

**IBM Content Manager OnDemand for iSeries
Common Server**



**Web Enablement Kit
インストールおよび構成ガイド**

バージョン 5 リリース 3

**IBM Content Manager OnDemand for iSeries
Common Server**



**Web Enablement Kit
インストールおよび構成ガイド**

バージョン 5 リリース 3

ご注意

本書および本書で紹介する製品をご使用になる前に、161ページの『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 Content Manager OnDemand for iSeries Common Server のバージョン 5 リリース 3 に適用されます。また、改訂版で断りがない限り、それ以降のすべてのリリースおよびモディフィケーションに適用されます。本書は、SC88-4013-00 の改訂版です。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。
<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは
<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

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： SC27-1163-02
IBM Content Manager OnDemand for iSeries Common Server
Web Enablement Kit Installation and Configuration Guide
Version 5 Release 3

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2004.4

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2001, 2004. All rights reserved.

© Copyright IBM Japan 2004

目次

IBM Content Manager OnDemand for iSeries Common Server Web

Enablement Kit インストールおよび構成ガイド (SC88-4013) について v

本書の対象読者	v
本書の構成	v
前提条件および関連情報	vi
WWW で入手可能なその他の情報	vi
iSeries ナビゲーター	vi

I 変更の要約 vii

第 1 章 概要 1

I プログラミング・インターフェースについて	2
ビューアーについて	4
ODWEK の使用	5
製品関数	6
Add Annotation	6
Change Password	6
Document Hit List	6
Logoff	7
Logon	7
Retrieve Document	7
Search Criteria	7
Server Print Document	7
Update Document	7
View Annotations	7
サーバーとデータのセキュリティ	8

第 2 章 HTTP サーバーのインストールおよび構成 11

インストール要件	11
その他の要件	11
OS/400 でのインストール	12
次のステップ	13
ARSWWW.INI ファイルの指定	13
[@SRV@_DEFAULT]	13
[@SRV@_server]	14
[CONFIGURATION]	15
[SECURITY]	23
[AFP2HTML]	24
[AFP2PDF]	25
[MIMETYPES]	27
[ATTACHMENT IMAGES]	32
[NO HTML]	33
[DEFAULT BROWSER]	34
[browser]	42
[DEBUG]	43
サンプルの ARSWWW.INI ファイル	43
次のステップ	45

第 3 章 サンプル・アプリケーションの構成 47

I LOGON.HTM	47
CREDIT.HTM	47
TEMPLATE.HTM	49
次のステップ	49

第 4 章 Web ビューアーのインストール 51

概要	51
要件	52
インストール	53
ユーザー定義ファイルの配布	54
AFP Web Viewer ファイルのインストール	55
サブディレクトリーの追加	55
ユーザー定義ファイルの保管	56
フォント・ファイルの構成	57
AFP Web Viewer インストール・ファイルの構築	58
ユーザーのワークステーションへの AFP Web Viewer のインストール	58
AFP フォントのマッピング	58
I AFP Web Viewer	59
I Image Web Viewer	59
I Java 行データ・ビューアー	60
次のステップ	64

付録 A. CGI API リファレンス 65

Add Annotation	66
Change Password	69
Document Hit List	72
Logoff	77
Logon	79
Print Document (サーバー)	82
Retrieve Document	87
Search Criteria	92
Update Document	95
View Annotations	97

I 付録 B. Java サブレット・リファレンス 99

I 付録 C. Java API リファレンス 101

I 付録 D. Java API プログラミング・ガイド 103

I クライアント/サーバー・アーキテクチャー	103
I Java 環境のパッケージ化	103
プログラミングのヒント	104
I システム環境のセットアップ	105
環境変数の設定	105
I トレースおよび診断情報	106

	トレース	106	AFP2PDF.INI ファイルの指定	147
	例外処理	107	変換された文書の表示	148
	定数	107	付録 G. HTTP サーバー構成ファイル 149	
	ODWEK アプリケーションの実行	107	HTTP Apache サーバー	149
	OnDemand サーバーへの接続	108	WebSphere Application Server	150
	接続の確立	108	付録 H. 非 HTML 出力 151	
	パスワードの設定と入手	109	区切り文字付き ASCII 出力	151
	OnDemand サーバーを使用した作業	109	Logon	152
	フォルダー内のアプリケーション・グループのリス		注	152
	ト表示	111	Search Criteria	152
	フォルダーの検索	113	注	152
	SQL ストリングを使用したフォルダーの検索	117	Document Hit List	153
	検索の取り消し	119	注	154
	検索基準のリスト表示	122	View Annotations	154
	フォルダーおよびフォルダー情報のリスト表示	125	エラー・メッセージ	155
	文書のリストの表示	126	注	155
	文書の検索	128	付録 I. 各国語サポート 157	
	文書の印刷	131	付録 J. 問題判別ツール 159	
	注に関する情報のリスト表示	133	特記事項 161	
	注の追加	135	商標	162
	文書の更新	137	索引 165	
	パスワードの変更	140		
	付録 E. AFP の HTML への変換 143			
	AFP2HTML.INI ファイルの形式	143		
	AFP2WEB Transform 用のオプション	145		
	変換された文書の表示	146		
	付録 F. AFP の PDF への変換 147			

IBM Content Manager OnDemand for iSeries Common Server Web Enablement Kit インストールおよび構成ガイド (SC88-4013) について

本書には、IBM® Content Manager OnDemand for iSeries™ バージョン 5 リリース 3 Common Server (OnDemand) Web Enablement Kit の計画、インストール、構成、および使用に役立つ情報を記載しています。

本書の対象読者

本書は、OnDemand Web Enablement Kit (ODWEK) ソフトウェアおよびアプリケーションのインプリメント、インストール、および保守を行う必要のあるシステム管理者を主に対象にしています。また、Web アプリケーションに OnDemand を組み込む必要のあるプログラマーも本書の対象です。

本書の構成

本書には、ODWEK をインストールおよび構成し、ユーザーが Web ブラウザーを通して IBM Content Manager OnDemand for iSeries Common Server システムのデータにアクセスできるようにする計画を立てる上で必要な情報が含まれています。本書は以下のセクションから構成されます。

- 1 ページの『第 1 章 概要』
- 11 ページの『第 2 章 HTTP サーバーのインストールおよび構成』
- 47 ページの『第 3 章 サンプル・アプリケーションの構成』
- 51 ページの『第 4 章 Web ビューアーのインストール』
- 65 ページの『付録 A. CGI API リファレンス』
- 99 ページの『付録 B. Java サブレット・リファレンス』
- 101 ページの『付録 C. Java API リファレンス』
- 103 ページの『付録 D. Java API プログラミング・ガイド』
- 143 ページの『付録 E. AFP の HTML への変換』
- 147 ページの『付録 F. AFP の PDF への変換』
- 149 ページの『付録 G. HTTP サーバー構成ファイル』
- 151 ページの『付録 H. 非 HTML 出力』
- 157 ページの『付録 I. 各国語サポート』
- 159 ページの『付録 J. 問題判別ツール』

前提条件および関連情報

iSeries の技術情報を検索する場合は、まず IBM iSeries Information Center をご利用ください。

Information Center にアクセスするには、次の 2 通りの方法があります。

- Web サイト <http://www.ibm.com/eserver/series/infocenter> より
- OS/400®に同梱されている CD-ROM (*iSeries V5R3 Information Center*, SK88-8055-03) より

このパッケージには、iSeries マニュアルの PDF 版である「*iSeries V5R2 Information Center : 補足資料*, SK88-8056-01」も含まれています。これは、従来のソフトコピー・ライブラリー CD-ROM に代わるものです。

Information Center には、Java™、TCP/IP、Web サービス、保護されたネットワーク、論理区画、クラスター化、CL コマンド、およびシステム・アプリケーション・プログラミング・インターフェース (API) などに関するアドバイザーや重要なトピックが含まれています。また、関連する IBM Redbooks™ へのリンク、および IBM ホーム・ページなどの他の IBM Web サイトへのインターネット・リンクも含まれています。

WWW で入手可能なその他の情報

WWW では、iSeries のより詳しい情報をご覧になることができます。iSeries のホーム・ページ (Web サイト: <http://www-1.ibm.com/servers/eserver/series/>) から全般的な情報を入手することができます。

iSeries の高度な機能に関するワークショップを利用するには、Web サイトの <http://www.iseries.ibm.com/tstudio/> にある Technical Studio にアクセスしてください。

インターネットを介せば、世界中どこからでも、iSeries のプログラム一時修正 (PTF) を参照、選択、発注、および入手することができます。iSeries インターネット PTF (ダウンロード) および予防サービス計画 (PSP) の情報は、インターネット・サイトの <http://as400service.ibm.com> から入手できます。

iSeries ナビゲーター

IBM iSeries ナビゲーターは、iSeries サーバーを管理するための強力なグラフィカル・インターフェースです。iSeries ナビゲーターは、ユーザーの作業を支援するシステム・ナビゲーション、構成、計画、およびオンライン・ヘルプなどの機能を備えています。iSeries ナビゲーターは、サーバーの運用と管理をさらに容易にすることによって、生産性を向上させるもので、また、OS/400® の新たな拡張機能を利用するための唯一のユーザー・インターフェースでもあります。また、複数のサーバーを中央のシステムから管理するための、マネージメント・セントラルも組み込まれています。

iSeries ナビゲーターの詳細については、IBM iSeries Information Center を参照するか、または Web サイト (<http://www.ibm.com/eserver/series/navigator/>) をご覧ください。

変更の要約

この版の「*IBM Content Manager OnDemand for iSeries Common Server: Web Enablement Kit* インストールおよび構成ガイド」には、新規および変更された技術情報が含まれます。場合によっては、変更が加えられていても変更バーが付けられていない箇所もあります。重要な変更内容は以下のとおりです。

- バージョン 5 リリース 1 で、Content Manager OnDemand for iSeries (OnDemand) は、OnDemand Common Server という新規のサーバー・インプリメンテーションを導入しました。Common Server では、OnDemand のユーザーと管理者に対し、索引付け、検索、表示、セキュリティー、PDF、および Web 対応機能を機能強化として提供します。スプール・ファイル保存を実装した現行の OnDemand をご使用のお客様は、AnyStore 機能または既存の Server 機能の有無に関わらず、「Content Manager OnDemand for iSeries Common Server 計画とインストール」の『付録 A』に記載されている手順に従って、新しい Common Server に移行できるようになりました。本書全体を通じて、スプール・ファイル保存のデータの移行について言及している箇所では、AnyStore がインストールされている場合には AnyStore データも含まれることに注意してください。
- 「Content Manager OnDemand for iSeries Common Server 索引付けリファレンス」資料には、OS/400 Indexer のサポートする機能に関連する内容が大幅に追加されました。これらの追加には、マルチキー索引、トランザクション・フィールド、テキスト検索フィールド、AFP オーバーレイ付きの SCS スプール・ファイル、およびアプリケーション・フィールドのマスクの定義に関するトピックが含まれています。
- Content Manager OnDemand for iSeries は、iSeries 対応の新しい Plasmon 光ディスク・ライブラリーをサポートするようになりました。
- OnDemand 保存記憶域管理の開始 (STRASMOND) コマンドでは、コマンドの使用を簡素化するために、2 つのコマンド・パラメーターが除去されました。詳しくは、「Content Manager OnDemand for iSeries Common Server 管理ガイド」の『付録 A』を参照してください。
- OS/400 では、オリジナルの HTTP サーバーのサポートを終了しました。これに伴い、Content Manager OnDemand Web Enablement Kit (ODWEK) でも、オリジナルの HTTP サーバーのサポートを終了しました。現在、ODWEK では HTTP Apache サーバーのみが、HTTP サーバーとしてサポートされています。

第 1 章 概要

ODWEK によって、ユーザーは、Web ブラウザーまたはユーザー作成プログラムを使用して、IBM Content Manager OnDemand サーバーに保管されているデータにアクセスできます。例えば、OnDemand サーバーにログオンすることができる Web ページの URL を使用できたり、特定のフォルダーを検索できる Web ページの URL を使用することができます。ODWEK は、OnDemand サーバー上のユーザー情報 (サーバーやアプリケーション・グループに保管されているデータに対するアクセス権など) の有効性を検証します。ユーザーが検索を実行依頼すると、ODWEK は、照会と一致する文書のリストが示された Web ページを表示します。ユーザーが表示する文書を選択すると、ODWEK はブラウザーにその文書を送信します。

図 1 では、Web ブラウザーを使用してワークステーションから OnDemand サーバーのデータにアクセスする様子を示します。

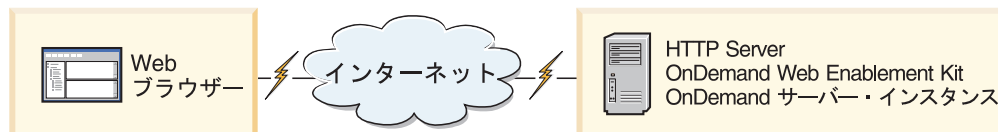


図 1. ODWEK を使用した、OnDemand に保管されているデータへのアクセス

ODWEK が文書を検索したり、取り出したりできるのは、IBM Content Manager OnDemand for iSeries Common Server バージョン 5、IBM Content Manager OnDemand for Multiplatforms バージョン 7、IBM Content Manager OnDemand for z/OS[®] および OS/390[®] バージョン 7.1 が稼働している OnDemand サーバーです。

ODWEK は、以下のいくつかのコンポーネントから構成されています。

- OnDemand プログラミング・インターフェース。このプログラミング・インターフェースは、標準の OnDemand インターフェースとプロトコルを使用して、OnDemand サーバーに保管されているデータにアクセスします。ODWEK をサポートするために、OnDemand サーバーで付加的なコードを使用する必要はありません。ODWEK の制御には、以下のいずれかのプログラミング・インターフェースを使用することができます。
 - 共通ゲートウェイ・インターフェース (CGI) プログラム。CGI プログラムは、Web ブラウザーから OnDemand データにアクセスする方法を提供します。CGI プログラムは、IBM HTTP Server のような Hypertext Transfer Protocol (HTTP) サーバーが稼働しているシステムで稼働します。
 - Java サブレット。CGI プログラムは、Web ブラウザーから OnDemand データにアクセスする方法を提供します。サブレットは、IBM WebSphere[®] Application Server のような Java アプリケーション・サーバーを実行している、Java に対応した HTTP サーバーで稼働します。
 - Java API。Java API は、ユーザー作成プログラムから OnDemand データにアクセスする方法を提供します。Java API には、Java バージョン 1.2.2 以降が必要です。

- IBM OnDemand Advanced Function Presentation™ (AFP™) Web Viewer。AFP Web Viewer を使用すると、Web ブラウザーからの AFP 文書の検索、取り出し、表示、ナビゲート、および印刷が可能になります。
- IBM OnDemand Image Web Viewer。Image Web Viewer を使用すると、Web ブラウザーからの BMP、GIF、JPEG、PCX、および TIFF 文書の検索、取り出し、表示、ナビゲート、および印刷が可能になります。
- Line Data Java アプレット。Line Data アプレットを使用すると、行データの文書を Web ブラウザーで表示することができます。管理者は、ARSWWW.INI ファイルを構成することによって、Line Data アプレットを使用可能にすることができます。
- AFP2HTML Java アプレット。AFP2HTML アプレットを使用すると、IBM AFP2WEB Transform サービス・オファリングによって生成された出力を表示することができます。AFP2WEB Transform では、AFP 文書や資源が AFP2HTML アプレットで表示可能なハイパーテキスト・マークアップ言語 (HTML) ファイルに変換されます。AFP2WEB Transform をインストールおよび構成した後、管理者は、ARSWWW.INI ファイルを構成することによって AFP2HTML アプレットを使用可能にすることができます。

注: OnDemand に保管されている他のタイプの文書を表示するには、適切なビューアーを取得およびインストールする必要があります。例えば、Adobe の Portable Data Format (PDF) 文書を表示する場合、IBM では、ご使用のブラウザーに合った Adobe Acrobat ビューアーを入手することをお勧めしています。

プログラミング・インターフェースについて

ODWEK のインスタンスは、OnDemand サーバー上のデータにアクセスする ODWEK コードです。インスタンスは、データに対して行うことができる操作を制御し、インスタンスに割り当てられているシステム資源を管理します。各インスタンスは、完全な 1 つの環境です。インスタンスには、他のインスタンスがアクセスできない固有の ASWWW.INI ファイルと ODWEK プログラミング・インターフェースがあります。以下の 3 つの ODWEK プログラミング・インターフェースがあります。

- CGI プログラム。これは、Web ブラウザーと OnDemand サーバー間のインターフェースです。
- Java サブレット。これは、Web ブラウザーと OnDemand サーバー間のインターフェースです。
- Java API。これは、ユーザー作成プログラムから OnDemand データにアクセスする際に使用されるメソッドの集合です。

非常に重要なのは、インスタンスが 1 つのプログラミング・インターフェースしか使用できないということです。プログラミング・インターフェースは、同時に使用できません。プログラミング・インターフェースを同時に同じインスタンスで使用することはできません。ただし、各インスタンスが異なるポート番号を使用するように構成することによって、単一のマシンで複数の ODWEK インスタンスを実行し、各インスタンスが異なるプログラミング・インターフェースを使用することができます。

1 つのシステムに単一インスタンスというのが、ODWEK の最も一般的なインプリメンテーションです。単一インスタンス構成は、通常、他のアプリケーションから独立して作動するアプリケーション・サーバー・インスタンスが含まれる、デベロッパーまたはスタンドアロン実動コンピューターに適しています。

図 2 に、CGI インターフェースを使用した単一インスタンスの例を示します。

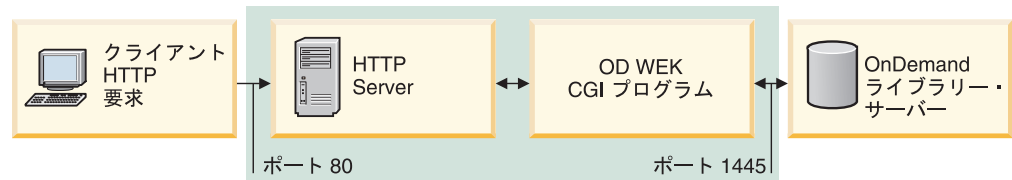


図 2. CGI インターフェースを使用した単一インスタンス

図 3 に、Java サブレット・インターフェースを使用した単一インスタンスの例を示します。

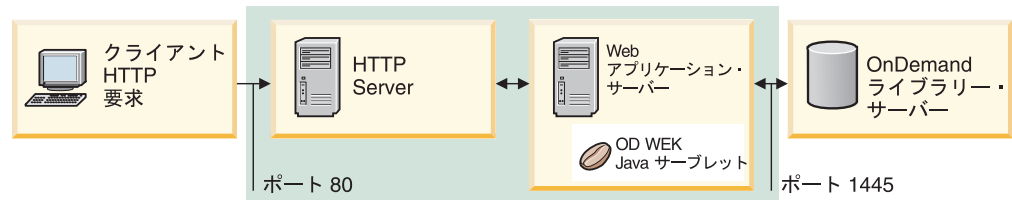


図 3. Java インターフェースを使用した単一インスタンス

図 4 に、Java API インターフェースを使用した単一インスタンスの例を示します。

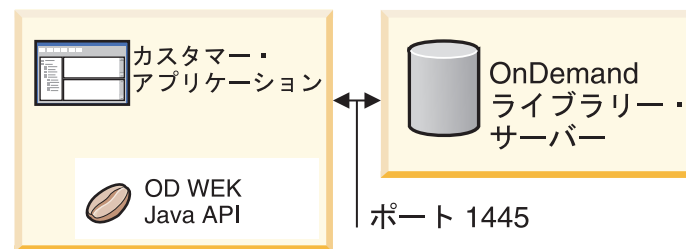


図 4. Java API インターフェースを使用した単一インスタンス

同じシステムで複数の ODWEK インスタンスを構成できます。各インスタンスは、固有のプログラミング・インターフェースと ARSWWW.INI ファイルを必要とします。このファイルで、プログラミング・インターフェースと OnDemand サーバー間で行われる通信に固有のポート番号を指定します。また、各インスタンスは、固有のストレージとセキュリティーも必要とします。複数インスタンス構成は、通常、1 つ以上のデベロッパー、テスト、または実動アプリケーションを同じシステムで実行する必要があるお客様に適しています。インスタンスは、互いに独立して作動します。

4 ページの図 5 に、複数インスタンス・トポロジーの例を示します。

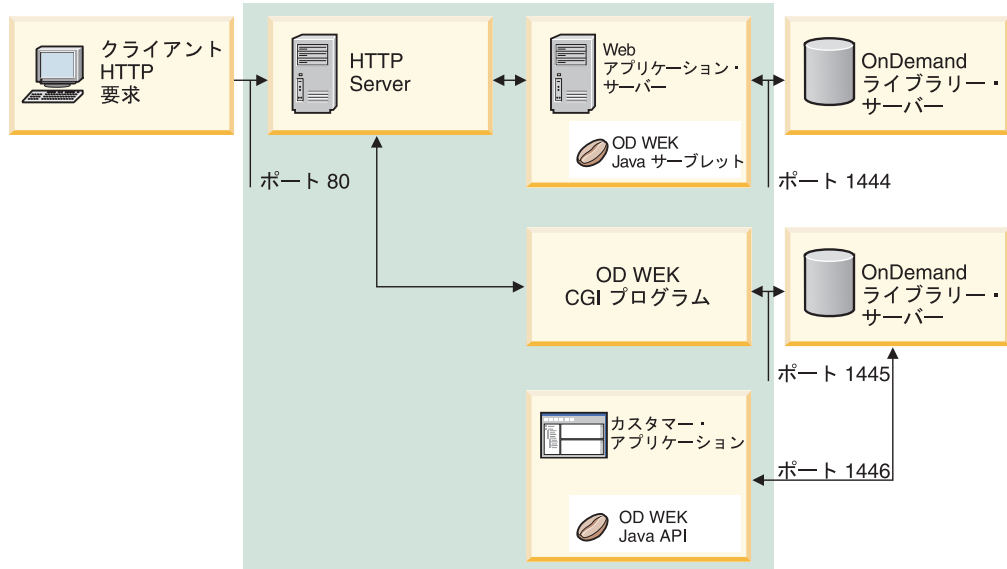


図5. 複数インスタンス・トポロジー

ビューアーについて

ODWEK には、以下のビューアーがあります。

- AFP Web Viewer
- Image Web Viewer
- Line Data Java アプレット
- AFP2HTML Java アプレット

AFP Web Viewer および Image Web Viewer は、独自の方法で Web ブラウザーの機能を拡張するソフトウェア・プログラムです。AFP Web Viewer を使用して、ユーザーは AFP 文書を表示することができます。Image Viewer を使用して、ユーザーは BMP、GIF、JPEG、PCX、および TIFF 文書を表示することができます。これらのビューアーを使用することにより、文書をブラウザーのウィンドウで表示できるようになります。表示ウィンドウの最上部には、それぞれのビューアーに応じてツールバーが表示されます。ビューアーのツールバーは、ブラウザーのツールバーに追加することができます。プラグインのツールバーには、文書の処理に役立つ種々のコントロールが備わっています。文書を表示するために Web ビューアーを使用する場合は、各ワークステーションに Web ビューアーをインストールする必要があります。

注: インストール・プログラムは、プラグインまたは ActiveX コントロールのいずれかとしてビューアーをインストールします。Internet Explorer がワークステーションにインストールされている場合、インストール・プログラムは ActiveX コントロールをインストールします。Netscape がワークステーションにインストールされている場合、インストール・プログラムはプラグインをインストールします。Internet Explorer と Netscape の両方がワークステーションにインス

トールされている場合、インストール・プログラムは Internet Explorer 用に ActiveX コントロールをインストールし、Netscape 用にプラグインをインストールします。

Line Data アプレットを使用すると、ユーザーは、SCS データおよび OnDemand に保管されている行データの文書を表示することができます。Line Data アプレットでは、行データの文書をブラウザのウィンドウに表示します。また、その表示ウィンドウの最上部にはツールバーが追加されます。Line Data アプレットのツールバーには、文書の処理に役立つ種々のコントロールが備わっています。管理者は、ARSWWW.INI ファイルを構成することによって、Line Data アプレットを使用可能にすることができます。

AFP2HTML アプレットを使用すると、IBM AFP2WEB Transform サービス・オフラインによって生成された出力を表示することができます。AFP2WEB Transform は、AFP の文書や資源を HTML 文書に変換します。AFP2WEB Transform をインストールおよび構成した後、管理者は、ARSWWW.INI ファイルを構成することによって AFP2HTML アプレットを使用可能にすることができます。AFP2HTML アプレットにもツールバーがあり、ラージ・オブジェクト用のコントロールも含め、文書の処理に役立つ種々のコントロールが備わっています。

アプレットが持つ利点の 1 つは、ワークステーションにインストールしなければならない Web ビューアーとは異なり、アプレットを使用するためにワークステーションにソフトウェアをインストールしたりアップグレードしたりする必要がない、という点です。また、IBM から Web ビューアーの新規バージョンが提供された場合は、ユーザーに更新版 Web ビューアーを配布する必要があります。

IBM が提供しているアプレットやビューアーを使用する場合、OnDemand サーバーから取り出される文書は、クライアントに達するまで圧縮されたまま処理されます。文書はクライアントで圧縮解除され、各ページは Web ブラウザーのウィンドウに表示されます。文書がラージ・オブジェクトとして OnDemand に保管されている場合は、ユーザーが文書のページを移動するのに応じて適宜クライアントで文書のセグメントが取り出され、圧縮解除されます。

ODWEK の使用

ODWEK を使用する際に最も一般的なのは、製品と共に提供されているサンプルの HTML アプリケーションをカスタマイズして用いる方法です。LOGON.HTM サンプル・アプリケーションは、いくつかの特定のフォルダーに対してアクセス権を持つユーザーをサポートします。LOGON.HTM ページではまず、実際に使用する OnDemand サーバーについての情報で変更を加えます。次いで、その LOGON.HTM ファイルの URL を公表します。ユーザーは、この URL へのリンクを作成し、指定されたサーバーにログオンすることができます。ODWEK は、OnDemand 文書を検索、取り出し、または表示するための一連の Web ページを自動的に表示します。CREDIT.HTM サンプル・アプリケーションは、特定のフォルダーに対する検索基準が含まれた Web ページを備え、非定常的な OnDemand の使用をサポートします。サンプルをカスタマイズした後、その URL に対するユーザーのリンクを作成し、検索基準を指定して、「Submit」ボタンをクリックします。ODWEK は、照会と一致する文書がリストされた Web ページを表示します。

重要: ODWEK を使用するためには、クライアントに Cookie データを書き込むことができない場合があります。各ユーザーのブラウザが必ず Cookie を受け付けるように構成してください。

ほとんどのお客様は、ODWEK を使用してサーバーにアクセスするための OnDemand ユーザー ID を 1 つしか定義しません。これは、多くの OnDemand ユーザーが定期的に同じフォルダーにアクセスする環境において一般的です。とはいえ、それぞれのユーザーに固有の OnDemand ユーザー ID を与えることも可能です。ODWEK でどのように OnDemand にアクセスするにしても、OnDemand のユーザー ID を管理することが必要です。これらのユーザー ID をサーバーに追加し、ユーザーに対してアプリケーション・グループとフォルダーの許可を設定する必要があります。

製品関数

ODWEK では、以下の OnDemand 関数がサポートされています。これらの関数は通常、ODWEK サーバー・プログラムへのリンクを含む Web ページを作成することによって呼び出されます。リンクごとに、特定の関数を呼び出します。1 つの関数の出力は別の Web ページとなり、それと共に次の論理関数へのリンクが示されます。例えば、最初の Web ページでは、Logon 関数が呼び出されます。Logon 関数によって表示される Web ページには、Search Criteria 関数へのリンクが含まれています。各関数は、アプリケーション・プログラミング・インターフェース (API) を使用して呼び出すことができます。詳細については、65 ページの『付録 A. CGI API リファレンス』を参照してください。

Add Annotation

Add Annotation 関数では、指定した文書に注釈を追加することができます。注釈を追加するためには、注釈を追加したい文書が含まれているアプリケーション・グループに対して、Add Annotation 許可を持っていない場合があります。(Application Group Access 許可を持っていれば、注釈を追加することができます。)

Change Password

Change Password 関数では、OnDemand パスワードを変更することができます。

Document Hit List

Document Hit List 関数は、検索基準と一致する項目のリストを作成します。このリストは、HTML 表に示されます。検索と一致する各項目は表のセルで表され、それぞれが Retrieve Document 関数にリンクします。

システムに **OnDemand スプール・ファイル保存**と **Common Server 環境**の両方があり、**ARS.CFG** ファイルの **ARS_MIGR_SERVER** 記入項目を使用して、単一のフォルダー選択リストで **Spool File Archive** フォルダーと **Common Server** フォルダーを結合しているお客様に対する**重要な注記:** ODWEK ユーザーの場合、Spool File Archive フォルダーは ODWEK フォルダー・リストに表示され、検索することができます。しかし、文書を取り出そうとすると、失敗します。

Logoff

Logoff 関数によって、ユーザーは OnDemand サーバーをログオフすることができます。

Logon

Logon 関数によって、ユーザーは OnDemand サーバーにログオンすることができます。Logon 関数が正常に実行されると、ログオンしたユーザーがオープンする権限を持っているフォルダーのリストが Web ページに表示されます。

Retrieve Document

Retrieve Document 関数は、OnDemand から文書を取り出します。サーバーから返されるデータ・ストリームの中には文書が組み込まれ、データ・タイプに応じて、その文書を表示するために必要な資源も組み込まれます。データ・ストリームは、いかなる方法でも変更することができません。ブラウザーやビューアーでは、このデータ・ストリームを解釈およびデコードして文書を表示します。文書がラージ・オブジェクトとして OnDemand に保管されている場合は、文書の最初のセグメントだけが戻されます。文書の後続のセグメントは、必要に応じて取り出され、表示されます。

Search Criteria

ログオンに成功すると、そのユーザーがオープンする権限を持っているフォルダーのリストが表示されます。ユーザーは、オープンするフォルダーを選択します。フォルダーをオープンすると、そのフォルダー用の検索フィールドを含む Web ページが表示されます。ユーザーは、デフォルトの検索基準をそのまま使用するか、あるいは検索基準を入力して、特定の文書を検索することができます。「Submit」ボタンが押されると、その検索の要求が OnDemand サーバーに送信されます。

Server Print Document

Server Print Document 関数は、文書のコピーを OnDemand サーバー・プリンターに送信します。サーバー印刷を使用するためには、印刷する必要のある文書が含まれている各アプリケーション・グループに対して、Document Print 許可を持っている必要があります。(Application Group Access 許可を持っている場合、文書を印刷することができます。) OnDemand サーバーには、少なくとも 1 つ以上のサーバー・プリンターが定義されていなければなりません。

Update Document

Update Document 関数では、データベースを更新することができます。Update Document 関数は、特定の文書に関する 1 つまたは複数のデータベース・フィールドを更新します。

View Annotations

View Annotations 関数では、特定の文書に付加されている注釈を表示することができます。注釈を表示するためには、表示する必要のある注釈が含まれている各アプ

リケーション・グループに対して、Annotation View 許可を持っていないければなりません。(Application Group Access 許可を持っていれば、注釈を表示することができます。)

サーバーとデータのセキュリティ

ODWEK を使用する際には、あらかじめ、以下の 2 つのレベルのセキュリティについて考慮する必要があります。

- ODWEK プログラムと Web ページにアクセスできるのは誰か
- OnDemand サーバー上のデータにアクセスできるのは誰か

HTTP サーバーとそのプログラム、および ODWEK へのフロントエンドが含まれている Web ページにアクセスできるすべてのユーザーは、潜在的に、OnDemand に保管されているデータにもアクセスすることができます。IBM では、プログラムや Web ページへのアクセスには制限を設けることを強くお勧めしています。HTTP サーバー上のプログラムや Web ページに対するアクセスは、さまざまな方法で制限することができます。例えば、多くの HTTP サーバーでは、ディレクトリーに対するアクセスを制限できるようにすることによって、機密情報を含む Web ページにセキュリティ・システムを提供しています。また、HTTP サーバー上でパスワード・ファイルを使用して、ユーザー ID とパスワードを入力しないと Web ページにアクセスできないようにすることもできます。ただし、HTTP サーバーのユーザー ID やパスワードがオペレーティング・システムのユーザー ID やパスワードと類似していても、これらの ID やパスワードとオペレーティング・システムのユーザー ID やパスワードとの間には何の対応関係もありません。また、HTTP サーバーのユーザー ID やパスワードと OnDemand のユーザー ID やパスワードとの間にも、対応関係はありません。

ODWEK では、標準の OnDemand API を使用して OnDemand サーバーおよびデータにアクセスすることができます。API は、OnDemand のユーザー ID にサーバーや要求されたデータにアクセスする権限があるかどうかを検査します。組織内に、OnDemand サーバーにおけるユーザーとデータのセキュリティを管理する担当者を設けてください。

セキュリティに関連して考慮できるもう 1 つの点は、クライアントとサーバーの間でフォームのパラメーターや値を変換する際に使用するメソッドです。ODWEK のフォームでは、POST メソッドを使用して、HTTP 要求の本文内でパラメーターや値を変換します。POST メソッドを使用すると、パラメーターと値はブラウザーの「場所」(または「アドレス」) フィールドに表示されません。例えば、典型的な例では、関数の呼び出しが以下のように表示されます。

```
http://www.company.com/cgi-bin/arswww.cgi
```

ただし、フォームの作成時にメソッドが指定されない場合は、デフォルトのメソッド GET が使用されます。このメソッドでは、URL そのものの中でパラメーターと値が変換されます。GET メソッドを使用した場合の典型的な例では、関数の呼び出しが以下のように表示されます。

```
http://www.company.com/cgi-bin/arswww.cgi?_function=logon
&_user=bob&_password=secret
```

パラメーターと値は、ブラウザー・ウィンドウの「場所」フィールドに明確なテキストとして表示されます。自分でフォームを作成する場合、IBM では、必ず POST メソッドを使用することをお勧めしています。デフォルトのメソッドを GET から POST に変更するには、フォーム・タグで METHOD 属性をコード化する必要があります。

注: GET メソッドを使用する必要がある場合は、ARSWWW.INI ファイルで ENCRYPTURL パラメーターを指定することにより、パラメーターおよび値を暗号化することができます。詳しくは、38 ページの『ENCRYPTURL』を参照してください。

第 2 章 HTTP サーバーのインストールおよび構成

このセクションでは、インストール要件を定義し、さらに、HTTP サーバーに ODWEK ソフトウェアをインストールし、ODWEK 構成ファイルを変更する方法について説明します。

IBM HTTP Server の現行バージョンが稼働している iSeries システムに ODWEK ソフトウェアをインストールする必要があります。さらに、Java サブレットの使用を計画している場合は、iSeries Web Application Server (WebSphere) の現行バージョンを稼働させるようにしてください。

ODWEK が文書を検索したり、取り出したりできるのは、IBM Content Manager OnDemand for iSeries バージョン 5 リリース 3 Common Server が稼働している OnDemand サーバーです。

インストール要件

ODWEK には以下の条件が必要です。

- IBM HTTP Apache サーバーの現行バージョン。さらに、Java サブレットの使用を計画している場合は、iSeries Web Application Server (WebSphere) の現行バージョンを稼働させるようにしてください。サーバーは、OS/400 バージョン 5 リリース 3 の下で稼働させる必要があります。
- インストールに適したメディア・タイプ。
- インストール・ファイル用に十分な大きさのディスク・スペース。HTTP サーバー上に約 30 MB が必要です。
- キャッシュ記憶機構用に十分なディスク・スペース。デフォルトでは、HTTP サーバー上に 10 MB が必要です。詳しくは、18 ページの『CACHESIZE』を参照してください。

その他の要件

ODWEK では、HTTP サーバー上の文書をキャッシュに入れる（一時的に保管する）ことができます。これにより、以前に表示された文書をユーザーに送る処理が速くなります。文書に対してキャッシュ記憶機構を使用できるようにするためには、ARSWWW.INI ファイルで CACHEDOCS パラメーターを構成します。詳細については、17 ページの『CACHEDOCS』を参照してください。

デフォルトでは、ODWEK は /QIBM/UserData/OnDemand/WWW/CACHE ディレクトリーにあるデータをキャッシュに入れます。ARSWWW.INI ファイルを変更することによって、別のキャッシュ・ディレクトリーを指定することができます。詳細については、17 ページの『CACHEDIR』を参照してください。

ODWEK プログラムを実行するプロセスが、プログラムの含まれているディレクトリーから読み取りを行い、キャッシュ・ディレクトリーに書き込みを行えることを確認してください。ODWEK がインストールされると、オブジェクトのすべてが権限リスト QONDADM によって保護され、ユーザー・プロファイル QTMHHTTP、

QTMHHTTP1、および QEJBSVR が *CHANGE 権限で権限リストに追加されます。また、QRDARS400 権限リストには、*USE 権限がある QTMHHTTP、QTMHHTTP1、QEJBSVR が必要です。

AFP2HTML アプレットの使用を計画している場合は、IBM から AFP2WEB Transform サービス・オファリングを入手し、これを HTTP サーバーにインストールし、構成する必要があります。AFP2WEB Transform サービス・オファリングに関する詳細は、IBM 担当員にご相談ください。さらに、AFP2WEB Transform で処理する AFP 文書や資源にも、構成オプションを指定する必要があります。構成ファイルについての詳細は、143 ページの『付録 E. AFP の HTML への変換』を参照してください。

OnDemand に保管されている AFP 文書を Adobe Acrobat ビューアーで表示できる PDF 文書に変換することを計画している場合は、IBM から AFP2PDF Transform サービス・オファリングを入手し、これを HTTP サーバーにインストールし、構成する必要があります。AFP2WEB Transform サービス・オファリングに関する詳細は、IBM 担当員にご相談ください。加えて、AFP2PDF Transform で処理する AFP 文書や資源にも、構成オプションを指定する必要があります。構成ファイルについての詳細は、147 ページの『付録 F. AFP の PDF への変換』を参照してください。変換された文書を表示するためには、ご使用のブラウザに合った Adobe Acrobat ビューアーを入手する必要があります。

OS/400 でのインストール

ODWEK のセットアップでは、一般に次のことを行う必要があります。

1. 最新の OnDemand README ファイルのコピーを入手します。ファイル全体を印刷し、内容を読んでから作業を始めてください。
2. ODWEK をインストールするには、「ソフトウェアの導入」(SD88-5002) という資料に記載されている説明に従ってください。ライセンス・プログラム番号は 5722RD1、フィーチャーは 11 です。

注: ODWEK をインストールするのにお勧めする方法は、「ライセンス・プログラムの処理 (Work with Licensed Programs)」メニュー (licpgm に進む) から「ライセンス・プログラムのインストール (Install licensed programs)」メニュー・オプションを使用することです。「ライセンス・プログラムのインストール (Install licensed programs)」画面からオプションを「追加」するために 1 を入力した後、「ライセンス・プログラム」に 5722RD1 および「製品オプション (Product Option)」に 11 を入力するか、またはライセンス・プログラムと製品オプションのリストをスクロールして ODWEK を見付けてその前の位置に 1 を入力してください。**他の方法で OnDemand をインストールした場合、使用時にエラーが発生する可能性があります。**

3. IBM では、ライセンス・プログラムを正しくインストールした後に、OnDemand に使用可能なすべての PTF をオーダーし、ロードし、適用することを推奨しています。OnDemand バージョン 5 リリース 3 の PTF の完全なリストについては、Informational APAR II13680 を参照してください。この Informational APAR をオンラインでオーダーするには、SNDPTFORD コマンドを使用し、PTF 番号に II13680 を指定します。必ず PTF カバー・レターを読み、特別な指示があれば、それに従ってください。

次のステップ

iSeries システム上で現行バージョンの IBM HTTP サーバーを稼働させるようにしてください。HTTP サーバーを構成する必要があります。HTTP サーバー構成ファイルの例については、149 ページの『付録 G. HTTP サーバー構成ファイル』を参照してください。

Java サブレットの使用を計画している場合は、iSeries Web Application Server (WebSphere) の現行バージョンを稼働させるようにしてください。WebSphere を構成する必要があります。手順については、Web サイトの www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/ にある IBM WebSphere Application Server for AS/400® Documentation Center を参照してください。該当するバージョンの WebSphere の『Installation and Initial Configuration』へのリンクに進んでください。

