライアントから Web Start クライアントに変更する。(定義済みのセッションおよびブリロードされたコンポーネント・リストが同じになるよう、これ以外は変更しないでください。) この 2 番目の HTML ページは、ユーザー・アクセス用にパブリッシュする必要があります。

HTML ファイルをパブリッシュしたら、次に、ユーザーが次の 2 つのステップを実行する必要があります。

1. 87 ページの『ユーザーのステップ』に従って、管理者が LAN または DVD にセットアップしたキャッシュ・クライアントをインストールする。
2. Host On-Demand サーバーから Web Start クライアントをインストールするステップ (94 ページの『Web ブラウザーを使用する場合』を参照) に従って、Web Start クライアント用に追加コンポーネントをインストールする。
Web Start クライアントのコードにより、Host On-Demand のアーカイブ・ファイルが既にダウンロードされていることが判別されるので、同じファイルが再びダウンロードされることはありません。その他のコンポーネントは直ちにダウンロードされ、Host On-Demand Web Start クライアントが開始します。

Web Start のための Web サーバーの構成

ブラウザーが Web Start アプリケーションを起動するようにするため、管理者は、Web サーバーで JNLP 拡張子を MIME タイプとして登録する必要があります。例として、以下のセクションでは、Apache HTTP Server、IBM HTTP Server、および Microsoft IIS を構成する方法を説明しています。

Apache HTTP Server または IBM HTTP Server

Apache HTTP Server または IBM HTTP Server を Web Start に構成するためには、以下の行を `mime.types` に追加します。

```
AddType Application/x-java-jnlp-file .jnlp
```

Microsoft IIS 7.0

Microsoft IIS を Web Start に構成するには、以下のステップを完了します。

1. 「コントロール パネル」>「管理ツール」>「インターネット インフォメーション サービス」から、「既定の Web サイト」をクリックします。
2. 「プロパティ」の「HTTP ヘッダー」タブをクリックします。
3. 「MIME マップ」の下で、「ファイルの種類」タブをクリックし、「追加」を選択します。
4. 「拡張子」フィールドで、`.jnlp` と入力します。
5. 「コンテンツの種類」フィールドで、`application/x-java-jnlp-file` と入力します。
6. 「OK」をクリックします。

Web Start クライアントのアップグレード

Web Start クライアントを初期インストールした後、デプロイメント・ウィザードで生成された HTML ファイルにユーザーがブラウザーでアクセスした場合、Host On-Demand サーバー上に使用可能な更新情報があると、Host On-Demand から更新を求めるプロンプトが表示されます。ユーザーが更新を選択すると、Java Web Start は、更新されたアーカイブ・ファイルをダウンロードして、Host On-Demand を起動します。ユーザーがアップグレードに同意しないと、Host On-Demand は、次の HTML ファイルの起動時に、再度同じプロンプトを出します。

初期インストール後の **Web Start** コンポーネントの追加

ユーザーが、Java Web Start クライアントにインストールされていない機能を要求する場合、Host On-Demand は、その機能を実行するために必要な追加コンポーネントをインストールするようにユーザーを促します。追加コンポーネントのインストールを選択する場合には、コンポーネントを使用するために Host On-Demand クライアントを再始動する必要があります。

Web Start と **Windows** 制限ユーザー

Java Web Start 1.0.1 を使用する Windows 制限ユーザーは、JRE および Java Web Start を除去し、Java Web Start 1.2 を使用する新しい JRE を再インストールする必要があります。

Web Start でのブックマーク・セッション

Web Start クライアントは、ブラウザの外で実行されるため、ブラウザの機能であるブックマークの設定は使用不可になっています。管理者は、以下のようにして、組み込まれたブックマーク・セッションを実行する時と同じ動作をユーザーに提供する Web Start クライアントを作成できます。

1. デプロイメント・ウィザードの「拡張オプション」ウィンドウで、`true` の値を持つ `HideHODDesktop` パラメーターを追加します。
2. 単一セッションを自動始動に構成します。
3. そのセッションが別のウィンドウで開始しないように構成します。

Web Start の **HTTPS** での使用

Web Start クライアントで HTTPS を使用したい場合、ご利用のセキュア HTTP 接続用に使用される認証局は、名の通った root 権限からのものである必要があります。Host On-Demand をアプレットとして使用し、HTTPS 接続を使用する場合、root 権限がブラウザにとって既知のものでない場合には、HTTPS 接続に使用される証明書を信頼するかを選択できます。Java Web Start は、アプリケーションとして実行されるため、このブラウザ機能は使用できません。Java Web Start が使用する Java 仮想マシンには、それが信頼するいくつかの root 権限が含まれています。HTTPS 接続から来る証明書が、JVM にとって既知のそれらの権限の中の 1 つの root 権限を持っている場合、セキュア接続が確立できます。デフォルトで JVM に既知のもの以外の認証局を使用したい場合 (例えば、自己署名証明書) には、この Java Web Start クライアントにアクセスするクライアントごとに、証明書を JVM の鍵ストアにインポートする必要があります。これは、セキュア HTTP 接続を確立するために必要です。

Web Start クライアントの除去

Web Start クライアントを除去するには、以下の両方のステップを完了してください。

1. Java Web Start Application Manager で、ご利用のアプリケーションを強調表示し、「除去」をクリックします。
2. ブラウザーで `HODRemove.html` を起動します。

ダウンロード・クライアント

キャッシュ・クライアントや Web Start クライアントとは異なり、ダウンロード・クライアントは、いつどのようにクライアント・コンポーネントがワークステーションのハード・ディスクにダウンロードされるかを制御しません。ダウンロード・クライアントは、すべてのキャッシングに関する決定をブラウザに任せます。

次の両方の 要件を満たす場合にダウンロード・クライアントを使用してください。

- キャッシュ・クライアントまたは Web Start クライアントをインストールすることによりクライアント・マシン上のディスク・スペースを消費したくない場合。
- 初期ダウンロード時間が問題とならない場合。

ダウンロード・クライアントの立ち上げ

ダウンロード・クライアントを立ち上げるには、これを Host On-Demand サーバーからブラウザ・ウィンドウにダウンロードします (81 ページの『エミュレーター・クライアントのロード』を参照)。

キャッシュ・クライアントまたは **Web Start** クライアントのインストール後のダウンロード・クライアントの起動

Java

Java クライアントの場合、キャッシュ・クライアントまたは Web Start クライアントをインストールした後に問題なくダウンロード・クライアントを立ち上げることができます。

事前定義エミュレーター・クライアント

Host On-Demand には、いくつかの定義済みエミュレーター・クライアント HTML ファイルが提供されています。これは、Host On-Demand クライアントの機能の範囲を示すためのものであり、カスタマイズされた HTML ファイルをデプロイメント・ウィザードで作成するためのサンプルとして使用します。これはすべて、構成サーバー・ベース・モデルを使用しています。これらのクライアントをロードするには、 81 ページの『エミュレーター・クライアントのロード』の説明に従ってください。



一般的に、事前定義のクライアント HTML ファイルを使用する代わりに、デプロイメント・ウィザードを使用して、ユーザー独自のカスタマイズした HTML ファイルを定義することをお勧めします。

Host On-Demand には、以下のエミュレーター・クライアント HTML ファイルが事前定義されています。

キャッシュ・クライアント (**HODCached.html**)

すべての Host On-Demand クライアント機能を提供します。

キャッシュ・クライアント (問題判別あり) (**HODCachedDebug.html**)¹

問題判別 (セッション・ログおよびトレース) を有効にしてキャッシュ・クライアントを開始します。

ダウンロード・クライアント (HOD.html)

問題判別を除くすべての Host On-Demand クライアント機能を提供します。



Java 対応の Web ブラウザーでは、定義済みダウンロード・クライアント・ファイル HOD.html は、あまり使用されていないいくつかの Host On-Demand コンポーネントを削除します。除外されるコンポーネントのリストおよび次善策の説明を含む詳細については、16 ページの『HTML ファイルにいくつかのコンポーネントが含まれていない』を参照してください。Java ブラウザーでの HOD.html へのアクセスには、機能に制限があります。

ダウンロード・クライアント (問題判別あり) (HODDebug.html)¹

問題判別 (セッション・ログおよびトレース) を有効にしてダウンロード・クライアントをロードします。

注:

1. 問題判別クライアントは、Host On-Demand インストール・システムにおいて問題が発生した際に、IBM サポートと一緒にその問題を解決する場合にだけ使用します。

クライアントのダウンロード・サイズの削減

一般的に、Host On-Demand クライアントのサイズは、ダウンロード・クライアント、Web Start クライアント、またはキャッシュ・クライアントのいずれの場合も、できるだけ小さくすることをお勧めします。これにより、ダウンロード時間を短縮し、クライアント・マシンのディスク・スペースを節約できます。

Host On-Demand クライアントのサイズを最小化するには、デプロイメント・ウィザードを使用して作成するのが最適な方法です。Host On-Demand に付属する事前定義クライアントは、Host On-Demand の全クライアント機能を含むことから、一般的にデプロイメント・ウィザードで作成されたカスタム・クライアントよりも大きくなります。デプロイメント・ウィザードで作成したクライアントには、あらかじめインストールするよう選択した機能のみが含まれます。また、デプロイメント・ウィザードで作成したクライアントは、圧縮形式でダウンロードされます。これにより、ダウンロード・サイズをさらに縮小できます。

デプロイメント・ウィザードを使用して、カスタマイズしたクライアントを作成する場合、デプロイメント・ウィザードの「プリロード・オプション」ウィンドウで、ユーザーが使用する機能のみを選択できます。例えば、ユーザーが 3270 端末および 3270 プリンター・セッションのみを使用する場合は、デプロイメント・ウィザードでクライアントを作成する際に、他のセッション・タイプは選択しないようにします。使用しないセッション・タイプのサポートを組み込むと、クライアントのサイズが増大しますが、機能が向上することはありません。

「プリロード・オプション」ウィンドウで「自動選択」をクリックした場合、デプロイメント・ウィザードは、セッション構成を基にして必要なコンポーネントを選択します。

また、頻繁に使用しない機能のコンポーネントをダウンロードしないよう選択できます。デプロイメント・ウィザードで機能を使用不可にするよう選択しないかぎ

り、ユーザーに何か別の機能が必要になった場合には、その機能を使用する際に必要なコンポーネントをダウンロードするよう指示されます。後で追加のセッション・タイプが必要な場合も、必ずしも新規クライアント・タイプを作成する必要はありません。その代わりに、「プリロード・オプション」ウィンドウのプリロード・リストに新しくセッション・タイプを追加できます。



Mac OS X では、初期のダウンロードの後に追加コンポーネントをインストールすることはできません。詳しくは、91 ページの『Mac OS X (Java クライアントのみ) の場合のキャッシュ・クライアントのサポート』を参照してください。

デプロイメント・ウィザードで生成したクライアントまたは事前定義クライアントのいずれでも、デバッグまたは問題判別を使用しないでください。これによりクライアントのサイズがかなり増加し、クライアントのパフォーマンスが低下します。デバッグおよび問題判別クライアントは、一般的な使用を目的とはしていません。これは、Host On-Demand テクニカル・サポートと一緒に、Host On-Demand システムの問題を診断および解決する場合にのみ使用します。

カスタマー提供の Java アーカイブおよび Java クラスのデプロイ

カスタマー提供の Java クラスおよび Java アーカイブは、Host On-Demand クライアントの一部としても、また Java ランタイム環境の一部としても組み込まれていない、Java クラス・ファイルおよび Java アーカイブ・ファイルです。そうしたファイルの例として、自分で実装したり、サード・パーティーから取得したりした Java クラスまたは Java アーカイブがあります。

以下の状況では、そうしたクラスまたはアーカイブをエミュレーター・クライアントで使用するためにデプロイしたいことでしょう。

- カスタマー提供の Java メソッドを呼び出すマクロをユーザーが実行できるようにしたい。
- カスタマー提供のアプレットをユーザーがセッションで実行できるようにしたい (セッションで自動的に開始するか、セッション・ウィンドウのメニュー上の「アクション (Actions)」>「アプレットの実行...」の選択を使用して起動する)。



カスタマー提供のアプレットを実行する際の Java の制限については、18 ページの『カスタマー提供のアプレットおよび Java に関する制限』を参照してください。

それらのファイルをデプロイするために選択可能ないくつかの方法がありますが、それぞれの方法は特定の状況下のみで行えます。以下の方法が考えられます。

- デプロイメント・ウィザードで `AdditionalArchives` HTML パラメーターを使用する。101 ページの『`AdditionalArchives` HTML パラメーターの使用』を参照してください。
- ファイルを Host On-Demand サーバーのパブリッシュ・ディレクトリーにコピーする。102 ページの『パブリッシュ・ディレクトリーからのデプロイ』を参照してください。

選択するデプロイメント方法は以下のものに依存します。

- デプロイされるファイルのタイプ (Java クラスおよび Java アーカイブ)

- デプロイされるファイルの場所 (Host On-Demand サーバーまたはクライアント・ワークステーション)
- クライアント・プラットフォームのタイプおよびブラウザのタイプ。

カスタマー提供の Java アーカイブおよび Java クラスをデプロイする場合に選択可能な 3 つの方法が、以下のセクションで説明されています。加えて、102 ページの『アーカイブ・ファイルについてのヒント』には、アーカイブ・ファイルの使用に関する詳しい情報があります。

AdditionalArchives HTML パラメーターの使用

Java アーカイブを Host On-Demand サーバーにデプロイする時にこの方法を使用できます。この方法は、キャッシュ・エミュレーター・クライアント、ダウンロード・エミュレーター・クライアント、および Web Start クライアントの場合に有効です。

Java アーカイブは、Java .JAR ファイルでなければなりません。

AdditionalArchives HTML パラメーターを使用する利点は、Host On-Demand サーバー上のキャッシュ・クライアントまたはダウンロード・クライアント HTML ファイルにユーザーの 1 人が接続する時に、ご利用の Java アーカイブが自動的にユーザーのワークステーションにダウンロードされるという点です。

この方法の欠点は、キャッシュ・クライアントまたはダウンロード・クライアントのどちらを使用しているかに関係なく、ユーザーがその HTML ファイルに接続するたびに、それらの Java アーカイブ・ファイルまたは Java クラス・ファイルが再度ダウンロードされるという点です。ユーザーが接続するたびにアーカイブをダウンロードする理由は、Host On-Demand クライアントが、最新のバージョンのアーカイブ・ファイルまたはクラス・ファイルを必ず持つようにするためです。結果として、Java アーカイブ・ファイルまたは Java クラス・ファイルが比較的少なくかつ比較的小さい時にこの方法は最も有効です。それは、それらのファイルをダウンロードするためにユーザーが長時間待つ必要がなく、それらのファイルをユーザーにダウンロードする時に、ご利用の Web サーバーに重い負荷をかけることがないからです。

この方法を使用するには、以下のステップを実行してください。

1. アーカイブを Host On-Demand パブリッシュ・ディレクトリーに置きます。
デフォルトのパブリッシュ・ディレクトリーは、ご利用の Host On-Demand サーバーのインストール・ディレクトリー内のサブディレクトリー HOD (c:\Program Files\IBM\HostOnDemand\HOD など) です。
2. デプロイメント・ウィザードで HTML ファイルを編集します。その後、次のようにします。
 - a. 「拡張オプション」パネルで、「HTML パラメーター」をクリックします。
 - b. 「名前」フィールドに、AdditionalArchives と入力します。
 - c. 「値」フィールドに、ファイル拡張子 (.jar) を付けずにコンマ区切りの Java アーカイブの名前を入力します。次に例を示します。
myCustomA,myCustomB,MyCustomC

詳しくは、オンライン・ヘルプの『AdditionalArchives』を参照してください。

パブリッシュ・ディレクトリーからのデプロイ

この方法は、以下の状況で有効です。

- Java クラス・ファイルを Host On-Demand サーバーにデプロイする場合。Java クラス・ファイルは、いずれの Host On-Demand パッケージにも属してはなりません。

この方法を使用するには、アーカイブを Host On-Demand パブリッシュ・ディレクトリーに置きます。デフォルトのパブリッシュ・ディレクトリーは、ご利用の Host On-Demand サーバーのインストール・ディレクトリー内のサブディレクトリー `HOD (c:\Program Files\IBM\HostOnDemand\HOD)` などです。

アーカイブ・ファイルについてのヒント

以下のヒントには、アーカイブ・ファイルの使用に関する役立つ情報が含まれています。

- アーカイブ (.jar) を作成する時には、各クラス・ファイルのパスが正確であることを確認してください。例えば、`com.mycompany.MyClass` のパスは、`com\mycompany\` です。それは `C:\MyTestDirectory\com\mycompany\` ではなく、空白でもありません (クラス・ファイルはパッケージの一部であるため)。
- ご使用のアーカイブ・ファイルに適切な許可が設定されていることを確認してください。すなわち、ファイル許可を使用するオペレーティング・システム (Linux、AIX、Unix、および z/OS など) では、アーカイブ・ファイルのファイル許可は 755 (すなわち、`rw-r--r--`) に設定する必要があります。
- 異なる AdditionalArchives パラメーターを指定する 2 つの異なるキャッシュ・クライアント・ページがある場合、あるページから別のページに切り替える時にブラウザーをクローズして再始動する必要があります。そうしないなら、あるページから別のページに切り替えた時に、キャッシュ・クライアントが再ロードされず、結果として、AdditionalArchives パラメーターがチェックされません。

第 11 章 Database On-Demand クライアントの使用

Database On-Demand クライアントは Java アプレットであり、エンド・ユーザーはこれを使用して SQL ステートメントや File Upload ステートメントを作成したり、これらのステートメントをリモート・データベース・サーバーに送信したり、SQL 照会 (SQL Select ステートメント) の結果をリモート・データベース・サーバーから取得したりすることができます。

適切な Java Database Connectivity (JDBC) ドライバーが Database On-Demand クライアント・ワークステーションにインストールされている場合には、ユーザーは、IBM System i サーバーや他のプラットフォーム上で稼働するデータベース・サーバーと通信できます。詳しくは、この資料の 106 ページの『JDBC ドライバーの入手とインストール』を参照してください。

Database On-Demand には以下の機能があります。

- SQL ステートメントおよび File Upload ステートメントを作成するためのテキスト・インターフェースおよびグラフィカル・インターフェース。
- SQL ステートメントおよび File Upload ステートメントを保管および再利用するための機能。
- SQL ステートメントの場合:
 - SQL ステートメントを実行して結果を表示するための機能。
 - SQL ステートメントの結果を、XML などのさまざまなファイル・フォーマットでファイルに保管するための機能 (本書の 107 ページの『データベース・アクセス用のファイル・フォーマット』を参照)。
- File Upload ステートメントの場合:
 - 作成、置換、追加、更新の各 File Upload タイプを使用するための機能。
 - XML などのさまざまなファイル・フォーマットでデータ・ファイルを読み取るための機能 (本書の 107 ページの『データベース・アクセス用のファイル・フォーマット』を参照)。

Database On-Demand クライアントは、3 つの定義済みクライアント HTML ファイルのいずれかを通してのみ使用可能です (105 ページの『Database On-Demand の定義済みクライアント』を参照)。デプロイメント・ウィザードを使用して Database On-Demand クライアントを作成することはできません。

ただし、Database On-Demand クライアントの代わりとして、Host On-Demand エミュレーション・クライアントおよびマクロのデータベース機能を使用できます (104 ページの『ディスプレイ・エミュレーション・クライアントおよびマクロにおけるデータベース機能』を参照)。

詳しくは、Host On-Demand オンライン・ヘルプの『データベース・アクセスの概要 (Overview of database access)』を参照してください。

Database On-Demand クライアントは、Java バージョンがあります。この場合、次のようになります。

- Java 対応のブラウザーを使用している場合は、Java バージョンの Database On-Demand クライアントが自動的に実行されます。

この Database On-Demand クライアントは、Java プラグインの拡張機能を利用できます。

ディスプレイ・エミュレーション・クライアントおよびマクロにおけるデータベース機能

Database On-Demand クライアントの代わりとして、Database On-Demand クライアントで使用可能な機能のほとんどすべては、以下のセッション・タイプを含むディスプレイ・エミュレーション・クライアントで使用できます。

- 3270 ディスプレイ・セッション
- 5250 ディスプレイ・セッション
- VT ディスプレイ・セッション

また、ディスプレイ・エミュレーション・クライアント・セッションでは、マクロ内の SQL ステートメントおよび File Upload ステートメントを使用することもできます (「マクロ・プログラミング・ガイド」の『SQLQuery アクション (SQLQuery action)』および『File Upload アクション (File Upload action)』を参照)。

例えば、3270 ディスプレイ・セッションのリモート・ホストに接続しているときには、マクロを起動して、3270 ディスプレイ・セッションのウィンドウから自動的にデータを読み取り、別のリモート・ホストに存在するデータベースのテーブルにデータを書き込むことができます。同様に、リモート・データベースのテーブルからデータを自動的に読み取って、3270 ディスプレイ・セッション・ウィンドウにデータを書き込むマクロを起動できます。

詳しくは、Host On-Demand オンライン・ヘルプの『データベース・アクセスの概要 (Overview of database access)』を参照してください。