ODWEK ソフトウェアをインストールし、HTTP サーバーを構成し、(オプションで) WebSphere を構成したら、オペレーティング環境に合わせて ODWEK 初期化ファイルを構成することができます。『ARSWWW.INI ファイルの指定』を参照してください。

ARSWWW.INI ファイルの指定

ARSWWW.INI ファイルは ASCII テキスト・ファイルで、ODWEK プログラム (CGI プログラムや Java サブレット) が読み取るパラメーターが含まれています。各パラメーターは、PARAMETER=value の形式で、それぞれ別個の行に分けて指定します。次に例を示します。

```
AFPVIEWING=plugin
CACHEDIR=/tmp/cache
LANGUAGE=ENU
```

ARSWWW.INI ファイル内のパラメーターは、セクションでグループ分けされています。パラメーターを指定する際には、[sectionHeader] という形式のセクション・ヘッダーを使用してセクションの先頭を指定します。そして、セクション・ヘッダーの下にある、セクションのパラメーターを指定します。次に例を示します。

```
[@SRV@_gunnar]
HOST=gunnar
PORT=1446
PROTOCOL=0
```

本製品には、サンプルの ARSWWW.INI 構成ファイルが付属しています。この構成ファイルの例では、一連の最も一般的に使用される値が使用されています。この例は、43 ページの『サンプルの ARSWWW.INI ファイル』にあります。

ARSWWW.INI ファイルには、以下のセクションとパラメーターがあります。

[@SRV@_DEFAULT]

デフォルト・サーバー・セクション。デフォルト・サーバー・セクションでは、ODWEK が通信を行う OnDemand サーバーに共通のパラメーターを指定することができます。サーバー・セクションでパラメーターと値が指定されない場合は、このセクションで指定されたパラメーターと値が使用されます。

このセクションの有効範囲は、すべてのサーバーに対してグローバルです。また、ARSWWW.INI ファイルの中で一度だけ指定します。

このセクションはオプションです。

このセクションでは、次のパラメーターを指定することができます。

PORT

OnDemand サーバーが ODWEK との通信に使用する TCP/IP ポート番号。PORT パラメーターが指定されない場合、サーバーは、サービス・テーブル (WRKSRVTBLE) で OnDemand 用に指定されているポート番号を使用します。PORT パラメーターが指定されず、サービス・テーブルにも OnDemand がリストされていない場合は、ポート番号 1445 の使用が試行されます。サービス・テーブルで OnDemand に指定されているポート番号を使用するように指定する場合は、0 (ゼロ) を指定してください。

このパラメーターは、デフォルトのセクションで一括して指定することができます。Logon API を使用すれば、_port パラメーターで指定されているポート番号を指定変更することが可能です。

このパラメーターはオプションです。

例:

```
[@SRV@_DEFAULT]
PORT=0
```

PROTOCOL

OnDemand サーバーが ODWEK との通信に使用するネットワーク・プロトコル。TCP/IP の場合には 0 (ゼロ) を指定する必要があります。

このパラメーターは、デフォルトのセクションで一括して指定しなければなりません。

このパラメーターはオプションです。このパラメーターを指定しなければ、値 0 (ゼロ) が使用されます。

例:

```
[@SRV@_DEFAULT]
PROTOCOL=0
```

[@SRV@_server]

サーバー・セクション。ODWEK が通信を行う各 OnDemand サーバーごとに、サーバー・セクションを 1 つずつ指定する必要があります。1 つのサーバー・セクションで指定されるパラメーターと値は、1 つの特定のサーバーに対するものです。セクション・ヘッダーには、特定のサーバーを識別するストリングが含まれていなければなりません。サーバー・セクションで指定されるパラメーターは、デフォルト・サーバー・セクションで指定されるパラメーターを指定変更します。

サーバー・セクションは、各サーバーに 1 つずつ指定する必要があります。

このセクションは必須です。

このセクションでは、次のパラメーターを指定することができます。

HOST

OnDemand サーバーの名前。サーバーの TCP/IP アドレス、ホスト名の別名、および完全修飾ホスト名を指定することができます。

このパラメーターは、サーバー・セクションにつき 1 回指定しなければなりません。

このパラメーターは必須です。

例:

```
[@SRV@_gunnar]
HOST=gunnar
```

PORT

OnDemand サーバーが ODWEK との通信に使用する TCP/IP ポート番号。PORT パラメーターが指定されない場合、サーバーは、デフォルト・サーバー・セクションで指定されている (またはデフォルトの) ポート番号を使用します。

このパラメーターは、サーバー・セクションにつき 1 回指定することができます。Logon API を使用すれば、_port パラメーターで指定されているポート番号を指定変更することが可能です。

このパラメーターはオプションです。

例:

```
[@SRV@_gunnar]
PORT=0
```

このポート番号は、インスタンスの ars.ini ファイルで指定したポート番号と一致している必要があります。

PROTOCOL

OnDemand サーバーが ODWEK との通信に使用するネットワーク・プロトコル。TCP/IP の場合には 0 (ゼロ) を指定する必要があります。

このパラメーターは、サーバー・セクションにつき 1 回指定することができます。

このパラメーターはオプションです。指定されない場合は、デフォルト・サーバー・セクションで指定された (またはデフォルトの) 値が使用されます。

例:

```
[@SRV@_gunnar]
PROTOCOL=0
```

[CONFIGURATION]

CONFIGURATION セクションには、HTTP サーバーで ODWEK が使用するパラメーターが含まれています。

このセクションの有効範囲はグローバルです。また、ARSWWW.INI ファイルの中で一度だけ指定します。

このセクションはオプションです。

このセクションでは、次のパラメーターを指定することができます。

APPLETCACHEDIR

Line Data アプレットおよび AFP2HTML アプレットが一時的に文書を保管するディレクトリーを指定します。ディレクトリーは、ユーザーのワークステーションに対してローカルであっても、ネットワーク・ドライブ上であっても構いません。すべてのユーザーが、指定したディレクトリーに対して書き込みアクセス権限を持つ必要があります。

例:

```
[Configuration]
APPLETCACHEDIR=/QIBM/UserData/OnDemand/www/cache
```

注:

1. APPLETCACHEDIR パラメーターは、グローバルな有効範囲を持ちます。
2. APPLETCACHEDIR パラメーターはオプションです。ただし、このパラメーターが指定されていないと、アプレットは Java 作業ディレクトリーに文書を保管しようとします。
3. 指定されたディレクトリーが無い場合、アプレットは Java 作業ディレクトリーに文書を保管しようとします。
4. アプレットは、ユーザーがアプレットを手放すと (例えば、文書をクローズした場合など)、キャッシュ・ディレクトリーから文書を除去します。

APPLETDIR

Line Data および AFP2HTML アプレットが含まれているディレクトリーを識別します。

注:

1. ディレクトリー名、または AliasMatch を指定することができます。
 - ディレクトリー名を指定する場合、ディレクトリーは、/QIBM/UserData/OnDemand/WWW ディレクトリーを起点とするディレクトリーでなければなりません。例えば、appletdir=applets と指定する場合、アプレットは /QIBM/UserData/OnDemand/WWW/APPLETS ディレクトリーになければなりません。
 - AliasMatch と指定する場合には、HTTP サーバー構成ファイルに AliasMatch を定義しておく必要があります。例えば、appletdir=/applets/ と指定する場合、HTTP サーバー構成ファイルに /applets/ の AliasMatch を定義しておかなければなりません。AliasMatch 規則の置換ファイル・パスは、サーバー上のディレクトリーの絶対パス名に設定する必要があります。

例:

```
AliasMatch ^/applets/com/ibm/edmslod/(.*)$ /QIBM/UserData/OnDemand/www/applets/$1
```

```
AliasMatch ^/applets/(.*)$ /QIBM/UserData/OnDemand/www/applets/$1
```

2. 指定したディレクトリーの許可を検査してください。ODWEK プログラムを実行するプロセスでは、アプレットのディレクトリーから読み取りを行う必要があります。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターは必須です。

例:

```
[CONFIGURATION]
APPLETDIR=applets
```

CACHEDIR

ODWEK が文書 (『CACHEDOCS』を参照) を一時保管する (キャッシュに入れる)、HTTP サーバー上のディレクトリーを指定するのに使用します。デフォルトでは、ODWEK は /QIBM/UserData/OnDemand/WWW/CACHE ディレクトリーにある文書をキャッシュに入れます。

注: 指定したディレクトリーの許可を检查してください。ODWEK プログラムを実行するプロセスでは、キャッシュ記憶機構のディレクトリーで書き込みや読み取りを行う必要があります。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[CONFIGURATION]
CACHEDIR=/QIBM/UserData/OnDemand/WWW/CACHE
```

CACHEDOCS

ODWEK が HTTP サーバー上に文書を一時保管する (キャッシュに入れる) かどうかを判別します。キャッシュ記憶機構を使用すると、以前に表示した文書をサーバーから取り出す処理が速くなります。デフォルトの値は 0 (ゼロ) で、文書に対してキャッシュ記憶機構を使用できないことを意味します。文書に対するキャッシュ記憶機構の使用を可能にするためには、1 を指定します。文書に対するキャッシュ記憶機構の使用を可能にする場合は、ODWEK が文書のキャッシュを行うディレクトリーを調べ (『CACHEDIR』を参照)、キャッシュ記憶機構のために確保されているディスク・スペースを確認してください (18 ページの『CACHESIZE』を参照)。

注: Microsoft® Internet Explorer ブラウザー、および AFP Web Viewer または Image Web Viewer を使用する場合、IBM では文書のキャッシュ記憶機構を常に使用可能にしておくことをお勧めしています。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。ただし、一般的には、ほとんどの環境で、常に文書のキャッシュ記憶機構を構成しておくことをお勧めします。

例:

```
[CONFIGURATION]
CACHEDOCS=1
```

CACHEMAXTHRESHOLD

ODWEK がキャッシュ記憶機構でデータや文書の削除を開始するタイミングを判別します。ODWEK は、キャッシュ記憶機構で使用されているディスク・スペースのパーセンテージが指定された値かそれ以上になったときに、データや文書の削除を開始します。デフォルトの値は 80 (パーセント) です。ODWEK は、古い項目から順に、しきい値に達するまでキャッシュ記憶機構内の項目を削除します (『CACHEMINTHRESHOLD』を参照)。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[CONFIGURATION]
CACHEMAXTHRESHOLD=80
```

CACHEMINTHRESHOLD

ODWEK がキャッシュ記憶機構でデータや文書の削除を停止するタイミングを判別します。ODWEK は、キャッシュ記憶機構で使用されているディスク・スペースのパーセンテージが指定された値かそれ以下になったときに、データや文書の削除を停止します。デフォルトの値は 40 (パーセント) です。ODWEK は、しきい値に達すると、古い項目から順に削除を開始します (『CACHEMAXTHRESHOLD』を参照)。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[CONFIGURATION]
CACHEMINTHRESHOLD=40
```

CACHESIZE

ODWEK が HTTP サーバー上にデータおよび文書を一時保管する (キャッシュに入れる) ために使用できるディスク・スペースを指定します。値はメガバイト単位で指定します。デフォルトの値は 10 (メガバイト) です。

注: 文書のキャッシュ記憶機構を使用可能にする方法については、17 ページの『CACHEDOCS』を参照してください。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。ただし、文書をキャッシュに入れる場合は、割り振るディスク・スペースが大きければ大きいほど、ODWEK が HTTP サーバー上に保管できる文書の数も増えます。一般的に、これにより、以前に表示された文書をユーザーに送る際の ODWEK の処理速度が向上します。

例:

```
[CONFIGURATION]
CACHESIZE=1024
```

CACHEUSERIDS

ODWEK がキャッシュ記憶機構のデータを使用してログオン・プロセスを完了する OnDemand ユーザー ID を、コンマで区切ったリストで指定します。ここでユーザー ID を指定すると、キャッシュ記憶機構にデータがない場合と、「非アクティブ・タイムアウト」値 (OnDemand サーバーのシステム・パラメーターを参照) に達した場合を除き、通常の OnDemand ログオン・プロセスをバイパスして複数のログオンが試行されます。各ユーザー ID は、コンマ文字で区切ります。

注:

1. サーバーでユーザー ID の大文字小文字が区別されている (OnDemand サーバーのシステム・パラメーターを参照) 場合は、OnDemand で定義されているユーザー ID を正確に指定する必要があります。
2. CACHEUSERIDS リストにリストされたユーザー ID では、キャッシュ記憶機構に名前やその他の情報が記憶されているフォルダーに対してのみアクセスが可能です。ユーザーが OnDemand サーバーにログオンした後で作成されたフォルダーに対して、ユーザーはアクセスできません。CACHEUSERIDS リストにリストされているユーザー ID で新しいフォルダーにアクセスできるようにするには、CACHEUSERIDS リストからそのユーザー名を削除するか、キャッシュの除去を行ってください。
3. ODWEK がすべての OnDemand ユーザーに対してキャッシュ記憶機構のデータを使用するように指定するには、CACHEUSERIDS=* を指定します。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[CONFIGURATION]
CACHEUSERIDS=user1,user2,user3
```

CODEPAGE

OnDemand データベースのコード・ページを識別します。デフォルトでは、ODWEK は HTTP サーバーのコード・ページを使用します。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。Logon API を使用すれば、_codepage パラメーターで指定されているコード・ページを指定変更することが可能です。

このパラメーターはオプションです。ただし、データベースと異なるコード・ページで HTTP サーバーが稼働している場合は、必ず CODEPAGE パラメーターを指定してください。

例:

```
[CONFIGURATION]
CODEPAGE=37
```

DOCSIZE

OnDemand サーバーから文書を取り出す際に、文書をまずディスクに書き込まず、メモリーに直接書き込むことができる文書の最大サイズ (バイト数) を決定します。指定した値以下の文書は、メモリーに直接書き込まれます。指定した値より大きい文書は、ディスクに書き込まれてから、文書がブラウザーに配布される前に、ディスクからメモリーに読み込まれます。値を小さくするとシステム資源が節約され、値を大きくすると表示のパフォーマンスが向上します。範囲は 0 (ゼロ) から n バイトまでで、 n は、システムで使用可能なメモリーの量です。値 0 は、サイズを 1 MB にデフォルト設定します。このパラメーターが指定されていないか、あるいは値が定義されていないまたは認識されない場合、サイズは 1 MB にデフォルト設定されます。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[CONFIGURATION]
DOCSIZE=524287
```

IMAGEDIR

ODWEK で使用するイメージ・ファイルが含まれるディレクトリーを識別します。

注:

1. ODWEK は、指定された値を HTML イメージ・タグにあるファイル名と連結させます。例えば、以下のような値を指定したとします。

```
imagedir=pictures
```

すると、View Document 関数の HTML イメージ・タグが、以下のような出力で示されます。

```
<IMG SRC="pictures/odic_vd.gif">
```

2. ディレクトリー名、または AliasMatch を指定することができます。

- ディレクトリー名を指定する場合、ディレクトリーは、/QIBM/UserData/OnDemand/WWW ディレクトリーを起点とするディレクトリーでなければなりません。例えば、imagedir=pictures を指定すると、イメージは /QIBM/UserData/OnDemand/WWW/PICTURES ディレクトリーの中になければなりません。
- AliasMatch 規則を指定する場合には、HTTP サーバー構成ファイルに AliasMatch を定義しておく必要があります。例えば、imagedir=/pictures/ と指定する場合、HTTP サーバー構成ファイルに /pictures/ の AliasMatch を定義しておかなければなりません。AliasMatch 規則は、サーバー上のディレクトリーの絶対パス名に設定する必要があります。次に例を示します。

```
AliasMatch ^/images/(.*)$ /QIBM/UserData/OnDemand/WWW/PICTURES/$1
```

3. 指定したディレクトリーの許可を検査してください。ODWEK プログラムを実行するプロセスでは、イメージ・ディレクトリーから読み取りを行う必要があります。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターは必須です。

例:

```
[CONFIGURATION]
IMAGEDIR=pictures
```

LANGUAGE

ODWEK がメッセージの表示に使用する言語を識別します。デフォルトの言語は英語 (ENU) です。ODWEK では、以下の言語がサポートされています。

値	地域
ARA	エジプト
CHS	中国
CHT	台湾
DAN	デンマーク
DEU	ドイツ
ENU	米国 (英語)
ESP	スペイン
FIN	フィンランド
FRA	フランス
FRC	カナダ
ITA	イタリア
JPN	日本
KOR	韓国
NLD	オランダ
NOR	ノルウェー
PTB	ブラジル
SVE	スウェーデン

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[CONFIGURATION]
LANGUAGE=JPN
```

ShowSearchString

自動検索機能がアクティブか、非アクティブかを決定します。自動検索機能は、Java 行データ・ビューアーからの行データ文書のトランザクションおよびテキスト検索をサポートします。自動検索機能は、ユーザーが指定した検索基準に一致する文書内の特定の行を自動的に検索し、強調表示します。

自動検索機能がアクティブにされている場合、ユーザーがトランザクション検索またはテキスト検索のいずれかを実行し、検索結果の文書リストから文書を開くと、システムは、指定された検索基準について文書のテキストを自動的に検索します。検索基準に一致するものが見つかり、検索基準が含まれている行が強調表示され、見つからなかった場合には、該当するメッセージが表示されます。ユーザーが別の文書を表示するために開くと（または、前に表示された文書を再度開くと）、検索が再度行われます。

自動検索機能をアクティブにするには、ShowSearchString パラメーターを 1 に設定します。自動検索機能を非アクティブにするには、ShowSearchString パラメーターを 0 (ゼロ) に設定します。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。指定されていない場合、デフォルト値は 0 (ゼロ; 非アクティブ) です。

例:

```
[CONFIGURATION]
ShowSearchString=1
```

TEMPDIR

ODWEK が一時ファイルを保管するディレクトリーは、この関数を使用して指定します。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。TEMPDIR パラメーターを指定しない場合、ODWEK はランタイム・ディレクトリーに一時ファイルを保管します。CGI プログラムを使用する場合、ランタイム・ディレクトリーは CGI プログラムがインストールされたディレクトリーです。サーブレットを使用する場合、ランタイム・ディレクトリーは、サーブレットを含むディレクトリーです。インストールによっては、ランタイム・ディレクトリーは java.exe ファイルのある場所である場合や、サーブレットのディレクトリーである場合もあります (ただし、正確な場所は Java アプリケーション・サーバーに依存)。

例:

```
[CONFIGURATION]
TEMPDIR=/QIBM/UserData/OnDemand/WWW/TMP
```

注: 指定したディレクトリーの許可を检查してください。ODWEK プログラムを実行するプロセスでは、一時ディレクトリーで書き込みや読み取りを行う必要があります。

TEMPLATEDIR

HTML テンプレート・ファイルが含まれているディレクトリーを識別します。ODWEK では、さまざまな製品関数 (Logon、Search、Retrieve Document など) に対する応答として Web ページを生成する際に、このテンプレート・ファイルを使用

します。デフォルトでは、ODWEK は /QIBM/UserData/OnDemand/WWW/SAMPLES ディレクトリーからテンプレート・ファイルを検索します。

注: 指定したディレクトリーの許可を検査してください。ODWEK プログラムを実行するプロセスでは、テンプレートのディレクトリーから読み取りを行う必要があります。

このパラメーターの有効範囲はグローバルです。また、CONFIGURATION セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[CONFIGURATION]
TEMPLATEDIR=/QIBM/UserData/OnDemand/WWW/SAMPLES
```

[SECURITY]

SECURITY セクションには、HTTP サーバーで ODWEK が使用するセキュリティーのパラメーターが含まれています。

このセクションの有効範囲はグローバルです。また、ARSWWW.INI ファイルの中で一度だけ指定します。

このセクションはオプションです。

このセクションでは、次のパラメーターを指定することができます。

REPORTSERVERTIMEOUT

このパラメーターを使用して、ODWEK が OnDemand サーバーの「非アクティブ・タイムアウト」パラメーターを使用するように指定します。「非アクティブ・タイムアウト」パラメーターによって、サーバーが非アクティブなユーザーとのセッションを終了することができる時点が決定されます。ODWEK が OnDemand サーバーの「非アクティブ・タイムアウト」パラメーターを使用するように指定するには、REPORTSERVERTIMEOUT を 1 に設定します。

このパラメーターの有効範囲はグローバルです。また、SECURITY セクションの中で一度だけ指定します。

このパラメーターはオプションです。REPORTSERVERTIMEOUT パラメーターを指定しない場合、ODWEK は「非アクティブ・タイムアウト」パラメーターを使用しません。したがって、この場合は、ODWEK は非アクティブなユーザーとのセッションを終了させません。「非アクティブ・タイムアウト」パラメーターの詳細については、管理クライアント用のオンライン・ヘルプを参照してください。

例:

```
[SECURITY]
REPORTSERVERTIMEOUT=1
```

SERVERACCESS

ODWEK がアクセスできる OnDemand サーバーをコマンドで区切ったリストで指定します。SERVERACCESS パラメーターを指定すると、ODWEK を使用するクライ

アントや API を使用するプログラムは指定されたサーバーにしかアクセスできなくなります。サーバーの TCP/IP アドレス、ホスト名の別名、および完全修飾ホスト名を指定することができます。

このパラメーターの有効範囲はグローバルです。また、SECURITY セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[SECURITY]
SERVERACCESS=dave,gunnar
```

[AFP2HTML]

AFP2HTML セクションには、AFP2WEB Transform で使用されるパラメーターが含まれます。AFP2WEB Transform では、AFP 文書や資源が AFP2HTML アプレットで表示可能な HTML 文書に変換されます。

注:

1. AFP 文書を HTML 文書に変換する場合、管理者は IBM から AFP2WEB Transform サービス・オフリングを入手し、これをサーバーにインストールおよび構成する必要があります。AFP2WEB Transform サービス・オフリングに関する詳細は、IBM 担当員にご相談ください。なお、組織内でも、AFP2WEB Transform の構成オプションを提供する必要があります。構成ファイルについての詳細は、143 ページの『付録 E. AFP の HTML への変換』を参照してください。
2. AFP2WEB Transform で文書の変換を行うためには、DEFAULT BROWSER セクション (または他のブラウザー・セクション) で AFPVIEWING=HTML パラメーターを指定する必要があります。詳細については、36 ページの『AFPVIEWING』を参照してください。(Retrieve Document API の使用を計画している場合は、_afp=HTML パラメーターを指定する必要があります。詳細については、87 ページの『Retrieve Document』を参照してください。)
3. デフォルトでは、ODWEK は変換された文書の表示に AFP2HTML アプレットを使用します。変換された文書がラージ・オブジェクトとして OnDemand に保管された場合、AFP2HTML アプレットでは、文書内でのページの移動を容易にするコントロールが使用できます。

このセクションの有効範囲はグローバルです。また、ARSWWW.INI ファイルの中で一度だけ指定します。

このセクションはオプションです。

このセクションでは、次のパラメーターを指定することができます。

CONFIGFILE

AFP 文書や資源を AFP2HTML アプレットで表示可能な HTML データ、フォント、およびイメージに変換する際に AFP2WEB Transform が使用するオプションが含まれる構成ファイルを指定します。

143 ページの『付録 E. AFP の HTML への変換』には、OnDemand に付属してい

るサンプルの構成ファイルが紹介されています。構成ファイルで指定できるオプションについての詳細は、AFP2WEB Transform の資料を参照してください。

このパラメーターの有効範囲はグローバルです。また、AFP2HTML セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[AFP2HTML]
CONFIGFILE=afp2html.ini
```

INSTALLDIR

AFP2WEB Transform プログラム、構成ファイル、およびマッピング・ファイルを置くディレクトリーを指定します。HTTP サーバー上のディレクトリーの絶対パス名を指定してください。

注: 指定したディレクトリーの許可を検査してください。ODWEK プログラムを実行するプロセスでは、インストール・ディレクトリーから読み取りを行う必要があります。

このパラメーターの有効範囲はグローバルです。また、AFP2HTML セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[AFP2HTML]
INSTALLDIR=/QIBM/UserData/OnDemand/www/bin
```

USEEXECUTABLE

ODWEK で AFP2WEB Transform を開始する際に共用ライブラリー (DLL) を使用するか、それとも実行可能 (EXE) を使用するかを決定します。

重要: iSeries 上の ODWEK では「実行可能」を使用しなければなりません。したがって、このパラメーターを常に 1 に設定する必要があります。

このパラメーターの有効範囲はグローバルです。また、AFP2HTML セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[AFP2HTML]
USEEXECUTABLE=1
```

[AFP2PDF]

AFP2PDF セクションには、IBM AFP2PDF Transform で使用されるパラメーターが含まれます。AFP2PDF Transform では、AFP 文書や資源を Adobe Acrobat ビューアーで表示可能な PDF 文書に変換します。

注:

1. AFP 文書を PDF 文書に変換する場合、管理者は IBM から AFP2PDF Transform サービス・オフリングを入手し、これを HTTP サーバーにインストールして、構成する必要があります。AFP2PDF Transform サービス・オフリングに関する詳細は、IBM 担当員にご相談ください。なお、組織内でも、AFP2PDF Transform の構成オプションを提供する必要があります。構成ファイルについての詳細は、147 ページの『付録 F. AFP の PDF への変換』を参照してください。
2. AFP2PDF Transform で文書の変換を行うためには、DEFAULT BROWSER (または他のブラウザ・セクション) で AFPVIEWING=PDF パラメーターを指定する必要があります。詳細については、36 ページの『AFPVIEWING』を参照してください。(Retrieve Document API の使用を計画している場合は、_afp=PDF パラメーターを指定する必要があります。詳細については、87 ページの『Retrieve Document』を参照してください。)
3. デフォルトでは、ODWEK は Adobe Acrobat ビューアーを使用して変換された文書を表示します。そのため、ご使用のブラウザに合ったビューアーを入手する必要があります。

このセクションの有効範囲はグローバルです。また、ARSWWW.INI ファイルの中で一度だけ指定します。

このセクションはオプションです。

このセクションでは、次のパラメーターを指定することができます。

CONFIGFILE

AFP 文書や資源を Adobe Acrobat ビューアーで表示可能な PDF 文書に変換する際に AFP2PDF Transform によって使用されたオプションを含む構成ファイルを指定します。147 ページの『付録 F. AFP の PDF への変換』には、OnDemand に付属しているサンプルの構成ファイルが紹介されています。構成ファイルで指定できるオプションについての詳細は、AFP2PDF Transform の資料を参照してください。

このパラメーターの有効範囲はグローバルです。また、AFP2PDF セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[AFP2PDF]
CONFIGFILE=afp2pdf.ini
```

INSTALLDIR

AFP2PDF Transform プログラム、構成ファイル、およびマッピング・ファイルを置くディレクトリーを指定します。HTTP サーバー上のディレクトリーの絶対パス名を指定してください。

注: 指定したディレクトリーの許可を検査してください。ODWEK プログラムを実行するプロセスでは、インストール・ディレクトリーから読み取りを行う必要があります。

このパラメーターの有効範囲はグローバルです。また、AFP2PDF セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[AFP2PDF]
INSTALLDIR=/QIBM/UserData/OnDemand/www/bin
```

USEEXECUTABLE

ODWEK で AFP2WEB Transform を開始する際に共有ライブラリー (DLL) を使用するか、それとも実行可能 (EXE) を使用するかを決定します。

重要: iSeries 上の ODWEK では「実行可能」を使用しなければなりません。したがって、このパラメーターを常に 1 に設定する必要があります。

このパラメーターの有効範囲はグローバルです。また、AFP2PDF セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[AFP2PDF]
USEEXECUTABLE=1
```

[MIMETYPES]

MIMETYPES セクションでは、OnDemand サーバーから取り出される文書の MIME (Multipurpose Internet Mail Extension) コンテンツ・タイプを識別します。ブラウザは、文書の形式設定と表示、文書をオープンするのに適切なアプレットやビューアーの選択、あるいは文書をオープンするためのユーザー定義プログラムの開始を行うために、この MIME コンテンツ・タイプを使用します。

注:

1. MIMETYPES セクションでは、OnDemand サーバーから取り出すことを計画している文書のタイプごとに、parameter=value のペアを指定する必要があります。parameter は、OnDemand 内の文書のデータ・タイプを識別します。(これは、「ビュー情報」ページで OnDemand アプリケーションに割り当てられるデータ・タイプです。) value は、文書をオープンするために開始するプログラムを判別します。value では大文字小文字が区別されます。
2. サンプル ARSWWW.INI ファイル (43 ページの『サンプルの ARSWWW.INI ファイル』を参照) の MIMETYPES セクションでは、OnDemand でサポートされている標準のデータ・タイプ (AFP、BMP、EMAIL、GIF、JFIF、LINE、PCX、PDF、および TIFF) のパラメーターがそれぞれ指定されています。
3. OnDemand では、標準のデータ・タイプに加えてユーザー定義のデータ・タイプもサポートされています。ユーザー定義のデータ・タイプには、システムに保管したい他の任意のデータ・タイプを定義することができます。ユーザー定義のデータ・タイプを持つ文書を表示するためには、あらかじめ MIMETYPE セクションにパラメーターを追加しておく必要があります。このパラメーターでは、「ビュー情報」ページで OnDemand アプリケーション用に指定されたデータの MIME コンテンツ・タイプとファイル拡張子を識別しなければなりません。な

お、ファイル拡張子もクライアントのオペレーティング・システムに登録しておく必要があります。例えば、OnDemand 内で、Lotus® WordPro 文書を保管するようにアプリケーションを定義するとします。アプリケーションの「ビュー情報」ページで、ファイル拡張子を LWP と指定します。このアプリケーションから取り出される文書を認識するようにシステムを構成するため、以下のパラメーターを ARSWWW.INI ファイルに追加します。

```
[MIMETYPES]
LWP=application/vnd.lotus-wordpro
```

こうして、ユーザーがアプリケーションから文書を取り出すと、ODWEK は MIME コンテンツ・タイプを application/vnd.lotus-wordpro に設定し、システムは Lotus WordPro を開始して文書をオープンします。Netscape の場合、MIME コンテンツ・タイプを「プリファレンス」->「ナビゲーター」->「アプリケーション」で定義する必要があります。

表 1 には、PC アプリケーション別に対応した MIME コンテンツ・タイプをリストします。

表 1. PC アプリケーション別の MIME コンテンツ・タイプ

アプリケーション	MIME コンテンツ・タイプ
Lotus アプリケーション	WK1=application/vnd.lotus-1-2-3 WK3=application/vnd.lotus-1-2-3 WK4=application/vnd.lotus-1-2-3 123=application/vnd.lotus-1-2-3 APR=application/vnd.lotus-approach VEW=application/vnd.lotus-approach LWP=application/vnd.lotus-wordpro SAM=application/vnd.lotus-wordpro MWP=application/vnd.lotus-wordpro SMM=application/vnd.lotus-wordpro PRE=application/vnd.lotus-freelance PRZ=application/vnd.lotus-freelance
Microsoft アプリケーション	DOC=application/msword XLS=application/vnd.ms-excel PPS=application/vnd.ms-powerpoint PPT=application/vnd.ms-powerpoint MPD=application/vnd.ms-project MPP=application/vnd.ms-project MPT=application/vnd.ms-project MPD=application/vnd.ms-project
HTML アプリケーション	HTML=application/html HTM=application/htm

このセクションの有効範囲はグローバルです。また、ARSWWW.INI ファイルの中で一度だけ指定します。

このセクションはオプションです。

このセクションでは、次のパラメーターを指定することができます。

AFP

[DEFAULT BROWSER] セクションで AFPVIEWING=NATIVE が指定された場合の、AFP 文書の MIME コンテンツ・タイプを指定します。詳しくは、36 ページの『AFPVIEWING』を参照してください。ここで指定される文書の MIME タイプは、文書の処理に使用するプログラムを決定するためにブラウザで使用されません。

このパラメーターの有効範囲はグローバルです。また、MIMETYPES セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[MIMETYPES]
AFP=application/afp
```

BMP

BMP 文書の MIME コンテンツ・タイプを指定します。デフォルトでは、BMP 文書は Image Web Viewer で表示されます。

このパラメーターの有効範囲はグローバルです。また、MIMETYPES セクションの中で一度だけ指定します。

このパラメーターはオプションです。ただし、このパラメーターが指定されない場合は MIME コンテンツ・タイプが image/bmp に設定され、クライアントのオペレーティング・システムで BMP ファイル・タイプに関連付けられているプログラムが開始されます。

例:

```
[MIMETYPES]
BMP=image/IBM-OnDemand
```

GIF

GIF 文書の MIME コンテンツ・タイプを指定します。デフォルトでは、GIF 文書は Image Web Viewer で表示されます。

このパラメーターの有効範囲はグローバルです。また、MIMETYPES セクションの中で一度だけ指定します。

このパラメーターはオプションです。ただし、このパラメーターが指定されない場合は MIME コンテンツ・タイプが image/gif に設定され、ブラウザに組み込まれているビューアーが GIF 文書の表示に使用されます。

例:

```
[MIMETYPES]
GIF=image/IBM-OnDemand
```

EMAIL

EMAIL 文書の MIME コンテンツ・タイプを指定します。クライアントに送信する前に EMAIL 文書に行う処理については、37 ページの『EMAILVIEWING』を参照してください。

注:

1. EMAIL 文書を HTML に変換すると、MIME コンテンツ・タイプは text/html に設定されます。ODWEK は EMAIL パラメーターが指定されていてもその値を無視します。
2. OnDemand から EMAIL 文書を抽出して圧縮解除する場合、ODWEK は EMAIL パラメーターの値を使用して文書のオープンに使用するプログラムを判別します。

このパラメーターの有効範囲はグローバルです。また、MIMETYPES セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[MIMETYPES]
EMAIL=text/plain
```

JFIF

JFIF (JPEG) 文書の MIME コンテンツ・タイプを指定します。デフォルトでは、JFIF 文書は Image Web Viewer で表示されます。

このパラメーターの有効範囲はグローバルです。また、MIMETYPES セクションの中で一度だけ指定します。

このパラメーターはオプションです。ただし、このパラメーターが指定されない場合は MIME コンテンツ・タイプが image/jpeg に設定され、クライアントのオペレーティング・システムで JPEG ファイル・タイプに関連付けられているプログラムが開始されます。

例:

```
[MIMETYPES]
JFIF=image/IBM-OnDemand
```

LINE

行データ文書の MIME コンテンツ・タイプを指定します。クライアントに送信する前に行データ文書に行う処理については、38 ページの『LINEVIEWING』を参照してください。

このパラメーターは、[DEFAULT BROWSER] セクションで LINEVIEWING=NATIVE が指定された場合に使用されます。OnDemand から行データ文書を抽出して圧縮解除する場合、ODWEK は LINE パラメーターの値を使用して文書のオープンを開始するプログラムを判別します。

このパラメーターの有効範囲はグローバルです。また、MIMETYPES セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[MIMETYPES]
LINE=text/html
```

PCX

PCX 文書の MIME コンテンツ・タイプを指定します。デフォルトでは、PCX 文書は Image Web Viewer で表示されます。

このパラメーターの有効範囲はグローバルです。また、MIMETYPES セクションの中で一度だけ指定します。

このパラメーターはオプションです。ただし、このパラメーターが指定されない場合は MIME コンテンツ・タイプが image/pcx に設定され、クライアントのオペレーティング・システムで PCX ファイル・タイプに関連付けられているプログラムが開始されます。

例:

```
[MIMETYPES]
PCX=image/IBM-OnDemand
```

PDF

PDF 文書の MIME コンテンツ・タイプを指定します。

注:

1. ODWEK は PDF パラメーターの値を使用して PDF 文書のオープンを開始するプログラムを判別します。デフォルトでは、PDF 文書は Adobe Acrobat ビューアーでオープンされます。
2. PDF 文書を表示するためには、ご使用のブラウザに合った Adobe Acrobat ビューアーを入手し、インストールする必要があります。

このパラメーターの有効範囲はグローバルです。また、MIMETYPES セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[MIMETYPES]
PDF=application/pdf
```

TIFF

TIFF 文書の MIME コンテンツ・タイプを指定します。デフォルトでは、TIFF 文書は Image Web Viewer で表示されます。

このパラメーターの有効範囲はグローバルです。また、MIMETYPES セクションの中で一度だけ指定します。

このパラメーターはオプションです。ただし、このパラメーターが指定されない場合は MIME コンテンツ・タイプが image/tiff に設定され、クライアントのオペレーティング・システムで TIFF ファイル・タイプに関連付けられているプログラムが開始されます。

例:

```
[MIMETYPES]
TIFF=image/IBM-OnDemand
```

[ATTACHMENT IMAGES]

ATTACHMENT IMAGES セクションは、文書への添付ファイルを表示する際に ODWEK が使用するイメージ・ファイルを識別します。それぞれのイメージ・ファイルには、添付ファイルの特定のタイプを表すアイコンが含まれています。例えば、テキストの添付ファイルやビットマップの添付ファイルなどを表すアイコンが含まれているイメージ・ファイルがあります。

注:

1. 指定するそれぞれのパラメーターでは、オペレーティング・システムが添付ファイルのタイプに関連付けているファイル・タイプを識別する必要があります。オペレーティング・システムが添付ファイルを処理するために開始するプログラムは、このファイル・タイプによって判別されます。例えば、オペレーティング・システムでテキスト添付ファイルにファイル・タイプ `TXT` が関連付けられている場合は、ATTACHMENT IMAGES セクションに `TXT=value` パラメーターを追加します。`value` の部分には、文書に対するテキストの添付ファイルを示すために使用するアイコンが含まれているファイルの名前を指定します。ユーザーがアイコンをクリックすると、オペレーティング・システムは `TXT` 文書をオープンするように登録されているプログラムを起動します。
2. デフォルトでは、文書への添付ファイルはすべて `odic_att.gif` ファイル (このファイルは、CONFIGURATION セクションの `IMAGEDIR` パラメーターで指定されたディレクトリーにあります) に示されます。加えて、OnDemand では、ATTACHMENT IMAGES セクションでパラメーターが指定されていないすべてのファイル・タイプに `odic_att.gif` ファイルを使用します。

このセクションの有効範囲はグローバルです。また、`ARSWWW.INI` ファイルの中で一度だけ指定します。

このセクションはオプションです。

このセクションでは、次のパラメーターを指定することができます。

BMP

このパラメーターは、ビットマップ・データ・タイプを識別します。パラメーターの値は、文書に添付されたビットマップ・イメージを表すアイコンが含まれているファイルを識別します。

このパラメーターの有効範囲はグローバルです。また、ATTACHMENT IMAGES セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[ATTACHMENT IMAGES]
BMP=userBitMap.gif
```

GIF

このパラメーターは、GIF データ・タイプを識別します。パラメーターの値は、文書に添付された GIF イメージを表すアイコンが含まれているファイルを識別します。

このパラメーターの有効範囲はグローバルです。また、ATTACHMENT IMAGES セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[ATTACHMENT IMAGES]
GIF=userGIF.gif
```

TXT

このパラメーターは、TXT データ・タイプを識別します。パラメーターの値は、文書に添付されたテキスト・ファイルを表すアイコンが含まれているファイルを識別します。

このパラメーターの有効範囲はグローバルです。また、ATTACHMENT IMAGES セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[ATTACHMENT IMAGES]
TXT=userText.gif
```

[NO HTML]

NO HTML セクションのパラメーターは、区切り文字付き ASCII 出力においてストリングを区切り、値のリストを区切るデフォルト文字を指定変更するために使用されます。関数は、_nohtml パラメーターが 1 に設定されている場合に区切り文字付き ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。

このセクションの有効範囲はグローバルです。また、ARSWWW.INI ファイルの中で一度だけ指定します。

このセクションはオプションです。

このセクションでは、次のパラメーターを指定することができます。

BEGIN

ODWEK がストリングや一連の値の先頭を区切るために使用する文字を指定します。ストリングにデフォルトの文字 ([文字) が含まれている場合は、BEGIN 区切り文字を変更する必要があります。

このパラメーターの有効範囲はグローバルです。また、NO HTML セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[NO HTML]
BEGIN=<
```

END

ODWEK がストリングや一連の値の末尾を区切るために使用する文字を指定します。ストリングにデフォルトの文字 (] 文字) が含まれている場合は、END 区切り文字を変更する必要があります。

このパラメーターの有効範囲はグローバルです。また、NO HTML セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[NO HTML]
END=>
```

SEPARATOR

ODWEK が一連の値を区切るために使用する文字を指定します。ストリングにデフォルトの文字 (\ 文字) が含まれている場合は、SEPARATOR 区切り文字を変更する必要があります。

このパラメーターの有効範囲はグローバルです。また、NO HTML セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[NO HTML]
SEPARATOR=;
```

[DEFAULT BROWSER]

| DEFAULT BROWSER セクションを使用して、組織で使用するブラウザのパラメ
| ターを指定することができます。42 ページの『[browser]』に記載されているよう
| に、指定したパラメーターは、特定のブラウザ・セクションで同じものを指定し
| ない限り、使用されます。(ブラウザ・セクションで指定したパラメーターは、
| DEFAULT BROWSER セクションでのパラメーターを指定変更します。)

このセクションの有効範囲は、すべてのブラウザに対してグローバルです。また、ARSWWW.INI ファイルの中で一度だけ指定します。

このセクションはオプションです。

このセクションでは、次のパラメーターを指定することができます。

ADDEXTENSION

文書のファイル拡張子 (3 文字) を、ブラウザに戻される URL の追加パス情報に追加するかどうかを判別します。ファイル拡張子を URL に追加すると、文書を開始するための正しいビューアーをブラウザが判別します。デフォルト値は 0 (ゼロ) で、これはファイル拡張子が URL に追加されないことを意味します。

注: Microsoft Internet Explorer ブラウザーを使用する場合、IBM では、ファイル拡張子が URL に追加されるように、ADDEXTENSION=1 を指定することをお勧めしています。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
ADDEXTENSION=1
```

ADDFIELDSTODOCID

フィールド値を文書 ID に追加するかどうかを判別します。(文書 ID は、Document Hit List 関数によって戻されます。) デフォルト値は 0 (ゼロ) で、これはフィールド値が文書 ID に追加されないことを意味します。ODWEK がフィールド値を文書 ID に追加できるようにすると、それらはシステム・ログにも現れます。ただし、アプリケーション・グループ・メッセージをシステム・ログに保管するようにシステムを構成していることが前提です。

注:

1. Update Document 関数を使用する場合、ADDFIELDSTODOCID=1 を指定しなければなりません。
2. 「Annotation Flags in the document database table field」が Yes に設定されている場合は、ADDFIELDSTODOCID=1 を指定しなければなりません。
OnDemand アプリケーション・グループ定義の「一般」ページから、「データベース情報」ダイアログ・ボックスの「Annotations Flags in document database table field」を設定することができます。(「拡張 (Advanced)」をクリックして、「データベース情報」ダイアログ・ボックスを開きます)

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
ADDFIELDSTODOCID=1
```

ADDNOTES

注釈を文書に追加できるかどうかを判別します。使用可能にすると、ODWEK は文書リスト内の各文書の横に、注釈の追加に使用するコントロールを配置します。デフォルト値は 0 (ゼロ) で、これは注釈を文書に追加できないことを意味します。

注: OnDemand アプリケーション・グループの Annotation 許可に基づいて、文書に注釈を追加する機能が許可または否認されます。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
ADDNOTES=1
```

AFPVIEWING

OnDemand サーバーから AFP 文書を取り出す場合、クライアントへ文書を送る前に ODWEK が行うアクションがあれば、このパラメーターの値によってそのアクションを判別します。例えば、AFP2WEB Transform を使用して AFP 文書を HTML に変換し、AFP2HTML アプレットを使用して HTML 出力を表示する場合があります。そのような場合、AFPVIEWING=HTML を指定すれば、ODWEK は AFP 文書を変換してから、それをクライアントに送ります。

パラメーターは以下の値のいずれかに設定することができます。

ASCII ODWEK は AFP 文書を ASCII テキストに変換します。

HTML ODWEK は AFP2WEB Transform を使用して、AFP 文書を HTML 文書に変換します。

NATIVE ODWEK は AFP 文書とそれらの資源を OnDemand から抽出し、圧縮解除します。

注: AFPVIEWING=NATIVE を指定する場合、AFP 文書の MIME コンテンツ・タイプで、使用するビューアーが識別されているかどうか検査してください。詳細については、27 ページの『[MIMETYPES]』を参照してください。

PDF ODWEK は AFP2WEB Transform を使用して、AFP 文書を PDF 文書に変換します。

注: AFPVIEWING=PDF を指定する場合、PDF 文書の MIME コンテンツ・タイプで、使用するビューアーが識別されているかどうか検査してください。詳細については、27 ページの『[MIMETYPES]』を参照してください。

PLUGIN ODWEK は AFP 文書の変換を行いません (デフォルト)。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。Retrieve Document 関数を使用する場合、_afp パラメーターを設定して、指定したアクションを指定変更することができます。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
AFPVIEWING=PLUGIN
```

AUTODOCRETRIEVAL

照会に一致する文書が 1 つだけである場合に、クライアントがその文書を自動的に表示するかどうかを指定します。つまりこの機能では、自分が把握している照会に一致する文書が 1 つしかない場合に、ユーザーがアクションを行わなくても、文書リストの Web ページがバイパスされて、その文書が表示されるように、システム

をセットアップすることができます。デフォルト値は 0 (ゼロ) で、これは、照会に一致する文書が 1 つしかない場合でも、ODWEK が文書リスト Web ページを表示することを意味します。

重要: Microsoft Internet Explorer ブラウザーを使用する場合、文書の自動取り出しを使用可能にしないでください。IBM では、Internet Explorer 用に定義するあらゆるブラウザー・セクションで、AUTODOCRETRIEVAL=0 を指定することをお勧めしています。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
AUTODOCRETRIEVAL=1
```

EMAILVIEWING

OnDemand サーバーから EMAIL 文書を取り出す場合、クライアントへ文書を送る前に ODWEK が行うアクションがあれば、このパラメーターの値によってそのアクションを判別します。

このパラメーターは以下の値のいずれかに設定することができます。

NATIVE ODWEK は OnDemand から EMAIL 文書を抽出し、圧縮解除します。

注: EMAIL=NATIVE を指定する場合、MIME コンテンツ・タイプで、使用するビューアーが識別されているかどうか検査してください。詳細については、27 ページの『[MIMETYPES]』を参照してください。

HTML ODWEK は EMAIL 文書を HTML 文書に変換します。これはデフォルト値です。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。Retrieve Document 関数を使用する場合、_email パラメーターを設定して、指定したアクションを指定変更することができます。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
EMAILVIEWING=HTML
```

ENCRYPTCOOKIES

ブラウザーに送信される Cookie を、ODWEK で暗号化するかどうかを判別します。デフォルト値は 0 (ゼロ) で、これは Cookie が暗号化されないことを意味します。ブラウザーに送信される Cookie を暗号化するには、1 を指定します。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
ENCRYPTCOOKIES=1
```

ENCRYPTURL

ブラウザに送信される URL に含まれるサーバー、ユーザー ID、パスワード、および文書 ID の値を、ODWEK で暗号化するかどうかを判別します。デフォルト値は 0 (ゼロ) で、これらの値が暗号化されないことを意味します。これらの値を暗号化するには、1 を指定します。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。ただし、ブラウザと HTTP サーバーとの間でフォームのパラメーターや値を転送するのに GET メソッドを使用する必要がある場合は、ENCRYPTURL=1 を指定してこれらの値を暗号化することができます。フォーム・タグのメソッド属性の詳細については、8 ページの『サーバーとデータのセキュリティ』を参照してください。

例:

```
[DEFAULT BROWSER]
ENCRYPTURL=1
```

FOLDERDESC

フォルダー記述を、フォルダー選択ページのフォルダー名の右側に表示するかどうかを指定します。デフォルト値は 0 (ゼロ) で、フォルダー記述を表示しないことを意味します。フォルダー記述を表示するには、1 を指定します。このパラメーターが指定されていないか、あるいは値が定義されていないまたは認識されない場合、フォルダー記述は表示されません。

このパラメーターは、ブラウザ・セクションで指定変更されない限り、グローバルな有効範囲を持ちます (42 ページの『[browser]』を参照してください)。このパラメーターは、DEFAULT BROWSER セクションで一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
FOLDERDESC=1
```

LINEVIEWING

OnDemand サーバーから行データ文書を取り出す場合、クライアントへ文書を送る前に ODWEK が行うアクションがあれば、このパラメーターの値によってそのアクションを判別します。

このパラメーターは以下の値のいずれかに設定することができます。

APPLET Line Data アプレットで表示するために、ODWEK は行データ文書を変換します (デフォルト)。

ASCII ODWEK は行データ文書を ASCII テキストに変換します。

NATIVE ODWEK は OnDemand から行データ文書を抽出し、圧縮解除します。

注: LINEVIEWING=NATIVE を指定する場合、MIME コンテンツ・タイプで、使用するビューアーが識別されているかどうか検査してください。詳細については、27 ページの『[MIMETYPES]』を参照してください。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。Retrieve Document 関数を使用する場合、_line パラメーターを設定して、指定したアクションを指定変更することができます。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
LINEVIEWING=APPLET
```

MAXHITS

文書リストに戻される項目の最大数 (照会に一致する項目の数とは関係がない)。

注:

1. 照会に一致する項目は、その項目がデータベースにロードされた順序で文書リストに入れられます。
2. ODWEK は以下で最初に指定された値を使用して、文書リストに戻す項目の数を判別します。
 - a. Document Hit List 関数での、「最大ヒット」フィールドの値 (「フォルダー許可」ページで指定される)。この値は他のすべての値を指定変更します。
 - b. Document Hit List および Print Document 関数での、_max_hits パラメーターの値 (関数で指定されている場合)。_max_hits パラメーターの値は、MAXHITS パラメーターを指定変更します。
 - c. MAXHITS パラメーターの値 (指定されている場合)。
 - d. 上記のいずれも指定されていない場合、ODWEK は最大で 200 個の項目を文書リストに戻します。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
MAXHITS=200
```

NOLINKS

文書の表示に使用するコントロールを、文書リストに含めるかどうかを判別します。使用可能にすると、ODWEK は各文書の横にコントロールを追加します。文書を表示する場合、ユーザーはこのコントロールを使用しなければなりません。デフ

オルト値は 0 (ゼロ) で、これはユーザーが文書を表示する際にテキスト・リンクを使用しなければならないことを意味します。

重要: Microsoft Internet Explorer ブラウザーを使用する場合、NOLINKS=0 を設定しなければなりません。IBM では、Internet Explorer 用に定義するあらゆるブラウザー・セクションで、NOLINKS=0 を指定することをお勧めしています。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
NOLINKS=1
```

ODApplet.jre.path.IE

60 ページの『Java 行データ・ビューアー』を参照してください。

ODApplet.jre.path.NN

60 ページの『Java 行データ・ビューアー』を参照してください。

ODApplet.jre.version

60 ページの『Java 行データ・ビューアー』を参照してください。

ODApplet.version

60 ページの『Java 行データ・ビューアー』を参照してください。

SERVERPRINT

サーバー・プリンターへの文書の送信に使用するコントロールを、文書リストに含めるかどうかを判別します。使用可能にすると、ODWEK は各文書の横にコントロールを追加します。デフォルト値は 0 (ゼロ) で、これは、ユーザーが文書をオープンしてからでなければ、その文書をサーバー・プリンターに送信できないことを意味します。

注:

1. サーバー印刷を使用するには、少なくとも 1 つ以上のサーバー・プリンターを OnDemand サーバーに定義していなければなりません。
2. OnDemand アプリケーション・グループの Print 許可に基づいて、文書を印刷する機能が許可または否認されます。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
SERVERPRINT=1
```

SERVERPRINTERS

このパラメーターを使用して、ユーザーにより選択可能なサーバー印刷装置のタイプを指定します。サーバー印刷装置には以下の 3 つのタイプがあります。

- P** サーバー・プリンター
- I** 通知付きサーバー・プリンター
- F** サーバー FAX

コンマで区切ったリストで、0 ～ 3 個のタイプを指定できます。

次の例では、サーバー・プリンターとサーバー FAX 装置をユーザーが選択できるということを指定する方法を示します。

```
[DEFAULT BROWSER]
SERVERPRINTERS=P,F
```

SHOWDOCLOCATION

HTML ではなく、区切り文字付き ASCII 出力を生成する場合（151 ページの『付録 H. 非 HTML 出力』を参照）に、文書の保管場所を出力に表示するかどうかを判別します。詳細については、153 ページの『Document Hit List』を参照してください。デフォルト値は 0（ゼロ）で、これは保管場所が出力に表示されないことを意味します。

注: 保管場所を表示するには、OnDemand フォルダーで Display Document Location プロパティも設定する必要があります。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
SHOWDOCLOCATION=1
```

VIEWNOTES

文書に対する注釈を表示できるかどうかを判別します。使用可能にすると、ODWEK は文書リスト内の各文書の横に、注釈の表示に使用するコントロールを配置します。デフォルト値は 0（ゼロ）で、これは注釈を表示できないことを意味します。

注: OnDemand アプリケーション・グループの Annotation 許可に基づいて、文書に対する注釈を表示する機能が許可または否認されます。

このパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中で一度だけ指定します。

このパラメーターはオプションです。

例:

```
[DEFAULT BROWSER]
VIEWNOTES=1
```

[browser]

組織で使用する特定のブラウザのオプションを指定することができます。browser セクションで指定するパラメーターは、ARSWWW.INI ファイルの DEFAULT BROWSER セクションからのパラメーターを指定変更します。(browser セクションでパラメーターを指定しなければ、DEFAULT BROWSER セクションで指定したパラメーターが使用されます。)

注:

1. セクション・ヘッダーには、オプションを指定したいブラウザを識別するストリングを入れなければなりません。ODWEK は、使用されるブラウザを判別する HTTP_USER_AGENT 環境変数の値を抽出します。それから、ODWEK は ARSWWW.INI ファイルを検索して、値が一致するブラウザ・セクションを探します。ブラウザ・セクションが見つからない場合、ODWEK は以下のセクションを探すために、ARSWWW.INI ファイルを検索します。

```
[browser version(major.minor)/platform]
```

```
[browser version(major.minor)]
```

```
[browser version(major)]
```

```
[browser]
```

```
[DEFAULT BROWSER]
```

ODWEK は、値が一致する最初のセクションからのオプションを使用します。

2. browser には、IE または Netscape を指定できます。
3. platform には、WinNT または Unix を指定できます。

browser セクションの有効範囲は、指定したブラウザでグローバルになります。各ブラウザごとに、1 つしか browser セクションは指定できません。DEFAULT BROWSER セクションから指定変更する必要があるパラメーターだけを指定してください。

このセクションはオプションです。

このセクションには、デフォルト・ブラウザ用に定義されているのと同じパラメーターを含めることができます。34 ページの『[DEFAULT BROWSER]』を参照してください。

例:

```
[IE 5]
AUTODOCRETRIEVAL=0
NOLINKS=0
```

```
[Netscape 4.7]
AUTODOCRETRIEVAL=1
NOLINKS=1
```

[DEBUG]

DEBUG セクションに含まれるオプションを使用して、組織内で ODWEK を使用する際の問題の解決に役立てることができます。DEBUG セクションは、arswww.ini ファイルで最初の実行可能ステートメントでなければなりません。

DEBUG セクションの有効範囲はグローバルです。また、ARSWWW.INI ファイルの中で一度だけ指定します。

このセクションはオプションです。

このセクションでは、次のパラメーターを指定することができます。

LOG

ODWEK がメッセージおよび他のプログラム情報をログ・ファイルに書き込めるようにします。(ログ・ファイルの名前は ARSWWW.LOG です。)

このパラメーターの有効範囲はグローバルです。また、DEBUG セクションの中で一度だけ指定します。

このパラメーターはオプションです。デフォルトでは、ODWEK はメッセージをログ・ファイルに書き込みません。メッセージをログに記録するには、1 を指定します。

LOGDIR

LOG パラメーターを使用してロギングが使用可能になった場合に ODWEK により ARSWWW.LOG ファイルが書き込まれるディレクトリーを判別します。

このパラメーターの有効範囲はグローバルです。また、DEBUG セクションの中で一度だけ指定します。

このパラメーターはオプションです。デフォルトでは、ロギングが使用可能な場合、ODWEK はログ・ファイルを /QIBM/UserData/OnDemand/WWW/LOG ディレクトリーに書き込みます。

例:

```
[DEBUG]
LOGDIR=/QIBM/UserData/OnDemand/WWW/LOG
LOG=1
```

サンプルの ARSWWW.INI ファイル

本製品には、サンプルの ARSWWW.INI 構成ファイルが付属しています。サンプルの構成ファイルは、サーバー、Web ブラウザー、およびビューアーで最も一般的に使用されるデフォルト値を設定します。

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; Server Configuration   ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;[DEBUG]
;log=1
;logdir=/QIBM/UserData/OnDemand/www/logs

[@SRV@_<host alias>];
```

```

HOST=<host name>;
PORT= ;
PROTOCOL= ;

[@SRV@_myiSeries.mycompany.com]
HOST=myiSeries.mycompany.com
PORT=1450
PROTOCOL=0

[CONFIGURATION]
CodePage=37
Language=ENU
TemplateDir=/QIBM/UserData/OnDemand/www/SAMPLES
ImageDir=/IMAGES/
AppletDir=/applets/
CacheDir=/QIBM/UserData/OnDemand/www/
CacheSize=0
CacheMinThreshold=0
CacheMaxThreshold=0
CacheDocs=0
CacheUserIDs=web,demo,mstephens

[SECURITY]
SERVERACCESS=

[AFP2HTML]
InstallDir=/QIBM/UserData/OnDemand/www/bin
ConfigFile=/QIBM/UserData/OnDemand/www/bin/afp2html.ini
UseExecutable=1

[AFP2PDF]
InstallDir=/QIBM/UserData/OnDemand/www/bin
ConfigFile=/QIBM/UserData/OnDemand/www/bin/afp2pdf.ini
UseExecutable=1

[MIMETYPES]
BMP=image/IBM_OnDemand
GIF=image/IBM_OnDemand
JFIF=image/IBM_OnDemand
PCX=image/IBM_OnDemand
TIFF=image/IBM_OnDemand
PNG=image/IBM_OnDemand
PDF=application/pdf
AFP=application/afp
LINE=application/line
EMAIL=text/html
META=application/unknown

[ATTACHMENT IMAGES]
TXT=userText.gif
BMP=userBitMap.gif
GIF=userGIF.gif

[NO HTML]
BEGIN=[
END=]
SEPARATOR=^

;;;;;;;;;;;;;;;;;;;;;;;;;
;;; Default Browser   ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;

[DEFAULT BROWSER]

;AfpViewing=[ascii,html,native,pdf,plugin,xenos]
AfpViewing=plugin
;LineViewing=[ascii,applet,native]

```

```
LineViewing=applet
;EmailViewing=[html,native]
EmailViewing=html
;MetaViewing=[xenos,native]
MetaViewing=native
NoLinks=1
ViewNotes=1
AddNotes=1
ServerPrint=1
ServerPrinters=P
AutoDocRetrieval=1
MaxHits=200
ShowDocLocation=1
```

```
[IE]
NoLinks=0
AddExtension=1
AddFieldsToDocid=0
```

次のステップ

ODWEK ソフトウェアをインストールし、ARSWWW.INI ファイルを構成したら、次に、サンプル・アプリケーションを構成してください。47 ページの『第 3 章 サンプル・アプリケーションの構成』を参照してください。

第 3 章 サンプル・アプリケーションの構成

この章では、ODWEK に付属している以下のサンプル・アプリケーションをカスタマイズする方法について説明します。

- LOGON.HTM。このアプリケーションは、いくつかのフォルダーに対するアクセスを許可されているユーザーをサポートします。各ユーザーは OnDemand サーバーに定義されます。サーバーへのログオン後、ODWEK はユーザーがオープンできるフォルダーのリストを表示します。
- CREDIT.HTM。このアプリケーションは、一時的な OnDemand の使用をサポートします。特定のフォルダーの検索基準が表示されます。OnDemand サーバー名、ユーザー ID とパスワード、フォルダー名およびフォルダー・フィールドが、アプリケーションでエンコードされます。『CREDIT.HTM』には、このアプリケーションのカスタマイズに関する指示が記載されています。

これらのサンプル・アプリケーションを変更したら、各ファイルの URL を公表して、ユーザーがそれらにリンクし、OnDemand にアクセスできるようにしてください。必要なカスタマイズの程度は、サンプルごとに異なります。この章には、サンプル・アプリケーションの 1 つをカスタマイズするための完全な指示を記載します。この指示をガイドとして使用し、必要に応じて他のアプリケーションをカスタマイズしてください。

注: IBM では、サンプル・アプリケーションの変更の他に、お客様の組織に合わせて TEMPLATE.HTM ファイルをカスタマイズすることもお勧めしています。TEMPLATE.HTM ファイルには、ODWEK が Web ページを表示する際に使用する、ユーザー定義の内容が含まれます。このファイルの変更についての詳細は、49 ページの『TEMPLATE.HTM』を参照してください。

LOGON.HTM

1. インストール・ディレクトリー (/QIBM/ProdData/OnDemand/www.samples) から HTTP サーバーの文書ルート・ディレクトリーに logon.htm ファイルをコピーします。
2. CGI プログラムの場合は、logon.htm ファイルに以下の行が含まれていることを確認してください。

```
<h4>Please enter your logon information:</h4>  
<FORM METHOD=POST ACTION="/arswww.cgi">
```

3. サブレットの場合は、logon.htm ファイルに以下の行が含まれていることを確認してください。

```
<FORM METHOD=POST ACTION="/ArsWWWServlet">
```

CREDIT.HTM

CREDIT.HTM サンプル・アプリケーションをカスタマイズする場合、ユーザーにアクセスを許可するフォルダーごとに、ファイルのコピーを作成します。ファイルの名前は、フォルダーの名前と同じにしなければなりません。

1. CREDIT.HTM ファイルを編集します (デフォルトでは、このファイルは /QIBM/UserData/OnDemand/WWW/SAMPLES ディレクトリーにあります)。
2. <body> ステートメント (行 11) で指定されている背景イメージを変更または削除します。
3. <body> ステートメント (行 11) で指定されている背景色を変更します (オプション)。
4. ステートメント (行 12) で指定されている製品イメージを変更または削除します。
5. <h1> ステートメント (行 15) で指定されているフォルダー名を置き換えます。
6. <p> ステートメント (行 17 ~ 25) で指定されているテキストを置き換えます。ユーザーに対する一般的な指示を入力します。
7. <FORM> ステートメント (行 29) で指定されている CGI-BIN ディレクトリー名を置き換えます。HTTP サーバー上の ODWEK プログラムおよびファイルが含まれている CGI-BIN ディレクトリーの名前を入力します。
8. <input> ステートメント (行 30) で指定されている値を置き換えます。これはコンマで区切ったストリングで、フォルダー表示フィールドの名前を含んでいます。
9. <input> ステートメント (行 31) で指定されている値を置き換えます。これはフォルダーの名前です。
10. <input> ステートメント (行 33) で指定されている値を置き換えます。これは文書リストに表示される項目の最大数であり、照会に一致する項目の数とは関係がありません。
11. <input> ステートメント (行 35) で指定されているサーバー名を置き換えます。これは、ODWEK が通信する OnDemand サーバーの名前です。指定されているサーバー名は gunnar です。
12. 文書リスト内の項目をソートする場合、<input> ステートメント (行 36) で指定されている値を検査します。それ以外の場合、行 36 は削除してください。
13. 文書リスト内の項目をソートする場合、<input> ステートメント (行 37) で指定されている値を検査します。それ以外の場合、行 37 は削除してください。
14. <input> ステートメント (行 38) で指定されている値を置き換えます。これは OnDemand ユーザー ID です。指定するユーザー ID には、フォルダーのオープンおよびアプリケーション・グループ・データへのアクセスを行うための許可を与えておかなければなりません。
15. <input> ステートメント (行 39) で指定されているテンプレート・ファイルの名前を変更します (オプション)。OnDemand はそのテンプレート・ファイルを使用して、以降の Web ページを生成します。指定されているテンプレート名は template.htm です。
16. 最初のフォルダー検索フィールドの行 40 ~ 43 を変更します。
 - a. ステートメントにフォルダー・フィールドの名前を入力します。
 - b. <input> ステートメントの名前フィールドに指定されている値を、実際のフォルダー・フィールド名に置き換えます。
 - c. <input> ステートメントの値フィールドに指定されている値を、デフォルトの検索値に置き換えます。

17. 行 40 ~ 43 をコピーし、別のフォルダー検索フィールドについてもステップ 16 を繰り返します。
18. 変更内容を保管し、テキスト・エディターをクローズします。

TEMPLATE.HTM

TEMPLATE.HTM ファイルは、さまざまな製品関数 (Logon など) への応答として、ODWEK が Web ページを生成する際に使用するデフォルトのテンプレート・ファイルです。このファイルは、ユーザー定義の内容を含んでいるファイルに置き換えなければなりません。ただし、テンプレート・ファイルには以下の HTML コメント行を含めなければなりません。

```
<!-- - - AOI# Marker - - -->
```

コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。

このテンプレート・ファイルは、デフォルトでは、ARSWWW.INI ファイル内の `TEMPLATEDIR` パラメーターで名前が指定されているディレクトリーにあります。詳細については、22 ページの『TEMPLATEDIR』を参照してください。

次のステップ

ODWEK ソフトウェアをインストールし、ARSWWW.INI ファイルを構成し、サンプル・アプリケーションを構成したら、次に、ユーザーのワークステーションに Web ビューアーをインストールする必要があります。51 ページの『第 4 章 Web ビューアーのインストール』を参照してください。

第 4 章 Web ビューアーのインストール

概要

IBM では、OnDemand から取り出せる標準的なタイプの文書を表示できるビューアーを提供しています。インストール要件は、組織内のユーザーが使用する必要があるビューアーによって異なります。

- IBM では、行データ文書を表示する場合は、Line Data アプレットを使用することをお勧めしています。Line Data アプレットは HTTP サーバーに保管されます。Line Data アプレットを使用可能にすると、行データ文書を表示するように選択した場合に、このアプレットがワークステーション上のメモリーに自動的にロードされます。ARSWWW.INI ファイル内の LINEVIEWING パラメーターで、ユーザーが使用するビューアーが指定されているかどうか検査してください。
- AFP 文書を表示するには、IBM OnDemand AFP Web Viewer、AFP2HTML アプレット、または Adobe Acrobat ビューアーを使用することができます。
 - IBM OnDemand AFP Web Viewer を使用して AFP 文書を表示するには、ユーザーのワークステーションにそのビューアーをインストールする必要があります。
 - AFP2HTML アプレットを使用して AFP 文書を表示する場合、管理者は HTTP サーバーで AFP2WEB Transform のインストールと構成を行い、ARSWWW.INI ファイルを構成しなければなりません。AFP2HTML アプレットは HTTP サーバーに保管されます。管理者が AFP2HTML アプレットを使用可能にすると、AFP 文書を表示するように選択した場合に、このアプレットがワークステーション上のメモリーに自動的にロードされます。
 - Adobe Acrobat ビューアーを使用して AFP 文書を表示する場合、管理者は HTTP サーバーで AFP2PDF Transform のインストールと構成を行い、ARSWWW.INI ファイルを構成しなければなりません。管理者がこの Transform を使用可能にすると、AFP 文書を表示するように選択した場合に、デフォルトで、ブラウザーが Adobe Acrobat ビューアーを開始させようとし、ユーザーは、ワークステーション上で Adobe Acrobat ビューアーの入手とインストールを行う必要があります。

ARSWWW.INI ファイル内の AFPVIEWING パラメーターで、ユーザーが使用するビューアーが指定されているかどうか検査してください。

- IBM では、BMP、GIF、JPEG、PCX、および TIFF 文書を表示するために、ユーザーが IBM OnDemand Image Web Viewer をワークステーションにインストールすることをお勧めしています。その他の場合は、これらのタイプの文書を処理する別のビューアーを使用する必要があります。(例えば、ほとんどのビューアーには、GIF および JPEG を表示する機能をもった組み込みビューアーが備わっています。) ユーザーが他のビューアーを使用することにした場合、それらのタイプの文書に合わせて、管理者はデフォルトの MIME コンテンツ・タイプを必ず変更してください。ARSWWW.INI ファイルの MIMETYPES セクション内のパラメーターで、ユーザーが使用するビューアーが指定されているかどうか検査してください。

注:

1. 他のタイプのデータを表示する場合、他のビューアーをインストールしなければならない場合があります。例えば、OnDemand サーバーから取り出した PDF 文書を表示する場合、IBM では、ご使用のブラウザに合った Adobe Acrobat ビューアーを入手し、インストールすることをお勧めしています。
2. PDF 文書を表示するには、ブラウザのプラグイン・ディレクトリー内に nppdf32.dll ファイルが必要です。Internet Explorer の場合は、¥Program Files¥Internet Explorer¥PLUGINS ディレクトリー内に、Netscape の場合は、¥Program Files¥Netscape¥Communicator¥Program¥Plugins ディレクトリー内になければなりません。ブラウザのディレクトリーにこのファイルがない場合は、Adobe ソフトウェアを再インストールする必要があります。

要件

IBM により提供されるビューアーには、Netscape Navigator 4.7 以降、または Microsoft Internet Explorer 5.5 以降が必要です。

ODWEK を使用するためには、PC に Cookie データを書き込むことができなければなりません。各ユーザーのブラウザが Cookie を受け付けるように構成されていることを確認してください。

IBM では、アプレット・ディレクトリー内に以下の 2 つのバージョンの Java 行データ・ビューアーを提供しています。

ODLineDataViewer.jar は、ブラウザで Java サポートを必要とする、古い Java 行データ・ビューアーです。Java のサポートは、通常、Java 仮想マシン (JVM) によって提供されます。**注:** Microsoft Internet Explorer をご使用の場合、ユーザーはブラウザのカスタム・インストール・オプションを使用して JVM のインストールが必要になる場合があります。

ODLineDataViewer2.jar は、バージョン 1.4.1 以降の Java プラグインを必要とする、新しい Java 行データ・ビューアーです。新しい Java 行データ・ビューアーは、ブラウザで Java サポートを使用しません。ユーザーは、新しい Java 行データ・ビューアーを使用するには、PC に Java プラグインをインストールする必要があります。構成に関する重要な情報は、60 ページの『Java 行データ・ビューアー』を参照してください。

デフォルトの製品インストールでは、古い Java 行データ・ビューアーを使用します。

ブラウザは、Windows® 2000、Windows XP、または Windows Server 2003 の下で稼働させなければならず、以下のハードウェアおよびソフトウェアが必要です。

- ネットワークへの物理接続 (トークンリングまたはイーサネットのネットワーク・アダプターなど)
- Windows TCP/IP サポート
- 最低 256 MB の RAM
- Pentium® または Pentium 互換の 800 MHz 以上の処理速度のプロセッサ
- 最低 800 x 600 の解像度を持つ SVGA ディスプレイおよびアダプター
- 文書の表示用の最低 20 MB の空きディスク・スペース

- IBM OnDemand AFP Web Viewer を必要とするワークステーションごとに約 3 MB、および IBM OnDemand Image Web Viewer を必要とするワークステーションごとに約 2 MB。

インストール

注: AFP Web Viewer と共にユーザー定義ファイルを配布する場合、ユーザーが AFP Web Viewer のインストールを開始する前に、AFP Web Viewer インストール・ファイルを構成して、ユーザー定義ファイルを保持する必要があります。詳しくは、54 ページの『ユーザー定義ファイルの配布』を参照してください。

IBM により提供されるビューアーは、自己解凍ファイルを使用してインストールされます。これらのファイルをユーザーの Windows システムにダウンロードし、該当するビューアーのインストールを実行します。インストールの進行中にユーザーがブラウザを稼働させている場合、ビューアーを使用できるようにするには、ユーザーはブラウザを停止してから再始動する必要があります。以下のビューアー・ファイルが /QIBM/ProdData/OnDemand/www/plugins ディレクトリー内にあります。

- afpplgus.exe - IBM OnDemand AFP Web Viewer - 英語のみ
- afpplgin.exe - IBM OnDemand AFP Web Viewer - DBCS サポートを含む全言語
- afpplgin.zip - IBM OnDemand AFP Web Viewer - DBCS サポートを含む全言語用の Zip 形式
- imgplgin.exe - IBM OnDemand Image Web Viewer - 全言語

インストール・プロセスでは、ユーザーが指定したディレクトリーにビューアーおよびそれに関連するファイルがコピーされます。AFP Web Viewer の場合、ワークステーション上に約 3 MB のスペースが必要です。Image Web Viewer の場合、ワークステーション上に約 2 MB のスペースが必要です。ユーザーに対し、インストール・プロセス中にブラウザがアクティブになっている場合は、必ずブラウザを再始動するように注意を促してください。

注: インストール・プログラムは、プラグインまたは ActiveX コントロールのいずれかとしてビューアーをインストールします。Internet Explorer がワークステーションにインストールされている場合、インストール・プログラムは ActiveX コントロールをインストールします。Netscape がワークステーションにインストールされている場合、インストール・プログラムはプラグインをインストールします。Internet Explorer と Netscape の両方がワークステーションにインストールされている場合、インストール・プログラムは Internet Explorer 用に ActiveX コントロールをインストールし、Netscape 用にプラグインをインストールします。

ユーザー定義ファイルの配布

IBM が提供している IBM OnDemand AFP Web Viewer ソフトウェアを使用すれば、ユーザー定義ファイルを配布することができます。例えば、組織内のだれかが OnDemand に保管されている文書用の AFP フォント・ファイルを作成したとします。AFP Web Viewer ソフトウェアを使用すれば、そのフォント・ファイルを配布することができます。これにより、ユーザーが AFP 文書を表示したとき、その文書は適切なフォントで表示されます。

ユーザー定義ファイルを AFP Web Viewer で配布するには、そのファイルをインストール・ファイルにパッケージし、そのインストール・ファイルを共用位置に保管する必要があります。ユーザーがインストール・ファイルを実行すると、Setup プログラムによって自動的に AFP Web Viewer とユーザー定義ファイルがユーザーのワークステーションにインストールされます。

AFP Web Viewer を使用すれば、以下のタイプのユーザー定義ファイルを配布することができます。

- AFP フォント・ファイル。これらのファイルは、ワークステーション上の AFP Web Viewer の宛先ディレクトリーにある FONT サブディレクトリーにコピーされます。
- Adobe Type 1 フォント・ファイル。これらのファイルは、ユーザーが指定したディレクトリーにコピーされ、Setup プログラムによって ATM にインストールされます。
- TrueType フォント・ファイル。これらのファイルは、Windows FONTS ディレクトリーにコピーされ、Setup プログラムによって Windows にインストールされます。
- その他のユーザー定義ファイル。これらのファイルは、ワークステーション上の AFP Web Viewer の宛先ディレクトリーにコピーされます。

注: Setup プログラムは、IBM が提供する AFP Web Viewer ファイルの後で、ユーザー定義ファイルをワークステーションにコピーします。ユーザー定義ファイルの名前を IBM が提供するファイルの名前と同じにすると、ユーザー定義ファイルによって IBM 提供のファイルが置き換えられます。この機能は、更新された FLDPORT2.INI ファイルを配布したり、組織が変更した IBM AFP フォント・ファイルを配布したりするときなどに利用することができます。

以下のトピックでは、AFP Web Viewer の構成と配布についての詳細について説明します。

- IBM が提供している AFP Web Viewer のインストール
- ユーザー定義ファイルを保持するサブディレクトリーの追加
- サブディレクトリーへのユーザー定義ファイルの保管
- フォント・ファイルの構成
- AFP Web Viewer インストール・ファイルの構築
- ユーザーのワークステーションへの AFP Web Viewer のインストール

AFP Web Viewer ファイルのインストール

ほとんどのお客様では、AFP Web Viewer でユーザー定義ファイルを配布するかどうかによって、以下の 2 つの方法のいずれかを使用してサーバーからビューアー・ファイルを配布します。

- 標準インストール。IBM が提供している AFP Web Viewer ファイルを配布し、AFP Web Viewer でユーザー定義ファイルの配布準備をするために使用します。管理者が ODWEK ソフトウェアを HTTP サーバーにインストールすると、ビューアーのインストール・ファイルがサーバー上のディレクトリーに保管されます。ビューアーごとに 1 つのインストール・ファイル (EXE)、および AFP Web Viewer 用に 1 つの ZIP アーカイブ・ファイルが必要です。通常、管理者は、インストール・ファイルをサーバー上の共用ディレクトリーに移動して、そのファイルへのリンクを含む Web ページを作成します。ユーザーは、Web ページをブラウザにロードし、適切なインストール・ファイルへのリンクをアクティブにすることにより、ビューアーをインストールします。
- AFP Web Viewer のカスタム・インストール。AFP Web Viewer でユーザー定義ファイルを配布するために使用します。
 1. 標準インストール用にサーバーをセットアップします。
 2. ユーザーが実際にビューアーをインストールする前に、AFP Web Viewer ZIP アーカイブ・ファイルのコピーを入手します。
 3. ファイルを ZIP アーカイブ・ファイルから空の作業ディレクトリーに抽出します。
 4. サブディレクトリーを作業ディレクトリーに追加し、ユーザー定義ファイルをそのディレクトリーに保管します。詳細については、『サブディレクトリーの追加』および 56 ページの『ユーザー定義ファイルの保管』を参照してください。
 5. ユーザー定義の Adobe Type 1 フォント・ファイルを配布する場合は、フォント構成ファイルを作成します。詳細については、57 ページの『フォント・ファイルの構成』を参照してください。
 6. すべてのディレクトリーとファイルが構成された後で、配布用の自己解凍 EXE ファイルを作成します。詳細については、58 ページの『AFP Web Viewer インストール・ファイルの構築』を参照してください。
 7. 標準インストール用に IBM が提供している EXE ファイルを、構築した自己解凍 EXE ファイルで置き換えます。
 8. 管理者がステップ 1 ~ 7 を完了した後で、ユーザーは、Web ページをブラウザにロードし、更新されたインストール・ファイルへのリンクをアクティブにすることにより、AFP Web Viewer およびユーザー定義ファイルをインストールすることができます。

サブディレクトリーの追加

配布する予定のユーザー定義ファイルは、メインのクライアント・インストール・ディレクトリー下の CUSTOM サブディレクトリー・ツリーに保管しておく必要があります。例えば、メインのクライアント・インストール・ディレクトリーを ¥ONDEMAND¥AFP32 と名付けたとします。

ユーザー定義ファイルを保持するメインのクライアント・インストール・ディレクトリを構成するには、

1. メインのクライアント・インストール・ディレクトリの下に CUSTOM ディレクトリを作成します。

例:

```
¥ondemand¥afp32¥custom
```

注: CUSTOM ディレクトリは、ユーザーに配布するその他の¹ユーザー定義ファイルを保持します。 Setup プログラムは、このディレクトリのファイルを、ワークステーション上の AFP Web Viewer の宛先ディレクトリにコピーします。

2. 以下のサブディレクトリのうち 1 つまたは複数 を CUSTOM ディレクトリに追加します。追加するサブディレクトリは、ユーザーに配布するユーザー定義ファイルのタイプによって異なります。

- CUSTOM ディレクトリの下に、AFP フォント・ファイル (ファイル・タイプ FNT および MAP) を保持する FONT サブディレクトリを作成します。

例:

```
¥ondemand¥afp32¥custom¥font
```

Setup プログラムは、これらのファイルを、ワークステーション上の AFP Web Viewer の FONT ディレクトリにコピーします。

- CUSTOM ディレクトリの下に、Adobe Type 1 フォント・ファイル (ファイル・タイプ PFB および PFM) とフォント構成ファイルを保持する TYPEONE サブディレクトリを作成します。

例:

```
¥ondemand¥afp32¥custom¥typeone
```

Setup プログラムは、これらのファイルをユーザーが指定したディレクトリにコピーして、フォントを ATM にインストールします。

- CUSTOM ディレクトリの下に、Windows TrueType フォント・ファイル (ファイル・タイプ TTF) を保持する TRUETYPE サブディレクトリを作成します。

例:

```
¥ondemand¥afp32¥custom¥truetype
```

Setup プログラムは、このディレクトリのファイルを Windows FONT ディレクトリにコピーして、フォントを Windows にインストールします。

ユーザー定義ファイルの保管

IBM 提供のインストール・ファイルを作業ディレクトリに抽出し、CUSTOM ディレクトリを作成した後は、個々のサブディレクトリ内にユーザー定義ファイルを保管することができます。例えば、ユーザーに配布する Adobe Type 1 フォント・ファイル (ファイル・タイプ PFB および PFM) を、
¥ONDEMAND¥AFP32¥CUSTOM¥TYPEONE ディレクトリにコピーします。