Database On-Demand クライアントの開始

クライアント・ワークステーションで Database On-Demand クライアントを開始するには、次の 2 つの方法のいずれかを使用します。

- ブラウザーのアドレス・フィールドに定義済みの Database On-Demand HTML ファイルの URL を入力して (またはブラウザーをその URL に誘導するリンクをクリックして)、ブラウザーをその HTML ファイルに接続します。URL のフォーマットは次のとおりです。

`http://server_name/hod_alias/client_name.html`

`server_name` は Host On-Demand サーバーのホスト名または IP アドレス、`hod_alias` はパブリッシュ・ディレクトリーの別名、`client_name` は HTML ファイルの名前です。例えば、`www.myHODServer.com` が Host On-Demand サーバーであり、その `hod` がパブリッシュ・ディレクトリーの別名であるとする、Database On-Demand クライアントのダウンロード・バージョンに対する URL は、次のようになります。

`http://www.myHODServer.com/hod/HODDatabase.html`

- ブラウザーを IBM Host On-Demand クライアントの HTML ファイルに接続し、実行する Database On-Demand クライアントに対するリンクをクリックします。クライアント HTML ファイルの URL は次のとおりです。

`http://server_name/hod_alias/HODMain_xx.html`

`server_name` と `hod_alias` の意味は、最初の方法の場合と同じです。ファイルの名前 `HODMain_xx` において、`xx` は使用する言語を示す 2 文字のニーモニックです。例えば、英語の場合は、ファイル名は `HODMain_en.html` となり、完全な URL は次のようになります (サーバーと別名は前の例と同じものとします)。

`http://www.myHODServer.com/hod/HODMain_en.html`

Database On-Demand の定義済みクライアント

Database On-Demand クライアントは、3 つの定義済みクライアント HTML ファイルのいずれかを通して使用できます。デプロイメント・ウィザードを使用して、Database On-Demand クライアント HTML ファイルを作成することはできません。定義済みのクライアントについては以下で説明します。

Database On-Demand クライアント (HODDatabase.html)

これはダウンロード・クライアントです。ダウンロードという名前は、エンド・ユーザーが Database On-Demand クライアントを開始するたびに、すべてのクライアント・コードがクライアント・ワークステーションにダウンロードされることを意味します。

キャッシュされた Database On-Demand クライアント (HODDatabaseCached.html)

これはキャッシュ・クライアントです。「キャッシュ」という名前は、エンド・ユーザーが Database On-Demand を初めて開始した時点で大部分のクライアント・コードがダウンロードされて、クライアント・ワークステーションに保管されることを意味します。最初のダウンロードの後には、大部分のクライアント・コードがクライアント・ワークステーション上で使用できるため、キャッシュ・クライアントはダウンロード・クライアントよりはるかに速く開始します。キャッシュされた Database On-Demand クライアントのコンポーネントの多くは、キャッシュされた Host On-Demand クライアントと共通しています。



キャッシュ・クライアントでは、エンド・ユーザーが複数のコード・ページを必要とする場合には、各追加コード・ページに対するアーカイブ・ファイル (.jar ファイル) の名前を、定義済み HTML ファイルのプリロード・リストに追加する必要があります。コード・ページ言語および対応するファイル名については、107 ページの『Database On-Demand での複数コード・ページの使用』を参照してください。

キャッシュされた Database On-Demand クライアント (問題判別あり) (HODDatabaseCachedDebug.html)

これは、キャッシュ・クライアントに、セッション・イベントをロギングしてトレースするための問題判別コードが追加されたものです。



問題判別クライアントは、Host On-Demand インストール・システムにおいて問題が発生した際に、IBM サポートと一緒にその問題を解決する場合にだけ使用します。

ユーザー用の Database On-Demand の構成

ユーザー用に Database On-Demand を構成するには、以下のステップを実行します。

1. 管理ユーティリティーを使用して、グループとユーザーを定義する (Host On-Demand オンライン・ヘルプの『ユーザーとグループの管理 (Managing users and groups)』を参照)。
2. グループおよびユーザーが実行できるようにするデータベース機能を指定し、新しい SQL ステートメントおよび File Upload ステートメントの一部のデータベース・パラメーターに対するデフォルト値を指定する (Host On-Demand オンライン・ヘルプの『Database On-Demand のグループ/ユーザー・オプション (Database On-Demand Group/User Options)』を参照)。

ユーザーおよびグループのために定義済みの SQL ステートメントおよび File Upload ステートメントを作成する場合は、以下のステップを実行します。

1. エンド・ユーザーとして Database On-Demand クライアントを実行し、SQL ステートメントおよび File Upload ステートメントを作成する (Host On-Demand オンライン・ヘルプの『Database On-Demand 入門 (Getting started with Database On-Demand)』を参照)。
2. 管理ユーティリティーを起動し、SQL ステートメントおよび File Upload ステートメントを他のユーザーまたはグループにコピーする (Host On-Demand オンライン・ヘルプの『Database On-Demand のグループ/ユーザー・ステートメント (Database On-Demand Group/User Statements)』を参照)。

JDBC ドライバーの入手とインストール

リモート・ホストで稼働するデータベース・サーバーに接続するには、クライアント・ワークステーション上に Java Database Connectivity (JDBC) ドライバーがインストールされている必要があります。

Host On-Demand クライアントおよび Database On-Demand クライアントには、IBM AS/400 Toolbox for Java の JDBC ドライバーがすでに組み込まれています。このドライバーを使用すると、クライアントは、適切に構成された IBM System i または AS/400 のホスト・システム上の DB2/400 データベースにアクセスできます。このドライバーを登録またはデプロイする必要はありません。

別の JDBC ドライバーが必要な場合は、次のようにします。

1. リモート・データベースのベンダーまたは管理者に連絡して、JDBC ドライバーを入手する。
2. JDBC ドライバーを Host On-Demand または Database On-Demand に登録する。Host On-Demand オンライン・ヘルプの『JDBC ドライバーの登録 (Registering a JDBC driver)』を参照してください。

3. エンド・ユーザーのワークステーションに JDBC ドライバーをデプロイする。
Host On-Demand オンライン・ヘルプの『JDBC ドライバーのデプロイ
(Deploying a JDBC driver)』を参照してください。

データベース・アクセス用のファイル・フォーマット

エンド・ユーザーは、「SQL ウィザード (SQL Wizard)」ウィンドウの「出力 (Output)」タブまたは「File Upload (File Upload)」ウィンドウの「ファイル」タブで、SQL ステートメントまたは File Upload ステートメントのファイル・タイプを選択します。

ファイル・フォーマットについては、Host On-Demand オンライン・ヘルプの『データベース・アクセス用のファイル・フォーマット (File formats for database access)』を参照してください。

Database On-Demand での複数コード・ページの使用

Database On-Demand で複数のコード・ページを使用する場合は、HTML ファイルに jar ファイルまたは cab ファイルを追加する必要があります。HTML ファイルの言語に対応するコード・ページのみが自動的にロードされます。例えば、フランス語のコンピューターから実行していてオランダ語のホストを利用する場合は、以下の方法で変更を行う必要があります。

CommonJars.js ファイルを編集します。ダウンロード・クライアントを使用する場合は、先頭が『dbaDownloadJars =』の行を探して、次の表にある適切なファイル名を追加します。クライアントで Internet Explorer を使用する場合でも、jar ファイル名を使用してください (名前は後から cab ファイル名に変換されます)。キャッシュ・クライアントを使用する場合は、先頭が『dbaCachedComps =』の行を探して、次の表の適切なコンポーネント名を追加します。

サポートされる Database On-Demand コード・ページ

次の表は、サポートされる Database On-Demand クライアントのコード・ページ言語、対応する .jar ファイル名、およびキャッシュ・コンポーネント名のリストです。

コード・ページ言語	.JAR ファイル名	コンポーネント名
アラビア語	hacpar.jar	HACPAR
チェコ語、ハンガリー語、ポーランド語、スロベニア語	hacpce.jar	HACPCE
デンマーク語、フィンランド語、オランダ語、ノルウェー語、スウェーデン語	hacp1b.jar	HACP1B
ドイツ語、スペイン語、フランス語、イタリア語、ポルトガル語、ブラジル・ポルトガル語	hacp1a.jar	HACP1A
ギリシャ語	hacpgr.jar	HACPGR
ヘブライ語	hacphe.jar	HACPHE

日本語	hacpja.jar	HACPJA
韓国語	hacpko.jar	HACPKO
ロシア語	hacpru.jar	HACPRU
中国語 (簡体字)	hacpzh.jar	HACPZH
タイ語	hacpth.jar	HACPTH
トルコ語	hacptr.jar	HACPTR
中国語 (繁体字)	hacptw.jar	HACPTW

第 12 章 サーバー・マクロ・ライブラリーの作成およびデプロイ

サーバー・マクロ・ライブラリーは、HTML モデル・ページで、また構成モデル・ユーザーが使用できます。HTML ページの場合、ユーザーはデプロイメント・ウィザードを使用してサーバー・マクロ・ライブラリーをカスタマイズできます。構成モデルの場合、ユーザーは Host On-Demand 管理コンソールを使用できます。管理者は、GUI ベースの構成を使用して各セッション用の構成を行うことができます。定義されたすべてのセッション用に構成を行う管理者の方は、HTML パラメーター **SetServerMacroLibraryPath** を使用してください。

SetServerMacroLibraryPath の値は共有パス または相対パス です。これらの値を使用して、ユーザーがその Host On-Demand セッションからアクセスする、マクロの中央リポジトリを作成および保守できます。これらのマクロは、必要な場合にのみユーザーのマシンにダウンロードされます。サーバー・マクロに対して変更を行うと、ユーザーは次のマクロへのアクセス時に、自動的に更新を取得します。

サーバー・マクロ・ライブラリーには、次のようないくつかの利点があります。

- マクロを簡単に保管、編集、および管理できるようにし、それらすべてはアクセスしやすい場所から行えます。
- 複数ユーザーおよび任意の数のセッションに渡り、マクロを容易に共用できるようにします。
- マクロを Host On-Demand セッションにインポートする必要がなくなるため、セッションのサイズを削減できます。マクロは、ユーザーがそれらにアクセスする場合のみ、アクセス時にユーザーのマシンにダウンロードされます。
- Host On-Demand セッションを再生成したり、HTML ファイルを変更したりせずに、いつでもサーバー・マクロ・ライブラリー内のマクロを編集したり、ファイルを置き換えたりできます。行ったすべての変更は、次回にユーザーがそのマクロを要求する時に自動的に使用可能になります。

サーバー・マクロ・ライブラリーは、Web サーバー上または共有ネットワーク・ドライブ上に常駐させることが可能です。両方のタイプのライブラリーにおいて、特定の Host On-Demand セッションに対してどのマクロを有効にするかを制御できます。Web ベースのマクロ・ライブラリーを使用する場合、構成しているセッション用に有効にしたい特定のマクロを識別するテキスト・ファイルを作成する必要があります。共有ドライブ・ベースのマクロ・ライブラリーを使用する場合には、指定したディレクトリー内のすべての ファイルが、セッションに対して有効となります。ユーザーは、Web ベースのマクロ・ライブラリーに書き込むことはできませんが、書き込みアクセス権限を持っている場合は、共有ドライブ・ベースのマクロ・ライブラリーを更新することができます。

サーバー・マクロ・ライブラリーの Web サーバーへのデプロイ

1. マクロを、ユーザーが Web サーバーを経由してアクセスできる場所に置く。この場所は、Host On-Demand パブリッシュ・ディレクトリーでなくても構いません。
2. マクロの別個のセットを必要とするセッションごとに、マクロ・ファイル名のリストを含むテキスト・ファイルを作成する。テキスト・ファイルの形式は、各行に 1 つのマクロ・ファイル名だけを記述するというものです。

```
macro1.mac  
macro2.mac  
macro3.mac
```

必ず以下の規則に従ってください。

- マクロ名は、行の最初の要素でなければならない。最初の要素より後のものはすべて無視されるからです。
 - 行の最初の要素が `//` で始まる場合、行はコメントとみなされ、無視される。
 - テキスト・ファイルにリストするそれぞれのマクロは、`.mac` の拡張子を持つ必要がある。
3. このテキスト・ファイルを、それが参照するマクロと同じ場所に置く。
 4. デプロイメント・ウィザードで、「ホスト・セッション (Host Sessions)」ウィンドウの「構成」メニューをクリックし、「サーバー・マクロ・ライブラリー... (Server macro library...)」を選択する。「このセッションのサーバー・マクロ・ライブラリーを使用します (Use a server macro library for this session)」ボックスにチェックマークを付け、Web サーバー・マクロ・ライブラリーを選択します。
 5. ステップ 2 で作成したマクロ・リストの完全修飾 URL (例えば、`http://servername/hod/macrolist.txt`) を指定する。「OK」をクリックします。

ユーザーがセッションをオープンする時、「マクロの再生」または「使用可能マクロ」ウィンドウを使用して、ユーザーのセッション用に作成したリストで指定されたマクロを参照することができます。これらのマクロは、ユーザーがマクロの場所としてサーバー・ライブラリーを選択したときに使用可能になります。サーバー・ライブラリーの場所は、セッションでサーバー・マクロ・ライブラリーを使用する構成をした場合のみ、使用可能になります。

注: サーバー・マクロ・ライブラリーは、管理クライアントでも構成できます。

サーバー・マクロ・ライブラリーの共有ドライブへのデプロイ

1. マクロをネットワーク上の共用ディレクトリーに置く。
2. デプロイメント・ウィザードの「ホスト・セッション (Host Sessions)」ウィンドウで、構成するセッションを選択し、「構成」メニューをクリックし、「サーバー・マクロ・ライブラリー (Server macro library)」を選択する。「このセッションのサーバー・マクロ・ライブラリーを使用します (Use a server macro library for this session)」ボックスにチェックマークを付け、共用ドライブ・マクロ・ライブラリーを選択します。

3. ディレクトリー・パスを指定する。以下は、有効なディレクトリーの例です。

- 絶対パス。マップされたネットワーク・ドライブ名も絶対パスで使用できます。サーバー・マクロ・ライブラリーがローカル・ドライブをポイントしてはいけないことに注意してください。
- リモート・コンピューター名または IP アドレスは、ユーザーのコンピューターがすでにリモート側で接続されており、ディレクトリーを共用するコンピューターに認証済みである場合に許可されます。共用ドライブ・マクロ・ライブラリーへのパスの 2 つの例を以下に示します。
 - `¥¥your_host¥macro_library`。 *your_host* はホスト名、 *macro_library* はマクロ・ディレクトリーです。
 - `¥¥123.45.67.89¥macro_library`。 123.45.67.89 はホストの IP アドレス、 *macro_library* はマクロ・ディレクトリーです。

複数のセッションに対してマクロ・ライブラリーを構成しており、各セッションが独自のマクロのセットを使用する場合、セッションごとに別個のディレクトリーを作成する必要があります。

4. 「OK」をクリックします。

ユーザーがセッションをオープンする時、「マクロの再生」または「使用可能マクロ」ウィンドウを使用して、ディレクトリー内のマクロのリストを参照することができます。これらのマクロは、ユーザーがマクロの場所としてサーバー・ライブラリーを選択したときに使用可能になります。サーバー・ライブラリーの場所は、セッションでサーバー・マクロ・ライブラリーを使用する構成をした場合のみ、使用可能になります。

第 13 章 セッション・プロパティの動的変更

Host On-Demand セッションは、管理者によって定義され、ユーザーが Host On-Demand HTML ファイルにアクセスするときに Host On-Demand クライアントによって取得されます。ユーザーが参照するセッション・プロパティは固定された値で、管理者の初期構成および任意のユーザー更新の組み合わせで成り立っています。しかし、いくつかの HTML ファイルまたは特定のセッション・プロパティにおいて、HTML のアクセス時に値を動的に設定するほうが便利な場合もあります。このタイプの制御によって、クライアントの IP アドレスまたは時刻などの情報に基づいて特定のセッション・プロパティ値を設定することができます。

HTML へのアクセス時にセッション・プロパティを動的に設定するには、管理者は、Web サーバー上で稼働し、HTML をクライアントへ送信する直前に効率良く変更するプログラムを作成する必要があります。初期セッション・プロパティが HTML で定義されていなくても、Host On-Demand は HTML 内の多くのセッション・プロパティを指定変更する機能を提供します。これらの指定変更値は、常にクライアントによって使用され、管理者による初期セッション・プロパティ・セットアップおよびユーザーによるプロパティへの任意更新よりも優先されます。HTML 指定変更値は保管されることはないため、管理者が指定変更を除去すれば、クライアントは再びプロパティの変更前の設定を使用することになります。また、指定変更されたプロパティは、ユーザーが変更できないようにロックされます。

管理者がプログラムを作成し、HTML を指定変更することによって 1 つまたは複数のセッション・プロパティを動的に設定するには、Java Server Pages (JSP)、サーブレット、Perl、REXX、または Active Server Pages (ASP) などさまざまな方法があります。この章では、管理者が共通して利用できることに焦点を当てたいいくつかの例を取り上げます。これらの例は、特定のプロパティを指定変更する構文および技法を説明するよう意図されています。これらのメカニズムは、管理者が選択するどのプログラミング・アプローチにも適用します。

初期 HTML ファイルのセットアップ

初期 HTML ファイルは、デプロイメント・ウィザードを使用して作成する必要があります。これにより、ダウンロードされるコードのサイズやユーザーが使用できる機能など、使用する上で重要な機能を適切にセットアップできるようになります。以下のセクションでは、組み込む必要がある HTML パラメーターを説明します。しかし、これらのパラメーターに必要な正確なフォーマットは、HTML のフォーマットに応じて変わることを頭に入れておいてください。Host On-Demand 7 以降では、いくつかの HTML が JavaScript を使用して生成され、HTML パラメーターは JavaScript の配列内で指定されるか、または JavaScript の `document.write` ステートメントを使用して指定されることに注意してください。また、HTML のフォーマットは、選択されたクライアント (キャッシュ・クライアントまたはダウンロード・クライアント) によっても異なります。

コード・ベースの設定

デプロイメント・ウィザードを使用して HTML を作成する時にコード・ベースを設定するには、以下のようにします。

1. 「追加オプション」ウィンドウで、「拡張オプション」をクリックしてツリー・ビューの「その他」ブランチに進みます。
2. 「コード・ベース」フィールドに、相対パス `/hod/` を入力します。
3. HTML ファイルをデフォルトの Host On-Demand パブリッシュ・ディレクトリー `your_install_directory¥HOD` に保管します。

HTML ファイルはこれで Host On-Demand のアーカイブ・ファイルと同じディレクトリーに配置されました。

コード・ベースは、デプロイメント・ウィザード・ファイルが発行されるディレクトリーではなく、インストール済み Host On-Demand パブリッシュ・ディレクトリーを参照します。「コード・ベース」フィールドに、完全修飾 URL を入力することはできますが、セッション・プロパティを動的に変更する時には、デフォルトのパブリッシュ・ディレクトリーの相対パス `/hod/` を入力することを強くお勧めします。完全修飾 URL を入力した場合、DNS エントリーが同じ IP アドレスに解決されたとしても、コード・ベースとして指定した方法とは異なる方法でホスト名を指定するすべてのユーザーは、ファイルにアクセスできなくなります。

ConfigBase パラメーターの追加

ConfigBase と呼ばれるパラメーターを HTML ファイルに追加します。『コード・ベースの設定』で説明されている Codebase としての `/hod/` の定義と同様に、ConfigBase パラメーターが必要です。これは、最終的には JSP ファイルをデフォルトのパブリッシュ・ディレクトリーとは異なる場所にデプロイすることになり、Host On-Demand のアプレットは `hostondemand/HOD/HODData` ディレクトリーに存在するセッション構成ファイルを見つける方法を知る必要があるためです。これらのファイルは、デプロイメント・ウィザードの HTML ファイルをパブリッシュ・ディレクトリーに保管した時点で作成されます。Codebase とは異なり、ConfigBase パラメーターでは完全修飾 URL が必要です。ConfigBase は、Host On-Demand に固有の用語です。



詳しくは、『WebSphere 拡張機能を含む JavaServer ページ・ファイルの開発 (Developing JavaServer Pages files with WebSphere extensions)』を参照してください。

HTML パラメーターの指定変更

セッション・プロパティを動的に設定するために従う必要のあるステップがいくつかあります (この章の後半に示される例は、これらのいくつかのパラメーターの指定方法を明確に理解するのに役立ちます)。

1. **HTML** 指定変更を使用可能にする。デフォルトでは、クライアントは HTML 指定変更を無視します。指定変更を使用可能にするには、`EnableHTMLOverrides` という HTML パラメーターを定義し、その値に `true` を設定する必要があります。

2. 指定変更されるセッションをリストする。HTML に関連付けられたセッションが複数ある場合があるため、どのセッションが指定変更されるかをリストする必要があります。TargetedSessionList という HTML パラメーターを定義し、その値に指定変更を受け入れるセッションの名前を正確に指定してください。この値は、"Session1Name, Session2Name" などのように、コンマで区切られたセッション名のリストでなければなりません。
3. 指定変更自体を指定する。指定変更されるセッション・プロパティごとに、希望する指定変更である値を使用して、そのプロパティ名を名前に持つ HTML パラメーターを定義する必要があります。そして、指定する値は、TargetedSessionList パラメーターでリストされたすべてのセッションに適用されます。TargetedSessionList のセッションの一部のみを指定変更する場合は、値を "Session1Name=value1, Session2Name=value2" などのフォーマットで指定できます。

指定変更可能な特定のセッション・プロパティ

以下の表に、指定変更可能なセッション・プロパティを説明し、パラメーターごとの許容値を示します。

表 12. 指定変更可能なセッション・プロパティ

パラメーター名	説明	有効な値
Host	ターゲット・サーバーのホスト名または IP アドレス。プロパティ・パネルに「宛先アドレス」として表示される。すべてのセッション・タイプに適用。	ホスト名または IP アドレス。
HostBackup1	backup1 サーバーのホスト名または IP アドレス。プロパティ・パネルに backup1 の「宛先アドレス」として表示される。すべてのセッション・タイプに適用。	ホスト名または IP アドレス。
HostBackup2	backup2 サーバーのホスト名または IP アドレス。プロパティ・パネルに backup2 の「宛先アドレス」として表示される。すべてのセッション・タイプに適用。	ホスト名または IP アドレス。
Port	ターゲット・サーバーが listen しているポート番号。プロパティ・パネルに「宛先ポート」として表示される。すべてのセッション・タイプに適用。	任意の有効 TCP/IP ポート番号。

表 12. 指定変更可能なセッション・プロパティ（続き）

パラメーター名	説明	有効な値
PortBackup1	backup1 サーバーが listen しているポート番号。プロパティ・パネルに backup1 の「宛先ポート」として表示される。すべてのセッション・タイプに適用。	任意の有効 TCP/IP ポート番号。
PortBackup2	backup2 サーバーが listen しているポート番号。プロパティ・パネルに backup2 の「宛先ポート」として表示される。すべてのセッション・タイプに適用。	任意の有効 TCP/IP ポート番号。
CodePage	セッションが接続する先のサーバーのコード・ページ。プロパティ・パネルに「ホスト・コード・ページ」として表示される。FTP を除くすべてのセッション・タイプに適用。	セッション・プロパティ・パネルにリストされるサポート済みホスト・コード・ページの数値の部分 (037 など)。
SessionID	このセッションへ割り当てるショート・ネーム (OIA 内に表示される)。この構成に固有である必要がある。プロパティ・パネルに「セッション ID」として表示される。すべてのセッション・タイプに適用。	1 文字: A-Z。
LUName	ターゲット・サーバーで定義され、このセッションを接続させる先の LU または LU プールの名前。プロパティ・パネルに「LU またはプール名」として表示される。3270 ディスプレイおよび 3270 印刷装置の各セッション・タイプに適用。	LU または LU プールの名前。

表 12. 指定変更可能なセッション・プロパティ（続き）

パラメーター名	説明	有効な値
LUNameBackup1	backup1 サーバーで定義され、このセッションを接続させる先の LU または LU プールの名前。プロパティ・パネルに backup1 の「LU またはプール名」として表示される。3270 ディスプレイおよび 3270 印刷装置の各セッション・タイプに適用。	LU または LU プールの名前。
LUNameBackup2	backup2 サーバーで定義され、このセッションを接続させる先の LU または LU プールの名前。プロパティ・パネルに backup2 の「LU またはプール名」として表示される。3270 ディスプレイおよび 3270 印刷装置の各セッション・タイプに適用。	LU または LU プールの名前。
WorkstationID	このワークステーションの名前。プロパティ・パネルに「ワークステーション ID」として表示される。5250 ディスプレイおよび 5250 印刷装置の各セッション・タイプに適用。	このワークステーションの固有名。

表 12. 指定変更可能なセッション・プロパティ (続き)

パラメーター名	説明	有効な値
ScreenSize	画面上の行数および列数を定義する。プロパティ・パネルに「画面サイズ」として表示される。3270 ディスプレイ、5250 ディスプレイ、および VT ディスプレイの各セッション・タイプに適用。	• 値=行 x 列 • 2=24x80 (3270、5250、VT) • 3=32x80 (3270) • 4=43x80 (3270) • 5=27x132 (3270、5250) • 6=24x132 (VT) • 7=36x80 (VT) • 8=36x132 (VT) • 9=48x80 (VT) • 10=48x132 (VT) • 11=72x80 (VT) • 12=72x132 (VT) • 13=144x80 (VT) • 14=144x132 (VT) • 15=25x80 (VT) • 16=25x132 (VT)
SLPScope	Service Location Protocol (SLP) スコープ。プロパティ・パネルの「SLP オプション」の下の「スコープ」として表示される。3270 ディスプレイ、3270 印刷装置、5250 ディスプレイ、および 5250 印刷装置の各セッション・タイプに適用。	このフィールドの正確な値は、担当の管理者に連絡して確認してください。
SLPAS400Name	セッションを特定の IBM System i へ接続させる。プロパティ・パネルには「iSeries 名 (SLP)」と表示される。5250 ディスプレイおよび 5250 印刷装置の各セッション・タイプに適用。	完全修飾 SNA CP 名 (USIBMNM.RAS400B など)。
FTPUser	FTP サーバーへ接続するときにセッションが使用するユーザー ID を指定する。プロパティ・パネルに「ユーザー ID」として表示される。FTP セッション・タイプに適用。	有効なユーザー ID。

表 12. 指定変更可能なセッション・プロパティ (続き)

パラメーター名	説明	有効な値
FTPPassword	FTP サーバーへ接続するときにセッションが使用するパスワードを指定する。プロパティ・パネルに「パスワード」として表示される。FTP セッション・タイプに適用。	有効なパスワード。
UseFTPAnonymousLogon	ユーザー ID に無名を使用して FTP サーバーへログインするセッションを使用可能にする。プロパティ・パネルに「無名ログイン」として表示される。FTP セッション・タイプに適用。	Yes または No
FTPEmailAddress	無名ログインを使用中に FTP サーバーへ接続するときに使用する E メール・アドレスを指定する。プロパティ・パネルに「E メール・アドレス」として表示される。FTP セッション・タイプに適用。	有効な E メール・アドレス。
PromptForDestinationAddress	FTP サーバーへ接続するときに使用する宛先アドレスのプロンプトをユーザーに対して出すかどうかを指定する。プロパティ・パネルに「宛先アドレス」として表示される。FTP セッション・タイプに適用。	yes または no
CICSInitialTransEnabled	CICS ゲートウェイ・セッションが確立された時に初期トランザクションが開始されるようにする。	true または false

表 12. 指定変更可能なセッション・プロパティ (続き)

パラメーター名	説明	有効な値
CICSInitialTrans	CICS ホストへの接続時に開始される初期トランザクションの名前を指定する。CICS ゲートウェイ・セッションに限り適用。 CICSInitialTransEnabled パラメーターは、開始される指定されたトランザクションに対して true を設定する必要があります。	有効なトランザクション ID は、1 から 128 文字のストリングです。このストリングは、サーバーへの接続時に実行される初期トランザクションおよびすべてのパラメーターを識別します。最初の 4 文字またはストリング内の最初のブランクまでの文字は、トランザクションとして取られます。残ったデータは、トランザクションが呼び出される時にトランザクションに渡されます。
Netname	インストールまたは予約する端末リソースの名前。このフィールドがブランクの場合、選択される端末タイプは予測不可能です。CICS セッションに限り適用。	有効な端末リソース名。

HTML パラメーターの処理中に生じたエラーは、すべて Java コンソール内に表示されます。

例 1: クライアントの IP アドレスに基づいて LU 名を指定変更する

管理者は、セッション定義に直接 LU 名を指定しないようにすることができます。この例では、クライアントの IP アドレスを使用して、テキスト・ファイル内にリストされた LU 名を検索し、それをセッション内の指定変更値として使用する簡単な方法を示します。

この例は、JSP を使用して作成されています。デプロイメント・ウィザードが、3270 ディスプレイおよび 5250 ディスプレイと呼ばれる 2 つのセッションを含む HTML ファイルを作成するために使用されました。Host On-Demand 7 以降では、いくつかの HTML が JavaScript を使用して生成され、HTML パラメーターは JavaScript の配列内で指定されるか、または JavaScript の document.write ステートメントを使用して指定されることに注意してください。また、HTML のフォーマットは、選択されたクライアント (キャッシュ・クライアントまたはダウンロード・クライアント) によっても異なります。

この例では、開始位置としてキャッシュ Java ページを使用しており、HTML の指定変更に対する必要な変更は太字で表示してあります。デプロイメント・ウィザードがキャッシュ Java2 ページを生成するために使用される時、それは以下のファイルを生成します。

- Example1.html
- z_Example1.html

- Example_J2.html

Macintosh クライアントは、Example_J2.html ページを利用します。

IP アドレス / LU 名のペアを含むファイル (c:\luname.table) が読み取られます。クライアントの IP アドレスが、「3270 ディスプレイ」セッションで指定変更される正しい LU 名を検索するために使用されます。詳細は、例に含まれるコメントを参照してください。デプロイメント・ウィザードの出力へ追加した行は、太字で表示してあります。

```
<!doctype html public "-//W3C//DTD HTML 3.2 Final//EN">
<%
// Read the luname.table file into a properties variable.
// The luname.table file contains lines in the following format:
//   ipaddress=luname
Properties lunames = new Properties();
lunames.load(new FileInputStream("c:%luname.table"));
%>
<HTML>
<HEAD>
<META http-equiv="content-type" content="text/html; charset=UTF-8">
<!-- TITLE Begin -->
<TITLE>Example1 page title</TITLE>
<!-- TITLE End -->
<!-- SUMMARY Begin -->
<!--
Configuration Model
What configuration model would you like to use?
-HTML-based model
Host Sessions
-3270 Display
-5250 Display
Additional Options
-Cached = Cached client
-Java Type = java2
Disable Functions
Preload Options
-5250 Sessions = True
-Change Session Properties = True
-3270 Sessions = True
Cached Client/Web Start Options
Basic Options
-Debug = False
-Height (in pixels) = 250
-Width (in pixels) = 550
Upgrade Options
-Percent of users who can upgrade by default = 100
-Prompt user (user decides foreground or background)
Advanced Options
HTML parameters
-None
Code base
- /hod/
HTML templates
-Default
Problem determination
-Debug = False
User updates
-Persist user updates? = True
Appearance
-Standard Host On-Demand Client
Applet size
-Autosize to browser
Session Manager API
-Enable Session Manager JavaScript API = False
Server connection
Language
-Locale = Use the system Locale
Maximum sessions
- 26
-->
<!-- SUMMARY End -->
</HEAD>

<BODY BACKGROUND="/hod/hodbkngd.gif">
<CENTER>
```

```

<IMG src="/hod/hodlogo.gif" ALT="hodlogo.gif">
<P>

<SCRIPT LANGUAGE="JavaScript">
function writeAppletParameters()
{
    return "";
}
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript" SRC="/hod/HODVersion.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CommonJars.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CommonParms.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CommonJ2Parms.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
var db = parent.location;
var hod_Locale = '';
var hod_AppName = '';
var hod_AppHgt = '340';
var hod_AppWid = '550';
var hod_CodeBase = '/hod/';
var hod_Comps = 'HABASE;HODBASE;HODIMG;HACP;HAFNTIB;HAFNTAP;HA3270;HODCFG;HA5250';
var hod_Archs = 'habasen.jar,hodbasen.jar,hodimg.jar,hacp.jar,hafntib.jar,hafntap.jar,
                ha3270n.jar,hodcfgn.jar,ha5250n.jar';

var hod_URL = new String(window.location);
var hod_DebugOn = false;

// put cached client installation applet parameters here
var hHod_AppletParams = new Array;
hHod_AppletParams[0] = '<PARAM NAME="DebugCachedClient" VALUE="false">';
hHod_AppletParams[1] = '<PARAM NAME="ShowDocument" VALUE="_parent">';
hHod_AppletParams[2] = '<PARAM NAME="CachedClient" VALUE="true">';
hHod_AppletParams[3] = '<PARAM NAME="ParameterFile" VALUE="HODData¥¥Example1¥¥params.txt">';
hHod_AppletParams[4] = '<PARAM NAME="JavaScriptAPI" VALUE="false">';
hHod_AppletParams[5] = '<PARAM NAME="BookmarkPage" VALUE="Example1.html">';

// The next 2 lines are required in order to override session properties.
// The first line turns on the processing for this function and does not
// need to be modified. The second line identifies the sessions that you
// want to change. In this example, there are 2 sessions identified
// named: "3270 Display" and "5250 Display".

hHod_AppletParams[6]='<PARAM NAME="EnableHTMLOverrides" VALUE="true">';
hHod_AppletParams[7]='<PARAM NAME="TargetedSessionList" VALUE="3270 Display,5250 Display">';

// The following line changes the LUName session parameter for the session named
// "3270 Display". In this example, the LUName is being set to the value
// contained in the c:¥luname.table for the IP address of the client.
// When you are initially testing your changes, you may want to use a constant
// value to verify that the syntax is correct before you insert your
// calculations.
hHod_AppletParams[8]='<PARAM NAME="Luname" VALUE="3270
                        Display=<%=lunames.get(request.getRemoteAddr())%>">';

//hHod_AppletParams[x] = '<PARAM NAME="DebugCode" VALUE="65535">';

var pg = buildJ2Page(db);
pg += writeAppletParameters();
pg += '</APPLET>';
if(hod_DebugOn) alert('J2 page complete, result = ¥n' + pg);
document.write(pg);
</SCRIPT>

</CENTER>
</BODY>
</HTML>

```

例 2: HTML のフォームを使用してユーザーが接続先ホストを指定できるようにする

管理者は、指定変更値を算出するのではなく、HTML のフォームを使用して値を指定することもできます。以下の例は、ホスト名の記入用の簡単なフォームを表示するものです。このフォームは、フォーム内に指定されたホスト名を使用して 3270 ディスプレイ・セッションのホスト名を指定変更する JSP プログラムへ post を実行します。

この例は、JSP を使用して作成されています。デプロイメント・ウィザードが、「3270 ディスプレイ」および「5250 ディスプレイ」と呼ばれる 2 つのセッションを含む HTML ファイルを作成するために使用されました。Host On-Demand 7 以降では、いくつかの HTML が JavaScript を使用して生成され、HTML パラメーターは JavaScript の配列内で指定されるか、または JavaScript の `document.write` ステートメントを使用して指定されることに注意してください。また、HTML のフォーマットは、選択されたクライアント (キャッシュ・クライアントまたはダウンロード・クライアント) によっても異なります。

フォームを使用する場合、フォーム・データはプログラムへの複数の要求を通じて保持される必要があります。これは、構成サーバー・ベース・モデルのページを使用している場合、Host On-Demand HTML ファイルが、Java を検出するために、およびブックマークをサポートするために自ら再ロードするからです。構成サーバー・ベース・モデルを使用している場合に、Java 1 を選択し、ブックマーク・サポートを使用不可にしている場合は、ページの再ロードの必要はなく、フォーム・データを保持する必要もありません。この例は、再ロードが行われてもフォーム・データを保持するために、JSP セッションを使用します。

以下に、ホスト名を入力できる簡単な HTML フォームを示します。このフォームは、JSP プログラム (example2.jsp) に対して post を実行します。

```
<form method="POST" action="hod/example2.jsp">
Hostname <input name="form.hostname"><br>
<input type="submit">
</form>
```

以下は、デプロイメント・ウィザードからの変更出力です。詳細は、例に含まれるコメントを参照してください。デプロイメント・ウィザードの出力へ追加した行は、太字で表示してあります。

```
<HTML>
<%
// Get a session or create if necessary and store the hostname
// entered in the form in the session.
HttpSession session = request.getSession(true);
String hostname = request.getParameter("form.hostname");
if (hostname!=null) {
    session.putValue("session.hostname", hostname);
}
%>
<!-- HOD WIZARD HTML -->
<!-- Deployment Wizard Build : 8.0.0-B20030605 -->
<HEAD>
<META http-equiv="content-type" content="text/html; charset=UTF-8">
<TITLE>Example 2 page title</TITLE>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CommonJars.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/HODJavaDetect.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/hod/CommonParms.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript">

//---- Start JavaScript variable declarations ----//
```

```

var hod_Locale = '';
var hod_jsapi=false;
var hod_AppName = '';
var hod_AppHgt = '80%';
var hod_AppWid = '80%';
var hod_CodeBase = '/hod/';
var hod_FinalFile = 'z_example2.html';
var hod_JavaType = 'java2';
var hod_Obplet = '';
var hod_jars = 'habasen.jar,hodbasen.jar,hodimg.jar,hacp.jar,hodsignn.jar,ha3270n.jar,
               hodcfgn.jar,ha5250n.jar';

var hod_URL = new String(window.location);
var hod_DebugOn = false;
var hod_SearchArg = window.location.search.substring(1);

var hod_AppletParams = new Array;
hod_AppletParams[0] = '<PARAM NAME="ParameterFile" VALUE="HODData¥¥example2¥¥params.txt">';
hod_AppletParams[1] = '<PARAM NAME="ShowDocument" VALUE="parent">';
hod_AppletParams[2] = '<PARAM NAME="JavaScriptAPI" VALUE="1" + hod_jsapi + ">';
hod_AppletParams[3] = '<PARAM NAME="PreloadComponentList" VALUE="HABASE;HODBASE;HODIMG;
                        HACP;HAFNTIB;HAFNTAP;
                        HA3270;HODCFG;HA5250">';

// The next 2 lines are required in order to override session properties.
// The first line turns on the processing for this function and does not
// need to be modified. The second line identifies the sessions that you
// want to change. In this example, there are 2 sessions identified
// named: "3270 Display" and "5250 Display".
// Be careful to increment the array index correctly.

hod_AppletParams[4] = '<PARAM NAME="EnableHTMLOverrides" VALUE="true">';
hod_AppletParams[5] = '<PARAM NAME="TargetedSessionList" VALUE="3270 Display,5250 Display">';

// The following line changes the Host or Destination Address session parameter
// for the session named "3270 Display". In this example, the Host is being set
// to the value saved in the JSP session from the HTML form.
// When you are initially testing your changes, you may want to use a constant
// value to verify that the syntax is correct before you insert your
// calculations.
// Here we override the host for the 3270 session to the value saved in the
// jsp session from the html form.

hod_AppletParams[6] = '<PARAM NAME="Host" VALUE="3270
Display=<%=session.getValue("session.hostname")%>">';

//hod_AppletParams[x] = '<PARAM NAME="DebugCode" VALUE="65535">';

//---- End JavaScript variable declarations ----//

function getHODMsg(msgNum) {
    return HODFrame.hodMsgs[msgNum];
}

function getHODFrame() {
    return HODFrame;
}

var lang = detectLanguage(hod_Locale);
document.writeln('<FRAMESET cols="*,10" border=0 FRAMEBORDER="0">');
document.writeln('<FRAME src="/hod/hoddetect_' + lang + '.html" name="HODFrame">');
document.writeln('</FRAMESET>');

</SCRIPT>
</HEAD>
</HTML>

```

第 14 章 zSeries での Host On-Demand の構成

この章では、zSeries システムに Host On-Demand を構成するために、読み取り/書き込みプライベート・ディレクトリーとパブリッシュ・ディレクトリーを別にセットアップする方法について説明します。

この構成シナリオの目的は、zSeries に共通の構成作業を解説することです。

別個の読み取り/書き込みプライベートおよびパブリッシュ・ディレクトリーのセットアップ

Host On-Demand プライベート・ディレクトリー用に別個のファイル・システムをセットアップ

Host On-Demand をインストールすると、製造リフレッシュ・リリースによってだけでなく、実行環境の `/usr/lpp/HOD/hostondemand/private` ディレクトリーのファイルも更新されます。このディレクトリーは Host On-Demand ソフトウェアの実行中に更新されるようになったため、別個の (非サービス) ファイル・システムをマウントすることをお勧めします。これは、次のいずれかの方法で実行できます。

- 現行プライベート・ディレクトリー・ロケーション (`/usr/lpp/HOD/hostondemand/private` など) 上に別個のファイル・システムをマウントする。
- 以下のように、プライベート・ディレクトリー・ロケーションへのシンボリック・リンクを作成する。
 1. TSO MKDIR を実行して、`/etc/HOD/private` などの異なるマウント・ポイントを作成する。
 2. 元のプライベート・ディレクトリーを名前変更するか、バックアップして削除する。
 3. 予期されたロケーション `/usr/lpp/HOD/hostondemand/private` から実際のロケーション `/etc/HOD/private` を指すシンボリック・リンクを作成する。以下のリンク・コマンドを使用します。

```
ln -s /etc/HOD/private /usr/lpp/HOD/hostondemand/private
```

LDAP およびネイティブ認証を使用している場合は、HODrapd および `/keys` ディレクトリーを、システム特定の `/private` ディレクトリーへ手動でコピーします。

システム特定の `/private` ディレクトリーは、マウントされるとオーバーレイしますが、マスターの `/private` ディレクトリーを壊すことはありません。保守リリースが適用されるときは、マスターの `/private` ディレクトリーを使用します。これらのファイルが変更された場合は、これらをシステム固有の `/private` ディレクトリーにコピーします。