1. AFP フォント・ファイル、Adobe Type 1 フォント・ファイル、および Windows TrueType フォント・ファイル以外のもの。

フォント・ファイルの構成

ユーザー定義の Adobe Type 1 フォント・ファイルをユーザーに配布する予定の場合は、以下のステップを完了する必要があります。

1. ユーザー定義の Type 1 フォント・ファイル (ファイル・タイプ PFB および PFM) を、CUSTOM ディレクトリーの TYPEONE サブディレクトリーに保管します。詳しくは、55 ページの『サブディレクトリーの追加』を参照してください。
2. Type 1 フォント構成ファイルを作成します。以降の部分で、Type 1 フォント構成ファイルを作成する方法について説明します。

Type 1 フォント構成ファイルは、ATM_INI.CFG という名前でなければならず、CUSTOM ディレクトリーの TYPEONE サブディレクトリーに保管する必要があります。配布ディレクトリーについての詳細は、55 ページの『サブディレクトリーの追加』を参照してください。

Type 1 フォント構成ファイルの各レコード (行) は、ユーザーに配布する唯一のユーザー定義の Adobe Type 1 フォントを識別します。レコードの形式は以下のとおりです。

```
fontname=filename.PFM,filename.PFB
```

ここで、fontname は ATM Control Panel のフォント・リストに表示される Type 1 フォントの名前、filename.PFM はフォントの PFM ファイルの名前、filename.PFB はフォントの PFB ファイルの名前です。以下の例は、2 つのレコードのある Type 1 フォント構成ファイルを示しています。

```
Courier,BOLD=coub.pfm,coub.pfb  
SonoranSansSerif_36,BOLDITALIC=c0a175z0.pfm,c0a175z0.pfb
```

ファイルの最初のレコードは、Courier,BOLD という名前のフォントと、PFM フォント・ファイル coub.pfm および PFB フォント・ファイル coub.pfb を識別します。ファイルの 2 番目のレコードは、SonoranSansSerif_36,BOLDITALIC という名前のフォントと、PFM フォント・ファイル c0a175z0.pfm および PFB フォント・ファイル c0a175z0.pfb を識別します。

ユーザー定義の Adobe Type 1 フォント・ファイルを含む AFP Web Viewer インストール・ファイルをユーザーが実行すると、Setup プログラムは、以下のようにフォント・ファイルを処理します。

1. TYPEONE ディレクトリーで見つかったすべてのユーザー定義の Adobe Type 1 フォント・ファイル (ファイル・タイプ PFB および PFM) を、宛先ディレクトリーにコピーします。ユーザーは宛先ディレクトリーを指定します。
2. Type 1 フォント構成ファイル (ATM_INI.CFG) で識別された各フォントごとに 2 つのフォント・ファイルがコピーされたことを検査します。ワークステーションにコピーされたファイルの名前は、フォント構成ファイルで指定された名前と一致しなければなりません。

注: フォント構成ファイルで指定されたフォント・ファイルの名前が、ワークステーションにコピーされたファイルの名前と一致しない場合、Setup プログラムは警告メッセージを表示し、そのフォントをインストールしません。

3. ユーザーが指定した宛先ディレクトリーを使用して、PFB および PFM ファイルのパス情報を追加します。
4. フォントを ATM にインストールします。

AFP Web Viewer インストール・ファイルの構築

ディレクトリーを作成し、ファイルを CUSTOM ディレクトリー・ツリーに保管し終えたら、ユーザー定義ファイルと IBM 提供の AFP Web Viewer ファイルを含むインストール・ファイルを作成する必要があります。インストール・ファイルは、通常、Setup.exe という名前になっています。

配布用にファイルとアプリケーションを単一の自己解凍形式の AFP Web Viewer 実行可能ファイルにパッケージするためのソフトウェアを開発している会社もあります。例えば、InstallShield Software Corporation は、PackageForTheWeb という製品を提供しています。

注: IBM は他社が提供するソフトウェアをサポートしていません。

パッケージ用のソフトウェアを入手後それを実行し、提供される指示にしたがって、ユーザー定義ファイルと IBM 提供の AFP Web Viewer ファイルを含む AFP Web Viewer インストール・ファイルを作成します。

ユーザーのワークステーションへの AFP Web Viewer のインストール

CUSTOM ディレクトリー・ツリーをセットアップし、AFP Web Viewer インストール・ファイルを構築し、サーバー上の AFP Web Viewer インストール・ファイルを置き換えたら、ユーザーは AFP Web Viewer およびユーザー定義ファイルのインストールを開始することができます。ユーザーが次回、AFP Web Viewer インストール・ファイルへのリンクをサーバーからアクティブにするときに、Setup プログラムは AFP Web Viewer をユーザーのワークステーションにインストールし、AFP Web Viewer インストール・ファイルでパッケージされたすべてのユーザー定義ファイルをユーザーのワークステーションにコピーします。

AFP フォントのマッピング

文書を作成する際に使用する AFP フォントは、AFP プラグインの使用時に表示できるフォントにマップする必要があります。ODWEK には、IBM Core Interchange (Latin のみ) フォントと、互換フォントを TrueType フォントにマップするのに使用されるフォント定義ファイルが用意されています。フォント定義ファイルおよびフォント・マップ・ファイルは、AFP Web Viewer コードがある FONT サブディレクトリーに保管されています。

AFP Web Viewer に定義されていないフォントを文書で使用する場合、自身または組織内の他の人が IBM Core フォントを変更した場合、あるいは AFP フォントを作成した場合、AFP Web Viewer で文書を正しく表示するためには、フォント定義ファイルでフォントを定義しなければなりません。AFP フォントのマッピング方法、フォント定義ファイル、および AFP と TrueType フォントに関連する他の技術情報については、「AFP Workbench Technical Reference」を参照してください。

AFP Web Viewer

以下の設定を、サーバーの論理ビューから AFP Web Viewer に適用することができます。

- 背景色。以下の色がサポートされています。他の色はサポートされていません。

緑色のバー (白の背景で表示されます)

緑

赤

黄

黒

白

グレー

- イメージ・カラー。以下の色がサポートされています。他の色はサポートされていません。

黄

青

赤

マゼンタ

緑

シアン

デフォルト (黒で表示)

- ズーム。

注: 選択エリア・カラーは AFP Web Viewer には適用されません。選択エリアは、常に、黒の背景に白のテキストで表示されます。

Image Web Viewer

Image Web Viewer を使用して複数ページのイメージを表示する場合には、以下の情報が該当します。

注: 以下の手順では、コンピューターのレジストリーを編集する必要があります。レジストリーは、絶対に必要な場合以外は、編集してはなりません。レジストリーにエラーがあると、コンピューターが正しく機能しない可能性があります。作業を行う前に、レジストリーのバックアップ・コピーを作成して、最後にコンピューターを正常に始動した際に使用していたものと同じバージョンにレジストリーを復元する方法を十分に理解しておく必要があります。詳しい説明は、ご使用の Windows の情報を参照してください。

複数ページのイメージの場合、垂直スクロール・バー・タブがドラッグされると、タブの横に小さなウィンドウが表示されます。このウィンドウには、タブの位置に対応するページ番号と、イメージのページ数が表示されます。例えば、5/10 と表示されている場合、イメージは 10 ページあり、タブが保留解除されると、ページ番号 5 が現行ページになることを示しています。

この振る舞いは、以下のキー内のレジストリー設定によって抑止することができます。

HKEY_LOCAL_MACHINE¥Software¥IBM¥OnDemand Image Web
Viewer¥Preferences

ストリング値 PageNumberScroll が 0 (ゼロ) に設定されていると、スクロール・バー・タブがドラッグされても、ページ番号ウィンドウは表示されません。

同じレジストリー・キー内で、ストリング値 PageNumberToolbar が 1 に設定されていると、複数ページのイメージの場合、ページ番号情報がツールバーに表示されます。例えば、3/5 と表示されている場合、イメージは 5 ページあり、ページ番号 3 が現行ページであることを示しています。

Java 行データ・ビューアー

IBM では、拡張 Java 行データ・ビューアーを提供するようになりました。改善された機能としては、ページ幅全体の印刷などの、拡張印刷機能があります。グラフィカル・ユーザー・インターフェースは、Swing ライブラリーに基づいています。

IBM では、現在、アプレット・ディレクトリー内に以下の 2 つのバージョンの Java 行データ・ビューアーを提供しています。

ODLineDataViewer.jar は、バージョン 1.1.8 以降の Java プラグインを必要とする、古い Java 行データ・ビューアーです。

ODLineDataViewer2.jar は、バージョン 1.4.1 以降の Java プラグインを必要とする、新しい Java 行データ・ビューアーです。

お客様は、新しい Java 行データ・ビューアーまたは古い Java 行データ・ビューアーを使用できます。この選択は、ARSWWW.INI ファイルの [DEFAULT BROWSER] セクションでパラメーターを設定することによって指定します。また、新しい Java 行データ・ビューアーでは、ブラウザーにバージョン 1.4.1 以降の Java プラグインが必要です。ARSWWW.INI ファイルの追加パラメーターは、必要なバージョンの Java プラグインをワークステーションにインストールしていないユーザーのために、Java プラグイン・インストール・ファイルのバージョン番号とロケーションを判別します。

61 ページの表 2 に、Java 行データ・ビューアーをサポートする ARSWWW.INI ファイル内の新規パラメーターを示します。

表 2. Java 行データ・ビューアーに関する ARSWWW.INI ファイル内のパラメーター

パラメーター	値	コメント
ODApplet.version	1	古い Java 行データ・ビューアーを呼び出すことを指定します。指定する場合は、他のパラメーターは無視してください。 注: これはデフォルト値です。また、このパラメーターが省略された場合、ODWEK は古い Java 行データ・ビューアーを使用します。
	2	新しい Java 行データ・ビューアー (拡張バージョン) を呼び出すことを指定します。指定する場合は、以下の 3 つのパラメーターを使用してください。
ODApplet.jre.path.IE	http://java.sun.com/getjava/installer.html	Internet Explorer の場合、java.sun.com Web サイトから、最新バージョンの Java プラグインを自動的にダウンロードしてインストールすることを指定します。Java プラグインを自動的にダウンロードおよびインストールするとどのようなかプレビューは、 http://java.sun.com/getjava/install-windows.html を参照してください。 注: プラグインのインストール後に、ブラウザを再始動する必要があります。
	<location>	会社のイントラネット内の Java プラグイン・インストール・ファイルのロケーションを指定します。ロケーションには、http、file、ftp などの有効なブラウザ・プロトコルを指定しなければなりません。次に例を示します。 file://shareName/java/plugins/plugin.exe 注: 管理者は、Java プラグイン・インストール・ファイルをダウンロードして、指定されたロケーションに保管しなければなりません。 インストール・ファイルのロケーションを指定することにより、ブラウザは、Java プラグインをワークステーション上に自動的にインストールします。インストール完了後、場合によってはブラウザを再始動する必要があります。

表2. Java 行データ・ビューアーに関する ARSWWW.INI ファイル内のパラメーター (続き)

パラメーター	値	コメント
ODApplet.jre.path.NN	http://java.sun.com/j2se/1.4.1/download.html	Netscape の場合。インストールする Java プラグインを選択するために、JRE/J2SE のダウンロードのページを開くことを指定します。ユーザーは、リンクに従って、使用しているプラットフォームの Java プラグイン・インストール・ファイルをダウンロードします。Java プラグイン・インストール・ファイルをダウンロードしてから、ユーザーはプラグインをワークステーションにインストールする必要があります。プラグインのインストール後、場合によっては、ユーザーはブラウザを再始動する必要があります。
	<location>	<p>会社のイントラネット内のプラグイン・ファイルのロケーションを指定します。ロケーションには、http、file、ftp などの有効なブラウザ・プロトコルを指定しなければなりません。次に例を示します。</p> <pre>http://webServer/ tmp/ondemand/java/ plugins</pre> <p>注: 管理者は、プラグイン・ファイルをダウンロードして、指定されたロケーションに保管しなければなりません。特定のファイルに対するパスを指定することはできません。Netscape が稼働しているオペレーティング・システムで、そのパスを認識していないからです。また、指定されている形式によって、管理者は、必要に応じて、他のプラットフォームのプラグインをダウンロードすることができます。</p> <p>ユーザーは、ワークステーションに Java プラグインをインストールする必要があります。プラグインのインストール後に、ユーザーはブラウザを再始動する必要があることがあります。</p>

表 2. Java 行データ・ビューアーに関する ARSWWW.INI ファイル内のパラメーター (続き)

パラメーター	値	コメント
ODApplet.jre.version	<version>	<p>使用する Java プラグインのバージョンを指定します。バージョン 1.4 以降を指定してください。そのレベルのリリースのプラグイン (1.4.0、1.4.0_03、1.4.1_01 など) をサポートするようにするには、メジャー・バージョン番号 (1.4 など) を指定してください。特定のバージョンの Java プラグインのみをサポートするには、特定のバージョン番号 (1.4.1_01 など) を指定してください。 java.sun.com Web サイトから有効なバージョン番号を入手してください。次に例を示します。</p> <p>1.4</p> <p>または、</p> <p>1.4.1_01</p>

以下の例では、古い Java 行データ・ビューアーをサポートする場合の ARSWWW.INI ファイルの構成方法を示します。

```
[DEFAULT BROWSER]
ODApplet.version=1
```

注:

1. ARSWWW.INI ファイルで ODApplet.version パラメーターを省略すると、ODWEK は、古い Java 行データ・ビューアーを使用します。
2. ODApplet パラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションでしか指定できません。(このパラメーターが別のブラウザ・セクションで指定されても、無視されます。)

以下の例は、新しい Java 行データ・ビューアー (拡張バージョン) とバージョン 1.4 以降の Java プラグインをサポートする場合の ARSWWW.INI ファイルの構成方法を示しています。Internet Explorer の場合、java.sun.com Web サイトから、最新バージョンの Java プラグインを自動的にダウンロードしてインストールすることができます。Netscape の場合、管理者が、ローカル Web サーバー上の指定されたロケーションに、さまざまなプラットフォーム用の Java プラグイン・インストール・ファイルのコピーを保管しているので、ユーザーは java.sun.com の JRE/J2SE のダウンロードのページにアクセスする必要はありません。**注:** バージョン 1.4 以降の Java プラグインをワークステーションにインストールしていないユーザーに対してのみ、プラグインをダウンロード/インストールするようにプロンプトが出されません。

```
[DEFAULT BROWSER]
ODApplet.version=2
ODApplet.jre.path.IE=http://java.sun.com/getjava/installer.html
ODApplet.jre.path.NN=http://localWebServer/java/plugins
ODApplet.jre.version=1.4
```

次のステップ

ODWEK ソフトウェアをインストールし、ARSWWW.INI ファイルを構成し、サンプル・アプリケーションを構成し、Web ビューアーをインストールしたら、これで ODWEK の使用を開始することができます。

付録 A. CGI API リファレンス

この章では、ODWEK で使用できるプログラミング関数に関する情報を扱います。この章は主に、ODWEK の Web ブラウザーへの統合を担当するプログラマーを対象にしています。

注: パラメーター値は標準テキストになります。このテキストには、ブラウザーで混乱が生じる文字が含まれている場合があります。起こりうるエラーを避けるために、すべての特殊文字を、それらに対応する 16 進数コードにエンコードしなければなりません。これらの特殊文字には、制御文字および特定の英数字記号が含まれます。例えば、以下のストリングであれば、

```
The post date is 12/31/95
```

以下のように変換されます。

```
The%20post%20date%20is%2012%2f31%2f95
```

パラメーター値には、フォルダー名、フォルダー・フィールド名、および検索基準が含まれます。

Add Annotation

指定した文書に注釈を追加します。

目的

Add Annotation 関数では、指定した文書に注釈を追加することができます。注釈を追加するためには、OnDemand アプリケーション・グループでの Add Annotation 許可を持っていないければなりません。(Access 許可を持っていれば、注釈を追加することができます。)

パラメーター

表 3. Add Annotation 関数

名前=値	目的
_function=addnote	注釈を追加します。
_server=value	OnDemand サーバーの名前。
_user=value	OnDemand ユーザー ID。注釈を付ける文書が含まれているアプリケーション・グループごとに、Add Annotation 許可を持っていないければなりません。(Application Group Access 許可を持っていれば、注釈を追加することができます。)
_password=value	ユーザーのパスワード。
_folder=value	フォルダーの名前。
_perm=value	注釈が共通 (0)、専用 (1)、またはグループ専用 (2) のいずれであるかを判別します。共通注釈を表示できるのは、そのアプリケーション・グループの View Annotation 許可を持っているユーザーです。専用注釈を表示できるのは、その注釈を作成したユーザー、アプリケーション・グループ管理者、およびシステム管理者です。グループ専用注釈を表示できるのは、指定したグループ内のユーザー、アプリケーション・グループ管理者、およびシステム管理者です。 _group パラメーターに、グループの名前を入れます。デフォルト値は 0 (共通) です。
_group=groupName	_perm パラメーターを 2 (グループ専用) に設定した場合、グループの名前を付けます。
_copy=value	文書を他のサーバーにエクスポートした場合に、その文書に注釈を添付したままにするかどうかを決定します。デフォルト値は off で、これは注釈が文書に添付されないことを意味します。値 on は、文書を他のサーバーにエクスポートした場合に、その文書に注釈を添付することを意味します。
_text=value	注釈のテキスト

表 3. Add Annotation 関数 (続き)

名前=値	目的
<code>_html=value</code>	<p>ODWEK が出力 Web ページを生成する際に、テンプレートとして使用する HTML ファイルを判別します。この値には、ファイル名か * (アスタリスク) を指定することができます。この値がアスタリスクの場合、ODWEK は ARSWWW.INI ファイル内の <code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーにある、<code>ADDNOTE.HTML</code> ファイルを使用します。パス名を指定せずにファイル名を指定することもできますが、その場合、ファイルは <code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーに置かれていなければなりません。値にパス名を含める場合、指定するパスは、<code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーを起点とするパスでなければなりません。</p> <p>HTML ファイルの内容はすべてお客様が定義します。ただし、このファイルには以下のコメント行を含めなければなりません。</p> <pre><!--AOI# Marker--></pre> <p>コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。</p> <p><code>TEMPLATE.HTM</code> ファイルは、ODWEK で用意されているサンプル・テンプレート・ファイルです。このサンプル・テンプレート・ファイルを使用して、Add Annotation 関数で使用される、独自のテンプレート・ファイルの作成に役立てることができます。</p>
<code>_nohtml=value</code>	<p>ODWEK が生成する出力のタイプを判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が HTML 出力を生成することを意味します。1 を指定した場合、ODWEK は区切り文字付きの ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。</p>
<code>_docid=documentID</code>	<p>注釈を添付する文書の ID。文書 ID は、Document Hit List 関数によって戻されます。</p>
<code>_port=value</code>	<p>OnDemand サーバーのポート番号。デフォルト値である 0 (ゼロ) は、サービス・テーブル (<code>WRKSRVTBLE</code>) で指定されているポート番号をサーバーが使用することを意味します。サービス・テーブルでポート番号が指定されていない場合、OnDemand はポート番号 1445 の使用を試行します。指定した任意の値によって、ARSWWW.INI ファイル内の <code>PORT</code> パラメーターの値が指定変更されます。</p>
<code>_codepage=value</code>	<p>OnDemand データベースのコード・ページ。デフォルトのコード・ページは HTTP サーバーのコード・ページです。サーバーのコード・ページがデータベースのコード・ページと異なる場合、コード・ページを指定しなければなりません。指定した任意の値によって、ARSWWW.INI ファイル内の <code>CODEPAGE</code> パラメーターの値が指定変更されます。</p>
<code>_logoff=1</code>	<p>注釈の追加後、ユーザーを OnDemand サーバーから自動的に切断します。このパラメーターを指定することによって、アプリケーションで Logoff 関数を呼び出して、ユーザーを切断する必要がなくなります。このパラメーターに有効な値は 1 のみです。</p>

使用法

以下のパラメーターは必須です。

```
_function  
_server  
_user  
_password  
_text  
_docid
```

以下のパラメーターはオプションです。

```
_perm  
_group (_perm でグループ専用を指定した場合に必要)  
_html  
_nohtml  
_port  
_codepage  
_logoff
```

関数呼び出しのサンプル

```
http://www.company.com/cgi-bin/arswww.cgi?_function=addnote  
&_server=od400&_user=web&_password=web  
&_folder=credit%20card%20statements  
&_text=Test%20note%20from%20the%20OnDemand%20Internet%20Client  
&_docid=6850-6851-SUA17-1FAAA-225712-1634-132014-132172-89-76-11-25-0  
&_perm=1&_logoff=1
```

Change Password

OnDemand ログオン・パスワードを変更します。

目的

Change Password 関数では、OnDemand パスワードを変更することができます。

パラメーター

表 4. Change Password 関数

名前=値	目的
<code>_function=chgpassword</code>	ユーザー ID の OnDemand パスワードを変更します。
<code>_server=value</code>	OnDemand サーバーの名前。
<code>_user=value</code>	OnDemand ユーザー ID。
<code>_password=value</code>	ユーザー ID のパスワード。
<code>_new_password=value</code>	ユーザー ID の新しいパスワード。
<code>_html=value</code>	<p>ODWEK が出力 Web ページを生成する際に、テンプレートとして使用する HTML ファイルを判別します。この値には、ファイル名か * (アスタリスク) を指定することができます。この値がアスタリスクの場合、ODWEK は ARSWWW.INI ファイル内の TEMPLATEDIR パラメーターで名前が指定されているディレクトリーにある、CHGPASSWORD.HTML ファイルを使用します。パス名を指定せずにファイル名を指定することもできますが、その場合、ファイルは TEMPLATEDIR パラメーターで名前が指定されているディレクトリーに置かれていなければなりません。値にパス名を含める場合、指定するパスは、TEMPLATEDIR パラメーターで名前が指定されているディレクトリーを起点とするパスでなければなりません。</p> <p>HTML ファイルの内容はすべてお客様が定義します。ただし、このファイルには以下のコメント行を含めなければなりません。</p> <pre><!--AOI# Marker--></pre> <p>コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。</p> <p>TEMPLATE.HTM ファイルは、ODWEK で用意されているサンプル・テンプレート・ファイルです。このサンプル・テンプレート・ファイルを使用して、Change Password 関数で使用される、独自のテンプレート・ファイルの作成に役立てることができます。</p>
<code>_nohtml=value</code>	ODWEK が生成する出力のタイプを判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が HTML 出力を生成することを意味します。1 を指定した場合、ODWEK は区切り文字付きの ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。

表 4. Change Password 関数 (続き)

名前=値	目的
<code>_port=value</code>	OnDemand サーバーのポート番号。デフォルト値である 0 (ゼロ) は、サービス・テーブル (WRKSRVTBLE) で指定されているポート番号をサーバーが使用することを意味します。サービス・テーブルでポート番号が指定されていない場合、OnDemand はポート番号 1445 の使用を試行します。指定した任意の値によって、ARSWWW.INI ファイル内の PORT パラメーターの値が指定変更されます。
<code>_codepage=value</code>	OnDemand データベースのコード・ページ。デフォルトのコード・ページは HTTP サーバーのコード・ページです。サーバーのコード・ページがデータベースのコード・ページと異なる場合、コード・ページを指定しなければなりません。指定した任意の値によって、ARSWWW.INI ファイル内の CODEPAGE パラメーターの値が指定変更されます。
<code>_cgibin=program</code>	次の出力ページを生成するときに CGI プログラムが使用します。これを指定すると、デフォルトのプログラム (ARSWWW.CGI) ではなく、指定したプログラムの呼び出しがそのページに入ります。このパラメーターは、IBM が提供する CGI プログラムまたはサーブレットに対してフロントエンド CGI プログラムまたはサーブレットを作成するプログラマーが主として使用するものです。 <i>program</i> には、ServerRoot ディレクトティブに関連したディレクトリーの名前か、HTTP サーバー構成ファイルで定義されている <i>alias</i> の名前を指定することができます。デフォルトでは、ODWEK は CGI-BIN ディレクトリーから CGI プログラムを取り出します。
<code>_logoff=1</code>	パスワードの変更後、ユーザーを OnDemand サーバーから自動的に切断します。このパラメーターを指定することによって、アプリケーションで Logoff 関数を呼び出して、ユーザーを切断する必要がなくなります。このパラメーターに有効な値は 1 のみです。

使用法

以下のパラメーターは必須です。

```

_function
_server
_user
_password
_new_password

```

以下のパラメーターはオプションです。

```

_html
_nohtml
_port
_codepage
_logoff
_cgibin

```


関数呼び出しのサンプル

```
http://www.company.com/cgi-bin/arswww.cgi?_function=chgpassword  
&_server=od400&_user=web&_password=web  
&_newpassword=newpw&_html=template.htm&_logoff=1
```

|

Document Hit List

検索基準に一致する文書のリストを表示します。

目的

Document Hit List 関数は、特定のフォルダーの検索基準と一致する文書のリストを表示します。各文書は、OnDemand サーバー上の文書へのリンクで示されます。文書をクリックすると、ODWEK はサーバーからその文書を取得し、適切なビューアーを使用してブラウザ・ウィンドウに表示します。

パラメーター

表 5. Document Hit List 関数

名前=値	目的
_function=dochitlist	検索基準に一致する文書のリストを表示します。
_server=value	OnDemand サーバーの名前。
_user=value	OnDemand ユーザー ID。
_password=value	ユーザー ID のパスワード。
_folder=value	フォルダーの名前。
<i>folder field name=value</i>	フォルダー検索フィールドの名前および検索値。フィールド名と検索値のセットは、1 つまたは複数 (最大でフォルダーに定義されているフィールドの数まで) 指定できます。
<i>folder field name2=value</i>	フォルダー検索フィールドで BETWEEN または NOT BETWEEN 検索演算子を使用する場合の、検索するフィールドの上限值。
<i>folder field nameOP=value</i>	フォルダー検索フィールドのデフォルト演算子を指定変更するために使用する演算子。value は、以下のいずれかでなければなりません。 1 Equal を表します 2 Not Equal を表します 4 Less Than を表します 8 Less Than または Equal を表します 16 Greater Than を表します 32 Greater Than または Equal を表します 64 In を表します 128 Not In を表します 256 Like を表します 512 Not Like を表します 1024 Between を表します 2048 Not Between を表します
_display_fields=value[,value,...]	フォルダー表示フィールドの名前を含んでいる、コンマで区切ったリスト。1 つまたは複数のフィールド名を指定できます。このパラメーターを指定しなければ、すべてのフォルダー表示フィールドが出力ページに含められます。
_sort_field=value[,value,...]	OnDemand が文書リスト内の項目のソートに使用する、フォルダー検索フィールドを判別します。複数のフィールドを指定する場合は、フィールド名をコンマで区切ります。例えば、_sort_field=Account,Account+Balance,Date のように指定します。デフォルト・ソート・フィールドは、フォルダーの「フィールド情報 (Field Information)」ページで定義されます。

表 5. Document Hit List 関数 (続き)

名前=値	目的
<code>_sort_order=value[,value,...]</code>	<p>sort_field パラメーターで指定したフォルダー検索フォルダーごとに、最初から最後、あるいは最後から最初のいずれの順序で OnDemand が項目をソートするかを判別します。最初から最後へ項目をソートする場合、A (昇順) を指定します。それ以外の文字を指定すると、最後から最初へ項目がソートされます (降順)。例えば、_sort_order=A,D,A のように指定します。デフォルトのソート順は、フォルダーの「フィールド情報 (Field Information)」ページで定義されているソート順序によって決まります。</p>
<code>_max_hits=value</code>	<p>照会に一致する項目の数とは関係なく、ODWEK が文書リストに戻す項目の最大数を判別します。ODWEK は、照会に一致する項目を、一致する項目がデータベースにロードされた順序で文書リストに入れます。</p> <p>ODWEK は以下で最初に指定された値を使用して、文書リストに戻す項目の数を判別します。</p> <ol style="list-style-type: none"> 1. 「最大ヒット」フィールドの値 (「フォルダー許可」ページで指定される)。この値は他のすべての値を指定変更します。 2. <code>_max_hits</code> パラメーターの値 (指定されている場合)。この値は、ARSWWW.INI ファイルからの MAXHITS パラメーターを指定変更します。 3. MAXHITS パラメーターの値 (指定されている場合)。 4. 上記のいずれも指定されていない場合、ODWEK は最大で 200 個の項目を文書リストに戻します。
<code>_html=value</code>	<p>ODWEK が出力 Web ページを生成する際に、テンプレートとして使用する HTML ファイルを判別します。この値には、ファイル名か * (アスタリスク) を指定することができます。この値がアスタリスクの場合、ODWEK は ARSWWW.INI ファイル内の TEMPLATEDIR パラメーターで名前が指定されているディレクトリーにある、DOCHITLIST.HTML ファイルを使用します。パス名を指定せずにファイル名を指定することもできますが、その場合、ファイルは TEMPLATEDIR パラメーターで名前が指定されているディレクトリーに置かれていなければなりません。値にパス名を含める場合、指定するパスは、TEMPLATEDIR パラメーターで名前が指定されているディレクトリーを起点とするパスでなければなりません。</p> <p>HTML ファイルの内容はすべてお客様が定義します。ただし、このファイルには以下のコメント行を含めなければなりません。</p> <pre><!--AOI# Marker--></pre> <p>コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。</p> <p>TEMPLATE.HTM ファイルは、ODWEK で用意されているサンプル・テンプレート・ファイルです。このサンプル・テンプレート・ファイルを使用して、Document Hit List 関数で使用される、独自のテンプレート・ファイルの作成に役立てることができます。</p>
<code>_frame=value</code>	<p>このコマンドの出力には、target=value 属性が含まれます。このパラメーターを使用すれば、HTML フレームをより簡単に作成できます。これはオプション・パラメーターです。</p>

表 5. Document Hit List 関数 (続き)

名前=値	目的
<code>_datefmt=value</code>	ODWEK がデータベースの検索および照会に一致する項目の表示に使用する、日付値の形式を判別します。デフォルトの日付形式は、フォルダーの「Field Information (フィールド情報)」ページで設定されます。OnDemand によりサポートされるデータ・フォーマットの詳細については、「 <i>IBM Content Manager OnDemand for iSeries V5R1 Common Server 管理ガイド</i> 」(SC88-4011) を参照してください。
<code>_nohtml=value</code>	ODWEK が生成する出力のタイプを判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が HTML 出力を生成することを意味します。1 を指定した場合、ODWEK は区切り文字付きの ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。
<code>_port=value</code>	OnDemand サーバーのポート番号。デフォルト値である 0 (ゼロ) は、サービス・テーブル (WRKSRVTBLE) で指定されているポート番号をサーバーが使用することを意味します。サービス・テーブルでポート番号が指定されていない場合、OnDemand はポート番号 1445 の使用を試行します。指定した任意の値によって、ARSWWW.INI ファイル内の PORT パラメーターの値が指定変更されます。
<code>_codepage=value</code>	OnDemand データベースのコード・ページ。デフォルトのコード・ページは HTTP サーバーのコード・ページです。サーバーのコード・ページがデータベースのコード・ページと異なる場合、コード・ページを指定しなければなりません。指定した任意の値によって、ARSWWW.INI ファイル内の CODEPAGE パラメーターの値が指定変更されます。
<code>_sql=string</code>	<p>OnDemand がフォルダーの検索に使用する、SQL 照会を指定します。このパラメーターを指定すると、指定できるフォルダーの名前/値のペアではなく、SQL 照会が使用されて、フォルダーが検索されます。OnDemand は照会ストリングの妥当性検査は行いません。</p> <p>SQL ストリングを使用する場合、アプリケーション・グループ・データベース・フィールドの名前と値を指定しなければなりません。日付フィールドの照会を計画する場合、OnDemand 内部日付値を指定しなければなりません。例えば、日付が 1999 年 1 月 1 日の場合、10593 と指定します。ARSDATE コマンドを使用して、指定した日付の内部日付値をリストできます。</p> <p>この SQL ストリングを使用して、フォルダー内にあるすべてのアプリケーション・グループを検索します。SQL ストリングに指定したデータベース・フィールド名が、一部のアプリケーション・グループにしか含まれていない場合、照会は失敗します。</p>
<code>_date1=value</code>	検索する日付の範囲の開始日付を指定する場合に使用します。_date1 および _date2 パラメーターを指定した場合、OnDemand は照会を、指定した日付のいずれかまたは両方を含む表 (複数の場合もある) に限定します。指定する日付ストリングの形式は、フォルダー・フィールドの表示形式と一致していなければなりません。(管理クライアントを使用して、フォルダー・フィールドの表示形式をリストすることができます。)

表 5. Document Hit List 関数 (続き)

名前=値	目的
<code>_date2=value</code>	検索する日付の範囲の終了日付を指定する場合に使用します。 <code>_date1</code> および <code>_date2</code> パラメーターを指定した場合、OnDemand は照会を、指定した日付のいずれかまたは両方を含む表 (複数の場合もある) に限定します。指定する日付ストリングの形式は、フォルダー・フィールドの表示形式と一致していなければなりません。(管理クライアントを使用して、フォルダー・フィールドの表示形式をリストすることができます。)
<code>_cgibin=program</code>	次の出力ページを生成するとき CGI プログラムが使用します。これを指定すると、デフォルトのプログラム (ARSWWW.CGI) ではなく、指定したプログラムの呼び出しがそのページに入ります。このパラメーターは、IBM が提供する CGI プログラムまたはサーブレットに対してフロントエンド CGI プログラムまたはサーブレットを作成するプログラマーが主として使用するものです。 <code>program</code> には、ServerRoot ディレクティブに関連したディレクトリーの名前か、HTTP サーバー構成ファイルで定義されている <code>alias</code> の名前を指定することができます。デフォルトでは、ODWEK は <code>/QIBM/Proddata/OnDemand/www/bin</code> ディレクトリーから CGI プログラムを取り出します。
<code>_or=value</code>	OR 論理演算子を使用して検索フィールドに接続する場合、1 を指定します。この場合、項目は指定した検索値の 1 つ以上に一致しなければなりません。デフォルト値は 0 (ゼロ) で、これは AND 論理演算子 (項目は指定されている検索値すべてに一致しなければなりません) を使用して、OnDemand が検索フィールドを接続することを意味します。
<code>_logoff=1</code>	文書リストの作成後、ユーザーを OnDemand サーバーから自動的に切断します。このパラメーターを指定することによって、アプリケーションで Logoff 関数を呼び出して、ユーザーを切断する必要がなくなります。このパラメーターに有効な値は 1 のみです。

使用法

以下のパラメーターは必須です。

`_function`
`_server`
`_user`
`_password`
`_folder`

以下のパラメーターはオプションです。

`folder field name`
`folder field name2`
`folder field nameOP`
`_display_fields`
`_sort_field`
`_sort_order`
`_max_hits`
`_frame`
`_datefmt`

```
|          _sql
|          _date1
|          _date2
|          _or
|          _html
|          _nohtml
|          _port
|          _codepage
|          _logoff
|          _cgibin
```

関数呼び出しのサンプル

```
|          http://www.company.com/cgi-bin/arswww.cgi?_function=dochitlist
|          &_server=od400&_user=web&_password=web
|          &_folder=credit%20card%20statements
|          &_account%20number=1000100010009999&date=1%2f1%2f96&date2=12%2f31%2f96
|          &_nameOP=256&name=%AA
|          &_sort_field=Account,Account%20Balance,Date&_sort_order=A,D,A
|          &_logoff=1
|          &_html=template.htm
```

Logoff

OnDemand サーバーからログオフします。

目的

Logoff 関数は、OnDemand サーバーからのユーザーのログオフを試行します。ログオフするサーバーの名前およびユーザー ID は、Logon 関数によって、クライアント上のブラウザの Cookie に保管されています。サーバーが有効な OnDemand サーバーでない場合、エラー・メッセージが戻されます。指定したサーバーにそのユーザー ID でログオンされていない場合、エラー・メッセージが表示されます。

パラメーター

表 6. Logoff 関数

名前=値	目的
<code>_function=logoff</code>	OnDemand サーバーからログオフします。
<code>_html=value</code>	<p>ODWEK が出力 Web ページを生成する際に、テンプレートとして使用する HTML ファイルを判別します。この値には、ファイル名か * (アスタリスク) を指定することができます。この値がアスタリスクの場合、ODWEK は ARSWWW.INI ファイル内の <code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーにある、<code>LOGOFF.HTML</code> ファイルを使用します。パス名を指定せずにファイル名を指定することもできますが、その場合、ファイルは <code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーに置かれていなければなりません。値にパス名を含める場合、指定するパスは、<code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーを起点とするパスでなければなりません。</p> <p>HTML ファイルの内容はすべてお客様が定義します。ただし、このファイルには以下のコメント行を含めなければなりません。</p> <pre><!-- -AOI# Marker- -></pre> <p>コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。</p> <p><code>TEMPLATE.HTM</code> ファイルは、ODWEK で用意されているサンプル・テンプレート・ファイルです。このサンプル・テンプレート・ファイルを使用して、<code>logoff</code> 関数で使用される、独自のテンプレート・ファイルの作成に役立てることができます。</p>
<code>_nohtml=value</code>	ODWEK が生成する出力のタイプを判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が HTML 出力を生成することを意味します。1 を指定した場合、ODWEK は区切り文字付きの ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。

表 6. Logoff 関数 (続き)

名前=値	目的
<code>_port=value</code>	OnDemand サーバーのポート番号。デフォルト値である 0 (ゼロ) は、サービス・テーブル (WRKSRVTBLE) で指定されているポート番号をサーバーが使用することを意味します。サービス・テーブルでポート番号が指定されていない場合、OnDemand はポート番号 1445 の使用を試行します。指定した任意の値によって、ARSWWW.INI ファイル内の PORT パラメーターの値が指定変更されます。

使用法

以下のパラメーターは必須です。

`_function`

以下のパラメーターはオプションです。

`_html`

`_nohtml`

`_port`

関数呼び出しのサンプル

```
http://www.company.com/cgi-bin/arswww.cgi?_function=logoff
&_html=template.htm
```


Logon

OnDemand サーバーにログオンします。

目的

Logon 関数は、サーバー、ユーザー、およびパスワードのパラメーターの値を使用して、OnDemand サーバーへのアクセスを試行します。Logon 関数は、指定されているユーザーが、指定されているサーバーへのログオンを許可されているか検査し、さらにパスワードを検査します。ユーザーがそのサーバーへのログオンを許可されていない場合、エラー・メッセージが戻されます。サーバーが有効な OnDemand サーバーでない場合、エラー・メッセージが戻されます。ユーザーのパスワードが無効な場合、エラー・メッセージが戻されます。正常にログオンすると、Logon 関数は、ユーザーがアクセスを許可されているフォルダーのリストが含まれている Web ページを表示します。

パラメーター

表 7. Logon 関数

名前=値	目的
<code>_function=logon</code>	OnDemand サーバーにログオンします。
<code>_server=value</code>	OnDemand サーバーの名前。
<code>_user=value</code>	OnDemand ユーザー ID。
<code>_password=value</code>	ユーザー ID のパスワード。
<code>_new_password=value</code>	ユーザー ID の新しいパスワード。パスワードを変更できるのは、OnDemand へ正常にログオンしてからです。これはオプション・パラメーターです。
<code>_html=value</code>	<p>ODWEK が出力 Web ページを生成する際に、テンプレートとして使用する HTML ファイルを判別します。この値には、ファイル名か * (アスタリスク) を指定することができます。この値がアスタリスクの場合、ODWEK は ARSWWW.INI ファイル内の TEMPLATEDIR パラメーターで名前が指定されているディレクトリーにある、LOGON.HTML ファイルを使用します。パス名を指定せずにファイル名を指定することもできますが、その場合、ファイルは TEMPLATEDIR パラメーターで名前が指定されているディレクトリーに置かれていなければなりません。値にパス名を含める場合、指定するパスは、TEMPLATEDIR パラメーターで名前が指定されているディレクトリーを起点とするパスでなければなりません。</p> <p>HTML ファイルの内容はすべてお客様が定義します。ただし、このファイルには以下のコメント行を含めなければなりません。</p> <pre><!-- -AOI# Marker- -></pre> <p>コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。</p> <p>TEMPLATE.HTM ファイルは、ODWEK で用意されているサンプル・テンプレート・ファイルです。このサンプル・テンプレート・ファイルを使用して、Logon 関数で使用される、独自のテンプレート・ファイルの作成に役立てることができます。</p>