別個のユーザー・パブリッシュ・ディレクトリーのセットアップ

デプロイメント・ウィザードで生成されるファイルは、Host On-Demand のパブリッシュ・ディレクトリーとは別のユーザー定義ディレクトリー内に置くことができ

ます。これによって、今後の Host On-Demand アップグレードの適用が容易になります。このソリューションにより、Host On-Demand パブリッシュ・ディレクトリーは読み取り専用のまま保持され、デプロイメント・ウィザードのファイルを展開するために別個の書き込み可能ロケーションが提供されます。

別個のユーザー・パブリッシュ・ディレクトリーにデプロイメント・ウィザード・ファイルをデプロイする手順について、およびパブリッシュ・ディレクトリー外に配置可能なその他のユーザー変更ファイルに関する情報については、デプロイメント・ウィザードのマイグレーション手順を参照してください。

ユーザー定義パブリッシュ・ディレクトリー用に別個のファイル・システムを作成およびマウントすることができます。生成されたデプロイメント・ウィザード zip ファイルはこのディレクトリーに転送され、DWUnzip ユーティリティによって unzip されます。Web サーバーは、ユーザー定義パブリッシュ・ディレクトリーに固有の別名ステートメントを組み込む必要があります。

ユーザー定義パブリッシュ・ディレクトリーの別名を指定する URL を使用してページにアクセスすることができます。例えば、パブリッシュ・ディレクトリーが /usr/lpp/HOD/publish であり、別名が userpublish である場合、クライアント・ページにアクセスするための URL は `http://<servername>/userpublish/<pagename>.html` となります。

z/OS でのマイグレーション考慮事項

旧レベルの Host On-Demand から Host On-Demand V13.0 にアップグレードを行うときは、以前のカスタマイズについて考慮する必要があります。HOD V13.0 のインストール後に、以前に定義したすべてのグループ/ユーザーおよびセッションに関して旧プライベート・ディレクトリーを新規プライベート・ディレクトリーにコピーします。次に `pax` または `tar` コマンドを使用して、既存のプライベート・ディレクトリーを HOD V13.0 ファイル・システムにコピーします。『プライベート・ディレクトリーのバックアップ』を参照してください。

デプロイメント・ウィザードで作成された以前のクライアントに関しては、デプロイメント・ウィザードを Windows コンピューターにインストールする必要があります。続いて、クライアントを編集し、HOD V12.0 サーバーに再デプロイします。127 ページの『デプロイメント・ウィザードを z/OS サーバーからインストール』を参照してください。

プライベート・ディレクトリーのバックアップ

プライベート・ディレクトリーは、`pax` コマンドまたは `tar` コマンドのいずれかを使用してバックアップできます。HOD V11 の現行プライベート・ディレクトリーが `/usr/lpp/HOD/hostondemand/private` であるとしします。

1. Host On-Demand V11 ファイル・システムから、ディレクトリーをプライベート・ディレクトリー `cd /usr/lpp/HOD/hostondemand/private` に変更します。
2. プライベート・ディレクトリーを `/tmp` ディレクトリーにアーカイブします。
-z オプションはファイルを圧縮し、-v はアーカイブするファイルおよびサブディレクトリーのリストを提供します (オプション): `pax -wzof /tmp/private.pax.Z *`。

3. `private.tar.Z` ファイルを、Host On-Demand V12 のシステム上の `/tmp` ディレクトリーにコピーします (異なるシステム上にある場合)。
4. Host On-Demand V12.0 HFS で、ディレクトリーを、ファイルを解凍するプライベート・ディレクトリーに変更します: `cd /usr/lpp/HOD/hostondemand/private`。
5. `pax` コマンドを発行して `private.pax.Z` ファイルを解凍します。 `-z` オプションは圧縮ファイルを指定し、`-v` は解凍するファイルおよびサブディレクトリーのリストを提供します (オプション): `pax -rzvf /tmp/private.pax.Z`。

デプロイメント・ウィザードを z/OS サーバーからインストール

デプロイメント・ウィザードは通常、製品のインストール時には Windows マシン上にあります。z/OS では、z/OS HOD サーバー用にクライアント・ページを生成できるように、Windows 上にデプロイメント・ウィザードをインストールするためのダウンロードが提供されます。デプロイメント・ウィザードの z/OS サーバーからのインストールについては、以下の手順を参照してください。

1. FTP をバイナリーで使用して、Windows ワークステーションのファイル `/usr/lpp/HOD/hostondemand/HOD/depwiz/DW.zip` を再配置します。
2. `zip` ファイルをフォルダーに解凍します。
3. インストールを始めるには、エクスプローラーで `<folder>%DeploymentWizard%disk1` に移動します。
4. `imLauncherWindows.bat` をダブルクリックして、Installation Manager ユーザー・インターフェースを起動します。
5. インストールの終了手順に従います。

デプロイメント・ウィザードがインストールされたら、これを起動できます。「スタート」 > 「すべてのプログラム」 > 「IBM Host On-Demand デプロイメント・ウィザード」の順に選択します。

第 15 章 IBM System i での Host On-Demand の構成

IBM System i プラットフォームに Host On-Demand をインストールしたら、以下のようにしてソフトウェアを構成します。

- サービス・マネージャーをセットアップするには、『IBM System i 上での Host On-Demand サービス・マネージャーの構成、開始および停止』の指示に従ってください。
- IBM System i システムでデプロイメント・ウィザードを使用するには、132 ページの『IBM System i でのデプロイメント・ウィザードの使用』の指示に従ってください。
- セキュリティーを構成するには、133 ページの『セキュア接続のための IBM System i サーバーの構成』の指示に従ってください。
- コード化文字セット ID を使用したユニコード・サポートの要件を知るには、137 ページの『i/OS および OS/400 でのユニコード・サポート』を参照してください。

IBM System i 上での Host On-Demand サービス・マネージャーの構成、開始および停止

以下のコマンドは、IBM iv7r1 または OS/400 のコマンド行から使用できます。

構成

サービス・マネージャーを構成するには、NCServiceManager-OS400.sh スクリプト・ファイルを使用できます。NCServiceManager-OS400.sh は、IBM System i のディレクトリー

```
HOD_install_directory>/lib/samples/NCServiceManager/
```

にあります。

サービス・マネージャー設定を構成するには、以下の作業を行います。

1. ディレクトリー `<HOD_install_directory>/lib/samples/NCServiceManager/` にアクセスします。`<HOD_install_directory>` は、Host On-Demand がインストールされている場所またはパスです。例えば、`/QIBM/ProdData/HostOnDemand/` です。
2. NCServiceManager-OS400.sh ファイルを開きます。
3. ランタイム変数が正しく、ご使用の環境に適合していることを確認します。ランタイム変数のデフォルト値がご使用の環境に適合しない場合は、これらのデフォルト値を変更してください。ランタイム変数には以下のようなものがあります。
 - JRE の場所: `JAVA_ENGINE`

`JAVA_ENGINE` の値を、システムにインストールされた `jre` の完全パスまたは場所に更新します。Java V6 以上であることが必要です。この値は、Java インストール・ディレクトリーの `<java_installation>/bin/java` を指していなければなりません。

- サーバー上の Host On-Demand パブリッシュ・ディレクトリーの場所:
MY_HOD_DIRECTORY

MY_HOD_DIRECTORY の値が HostOnDemand インストール・ディレクトリーの完全パスに設定されていることを確認し、必要であれば更新します。これは Host On-Demand のインストール・ディレクトリーでなければなりません。このディレクトリーには /bin、/lib、およびその他の Host On-Demand フォルダーが含まれます。一般的に、この値はインストール時に一度更新されます。例えば、/QIBM/ProdData/HostOnDemand です。

- コマンド・ファイル内で指定されたターゲット・パス:
MY_PUBLISHED_DIRECTORY

MY_PUBLISHED_DIRECTORY の値が Host On-Demand パブリッシュ・ディレクトリーの完全パスに設定されていることを確認し、必要であれば更新します。一般的に、これは <HOD_Installation>/HOD ディレクトリーです。<HOD_Installation> は Host On-Demand インストール・ディレクトリーです。

4. NCSERVICEManager-OS400.sh が必要な実行権限を持ち、サーバー上の Host On-Demand インストール済み環境にあるディレクトリーへの書き込みが許可されていることを確認します。

開始

Host On-Demand サービス・マネージャーを開始するには、このサービス・マネージャーがバックグラウンドで開始され、実行が継続されるように、NCSERVICEManager-OS400.sh を実行します。

IBM i Series でこれを行う 1 つの方法として、IBM PASE for System i を呼び出すことによってジョブを実行依頼し、スクリプトを実行する方法があります。お客様の i Series セットアップおよび要件に適した最良のジョブの実行依頼方法について詳しくは、IBM i Series 管理者にお問い合わせください。

ジョブを実行依頼するコマンドの例:

```
sbmjob cmd(call pgm(qp2shell) parm('/QOpenSys/usr/bin/-sh'
'/QIBM/ProdData/HostOnDemand/lib/samples/NCSERVICEManager/NCSERVICEManager-OS400.sh'))
```

停止

サービス・マネージャーを停止するには、Iseries 上でジョブを終了します。サービスを停止するための適切な方法について詳しくは、Iseries 管理者にお問い合わせください。

これを行う方法の 1 つを、以下の例で説明します。

1. **WRKACTJOB** と入力して、アクティブ・ジョブのリストを開きます。
2. 「アクティブ・ジョブの処理 (**Work with Active Jobs**)」メニューに、機能名 **JVM-NCSERVICEM** で Host On-Demand サービス・マネージャー・ジョブがリストされます。このジョブ項目までメニューをスクロールダウンし、「処理 **..(Work With..)**」オプション (通常はオプション 5) を選択します。

3. 「ジョブの終了 (End job)」オプションを選択します。これを行うには、41 を入力してジョブを終了し、**Enter** ボタンを押します。これにより、サービス・マネージャー・ジョブが終了し、サービス・マネージャーは停止します。

HOD サーバー状況での作業

サービス・マネージャーが実行中であるかどうかを判別するには、Java プログラム `NCServiceManager` (スクリプト `NCServiceManager-OS400.sh` で開始) が実行されているかどうかを確認する必要があります。このため、サーバー状況を確認する方法は、サービス・マネージャーを開始した方法によって異なる場合があります。

上記の例では、サービス・マネージャーは `NCServiceManager-OS400.sh` スクリプトを実行するジョブを実行依頼することによって開始されています。したがって、以下の 2 つの方法で状況を確認できます。

1. `WRKACTJOB` コマンドを使用して状況を確認します。

- a. 次のコマンドを入力します。

```
WRKACTJOB
```

これにより、アクティブ・ジョブのリストが表示されます。

- b. 「アクティブ・ジョブの処理 (Work with Active Jobs)」メニューに、Host On-Demand サービス・マネージャー・ジョブが機能名 **JVM-NCServiceM** でリストされます。 **PageDown** ボタンまたは **PageUp** ボタンを使用して、このジョブ項目までメニューをスクロールダウンし、ジョブを「処理 **..(Work with..)**」するための適切なオプション番号 (通常はオプション 5) を入力します。

- c. メニュー・オプションを使用してジョブ状況を確認します。

2. コマンド行でプロセス状況を照会します。

130 ページの『開始』の例では、`SBMJOB` コマンドで IBM PASE for System i (qp2shell) を呼び出すことによってスクリプト `NCServiceManager-OS400.sh` が実行されます。したがって、この場合は、以下の手順も状況の確認に役立ちます。

1. IBM System i で、グリーン・スクリーン・コマンド行にサインオンします。
2. PASE シェル環境に入ります。グリーン・スクリーン・コマンド行で、コマンド `call qp2term`

を入力します。

3. PASE シェルで、コマンド

```
ps -ef | grep NCServ
```

を入力します。

注: `NCServiceManager` は、サービス・マネージャーを実行する Java プログラムの名前です。

コマンドは、サービス・マネージャーが稼働していることを検出すると、以下のような出力を提供します。

```
$
> ps -ef | grep NCServ
kushald 3146    1    0 15:23:30      -  0:00 /QIBM/ProdData/OS400/Java400/jFr
omPASE java -classpath .:sm.zip:ibmjndi.jar:jndi.jar:jsdk.jar:ods.jar:jt400.j
ar -Djava.net.preferIPv4Stack=true -DFIPS=on com.ibm.eNetwork.HODUtil.service
s.admin.NCServiceManager /QIBM/ProdData/HostOnDemand
$
```

注: PASE シェルでは大/小文字が区別されます。そのため、コマンドでアルファベットの正しい大/小文字を保つことが重要です。

証明書管理

証明書管理機能は、Host On-Demand に用意されている P12Keyring ユーティリティを使用して実行することができます。このユーティリティは、SSL 鍵リング・データベースを簡単に作成およびデプロイするための方法を提供します。Host On-Demand 鍵リングのいずれかにある SSL 証明書を処理するには、このオプションを使用します。SSL 関連のセッションに関する一般情報については、21 ページの『第 4 章 セキュリティの計画』を参照してください。

P12Keyring とその使用法に関する情報が『付録 C. P12 鍵リング・ユーティリティ』にあります。

いくつかのコマンドの例を、リンク「How to create, add or convert certificates to CustomizedCAs.p12 file on z/OS for Host On-Demand」で参照できます。

情報バンドラーの開始

IBM サポート・センターに支援を要請することが必要な場合は、すでに使用可能になっている「情報バンドラー」スクリプト・ファイルを使用して、Host On-Demand 構成に関する情報を収集することができます。

使用法に関する情報については、HOD V10 資料のセクション『情報バンドラーの実行』を参照してください。

HOD プリンター定義テーブルの作成

Host On-Demand 3270 プリンター・セッション用のカスタム・プリンター定義テーブルを作成します。この機能の使用については、『iSeries サーバーでの PDT のコンパイル』にあるセクションを参照してください。

カスタム・プリンター定義が必要になるのは、特殊な用紙を使用する場合や、そのプリンターがサポートされていない場合などです。以下のオプションは HOD V13.0 では使用できません。

IBM System i でのデプロイメント・ウィザードの使用

IBM System i ベース Host On-Demand サーバーに画面を展開するために、デプロイメント・ウィザードを使用するには、次のようにします。

1. Windows ワークステーションから、ネットワーク・ドライブを、Host On-Demand サーバーである IBM System i システム上の /qibm ディレクトリにマップします。詳しくは、IBM System i Web サイトを参照してください。
2. デプロイメント・ウィザードのインストール・イメージを、インストール済みの Host On-Demand v12 サーバーからダウンロードします。 HODMain.html (例えば <http://hodserver.name.com/hod/HODMain.html>) にアクセスし、「**Windows 版デプロイメント・ウィザード・インストール・イメージ (Deployment Wizard Installation Image for Windows)**」をクリックします。
3. デプロイメント・ウィザードをインストールするためのインストール手順を参照します。これは Host On-Demand サーバー全体をインストールしなくても実行できます。
4. カスタム機能および選択を設計します。
5. カスタマイズした HTML ファイルをマップ済みネットワーク・ドライブ (例えば、`y:¥ProdData¥hostondemand¥hod¥myweb` など) に保管します。
6. ブラウザーを使用して、ファイル (例えば <http://iSeries.name.com/hod/myweb.html>) をテストします。

セキュア接続のための IBM System i サーバーの構成

自己署名証明書または既知リストにない署名機関からの証明書を使用している場合は、P12Keyring ユーティリティを使用して CustomizedCAs 鍵リングを構成します。詳しくは、『付録 C. P12 鍵リング・ユーティリティ』を参照してください。

CustomizedCAs 鍵リングを構成するには、以下の手順に従ってください。

1. Java がシステムにインストールされていることを確認します。
2. UNIX または AIX ベースのコマンド行を開きます。これは、例えば QShell や IBM I PASE シェルなどです。
3. Host On-Demand インストール・ディレクトリ内の Host On-Demand パブリッシュ・フォルダーにナビゲートします。一般的に、これは `/QIBM/ProdData/HostOnDemand/HOD/` です。
4. コマンド

```
java -classpath .:your_install_dir/lib/sm.zip com.ibm.hod5ssligh.tools.P12Keyring  
CustomizedCAs connect myServer.raleigh.ibm.com:702
```

を入力します。このコマンドが完了するまで数分かかる場合があります。パスワードの入力を求められた場合は、`hod` と入力して **Enter** を押します。

5. 鍵リングに追加する認証局 (CA) に対応した証明書番号を選択します。サイト証明書ではなく、CA 証明書を追加するようにしてください。ポートが応答していない場合は、「セキュア接続のための IBM i 7.1 サーバーの構成」を参照してください。
6. それぞれのターゲット・サーバーごとにステップ 3 ~ 5 を繰り返してください。

CustomizedCAs 鍵リングの内容を表示するには、以下の手順を実行します。

1. Java がシステムにインストールされていることを確認します。

2. linux ベースのシェル (例えば QSHELL や IBM i PASE シェルなど) を開きます。
3. Host On-Demand インストール・ディレクトリー内の Host On-Demand パブリッシュ・フォルダーにナビゲートします。一般的に、これは `/QIBM/ProdData/HostOnDemand/HOD/` です。
4. コマンド

```
java -classpath .: your_install_dir/lib/sm.zip com.ibm.hod5ssligh.tools.P12Keyring CustomizedCAs list
```

を入力します。

i/OS および OS/400 での TLS を使用する Host On-Demand のインストールと構成

次に示すリストは、TLS を使用する Host On-Demand をインストールして構成するために必要なステップの概要です。

1. ソフトウェアとハードウェアに関するすべての要件が満たされていることを確認する。
2. すべての必要な IBM System i ソフトウェア・プロダクトをインストールする。詳しくは、IBM System i の資料を参照してください。
3. すべての必要な PTF をインストールする。最新の PTF は、IBM eServer System i サポート・ポータルにあります。
4. IBM HTTP Server または IBM Application Server をインストールして構成する。詳しくは製品資料を参照してください。
5. IBM Administrative Server 上の Digital Certificate Manager から認証局 (CA) を作成するか、または公開されている CA を購入する。詳しくは、IBM System i の資料を参照してください。
6. IBM HTTP Server または IBM Application Server に TLS を構成する。詳しくは製品資料を参照してください。
7. TLS を使用するように Host On Demand を構成する。詳しくは、オンライン・ヘルプの『TLS の構成 (Configuring TLS)』を参照してください。

セキュア接続のための Telnet サーバーの構成

TLS を有効にするために必要な手順については、IBM System i Knowledge Center にアクセスし、TLS について検索してください。セキュア接続を使用したい IBM System i7 システムごとに、この手順を繰り返すことをお勧めします。

Host On-Demand の CustomizedCAs 鍵リングの構成

自己署名証明書または既知リストにない署名機関からの証明書を使用している場合は、P12Keyring ユーティリティを使用して CustomizedCAs 鍵リングを構成します。詳しくは、『付録 C. P12 鍵リング・ユーティリティ』を参照してください。

CustomizedCAs 鍵リングを構成するには、以下の手順を実行します。

1. Java がシステムにインストールされていることを確認します。

2. linux ベースのシェル (例えば QSHELL や IBM I PASE シェルなど) を開きます。
3. Host On-Demand インストール・ディレクトリー内の Host On-Demand パブリッシュ・フォルダーにナビゲートします。一般的に、これは `/QIBM/ProdData/HostOnDemand/HOD/` です。
4. コマンド

```
java -classpath .:your_install_dir/lib/sm.zip com.ibm.hod5ssligh.tools.P12Keyring CustomizedCAs connect myServer.raleigh.ibm.com:702
```

を入力します。このコマンドが完了するまで数分かかる場合があります。パスワードの入力を求めるプロンプトが出された場合は、`hod` と入力して **Enter** を押します。

5. 鍵リングに追加する認証局 (CA) に対応した証明書番号を選択します。サイト証明書ではなく、CA 証明書を追加するようにしてください。ポートが応答していない場合は、「セキュア接続のための IBM System i サーバーの構成」を参照してください。
6. それぞれのターゲット・サーバーごとにステップ 3 ~ 5 を繰り返してください。

CustomizedCAs 鍵リングの内容を表示するには、以下のようになります。

1. Java がシステムにインストールされていることを確認します。
2. linux ベースのシェル (例えば QSHELL や IBM I PASE シェルなど) を開きます。
3. Host On-Demand インストール・ディレクトリー内の Host On-Demand パブリッシュ・フォルダーにナビゲートします。一般的に、これは `/QIBM/ProdData/HostOnDemand/HOD/` です。
4. コマンド

```
java -classpath .: your_install_dir/lib/sm.zip com.ibm.hod5ssligh.tools.P12Keyring CustomizedCAs list
```

を入力します。



複数の IBM System i マシンがあり、すべてのマシンが使用できる単一証明書を作成したい場合は、クロス認証を考慮してください。クロス認証について詳しくは、『セキュリティーの管理』、『Cryptographic Services APIs』、および『Application System/400 Cryptographic Support/400 Version 3』を参照してください。

クライアント認証

補足的なセキュリティーについては、インターネットを介してユーザーのシステムに Telnet でログインできる人を緊密に制御するためにクライアント認証を持つ TLS を考慮してください。例えば、クライアント証明書がユーザーの IBM System i (例えばデジタル証明書マネージャーを介して) によって発行された場合にのみ認証を許可するように、Telnet サーバーを構成できます。

クライアント証明書には、有効期間の制限があります (例えば、90 日)。証明書の有効期限が切れたら、ユーザーは継続するためにクライアント証明書ダウンロード・プロセスを実行しなければなりません。このプロセスでは有効な IBM System i のユーザー ID とパスワードが必要です。



すべての Telnet クライアント・ソフトウェアがクライアント認証の機能に対応しているわけではありません。使用可能であれば、IBM System i についてのすべての TLS 可能 Telnet 接続はユーザー証明書を必要とします。

詳しくは、IBM System i Web サイトを参照してください。

セキュア接続用の Host On-Demand OS/400 プロキシの構成

OS/400 プロキシを構成して、ファイル転送と Database On-Demand 接続を暗号化することができます。これを行うには、それぞれのターゲット IBM System i に次の追加ソフトウェアがインストールされていなければなりません。

- IBM Cryptographic Access Provider
- IBM Client Encryption
- Host Servers
- デジタル証明書マネージャー

TLS ユーザー許可のセットアップ

ファイルに対するユーザーの許可を制御する必要があります。TLS リーガル責任を満たすために、そのファイルに対するユーザー・アクセスを制御する TLS ファイルを含むディレクトリーの権限を変更しなければなりません。権限を変更するには、次のようにします。

1. コマンド `wrklnk '/QIBM/ProdData/HTTP/Public/jt400/*'` を入力します。
2. ディレクトリーでオプション 9 を選択します。
 - a. *PUBLIC が *EXCLUDE 権限を持っていることを確認してください。
 - b. TLS ファイルにアクセスする必要のあるユーザーにディレクトリーに対する *RX 権限を与えます。個々のユーザーまたはユーザーのグループを許可することができます。*ALLOBJ 特殊権限を持っているユーザーは TLS ファイルへのアクセスを拒否できないことに注意してください。

セキュア Web サービス

Host On-Demand サーバーはブラウザーにプログラム・オブジェクトをダウンロードするために Web サーバーを使用します。この情報は暗号化できますが、かなりパフォーマンスに影響します。

セキュア Web サービスのデフォルトのポートは 443 です。そのポートが使用可能ではない場合は、ポート 80 が使用されます。セキュア Web サービスを使用可能にするには、次のステップを実行してください。

1. Web ブラウザーで `http://<server.name>:2001` と入力する (<server.name> は、使用する IBM System i の TCP/IP ホスト名)。接続できない場合は、次の i/OS および OS/400 のコマンドを使用して、HTTP サーバーを開始してください。

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

2. i/OS または OS/400 のユーザー・プロファイルとパスワードを (プロンプトに応じて) 入力する。残りの構成アクティビティーを完了するには、*ALLOBJ および *SECADM 権限が必要です。
3. 「IBM HTTP Server for AS/400」をクリックする。
4. 「構成と管理 (Configuration and Administration)」をクリックする。
5. 「構成」をクリックする。
6. リストから「CONFIG」構成を選択する。
7. 「セキュリティ構成 (Security Configuration)」をクリックする。
8. 「HTTP 接続の許可 (Allow HTTP connections)」と「TLS 接続の許可 (Allow TLS connections)」選択項目で以下のようにする。
 - ポート番号 (443)
 - 「TLS クライアント認証なし」を選択します。
 - 「適用」を選択します。
9. 画面左下の「AS/400 タスク (AS/400 Tasks)」ボタンをクリックする。
10. 「デジタル証明書マネージャー (Digital Certificate Manager)」をクリックする。
11. 「システム証明書 (System Certificates)」をクリックする。
12. 「セキュア・アプリケーションでの作業 (Work with Secure Applications)」をクリックする。
13. 「QIBM_HTTP_SERVER_CONFIG」をクリックした後、「システム証明書での作業 (Work with System Certificate)」をクリックする。
14. 「新規証明書の割り当て (Assign New Certificate)」をクリックする。
15. 次の i/OS および OS/400 のコマンドを使用して、管理 HTTP サーバー・インスタンスを終了する。

```
ENDTCPSVR SERVER(*HTTP) HTTPSVR(DEFAULT)
```
16. HTTP インスタンスがシャットダウンするまで 10 秒待つ。
17. 次の i/OS および OS/400 のコマンドを使用して、管理 HTTP サーバー・インスタンスを開始する。

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(DEFAULT)
```
18. Web ブラウザーで `https://server.name/hod/hodmain.html` と入力する (`server.name` は使用する IBM System i の TCP/IP ホスト名)。

IBM System i の各種トピックについて詳しくは、『IBM i PDF ファイルおよびマニュアル』を参照してください。

i/OS および OS/400 でのユニコード・サポート

一般情報

5250 ディスプレイのセッションにおいて、Host On-Demand は、コード化文字セット ID (CCSID) のタグが付けられたフィールドに置かれるユニコード・データの表示をサポートします。詳しくは、『Unicode support for i/OS and OS/400 using Coded Character Set Identifiers』を参照してください。

ホスト・プログラミング情報

ホスト・プログラミングについては、IBM System i Web サイト を参照してください。

第 16 章 WebSphere Portal での Host On-Demand のデプロイ

ユーザーは、HTML ファイルを使用して Host On-Demand にアクセスする代わりに WebSphere Portal のコンポーネントである Portal Server を使用して、これにアクセスすることができます。Portal Server は、ポートレットと呼ばれるコンテンツ拡張機能を Web サイトに組み込むためのフレームワークを提供します。ポートレットは、Portal Server 内で実行されるアプリケーションです。ポートレットは、さまざまなソース (Web サイト、E メール、およびビジネス・アプリケーションなど) からコンテンツを編成し、それを単一の HTML ファイルとしてブラウザ・ウィンドウに表示します。デプロイメント・ウィザードにより生成される Host On-Demand セッションを起動するための WAR ファイルは、ポートレットとしてデプロイすることができ、ユーザーはポータル・インターフェースを通して Host On-Demand にアクセスできます。Host On-Demand および Portal Server をファイアウォールと共に使用する場合は、33 ページの『ファイアウォールでの Host On-Demand の使用』を参照してください。また、ユーザーのポータル ID や Portal Server クリデンシアル・ポルトなどの WebSphere Portal のセキュリティ機能を使用する場合は、『Web Express Logon 解説書』を参照してください。

Host On-Demand ポートレットを実行するには、Host On-Demand と Portal Server の両方がインストールされている必要があります。

Host On-Demand による Portal Server の処理方法

図 8 は、Host On-Demand による Portal Server の処理方法を示しています。

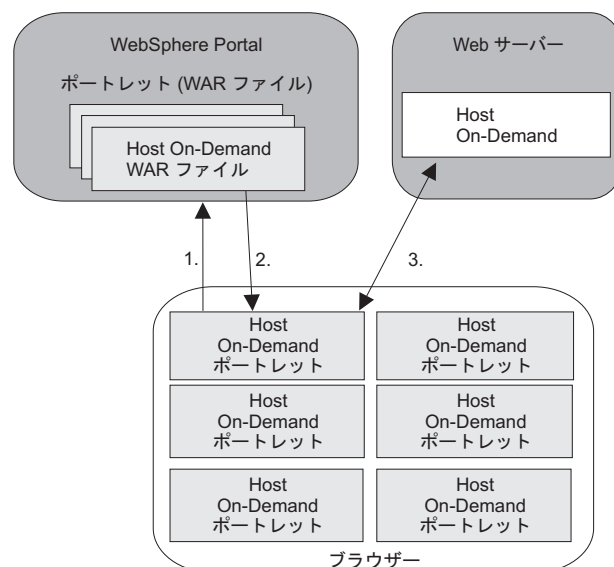


図 8. Host On-Demand による Portal Server の処理方法

1. ユーザーがブラウザーからポータルにログインし、ユーザー ID とパスワードによる認証を受けます。
2. ユーザーがカスタマイズしたポートレット・セットがユーザーのマシンにダウンロードされ、ブラウザーに表示されます。
3. ユーザーが Host On-Demand ポートレットを構成していれば、Host On-Demand が始動します。この場合、セッションの開始や他の Host On-Demand タスクの実行など、ユーザーはポートレット・ウィンドウ内でのすべての Host On-Demand の機能を利用できます。

Portal Server での Host On-Demand クライアントの使用

Portal Server で Host On-Demand を使用するには、Host On-Demand ポートレットが必要になります。デプロイメント・ウィザードを使用すると、独自のカスタム・ポートレットを迅速かつ容易に作成することができます。ポートレットの作成についての詳細は、デプロイメント・ウィザードのオンライン・ヘルプを参照してください。また、Host On-Demand Service Key の「Tools and Utilities」にある「Host On-Demand manufacturing refresh」ページから、サンプル Host On-Demand ポートレットをダウンロードすることもできます。

カスタム・ポートレットを作成するかサンプル・ポートレットを入手したら、他のポートレットと同様、それらを Portal Server へ直接インポートすることができます。詳しくは、『IBM WebSphere Portal Express V8 documentation』を参照してください。

ポートレットによる Host On-Demand へのアクセスに関する制限

Portal 環境は Host On-Demand のすべての機能をサポートしていますが、以下の制限があります。

- Host On-Demand はクライアント・ブラウザーとして Mac OS をサポートしていますが、これらのブラウザーは Portal 環境では推奨されません。サポートされるブラウザーについて詳しくは、『IBM WebSphere Portal Express V8 documentation』を参照してください。
- 1 つの WebSphere Portal ページで複数のポートレットを実行するときは、以下の点に注意してください。
 - HTML ベースの構成モデルを使用する。
 - キャッシュ・クライアントとしてポートレットを構成するときは、Java を使用する。
 - ポートレットは、ダウンロード・クライアントまたはキャッシュ・クライアントのどちらか一方として構成し、両方の混合としては構成しない。
- 独立したウィンドウで実行するように構成され、AssociateEmbeddedMenuBar パラメーターが false に設定されているセッションに対し、Java 対応ブラウザーを使用すると、3270 および 5250 ホスト・セッションのメニューは、ポップアップ・メニューとして表示されます。Host Print セッションおよび FTP セッションの場合は、デフォルトでポップアップ・メニューは表示されません。Host Print セッションまたは FTP セッションに対してメニューを表示するには、別のウィンドウで開始するようにセッションを構成する必要があります。

- 別のウィンドウで実行しないように構成されている Host On-Demand セッションにメニュー・バーを組み込むには、Java 対応のブラウザを使用し、AssociateEmbeddedMenuBar パラメーターを true (デフォルト) に設定する必要があります。次の環境では、3270、5250、VT、CICS の各ホスト・セッションに対するメニュー・バーは、ポップアップ・メニューとして表示され、セッションには組み込まれません。
 - クライアント・ブラウザが Java 対応で、AssociateEmbeddedMenuBar パラメーターが false に設定されている。

Host On-Demand セッションが別のウィンドウで開始するように構成されている場合は、メニュー・バーは常にセッション・ウィンドウに関連付けられるので、ポップアップ・メニューとして表示することはできません。

- ポートレットが Host On-Demand でキャッシングを行う場合 (デプロイメント・ウィザードで構成された場合)、ポートレットへのアクセスに使用するそれぞれのマシンで、Host On-Demand クライアントがキャッシュに入ります。
- Host On-Demand のブックマークは、ポータル環境では機能しません。
- デプロイメント・ウィザードでアプレット・サイズを構成しない場合は、デフォルトの固定サイズ「ミディアム」に設定されます。
- Host On-Demand ポートレットが稼働していると、`java.io.FileNotFoundException` のような警告メッセージが Java コンソールに表示される場合があります。このメッセージは、単一のポータル・ページで複数の Host On-Demand ポートレットを実行できるようにするために Host On-Demand ポートレットが使用している、ダミーのアーカイブ・ファイル名が原因で発生します。これらのメッセージは、ポートレットのパフォーマンスに影響を与えないので、無視してかまいません。

Host On-Demand ポートレット使用時の特殊な考慮事項

Portal Server で Host On-Demand を使用する際、以下の問題を考慮するようにします。

- ユーザーが **Portal Server** からログアウトする際の **Host On-Demand** セッション。Host On-Demand はユーザーのマシン上でアプレットとして実行されています。そのため、ユーザーが Portal Server からログアウトしてもそのことを認識しません。セッションが独立したウィンドウで稼働している場合 (デフォルト)、Host On-Demand セッションは、ユーザーがセッションをクローズするか、ブラウザをクローズするまで持続します。Host On-Demand セッションが Portal Server ウィンドウ内に組み込まれて稼働している場合、ユーザーが Portal Server からログアウトするとセッションが終了するように見えますが、接続はブラウザ・ウィンドウがクローズされるまで持続します。Portal Server からログアウトする場合には、ブラウザ・ウィンドウをクローズすることを強くお勧めします。また、セッションに対して非アクティブ・セッションのタイムアウトを構成することもできます。
- 非アクティブ・セッションのタイムアウト。デフォルトでは、Host On-Demand はセッション接続に対してタイムアウトを強制しません。しかし、ポートレットを実行する際には、リソースの消費を抑えるために非アクティブのセッションのタイムアウトを構成することが望まれます。非アクティブ・セッションのタイムアウトは、3270 ディスプレイおよびプリンターのセッション、

5250 ディスプレイおよびプリンターのセッション、および VT を含むほとんどのエミュレーター・タイプに対して設定することができます。タイムアウトのパラメーター「非アクティブ・セッションのタイムアウト (Session Inactivity Timeout)」は、セッションの「プロパティー」にある「接続」ウィンドウで、これらすべてのセッションに対して、分単位で設定および使用可能にすることができます。

- 異なるサーバーへの **WebSphere Portal** および **Host On-Demand** のインストール。WebSphere Portal と Host On-Demand を異なるサーバーにインストールすると、特定のブラウザにおいて、Host On-Demand ポートレットへのアクセス時にセキュリティ違反が発生することがあります。この問題が起こる理由は、Host On-Demand の機能が Java (Host On-Demand サーバー側) と JavaScript (WebSphere Portal 側) の間の対話に負うところが非常に大きいため、一部のブラウザでは両者が異なるサーバー上にあるというだけで、対話できなくなります。1 つの解決策としては、プロキシを使用することで、ブラウザに対して WebSphere Portal と Host On-Demand が同じサーバーにあるように見せかける方法があります。Apache/IBM HTTP サーバーでプロキシをセットアップするステップの例を以下に示します。

- Host On-Demand ポートレットの「HOD サーバー URL」(hodCodeBase) が、WebSphere Portal の存在するホストを指すように構成する。コンテキスト・ルートの /hod/ を指定します (例えば、http://portal.company.com/hod)。
- httpd.conf 内の LoadModule proxy_module で始まる行をアンコメントする (# を取り除く)。
- httpd.conf に ProxyPass 規則を追加して、HOD サーバーの URL 要求を実際の Host On-Demand サーバーの要求に変換する (例えば、ProxyPass/hod/ http://hod.company.com/hod/)。
- Web サーバーを再始動する。

•

これでクライアントのブラウザはポータルと同じホストから Host On-Demand ファイルを要求するようになりますが、これらの要求は Web サーバーによって Host On-Demand システムの実際のロケーションへ内部的に転送されます。

- キャッシングの使用/不使用。デプロイメント・ウィザードのデフォルトでは、各ユーザーのマシンで Host On-Demand をキャッシュに入れるよう設定されます。多くの場合、Host On-Demand にこのオプションを指定することが望まれます。これを指定すれば、必要なすべてのコードがユーザーのマシンに効果的にインストールされ、HTML ファイルやポートレットにアクセスするたびにネットワークに負荷をかけずに済むからです。しかし、キャッシング動作は多くの Portal Server ユーザーにとって不慣れなものである可能性もあり、キャッシング・オプションを使用しないようにすることもできます。
- デプロイメント・ウィザードのモデルの選択。ポートレットにどのモデル (構成サーバー、HTML、結合) を選択するかによって、セッションが構成される場所や、ユーザーによる変更の保管方法が変わってきます。Host On-Demand ではポートレットを HTML ファイルと同じように扱いますが、ポートレットの構成方法を決定する際には以下の特性を考慮に入れてください。

- HTML モデル: Host On-Demand ポートレットに対しては、この構成モデルが推奨されます。このモデルは、Host On-Demand の構成サーバーに依存しません。ユーザーが更新を行うことを許可されている場合、行われた更新は、ユーザーのローカル・マシンではなく、WebSphere Portal 構成の一部として保管されます。これにより、ユーザーは、マシン間を移動した場合でも、この更新情報にアクセスできます。



ポートレットおよびポートレットにアクセスする Web ページに対する適切なアクセスをユーザーに許可した場合にのみ、ユーザー設定は WebSphere Portal に保管されます。WebSphere Portal のユーザーには Privileged User、Editor、Manager、または Administrator のアクセス権が必要です。ユーザーにアクセスを許可する方法については、WebSphere Portal の資料を参照してください。

- 構成サーバー・ベース・モデル: このモデルでは、ユーザーは Host On-Demand 構成サーバーにアクセスする必要があります。ユーザーは、マシン間を移動しても、自分が行ったセッションへの変更を使用することができます。ただし、ユーザーは、Host On-Demand 構成サーバーと WebSphere Portal の両方に対して、認証を受ける必要があります。
- 結合モデル: このモデルでは、ユーザーには、初期セッション構成を取得するために、Host On-Demand 構成サーバーへのアクセス権が必要です。ユーザーの変更は、ローカルではなく WebSphere Portal の構成の一部として保管されるので、ユーザーは、マシン間を移動しても、自分が行ったセッションへの変更を使用することができます。ただし、ユーザーは、Host On-Demand 構成サーバーと WebSphere Portal の両方に対して、認証を受ける必要があります。



ポートレットおよびポートレットにアクセスする Web ページに対する適切なアクセスをユーザーに許可した場合にのみ、ユーザー設定は WebSphere Portal に保管されます。WebSphere Portal V5 のユーザーには Privileged User、Editor、Manager、または Administrator のアクセス権が必要です。ユーザーにアクセスを許可する方法については、WebSphere Portal の資料を参照してください。

- 追加パラメーターの構成。Host On-Demand ポートレットの使用時には、以下の追加パラメーターを構成し、ポータル・ページを望ましい外観にすることができます。
 - 「自動的に開始する (Start Automatically)」: セッション・プロパティの「設定 (Preferences)」>「開始オプション (Start Options)」ウィンドウでこのオプションを「はい」に設定すると、Host On-Demand ポートレットを自動的に開始することができます。
 - 「別のウィンドウで開始する (Start in Separate Window)」: セッション・プロパティの「設定 (Preferences)」>「開始オプション (Start Options)」ウィンドウでこのオプションを「いいえ」に設定すると、Host On-Demand ポートレットを埋め込みポートレットとして表示することができます。
 - 「開始時に HOD デスクトップを表示しない (Hide HOD Desktop at Startup)」: 「拡張オプション」>「外観」ウィンドウでこのオプションを選択すると、Host On-Demand デスクトップは表示されません。

- **Portal Server** での固有のポートレット名の指定。 デプロイメント・ウィザードで、「ファイル名と出力形式」ページの「ページ・タイトル」を使用して、Portal Server 内での固有のポートレット名を指定します。

Host On-Demand ポートレットの拡張

特定の状況においては、Host On-Demand ポートレットの外観や機能を変更する必要がある出てくることがあります。ここでは、ポートレットの拡張について、いくつかのヒントやガイドラインを示します。

- ポートレットのテンプレート・ファイルは、Host On-Demand のパブリッシュ・ディレクトリーのポータル・サブディレクトリー (あるいはポートレットを個別にインストールした場合にはデプロイメント・ウィザードのインストール・ディレクトリー) にあります。これらのテンプレートを変更すると、以後生成されるすべてのポートレットに影響します。したがって、これらのファイルを変更する場合には、バックアップしておくようにしてください。テンプレート・ファイルには、Host On-Demand アプレットを表示するための JSP 用のファイルや、WebSphere Portal にポートレットを展開するための XML 記述子用のファイルなどが含まれています。
- 各ポートレットは、JRE にパッケージされている ZIP ユーティリティや JAR ユーティリティなどを使用して簡単に抽出および再アーカイブが可能なアーカイブ・ファイルです。各ディレクトリー名を維持したまま、ポートレットを一時ディレクトリーに抽出します。次に、該当するファイルを変更し、ポートレットを一時ディレクトリーのトップレベルから再アーカイブします。
- XML 記述子は、ポートレットのトップレベル・ディレクトリーにあります。JSP ファイルは、WebSphere Portal 6 の場合は /WEB-INF/hod/html ディレクトリーにあります。
- ポートレットにはカスタム・ヘルプ・ファイルを追加することができます。これを行うには、portlet.xml ファイルで *help* マークアップ・モードをサポートすることを示さなければなりません。ヘルプ情報を含む WpsHODHelp.jsp という名前のファイル (大文字と小文字を区別) および HTML フォーマッターを、ポートレットの JSP ディレクトリーに追加します。
- セッション・プロパティを動的に変更するカスタム・ポートレットを開発することができます。アクセスしようとしている何らかの有用なデータは、ポータル・ユーザーのユーザー名であったり、ページを要求するクライアントの IP アドレスであったりするかもしれません。このデータへのアクセス方法は、ポートレットの API から確認してください。 113 ページの『第 13 章 セッション・プロパティの動的変更』で説明されている HTML の指定変更構文を使用し、この情報から得たデータをアプレットのパラメーター・セットに挿入します。
- ポートレットの開発および API の詳細については、WebSphere Portal と共にインストールされる WebSphere Portal の資料を参照してください。

第 17 章 Eclipse-Plugin サポート

この章では、IBM Eclipse-Plugin 用に Host On-Demand をセットアップする方法について説明します。

注: Host On-Demand は現在、Windows プラットフォームでのみ Eclipse-Plugin をサポートしています。プラットフォームが追加された場合に更新される追加サポートについては、README を確認してください。

Eclipse-Plugin は、次世代のネットワーク中心のコンピューティングのための基盤となるものです。Eclipse リッチ・クライアント・プラットフォーム上に構築されており、アプリケーションを簡単に管理およびデプロイするための追加機能をエンド・ユーザーに提供します。

Eclipse-Plugin では、アプリケーションはすべて、「プラグイン」と「フラグメント」からなる Eclipse「機能」としてパッケージされています。Eclipse 機能は通常、Web にアクセス可能なマシンのディレクトリーである「更新サイト」からインストールされます。

Eclipse-Plugin 用の Host On-Demand プラグインを構築するために、Host On-Demand には「更新サイト・ユーティリティー」と呼ばれる Java アプレットが用意されています。更新サイト・ユーティリティーは、Host On-Demand JAR ファイルを Eclipse プラグインおよびフラグメントに変換して、新規または既存の更新サイト・ディレクトリーに入れます。

更新サイトから機能をインストールする手順は、Workplace Managed Client (WMC) や WebSphere Everyplace Deployment (WED) などの Eclipse-Plugin プラットフォームによって異なります。WMC を使用する場合、片方の Workplace Collaboration Service (WCS) サーバーで追加の構成ステップが必要になります。更新サイト・ユーティリティーは、WCS での構成ステップを簡単にする XML ファイルを生成します。

Host On-Demand プラグインの作成

Eclipse-Plugin で実行するこれらの Host On-Demand プラグインを作成およびデプロイするには、以下を行います。

1. プラグインのセッションを定義する HTML モデルのデプロイメント・ウィザード・ページがあることを確認する。既存の HTML モデル・ページを使用するか、新規作成することができます。

注: Eclipse-Plugin 機能では HTML モデル・ページのみがサポートされています。

ページが完了したら、unzip したデプロイメント・ウィザードの出力ファイルを Host On-Demand パブリッシュ・ディレクトリーに入れます。

2. プラグインの Eclipse 更新サイトとして使用するディレクトリー (c:\update など) をまだ定義していない場合は、作成します。次に、

3. Web サーバー構成でこのディレクトリーの別名を定義して、Web サーバーを再始動します。
4. これで、Host On-Demand プラグインを作成する準備ができました。 Eclipse 更新サイト・マシンで、Java JRE (1.6 以降) を実行するブラウザーを開き、Host On-Demand URL (`http://<hostname>/<alias>/WCTConfig.html`) にアクセスします。

注: Linux では、IBM 1.4.2 Java プラグイン・サービス・リリース 2 以降を使用する場合は、LD_LIBRARY_PATH 環境変数を設定する必要があります。

例えば、Linux 用の Host On-Demand サーバーに付属の Java プラグインを使用したい場合は、以下のようにして export コマンドを使用して、LD_LIBRARY_PATH 環境変数を設定します。