表 7. Logon 関数 (続き)

名前=値	目的
<code>_frame=value</code>	このコマンドの出力には、 <code>target=value</code> 属性が含まれます。このパラメーターを使用すれば、HTML フレームをより簡単に作成できます。これはオプション・パラメーターです。
<code>_datefmt=value</code>	ODWEK がデータベースの検索および照会に一致する項目の表示に使用する、日付値の形式を判別します。デフォルトの日付形式は、フォルダーの「Field Information (フィールド情報)」ページで設定されます。OnDemand によりサポートされるデータ・フォーマットの詳細については、「IBM Content Manager OnDemand for iSeries V5R1 Common Server 管理ガイド」(SC88-4011) を参照してください。
<code>_nohtml=value</code>	ODWEK が生成する出力のタイプを判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が HTML 出力を生成することを意味します。1 を指定した場合、ODWEK は区切り文字付きの ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。
<code>_port=value</code>	OnDemand サーバーのポート番号。デフォルト値である 0 (ゼロ) は、サービス・テーブル (WRKSRVTBLE) で指定されているポート番号をサーバーが使用することを意味します。サービス・テーブルでポート番号が指定されていない場合、OnDemand はポート番号 1445 の使用を試行します。指定した任意の値によって、ARSWWW.INI ファイル内の PORT パラメーターの値が指定変更されます。
<code>_codepage=value</code>	OnDemand データベースのコード・ページ。デフォルトのコード・ページは HTTP サーバーのコード・ページです。サーバーのコード・ページがデータベースのコード・ページと異なる場合、コード・ページを指定しなければなりません。指定した任意の値によって、ARSWWW.INI ファイル内の CODEPAGE パラメーターの値が指定変更されます。
<code>_cgibin=program</code>	次の出力ページを生成するときに CGI プログラムが使用します。これを指定すると、デフォルトのプログラム (ARSWWW.CGI) ではなく、指定したプログラムの呼び出しがそのページに入ります。このパラメーターは、IBM が提供する CGI プログラムまたはサーブレットに対してフロントエンド CGI プログラムまたはサーブレットを作成するプログラマーが主として使用するものです。 <i>program</i> には、ServerRoot ディレクティブに関連したディレクトリーの名前か、HTTP サーバー構成ファイルで定義されている <i>alias</i> の名前を指定することができます。デフォルトでは、ODWEK は CGI-BIN ディレクトリーから CGI プログラムを取り出します。

使用法

以下のパラメーターは必須です。

```

_function
_server
_user
_password

```

以下のパラメーターはオプションです。

```

_new_password
_frame

```

| _datefmt
| _html
| _nohtml
| _port
| _codepage
| _logoff
| _cgibin

関数呼び出しのサンプル

```
http://www.company.com/cgi-bin/arswww.cgi?_function=logon  
&_server=od400&_user=web&_password=web  
&_html=template.htm
```

Print Document (サーバー)

指定したサーバー・プリンターに 1 つまたは複数の文書を送信します。

目的

Print Document 関数は、文書のコピーを OnDemand サーバー・プリンターに送信します。サーバー印刷機能を使用するためには、OnDemand アプリケーション・グループでの Print Document 許可を持っていないければなりません。(Access 許可を持っているれば、文書を印刷することができます。) OnDemand サーバーには、少なくとも 1 つ以上のサーバー・プリンターが定義されていなければなりません。

パラメーター

表 8. Print Document 関数

名前=値	目的
<code>_function=printdocs</code>	文書を印刷します。
<code>_server=value</code>	OnDemand サーバーの名前。
<code>_user=value</code>	OnDemand ユーザー ID。印刷する文書が含まれているアプリケーション・グループごとに、Document Print 許可を持っていないければなりません。(Application Group Access 許可を持っているれば、文書を印刷することができます。)
<code>_password=value</code>	ユーザーのパスワード。
<code>_folder=value</code>	フォルダーの名前。

表 8. *Print Document* 関数 (続き)

名前=値	目的
<p>_printer=value</p>	<p>OnDemand サーバー・プリンターの名前。</p> <p>指定されたプリンターが通知付きの FAX またはプリンターの場合、以下の追加パラメーターを指定することができます。</p> <p>_recv_name=value レシーバーの名前。</p> <p>_recv_comp=value レシーバーの会社の名前。</p> <p>_recv_fax=value レシーバーの FAX 番号。</p> <p>_send_name=value 送信側の名前。</p> <p>_send_comp=value 送信側の会社の名前。</p> <p>_send_tel=value 送信側の電話番号。</p> <p>_send_fax=value 送信側の FAX 番号。</p> <p>_send_cover=value 文書のカバー・ページを生成するために、ヘッダー・ページ出口プログラムが他のパラメーターと値をマージするユーザー定義オーバーレイ。</p> <p>_subject=value 文書の件名を示すストリング。</p> <p>_notes=value 文書に関する注釈を示すストリング。</p>

表 8. Print Document 関数 (続き)

名前=値	目的
_html=value	<p>ODWEK が出力 Web ページを生成する際に、テンプレートとして使用する HTML ファイルを判別します。この値には、ファイル名か * (アスタリスク) を指定することができます。この値がアスタリスクの場合、ODWEK は ARSWWW.INI ファイル内の TEMPLATEDIR パラメーターで名前が指定されているディレクトリーにある、PRINTDOCS.HTML ファイルを使用します。パス名を指定せずにファイル名を指定することもできますが、その場合、ファイルは TEMPLATEDIR パラメーターで名前が指定されているディレクトリーに置かれていなければなりません。値にパス名を含める場合、指定するパスは、TEMPLATEDIR パラメーターで名前が指定されているディレクトリーを起点とするパスでなければなりません。</p> <p>HTML ファイルの内容はすべてお客様が定義します。ただし、このファイルには以下のコメント行を含めなければなりません。</p> <pre><!-- -AOI# Marker- --></pre> <p>コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。</p> <p>TEMPLATE.HTM ファイルは、ODWEK で用意されているサンプル・テンプレート・ファイルです。このサンプル・テンプレート・ファイルを使用して、Print Document 関数で使用される、独自のテンプレート・ファイルの作成に役立てることができます。</p>
_nohtml=value	<p>ODWEK が生成する出力のタイプを判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が HTML 出力を生成することを意味します。1 を指定した場合、ODWEK は区切り文字付きの ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。</p>
_docids=documentIDList	<p>印刷する文書の文書 ID のリスト。文書 ID は、Document Hit List 関数によって戻されます。複数の文書 ID を指定する場合、文書 ID の間に ¥003 文字を入れて区切る必要があります。</p> <p>注: 文書 ID の数が 200 を超える場合、_max_hits パラメーターを指定しなければなりません。</p>
_port=value	<p>OnDemand サーバーのポート番号。デフォルト値である 0 (ゼロ) は、サービス・テーブル (WRKSRVTBLE) で指定されているポート番号をサーバーが使用することを意味します。サービス・テーブルでポート番号が指定されていない場合、OnDemand はポート番号 1445 の使用を試行します。指定した任意の値によって、ARSWWW.INI ファイル内の PORT パラメーターの値が指定変更されます。</p>
_codepage=value	<p>OnDemand データベースのコード・ページ。デフォルトのコード・ページは HTTP サーバーのコード・ページです。サーバーのコード・ページがデータベースのコード・ページと異なる場合、コード・ページを指定しなければなりません。指定した任意の値によって、ARSWWW.INI ファイル内の CODEPAGE パラメーターの値が指定変更されます。</p>

表 8. *Print Document* 関数 (続き)

名前=値	目的
<code>_max_hits=value</code>	<p>処理する文書 ID の数を指定する場合、このパラメーターを使用します。 <code>_docids</code> パラメーターで指定されている文書 ID の数と同じか、それより大きい値を指定します。</p> <p>注: 文書 ID の数が、<code>ARSWWW.CGI</code> ファイルの <code>MAXHITS</code> パラメーターで指定されている値 (指定されていない場合は 200) を超える場合、<code>_max_hits</code> パラメーターを指定しなければなりません。<code>_max_hits</code> パラメーターを指定しないと (あるいは <code>MAXHITS</code> パラメーターの値を指定しないと)、<code>_docids</code> パラメーターで指定した文書 ID の数に関係なく、最大で 200 個の文書 ID が処理されます。</p> <p><code>ODWEK</code> は以下の値のいずれかを使用して、処理する文書 ID の数を判別します。</p> <ul style="list-style-type: none"> • <code>_max_hits</code> パラメーターの値 (指定されている場合)。この値は、<code>MAXHITS</code> パラメーターの値を指定変更します。 • <code>MAXHITS</code> パラメーターの値 (指定されている場合)。 • 上記のいずれも指定されていない場合、<code>ODWEK</code> は最大で 200 個の文書 ID を処理します。
<code>_logoff=1</code>	<p>文書の印刷後、ユーザーを <code>OnDemand</code> サーバーから自動的に切断します。このパラメーターを指定することによって、アプリケーションで <code>Logoff</code> 関数を呼び出して、ユーザーを切断する必要がなくなります。このパラメーターに有効な値は 1 のみです。</p>

使用法

以下のパラメーターは必須です。

```

_function
_server
_user
_password
_folder
_printer
_docids

```

以下のパラメーターはオプションです。

```

_recv_name
_recv_comp
_recv_fax
_send_name
_send_comp
_send_tel
_send_fax
_send_cover
_subject
_notes
_max_hits
_html
_nohtml

```

| _port
| _codepage
| _logoff

関数呼び出しのサンプル

| http://www.company.com/cgi-bin/arswww.cgi?_function=printdocs
| &_server=od400&_user=web&_password=web
| &_folder=credit%20card%20statements
| &_printer=infoprint60
| &_docids=6850-6851-SUA17-1FAAA-225712-1634-132014-132172-89-76-11-25-0
| &_logoff=1

Retrieve Document

選択した文書を OnDemand から取り出します。

目的

Retrieve Document 関数は、選択した文書を OnDemand サーバーから取得します。ODWEK は文書タイプに関連付けられているアプレット、ビューアー、または他のプログラムを使用して、ブラウザー・ウィンドウに文書を表示します。

パラメーター

表 9. Retrieve Document 関数

名前=値	目的
<code>_function=retrieve</code>	選択した文書を取り出します。
<code>_server=value</code>	OnDemand サーバーの名前。
<code>_user=value</code>	OnDemand ユーザー ID。
<code>_password=value</code>	ユーザー ID のパスワード。
<code>_folder=value</code>	フォルダーの名前。
<code>folder field name=value</code>	フォルダー検索フィールドの名前および検索値。フィールド名と検索値のセットは、1 つまたは複数 (最大でフォルダーに定義されているフィールドの数まで) 指定できます。
<code>_html=value</code>	<p>文書の取得でエラーが発生した場合に、ODWEK が (エラーの) 出力 Web ページを生成する際にテンプレートとして使用する、HTML ファイルを判別します。この値には、ファイル名か * (アスタリスク) を指定することができます。この値がアスタリスクの場合、ODWEK は ARSWWW.INI ファイル内の <code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーにある、<code>RETRIEVE.HTML</code> ファイルを使用します。パス名を指定せずにファイル名を指定することもできますが、その場合、ファイルは <code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーに置かれていなければなりません。値にパス名を含める場合、指定するパスは、<code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーを起点とするパスでなければなりません。</p> <p>HTML ファイルの内容はすべてお客様が定義します。ただし、このファイルには以下のコメント行を含めなければなりません。</p> <pre><!-- -AOI# Marker- -></pre> <p>コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。</p> <p><code>TEMPLATE.HTM</code> ファイルは、ODWEK で用意されているサンプル・テンプレート・ファイルです。このサンプル・テンプレート・ファイルを使用して、Retrieve Document 関数で使われる、独自のテンプレート・ファイルの作成に役立てることができます。</p>

表 9. Retrieve Document 関数 (続き)

名前=値	目的
<code>_nohtml=value</code>	ODWEK が生成する出力のタイプを判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が HTML 出力を生成することを意味します。1 を指定した場合、ODWEK は区切り文字付きの ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。
<code>_port=value</code>	OnDemand サーバーのポート番号。デフォルト値である 0 (ゼロ) は、サービス・テーブル (WRKSRVTBLE) で指定されているポート番号をサーバーが使用することを意味します。サービス・テーブルでポート番号が指定されていない場合、OnDemand はポート番号 1445 の使用を試行します。指定した任意の値によって、ARSWWW.INI ファイル内の PORT パラメーターの値が指定変更されます。
<code>_codepage=value</code>	OnDemand データベースのコード・ページ。デフォルトのコード・ページは HTTP サーバーのコード・ページです。サーバーのコード・ページがデータベースのコード・ページと異なる場合、コード・ページを指定しなければなりません。指定した任意の値によって、ARSWWW.INI ファイル内の CODEPAGE パラメーターの値が指定変更されます。
<code>_cgibin=program</code>	次の出力ページを生成するとき CGI プログラムが使用します。これを指定すると、デフォルトのプログラム (ARSWWW.CGI) ではなく、指定したプログラムの呼び出しがそのページに入ります。このパラメーターは、IBM が提供する CGI プログラムまたはサープレットに対してフロントエンド CGI プログラムまたはサープレットを作成するプログラマーが主として使用するものです。 <i>program</i> には、ServerRoot ディレクティブに関連したディレクトリーの名前か、HTTP サーバー構成ファイルで定義されている <i>alias</i> の名前を指定することができます。デフォルトでは、ODWEK は CGI-BIN ディレクトリーから CGI プログラムを取り出します。
<code>_or=value</code>	OR 論理演算子を使用して検索フィールドに接続する場合、1 を指定します。この場合、項目は指定した検索値の 1 つ以上に一致しなければなりません。デフォルト値は 0 (ゼロ) で、これは AND 論理演算子 (項目は指定されている検索値すべてに一致しなければなりません) を使用して、OnDemand が検索フィールドを接続することを意味します。

表 9. Retrieve Document 関数 (続き)

名前=値	目的
_afp=value	<p>OnDemand サーバーから AFP 文書を取り出す場合、クライアントへ文書を送る前に ODWEK が行うアクションがあれば、このパラメーターの値によってそのアクションを判別します。例えば、AFP2WEB Transform を使用して AFP 文書を HTML に変換し、AFP2HTML アプレットを使用して HTML 出力を表示する場合があります。そのような場合は、ODWEK が AFP 文書を変換してからそれをクライアントに送るように、_afp=HTML を指定する必要があります。</p> <p>value には、以下を指定することができます。</p> <p>ASCII ODWEK は AFP 文書を ASCII テキストに変換します。</p> <p>HTML ODWEK は AFP2WEB Transform を使用して、AFP 文書を HTML に変換します。</p> <p>NATIVE ODWEK は AFP 文書とその資源を OnDemand から抽出し、圧縮解除します。</p> <p>注: _afp=NATIVE を指定する場合、MIME コンテンツ・タイプで、使用するビューアーが識別されているかどうか検査してください (詳細については、27 ページの『[MIMETYPES]』を参照)。</p> <p>PDF ODWEK は AFP2WEB Transform を使用して、AFP 文書を PDF に変換します。</p> <p>PLUGIN ODWEK は AFP 文書の変換を行いません (デフォルト)。</p>
_email=value	<p>OnDemand サーバーから EMAIL 文書を取り出す場合、クライアントへ文書を送る前に ODWEK が行うアクションがあれば、このパラメーターの値によってそのアクションを判別します。value には、以下を指定することができます。</p> <p>NATIVE ODWEK は OnDemand から EMAIL 文書を抽出し、圧縮解除します。</p> <p>注: _email=NATIVE を指定する場合、MIME コンテンツ・タイプで、使用するビューアーが識別されているかどうか検査してください (詳細については、27 ページの『[MIMETYPES]』を参照)。</p> <p>HTML ODWEK は EMAIL 文書を HTML に変換します。</p>

表 9. Retrieve Document 関数 (続き)

名前=値	目的
<code>_line=value</code>	<p>OnDemand サーバーから行データ文書を取り出す場合、クライアントへ文書を送る前に ODWEK が行うアクションがあれば、このパラメーターの値によってそのアクションを判別します。value には、以下を指定することができます。</p> <p>APPLET Line Data アプレットで表示するために、ODWEK は行データ文書を変換します (デフォルト)。</p> <p>ASCII ODWEK は行データ文書を ASCII テキストに変換します。</p> <p>NATIVE ODWEK は OnDemand から行データ文書を抽出し、圧縮解除します。 注: <code>_line=NATIVE</code> を指定する場合、MIME コンテンツ・タイプで、使用するビューアーが識別されているかどうか検査してください (詳細については、27 ページの『[MIMETYPES]』を参照)。</p>
<code>_docid=documentID</code>	<p>取り出す文書の ID。文書 ID は、Document Hit List 関数によって戻されます。</p>
<code>_logoff=1</code>	<p>文書を取り出した後、ユーザーを OnDemand サーバーから自動的に切断します。このパラメーターを指定することによって、アプリケーションで Logoff 関数を呼び出して、ユーザーを切断する必要がなくなります。このパラメーターに有効な値は 1 のみです。</p>

使用法

以下のパラメーターは必須です。

```

_function
_server
_user
_password
_folder

```

以下のパラメーターはオプションです。

```

folder field name
_docid
_or
_afp
_email
_line
_html
_nohtml
_port
_codepage
_logoff
_cgibin

```

関数呼び出しのサンプル

```
http://www.company.com/cgi-bin/arswww.cgi?_function=retrieve  
&_server=od400&_user=web&_password=web  
&_folder=credit%20card%20statements  
&account%20number=1000100010009999&date=1%2f1%2f96  
&_html=template.htm&_logoff=1
```

|

Search Criteria

特定のフォルダーの検索基準を表示します。

目的

Search Criteria 関数は、フォームを使用して特定のフォルダーの検索基準を表示します。ユーザーは、デフォルトの検索基準をそのまま使用するか、あるいは検索基準を入力して、特定の文書を検索することができます。「Submit」ボタンをクリックすると、ODWEK は検索基準と一致する文書がリストされた Web ページを表示します。

パラメーター

表 10. Search Criteria 関数

名前=値	目的
<code>_function=searchcrit</code>	特定のフォルダーの検索基準を表示します。
<code>_server=value</code>	OnDemand サーバーの名前。
<code>_user=value</code>	OnDemand ユーザー ID。
<code>_password=value</code>	ユーザー ID のパスワード。
<code>_folder=value</code>	検索するフォルダーの名前。
<code>_html=value</code>	<p>ODWEK が出力 Web ページを生成する際に、テンプレートとして使用する HTML ファイルを判別します。この値には、ファイル名か * (アスタリスク) を指定することができます。この値がアスタリスクの場合、ODWEK は ARSWWW.INI ファイル内の TEMPLATEDIR パラメーターで名前が指定されているディレクトリーにある、SEARCHCRIT.HTML ファイルを使用します。パス名を指定せずにファイル名を指定することもできますが、その場合、ファイルは TEMPLATEDIR パラメーターで名前が指定されているディレクトリーに置かれていなければなりません。値にパス名を含める場合、指定するパスは、TEMPLATEDIR 変数で名前が指定されているディレクトリーを起点とするパスでなければなりません。</p> <p>HTML ファイルの内容はすべてお客様が定義します。ただし、このファイルには以下のコメント行を含めなければなりません。</p> <pre><!-- -AOI# Marker- -></pre> <p>コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。</p> <p>TEMPLATE.HTM ファイルは、ODWEK で用意されているサンプル・テンプレート・ファイルです。このサンプル・テンプレート・ファイルを使用して、Search Criteria 関数で使用される、独自のテンプレート・ファイルの作成に役立てることができます。</p>
<code>_frame=value</code>	このコマンドの出力には、 <code>target=value</code> 属性が含まれます。このパラメーターを使用すれば、HTML フレームをより簡単に作成できます。これはオプション・パラメーターです。

表 10. Search Criteria 関数 (続き)

名前=値	目的
<code>_datefmt=value</code>	ODWEK がデータベースの検索および照会に一致する項目の表示に使用する、日付値の形式を判別します。デフォルトの日付形式は、フォルダーの「Field Information (フィールド情報)」ページで設定されます。OnDemand によりサポートされるデータ・フォーマットの詳細については、「 <i>IBM Content Manager OnDemand for iSeries V5R1 Common Server 管理ガイド</i> 」(SC88-4011) を参照してください。
<code>_nohtml=value</code>	ODWEK が生成する出力のタイプを判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が HTML 出力を生成することを意味します。1 を指定した場合、ODWEK は区切り文字付きの ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。
<code>_port=value</code>	OnDemand サーバーのポート番号。デフォルト値である 0 (ゼロ) は、サービス・テーブル (WRKSRVTBLE) で指定されているポート番号をサーバーが使用することを意味します。サービス・テーブルでポート番号が指定されていない場合、OnDemand はポート番号 1445 の使用を試行します。指定した任意の値によって、ARSWWW.INI ファイル内の PORT パラメーターの値が指定変更されます。
<code>_codepage=value</code>	OnDemand データベースのコード・ページ。デフォルトのコード・ページは HTTP サーバーのコード・ページです。サーバーのコード・ページがデータベースのコード・ページと異なる場合、コード・ページを指定しなければなりません。指定した任意の値によって、ARSWWW.INI ファイル内の CODEPAGE パラメーターの値が指定変更されます。
<code>_cgibin=program</code>	次の出力ページを生成するときに CGI プログラムが使用します。これを指定すると、デフォルトのプログラム (ARSWWW.CGI) ではなく、指定したプログラムの呼び出しがそのページに入ります。このパラメーターは、IBM が提供する CGI プログラムまたはサーブレットに対してフロントエンド CGI プログラムまたはサーブレットを作成するプログラマーが主として使用するものです。 <i>program</i> には、ServerRoot ディレクティブに関連したディレクトリーの名前か、HTTP サーバー構成ファイルで定義されている <i>alias</i> の名前を指定することができます。デフォルトでは、ODWEK は CGI-BIN ディレクトリーから CGI プログラムを取り出します。
<code>_logoff=1</code>	検索条件の表示後、ユーザーを OnDemand サーバーから自動的に切断します。このパラメーターを指定することによって、アプリケーションで Logoff 関数を呼び出して、ユーザーを切断する必要がなくなります。このパラメーターに有効な値は 1 のみです。

使用法

以下のパラメーターは必須です。

```

_function
_server
_user
_password
_folder

```

以下のパラメーターはオプションです。

| _frame
| _datefmt
| _html
| _nohtml
| _port
| _codepage
| _logoff
| _cgibin

関数呼び出しのサンプル

| http://www.company.com/cgi-bin/arswww.cgi?_function=searchcrit
| &_server=od400&_user=web&_password=web
| &_folder=credit%20card%20statements&_html=template.htm&_logoff=1

Update Document

指定した文書に関する 1 つまたは複数のデータベース値を更新します。

目的

Update Document 関数では、許可ユーザーがデータベースを更新することができます。Update Document 関数は、特定の文書に関する 1 つまたは複数のデータベース値を更新します。

パラメーター

表 11. Update Document 関数

名前=値	目的
<code>_function=updatedoc</code>	データベースを更新します。
<code>_server=value</code>	OnDemand サーバーの名前。
<code>_user=value</code>	OnDemand ユーザー ID。ユーザーは、そのアプリケーション・グループの Update Document 許可を持っていないければなりません。
<code>_password=value</code>	ユーザーのパスワード。
<code>_folder=value</code>	フォルダーの名前。
<code>folder field name=value</code>	更新したいフィールドの名前と、フィールドに入れたい値。フィールド名と値のセットは、1 つまたは複数 (最大でフォルダーに定義されているフィールドの数まで) 指定できます。
<code>_html=value</code>	<p>ODWEK が出力 Web ページを生成する際に、テンプレートとして使用する HTML ファイルを判別します。この値には、ファイル名か * (アスタリスク) を指定することができます。この値がアスタリスクの場合、ODWEK は ARSWWW.INI ファイル内の <code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーにある、<code>UPDATE.HTML</code> ファイルを使用します。パス名を指定せずにファイル名を指定することもできますが、その場合、ファイルは <code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーに置かれていなければなりません。値にパス名を含める場合、指定するパスは、<code>TEMPLATEDIR</code> パラメーターで名前が指定されているディレクトリーを起点とするパスでなければなりません。</p> <p>HTML ファイルの内容はすべてお客様が定義します。ただし、このファイルには以下のコメント行を含めなければなりません。</p> <pre><!-- -AOI# Marker- -></pre> <p>コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。</p> <p><code>TEMPLATE.HTM</code> ファイルは、ODWEK で用意されているサンプル・テンプレート・ファイルです。このサンプル・テンプレート・ファイルを使用して、Update Document 関数で使用される、独自のテンプレート・ファイルの作成に役立てることができます。</p>

表 11. Update Document 関数 (続き)

名前=値	目的
<code>_nohtml=value</code>	ODWEK が生成する出力のタイプを判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が HTML 出力を生成することを意味します。1 を指定した場合、ODWEK は区切り文字付きの ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。
<code>_docid=documentID</code>	更新する文書の ID。文書 ID は、Document Hit List 関数によって戻されます。
<code>_port=value</code>	OnDemand サーバーのポート番号。デフォルト値である 0 (ゼロ) は、サービス・テーブル (WRKSRVTBLE) で指定されているポート番号をサーバーが使用することを意味します。サービス・テーブルでポート番号が指定されていない場合、OnDemand はポート番号 1445 の使用を試行します。指定した任意の値によって、ARSWWW.INI ファイル内の PORT パラメーターの値が指定変更されます。
<code>_codepage=value</code>	OnDemand データベースのコード・ページ。デフォルトのコード・ページは HTTP サーバーのコード・ページです。サーバーのコード・ページがデータベースのコード・ページと異なる場合、コード・ページを指定しなければなりません。指定した任意の値によって、ARSWWW.INI ファイル内の CODEPAGE パラメーターの値が指定変更されます。
<code>_logoff=1</code>	文書の更新後、ユーザーを OnDemand サーバーから自動的に切断します。このパラメーターを指定することによって、アプリケーションで Logoff 関数を呼び出して、ユーザーを切断する必要がなくなります。このパラメーターに有効な値は 1 のみです。

使用法

以下のパラメーターは必須です。

```

_function
_server
_user
_password
_folder

```

以下のパラメーターはオプションです。

```

folder field name
_docid
_html
_nohtml
_port
_codepage
_logoff

```

関数呼び出しのサンプル

```

http://www.company.com/cgi-bin/arswww.cgi?_function=updatedoc
&_server=od400&_user=web&_password=web
&_folder=credit%20card%20statements
&account%20number=1000100010009999
&_docid=6850-6851-SUA17-1FAAA-225712-1634-132014-132172-89-76-11-25-0
&_html=template.htm&_logoff=1

```

View Annotations

指定した文書に添付されている注釈を表示します。

目的

View Annotations 関数では、特定の文書に付加されている注釈を表示することができます。注釈を表示するためには、OnDemand アプリケーション・グループでの View Annotation 許可を持っていないければなりません。(Access 許可でも、注釈を表示することができます。)

パラメーター

表 12. View Annotations 関数

名前=値	目的
<code>_function=getnotes</code>	注釈を表示します。
<code>_server=value</code>	OnDemand サーバーの名前。
<code>_user=value</code>	OnDemand ユーザー ID。ユーザーは、表示する注釈が含まれているアプリケーション・グループごとに、Annotation View 許可を持っていないければなりません。(Application Group Access 許可を持っていれば、注釈を表示することができます。)
<code>_password=value</code>	ユーザーのパスワード。
<code>_folder=value</code>	フォルダーの名前。
<code>_html=value</code>	<p>ODWEK が出力 Web ページを生成する際に、テンプレートとして使用する HTML ファイルを判別します。この値には、ファイル名か * (アスタリスク) を指定することができます。この値がアスタリスクの場合、ODWEK は ARSWWW.INI ファイル内の TEMPLATEDIR パラメーターで名前が指定されているディレクトリーにある、GETNOTES.HTML ファイルを使用します。パス名を指定せずにファイル名を指定することもできますが、その場合、ファイルは TEMPLATEDIR パラメーターで名前が指定されているディレクトリーに置かれていないければなりません。値にパス名を含める場合、指定するパスは、TEMPLATEDIR パラメーターで名前が指定されているディレクトリーを起点とするパスでなければなりません。</p> <p>HTML ファイルの内容はすべてお客様が定義します。ただし、このファイルには以下のコメント行を含めなければなりません。</p> <pre><!--AOI# Marker--></pre> <p>コメント行の位置によって、ODWEK プログラムが出力を配置する場所が決まります。コメント行の上にあるすべての行は、ODWEK が生成する出力の前に書き込まれます。コメント行の下にあるすべての行は、ODWEK が生成する出力の後に書き込まれます。</p> <p>TEMPLATE.HTM は、ODWEK で用意されているサンプル・テンプレート・ファイルです。このサンプル・テンプレート・ファイルを使用して、View Annotations 関数で使用される、独自のテンプレート・ファイルの作成に役立てることができます。</p>

表 12. View Annotations 関数 (続き)

名前=値	目的
<code>_nohtml=value</code>	ODWEK が生成する出力のタイプを判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が HTML 出力を生成することを意味します。1 を指定した場合、ODWEK は区切り文字付きの ASCII 出力を生成します。区切り文字付き ASCII 出力の詳細については、151 ページの『付録 H. 非 HTML 出力』を参照してください。
<code>_docid=documentID</code>	表示する注釈を含んでいる文書の ID。文書 ID は、Document Hit List 関数によって戻されます。
<code>_port=value</code>	OnDemand サーバーのポート番号。デフォルト値である 0 (ゼロ) は、サービス・テーブル (WRKSRVTBLE) で指定されているポート番号をサーバーが使用することを意味します。サービス・テーブルでポート番号が指定されていない場合、OnDemand はポート番号 1445 の使用を試行します。指定した任意の値によって、ARSWWW.INI ファイル内の PORT パラメーターの値が指定変更されます。
<code>_codepage=value</code>	OnDemand データベースのコード・ページ。デフォルトのコード・ページは HTTP サーバーのコード・ページです。サーバーのコード・ページがデータベースのコード・ページと異なる場合、コード・ページを指定しなければなりません。指定した任意の値によって、ARSWWW.INI ファイル内の CODEPAGE パラメーターの値が指定変更されます。
<code>_logoff=1</code>	注釈の表示後、ユーザーを OnDemand サーバーから自動的に切断します。このパラメーターを指定することによって、アプリケーションで Logoff 関数を呼び出して、ユーザーを切断する必要がなくなります。このパラメーターに有効な値は 1 のみです。

使用法

以下のパラメーターは必須です。

```

_function
_server
_user
_password
_folder
_docid

```

以下のパラメーターはオプションです。

```

_html
_nohtml
_port
_codepage
_logoff

```

関数呼び出しのサンプル

```

http://www.company.com/cgi-bin/arswww.cgi?_function=getnotes
&_server=od400&_user=web&_password=web
&_folder=credit%20card%20statements
&_docid=6850-6851-SUA17-1FAAA-225712-1634-132014-132172-89-76-11-25-0
&_logoff=1

```

付録 B. Java サークレット・リファレンス

Java サークレットは、OnDemand サーバーへの接続の管理など、アクションの前後に関数や共通タスクを実行する、Web アプリケーションのコントローラーとして作動します。

関数は、以下の標準的なアプリケーション・タスクに対して提供されています。

- ログオンおよびログオフ
- 検索
- 文書の取り出し、印刷、更新
- 注釈の追加および表示
- パスワードの変更

アプリケーション関数およびパラメーターのセットを使用して、アプリケーションでサーケットを使用します。

Java サークレットは、CGI プログラムと同じ関数を使用します。関数のリファレンス、説明、およびパラメーターについては、65 ページの『付録 A. CGI API リファレンス』を参照してください。

付録 C. Java API リファレンス

Java API の資料は、ODWEK ソフトウェアを使用して HTML 形式で提供されます。

資料を表示する前に、システムに ODWEK ソフトウェアをインストールして、/QIBM/ProdData/OnDemand/www/ ディレクトリーにある ODApiDoc.zip ファイルから文書ファイルを抽出する必要があります。アーカイブ内のファイルのディレクトリー構造を保存する抽出メソッドを使用してください。

文書を表示するには、ファイルを抽出した後、Web ブラウザーで index.html ファイルを開きます。

付録 D. Java API プログラミング・ガイド

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) は、OnDemand サーバー上のデータにアクセスして操作する一連のクラスです。このセクションでは、Java API、Document 関数の Java インプリメンテーション、およびインターネット接続について説明します。

Java API は、以下をサポートします。

- データ・アクセスのための共通オブジェクト・モデル
- OnDemand サーバー間の検索および更新。**注:** OnDemand for OS/390 バージョン 2 サーバーにアクセスする際の制限については、1 ページの『第 1 章 概要』を参照してください。
- Java アプリケーション・ユーザーのクライアント/サーバー・インプリメンテーション

クライアント/サーバー・アーキテクチャー

API は、アプリケーション・ユーザーに便利なプログラミング・インターフェースを提供します。API は、OnDemand サーバーおよびクライアントの両方に常駐可能で (両方とも同じインターフェースを提供します)、アプリケーションはローカルにあってリモートにあっても構いません。クライアント API は、サーバーと通信し、ネットワークを介してデータにアクセスします。クライアントとサーバー間の通信は、クラスによって行われます。追加プログラムを追加する必要はありません。

API クラスは、`com.ibm.edms.od` という 1 つのパッケージから構成されます。

Java 環境のパッケージ化

API クラスは、`com.ibm.edms.od` という 1 つのパッケージに含まれています。クラスには、次のものがあります。

com.ibm.edms.od.ODCallback

このクラスは、サーバー操作で処理中にデータを戻す際に、すべてのメソッドで使用されます。

com.ibm.edms.od.ODCriteria

OnDemand フォルダーの検索基準を表すクラス。基準クラスには、検索演算子および検索値を設定するメソッドが含まれます。

com.ibm.edms.od.ODException

このクラスは、API 使用時に発生する例外を表します。

com.ibm.edms.od.ODFolder

OnDemand フォルダーを表すクラス。このオブジェクトは、`ODServer.openFolder()` への呼び出しが正常に終了した場合に戻されます。

このクラスには、フォルダー基準情報が含まれています。これらの基準オブジェクトは、サーバーでの照会を絞り込むために、変更する必要があるものです。

com.ibm.edms.od.ODHit

このクラスは、OnDemand 文書を表します。

com.ibm.edms.od.ODNote

このクラスは、OnDemand 注釈を表します。

com.ibm.edms.od.ODServer

このクラスは、OnDemand サーバーへの接続を表します。このクラスから、ログオン、ログオフ、およびパスワードの変更が可能です。正常にログオンした後、このオブジェクトには、セッションがアクセスできるすべてのフォルダーのリストが含まれています。**注:** このサーバー・オブジェクトへのアクセスは、単一スレッド環境で行わなければなりません。唯一の例外は、サーバーの操作を取り消す場合です。

プログラミングのヒント

com.ibm.edms.od パッケージを ODWEK アプリケーションにインポートする必要があります。

Java API を使用する ODWEK アプリケーションを実行するために、HTTP サーバーや Web アプリケーション・サーバーは必要ありません。ODWEK アプリケーションで Java インタープリターを実行できます。

ODWEK アプリケーションで Java インタープリターを実行するには、次のようにします。

1. arswww.ini ファイルをユーザー定義ランタイム・ディレクトリーにコピーします。
2. 共用ライブラリーを、arswww.ini ファイルをコピーしたディレクトリーにコピーします。

表 13. 共用ライブラリー・ファイル名

オペレーティング・システム	共用ライブラリー
AIX®	libarswwwsl.a
HP-UX	libarswwwsl.sl
Linux	libarswwwsl.so
Solaris	libarswwwsl.so
Windows	arswwwsl.dll

3. Windows システムの場合は、以下のファイルを、arswww.ini ファイルをコピーしたディレクトリーにコピーします。

ARSSCKNT.DLL
ARSCT32.DLL

4. アプリケーションで Java インタープリターを実行する際に、ユーザー定義ディレクトリーの名前を指定します。107 ページの『ODWEK アプリケーションの実行』の例を参照してください。

システム環境のセットアップ

AIX、HP-UX、Linux、Solaris、または Windows 環境をセットアップする場合には、以下を設定する必要があります。

パッケージ

すべての ODWEK アプリケーション用にインポートします。

- com.ibm.edms.od

ライブラリー・ファイル

AIX、HP-UX、Linux、および Solaris 用の共用オブジェクト

Windows 用の DLL

環境変数の設定

ODWEK アプリケーションを作成する場合は、環境をセットアップする必要があります。

AIX

AIX 環境で ODWEK アプリケーションを開発するための開発環境をセットアップするには、以下の環境変数を設定する必要があります。

PATH PATH に /usr/lpp/ars/www を含めるようにしてください。

LIBPATH LIBPATH に /usr/lpp/ars/www を含めるようにしてください。

LD_LIBRARY_PATH

LD_LIBRARY_PATH に /usr/lpp/ars/www を含めるようにしてください。

CLASSPATH CLASSPATH に /usr/lpp/ars/www/api/ODApi.jar (クラス・ライブラリー) を含めるようにしてください。

HP-UX

HP-UX 環境で ODWEK アプリケーションを開発するための開発環境をセットアップするには、以下の環境変数を設定する必要があります。

PATH PATH に /opt/ondemand/www を含めるようにしてください。

LIBPATH LIBPATH に /opt/ondemand/www を含めるようにしてください。

LD_LIBRARY_PATH

LD_LIBRARY_PATH に /opt/ondemand/www を含めるようにしてください。

CLASSPATH CLASSPATH に /opt/ondemand/www/api/ODApi.jar (クラス・ライブラリー) を含めるようにしてください。

Linux

Linux 環境で ODWEK アプリケーションを開発するための開発環境をセットアップするには、以下の環境変数を設定する必要があります。

PATH PATH に /opt/ondemand/www を含めるようにしてください。

LIBPATH LIBPATH に /opt/ondemand/www を含めるようにしてください。

LD_LIBRARY_PATH

LD_LIBRARY_PATH に /opt/ondemand/www を含めるようにしてください。

CLASSPATH CLASSPATH に /opt/ondemand/www/api/ODApi.jar (クラス・ライブラリー) を含めるようにしてください。

Solaris

Solaris 環境で ODWEK アプリケーションを開発するための開発環境をセットアップするには、以下の環境変数を設定する必要があります。

PATH PATH に /opt/ondemand/www を含めるようにしてください。

LIBPATH LIBPATH に /opt/ondemand/www を含めるようにしてください。

LD_LIBRARY_PATH

LD_LIBRARY_PATH に /opt/ondemand/www を含めるようにしてください。

CLASSPATH CLASSPATH に /opt/ondemand/www/api/ODApi.jar (クラス・ライブラリー) を含めるようにしてください。

Windows

Windows 環境で ODWEK アプリケーションを開発するための開発環境をセットアップするには、以下の環境変数を設定する必要があります。

PATH PATH に x :%yyyyyyyyy %DLL; を含めるようにしてください。x は ODWEK をインストールしたドライブで、yyyyyyyyy は ODWEK ソフトウェアのインストール・ディレクトリーです。

CLASSPATH CLASSPATH に x :%yyyyyyyyy %WWW %API%ODApi.jar を含めるようにしてください。x は ODWEK をインストールしたドライブで、yyyyyyyyy はクラス・ライブラリーのインストール・ディレクトリーです。

トレースおよび診断情報

Java API アプリケーションで発生する問題に対処するために、トレースおよび例外処理を使用することができます。

トレース

ARSWWW.INI ファイルの以下のパラメーターは、指定されたディレクトリー内の arswww.log ファイルにトレース情報を書き込みます。

```
[DEBUG]
LOG=1
LOGDIR=/ars/www/log
```

注: ログ・ファイルには膨大な量の情報が書き込まれる場合があるので、IBM では、ロギングは必要な場合 (問題を再現する場合など) にのみ使用可能にするようにお勧めしています。長期間ロギングを使用可能にする必要がある場合、ログ・ファイルのパスでは、必ず十分なフリー・スペースがあるストレージ・デバイス指定するようにしてください。システムから古いログ・ファイルを定期的に削除することを忘れないでください。

システムおよび文書に関する情報の収集に使用できる他のツールについては、159 ページの表 17 を参照してください。

例外処理

Java API は、問題を検出すると、例外をスローします。例外をスローすると、`ODException` クラスまたはそのいずれかのサブクラスの例外オブジェクトが作成されます。

`ODException` が作成されると、API は、ロギングが使用可能である場合、ログ・ファイルに診断情報を記録します。Java API が使用するログ・ファイルについての詳細は、106 ページの『トレース』を参照してください。

`ODException` をキャッチした場合、実行中に発生したエラー・メッセージ、エラー・コード、およびエラー状態を見ることができます。エラーをキャッチすると、例外がスローされたロケーションが示されているエラー・メッセージが発行されます。エラー ID と例外 ID も示されています。以下のコードは、スローおよびキャッチ・プロセスの例を示しています。

```
try
{
    odServer = new ODServer( );
    odServer.initialize( argv[9], "TcUpdate.java" );
    System.out.println( "Logging on to " + argv[0] + "..." );
    odServer.logon( argv[0], argv[1], argv[2] );
    odServer.logoff( );
    odServer.terminate( );
}

catch ( ODException e )
{
    System.out.println( "ODException: " + e );
    System.out.println( "  id = " + e.getErrorId( ) );
    System.out.println( "  msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}
```

定数

Java API で使用する定数は、オンライン・リファレンスに記載されています。詳しくは、101 ページの『付録 C. Java API リファレンス』を参照してください。

ODWEK アプリケーションの実行

Java インタープリターを使用して、ODWEK アプリケーションを実行することができます。ODWEK アプリケーションを作成、コンパイル、実行する際には、以下の点に注意してください。

1. Java API で使用可能なメソッドを使用して、ODWEK アプリケーションを作成します。Java API パッケージを ODWEK アプリケーション・ファイルにインポートします。次に例を示します。

```
//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class Logon
```

```

    {
        public static void main ( String argv[] )
        {
            .
            .
            .
        }
    }
}

```

2. ODWEK アプリケーション・ファイル (.java) を javac でコンパイルして、.class ファイルを作成します。Java アプリケーションのコンパイル手順は、ご使用の Java リファレンスを参照してください。
3. アプリケーション (.class ファイル) で Java インタープリターを実行します。次に例を示します。

```
java Logon server userid passwd /tmp/ondemand/www
```

ここで、Logon は .class ファイルの名前、server、userid、および passwd はアプリケーションのパラメーター、/tmp/ondemand/www は arswww.ini ファイルのコピーが含まれているユーザー定義のランタイム・ディレクトリーです。**注:** この例は、システム環境変数 (105 ページの『システム環境のセットアップ』を参照) を使用して ODWEK クラスおよびサブレット・ライブラリーへのパスを指定していることを前提としています。

OnDemand サーバーへの接続

クラス ODServer のオブジェクトは、OnDemand サーバーへの接続を提示および管理し、トランザクション・サポートを提供し、サーバー・コマンドを実行します。101 ページの『付録 C. Java API リファレンス』に、メソッドとその説明のオンライン・リファレンスの場所が記載されています。

OnDemand サーバーに接続する際は、サーバーの要件 (例えば、OnDemand のパスワードは 8 文字を超えてはならないなど) に注意してください。

接続の確立

ODServer クラスは、OnDemand サーバーへの接続および OnDemand サーバーからの切断に関するメソッドを提供します。以下の例では、LIBSRVR1 という OnDemand ライブラリー・サーバー、ユーザー ID ADMIN、およびパスワード PASSWD を使用します。例では、OnDemand サーバー用の ODServer オブジェクトを作成し、そのオブジェクトに接続して、それを使用して作業を行い (例には示されていません)、オブジェクトから切断します。

```

odServer = new ODServer( );
odServer.initialize( "c:%odwekdir", "Sample" );
System.out.println( "Logging on to " + "LIBSRVR1" + "..." );
odServer.logon( "LIBSRVR1", "ADMIN", "PASSWD" );
.
.
.
odServer.logoff( );
odServer.terminate( );

```

この例の引用元である完全なサンプル・アプリケーションについては、109 ページの『OnDemand サーバーを使用した作業』を参照してください。

パスワードの設定と入手

ODServer のメソッドを使用して、OnDemand サーバーにアクセスしたり、ユーザー・パスワードを設定することができます。以下の例で、OnDemand ライブラリー・サーバーでのユーザー・パスワードの設定および入手方法を示します。

```
odServer = new ODServer( );
odServer.setServer( "LIBSRVR1" );
odServer.setUserId( "ADMIN" );
odServer.setPassword( "PASSWD" );

System.out.println( "Logging on to " + "LIBSRVR1" + "..." );

odServer.logon( odServer.getServerName( ),
               odServer.getUserId( ),
               odServer.getPassword( ),
               ODConstant.CONNECT_TYPE_LOCAL,
               0 );
```

この例の引用元である完全なサンプル・アプリケーションについては、『OnDemand サーバーを使用した作業』を参照してください。

OnDemand サーバーを使用した作業

クラス ODServer のオブジェクトは、OnDemand サーバーへの接続を提示および管理し、トランザクション・サポートを提供し、サーバー・コマンドを実行します。

以下の例は、ODServer メソッドを使用して、ログオンの準備、アプリケーション名の設定、ローカル・ディレクトリーの表示 (オプション)、サーバー名、ユーザー ID およびパスワードの表示、接続タイプの表示および設定、ポートの表示および設定、サーバーからの切断を行います。

この例には、以下の ODServer メソッドを示します。

- initialize
- Logon
- Logoff
- terminate
- getConnectType
- getLocalDir
- getPassword
- getPort
- getServerName
- getUserId
- setApplicationName
- setConnectType
- setLocalDir
- setPassword
- setPort
- setServer
- setUserId

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID

- パスワード
- 構成ディレクトリー (arswww.ini ファイルのロケーション)
- (オプション) ローカル・サーバー・ディレクトリー

OnDemand サーバーを使用した作業の例:

```
//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcServerMisc
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        String str;
        int j;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 4 )
        {
            System.out.println( "usage: java TcServerMisc <server> <userid> <password> <config dir> [<local server dir>]" );
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "This testcase should:" );
            System.out.println( " Use ODServer methods setServer, setUserId, and setPassword" );
            System.out.println( " to prepare for logon" );
            System.out.println( " Set the application name" );
            System.out.println( " Display the" );
            System.out.println( " Local Directory" );
            System.out.println( " Server name" );
            System.out.println( " User Id" );
            System.out.println( " Password" );
            System.out.println( " Connect Type" );
            System.out.println( " Set and display the port" );
            System.out.println( " Set the connect type" );
            System.out.println( " Logoff" );
            System.out.println( "" );
            System.out.println( "Ensure that all information is correct." );
            System.out.println( "" );
            System.out.println( "-----" );
            System.out.println( "" );

            //-----
            // Logon to specified server
            //-----
            odServer = new ODServer( );
            odServer.initialize( argv[3], "TcServerMisc.java" );
            odServer.setServer( argv[0] );
            odServer.setUserId( argv[1] );
            odServer.setPassword( argv[2] );

            System.out.println( "Logging on to " + argv[0] + "..." );
            if ( argv.length == 4 )
                odServer.logon( );
            else
            {
                if ( argv.length == 5 )
                {
                    odServer.setLocalDir( argv[4] );
                    odServer.logon( odServer.getServerName( ),
                                    odServer.getUserId( ),
                                    odServer.getPassword( ),
                                    ODConstant.CONNECT_TYPE_LOCAL,
                                    0,
                                    odServer.getLocalDir( ) );
                }
            }

            //-----
            // Test miscellaneous methods
            //-----
            System.out.println( "Setting application name to TcServerMisc.java..." );
            odServer.setApplicationName( "TcServerMisc.java" );

            System.out.println( "Local Dir: " + odServer.getLocalDir( ) );
            System.out.println( "Server Name: " + odServer.getServerName( ) );
            System.out.println( "User Id: " + odServer.getUserId( ) );
            System.out.println( "Password: " + odServer.getPassword( ) );
            System.out.println( "Connect Type: " + getConnectTypeName( odServer.getConnectType( ) ) );

            j = odServer.getPort( );
        }
    }
}
```



```

System.out.println( "Setting port to " + j + "..." );
odServer.setPort( j );
System.out.println( "Port: " + j );

if ( argv.length == 4 )
{
    System.out.println( "Setting connect type to ODConstant.CONNECT_TYPE_TCPIP..." );
    odServer.setConnectType( ODConstant.CONNECT_TYPE_TCPIP );
}
else
{
    System.out.println( "Setting connect type to ODConstant.CONNECT_TYPE_LOCAL..." );
    odServer.setConnectType( ODConstant.CONNECT_TYPE_LOCAL );
}

//-----
// Cleanup
//-----
System.out.println( "Logging off..." );
odServer.logoff( );
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed - analyze if required" );
System.out.println( "" );
}

catch ( ODEException e )
{
    System.out.println( "ODEException: " + e );
    System.out.println( " id = " + e.getErrorId( ) );
    System.out.println( " msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

catch ( Exception e2 )
{
    System.out.println( "exception: " + e2 );
    e2.printStackTrace( );
}
}

static String getConnectTypeName( char type )
{
    String str;

    switch( type )
    {
        case ODConstant.CONNECT_TYPE_TCPIP:
            str = "TCPIP";
            break;
        case ODConstant.CONNECT_TYPE_LOCAL:
            str = "LOCAL";
            break;
        default:
            str = "*** Unknown connect type";
            break;
    }

    return str;
}
}

```

フォルダー内のアプリケーション・グループのリスト表示

クラス `ODFolder` のオブジェクトは、`OnDemand` フォルダを表します。

以下の例では、`ODFolder` メソッドを使用して、フォルダーから検索されたアプリケーション・グループの数を表示し、各アプリケーション・グループの名前を表示します。

この例には、以下の `ODFolder` メソッドを示します。

- `getNumApplGroups`
- `getApplGroups`
- `close`

また、この例では、`ODServer` メソッドを使用して、ログオンの準備、指定したフォルダーのオープン、ログオフを行います。この例には、以下の `ODServer` メソッドを示します。

- initialize
- Logon
- openFolder
- Logoff
- terminate

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダー名
- 構成ディレクトリー (arswww.ini ファイルのロケーション)
- (オプション) ローカル・サーバー・ディレクトリー

フォルダー内のアプリケーション・グループのリスト表示の例:

```

//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcAppGrp
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        ODFolder odFolder;
        Object[] appl_grps;
        int j;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 5 )
        {
            System.out.println( "usage: java TcAppGrp <server> <userid> <password> <folder> <config dir> [<local server dir>]" );
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "This testcase should:" );
            System.out.println( "  Logon to the specified server" );
            System.out.println( "  Open the specified folder" );
            System.out.println( "  Display the folder name" );
            System.out.println( "  Display the number of application groups" );
            System.out.println( "  Display the name of each application group" );
            System.out.println( "" );
            System.out.println( "-----" );
            System.out.println( "" );

            //-----
            // Logon to the specified server
            //-----
            odServer = new ODServer( );
            odServer.initialize( argv[4], "TcListCriteria.java" );

            System.out.println( "Logging on to " + argv[0] + "..." );
            if ( argv.length == 5 )
                odServer.logon( argv[0], argv[1], argv[2] );
            else
                if ( argv.length == 6 )
                    odServer.logon( argv[0], argv[1], argv[2], ODConstant.CONNECT_TYPE_LOCAL, 0, argv[5] );

            //-----
            // Open the specified folder
            //-----
            System.out.println( "Opening " + argv[3] + " folder..." );
            odFolder = odServer.openFolder( argv[3] );

            //-----
            // Display number and names of application groups
            //-----
            System.out.println( "There is(are) " + odFolder.getNumAppGroups( ) + " application group(s) in the folder:" );
            appl_grps = odFolder.getAppGroups( );
            for ( j = 0; j < appl_grps.length; j++ )
                System.out.println( "  " + appl_grps[j].toString( ) );

            //-----
            // Cleanup
            //-----
            odFolder.close( );
        }
    }
}

```

```

        odServer.logoff( );
        odServer.terminate( );
        System.out.println( "" );
        System.out.println( "-----" );
        System.out.println( "" );
        System.out.println( "Testcase completed - analyze results if required" );
        System.out.println( "" );
    }

    catch ( ODEException e )
    {
        System.out.println( "ODEException: " + e );
        System.out.println( " id = " + e.getErrorId( ) );
        System.out.println( " msg = " + e.getErrorMsg( ) );
        e.printStackTrace( );
    }

    catch ( Exception e2 )
    {
        System.out.println( "exception: " + e2 );
        e2.printStackTrace( );
    }
}
}

```

フォルダーの検索

クラス `ODFolder` のオブジェクトは、`OnDemand` フォルダーを表します。クラス `ODCriteria` のオブジェクトは、`OnDemand` フォルダーの検索基準を表します。クラス `ODHit` のオブジェクトは、`OnDemand` 文書を表します。

以下の例では、`ODFolder` メソッドを使用して、指定したフォルダーのオープン、フォルダー名、記述、表示順序、および検索基準の表示、フォルダーの検索、フォルダーのクローズを行います。この例では、`ODCriteria` メソッドを使用して、現行の検索オペランドおよび検索値を設定します。また、この例では、`ODHit` メソッドを使用して、文書の表示値、文書タイプ、文書の永続 ID、文書のロケーション、および文書の MIME コンテンツ・タイプを入手します。

この例には、以下の `ODFolder` メソッドを示します。

- `getName`
- `getDescription`
- `getDisplayOrder`
- `getCriteria`
- `search`
- `getSearchMessage`
- `close`

この例には、以下の `ODCriteria` メソッドを示します。

- `getName`
- `setOperand`
- `setSearchValue`
- `setSearchValues`

この例には、以下の `ODHit` メソッドを示します。

- `getDisplayValue`
- `getDisplayValues`
- `getDocType`
- `getMimeType`
- `getDocLocation`
- `getDocId`

また、この例では、ODServer メソッドを使用して、ログオンの準備、指定したフォルダーのオープン、ログオフを行います。この例には、以下の ODServer メソッドを示します。

- initialize
- Logon
- openFolder
- terminate

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダー名
- 基準名
- 演算子 (eq, ne, lt, le, gt, ge, in, ni, li, nl, be, nb のいずれかでなければなりません)
- 検索値 1
- (オプション) 検索値 2
- 構成ディレクトリー (arswww.ini ファイルのロケーション)

注: ヒット回数は、arswww.ini ファイル内の MAXHITS パラメーターによって制限されることがあります。

フォルダーの検索の例:

```
//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcSearch
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        ODFolder odFolder;
        ODCriteria odCrit;
        ODHit odHit;
        Enumeration values_enum;
        Vector hits;
        String[] display_crit;
        String header, line1, line2, hit_value, useable_value;
        boolean mismatch_detected;
        int j, k, opr;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 9 )
        {
            System.out.println( "usage: java TcSearch <server> <userid> <password> <folder> <criteria> <opr> <value1> <value2> <config dir>" );
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "This testcase should:" );
            System.out.println( "  Logon to the specified server" );
            System.out.println( "  Open the specified folder" );
            System.out.println( "  Display the folder name and description" );
            System.out.println( "  Get the specified criteria" );
            System.out.println( "  Set the operator" );
            System.out.println( "  Set the operand(s)" );
            System.out.println( "  Search the folder" );
            System.out.println( "  Display search message (if any)" );
            System.out.println( "  Display the number of hits" );
            System.out.println( "  Display the hitlist with each hit using 3 lines:" );
            System.out.println( "    1. The hit values returned by the ODHit.getDisplayValue method" );
            System.out.println( "    2. The hit values returned by the ODHit.getDisplayValues method" );
            System.out.println( "    3. The doc type, mime type, doc location, and doc id values" );
            System.out.println( "" );
            System.out.println( "Ensure that lines 1 and 2 of the hitlist are the same and that the" );
            System.out.println( "hitlist values are the same as those displayed using the Windows Client." );
            System.out.println( "If arswww.ini is restricting the number of hits, there may be fewer" );
            System.out.println( "hits than displayed using the Windows Client." );
            System.out.println( "" );
            System.out.println( "-----" );
            System.out.println( "" );
        }
    }
}
```

```

//-----
// Logon to specified server
//-----
odServer = new ODServer( );
odServer.initialize( argv[8], "TcSearch.java" );
System.out.println( "Logging on to " + argv[0] + "..." );
odServer.logon( argv[0], argv[1], argv[2] );

//-----
// Open the specified folder and find the requested criteria
//-----
System.out.println( "Opening " + argv[3] + " folder..." );
odFolder = odServer.openFolder( argv[3] );
System.out.println( "Name=" + odFolder.getName( ) + " Desc=" + odFolder.getDescription( ) + " " );
System.out.println( "Getting " + argv[4] + " criteria..." );
odCrit = odFolder.getCriteria( argv[4] );

//-----
// Convert the operator parameter to the internal operator value and set
// the criteria operator
//-----
System.out.println( "Setting operator to " + argv[5] + "..." );
if ( argv[5].equals( "eq" ) )
    opr = ODConstant.OPEqual;
else if ( argv[5].equals( "ne" ) )
    opr = ODConstant.OPNotEqual;
else if ( argv[5].equals( "lt" ) )
    opr = ODConstant.OPLessThan;
else if ( argv[5].equals( "le" ) )
    opr = ODConstant.OPLessThanEqual;
else if ( argv[5].equals( "gt" ) )
    opr = ODConstant.OPGreaterThan;
else if ( argv[5].equals( "ge" ) )
    opr = ODConstant.OPGreaterThanEqual;
else if ( argv[5].equals( "in" ) )
    opr = ODConstant.OPIn;
else if ( argv[5].equals( "ni" ) )
    opr = ODConstant.OPNotIn;
else if ( argv[5].equals( "li" ) )
    opr = ODConstant.OPLike;
else if ( argv[5].equals( "nl" ) )
    opr = ODConstant.OPNotLike;
else if ( argv[5].equals( "be" ) )
    opr = ODConstant.OPBetween;
else if ( argv[5].equals( "nb" ) )
    opr = ODConstant.OPNotBetween;
else
    opr = -1;

System.out.println( "Setting operand(s)..." );
odCrit.setOperand( opr );

if ( opr == ODConstant.OPBetween || opr == ODConstant.OPNotBetween )
{
    odCrit.setSearchValues( argv[6], argv[7] );
    System.out.println( " " + odCrit.getName( ) + " " + getOperatorName( opr ) + " " + argv[6] + " and " + argv[7] );
}
else
{
    odCrit.setSearchValue( argv[6] );
    System.out.println( " " + odCrit.getName( ) + " " + getOperatorName( opr ) + " " + argv[6] );
}

//-----
// Search the folder
//-----
System.out.println( " Searching " + argv[3] + "..." );
hits = odFolder.search( );
System.out.println( " Search message: " + odFolder.getSearchMessage( ) );
System.out.println( " Number of hits: " + hits.size( ) );

//-----
// Display the hits
//-----
mismatch_detected = false;
if ( hits != null && hits.size( ) > 0 )
{
    display_crit = odFolder.getDisplayOrder( );
    header = " ";
    for ( j = 0; j < display_crit.length; j++ )
        header = header + display_crit[j] + "-";
    System.out.println( " -----" );
    System.out.println( header + " (from ODHit.getDisplayValue method)" );
    System.out.println( header + " (from ODHit.getDisplayValues method)" );
    System.out.println( " DocType--MimeType--DocLocation--DocId" );
    System.out.println( " -----" );
    for ( j = 0; j < hits.size( ); j++ )
    {
        odHit = (ODHit)hits.elementAt( j );
        line1 = " ";
        for ( k = 0; k < display_crit.length; k++ )
        {
            hit_value = odHit.getDisplayValue( display_crit[k] );
            useable_value = ( hit_value.equals( "" ) ) ? " " : hit_value;
            line1 = line1 + useable_value + "-";
        }
        System.out.println( line1 );
        line2 = " ";
        for ( values_enum = odHit.getDisplayValues( ); values_enum.hasMoreElements( ); )
        {
            hit_value = (String)values_enum.nextElement( );
            useable_value = ( hit_value.equals( "" ) ) ? " " : hit_value;
            line2 = line2 + useable_value + "-";
        }
        System.out.println( line2 );
        System.out.println( " " + getDocTypeString( odHit.getDocType( ) ) +
            " " + odHit.getMimeType( ) +
            " " + getLocationString( odHit.getDocLocation( ) ) +
            " " + odHit.getDocId( ) );
    }
}

```

```

        if ( !line1.equals( line2 ) )
            mismatch_detected = true;
    }
}

//-----
// Cleanup
//-----
odFolder.close( );
odServer.logoff( );
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed - analyze if required" );
System.out.println( "" );
if ( mismatch_detected )
{
    System.out.println( "*** At least one mismatch was found between" );
    System.out.println( "*** lines 1 and 2 of a hit" );
    System.out.println( "" );
}
}

catch ( ODEException e )
{
    System.out.println( "ODEException: " + e );
    System.out.println( " id = " + e.getErrorId( ) );
    System.out.println( " msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

catch ( Exception e2 )
{
    System.out.println( "exception: " + e2 );
    e2.printStackTrace( );
}
}

static String getOperatorName( int oper )
{
    String str;

    switch( oper )
    {
        case ODConstant.OPEqual:
            str = "Equals";
            break;
        case ODConstant.OPNotEqual:
            str = "Not Equal";
            break;
        case ODConstant.OPLessThan:
            str = "Less Than";
            break;
        case ODConstant.OPLessThanEqual:
            str = "Less Than or Equal";
            break;
        case ODConstant.OPGreaterThan:
            str = "Greater Than";
            break;
        case ODConstant.OPGreaterThanEqual:
            str = "Greater Than or Equal";
            break;
        case ODConstant.OPIn:
            str = "In";
            break;
        case ODConstant.OPNotIn:
            str = "Not In";
            break;
        case ODConstant.OPLike:
            str = "Like";
            break;
        case ODConstant.OPNotLike:
            str = "Not Like";
            break;
        case ODConstant.OPBetween:
            str = "Between";
            break;
        case ODConstant.OPNotBetween:
            str = "Not Between";
            break;
        default:
            str = "Operator unknown";
            break;
    }
}

return str;
}

static String getDocTypeString( char type )
{
    String str;

    switch( type )
    {
        case ODConstant.FileTypeAFP:
            str = "AFP";
            break;
        case ODConstant.FileTypeBMP:
            str = "BMP";
            break;
        case ODConstant.FileTypeEMAIL:
            str = "EMAIL";
            break;
        case ODConstant.FileTypeGIF:
            str = "GIF";
            break;
        case ODConstant.FileTypeJFIF:
            str = "JFIF";
            break;
    }
}

```

```

        case ODCConstant.FileTypeLINE:
            str = "LINE";
            break;
        case ODCConstant.FileTypeMETA:
            str = "META";
            break;
        case ODCConstant.FileTypeNONE:
            str = "NONE";
            break;
        case ODCConstant.FileTypePCX:
            str = "PCX";
            break;
        case ODCConstant.FileTypePDF:
            str = "PDF";
            break;
        case ODCConstant.FileTypePNG:
            str = "PNG";
            break;
        case ODCConstant.FileTypeTIFF:
            str = "TIFF";
            break;
        case ODCConstant.FileTypeUSRDEF:
            str = "USRDEF";
            break;
        default:
            str = "*** Invalid Doc Type ***";
            break;
    }

    return str;
}

static String getLocationString( int loc )
{
    String str;

    switch( loc )
    {
        case ODCConstant.DocLocationCache:
            str = "Cache";
            break;
        case ODCConstant.DocLocationArchive:
            str = "Archive";
            break;
        case ODCConstant.DocLocationExternal:
            str = "External";
            break;
        case ODCConstant.DocLocationUnknown:
            str = "Unknown";
            break;
        default:
            str = "*** Invalid Doc Location ***";
            break;
    }

    return str;
}
}

```

SQL ストリングを使用したフォルダーの検索

以下の例では、ODFolder メソッドを使用して、指定したフォルダーのオープン、指定した SQL ストリングによるフォルダーの検索、フォルダーのクローズを行います。この例は、ODHit メソッドを使用して、照会に一致する項目の数、および文書リストを表示します。

この例には、以下の ODFolder メソッドを示します。

- setAppGroupForSearchWithSQL
- search
- getDisplayOrder
- close

この例には、以下の ODHit メソッドを示します。

- getDisplayValue

また、この例では、ODServer メソッドを使用して、ログオンの準備、指定したフォルダーのオープン、ログオフを行います。この例には、以下の ODSERVER メソッドを示します。

- initialize
- Logon
- openFolder
- Logoff

- terminate

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダー名
- アプリケーション・グループ名
- SQL ストリング
- 構成ディレクトリー (arswww.ini ファイルのロケーション)

SQL ストリングを使用したフォルダーの検索の例:

```
//*****
//
// Testcase: TcSearchWithSQL
//
// This testcase will:
// Logon to the specified server
// Open the specified folder
// Search the folder with the SQL string
// Display the number of hits
// Display the hitlist
//
// Tests the following methods:
// ODServer
//   initialize
//   logon
//   openFolder
//   logoff
//   terminate
// ODFolder
//   setApplGroupForSearchWithSQL
//   search
//   getDisplayOrder
//   close
// ODHit
//   getDisplayValue
//
// Parameters:
// 1. Server name
// 2. User Id
// 3. Password
// 4. Folder name
// 5. Appl Group name
// 6. SQL string
// 7. Configuration directory (contains arswwww.ini)
//
//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcSearchWithSQL
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        ODFolder odFolder;
        ODHit odHit;
        Enumeration values_enum;
        Vector hits;
        String[] display_crit;
        String server, userid, password, folder, directory;
        String sql, appl_group;
        String header, line, hit_value, useable_value;
        int j, k;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 6 )
        {
            System.out.println( "usage: java TcSearchWithSQL <server> <userid> <password> <folder> <appl group> <sql string> <config dir>" );
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "This testcase should:" );
            System.out.println( " Logon to the specified server" );
            System.out.println( " Open the specified folder" );
            System.out.println( " Search the folder with the specified SQL string" );
            System.out.println( " Display the number of hits" );
            System.out.println( " Display the hitlist" );
            System.out.println( "" );
            System.out.println( "-----" );
            System.out.println( "" );

            //-----
            // Logon to specified server
            //-----

```



```

server    = argv[0];
userid   = argv[1];
password = argv[2];
folder   = argv[3];
appl_group = argv[4];
sql      = argv[5];
directory = argv[6];

odServer = new ODServer( );
odServer.initialize( directory, "TcSearchWithSQL.java" );
System.out.println( "Logging on to " + server + "..." );
odServer.logon( server, userid, password );

//-----
// Open the specified folder
//-----
System.out.println( "Opening " + folder + " folder..." );
odFolder = odServer.openFolder( folder );

//-----
// Search the folder
//-----
if ( appl_group.length( ) > 0 )
{
    System.out.println( "Setting Appl Group to search: " + appl_group );
    odFolder.setAppGroupForSearchWithSQL( appl_group );
}

//-----
// Search the folder
//-----
System.out.println( " Searching " + folder + "..." );
hits = odFolder.search( sql );
System.out.println( "    Number of hits: " + hits.size( ) );

//-----
// Display the hits
//-----
if ( hits != null && hits.size( ) > 0 )
{
    display_crit = odFolder.getDisplayOrder( );
    header = " ";
    for( j = 0; j < display_crit.length; j++ )
        header = header + display_crit[j] + "--";
    System.out.println( " -----" );
    System.out.println( header );
    System.out.println( " -----" );
    for ( j = 0; j < hits.size( ); j++ )
    {
        odHit = (ODHit)hits.elementAt( j );
        line = " ";
        for ( k = 0; k < display_crit.length; k++ )
        {
            hit_value = odHit.getDisplayValue( display_crit[k] );
            useable_value = ( hit_value.equals( "" ) ) ? " " : hit_value;
            line = line + useable_value + "--";
        }
        System.out.println( line );
    }
}

//-----
// Cleanup
//-----
odFolder.close( );
odServer.logoff( );
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed - analyze if required" );
System.out.println( "" );
}

catch ( ODEException e )
{
    System.out.println( "ODEException: " + e );
    System.out.println( "    id = " + e.getErrorId( ) );
    System.out.println( "    msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

catch ( Exception e2 )
{
    System.out.println( "exception: " + e2 );
    e2.printStackTrace( );
}
}
}

```

検索の取り消し

以下の例は、ODServer.cancel メソッドを使用して、進行中の検索を取り消します。

この例では、ODServer、ODFolder、および ODCriteria メソッドを使用して、サーバーにログオンし、フォルダーをオープンして、日付基準を 1970-2001 に設定します。その後、2 番目のスレッドが開始され、検索が行われます。2 番目のスレッド

が完了したら、ヒット数が表示されます。再度、2 番目のスレッドが開始され、検索が行われます。プロセスは、.5 秒間スリープに入り、その後、検索が取り消されます。2 番目のスレッドが完了したら、ヒット数が表示されます。

この例には、以下の ODServer メソッドを示します。

- initialize
- Logon
- openFolder
- Logoff
- terminate

この例には、以下の ODFolder メソッドを示します。

- getCriteria
- search
- close

この例には、以下の ODCriteria メソッドを示します。

- setOperand
- setSearchValues

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダー名
- 構成ディレクトリー (arswww.ini ファイルのロケーション)
- (オプション) ローカル・サーバー・ディレクトリー

検索の取り消しの例:

```
//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

class TestThread extends Thread
{
    ODFolder odFolder;

    TestThread( ODFolder fld )
    {
        odFolder = fld;
    }

    public void run( )
    {
        Vector hits;

        try
        {
            System.out.println( " Second thread Searching..." );
            hits = odFolder.search( );
            System.out.println( " Search completed - Number of hits: " + hits.size( ) );
        }

        catch ( ODEException e )
        {
            System.out.println( "ODEException: " + e );
            System.out.println( " id = " + e.getErrorId( ) );
            System.out.println( " msg = " + e.getErrorMsg( ) );
            e.printStackTrace( );
        }

        catch ( Exception e2 )
        {
            System.out.println( "exception: " + e2 );
            e2.printStackTrace( );
        }
    }
}

public class TcancelSearch
{
    public static void main ( String argv[] )
    {

```

```

ODServer odServer;
ODFolder odFolder;
ODCriteria odCrit;
TestThread search_thread;
int j;

//-----
// If too few parameters, display syntax and get out
//-----
if ( argv.length < 5 )
{
    System.out.println( "usage: java TcCancelSearch <server> <userid> <password> <folder> <config dir> [<local server dir>]" );
    return;
}

try
{
    //-----
    // Set the stage
    //-----
    System.out.println( "This testcase should:" );
    System.out.println( "  Logon to the specified server" );
    System.out.println( "  Open the specified folder" );
    System.out.println( "  Set the Date criteria to 1970-2001" );
    System.out.println( "  Initiate a second thread to perform the search" );
    System.out.println( "  When second thread completes, display the number of hits" );
    System.out.println( "  Initiate a second thread to perform the search" );
    System.out.println( "  Sleep for .5 seconds" );
    System.out.println( "  Cancel the search" );
    System.out.println( "  When second thread completes, display the number of hits" );
    System.out.println( "" );
    System.out.println( "Ensure that a folder is chosen that includes a criteria named Date." );
    System.out.println( "Ensure that the folder contains many hits and that arswww.ini is" );
    System.out.println( "not overly restricting the number of hits which can be returned." );
    System.out.println( "" );
    System.out.println( "-----" );
    System.out.println( "" );

    //-----
    // Logon to specified server
    //-----
    odServer = new ODServer( );
    odServer.initialize( argv[4], "TcCancelSearch.java" );

    System.out.println( "Logging on to " + argv[0] + "..." );
    if ( argv.length == 5 )
        odServer.logon( argv[0], argv[1], argv[2] );
    else
        if ( argv.length == 6 )
            odServer.logon( argv[0], argv[1], argv[2], ODConstant.CONNECT_TYPE_LOCAL, 0, argv[5] );

    //-----
    // Open the specified folder and display its name and description
    //-----
    System.out.println( "Opening " + argv[3] + "..." );
    odFolder = odServer.openFolder( argv[3] );
    odCrit = odFolder.getCriteria( "Date" );
    odCrit.setOperand( ODConstant.OPBetween );
    odCrit.setSearchValues( "01/01/70", "01/01/01" );

    //-----
    // Start a search on a different thread, sleep briefly, awake and cancel search
    //-----
    System.out.println( "Main thread initiating search (will not attempt to cancel)..." );
    search_thread = new TestThread( odFolder );
    search_thread.start( );
    search_thread.join( );

    System.out.println( "Main thread initiating search (will attempt to cancel)..." );
    search_thread = new TestThread( odFolder );
    search_thread.start( );
    System.out.println( "Main thread sleeping for .5 seconds..." );
    ( Thread.currentThread( ) ).sleep( 500 );
    System.out.println( "Main thread attempting to cancel search..." );
    odServer.cancel( );
    System.out.println( "Main thread returned from attempt to cancel" );
    search_thread.join( );

    //-----
    // Cleanup
    //-----
    odFolder.close( );
    odServer.logoff( );
    odServer.terminate( );
    System.out.println( "" );
    System.out.println( "-----" );
    System.out.println( "" );
    System.out.println( "Testcase completed - Ensure that the second search," );
    System.out.println( "  which was cancelled, yielded fewer hits than the first" );
    System.out.println( "" );
}

catch ( ODEException e )
{
    System.out.println( "ODEException: " + e );
    System.out.println( "  id = " + e.getErrorId( ) );
    System.out.println( "  msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

catch ( Exception e2 )
{
    System.out.println( "exception: " + e2 );
}

```

```

        e2.printStackTrace( );
    }
}

```

検索基準のリスト表示

以下の例では、ODCriteria メソッドを使用して、指定したフォルダーに対する検索基準をリストする方法を示します。この例では、それぞれの検索フィールドごとに、検索フィールドの名前、デフォルト演算子、フィールドに有効な演算子、フィールド・タイプ、およびデフォルト検索値をリストします。デフォルト値は、ODCriteria.getSearchValues および ODCriteria.getValues メソッドによってリストされます。固定検索値は、FixedChoice または Segment として定義されている検索フィールドについてリストされます。

この例には、以下の ODCriteria メソッドを示します。

- setOperand
- getValidOperands
- getType
- getValues
- setSearchValues
- getFixedValues

この例には、以下の ODServer メソッドを示します。

- initialize
- Logon
- openFolder
- Logoff
- terminate

この例には、以下の ODFolder メソッドを示します。

- getCriteria
- close

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダー名
- 構成ディレクトリ (arswww.ini ファイルのロケーション)
- (オプション) ローカル・サーバー・ディレクトリ

検索基準へのアクセスの例:

```

//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcListCriteria
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        ODFolder odFolder;
        ODCriteria odCrit;
        Enumeration crit_enum;
        Vector value_vec;
        String[] search_values, fixed_values;
        int[] valid_oprs;
        int j, opr;
        char field_type;
    }
}

```

```

//-----
// If too few parameters, display syntax and get out
//-----
if ( argv.length < 5 )
{
    System.out.println( "usage: java TcListCriteria <server> <userid> <password> <folder> <config dir> [<local server dir>]" );
    return;
}

try
{
    //-----
    // Set the stage
    //-----
    System.out.println( "This testcase should:" );
    System.out.println( " Logon to the specified server" );
    System.out.println( " Open the specified folder" );
    System.out.println( " Display the folder name and description" );
    System.out.println( " Display the number of folder criteria" );
    System.out.println( " For each criteria, display the" );
    System.out.println( "     Name" );
    System.out.println( "     Default operator" );
    System.out.println( "     Valid operators" );
    System.out.println( "     Field Type" );
    System.out.println( "     Default values (by ODCrit.getSearchValues method)" );
    System.out.println( "     Default values (by ODCrit.getValues method)" );
    System.out.println( "     Fixed values (only for FixedChoice and Segment criteria)" );
    System.out.println( "" );
    System.out.println( "Ensure that none of the operators indicates 'Unknown operator,'" );
    System.out.println( "that none of the field types indicates 'Unknown type', that the" );
    System.out.println( "default values are the same for each method, and that all" );
    System.out.println( "information is the same as that displayed using the Windows Client." );
    System.out.println( "" );
    System.out.println( "-----" );
    System.out.println( "" );

    //-----
    // Logon to the specified server
    //-----
    odServer = new ODServer( );
    odServer.initialize( argv[4], "TcListCriteria.java" );

    System.out.println( "Logging on to " + argv[0] + "..." );
    if ( argv.length == 5 )
        odServer.logon( argv[0], argv[1], argv[2] );
    else
        if ( argv.length == 6 )
            odServer.logon( argv[0], argv[1], argv[2], ODCConstant.CONNECT_TYPE_LOCAL, 0, argv[5] );

    //-----
    // Open the specified folder and display its name and description
    //-----
    System.out.println( "Opening " + argv[3] + " folder..." );
    odFolder = odServer.openFolder( argv[3] );
    System.out.println( "Name=" + odFolder.getName( ) + " Desc=" + odFolder.getDescription( ) + "" );
    System.out.println( "There are " + odFolder.getNumCriteria( ) + " criteria:" );

    //-----
    // For each folder criteria,
    //-----
    for ( crit_enum = odFolder.getCriteria( ); crit_enum.hasMoreElements( ); )
    {
        //-----
        // Display criteria name
        //-----
        System.out.println( "" );
        odCrit = (ODCriteria)crit_enum.nextElement( );
        System.out.println( odCrit.getName( ) );

        //-----
        // Display default operator
        //-----
        opr = odCrit.getOperand( );
        System.out.println( " Default operator: " );
        System.out.println( " " + getOperatorName( opr ) );

        //-----
        // Display valid operators
        //-----
        valid_oprs = odCrit.getValidOperands( );
        System.out.println( " Valid operators:" );
        for ( j = 0; j < valid_oprs.length; j++ )
            System.out.println( " " + getOperatorName( valid_oprs[j] ) );

        //-----
        // Display field type
        //-----
        field_type = odCrit.getType( );
        System.out.println( " Type:" );
        System.out.println( " " + getTypeName( field_type ) );

        //-----
        // Display default value(s) using ODCrit.getValues( )
        //-----
        value_vec = odCrit.getValues( );
        System.out.println( " Default Value(s) (ODCrit.getValues method):" );
        System.out.println( " " + value_vec.elementAt( 0 ) + "" );
        System.out.println( " " + value_vec.elementAt( 1 ) + "" );

        //-----
        // Display default value(s) using ODCrit.getSearchValues( )
        //-----
        search_values = odCrit.getSearchValues( );

```

```

System.out.println(" Default Values (ODCrit.getSearchValues method):");
for ( j = 0; j < search_values.length; j++ )
    System.out.println(" " + search_values[j] + " ");

//-----
// Display fixed choices
//-----
switch ( field_type )
{
    case ODConstant.InputTypeChoice:
    case ODConstant.InputTypeSegment:
        fixed_values = odCrit.getFixedValues( );
        System.out.println(" Fixed Values (only for field types FixedChoice and Segment:");
        for ( j = 0; j < fixed_values.length; j++ )
            System.out.println(" " + fixed_values[j] + " ");
        break;
}

//-----
// Cleanup
//-----
odFolder.close( );
odServer.logoff( );
odServer.terminate( );
System.out.println( " " );
System.out.println( " " );
System.out.println( "-----" );
System.out.println( "Testcase completed - analyze and compare results to" );
System.out.println( " Windows Client if required" );
System.out.println( " " );
}

catch ( ODEException e )
{
    System.out.println( "ODEException: " + e );
    System.out.println( " id = " + e.getErrorId( ) );
    System.out.println( " msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

catch ( Exception e2 )
{
    System.out.println( "exception: " + e2 );
    e2.printStackTrace( );
}
}

static String getOperatorName( int oper )
{
    String str;

    switch( oper )
    {
        case ODConstant.OPEqual:
            str = "Equal";
            break;
        case ODConstant.OPNotEqual:
            str = "Not Equal";
            break;
        case ODConstant.OPLessThan:
            str = "Less Than";
            break;
        case ODConstant.OPLessThanEqual:
            str = "Less Than or Equal";
            break;
        case ODConstant.OPGreaterThan:
            str = "Greater Than";
            break;
        case ODConstant.OPGreaterThanEqual:
            str = "Greather Than or Equal";
            break;
        case ODConstant.OPIn:
            str = "In";
            break;
        case ODConstant.OPNotIn:
            str = "Not In";
            break;
        case ODConstant.OPLike:
            str = "Like";
            break;
        case ODConstant.OPNotLike:
            str = "Not Like";
            break;
        case ODConstant.OPBetween:
            str = "Between";
            break;
        case ODConstant.OPNotBetween:
            str = "Not Between";
            break;
        default:
            str = "*** Unknown operator";
            break;
    }

    return str;
}

static String getTypeName( char type )
{
    String str;

    switch( type )