```
export LD_LIBRARY_PATH=/opt/ibm/HostOnDemand/hod_jre/jre/bin:
$LD_LIBRARY_PATH
```

5. この URL で、プラグインの作成を支援する、特別な更新サイト・ユーティリティー・アプレットが実行される。
6. 次のようにして、更新サイト・ユーティリティーの「基本情報 (Basic Information)」パネルに入力する。
 - 更新サイト宛先ディレクトリー (**Update Site Destination Directory**) (必須) ステップ 2 で作成した Eclipse 更新サイト・ディレクトリー (例えば、`c:\updates`) を指定する。
 - HOD コード・ベース (**HOD Code Base**) (必須) ステップ 3 の説明どおりに `WCTConfig.html` にアクセスする場合は、このフィールドは既に正しく入力されているはずです。このフィールドには、Host On-Demand パブリッシュ・ディレクトリーの場所を `http://<hostname>/<alias>` の形式で指定する必要があります。Host On-Demand サーバー名は完全修飾名にしなければなりません。相対 URL 名または「localhost」や「127.0.0.1」のような名前にすることはできません。
 - デプロイメント・ウィザードの出力ファイル (**Deployment Wizard Output File**) (必須) ステップ 1 で作成した HTML モデルのデプロイメント・ウィザード・ページの名前を指定する。
 - 機能バージョン (**Feature Version**) (必須) 生成される機能で使用するバージョン・ストリングを `major.minor.service` の形式で (例えば、`1.0.0`) 指定する。
 - ユーザー JAR ファイル・パス (**User JAR File Path**) (オプション) Host On-Demand セッションと対話するためにカスタム・コードを必要とするソリューションで使用される、カスタム・コードが含まれた JAR ファイルのパスを指定する。コンマ (,) で区切って、複数のファイルを指定することができます。

注: 「アプレットの実行」機能を使用する必要がある場合は、アプレットを JAR ファイルにパッケージして、ここでファイル・パスを指定する必要があります。

7. 「更新サイト・ユーティリティー (**Update Site Utility**)」パネルの「ランタイム・コード (**Runtime Codes**)」パネルと「コード・ページ (**Code Pages**)」パ

ネルで、不要な機能またはホスト・コード・ページのチェック・マークを外して、作成される Eclipse プラグインのサイズを削減することができます。

8. すべてのフィールドが完了したら、「プラグインの生成およびデプロイ (**Generate and Deploy Plugin**)」を選択する。アプレットが Host On-Demand プラグインを作成して、指定した更新サイトに配置します。
9. 更新サイト宛先ディレクトリーとして指定したディレクトリーで、以下のファイルが作成または変更される。
 - **サイト・マップ・ファイル (site.xml)**: このファイルには、この更新サイトからインストール可能な機能がリストされています。
 - **XMLAccess** スクリプト・ファイル: このファイルは、WCS に Host On-Demand 機能をインストールするための WebSphere Portal XMLAccess ユーティリティーの入力です。ファイル名は、(デプロイメント・ウィザードの出力ファイル名)_DeployScript.xml の形式で付けられます。XMLAccess については、『IBM Accelerators for WebSphere Portal family』を参照してください。
 - **features** サブディレクトリー: このサブディレクトリーには、Host On-Demand 機能のアーカイブが含まれています。
 - **plugins** サブディレクトリー: このサブディレクトリーには、次のものが含まれています。

Host On-Demand プラグイン	プラグイン自体。ファイル名は、 <i>com.ibm.eNetwork.HOD.wct_(プラグイン・バージョン).jar</i> の形式で付けられる。
Host On-Demand コード・フラグメント	Host On-Demand ランタイム・コード。ファイル名は、 <i>com.ibm.eNetwork.HOD.wct.(機能名)_(プラグイン・バージョン).jar</i> の形式で付けられる。
構成フラグメント	構成情報を保管するフラグメント。ファイル名は、 <i>com.ibm.eNetwork.HOD.wct.configs.(デプロイメント・ウィザードの出力ファイル名)_(機能バージョン).jar</i> の形式で付けられる。

- **images** サブディレクトリー: このサブディレクトリーには、WMC/WCS で使用されるイメージ・ファイルが含まれています。

プラグインのクライアントへのインストールについては、ご使用の Eclipse-Plugin プラットフォームに付属の資料を参照してください。

セッション・プロパティーの動的設定

Eclipse-Plugin プラットフォームでは、Host On-Demand プラグインの実行に HTML ファイルは使用されないため、セッション・プロパティーを動的に設定するために HTML オーバーライドを使用することはできません。同様の機能が必要な場合は、以下のステップを実行してください。

1. *com.ibm.eNetwork.HOD.wct.IHODConfigFactory* インターフェースを実装する Java クラスを実装する。このインターフェースは、*wct.jar* ファイルに保管されています。*wct.jar* ファイルは、Host On-Demand パブリッシュ・ディレクトリーにインストールされます。インターフェースには、以下の 2 つの **public** メソッドがあります。

```
public String setHodHtmlFileName()
public Properties getHodHtmlParameters()
```

以下に、そのような Java クラスの例を示します。

```
package com.ibm.eNetwork.HOD.wct.samples;

import java.util.Properties;

import com.ibm.eNetwork.HOD.wct.IHODConfigFactory;

public class ConfigOverride implements IHODConfigFactory {
    /* (non-Javadoc)
     * @see com.ibm.eNetwork.HOD.wct.IHODConfigFactory#getHodHtmlFileName()
     */
    public String getHodHtmlFileName() {
        return "hodwmc";
    }

    /* (non-Javadoc)
     * @see com.ibm.eNetwork.HOD.wct.IHODConfigFactory#getHodHtmlParameters()
     */
    public Properties getHodHtmlParameters() {
        Properties p = new Properties();
        p.put("EnableHTMLOverrides", "true");
        p.put("TargetedSessionList", "3270 Display");
        p.put("host", "3270 Display=hostname");
        return p;
    }
}
```

図 9. Java クラスの例

2. Java クラスを JAR ファイルにパッケージする。
3. Host On-Demand パブリッシュ・ディレクトリー内にある更新サイト・ユーティリティーの HTML ファイル (WCTConfig.html) を編集して、showUserClass パラメーターを true に設定する。
var showUserClass="true";
4. 更新サイト・ユーティリティーを実行して、次のように追加のパラメーターを指定する。「ユーザー JAR ファイル・パス (User JAR File Path)」: ステップ 2 で作成された JAR ファイルのファイル・パス。「ユーザー構成ファクトリー・クラス (User Configuration Factory Class)」: ステップ 1 (147 ページ) で実装した Java クラスの名前。
5. Host On-Demand プラグインを生成し、それを Eclipse-Plugin プラットフォームにデプロイする。

別個のユーザー・パブリッシュ・ディレクトリーを使用

Host On-Demand パブリッシュ・ディレクトリー以外のユーザー・パブリッシュ・ディレクトリーを使用する場合は、以下の手順を使用して、更新サイト・ユーティリティーでディレクトリーを指定する必要があります。

1. Host On-Demand パブリッシュ・ディレクトリー内にある更新サイト・ユーティリティーの HTML ファイル (WCTConfig.html) を編集して、showAlternatePublishDirectory パラメーターを true に設定する。
var showAlternatePublishDirectory ="true";

2. 更新サイト・ユーティリティを実行して、「代替パブリッシュ・ディレクトリー」入力フィールドに別のユーザー・パブリッシュ・ディレクトリーを指定する。

Host On-Demand プラグインで使用されるビュー ID

以下に、Host On-Demand プラグインで使用されるビュー ID のリストを示します。WCS でページ・レイアウトを手動で構成する場合は、これらの ID についての知識が必要です。

ID	説明
com.ibm.eNetwork.HOD.wct.SessionsView	構成済みセッション
com.ibm.eNetwork.HOD.wct.SessionLabelsView	アクティブ・セッション
com.ibm.eNetwork.HOD.wct.TerminalView	端末 (ディスプレイ、プリンター、FTP など)

Eclipse-Plugin 環境での Host On-Demand の使用に関する制限

Eclipse-Plugin 環境での Host On-Demand の使用に関する、前述されていない制限を以下に示します。

1. Host On-Demand モーダル・ダイアログが、Eclipse-Plugin シェル・ウィンドウの背後に隠れることがある。これは、Host On-Demand でダイアログが開いている場合に、ユーザーが Eclipse-Plugin の外部にある別のアプリケーションに切り替えると発生します。ユーザーは、ALT-TAB を使用して、確認が必要な HOD ダイアログを見つける必要があります。
2. 「終了の確認」が機能しない。Eclipse-Plugin 環境では、「終了の確認」設定は無視されます。この設定はサポートされていないため、このオプションはセッション・プロパティーから除去されました。
3. セッションを起動したときに、宛先アドレスが構成されていないと、Host On-Demand アプレットがセッション・プロパティー・ダイアログを起動できない。Eclipse-Plugin 環境では、ユーザーが宛先アドレスを必要とするメッセージを受信しても、プロパティー・ダイアログは開きません。
4. マクロ管理プログラム、キーパッド、ツールバーなどの GUI エレメントを、実行中のセッションに動的に追加できない。代わりに、セッション・プロパティーの「設定」セクションで既存のプロパティーを使用して、これらの項目を使用可能にする必要があります。
5. セッションは常にエディター・ペイン内にあるため、この環境では「別のウィンドウで開始」するためのオプションには意味がない。このオプションは、セッション・プロパティーから除去されています。
6. デバッグ機能を備えたクライアントしか使用できない。占有スペースを少なくするために、デプロイメント・ウィザードのプリロード・オプションを使用して、コンポーネントのプリロードを減らすことはできません (ホスト・コード・ページおよび 5250 ファイル転送を除く)。
7. Host On-Demand キャッシュ・クライアントと異なり、クライアントが新規コード・レベルに自動的に更新されない。管理者は、Eclipse-Plugin プラット

フォームが新しいプラグインやフラグメントをインストールできるように、更新サイトを再構成する必要があります。

8. アプレットが JAR ファイルにパッケージされ、クライアント・マシンにインストールされている場合しか、「アプレットの実行」が機能しない。
9. IPMON トレースは、「通常」モードでのみサポートされる。「自動モード」モードはサポートされません。IPMON の実行モードについては、オンライン・ヘルプの『IPMON トレースの概要 (Overview of IPMON tracing)』のトピックを参照してください。
10. 複数の Host On-Demand 機能がインストールされていると、Host On-Demand プラグインには、インストール済みの Host On-Demand 機能のリストが構成済みセッション・ビューで表示される。このビューでは、ユーザーが 1 つの機能を選択することができます。1 つの機能を選択した後は、別の機能を選択するには WED を再始動しなければなりません。
11. Alt キーを押して放すと、Java コンソールで例外がスローされる。これは IBM 1.4.2 JRE の既知の問題で、IBM 1.4.2 サービス・リリース 4.1 以降では解決されています。

第 18 章 LDAP を使用する Host On-Demand サーバーの構成

Host On-Demand サーバーは、構成サーバー・ベース・モデルおよび結合モデルの構成データを管理するために使用されます。Host On-Demand サーバーのデフォルト動作モードでは、このデータは非共有のプライベート・データ・ストアに保管されます。顧客企業によっては、構成情報を複数の Host On-Demand サーバーの間で管理する必要がある場合もあります。このような企業で非共有のプライベート・データ・ストアを使用した場合、管理者はそれぞれの Host On-Demand サーバーのデータを個別に管理しなければなりません。Lightweight Directory Access Protocol (LDAP) サーバーのディレクトリーでは、Host On-Demand 構成サーバーの異なるインスタンス間で、ユーザーおよびグループの構成情報を共有することができます。

LDAP ディレクトリー・サーバーを使用して、複数の Host On-Demand サーバー間でユーザーの定義を管理および共有することを選択するのであれば、慎重に計画し実行する必要があります。プライベート・データ・ストアからのマイグレーションは、特に構成データに大きく影響します。LDAP では、ユーザーを階層ツリーのグループとして編成することによって、構成情報を管理可能にします。既存のユーザーが複数のグループのメンバーであった場合、一部の情報が失われます。プライベート・データ・ストア内の構成データが、LDAP へのマイグレーション時に変更されないことに注意してください。詳しくは、Host On-Demand のオンライン・ヘルプの『LDAP へのマイグレーションによる影響 (implications of migrating to LDAP)』を参照してください。

LDAP サポートの設定

1. 使用する LDAP ディレクトリー・サーバーを決定し、必要に応じてそれをインストールする。
2. Host On-Demand のスキーマをサポートしていない LDAP バージョンを実行している場合は、152 ページの『スキーマ拡張機能のインストール』で説明されているとおりに、Host On-Demand のスキーマ拡張ファイルをインストールする。(スキーマ拡張ファイルは IBM LDAP バージョン 3.x 以降では不要です。)
3. LDAP 管理者に、Host On-Demand が構成情報を保管するために使用する接尾部を問い合わせる。この接尾部の識別名 (DN) を書き留めておいてください。この情報は、LDAP のセットアップを完了するために必要となります。
4. LDAP 管理者に、Host On-Demand の管理者 DN およびパスワードを問い合わせる。これらは、LDAP サーバーへの認証に使用されます。管理者 DN には、前のステップで言及されている接尾部に対する作成、変更、および削除権限がなければなりません。DN およびパスワードを書き留めておいてください。この情報は、LDAP のセットアップを完了するために必要となります。
5. 管理ユーティリティの「ディレクトリー・サービス」ウィンドウで LDAP を使用可能にする。また、任意でプライベート・データ・ストアの構成情報を

LDAP ディレクトリー・サーバーにマイグレーションしてください。詳しくは、151 ページの『第 18 章 LDAP を使用する Host On-Demand サーバーの構成』を参照してください。



他の目的で既に LDAP に定義されているユーザーおよびグループは、Host On-Demand では使用されません。Host On-Demand のユーザーおよびグループは、構成情報をプライベート・データ・ストアからマイグレーションするか、LDAP を使用可能にした後で Host On-Demand にユーザーおよびグループをセットアップすることによって、個別に定義しなければなりません。



Windows または AIX オペレーティング・システム上で IBM LDAP サーバーを使用していて、多数のユーザーを作成する場合は、DB2 が APP_CTL_HEAP_SZ の正しい値で構成されていることを確認してください。この変数の値はそれぞれのインストールで異なりますが、開始時に APP_CTL_HEAP_SZ を 512 に設定すると良いでしょう。

Windows または AIX 環境で DB2 ヒープ・サイズを構成するには、次のコマンドを発行します。

1. set DB2INSTANCE=ldapdb2
2. db2 connect to ldapdb2
3. db2 update db cfg for ldapdb2 using APP_CTL_HEAP_SZ 512
4. db2 force application all
5. db2 terminate
6. db2stop
7. db2start

さらに、STMTHEAP の大きさが十分あるようにします。このパラメーターのサイズは、それぞれのカスタマー構成と、LDAP にマイグレーションする Host On-Demand ユーザーの数にのみ依存します。

スキーマ拡張機能のインストール

LDAP ディレクトリー・スキーマへの Host On-Demand 拡張機能は、パブリッシュ・ディレクトリーの LDAP サブディレクトリー (例えば、`your_install_directory%HOST%ldap`、ここで `your_install_directory` はご使用の Host On-Demand インストール・ディレクトリー) にあるいくつかのファイルで提供されます。これらのファイルは、LDAP スキーマへの拡張機能を含み、標準の slapd 形式で保管されています。Host On-Demand が LDAP サーバーに構成情報を保管するためには、スキーマ拡張機能が有効になっていなければなりません。LDAP 管理者に、これらのスキーマ拡張機能をインストールするよう要請してください。

zSeries でのスキーマ拡張機能のインストールに関する説明は、プログラム・ディレクトリーを参照してください。



LDAP 管理者は、これらのスキーマ拡張機能をすでに別の IBM 製品用にインストールしている可能性があります。その場合は、以下のステップをスキップしてください。IBM Directory Server バージョン 3.1.1 以降を使用している場合、スキーマはプリインストールされているので、この場合も以下のステップをスキップすることができます。

Netscape LDAP Directory Server 上に Host On-Demand のスキーマ拡張機能をインストールするには、以下のようにします。

1. 以下の slapd ファイルを <Host On-Demand パブリッシュ・ディレクトリー>/ldap ディレクトリーから LDAP サーバー上の Netscape の LDAP config ディレクトリーにコピーする。

```
Netscape.IBM.at  
Netscape.IBM.oc
```

2. LDAP サーバーを停止する。
3. <Netscape の LDAP config ディレクトリー>/slapd.conf ファイルを編集して、以下のステートメントを追加する。

```
userat "<Netscape LDAP config ディレクトリー>/Netscape.IBM.at"  
useroc "<Netscape LDAP config ディレクトリー>/Netscape.IBM.oc"
```

4. LDAP サーバーを再始動する。

IBM LDAP Directory Server に Host On-Demand のスキーマ拡張機能をインストールするには、以下のようにします。

1. 以下の slapd ファイルを <Host On-Demand パブリッシュ・ディレクトリー>/ldap ディレクトリーから LDAP サーバー上の <インストール・ディレクトリー>/etc ディレクトリーにコピーする。

```
V2.1.IBM.at  
V2.1.IBM.oc
```

2. LDAP サーバーを停止する。
3. <インストール・ディレクトリー>/etc/slapd.at.conf ファイルを編集して、以下のステートメントをファイルの最後に追加する。

```
include /etc/V2.1.IBM.at
```

4. <インストール・ディレクトリー>/etc/slapd.oc.conf ファイルを編集して、以下のステートメントをファイルの最後に追加する。