```

```

    {
        case ODConstant.InputTypeNormal:
            str = "Normal";
            break;
        case ODConstant.InputTypeTextSearch:
            str = "TextSearch";
            break;
        case ODConstant.InputTypeNoteTextSearch:
            str = "NoteTextSearch";
            break;
        case ODConstant.InputTypeNoteColor:
            str = "NoteColor";
            break;
        case ODConstant.InputTypeChoice:
            str = "FixedChoice";
            break;
        case ODConstant.InputTypeSegment:
            str = "Segment";
            break;
        default:
            str = "*** Unknown type";
            break;
    }
    return str;
}
}

```

フォルダーおよびフォルダー情報のリスト表示

以下の例では、ODServer メソッドを使用して、指定したユーザー ID により検索することができる、指定したサーバー上のフォルダーの数を示す行を印刷します。例では、各フォルダーごとに、フォルダー名と記述を示す行を 1 行ずつ印刷します。

この例には、以下の ODServer メソッドを示します。

- initialize
- Logon
- getNumFolders
- getFolderNames
- getFolderDescription
- Logoff
- terminate

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- 構成ディレクトリー (arswww.ini ファイルのロケーション)
- (オプション) ローカル・サーバー・ディレクトリー

フォルダーおよびフォルダー情報のリスト表示の例:

```

//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcListFolders
{
    public static void main ( String argv[] )
    {
        ODServer    odServer;
        Enumeration folders_enum;
        String folder_name, folder_desc;
        int num_folders;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 4 )
        {
            System.out.println( "usage: java TcListFolders <server> <userid> <password> <config dir> [<local server dir>]" );
            return;
        }
    }

    try

```

```

{
//-----
// Set the stage
//-----
System.out.println( "This testcase should:" );
System.out.println( " Display a line showing number of folders on the server available to the userid" );
System.out.println( " Display one line for each folder, showing name and description" );
System.out.println( "" );
System.out.println( "The information should be the same as that displayed using the Windows Client" );
System.out.println( "(with the 'All' button checked if available), but the sequence of the folders" );
System.out.println( "may be different depending on the server specified" );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );

//-----
// Logon to specified server
//-----
odServer = new ODServer( );
odServer.initialize( argv[3], "TcListFolders.java" );

System.out.println( "Logging on to " + argv[0] + "..." );
if ( argv.length == 4 )
odServer.logon( argv[0], argv[1], argv[2] );
else
if ( argv.length == 5 )
odServer.logon( argv[0], argv[1], argv[2], ODConstant.CONNECT_TYPE_LOCAL, 0, argv[4] );

//-----
// Display the number of folders available.
//-----
num_folders = odServer.getNumFolders( );
System.out.println( "" );
System.out.println( "There are " + num_folders + " folders available to " + argv[1] + " on " + argv[0] + ":" );

//-----
// Display the folder names and descriptions
//-----
for ( folders_enum = odServer.getFolderNames( ); folders_enum.hasMoreElements( ); )
{
folder_name = (String)folders_enum.nextElement( );
folder_desc = odServer.getFolderDescription( folder_name );
System.out.println( " " + folder_name + " --- " + folder_desc );
}

//-----
// Cleanup
//-----
odServer.logoff( );
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed - compare results to Windows Client if required" );
System.out.println( "" );
}

catch ( ODEException e )
{
System.out.println( "ODEException: " + e );
System.out.println( " id = " + e.getErrorId( ) );
System.out.println( " msg = " + e.getErrorMsg( ) );
e.printStackTrace( );
}

catch ( Exception e2 )
{
System.out.println( "exception: " + e2 );
e2.printStackTrace( );
}
}
}

```

文書のリストの表示

以下の例では、ODFolder および ODHit メソッドを使用して、デフォルトの検索基準によるフォルダーの検索、照会で一致した文書の数の印刷、照会で一致した文書をリストします。

この例には、以下の ODFolder メソッドを示します。

- getName
- getDisplayOrder
- search
- close

この例には、以下の ODHit メソッドを示します。

- getDisplayValue

この例には、以下の ODServer メソッドも示します。

- initialize
- Logon
- openFolder
- Logoff
- terminate

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダー名
- 構成ディレクトリー (arswww.ini ファイルのロケーション)

文書のリストの表示の例:

```
//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcSortedHitlist
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        ODFolder odFolder;
        ODHit odHit;
        Vector hits;
        String[] display_crit;
        String server, userid, password, folder, value;
        int j, k;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 5 )
        {
            System.out.println( "usage: java TcSortedHitlist <server> <userid> <password> <folder> <config dir>" );
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "This testcase should:" );
            System.out.println( "  Logon to the specified server" );
            System.out.println( "  Open the specified folder" );
            System.out.println( "  Search the folder using the default criteria" );
            System.out.println( "  Display search message (if any)" );
            System.out.println( "  Display the number of hits" );
            System.out.println( "  Display the hitlist" );
            System.out.println( "  " );
            System.out.println( "-----" );
            System.out.println( "  " );

            //-----
            // Logon to the server
            //-----
            server = argv[0];
            userid = argv[1];
            password = argv[2];
            folder = argv[3];
            odServer = new ODServer( );
            odServer.initialize( argv[4], "TcSortedHitlist.java" );
            System.out.println( "Logging on to " + server + " as " + userid + "/" + password + "..." );
            odServer.logon( server, userid, password );

            //-----
            // Open and search the folder
```

```

//-----
System.out.println( "Opening " + folder + "..." );
odFolder = odServer.openFolder( folder );
System.out.println( "Searching folder with default criteria..." );
hits = odFolder.search( );
System.out.println( " Number of hits: " + hits.size( ) );

//-----
// Display the hits
//-----
if ( hits != null && hits.size( ) > 0 )
{
    display_crit = odFolder.getDisplayOrder( );
    value = " ";
    for( j = 0; j < display_crit.length; j++ )
        value = value + display_crit[j] + " ";
    System.out.println( value );
    for ( j = 0; j < hits.size( ); j++ )
    {
        odHit = (ODHit)hits.elementAt( j );
        value = " ";
        for ( k = 0; k < display_crit.length; k++ )
            value = value + odHit.getDisplayValue( display_crit[k] ) + " ";
        System.out.println( value );
    }
}

//-----
// Cleanup
//-----
odFolder.close( );
odServer.logoff( );
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed - Ensure that the order of the hits" );
System.out.println( " is the same as shown by the Windows Client" );
System.out.println( "" );
}

catch ( ODException e )
{
    System.out.println( "ODException: " + e );
    System.out.println( " id = " + e.getErrorId( ) );
    System.out.println( " msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

catch ( Exception e2 )
{
    System.out.println( "exception: " + e2 );
    e2.printStackTrace( );
}
}
}

```

文書の検索

以下の例には、3 つの異なる文書検索メソッドを示します。

- ODServer
- ODFolder
- ODHit

この例では、指定したサーバーにログオンし、指定したフォルダーを開いて、デフォルト基準を使用してフォルダーを検索し、ヒット数を表示して、ODHit.retrieve を使用して最初にヒットしたデータを検索し、ODServer.retrieve を使用して最初にヒットしたデータを検索し、ODFolder.retrieve を使用して最初にヒットしたデータを検索します。さらに、この例では、各メソッドで検索したデータの長さを表示し、各メソッドで検索した長さおよびデータを比較して、比較結果を表示します。

この例には、以下の ODServer メソッドを示します。

- initialize

- Logon
- openFolder
- retrieve
- Logoff
- terminate

この例には、以下の ODFolder メソッドを示します。

- search
- retrieve
- close

この例には、以下の ODHit メソッドを示します。

- getDocId
- retrieve

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダー名
- 構成ディレクトリー (arswww.ini ファイルのロケーション)
- (オプション) ローカル・サーバー・ディレクトリー

文書の検索の例:

```

//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcRetrieve
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        ODFolder odFolder;
        ODHit odHit;
        TcCallback callback;
        Vector hits;
        Vector hit_to_retrieve;
        byte[] data_from_hit;
        byte[] data_from_server;
        byte[] data_from_folder;
        int j;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 5 )
        {
            System.out.println( "usage: java TcRetrieve <server> <userid> <password> <folder> <config dir> [<local server dir>]" );
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "This testcase should:" );
            System.out.println( " Logon to the specified server" );
            System.out.println( " Open the specified folder" );
            System.out.println( " Search the folder using the default criteria" );
            System.out.println( " Display the number of hits" );
            System.out.println( " Retrieve the data for the first hit using ODHit.retrieve" );
            System.out.println( " Retrieve the data for the first hit using ODServer.retrieve" );
            System.out.println( " Retrieve the data for the first hit using ODFolder.retrieve" );
            System.out.println( " Display length of data retrieved from each method" );
            System.out.println( " Compare the lengths and data retrieved from each method" );
            System.out.println( " Display the result of the comparisons" );
            System.out.println( "" );
            System.out.println( "-----" );
            System.out.println( "" );

            //-----
            // Logon to specified server
            //-----
            odServer = new ODServer( );

```

```

odServer.initialize( argv[4], "TcRetrieve.java" );
System.out.println( "Logging on to " + argv[0] + "..." );
if ( argv.length == 5 )
    odServer.logon( argv[0], argv[1], argv[2] );
else
    odServer.logon( argv[0], argv[1], argv[2], ODConstant.CONNECT_TYPE_LOCAL, 0, argv[5] );

//-----
// Open the specified folder and search with the default criteria
//-----
System.out.println( "Opening " + argv[3] + " folder..." );
odFolder = odServer.openFolder( argv[3] );
System.out.println( "Searching with default criteria..." );
hits = odFolder.search( );
System.out.println( "Number of hits: " + hits.size( ) );

//-----
// Do some retrieves and comparisons
//-----
if ( hits.size( ) > 0 )
{
    odHit = (ODHit)hits.elementAt( 0 );
    System.out.println( "Retrieving data from first hit using ODHit.retrieve..." );
    data_from_hit = odHit.retrieve( "" );
    System.out.println( "Retrieving data from first hit using ODServer.retrieve..." );
    data_from_server = odServer.retrieve( odHit.getDocId( ), argv[3], "" );
    hit_to_retrieve = new Vector( );
    hit_to_retrieve.addElement( odHit );
    System.out.println( "Retrieving data from first hit using ODFolder.retrieve (uses callback method)..." );
    callback = new TcCallback( );
    odFolder.retrieve( hit_to_retrieve, callback );
    data_from_folder = callback.getData( );
    System.out.println( "Length of data from:" );
    System.out.println( "  ODHit.retrieve=" + data_from_hit.length );
    System.out.println( "  ODServer.retrieve=" + data_from_server.length );
    System.out.println( "  ODFolder.retrieve=" + data_from_folder.length );
    if ( data_from_hit.length == data_from_server.length )
    {
        for ( j = 0; j < data_from_hit.length; j++ )
        {
            if ( data_from_hit[j] != data_from_server[j] )
                break;
        }
        if ( j == data_from_hit.length )
        {
            System.out.println( "ODHit vs. ODServer: Length and content of data match" );
            if ( data_from_hit.length == data_from_folder.length )
            {
                for ( j = 0; j < data_from_folder.length; j++ )
                {
                    if ( data_from_hit[j] != data_from_folder[j] )
                        break;
                }
                if ( j == data_from_folder.length )
                    System.out.println( "ODHit vs. ODFolder: Length and content of data matches" );
                else
                {
                    System.out.println( "*** ODHit vs. ODFolder: Data mismatch at offset " + j );
                    System.out.println( "  ODHit data is " + data_from_hit[j] );
                    System.out.println( "  ODFolder data is " + data_from_folder[j] );
                }
            }
            else
                System.out.println( "*** ODHit vs. ODFolder: Length mismatch" );
        }
        else
        {
            System.out.println( "*** ODHit vs. ODServer: Data mismatch at offset " + j );
            System.out.println( "  ODHit data is " + data_from_hit[j] );
            System.out.println( "  ODServer data is " + data_from_server[j] );
        }
    }
    else
        System.out.println( "*** ODHit vs. ODServer: Length mismatch" );
}
else
    System.out.println( "There is no document to retrieve" );

//-----
// Cleanup
//-----
odFolder.close( );
odServer.logoff( );
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed - analyze the result of the comparisons" );
System.out.println( "" );
System.out.println( "If the arwww.ini file specifies 'native' for the data type, all" );
System.out.println( "lengths and data should match; otherwise, differences are expected." );
System.out.println( "" );
}

catch ( ODEException e )
{
    System.out.println( "ODEException: " + e );
    System.out.println( "  id = " + e.getErrorId( ) );
    System.out.println( "  msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

```

```

    }
    catch ( Exception e2 )
    {
        System.out.println( "exception: " + e2 );
        e2.printStackTrace( );
    }
}

```

以下の例では、文書データのバルク検索に ODCallback メソッドを使用します。

```

//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcCallback extends ODCallback
{
    byte[] data_from_folder;
    boolean init = true;

    TcCallback( )
    {
    }

    public void HitHandleCallback( int hit, int off, int len )
    {
    }

    public boolean HitCallback( String docid, char type, String[] values )
        throws Exception
    {
        return true;
    }

    public boolean DataCallback( byte[] data )
    {
        byte[] temp;
        int j, k;

        //-----
        // If first data block received, initialize container; otherwise,
        // append new data to that previously received.
        //-----
        if ( init )
        {
            data_from_folder = data;
            init = false;
        }
        else
        {
            temp = new byte[ data_from_folder.length + data.length ];
            for ( j = 0; j < data_from_folder.length; j++ )
                temp[j] = data_from_folder[j];
            k = data_from_folder.length;
            for ( j = 0; j < data.length; j++ )
                temp[k++] = data[j];
            data_from_folder = temp;
        }

        return true;
    }

    public byte[] getData( )
    {
        return data_from_folder;
    }
}

```

文書の印刷

以下の例では、ODServer および ODFolder メソッドを使用して、サーバーで使用可能なプリンターをリストし、指定したサーバー・プリンターで文書を印刷します。また、この例では、ODServer メソッドを使用して、ログオンの準備、指定したフォルダーのオープン、ログオフを行います。

この例には、以下の ODServer メソッドを示します。

- initialize
- Logon
- openFolder
- getServerPrinters
- Logoff
- terminate

この例には、以下の ODFolder メソッドを示します。

- search
- printDocs
- close

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダー名
- プリンター名
- 構成ディレクトリー (arswww.ini ファイルのロケーション)
- (オプション) ローカル・サーバー・ディレクトリー

文書の印刷の例:

```
//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcPrintHit
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        ODFolder odFolder;
        ODHit odHit;
        Vector hits, hit_to_print;
        String [] printers;
        String printer_name;
        boolean match;
        int j;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 6 )
        {
            System.out.println( "usage: java TcPrintHit <server> <userid> <password> <folder> <printer> <config dir> [<local server dir>]" );
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "This testcase should:" );
            System.out.println( "  Logon to the specified server" );
            System.out.println( "  Display the list of printers available on the server" );
            System.out.println( "  Open the specified folder" );
            System.out.println( "  Search the folder using the default criteria" );
            System.out.println( "  Display the number of hits" );
            System.out.println( "  Print the first hit to the specified server printer" );
            System.out.println( "" );
            System.out.println( "-----" );
            System.out.println( "" );

            //-----
            // Logon to specified server
            //-----
            odServer = new ODServer();
            odServer.initialize( argv[5], "TcPrintHit.java" );
            System.out.println( "Logging on to " + argv[0] + "..." );
            if ( argv.length == 6 )
                odServer.logon( argv[0], argv[1], argv[2] );
            else
                odServer.logon( argv[0], argv[1], argv[2], ODConstant.CONNECT_TYPE_LOCAL, 0, argv[6] );

            //-----
            // If any server printers are available on the server
            //-----
            System.out.println( "Retrieving list of server printers..." );
        }
    }
}
```

```

printer_name = argv[4];
printers = odServer.getServerPrinters( );
if ( printers.length > 0 )
{
    //-----
    // List the available server printers
    //-----
    System.out.println( "There are " + printers.length + " printers available on the server:" );
    match = false;
    for( j = 0; j < printers.length; j++ )
    {
        System.out.println( " " + printers[j] );
        if ( printers[j].equals( printer_name ) )
            match = true;
    }

    if ( match )
    {
        //-----
        // Open the specified folder and search with the default criteria
        //-----
        System.out.println( "Opening " + argv[3] + " folder..." );
        odFolder = odServer.openFolder( argv[3] );
        System.out.println( "Searching with default criteria..." );
        hits = odFolder.search( );
        System.out.println( " Number of hits: " + hits.size( ) );

        //-----
        // Print the first hit to the specified server printer
        //-----
        if ( hits.size( ) > 0 )
        {
            hit_to_print = new Vector( );
            odHit = (ODHit)hits.elementAt( 0 );
            hit_to_print.addElement( odHit );
            System.out.println( "Printing first hit to " + printer_name + "..." );
            odFolder.printDocs( hit_to_print, printer_name );
        }
        else
            System.out.println( "There is no document to print" );

        odFolder.close( );
    }
    else
        System.out.println( "The specified printer ( " + printer_name + " ) is not available on this server" );
}
else
    System.out.println( "No printers are available on this server" );

//-----
// Cleanup
//-----
odServer.logoff( );
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed - Analyze the results" );
System.out.println( "" );
}

catch ( ODEException e )
{
    System.out.println( "ODEException: " + e );
    System.out.println( " id = " + e.getErrorId( ) );
    System.out.println( " msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

catch ( Exception e2 )
{
    System.out.println( "exception: " + e2 );
    e2.printStackTrace( );
}
}
}

```

注に関する情報のリスト表示

以下の例では、ODNote メソッドを使用して、注に関する詳細な情報をリストします。この例では、指定したサーバーにログオンし、指定したフォルダーを開いて、デフォルト基準を使用してフォルダーを検索し、ヒット数を表示して、最初の文書に関連した注の数を表示し、その文書に添付されている各注ごとに詳細情報を表示します。情報には、文書ページ上の注の位置、背景色、注が文書に添付された日時、注を作成したユーザー ID およびその他の属性が含まれています。

この例には、以下の ODNote メソッドを示します。

- getColor
- getDateTime
- getGroupName
- getOffsetX

- getOffsetY
- getPageNum
- getText
- getUserid
- isOkToCopy
- isPublic

この例には、以下の ODServer メソッドも示します。

- initialize
- Logon
- openFolder
- Logoff
- terminate

この例には、以下の ODFolder メソッドも示します。

- search
- close

この例には、以下の ODHit メソッドも示します。

- getNotes

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダー名
- 構成ディレクトリー (arswww.ini ファイルのロケーション)
- (オプション) ローカル・サーバー・ディレクトリー

注に関する情報のリスト表示の例:

```
//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcListNotes
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        ODFolder odFolder;
        ODHit odHit;
        ODNote odNote;
        Vector hits, notes;
        int j;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 5 )
        {
            System.out.println( "usage: java TcListNotes <server> <userid> <password> <folder> <config dir> [<local server dir>"];
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "This testcase should:" );
            System.out.println( " Logon to the specified server" );
            System.out.println( " Open the specified folder" );
            System.out.println( " Search the folder using the default criteria" );
            System.out.println( " Display the number of hits" );
            System.out.println( " Display the number of notes associated with the first hit" );
            System.out.println( " Display info for each note" );
            System.out.println( "" );
            System.out.println( "-----" );
            System.out.println( "" );

            //-----

```



```

// Logon to specified server
//-----
odServer = new ODServer( );
odServer.initialize( argv[4], "TcListNotes.java" );
System.out.println( "Logging on to " + argv[0] + "..." );
if ( argv.length == 5 )
    odServer.logon( argv[0], argv[1], argv[2] );
else
    odServer.logon( argv[0], argv[1], argv[2], ODConstant.CONNECT_TYPE_LOCAL, 0, argv[6] );

//-----
// Open the specified folder and search with the default criteria
//-----
System.out.println( "Opening " + argv[3] + " folder..." );
odFolder = odServer.openFolder( argv[3] );
System.out.println( "Searching with default criteria..." );
hits = odFolder.search( );
System.out.println( " Number of hits: " + hits.size( ) );

//-----
// List info for each note for the first hit
//-----
if ( hits.size( ) > 0 )
{
    odHit = (ODHit)hits.elementAt( 0 );
    notes = odHit.getNotes( );
    System.out.println( " There are " + notes.size( ) + " notes for the first hit" );
    for ( j = 0; j < notes.size( ); j++ )
    {
        odNote = (ODNote)notes.elementAt( j );
        System.out.println( " " + (j+1) + ". Text='" + odNote.getText( ) + "' " );
        System.out.println( "   UserId=" + odNote.getUserId( ) );
        System.out.println( "   Page=" + odNote.getPageNum( ) );
        System.out.println( "   Color=" + odNote.getColor( ) );
        System.out.println( "   Date=" + odNote.getDateTime( ) );
        System.out.println( "   Group=" + odNote.getGroupName( ) );
        System.out.println( "   Offset={ " + odNote.getOffsetX( ) + ", " + odNote.getOffsetY( ) + " } " );
        System.out.println( "   OkToCopy=" + odNote.isOkToCopy( ) );
        System.out.println( "   Public=" + odNote.isPublic( ) );
    }
}
else
    System.out.println( "There is no document - cannot list notes" );

//-----
// Cleanup
//-----
odFolder.close( );
odServer.logoff( );
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed - Ensure that the information" );
System.out.println( " is the same as shown by the Windows Client" );
System.out.println( "" );
}

catch ( ODEException e )
{
    System.out.println( "ODEException: " + e );
    System.out.println( " id = " + e.getErrorId( ) );
    System.out.println( " msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

catch ( Exception e2 )
{
    System.out.println( "exception: " + e2 );
    e2.printStackTrace( );
}
}
}

```

注の追加

クラス ODHit のオブジェクトは、OnDemand 文書を表します。以下の例では、ODHit メソッドを使用して、文書に関連した注の数を表示し、以下の属性を持つ新規の注を追加します。

- 指定した注テキスト
- OkToCopy=false
- Public=false (つまり、専用の注)
- 空のグループ名

この例には、以下の ODHit メソッドを示します。

- getNotes
- addNote

この例では、ODServer メソッドを使用して、ログオンの準備、指定したフォルダーのオープン、およびログオフを行います。また、ODFolder メソッドを使用して、フォルダーの検索、照会で一致したヒット数の入手、フォルダーのクローズを行います。この例には、以下の ODServer メソッドを示します。

- initialize
- Logon
- openFolder
- Logoff
- terminate

この例には、以下の ODFolder メソッドを示します。

- search
- getHits
- close

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダー名
- 注のテキスト
- 構成ディレクトリー (arswww.ini ファイルのロケーション)
- (オプション) ローカル・サーバー・ディレクトリー

注釈の追加の例:

```
//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcAddNote
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        ODFolder odFolder;
        ODHit odHit;
        ODNote odNote;
        Vector hits, notes;
        int j;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 6 )
        {
            System.out.println( "usage: java TcAddNote <server> <userid> <password> <folder> <note text> <config dir> [<local server dir>" );
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "This testcase should:" );
            System.out.println( " Logon to the specified server" );
            System.out.println( " Open the specified folder" );
            System.out.println( " Search the folder using the default criteria" );
            System.out.println( " Display the number of hits" );
            System.out.println( " Display the number of notes associated with the first hit" );
            System.out.println( " Add a new note with the these attributes" );
            System.out.println( " The specified note text" );
            System.out.println( " OKToCopy=false" );
            System.out.println( " Public=false (i.e. a private note)" );
            System.out.println( " An empty group name" );
            System.out.println( "" );
            System.out.println( "-----" );
            System.out.println( "" );

            //-----
            // Logon to specified server
            //-----
            odServer = new ODServer( );
            odServer.initialize( argv[5], "TcAddNote.java" );
            System.out.println( "Logging on to " + argv[0] + "..." );
            if ( argv.length == 6 )
                odServer.logon( argv[0], argv[1], argv[2] );
            else

```

```

odServer.logon( argv[0], argv[1], argv[2], ODConstant.CONNECT_TYPE_LOCAL, 0, argv[6] );

//-----
// Open the specified folder and search with the default criteria
//-----
System.out.println( "Opening " + argv[3] + " folder..." );
odFolder = odServer.openFolder( argv[3] );
System.out.println( "Searching with default criteria..." );
odFolder.search( );
hits = odFolder.getHits( );
System.out.println( " Number of hits: " + hits.size( ) );

//-----
// Add a new note
//-----
if ( hits.size( ) > 0 )
{
    odHit = (ODHit)hits.elementAt( 0 );
    notes = odHit.getNotes( );
    System.out.println( " There are " + notes.size( ) + " notes for the first hit" );

    odNote = new ODNote( );
    odNote.setText( argv[4] );
    odNote.setGroupName( "" );
    odNote.setOkToCopy( false );
    odNote.setPublic( false );

    System.out.println( " Adding a new note with:" );
    System.out.println( "   Text=" + odNote.getText( ) + "" );
    System.out.println( "   OkToCopy=" + odNote.isOkToCopy( ) );
    System.out.println( "   Public=" + odNote.isPublic( ) );
    System.out.println( "   Group=" + odNote.getGroupName( ) );

    odHit.addNote( odNote );
}
else
    System.out.println( "No document - cannot list notes" );

//-----
// Cleanup
//-----
odFolder.close( );
odServer.logoff( );
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed - Ensure that the new note was correctly" );
System.out.println( " added by displaying it with the Windows Client" );
System.out.println( "" );
}

catch ( ODException e )
{
    System.out.println( "ODException: " + e );
    System.out.println( "   id = " + e.getErrorId( ) );
    System.out.println( "   msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

catch ( Exception e2 )
{
    System.out.println( "exception: " + e2 );
    e2.printStackTrace( );
}
}
}

```

文書の更新

以下の例では、文書の更新方法を示します。

この例では、ODServer、ODFolder、および ODCriteria メソッドを使用して、指定したユーザー ID およびパスワードでサーバーに接続し、指定したフォルダーを開いて、2 つの検索フィールドに検索値を設定し、「日付 (Date)」検索フィールドをヌルに設定して、フォルダーを検索します。照会で一致する文書が見つかったら、ODHit メソッドを使用して、1 つ以上のデータベース値を更新します。

この例には、以下の ODServer メソッドを示します。

- initialize
- Logon
- openFolder
- Logoff
- terminate

この例には、以下の ODFolder メソッドを示します。

- getName
- getDisplayOrder
- getCriteria
- search
- closeinitialize

この例には、以下の ODCriteria メソッドを示します。

- setOperand
- setSearchValue

この例には、以下の ODHit メソッドを示します。

- getDisplayValue
- update

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- フォルダ名
- 基準名 1
- 検索値 1
- 基準名 2
- 検索値 2
- 検索値 2 を置き換える新規検索値
- 構成ディレクトリー (arswww.ini ファイルのロケーション)

文書の更新の例:

```

/*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcUpdate
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        ODFolder odFolder;
        ODCriteria odCrit;
        ODHit odHit;
        Hashtable hash;
        Vector hits;
        String[] display_crit;
        String line, crit1, crit2, value1, value2, new_value;
        int j;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 10 )
        {
            System.out.println( "usage: java TcUpdate <server> <userid> <password> <folder> <criteria1> <value1>" +
                "<criteria2> <value2> <new value2>" );
            return;
        }

        try
        {
            System.out.println( "This testcase should:" );
            System.out.println( " Logon to the specified server" );
            System.out.println( " Open the specified folder" );
            System.out.println( " Set the search values" );
            System.out.println( " Search the folder" );
            System.out.println( " For the first hit, change the value of 2nd specified criteria" );
            System.out.println( " to the new value" );
            System.out.println( "" );
            System.out.println( "Using the Windows Client, ensure that the value has been changed." );
            System.out.println( "" );
            System.out.println( "-----" );

```

```

System.out.println( "" );

//-----
// Logon to specified server
//-----
odServer = new ODServer( );
odServer.initialize( argv[9], "TcUpdate.java" );
System.out.println( "Logging on to " + argv[0] + "..." );
odServer.logon( argv[0], argv[1], argv[2] );

//-----
// Open the specified folder and set the requested criteria
//-----
crit1 = argv[4];
crit2 = argv[6];
value1 = argv[5];
value2 = argv[7];
new_value = argv[8];
System.out.println( "Opening " + argv[3] + " folder..." );
odFolder = odServer.openFolder( argv[3] );
odCrit = odFolder.getCriteria( crit1 );
odCrit.setOperand( ODConstant.OPEqual );
odCrit.setSearchValue( value1 );
odCrit = odFolder.getCriteria( crit2 );
odCrit.setOperand( ODConstant.OPEqual );
odCrit.setSearchValue( value2 );

//-----
// Search the folder
//-----
System.out.println( " Searching for " + crit1 + " = " + value1 + " and " + crit2 + " = " + value2 + "..." );
hits = odFolder.search( );

//-----
// If there was at least one hit
//-----
if ( hits != null && hits.size( ) > 0 )
{
//-----
// Display the values for the first hit
//-----
System.out.println( " For first hit:" );
line = " ";
display_crit = odFolder.getDisplayOrder( );
for( j = 0; j < display_crit.length; j++ )
line = line + display_crit[j] + " ";
System.out.println( line );
line = " ";
odHit = (ODHit)hits.elementAt( 0 );
for ( j = 0; j < display_crit.length; j++ )
line = line + odHit.getDisplayValue( display_crit[j] ) + " ";
System.out.println( line );

//-----
// Create a hash table of existing criteria/value pairs, except for criteria 2
// which will be set to the new value. Update the hit values
//-----
System.out.println( " Replacing " + crit2 + " = " + value2 + " with " + crit2 + " = " + new_value );
hash = new Hashtable( );
for ( j = 0; j < display_crit.length; j++ )
{
if ( display_crit[j].equals( crit2 ) )
hash.put( display_crit[j], new_value );
else
hash.put( display_crit[j], odHit.getDisplayValue( display_crit[j] ) );
}
odHit.update( hash );
}
else
System.out.println( "There were no hits" );

//-----
// Cleanup
//-----
odFolder.close( );
odServer.logoff( );
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed - Using the Windows Client," );
System.out.println( " ensure that the value has been changed." );
System.out.println( "" );
}

catch ( ODEException e )
{
System.out.println( "ODEException: " + e );
System.out.println( " id = " + e.getErrorId( ) );
System.out.println( " msg = " + e.getErrorMsg( ) );
e.printStackTrace( );
}

```

```

        catch ( Exception e2 )
        {
            System.out.println( "exception: " + e2 );
            e2.printStackTrace ( );
        }
    }
}

```

パスワードの変更

以下の例では、ODServer メソッド `changePassword` を使用して、指定したユーザーのパスワードを新規パスワードに変更します。また、この例では、ODServer メソッドを使用して、ログオンおよびログオフの準備を行います。

この例には、以下の ODServer メソッドを示します。

- initialize
- Logon
- changePassword
- Logoff
- terminate

この例では、以下のランタイム・パラメーターを使用します。

- サーバー名
- ユーザー ID
- パスワード
- 新規パスワード
- 構成ディレクトリー (`arswww.ini` ファイルのロケーション)
- (オプション) ローカル・サーバー・ディレクトリー

パスワードの変更の例:

```

//*****
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class TcChangePassword
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        String server, userid, original_password, new_password;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 5 )
        {
            System.out.println( "usage: java TcChangePassword <server> <userid> <password> <new password> <config dir> [<local server dir>]" );
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "This testcase should:" );
            System.out.println( " Logon to the server using the specified password" );
            System.out.println( " Change the password to the new password" );
            System.out.println( " Logoff" );
            System.out.println( " Logon to the server using the new password" );
            System.out.println( " Change the password back to the original password" );
            System.out.println( " Logoff" );
            System.out.println( "" );
            System.out.println( "If the testcase executes without exception, no further analysis" );
            System.out.println( "is required." );
            System.out.println( "" );
            System.out.println( "-----" );
            System.out.println( "" );

            //-----
            // Create the specified server
            //-----
            server = argv[0];
            userid = argv[1];
            original_password = argv[2];
            new_password = argv[3];
            odServer = new ODServer ( );
            odServer.initialize( argv[4], "TcChangePassword.java" );

```

```

//-----
// Logon to the server using the original password
//-----
System.out.println( "Logging on to " + server + " using original password..." );
if ( argv.length == 5 )
    odServer.logon( server, userid, original_password );
else
    if ( argv.length == 6 )
        odServer.logon( server, userid, original_password, ODConstant.CONNECT_TYPE_LOCAL, 0, argv[5] );

//-----
// Change to the new password and logoff
//-----
System.out.println( "Changing to new password..." );
odServer.changePassword( new_password );
System.out.println( "Logging off..." );
odServer.logoff( );

//-----
// Logon to the server using the new password
//-----
System.out.println( "Logging on to " + server + " using new password..." );
if ( argv.length == 5 )
    odServer.logon( server, userid, new_password );
else
    if ( argv.length == 6 )
        odServer.logon( server, userid, new_password, ODConstant.CONNECT_TYPE_LOCAL, 0, argv[5] );

//-----
// Change back to the original password and logoff
//-----
System.out.println( "Changing back to original password..." );
odServer.changePassword( original_password );
System.out.println( "Logging off..." );
odServer.logoff( );

//-----
// Cleanup
//-----
odServer.terminate( );
System.out.println( "" );
System.out.println( "-----" );
System.out.println( "" );
System.out.println( "Testcase completed successfully" );
System.out.println( "" );
}

catch ( ODEException e )
{
    System.out.println( "ODEException: " + e );
    System.out.println( "    id = " + e.getErrorId( ) );
    System.out.println( "    msg = " + e.getErrorMsg( ) );
    e.printStackTrace( );
}

catch ( Exception e2 )
{
    System.out.println( "exception: " + e2 );
    e2.printStackTrace( );
}
}
}

```

付録 E. AFP の HTML への変換

AFP/HTML 間変換プロセスは、AFP 文書や資源を HTML 文書に変換します。AFP/HTML 間変換プロセスには、IBM Printing Systems Division からの AFP2WEB Transform サービス・オフリングが必要です。管理者は、HTTP サーバーで AFP2WEB Transform のインストールと構成を行う必要があります。AFP2WEB Transform サービス・オフリングに関する詳細は、IBM 担当員にご相談ください。加えて、AFP2WEB Transform で処理する AFP 文書や資源用に、組織内で構成オプションを指定する必要があります。このセクションでは、構成オプションを指定する方法について説明します。

注: 本書では、AFP2HTML.INI という名前で構成ファイルを参照します。構成オプションを含むファイルを指定する方法については、24 ページの『CONFIGFILE』を参照してください。

AFP2HTML.INI ファイルは、AFP2WEB Transform の構成オプションを提供します。通常、AFP2HTML.INI ファイルでは、具体的な AFP アプリケーション用のオプションを構成しますが、デフォルト・オプションのセットを使用することもできます。AFP2WEB Transform は、AFP2HTML.INI ファイルで識別されていない AFP アプリケーションの文書や資源を変換するときに、デフォルト・オプションを使用します。オプションと変換プロセスの詳細については、AFP2WEB Transform の資料を参照してください。

以下のトピックでは、AFP2HTML.INI ファイルについての追加情報を扱います。

- AFP2HTML.INI ファイルの形式
- AFP2WEB Transform 用のオプション
- 変換された文書の表示

注: AFP2HTML アプレットで文書を変換するには、ARSWWW.INI ファイルの DEFAULT BROWSER セクション (または、他のブラウザ・セクション) で AFPVIEWING=HTML パラメーターを指定することも必要です。詳細については、36 ページの『AFPVIEWING』を参照してください。(Retrieve Document API の使用を計画している場合は、_afp=HTML パラメーターを指定してください。詳細については、87 ページの『Retrieve Document』を参照してください。) また、AFP2WEB Transform プログラムが含まれているディレクトリーを指定する必要もあります (24 ページの『CONFIGFILE』を参照)。

AFP2HTML.INI ファイルの形式

以下に、AFP2HTML.INI ファイルの例を示します。

```
[CREDIT-CREDIT]
UseApplet=FALSE
ScaleFactor=1.0
CreateGIF=TRUE
SuppressFonts=FALSE
FontMapFile=creditFontMap.cfg
ImageMapFile=creditImageMap.cfg
```

```
[default]
ScaleFactor=1.0
CreateGIF=TRUE
SuppressFonts=FALSE
FontMapFile=fontmap.cfg
ImageMapFile=imagemap.cfg
```

このファイルの構造は Windows INI ファイルに似ており、各 AFP アプリケーション用のスタンザ 1 つと、デフォルトのスタンザが含まれています。スタンザのタイトル行は、アプリケーション・グループとアプリケーションを識別します。例えば、以下のタイトル行であれば、

```
[CREDIT-CREDIT]
```

CREDIT アプリケーション・グループと CREDIT アプリケーションを識別します。タイトル行で名前を分離するには、- (ダッシュ) 文字を使用します。この名前は、OnDemand サーバーで定義されているアプリケーション・グループおよびアプリケーションの名前と一致しなければなりません。アプリケーション・グループに複数のアプリケーションがある場合、アプリケーションごとに 1 つのスタンザを作成してください。

[default] スタンザ内のオプションは、AFP2WEB Transform が AFP2HTML.INI ファイルで識別されていない AFP アプリケーション用に文書进行处理するために使用します。これらのデフォルトは、AFP アプリケーション・スタンザに何らかのオプションが含まれていない場合にも使用されます。

UseApplet オプションは、ODWEK へのディレクティブです。これは、AFP2WEB Transform からの出力を表示するために AFP2HTML アプレットを使用するかどうかを判別します。デフォルト値は TRUE です。FALSE を指定した場合 (出力を表示するために AFP2HTML アプレットを使用しない場合)、出力は Web ブラウザーによってフォーマットされ、表示されます。

残りの 5 つのオプションは、AFP2WEB Transform へのディレクティブです。145 ページの『AFP2WEB Transform 用のオプション』では、これらのオプションを AFP2WEB Transform で使用方法について簡単に説明します。

AFP2WEB Transform 用のオプション

表 14 では、AFP2WEB Transform で文書を変換するために AFP2HTML.INI ファイルで指定できるオプションをリストします。

表 14. AFP2WEB Transform 用のオプション

AFP2HTML.INI ファイル内のオプション	説明
AllObjects	ラージ・オブジェクトとして OnDemand に保管されている文書の処理方法を決定します。デフォルト値は 0 (ゼロ) で、これは ODWEK が文書の最初のセグメントのみを検索することを意味します。1 を指定した場合、ODWEK は全セグメントを検索し、それらをテキストに変換してから、文書をクライアントに送信します。 注: 非常に大きな文書に対してラージ・オブジェクト・サポートを使用可能にすると、クライアント側での文書の表示が著しく遅れることがあります。
ScaleFactor	指定したスケール因数で出力を位取りします。デフォルト値は 1.0 です。例えば、ScaleFactor=2.0 の値を指定すると、出力はデフォルト・サイズの 2 倍のスケールに位取りされます。ScaleFactor=0.5 の値を指定すると、出力はデフォルト・サイズの半分のスケールに位取りされます。デフォルト・サイズは、OnDemand アプリケーションの「Logical Views」ページの「ズーム」設定から決定されます。
SuppressFonts	AFP テキスト・ストリングを変換するかどうかを決定します。SuppressFonts=TRUE を指定すると、Font Map ファイルでリストされているフォントを使用したテキストは変換されません。デフォルト値は FALSE です。このとき、すべての AFP テキスト・ストリングは変換されます。Font Map ファイルは FontMapFile オプションによって識別されます。
FontMapFile	Font Map ファイルの絶対パス名を識別します。Font Map ファイルには、特殊な処理を必要とするフォントのリストが含まれています。デフォルトの Font Map ファイルは imagfont.cfg で、これは、AFP2WEB Transform プログラムが含まれているディレクトリーに存在します。Font Map ファイルの詳細については、AFP2WEB Transform の資料を参照してください。
ImageMapFile	イメージ・マッピング・ファイルを識別します。イメージ・マッピング・ファイルを使用すれば、イメージを出力から除去したり、陰影付きイメージの外観を改善したり、AFP2WEB Transform で作成されたイメージで既存のイメージを置換したりすることができます。複数の AFP 文書で共通するイメージ (例えば、会社のロゴ) をマッピングすれば、文書の変換に必要な時間を削減することができます。指定する場合、イメージ・マッピング・ファイルは、AFP2WEB Transform プログラムが含まれているディレクトリーに存在していなければなりません。イメージ・マッピング・ファイルの詳細については、AFP2WEB Transform の資料を参照してください。