```
include /etc/V2.1.IBM.oc
```

5. LDAP サーバーを再始動する。

LDAP をデータ・ストアとして使用する場合の Host On-Demand サーバーの構成

1. 「管理」ウィンドウを開き、Host On-Demand にログオンする。
2. 「サービス」>「ディレクトリー・サービス (Directory Service)」をクリックする。
3. 「ディレクトリー・サービスの使用 (LDAP) (Use Directory Service (LDAP))」ボックスをクリックして、以下の LDAP サーバー情報を入力する。

宛先アドレス

LDAP ディレクトリーの IP アドレスを入力します。ホスト名、またはドット 10 進形式を使用します。デフォルトは、Host On-Demand サーバーのホスト名になっています。

宛先ポート

LDAP サーバーが LDAP クライアントからの接続を受け入れる TCP/IP ポートを入力します。デフォルト・ポートは 389 です。

管理者の識別名

Host On-Demand の情報更新を許可するディレクトリー管理者の識別名 (DN) を入力します。識別名には、LDAP のストリング表現を使用する必要があります (例: cn=Chris Smith、o=IBM、c=US)。

管理者パスワード

ディレクトリー管理者のパスワードを入力します。

識別名の接尾部

情報を保管するディレクトリー情報ツリー (DIT) の最上位項目の識別名 (DN) を入力します。Host On-Demand は、DIT の中に、この接尾部以下のすべての構成情報を保管します。識別名には LDAP のストリング表現を使用する必要があります (例: cn=HOD、o=IBM、c=US)。

ディレクトリー・サービスへの構成のマイグレーション

プライベート・データ・ストアから LDAP ディレクトリーにユーザーおよびグループをマイグレーションするには、このチェック・ボックスをクリックします。LDAP へのマイグレーションは、グループおよびユーザーの構成情報に対して重要な意味を持ちます。詳しくは、オンライン・ヘルプの『LDAP マイグレーションにおける考慮点 (LDAP Migration Implication)』を参照してください。このチェック・ボックスは、ディレクトリー・サーバーへ切り替え時、または切り替えを行った後に使用することができます。



リダイレクター構成は、ディレクトリー・サーバーにマイグレーションされません。



LDAP への接続およびマイグレーションに問題が生じた場合は、最初に LDAP への接続を試みてください。それから、接続が正しくできたら、マイグレーションを試みてください。

4. 「適用」をクリックする。

最初に LDAP ディレクトリーでの認証を求められたら、"admin" というユーザー ID と "password" というパスワードを指定してください。最初のログオンの後で、このパスワードを変更することができます。プライベート・データ保管用にパスワードを変更した場合でも、その ID とパスワードはプライベート・データ保管用にのみ有効なままです。LDAP ディレクトリーの場合は、別のユーザー ID とパスワードが必要です。混乱を避けるために、LDAP ディレクトリー・パスワードをプライベート・データ・ストア・パスワードと同じに変更することができます。

このパネルで行った変更はすぐに適用されます。LDAP サーバーに切り替えると、グループ、ユーザー、またはセッションの管理上の変更や、管理者またはユーザーによる新規パスワード、マクロ、キーボードの変更など、ユーザー関連の変更は LDAP サーバーにおいてのみ行われるようになります。

付録 A. ローカル・インストール・クライアントの使用

ローカル・インストール・クライアントは、ローカル・ディスクにインストールされます。クライアント・アプレットは直接デフォルトのシステム・ブラウザにロードされるため、サーバーからのダウンロードは行われません。ローカル・クライアントを構成する最も一般的な理由としては、低速の電話回線でリモート接続を行うユーザーにとっては、ダウンロードにかかる時間が問題であり接続性が予測不可能、ということがあげられます。また、ローカル・インストール・クライアントを使用することにより、Host On-Demand 製品をすべてインストールすることなく、ホスト・アクセス機能をテストすることができます。

ローカル・インストール・クライアントをサポートするオペレーティング・システム

Host On-Demand は、以下のオペレーティング・システム上にクライアントとしてインストールできます。

- Windows 7
- Windows 8
- Windows 10
- Windows Server 2012

ローカル・インストール・クライアントにはおよそ 320 MB のディスク・スペースが必要です。

ローカル・クライアントのインストール

Host On-Demand ローカル・クライアントを Windows ワークステーションにインストールするには、管理者グループのメンバーであることが必要です。

1. DVD を挿入し、DVD の ¥HODINST ディレクトリーから `hodinstallwin.exe -lc` を実行する。
2. 「インストール」をクリックする。
3. 残りのウィンドウで作業を続ける。
4. 最後のウィンドウで Readme ファイルを開いて読む (まだ読んでいない場合)。

インストールが終了すると、Host On-Demand サービス・マネージャーが自動的に構成され、開始します。Windows 7、Windows 8、および Windows 10 では、サービス・マネージャーはサービスとしてインストールされます。

ローカル・クライアントの開始

Host On-Demand をクライアントとして開始するには、「スタート」 > 「プログラム」 > 「IBM Host On-Demand」 > 「Host On-Demand」とクリックします。

ローカル・クライアントの削除

ローカル・クライアントを除去するには、「コントロール パネル」から「プログラムの追加と削除」を使用します。

付録 B. IKEYCMD コマンド行インターフェースの使用

Host On-Demand 証明書管理ユーティリティのほかに、IKEYCMD というコマンド行ツールが用意されています。これは、鍵、証明書、および認証要求を管理する際に使用することができます。これは機能的には証明書管理ユーティリティと類似したのですが、グラフィカル・インターフェースを使用せず、コマンド行から実行する目的で設計されています。アプリケーションが証明書および鍵の管理タスクにカスタム・インターフェースを追加する際に、ネイティブのシェル・スクリプトおよびプログラムから呼び出して使用することができます。証明書管理ユーティリティが現在サポートしているすべてのタイプの鍵データベース・ファイルを作成することができます。認証要求の作成、CA 署名証明書のインポート、自己署名証明書の管理を行うことが可能です。これは Java ベースであり、Windows、AIX、Linux Intel、および Linux zSeries の各プラットフォームでのみ使用できます。

公開/秘密鍵の作成および管理に関連する構成タスクには IKEYCMD を使用してください。サーバーの構成ファイル `httpd.conf` を更新する構成オプションで IKEYCMD を使用することはできません。サーバーの構成ファイルを更新するオプションには、IBM Administration Server を使用しなければなりません。

IKEYCMD コマンド行インターフェースの環境のセットアップ

IKEYCMD コマンド行インターフェースを使用するために、以下のようにして環境変数をセットアップしてください。

Windows プラットフォームの場合は以下のようにします。

- ユーザー・インターフェースを使用するか、コマンド・ウィンドウで `autoexec.bat` を変更することによって、以下のように `PATH` 変数に Java 実行可能ファイルのロケーションが組み込まれるように設定/変更してください。

```
set PATH=c:\Program Files\IBM\HostOnDemand\jre\bin;%PATH%;
```

- ユーザー・インターフェースを使用するか、コマンド・ウィンドウで `autoexec.bat` を変更することによって、以下のように `CLASSPATH` 環境変数を設定/変更してください。

```
set CLASSPATH=c:\Program Files\IBM\GSK7\classes\cfwk.zip;C:\Program Files\IBM\GSK7\classes\gsk7cls.jar;%CLASSPATH%;
```

AIX プラットフォームの場合は以下のようにします。

まず最初に、(標準 AIX C++ コンパイラー用のランタイム・ライブラリーを構成する) `xlC` ファイルが次のいずれかの要件を満たしていることを確認してください。

- AIX 5.2 の場合: ファイル・セット `xlC.aix50.rte` がレベル 6.0.0.3 以降でなければならない

次のコマンドを使用して、ご使用のバージョンを確認します。

```
lsllpp -ha "xlC.aix*.rte"
```

(ご使用の xIC ファイル・セットが古くて、Host On-Demand サービス・マネージャーを証明書管理をアクティブにして開始した場合は、エラーが発生します。)

次に、以下のように指定します。

- PATH を Java または JRE 実行可能ファイルがある場所に設定します。

```
EXPORT PATH=/opt/IBM/HostOnDemand/hod_jre/jre/bin:$PATH
```

- 以下の CLASSPATH 環境変数を設定します。

```
EXPORT CLASSPATH=/usr/local/ibm/gsk7/classes/cfwk.zip:/usr/local/ibm/gsk7/classes/gsk7cls.jar:$CLASSPATH
```

こうしたステップを終えると、任意のディレクトリーから IKEYCMD を実行できるようになります。IKEYCMD コマンドを実行するには、以下の構文を使用してください。

```
java com.ibm.gsk.ikeyman.ikeycmd <command>
```

IKEYCMD コマンド行構文

Java CLI の構文は以下のとおりです。

```
java [-Dikeycmd.properties=<properties_file>]  
com.ibm.gsk.ikeyman.ikeycmd <object> <action> [options]
```

ここで、次のように指定します。

- `-Dikeycmd.properties` には、この Java 呼び出しで使用するオプションのプロパティー・ファイルの名前を指定します。デフォルトのプロパティー・ファイルは `ikminit_hod.properties` です。このファイルは Host On-Demand のデフォルト設定を含むサンプル・ファイルとして提供されています。
- `object` は以下のいずれかになります。
 - `-keydb`: 鍵データベース (CMS 鍵データベース・ファイルまたは TLSight クラス) で実行されるアクション。
 - `-version`: IKEYCMD のバージョン情報の表示。
- アクションは次のいずれかです。
 - `-cert`: 証明書で実行されるアクション。
 - `-certreq`: 認証要求で実行されるアクション。
 - `-help`: IKEYCMD 呼び出しのヘルプの表示。

`action` はオブジェクトで実行される特定のアクション、`options` は必須または任意のオプションで、オブジェクトおよびアクションのペアに指定します。



オブジェクトおよびアクションのキーワードには定位置があるため、選択された順序どおりに指定しなければなりません。しかし、`options` は、オプションおよびオペランドのペアとして指定される場合には、定位置に関係なく任意の順序で指定することができます。

Host On-Demand の IKEYCMD タスクのリスト

この付録の以下のセクションでは、Host On-Demand で必要となる IKEYCMD コマンド行インターフェースのタスクを要約しています。

- 『新規鍵データベースの作成』
- 161 ページの『CA のリスト表示』
- 167 ページの『鍵データベースのデフォルト鍵の表示』
- 167 ページの『暗号化されたデータベース・パスワードの stash ファイルへの保管』
- 161 ページの『新規鍵ペアと証明書要求の作成』
- 162 ページの『サーバー証明書の保管』
- 164 ページの『自己署名証明書の作成』
- 164 ページの『サーバー証明書をクライアントに使用可能にする』
- 166 ページの『鍵のエクスポート』
- 166 ページの『鍵のインポート』

新規鍵データベースの作成

鍵データベースは、サーバーが 1 つ以上の鍵ペアおよび証明書を保管する際に使用するファイルです。このファイルは、Host On-Demand サーバーとクライアントの間でセキュア接続を使用可能にするために必要となります。 TLS 通信を構成する前に、Windows の場合には `your_install_directory\bin` に、AIX の場合には `your_install_directory/bin` に `HODServerKeyDb.kdb` 鍵データベース・ファイルを作成しなければなりません。このファイルは Host On-Demand に同梱されていないため、最初にインストールを行った後、作成しなければなりません。

例えば Windows プラットフォームの場合、IKEYCMD コマンド行インターフェースを使用して新規の鍵データベースを作成するには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -keydb -create
-db your_install_directory\bin\HODServerKeyDb.kdb
-pw <password> -type cms -expire <days> -stash
```

`your_install_directory` は Host On-Demand のインストール・ディレクトリーです。

以下の説明に注意してください。

- `<password>`: 各鍵データベースの操作にはパスワードが必須です。 `sslight` タイプのデータベースにも指定のパスワードが必要ですが、このパスワードは NULL スtringにできます (" と指定します)。
- `-type`: Host On-Demand サーバーが使用する `HODServerKeyDb.kdb` のタイプは CMS です。
- `-expire`: パスワードが失効するまでの日数。
 - このパラメーターを設定しないと、パスワードは失効しません。
 - 警告: このパラメーターを設定した場合、リダイレクターを含む鍵データベースを使用していると、パスワードが失効した後でリダイレクターが実行できなくなることにご注意ください。リダイレクターで障害が起きたとき、リ

ダイレクターからのエラー・メッセージでは、鍵データベースのパスワードの有効期限が切れたことは示されません。

- `-stash`: 鍵データベースのパスワードを隠しておきます。IBM HTTP Server および Host On-Demand サーバーでは、パスワードを隠しておく必要があります。

鍵データベースの作成時に `-stash` オプションが指定されると、パスワードは `HODServerKeyDb.sth` という名前のファイルに隠されます。

`HODServerKeyDb.kdb` ファイルが作成された後は、Host On-Demand サーバーが必要とするすべてのセキュリティ情報がここに保持されます。すべての追加および変更は、既存の `HODServerKeyDb.kdb` 鍵データベース・ファイルに対して行われます。



`HODServerKeyDb.kdb` ファイルを作成した場合、またはこのファイルに変更を行った場合は、Host On-Demand サービス・マネージャーを停止して再始動しなければなりません。

データベース・パスワードの設定

新規の鍵データベースを作成する際には、鍵データベースのパスワードを指定します。このパスワードにより、秘密鍵がプロテクトされます。秘密鍵は、文書に署名したり、公開鍵で暗号化されたメッセージを暗号化解除したりすることのできる唯一の鍵です。鍵データベース・パスワードを頻繁に変更することは良い習慣です。

パスワードを指定する場合は、以下のガイドラインを使用してください。

- パスワードは米国英語の文字セットで指定しなければなりません。
- パスワードは 6 文字以上で、少なくとも 2 つの連続しない数字を含んでいなければなりません。パスワードには、自分や配偶者あるいは子の頭文字や誕生日など、他人が入手し得る情報は使用しないようにしてください。
- パスワードは隠しておくようにしてください。



パスワードの有効期限を常に把握してください。パスワードが失効すると、エラー・ログにメッセージが書き込まれます。パスワードの有効期限が切れた場合、サーバーは起動しますが、セキュアなネットワーク接続は確立されません。

データベース・パスワードの変更

データベース・パスワードを変更するには、以下のようになります。

例えば Windows プラットフォームの場合、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -keydb -changepw
-db your_install_directory%bin%HODServerKeyDb.kdb
-pw <password> -new_pw <new_password> -expire <days> -stash
```

`your_install_directory` は Host On-Demand のインストール・ディレクトリーです。

以下の説明に注意してください。

- `-new_pw`: 新しい鍵データベースのパスワードです。このパスワードは旧パスワードと異なっていなければならない、また、NULL スtringにすることもできません。
- `-expire`: パスワードが失効するまでの日数。
- `-stash`: 鍵データベースのパスワードを隠しておきます。 IBM HTTP Server および Host On-Demand サーバーでは、パスワードを隠しておく必要があります。

CA のリスト表示

HODServerKeyDb.kdb 鍵データベースのトラステッド CA のリストを表示するには、以下のようにします。

例えば Windows プラットフォームの場合、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -list CA
-db your_install_directory%bin%HODServerKeyDb.kdb
-pw <password> -type cms
```

`your_install_directory` は Host On-Demand のインストール・ディレクトリーです。

HODServerKeyDb.kdb には、デフォルトで以下の既知のトラステッド CA の CA 証明書が用意されています。

- IBM World Registry CA
- Integrion CA Root (IBM World Registry から)
- VeriSign Class 1 Public Primary CA
- VeriSign Class 2 Public Primary CA
- VeriSign Class 3 Public Primary CA
- VeriSign Class 4 Public Primary CA
- VeriSign Test CA
- RSA Secure Server CA (VeriSign から)
- Thawte Personal Basic CA
- Thawte Personal Freemail CA
- Thawte Personal Premium CA
- Thawte Premium Server CA
- Thawte Server CA

新規鍵ペアと証明書要求の作成

公開鍵と秘密鍵のペアおよび認証要求を作成するには、以下のようにしてください。

1. 例えば Windows プラットフォームの場合、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -certreq -create
-db your_install_directory%bin%HODServerKeyDb.kdb
-pw <password> -size <1024 | 512> -dn <distinguished_name>
-file <filename> -label <label>
```

your_install_directory は Host On-Demand のインストール・ディレクトリーです。

以下の説明に注意してください。

- -size: 鍵のサイズ (512 または 1024)。
- -label: 証明書または認証要求に付加されるラベル。
- -dn: X.500 Distinguish Name (公開鍵持ち主情報)。これは、以下の書式の引用符付きストリングによる入力です (CN、O、および C のみが必須。
CN=common_name、O=organization、OU=organization_unit、
L=location、ST=state/province、C=country)。
"CN=weblinux.raleigh.ibm.com,O=ibm,OU=IBM HTTP Server,L=RTP,ST=NC,C=US"
- -file: 認証要求を保管するファイルの名前。デフォルトでは、Host On-Demand は certreq.arm という名前を使用し、HODServerKeyDb.kdb のある *your_install_directory*\bin (ここで、*your_install_directory* はご使用の Host On-Demand インストール・ディレクトリーです) に保管されます。

2. 証明書が正常に作成されたか検証します。

a. 作成した認証要求ファイルの内容を表示します。

b. 鍵データベースが以下のように認証要求を記録したかどうか確認します。

```
java com.ibm.gsk.ikeyman.ikeycmd -certreq -list  
-db <filename> -pw <password>
```

リストされている作成したラベルを確認する必要があります。

3. 新たに作成したファイルを認証局に送信します。

サーバー証明書の保管

CA 署名証明書の受信

以下の手順に従って、ユーザーのサーバーでトラステッド認証局 (CA) と指定された CA から電子メールで送信された証明書を受信してください。

HODServerKeyDb.kdb 鍵データベースには、デフォルトで以下の CA 証明書が保管され、トラステッド CA 証明書としてマークされています。

- IBM World Registry CA
- Integrion CA Root (IBM World Registry から)
- VeriSign Class 1 Public Primary CA
- VeriSign Class 2 Public Primary CA
- VeriSign Class 3 Public Primary CA
- VeriSign Class 4 Public Primary CA
- VeriSign Test CA
- RSA Secure Server CA (VeriSign から)
- Thawte Personal Basic CA
- Thawte Personal Freemail CA
- Thawte Personal Premium CA

- Thawte Premium Server CA
- Thawte Server CA

認証局からは複数の証明書が送信される場合があります。CA は、サーバー用の証明書に加え、追加の署名証明書または中間 CA 証明書も送信する場合があります。例えば、Verisign がグローバル・サーバー ID 証明書を送信する場合には、中間 CA 証明書が含まれます。サーバー証明書を受信する前に、追加の中間 CA 証明書をすべて受信してください。中間 CA 証明書の受信については、『CA 証明書の保管』の指示に従ってください。



CA 署名証明書を発行した CA が鍵データベースにおけるトラステッド CA でない場合は、まず CA 証明書を保管してから、その CA をトラステッド CA として指定します。すると、CA 署名証明書をデータベースに受信することができません。トラステッド CA でない CA から CA 署名証明書を受信することはできません。説明については、『CA 証明書の保管』を参照してください。

例えば Windows プラットフォームで CA 署名証明書を鍵データベースに受信するには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -receive -file <filename>
-db your_install_directory%bin%HODServerKeyDb.kdb -pw <password>
-format <ascii | binary> -default_cert <yes | no>
```

your_install_directory は Host On-Demand のインストール・ディレクトリーです。

以下の説明に注意してください。

- -format: 認証局は、CA 証明書を ASCII またはバイナリー・フォーマットで提供します。
- -label: CA 証明書に付加されるラベル。
- -trust: この CA がトラステッドであるかどうかを示します。CA 証明書を受信する際には enable オプションを使用してください。
- -file: CA 証明書を含むファイル。

CA 証明書の保管

例えば Windows プラットフォームでトラステッド CA でない CA からの証明書を保管するには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -add
-db your_install_directory%bin%HODServerKeyDb.kdb
-pw <password> -label <label> -format <ascii | binary>
-trust <enable | disable> -file <file>
```

your_install_directory は Host On-Demand のインストール・ディレクトリーです。

以下の説明に注意してください。

- -label: 証明書または認証要求に付加されるラベル。
- -format: 認証局はバイナリーまたは ASCII ファイルを提供します。
- -trust: この CA が信頼できるかどうかを示します。これは「Yes」にしてください。



コマンドの実行後には、Host On-Demand サービス・マネージャーを停止して再始動しなければなりません。

自己署名証明書の作成

既知の CA から証明書を取得するには、通常 2 ～ 3 週間かかります。発行される証明書を待つ間に、IKEYCMD を使用して自己署名サーバー証明書を作成して、クライアントとサーバーの間の TLS セッションを使用可能にすることができます。自身がプライベート Web ネットワークの CA である場合は、この手順を使用してください。

例えば Windows プラットフォームで自己署名証明書を作成するには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -create
-db your_install_directory\bin\HODServerKeyDb.kdb
-pw <password> -size <1024 | 512> -dn <distinguished name>
-label <label> -default_cert <yes or no>
```

your_install_directory は Host On-Demand のインストール・ディレクトリーです。

以下の説明に注意してください。

- -size: 鍵のサイズ (512 または 1024)。
- -label: データベースで鍵および証明書を識別するための説明コメントを入力します。
- -dn: X.500 Distinguish Name (公開鍵持ち主情報) を入力します。これは、以下の書式の引用符付きストリングによる入力です (CN、O、および C のみが必須。CN=common_name、O=organization、OU=organization_unit、L=location、ST=state/province、C=country)。
"CN=weblinux.raleigh.ibm.com,O=ibm,OU=IBM HTTP Server,L=RTP,ST=NC,C=US"
- -default_cert: この証明書を鍵データベースのデフォルト証明書にする場合は、yes を入力します。デフォルトにしない場合は、No を入力します。

サーバー証明書をクライアントに使用可能にする

HODServerKeyDb.kdb にある証明書は、すべて Host On-Demand サーバーで使用することができます。しかし構成によっては、これらの証明書のいずれかを、サーバーにアクセスするクライアントでも使用可能にしなければなりません。サーバーが不明な CA からの証明書を使用している場合、その証明書のルートがクライアントで使用できるようにしなければなりません。サーバーが自己署名証明書を使用している場合は、その証明書のコピーをクライアントで使用できるようにしなければなりません。

Host On-Demand の、ダウンロード・クライアントおよびキャッシュ・クライアントでこれを行うには、証明書を一時ファイルに抽出し、CustomizedCAs.p12 (Host On-Demand のパブリッシュ・ディレクトリーにあります) というファイルを作成するか、そのファイルを更新します。

ダウンロード・クライアントあるいはキャッシュ・クライアントで CustomizedCAs.p12 ファイルを作成するには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman -keydb -create -db  
CustomizedCAs.p12 -pw hod -type pkcs12
```

デフォルト・パスワードは hod です。

CustomizedCAs.p12 への不明な CA のルートの追加

初めに、HODServerKeyDb.kdb 鍵データベース・ファイルから CA のルート証明書または自己署名証明書を抽出します。Windows プラットフォームでそうするには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -extract  
-db your_install_directory\bin\HODServerKeyDb.kdb  
-pw <password> -label <label> -target cert.arm -format ascii
```

your_install_directory は Host On-Demand のインストール・ディレクトリーです。

以下の説明に注意してください。

- -label: 証明書に付加されるラベル。
- -pw: HODServerKeyDb.kdb 鍵データベース・ファイルをオープンするためのパスワード。
- -target: 宛先のファイルまたはデータベース。この場合、デフォルトのファイル名は cert.arm という Base-64 Armored ASCII フォーマット・ファイルの名前です。
- -format: ASCII またはバイナリー。

ここで、この CA ルート証明書を CustomizedCAs.p12 ファイルに追加します。CA ルート証明書または自己署名証明書を CustomizedCAs.p12 の署名者リストに追加するには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -add  
-db CustomizedCAs.p12 -pw hod -label <label>  
-file cert.arm -format ascii -trust <enable | disable>
```

古いクライアントの場合、この CA ルート証明書を CustomizedCAs.class ファイルに追加するには、次のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -add  
-db CustomizedCAs.class -label <label>  
-file cert.arm -format ascii -trust <enable | disable>
```

以下の説明に注意してください。

- -label: 追加する証明書のラベル。
- -file: 証明書の抽出先ファイル名。この場合、デフォルトのファイル名は cert.arm という Base-64 Armored ASCII フォーマット・ファイルの名前です。
- -format: ASCII またはバイナリー。

- **-trust:** トラストド・ルートとして設定するかどうかを決定します。 **enable** の場合は CA ルート証明書または自己署名証明書をトラストド・ルートとして設定します。 **disable** の場合は CA ルート証明書または自己署名証明書をトラストド・ルートとして設定しません。



このタスクが完了したら、Host On-Demand サービス・マネージャーを停止して再始動してください。

古いクライアントの場合は、次のコマンドを入力して、ダウンロード・クライアントまたはキャッシュ・クライアントに対する CustomizedCAs.p12 ファイルを CustomizedCAs.class ファイルに変換する必要があります。コマンドは表示上は 3 行になっていますが、1 行で入力してください。

```
..¥hod_jre¥jre¥bin¥java -cp ..¥lib¥sm.zip;  
com.ibm.eNetwork.HOD.convert.CVT2SSLIGHT  
CustomizedCAs.p12 hod CustomizedCAs.class
```

鍵のエクスポート

鍵を別の鍵データベースや PKCS12 ファイルにエクスポートするには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -export -db <filename>  
-pw <password> -label <label> -type <cms | jks | jceks | pks12>  
-target <filename> -target_pw <password>  
-target_type <cms | jks | jceks | pkcs12> -encryption <strong | weak>
```

以下の説明に注意してください。

- **-label:** 証明書に付加されるラベル。
- **-target:** 宛先のファイルまたはデータベース。
- **-target_pw:** ターゲット鍵データベースのパスワード。
- **-target_type:** **-target** オペランドで指定されたデータベースのタイプ。
- **-encryption:** 暗号化の強度。デフォルトは **strong** です。

鍵のインポート

別の鍵データベースから鍵をインポートするには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -import -db <filename>  
-pw <password> -label <label> -type <cms | jks | jceks | pks12> -target  
<filename> -target_pw <password> -target_type <cms | jks | jceks | pks12>
```

PKCS12 ファイルから鍵をインポートするには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -import -file <filename>  
-pw <password> -type pkcs12 -target <filename>  
-target_pw <password> -target_type <cms | jks | jceks | pks12>
```

以下の説明に注意してください。

- **-label:** 証明書に付加されるラベル。
- **-target:** 宛先データベース。

- `-target_pw: -target` で鍵データベースが指定されている場合の鍵データベースのパスワード。
- `-target_type: -target` オペランドで指定されたデータベースのタイプ。

鍵データベースのデフォルト鍵の表示

例えば Windows プラットフォームでデフォルトの鍵項目を表示するには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -cert -getdefault
-db your_install_directory%bin%HODServerKeyDb.kdb
-pw <password>
```

`your_install_directory` は Host On-Demand のインストール・ディレクトリーです。

暗号化されたデータベース・パスワードの **stash** ファイルへの保管

セキュア・ネットワーク接続では、暗号化されたデータベース・パスワードを **stash** ファイルに保管します。例えば Windows プラットフォームで、データベースの作成時にパスワードを保管するには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -keydb -create
-db your_install_directory%bin%HODServerKeyDb.kdb
-pw <password> -type cms -expire <days> -stash
```

`your_install_directory` は Host On-Demand のインストール・ディレクトリーです。

例えば Windows プラットフォームで、データベースの作成後にパスワードを保管するには、以下のコマンドを入力します。

```
java com.ibm.gsk.ikeyman.ikeycmd -keydb -stashpw
-db your_install_directory%bin%HODServerKeyDb.kdb -pw <password>
```

`your_install_directory` は Host On-Demand のインストール・ディレクトリーです。

IKEYCMD コマンド行パラメーターの概要

以下の表は、特定のオブジェクトにおいて実行可能なそれぞれのアクションについて説明したものです。

オブジェクト	アクション	説明
-keydb	-changepw	鍵データベースのパスワードを変更する。
	-convert	鍵データベースをあるフォーマットから別のフォーマットへ変換する。
	-create	鍵データベースを作成する。
	-delete	鍵データベースを削除する。
	-stashpw	鍵データベースのパスワードをファイルに隠しておく。

-cert	-add	CA 証明書をファイルから鍵データベースに追加します。
	-create	自己署名証明書を作成する。
	-delete	CA 証明書を削除する。
	-details	特定の証明書の詳細情報をリストする。
	-export	個人用証明書およびそれに関連付けられた秘密鍵を、鍵データベースから PKCS#12 ファイルまたは別の鍵データベースへエクスポートする。
	-extract	鍵データベースから証明書を抽出する。
	-getdefault	デフォルトの個人用証明書を取得する。
	-import	鍵データベースまたは PKCS#12 ファイルから証明書をインポートする。
	-list	すべての証明書をリストする。
	-modify	証明書に変更を行う (注: 現在、変更可能なフィールドは「証明書の信頼」フィールドのみ)。
	-receive	証明書をファイルから鍵データベースに受信します。
	-setdefault	デフォルトの個人用証明書を設定する。
	-sign	ファイルに保管された証明書を、鍵データベースに保管された証明書で署名し、その署名済み証明書をファイルに保管する。
-certreg	-create	認証要求を作成する。
	-delete	認証要求データベースから認証要求を削除する。
	-details	特定の認証要求の詳細情報をリストする。
	-extract	認証要求データベースから認証要求をファイルに抽出する。
	-list	認証要求データベース内のすべての認証要求をリストする。
	-recreate	認証要求を再作成する。
-help		IKEYCMD コマンドのヘルプ情報を表示する。

-version	IKEYCMD のバージョン情報を表示する。
----------	------------------------

IKEYCMD コマンド行オプションの概要

以下の表は、コマンド行で指定できるそれぞれのオプションを示したものです。オプションは完全なグループとしてリストされていますが、オプションの使用はコマンド行で指定されるオブジェクトおよびアクションによって異なります。

オプション	説明
-db	鍵データベースの完全修飾パス名。
-default_cert	クライアント認証でデフォルトの証明書として使用される証明書を設定する (yes または no)。デフォルトは no。
-dn	X.500 Distinguish Name (公開鍵持ち主情報)。以下の書式の引用符付きストリングによる入力 (CN、O、および C のみが必須)。 "CN=Jane Doe,O=IBM,OU=Java Development,L=Endicott,ST=NY,ZIP=13760,C=country"
-encryption	証明書のエクスポート・コマンドで使用する暗号化の強度 (strong または weak)。デフォルトは strong。
-expire	証明書またはデータベース・パスワードの有効期限 (日数)。デフォルトは、証明書の場合は 365 日、データベース・パスワードの場合は 60 日。
-file	証明書または認証要求のファイル名 (指定されたオブジェクトによって異なる)。
-format	証明書のフォーマット (Base64_encoded ASCII の場合は ascii、バイナリー DER データの場合は binary)。デフォルトは ascii。
-label	証明書または証明書要求に付加されたラベル
-new_format	鍵データベースの新しいフォーマット。
-new_pw	新しいデータベース・パスワード。
-old_format	鍵データベースの古いフォーマット。
-pw	鍵データベースまたは PKCS#12 ファイルのパスワード。 159 ページの『新規鍵データベースの作成』を参照してください。
-size	鍵のサイズ (512 または 1024)。デフォルトは 1024。
-stash	鍵データベースのパスワードをファイルへ隠しておく場合の標識。これを指定すると、パスワードはファイルへ隠される。
-target	宛先のファイルまたはデータベース。

-target_pw	-target で鍵データベースが指定されている場合の鍵データベースのパスワード。 159 ページの『新規鍵データベースの作成』を参照してください。
-target_type	-target オペランドで指定されているデータベースのタイプ (-type を参照)。
-trust	CA 証明書の信頼状況 (enable または disable)。デフォルトは enable。
-type	データベースのタイプ。指定可能な値は、cms (CMS 鍵データベースを示す)、jce (Sun 専用の Java Cryptography Extension を示す)、jceks (Sun 専用の Java Cryptography Extension Key Store を示す)、または pkcs12 (PKCS#12 ファイルを示す) です。
-x509version	作成する X.509 証明書のバージョン (1、2 または 3)。デフォルトは 3。

コマンド行呼び出し

以下に示すのは、各コマンド行呼び出しのリストです (オプション・パラメーターはイタリックで示しています)。

簡略化するために、実際には存在する Java 呼び出しの部分 (java com.ibm.gsk.ikeyman.ikeycmd) を、各コマンド呼び出しから省略しています。

```
-keydb -changepw -db <filename> -pw <password>
-keydb -new_pw <new_password> -stash -expire <days>

-keydb -convert -db <filename> -pw <password>
-keydb -old_format <cms | webdb> -new_format <cms>

-keydb -create -db <filename> -pw <password> -type <cms | jks | jceks | pkcs12>
-keydb -expire <days> -stash

-keydb -delete -db <filename> -pw <password>

-keydb -stashpw -db <filename> -pw <password>

-cert -add -db <filename> -pw <password> -label <label>
-cert -file <filename> -format <ascii | binary> -trust <enable | disable>

-cert -create -db <filename> -pw <password> -label <label>
-cert -dn <distinguished_name> -size <1024 | 512> -x509version <3 | 1 | 2>
-cert -default_cert <no | yes>

-cert -delete -db <filename> -pw <password> -label <label>

-cert -details -db <filename> -pw <password> -label <label>

-cert -export -db <filename> -pw <password> -label <label>
-cert -type <cms | jks | jceks | pkcs12> -target <filename> -target_pw <password>
-cert -target_type <cms | jks | jceks | pkcs12> -encryption <strong | weak>

-cert -extract -db <filename> -pw <password> -label <label>
-cert -target <filename> -format <ascii | binary>

-cert -getdefault -db <filename> -pw <password>

-cert -import -db <filename> -pw <password> -label <label>
-cert -type <cms | jks | jceks | pkcs12> -target <filename> -target_pw <password>
-cert -target_type <cms | jks | jceks | pkcs12>
```

```

-cert -import -file <filename> -type <pkcs12> -target <filename>
-target_pw <password> -target_type <cms | jks | jceks | pks12>

-cert -list <all | personal | CA | site> -db <filename>
-pw <password> -type <cms | jks | jceks | pks12>

-cert -modify -db <filename> -pw <password> -label <label>
-trust <enable | disable>

-cert -receive -file <filename> -db <filename> -pw <password>
-format <ascii | binary> -default_cert <no | yes>

-cert -setdefault -db <filename> -pw <password> -label <label>

-cert -sign -file <filename> -db <filename> -pw <password>
-label <label> -target <filename> -format <ascii | binary>
-expire <days>

-certreq -create -db <filename> -pw <password> -label <label>
-dn <distinguished_name> -size <1024 | 512> -file <filename>

-certreq -delete -db <filename> -pw <password> -label <label>

-certreq -details -db <filename> -pw <password> -label <label>

-certreq -extract -db <filename> -pw <password> -label <label>
-target <filename>

-certreq -list -db <filename> -pw <password>

-certreq -recreate -db <filename> -pw <password> -label <label>
-target <filename>

-help

-version

```

ユーザー・プロパティー・ファイル

Java CLI 呼び出しで一部を入力しなくても済むように、プロパティー・ファイルにユーザー・プロパティーを指定することができます。 `-Dikeycmd.properties` Java オプションを使用することによって、Java コマンド行呼び出しでプロパティー・ファイルを指定することができます。 Windows プラットフォームの場合、サンプルのプロパティー・ファイル `ikminit_hod.properties` が `your_install_directory\bin` に用意されています (ここで `your_install_directory` は Host On-Demand のインストール・ディレクトリーです)。 AIX プラットフォームの場合は、このファイルは `your_install_directory/bin` に用意されています。これらのインストール・ディレクトリーには Host On-Demand のデフォルト設定が含まれています。

付録 C. P12 鍵リング・ユーティリティー

グラフィカルな証明書管理ユーティリティー (Windows および AIX プラットフォームで使用可能) が提供されているので、認証要求を作成し、証明書を受け取って保管し、さらには自己署名証明書を作成することができます。P12 鍵リング・ユーティリティーは、証明書管理ユーティリティーのないプラットフォームで、自己署名のあるルート証明書および不明な認証局証明書を有する鍵リング・データベースを作成するために主に準備されています。しかし、このユーティリティーはどの Host On-Demand オペレーティング・システムでも使用できます。このユーティリティーによって、システム管理者は簡単に TLS 鍵リング・データベースを作成してデプロイできます。

P12 鍵リング・ユーティリティーは Java で書かれています。TLS 用に構成されている Telnet または FTP サーバー (あるいはリダイレクター) からサーバー証明書を入手します。指定されたサーバーおよび TLS ポートに対して TLS 接続が行われます。ポートが準備されていない場合には、既知のセキュアな Telnet または FTP ポートが使用されます。サーバーの証明書は抽出され、指定された p12 ファイルに追加されます。

鍵リング・データベースへのアクセスは、パスワードによって保護されます。コマンドが実行される前に、パスワード・プロンプトが出されます。指定された鍵リング・ファイルが存在しない場合には作成され、パスワードがそのファイルに保管されます。



Host On-Demand TLS サポートのためには、パスワードが hod であることが必要です。私用証明書を鍵リング・データベースに追加する場合には、別のパスワード・プロンプトが 2 番目の p12 ファイル用に出されます。

使用法

```
P12Keyring p12FileName connect ipaddr[:port] [ftp]  
P12Keyring p12FileName add p12FileName2  
P12Keyring p12FileName list
```

オプション

connect - 指定された ipaddr およびポートへの TLS 接続を確立します。ポート番号および ftp キーワードは、オプションです。ポート番号が指定されていない場合には、デフォルトのセキュアな Telnet ポート 443 またはデフォルトのセキュアな FTP ポート 990 が使用されます。

ftp キーワードが指定されている場合は、セキュリティ用に構成されているセキュア FTP サーバーに接続されます。FTP サーバー用のセキュリティ・オプションには 2 つのタイプがあります。

- ポート 990 への暗黙的セキュリティ
- その他のポートへの明示的セキュリティ

ftp キーワードが指定されていても、ポート番号が指定されていない場合、またはポート番号が 990 の場合には、暗黙的セキュリティー・ネゴシエーションが行われます。**ftp** キーワードが指定され、ポート番号が 990 ではない場合には、AUTH TLS コマンドをまず発行して、明示的セキュリティー・ネゴシエーションを行います。

add - 指定された鍵リング・データベースに私用クライアント証明書を追加します。

list - 指定された鍵リング・データベースに保管された証明書のリストを表示します。

例

Windows:

```
C:¥your_install_dir¥lib¥P12Keyring c:¥your_install_dir¥HOD¥CustomizedCAs  
connect myServer.raleigh.ibm.com:702
```

```
C:¥your_install_dir¥lib¥P12Keyring c:¥your_install_dir¥HOD¥CustomizedCAs  
connect myFTPServer.raleigh.ibm.com:5031 ftp
```

your_install_dir は Host On-Demand のインストール・ディレクトリーです。

Unix:

```
cd your_install_directory/HOD  
Java -classpath .;your_install_dir/lib/sm.zip ¥  
com.ibm.hod5ssligh.tools.P12Keyring CustomizedCAs connect  
myServer.raleigh.ibm.com:702
```

your_install_dir は Host On-Demand のインストール・ディレクトリーです。

付録 D. ネイティブ・プラットフォーム・ランチャーのコマンド行オプション

ネイティブ・プラットフォーム・ランチャーで以下のコマンド行オプションを入力すると、ランチャーはそれらをインストール・パラメーターとして Host On-Demand インストールに渡します。GUI ウィザードを使用しないオプションには、相応の注が付けられています。

表 13. コマンド行オプション

オプション	目的	使用例
-console (GUI ウィザードを使用しません)	コンソール・モードで Host On-Demand をインストールします。	install.exe
-log #!filename ここで、# は標準出力へ表示をエコーし、!filename はログ・ファイルの名前です。ファイル名なしで！を指定すると、デフォルト・ログ・ファイル名が使用されます。	指定された名前ですインストール・ログ・ファイルを生成します。	hodinstallwin.exe -log #!¥mydirectory¥logfile
-options filename	指定されたプロパティをインストールに設定するコマンド行オプションを使用して、Host On-Demand をインストールします。	hodinstallwin.exe -silent -options c:¥mydirectory¥responseFile
-options-record filename	応答を Host On-Demand インストール・ウィザードに記録するオプション・テキスト・ファイルを生成します。それを、インストール変数のデフォルト値として設定します。	hodinstallwin.exe -options-record responses.txt
-options-template filename	デフォルトのインストール値を含む、オプション・テキスト・ファイルを生成します。	hodinstallwin.exe -options-template template.txt
-silent (GUI ウィザードを使用しません)	サイレント・モードで Host On-Demand をインストールし、デフォルトのインストール値すべてを受け入れます。	hodinstallwin.exe -silent

以下の追加のコマンド行オプションは、インストール・プログラムを呼び出して実行するプロセス にのみ適用されます。ネイティブ・プラットフォーム・ランチャーを使用して、コマンド行でこれらを入力します。

表 14. 起動固有のコマンド行オプション

オプション	目的	使用例
<code>-is:logfile<i>filename</i></code>	ネイティブ・ランチャーの JVM 検索用のログ・ファイルを生成します。	<code>hodinstallwin.exe -is:log myLogFile.txt</code>
<code>-is:silent</code>	JVM 検索および他の初期化の実行時に、ランチャー・ユーザー・インターフェース (UI) が表示されないようにします。(通常、 <code>silent</code> コマンド行オプションで使用されます。)	<code>hodinstallwin.exe -is:silent</code>
<code>-is:tempdir<i>directory</i></code>	Host On-Demand インストールで使用される一時ディレクトリを設定します。	<code>hodinstallwin.exe -is:tempdir "c:¥temp"</code>

付録 E. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

付録 F. 商標

IBM、IBM ロゴおよび **ibm.com** は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Microsoft、Windows および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。



Printed in Japan

SC43-3077-02



日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21