注: ODWEK は、文書を変換するときに以下のオプションを AFP2WEB Transform に送ります。これらのオプションは、AFP2HTML.INI ファイルでは指定されません。

- 「方向」。文書を表示するときに使用する回転値を判別します。デフォルト・サイズは、OnDemand アプリケーションの「ビュー情報」ページの「方向」設定から決定されます。
- 「イメージ・カラー」。イメージとグラフィックスを表示するときに使用する色を判別します。デフォルト・サイズは、OnDemand アプリケーションの「論理ビュー」ページの「イメージ・カラー」設定から決定されます。

変換された文書の表示

AFP2HTML.INI ファイルの UseApplet オプションは、ODWEK へのディレクティブで、変換された出力を表示するために AFP2HTML アプレットを使用するかどうかを判別します。デフォルト値は TRUE です。FALSE を指定した場合 (出力を表示するために AFP2HTML アプレットを使用しない場合)、出力は Web ブラウザーによってフォーマットされ、表示されます。

IBM では一般に、変換された文書を表示するには、常に AFP2HTML アプレットを使用することをお勧めしています。文書がラージ・オブジェクトとして OnDemand に保管された場合、AFP2HTML アプレットは、ユーザーがその文書内の任意のページに容易に移動できるようにするコントロールを追加します。

付録 F. AFP の PDF への変換

AFP2PDF Transform は、AFP の文書や資源を PDF 文書に変換します。AFP2PDF Transform は、IBM Printing Systems Division からのサービス・オファリングです。管理者は、HTTP サーバーで AFP2PDF Transform のインストールと構成を行う必要があります。AFP2PDF Transform サービス・オファリングの詳細については、IBM 担当員にご相談ください。加えて、AFP2PDF Transform で処理する AFP 文書や資源用に、組織内で構成オプションを指定する必要があります。このセクションでは、構成オプションを指定する方法について説明します。

注: 本書では、AFP2PDF.INI という名前で構成ファイルを参照します。構成オプションを含むファイルを指定する方法については、26 ページの『CONFIGFILE』を参照してください。

AFP2PDF.INI ファイルは、AFP2PDF Transform の構成オプションを提供します。通常、AFP2PDF.INI ファイルでは、具体的な AFP アプリケーション用のオプションを構成しますが、デフォルト・オプションのセットを使用することもできます。AFP2PDF Transform は、AFP2PDF.INI ファイルで識別されていない AFP アプリケーションの文書や資源を変換するときに、デフォルト・オプションを使用します。オプションと変換プロセスの詳細については、AFP2PDF Transform の資料を参照してください。

以下のトピックでは、AFP2PDF.INI ファイルについての追加情報を扱います。

- AFP2PDF.INI ファイルの指定
- 変換された文書の表示

注: 文書を変換するには、ARSWWW.INI ファイルの DEFAULT BROWSER セクション (または、他のブラウザ・セクション) で AFPVIEWING=PDF パラメーターを指定することも必要です。詳細については、36 ページの『AFPVIEWING』を参照してください。(Retrieve Document API の使用を計画している場合は、_afp=PDF パラメーターを指定してください。詳細については、87 ページの『Retrieve Document』を参照してください。)

AFP2PDF.INI ファイルの指定

以下に、AFP2PDF.INI ファイルの例を示します。

```
[CREDIT-CREDIT]
OptionsFile=
ImageMapFile=creditImageMap.cfg

[default]
OptionsFile=
ImageMapFile=imagemap.cfg
AllObjects=0
```

このファイルの構造は Windows INI ファイルに似ており、各 AFP アプリケーション用のスタンザ 1 つと、デフォルトのスタンザが含まれています。スタンザのタイトル行は、アプリケーション・グループとアプリケーションを識別します。例えば、以下のタイトル行であれば、

```
[CREDIT-CREDIT]
```

CREDIT アプリケーション・グループと CREDIT アプリケーションを識別します。タイトル行で名前を分離するには、- (ダッシュ) 文字を使用します。この名前は、OnDemand サーバーで定義されているアプリケーション・グループおよびアプリケーションの名前と一致しなければなりません。アプリケーション・グループに複数のアプリケーションがある場合、アプリケーションごとに 1 つのスタンザを作成してください。

[default] スタンザで指定したパラメーターは、AFP2PDF Transform が AFP2PDF.INI ファイルで識別されていない AFP アプリケーション用に文書を処理するために使用します。これらのデフォルト・パラメーターは、AFP アプリケーション・スタンザで何らかのパラメーターが指定されていない場合にも使用されます。

OptionsFile パラメーターは、AFP2PDF Transform で使用される変換オプションが含まれているファイルの絶対パス名を識別します。これらの変換オプションは、特殊な処理を必要とする AFP 文書で使用されます。変換オプションのファイルの詳細については、AFP2PDF Transform の資料を参照してください。

ImageMapFile パラメーターは、イメージ・マッピング・ファイルを識別します。イメージ・マッピング・ファイルを使用すれば、イメージを出力から除去したり、陰影付きイメージの外観を改善したり、AFP2PDF Transform で作成されたイメージで既存のイメージを置換したりすることができます。ほとんどの AFP 文書で共通するイメージ (例えば、会社のロゴ) をマッピングすれば、文書の変換に必要な時間を削減することができます。指定する場合、イメージ・マッピング・ファイルは、AFP2PDF Transform プログラムが含まれているディレクトリーに存在していなければなりません。AFP2PDF Transform 用のプログラムが含まれているディレクトリーを指定する方法については、26 ページの『INSTALLDIR』を参照してください。イメージ・マッピング・ファイルの詳細については、AFP2PDF Transform の資料を参照してください。

AllObjects パラメーターは、ラージ・オブジェクトとして OnDemand に保管されている文書を ODWEK が処理する方法を判別します。デフォルト値は 0 (ゼロ) で、これは ODWEK が文書の最初のセグメントのみを検索することを意味します。1 を指定した場合、ODWEK は全セグメントを検索し、それらをテキストに変換してから、文書をクライアントに送信します。**注:** 非常に大きな文書に対してラージ・オブジェクト・サポートを使用可能にすると、クライアント側での文書の表示が著しく遅れることがあります。

変換された文書の表示

変換された文書を Adobe Acrobat ビューアーで表示するには、ご使用のブラウザーに合ったプラグインを入手する必要があります。

付録 G. HTTP サーバー構成ファイル

このセクションでは、次の HTTP サーバーの構成ファイルの例を示します。

- オリジナルの HTTP サーバー (現在、サポートされていません)
- HTTP Apache サーバー
- WebSphere Application Server

注: 構成アシスタンスについては、HTTP 文書を調べてください。

HTTP Apache サーバー

注:

1. Listen の行のストリング **n.n.n.n** を、OnDemand サーバー・ソフトウェアが稼働する iSeries マシンの IP アドレスで置き換えてください。
2. Listen の行の **p** を、クライアントと HTTP サーバーの通信に使用される TCP/IP ポート番号で置き換えてください。デフォルトの TCP/IP ポート番号は 80 です。
3. DefaultNetCCSID 行のストリング **xxxx** を、OnDemand インスタンスの CCSID で置き換えてください。
4. ServerName の行のストリング **f.q.h.n** を、OnDemand サーバー・ソフトウェアが稼働する iSeries マシンの完全修飾 TCP/IP ホスト名で置き換えてください。
5. DBCS 言語の場合、CGIConvMode 行を次のように変更してください。

```
CGIConvMode %%EBCDIC_JCD/MIXED%%
```

```
#HTTP Apache Server Configuration for the ODWAPACHE server
```

```
Listen n.n.n.n:p
DefaultNetCCSID xxxxx
HostNameLookups off
RuleCaseSense OFF
LimitRequestBody 102400
ErrorLog /www/odwapache/logs/error_log
ServerName f.q.h.n
DocumentRoot /www/odwapache
UseCanonicalName Off
CGIConvMode %%EBCDIC/MIXED%%
ScriptLogLength 200
ScriptLog /www/odwapache/logs/cgi_log
Alias /logon /www/odwapache/odw_logon.htm
AliasMatch ^/images/(.*)$ /QIBM/UserData/OnDemand/www/images/$1
AliasMatch ^/applets/com/ibm/edms/od/(.*)$ /QIBM/ProdData/OnDemand/www/applets/$1
AliasMatch ^/applets/(.*)$ /QIBM/ProdData/OnDemand/www/applets/$1
ScriptAliasMatch ^/scripts/arswww.cgi$ /QSYS.LIB/QRDARS.LIB/ARS3WCGI.PGM
ScriptAliasMatch ^/scripts/arswww%.cgi/(.*)$ /QSYS.LIB/QRDARS.LIB/ARS3WCGI.PGM
AddType www/unknown cab
AddType www/unknown jar
AlwaysDirectoryIndex On
DirectoryIndex index.html
<Directory/>
    Options None
    Options +ExecCGI
    AllowOverride None
    Order deny,allow
    Deny from all
```

```
<LimitExcept GET HEAD OPTIONS TRACE POST PUT>
</LimitExcept>
</Directory>
<Directory /QIBM/ProdData/OnDemand/www/applets/>
  Order allow,deny
  Allow from All
</Directory>
<Directory /QIBM/ProdData/OnDemand/www/images/>
  Order allow,deny
  Allow from All
  Options +FollowSymLinks
</Directory>
<Directory /QIBM/ProdData/OnDemand/www/samples/>
  Order allow,deny
  Allow from All
  Options +FollowSymLinks
</Directory>
<Directory /www/odwapache/>
  Order allow,deny
  Allow from All
</Directory>
<Directory /QSYS.LIB/QRDARS.LIB>
  Order allow,deny
  Allow from All
  Options +ExecCGI
</Directory>
```

| WebSphere Application Server

| WebSphere 構成ファイルのサンプルは、IBM Content Manager OnDemand for iSeries
| サポート Web ページ (<http://www.ibm.com/software/data/ondemand/400/support.html>)
| から入手できます。 Technotes カテゴリーで、*ODWEK WebSphere example* を検索
| してください。

付録 H. 非 HTML 出力

ODWEK は `_nohtml` ディレクティブを使用して、関数 (Logon など) が生成する出力のタイプを判別します。デフォルトでは、ODWEK は HTML 出力を生成します。`_nohtml=1` を指定した場合、ODWEK は区切り文字付き ASCII 出力を生成します。この章では、ODWEK が生成する区切り文字付き ASCII 出力について説明します。

区切り文字付き ASCII 出力

ODWEK が生成する区切り文字付き ASCII 出力は、文字ストリング値、キーワード、関数区切り文字、レコード区切り文字、ストリング区切り文字、およびセパレーターを含む出力レコードのセットです。

- 文字ストリング値は、キーワード、区切り文字、およびセパレーター以外の、関数の出力データです。例えば、次に呼び出す関数、フォルダーの名前、フォルダー・フィールド名、検索演算子、およびフィールド値が、文字ストリング値です。
- キーワードは特定の文字ストリングから成っています。例えば、ACTION、DOC、FOLDER、NUMROWS、および ROW がキーワードです。
- 関数区切り文字は、特定の文字ストリング [BEGIN] および [END] から成っています。
- レコード区切り文字は改行文字 `¥n` です。すべてのレコードが改行文字で区切られます。
- デフォルトでは、ストリング区切り文字およびセパレーターは、脱字記号文字 (^)、左大括弧文字 ([)、右大括弧文字 (]) です。次に例を示します。

```
[folderName^folderDesc]
```

キーワード・レコードに複数の文字ストリング値が含まれている場合、値は脱字記号文字で区切られます。文字ストリング値の各キーワードのセットは、左大括弧文字と右大括弧文字で区切られます。

複数の文字ストリング値を脱字記号文字で区切り、左大括弧文字と右大括弧文字で囲んで、リストに保管することができます。例えば、フィールドでの有効な検索演算子のリストが、以下のように表示される場合があります。

```
[1^2^4^8^16^32]
```

ストリング区切り文字およびセパレーターのデフォルト文字を指定変更することができます。詳細については、33 ページの『[NO HTML]』を参照してください。

- 単一のヌル文字ストリング値は、内側に値がない 2 つの二重引用符文字 ("") によって示されます。ヌル・リストは、内側に値がない左大括弧文字と右大括弧文字 ([]) によって示されます。

Logon

以下に、Logon 関数によって生成される区切り文字付き ASCII 出力の例を示します。

```
[BEGIN]¥n
ACTION=searchCriteriaUrl¥n
FOLDER=[folderName^folderDesc]¥n
FOLDER=[folderName^folderDesc]¥n

:

[END]¥n
```

注

1. スtring searchCriteriaUrl は、次に実行される関数の名前およびそのパラメーターを識別します。
2. スtring folderName はフォルダー名を識別します。この名前には引用符は付きません。
3. スtring folderDesc はフォルダーの説明です。この説明には引用符は付きません。

Search Criteria

以下に、Search Criteria 関数によって生成される区切り文字付き ASCII データの例を示します。

```
[BEGIN]¥n
ACTION=hitListUrl¥n
DISPLAY_ORDER=[field1^field2^...fieldN]¥n
NUMROWS=numberOfRows¥n
ROW=[criteriaName^[validOp]^defOp]^ [inpType^inpAssocData]¥n

:

[END]¥n
```

注

1. スtring hitListUrl は、次に実行される関数の名前およびそのパラメーターを識別します。
2. DISPLAY_ORDER キーワードは、フォルダー・フィールドが表示される順序を示します。
3. スtring numberOfRows は、続く ROW キーワード・レコードの数を識別します。この関数は、検索フィールドごとに 1 つずつ ROW キーワード・レコードを生成します。
4. スtring criteriaName は、検索フィールドの検索基準を表します。検索基準には引用符は付きません。
5. スtring validOp は、以下のような、検索フィールドの有効な検索演算子を表す整数値のリストです。

1 Equal

- 2 Not Equal
- 4 Less Than
- 8 Less Than or Equal
- 16 Greater Than
- 32 Greater Than or Equal
- 64 In
- 128 Not In
- 256 Like
- 512 Not Like
- 1024 Between
- 2048 Not Between

6. ストリング defOp は、デフォルトの検索演算子を表す整数値です。
7. ストリング inpType は、以下のように、検索フィールドのタイプを表します。
 - A Annotation Text Search
 - C Choice
 - N Normal
 - S Segment
 - T Text Search
 - Z Annotation Color Search
8. ストリング inpAssocData は、 defOp および inpType に関連するリストです。
表 15 を参照してください。

表 15. inpAssocData に関連したデフォルト演算子および入力タイプ

defOp	inpType	inpAssocData
Between, Not Between	N	Null: [] またはリスト : [defaultField1^...^defaultFieldM] 例: ["01/31/96"^"01/31/97"] ["01/31/96"^" [""^"01/31/97"]
他の有効な演算子	A, N, T, Z	Null: [] または、デフォルトのフィールド値を表す単一 ストリング値。
他の有効な演算子	C, S	[[listOfChoices]^defaultChoice] 例: [["JFIF"^"TIFF"^"PCX"]^"TIFF"] [["JFIF"^"TIFF"^"PCX"]^" [""]

Document Hit List

以下に、 Document Hit List 関数によって生成される区切り文字付き ASCII 出力の例を示します。

```
[BEGIN]¶n
ACTION=hitListURL¶n
MSG=Only 20 documents can be listed for this folder.
DOC=[criteria1^criteria2^criteriaN^docid^fileType^docLocation]¶n
```

⋮
[END]¥n

注

1. スtring hitListURL は、次に実行される関数の名前およびその関数へのパラメーターを識別します。
2. MSG キーワードは、区切り文字付きの ASCII 出力内のエラー・メッセージの例を示します。デフォルトでは、ODWEK はクライアントにエラー・メッセージを送ります。ただし、関数に _nohtml=1 ディレクティブが含まれている場合、ODWEK は代わりに、区切り文字付きの ASCII 出力内にメッセージ・テキストを生成します。
3. スtring criteria1、criteria2、および criteriaM は、検索基準値を表します。値は、文書リストに表示されている順番でリストされます。これらの値に引用符は付きません。
4. スtring docid は文書の文書 ID です。
5. スtring fileType は、以下のように、文書のデータ・タイプを識別します。

A	AFP
B	BMP
E	電子メール
F	JFIF
G	GIF
L	行
N	なし
O	OD 定義
P	PDF
T	TIFF
U	ユーザー定義
X	PCX
6. スtring docLocation は、文書の保管場所を識別します。

0	不明
1	OnDemand キャッシュ記憶機構
2	アーカイブ・ストレージ
3	外部キャッシュ記憶機構

View Annotations

以下に、View Annotations 関数によって生成される区切り文字付きの ASCII 出力の例を示します。

```
[BEGIN]¥n
NOTE 4: 15:42:44 PM Mountain Standard Time Thursday November 19, 1998...¥n
Public - Cannot be copied to another server¥n
Test note from the OnDemand Internet Client.¥n
[END]¥n
```

エラー・メッセージ

以下に、エラーが発生した場合に生成される、区切り文字付き ASCII 出力の例を示します。

```
[ERROR]¥n
ID=nnnn¥n
MSG=errorMessageText¥n
```

注

1. スtring `nnnn` はエラー・メッセージの番号です。
2. スtring `errorMessageText` はエラー・メッセージのテキストです。

付録 I. 各国語サポート

このセクションには、管理者が DBCS 言語対応の ODWEK を構成する際に役立つ情報を記載します。

ARSWWW.INI ファイルの CODEPAGE および LANGUAGE パラメーターを使用して、各国語 (NL) 構成オプションを指定します。

CODEPAGE パラメーターは、ODWEK サーバーのコード・ページを識別します。また、OnDemand ライブラリー・サーバー上の OnDemand データベースとの互換性が必要です。CODEPAGE パラメーターは、ODWEK アプリケーションを実行しているワークステーションのコード・ページが、OnDemand ライブラリー・サーバー上の OnDemand データベースのコード・ページと異なる場合にのみ、指定する必要があります。システムは、ODWEK アプリケーションが稼働しているワークステーションのコード・ページをデフォルト値として使用します。

LANGUAGE パラメーターは、ODWEK がメッセージを表示する際に使用するメッセージ・カタログを判別します。

表 16 に、OnDemand によりサポートされる DBCS コード・ページと言語をリストします。CODEPAGE= 欄にはコード・ページの値がリストされていますが、これは、ODWEK アプリケーションを実行しているワークステーションのコード・ページが OnDemand データベースのコード・ページと異なる場合にのみ、指定する必要があります。LANGUAGE= 欄には、変換されたメッセージ・カタログに関連した値がリストされています。

注: Linux は、ODWEK アプリケーションのターゲット・サーバーになれないので、表にリストされていません。つまり、ODWEK は Linux システムで稼働できますが、OnDemand データベース (OnDemand ライブラリー・サーバー) は Linux システムで稼働できません。

表 16. DBCS コード・ページ、言語、コード・セット、およびロケール

地域	LANGUAGE=	OS	データベース・コード・ページ	CODEPAGE=	コード・セット	ロケール
中国 (PRC)	CHS	AIX	1383	1383	IBM_eucCN	zh_CN
		HP-UX	1383	1383	hp15CN	zh_CN. hp15CN
		Solaris	1383	1383	gb2312	zh
		Windows	1386	1386	GBK	—
		z/OS または OS/390 (EBCDIC)	935	935	IBM-935	—

表 16. DBCS コード・ページ、言語、コード・セット、およびロケール (続き)

地域	LANGUAGE=	OS	データベース・ コード・ページ	CODEPAGE=	コード・セット	ロケール
日本	JPN	AIX	954	954	IBM_eucJP	ja_JP
		HP-UX	954	954	eucJP	ja_JP.eucJP
		Solaris	954	954	eucJP	ja
		Windows	943	943	IBM-943	—
		z/OS または OS/390 (EBCDIC)	939	939	IBM-939	—
韓国	KOR	AIX	970	970	IBM_eucKR	ko_KR
		HP-UX	970	970	eucKR	ko_KR.eucKR
		Solaris	970	970	5601	ko
		Windows	1363	1363	1363	—
		z/OS または OS/390 (EBCDIC)	933	933	IBM-933	—
台湾	CHT	AIX	964	964	IBM_eucTW	zh_TW
		HP-UX	964	964	eucTW	zh_TW.eucTW
		Solaris	964	964	cns11643	zh_TW
		Windows	950	950	big5	—
		z/OS または OS/390 (EBCDIC)	937	937	IBM-937	—

DBCS 言語対応の OnDemand システムの構成については、「計画とインストール」の『各国語サポート』を参照してください。

付録 J. 問題判別ツール

表 17 でリストされているツールを使用して、システムおよび文書に関する情報を収集することができます。それらの情報を使用すると、ODWEK を構成する際の問題の解決に役立ち、さらに組織内のほかの人がアプレットおよびプラグインを使用した際に問題に直面した場合にも役立ちます。

表 17. 問題判別ツール

ツール	目的	使用可能にする方法
HTML Output	ODWEK がブラウザーに戻す HTML のコピーを保管します。	ブラウザーの「ファイル」メニューから「名前を付けて保存」を選択します。
Server Log Files	アクセス情報、エラー、およびサーバー情報を保管します。	<p>以下の処置を行います。</p> <ol style="list-style-type: none"> 1. ARSWWW.INI ファイルの DEBUG セクションで、LOG パラメーターを 1 に設定します。ODWEK が生成するログ・ファイルは、ARSWWW.LOG という名前が付けられて、LOGDIR パラメーターで指定したディレクトリーに書き込まれます。(デフォルトのディレクトリーは /QIBM/UserData/OnDemand/WWW/LOG です) 重要: 指定する場合には、DEBUG セクションが ARSWWW.INI ファイルで最初の実行可能ステートメントでなければなりません。 2. 使用している HTTP サーバーのロギングを構成します。(各 HTTP サーバーによって、ロギングの構成方法が異なる場合があります。また、詳細の度合いが異なる情報を収集するために使用可能にできるログおよびオプションが異なる場合もあります。) <p>注: ログ・ファイルには膨大な量の情報が書き込まれる場合があるので、IBM では、ロギングは必要な場合 (問題を再現する場合など) にのみ使用可能にするようにお勧めしています。長期間ロギングを使用可能にする必要がある場合、ログ・ファイルのパスでは、必ず十分なフリー・スペースがあるストレージ・デバイスを指定するようにしてください。忘れずに、サーバーから古いログ・ファイルを定期的に削除してください。</p>

表 17. 問題判別ツール (続き)

ツール	目的	使用可能にする方法
Java Console	<p>アプレットが生成するメッセージを表示します。</p>	<ul style="list-style-type: none"> • Netscape の場合: 「Communicator」メニューから「ツール」を選択し、「Java コンソール」を選択します。 • Internet Explorer の場合: <ol style="list-style-type: none"> 1. 「ツール」メニューから「インターネット オプション」を選択します。 2. 「詳細設定」ページから「Java コンソールの使用 (再起動が必要)」を選択します。 3. ブラウザーを再起動します。 4. 「ツール」メニューから「Java コンソール」を選択します。
AFP Web Viewer Trace Facility	<p>AFP Web Viewer で表示される AFP 文書に関する詳細な情報を取り込みます。</p>	<p>ユーザーのワークステーション上の FLDPORT2.INI ファイルに、以下のセクションが存在するかどうか確認してください。</p> <pre>[Misc] ViewTraceFile=d:%temp%afpplgin.log Trace=TRUE</pre> <p>ログ・ファイルのパスを検査します。必要な情報を収集したら、ロギングを必ずオフにしてください。</p>
OnDemand System Log	<p>文書および注釈と関係があるシステム・メッセージ (ログオンやログオフなど) およびアプリケーション・グループ・メッセージ (照会や検索など) を保管します。</p>	<p>以下の処置を行います。</p> <ol style="list-style-type: none"> 1. OnDemand サーバーのシステムおよびアプリケーション・グループ・ロギングを使用可能にします。管理クライアントを使用して、サーバーのシステム・パラメーターを更新します。 2. ログに記録する特定のアプリケーション・グループ・メッセージを使用可能にします。管理クライアントを使用して、そのアプリケーション・グループのメッセージ・ロギング・オプションを更新します。

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あり、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。使用許諾については、下記の宛先に書面にてご照会ください。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

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。

IBM Corporation
Software Interoperability Coordinator
3605 Highway 52 N
Rochester, MN 55901-7829
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。

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認できません。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。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って、IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

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。

商標

IBM、Advanced Function Presentation、AFP、AS/400、iSeries、Operating System/400、OS/390、OS/400、Redbooks、WebSphere および z/OS は、IBM Corporation の米国およびその他の国における商標です。

Lotus は、IBM Corporation の商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. 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、Windows NT 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。

他の会社名、製品名およびサービス名などはそれぞれ各社の商標または登録商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アプリケーション名

Java API 109

アプリケーション・プログラミング・インターフェース (API)

参照: API

アプレット 16

インストール 51

使用可能 26

説明 1

ディレクトリー 26

要件 52

ラージ・オブジェクトのサポート 24, 146

APPLETCACHEDIR パラメーター 16

暗号化 37, 38

一時記憶域 22

一時的作業ディレクトリー 22

イメージ・ディレクトリー 20

印刷

サーバー 40, 41

Java API 131

インストール

アプレット 51

カスタマイズ 54

プラグイン 51

ユーザー定義ファイル 54

ユーザー・ワークステーション 51

要件 11

AFP Web Viewer 51

AFP2HTML Java アプレット 51

ARSWWW.INI ファイル 13

HTTP サーバー 11

Image Web Viewer 51

Java アプレット 51

Line Data Java アプレット 51

ODWEK ソフトウェア 11

OS/400 12

エラー 106, 159

エラー・メッセージ

区切り文字付き ASCII 出力 155

オペランド、Java API 113

[カ行]

解説

サーブレット 99

解説 (続き)

API 65, 99, 101

CGI API 65

Java API 101

Java サブレット 99

概要 1

各国語サポート 157

環境変数、Java API 105

関数

Add Annotation 6

Annotations 6, 7

Change Password 6

Document Hit List 6

Logoff 7

Logon 7

Print Document 7

Retrieve Document 7

Search Criteria 7

Server Print Document 7

Update Document 7

View Annotations 7

キャッシュ記憶機構 17, 18, 19

キャッシュ文書 17

キャッシュ・サイズ 18

キャッシュ・ディレクトリー 17

行データ文書

媒体タイプ 30

表示 38

変換 38

MIME コンテンツ・タイプ 30

行データ・ビューアー

構成 60

ODApplet.jre.path.IE パラメーター 40

ODApplet.jre.path.NN パラメーター 40

ODApplet.jre.version パラメーター 40

ODApplet.version パラメーター 40

許可 11

区切り文字 33

区切り文字付き ASCII 出力

エラー・メッセージ 155

区切り文字 33

形式 151

メッセージ 155

Annotations 154

Document Hit List 153

Logon 152

OnDemand が生成する 151

Search Criteria 152

View Annotations 154

クラス 103

権限 11

言語 21, 157
検索、フォルダーの 113, 117, 119, 122, 126
検索、文書の 128
検索値、Java API 113, 117
コード・ページ 19, 157
構成

AFP2HTML 構成ファイル 143
AFP2PDF 構成ファイル 147
ARSWWW.INI ファイル 13
HTTP サーバー 11
ODWEK ソフトウェア 11

[サ行]

サーバー
Java API 109
サーバー印刷
関数の説明 7
関数呼び出しのサンプル 86
使用可能 40, 41
パラメーター 82
API 82
Java API 131
サーバー・アクセス・リスト 23
サーバー・セキュリティ 8, 23
サーブレット
解説 99
最大ヒット 39
サンプル・アプリケーション 47
サンプル・コード 103
システム環境、Java API 105
出力区切り文字 33
準備、OnDemand Internet Connection を使用するための 1
照会結果 39
診断情報 106
セキュリティ 8, 23, 37, 38
接続、サーバーへの 108, 109
接続タイプ
Java API 109
切断、サーバーからの 109
設定、パスワードの 140
説明、OnDemand Internet Connection の 1

[タ行]

タイムアウト 23
注 35, 41
通信プロトコル 14, 15
データ・セキュリティ 8
ディレクトリー許可 11
デバッグ・セクション 43
デフォルト・ブラウザ・オプション 34
添付ファイル 32, 33
テンプレート・ファイル 49
取り消し、検索の 119

取り出し
文書 36
トレース、問題の 159
トレースおよび診断情報 106

[ハ行]

パスワード
Java API 109, 140
パッケージ階層、Java 103
パラメーター
ADDEXTENSION 34
ADDFIELDSTODOCID 35
ADDNOTES 35
AFP2HTML セクション 24
AFP2PDF セクション 25
AFPVIEWING 36
APPLETCACHEDIR 16
APPLETDIR 16
ATTACHMENT IMAGES セクション 32
AUTODOCRETRIEVAL 36
BEGIN 33
CACHEDIR 17
CACHEDOCS 17
CACHEMAXTHRESHOLD 18
CACHEMINTHRESHOLD 18
CACHESIZE 18
CACHEUSERIDS 19
CODEPAGE 19
CONFIGFILE 24, 26
CONFIGURATION セクション 15
DOCSIZE 20
EMAILVIEWING 37
ENCRYPTCOOKIES 37
ENCRYPTURL 38
END 34
FOLDERDESC 38
HOST 15
IMAGEDIR 20
INSTALLDIR 25, 26
LANGUAGE 21
LINEVIEWING 38
LOG 43, 159
LOGDIR 43
MAXHITS 39
NOLINKS 39
ODApplet.jre.path.IE 40
ODApplet.jre.path.NN 40
ODApplet.version 40
PORT 14, 15
PROTOCOL 14, 15
REPORTSERVERTIMEOUT 23
SECURITY セクション 23
SEPARATOR 34
SERVERACCESS 23
SERVERPRINT 40

パラメーター (続き)
SERVERPRINTERS 41
SHOWDOCLOCATION 41
ShowSearchString 21
TEMPDIR 22
TEMPLATEDIR 22
USEEXECUTABLE 25, 27
VIEWNOTES 41
@SRV@_DEFAULT セクション 13
@SRV@_server セクション 14
非 HTML 出力 33, 151
非アクティブ・タイムアウト 23
表示、文書の場所の 41
表示値、Java API 113
フォルダー、Java API を使用したアプリケーション・グループ
のリスト表示 111
フォルダー、Java API を使用した検索 113, 117, 119, 122,
126
フォルダー記述、Java API 125
フォルダー内のアプリケーション・グループ
Java API 111
フォルダー名、Java API 125
フォント
マッピング 58
AFP 58
TrueType 58
ブラウザ
サポートされている 52
Cookies 52
Java 仮想マシン 52
JVM 52
ブラウザ・オプション
browser セクション 42
DEFAULT BROWSER セクション 34
プラグイン
インストール 51
説明 1
プログラミング・ガイド
API 103
Java API 103
プロトコル 14, 15
文書
キャッシュ記憶機構 17
行データ 38
取り出し 36
媒体タイプ 27
表示 36, 37, 38
変換 36, 37, 38
リンク 39
AFP 36
EMAIL 37
Java API を使用した印刷 131
Java API を使用した更新 137
MIME コンテンツ・タイプ 27
文書、Java API 126, 128
文書タイプ、Java API 113

文書の場所 41
ヘルプ 159
変更、パスワードの 140
ポート
Java API 109
ポート番号 14, 15
ホスト名 15
本書について 1

[マ行]

マッピング、AFP フォントの 58
メソッド属性、フォーム・タグの 8
メッセージ 21
区切り文字付き ASCII 出力 155
メディア・タイプ/サブタイプ 27
問題判別 159

[ヤ行]

ユーザー ID
キャッシュ記憶機構 19
Java API 109
ユーザー定義ファイル
インストール 54
要件
キャッシュ記憶機構 11
サーバー 11
文書キャッシュ 11
AFP2HTML Java アプレット 12
AFP2PDF Transform 12
AFP2WEB Transform 12
Cookies 52
HTTP サーバー 11
Java API 1
Java 仮想マシン 52

[ラ行]

ラージ・オブジェクト 24, 146
リンク 39
例 103
例外処理 107
ローカル・ディレクトリー
Java API 109
ロギング 43, 159
ログ・ファイル 43, 159

[数字]

2 バイト文字セット言語 157

A

Add Annotation

- 関数の説明 6
- 関数呼び出しのサンプル 68
- パラメーター 66
- API 66

ADDEXTENSION パラメーター 34

ADDFIELDSTODOCID パラメーター 35

ADDNOTES パラメーター 35

AFP Web Viewer

- インストール 51
- インストールのカスタマイズ 54
- 構成 59
- 説明 1
- フォント 58
- マッピング、AFP フォントの 58
- ユーザー定義ファイル 54
- ユーザー定義ファイルのインストール 54
- 要件 52
- AFP フォント 58

AFP フォント

- マッピング 58

AFP 文書

- 媒体タイプ 29
- 表示 36
- 変換 36
- MIME コンテンツ・タイプ 29

AFP2HTML Java アプレット

- インストール 51
- 説明 1, 5
- 要件 12, 52
- ラージ・オブジェクトのサポート 24, 146
- APPLETCACHEDIR パラメーター 16

AFP2HTML 構成ファイル 143

AFP2HTML セクション 24

AFP2PDF Java アプレット

- 使用可能 26
- ディレクトリー 26

AFP2PDF Transform

- インストール 12
- 構成 147
- 使用可能 25

AFP2PDF 構成ファイル 147

AFP2PDF セクション 25

AFP2WEB Transform

- インストール 12
- 構成 143
- 使用可能 24

AFPVIEWING パラメーター 36

Annotations 35, 41

- 関数の説明 6, 7
- 関数呼び出しのサンプル 68, 98
- 区切り文字付き ASCII 出力 154
- パラメーター 66, 97
- API 66, 97

Annotations (続き)

- Java API 133, 135

Apache HTTP server

- 構成ファイル 149

API

- 解説 65, 99, 101
- 環境変数 105
- クラス 103
- サーバー印刷 82
- サンプル・コード 103
- システム環境 105
- 診断情報 106
- トレースおよび診断情報 106
- パッケージ化 103
- プログラミング・ガイド 103
- 例 103
- 例外処理 107

Add Annotation 66

Annotations 66, 97

CGI API リファレンス 65

Change Password 69

Document Hit List 72

Java API プログラミング・ガイド 103

Java API リファレンス 101

Logoff 77

Logon 79

Print Document 82

Retrieve Document 87

Search Criteria 92

Update Document 95

View Annotations 97

APPLETCACHEDIR パラメーター 16

APPLETDIR パラメーター 16

ARSWWW.INI ファイル

構成 13

指定 13

デバッグ・セクション 43

ブラウザ・オプション 42

ADDEXTENSION パラメーター 34

ADDFIELDSTODOCID パラメーター 35

ADDNOTES パラメーター 35

AFP2HTML セクション 24

AFP2PDF Transform 25

AFP2PDF セクション 25

AFP2WEB Transform 24

AFPVIEWING パラメーター 36

APPLET パラメーター 16

APPLETCACHEDIR パラメーター 16

ATTACHMENT IMAGES セクション 32

AUTODOCRETRIEVAL パラメーター 36

BEGIN パラメーター 33

browser セクション 42

CACHEDIR パラメーター 17

CACHEDOCS パラメーター 17

CACHEMAXTHRESHOLD パラメーター 18

CACHEMINTHRESHOLD パラメーター 18

ARSWWW.INI ファイル (続き)

CACHESIZE パラメーター 18
CACHEUSERIDS パラメーター 19
CODEPAGE パラメーター 19
CONFIGFILE パラメーター 24, 26
CONFIGURATION セクション 15
DEFAULT BROWSER セクション 34
DOCSIZE パラメーター 20
EMAILVIEWING パラメーター 37
ENCRYPTCOOKIES パラメーター 37
ENCRYPTURL パラメーター 38
END パラメーター 34
FOLDERDESC パラメーター 38
HOST パラメーター 15
IMAGEDIR パラメーター 20
INSTALLDIR パラメーター 25, 26
LANGUAGE パラメーター 21
LINEVIEWING パラメーター 38
LOG パラメーター 43, 159
LOGDIR パラメーター 43
MAXHITS パラメーター 39
MIMETYPES セクション 27
NOHTML セクション 33
NOLINKS パラメーター 39
ODApplet.jre.path.IE パラメーター 40
ODApplet.jre.path.NN パラメーター 40
ODApplet.jre.version パラメーター 40
ODApplet.version パラメーター 40
PORT パラメーター 14, 15
PROTOCOL パラメーター 14, 15
REPORTSERVERTIMEOUT パラメーター 23
SECURITY セクション 23
SEPARATOR パラメーター 34
SERVERACCESS パラメーター 23
SERVERPRINT パラメーター 40
SERVERPRINTERS パラメーター 41
SHOWDOCLOCATION パラメーター 41
ShowSearchString パラメーター 21
TEMPDIR パラメーター 22
TEMPLATEDIR パラメーター 22
USEEXECUTABLE パラメーター 25, 27
VIEWNOTES パラメーター 41
@SRV@_DEFAULT セクション 13
@SRV@_server セクション 14

ASCII 出力

エラー・メッセージ 155
形式 151
メッセージ 155
Annotations 154
Document Hit List 153
Logon 152
OnDemand が生成する 151
Search Criteria 152
View Annotations 154

ATTACHMENT IMAGES セクション 32
AUTODOCRETRIEVAL パラメーター 36

B

BEGIN パラメーター 33
BMP 添付ファイル 32
BMP 文書
 媒体タイプ 29
 MIME コンテンツ・タイプ 29
browser セクション 42

C

CACHEDIR パラメーター 17
CACHEDOCS パラメーター 17
CACHEMAXTHRESHOLD パラメーター 18
CACHEMINTHRESHOLD パラメーター 18
CACHESIZE パラメーター 18
CACHEUSERIDS パラメーター 19
CGI API
 解説 65
Change Password
 関数の説明 6
 関数呼び出しのサンプル 71
 パラメーター 69
 API 69
CODEPAGE パラメーター 19, 157
CONFIGFILE パラメーター 24, 26
CONFIGURATION セクション 15
Cookies 37, 52
CREDIT.HTM 47

D

DBCS 157
DEFAULT BROWSER セクション 34
DOCSIZE パラメーター 20
Document Hit List
 関数の説明 6
 関数呼び出しのサンプル 76
 区切り文字付き ASCII 出力 153
 パラメーター 72
 API 72
 Java API 113, 117, 126, 128

E

EMAIL 文書
 媒体タイプ 30
 表示 37
 変換 37
 MIME コンテンツ・タイプ 30
EMAILVIEWING パラメーター 37
ENCRYPTCOOKIES パラメーター 37
ENCRYPTURL パラメーター 38
END パラメーター 34

F

FOLDERDESC パラメーター 38

G

GET メソッド 8

GIF 添付ファイル 33

GIF 文書

媒体タイプ 29

MIME コンテンツ・タイプ 29

H

HOST パラメーター 15

HTTP サーバー

構成ファイル 149

httpd.conf ファイル 149

HTTP サーバー・オプション

デバッグ 43

デフォルト・ブラウザ 34

ブラウザ 34, 42

AFP2HTML セクション 24

AFP2PDF Transform 25

AFP2PDF セクション 25

AFP2WEB Transform 24

APPLETDIR パラメーター 16

ATTACHMENT IMAGES セクション 32

BEGIN パラメーター 33

CACHEDIR パラメーター 17

CACHEDOCS パラメーター 17

CACHEMAXTHRESHOLD パラメーター 18

CACHEMINTHRESHOLD パラメーター 18

CACHESIZE パラメーター 18

CACHEUSERIDS パラメーター 19

CODEPAGE パラメーター 19

CONFIGFILE パラメーター 24, 26

CONFIGURATION セクション 15

END パラメーター 34

IMAGEDIR パラメーター 20

INSTALLDIR パラメーター 25, 26

LANGUAGE パラメーター 21

MIMETYPES セクション 27

NOHTML セクション 33

REPORTSERVERTIMEOUT パラメーター 23

SECURITY セクション 23

SEPARATOR パラメーター 34

SERVERACCESS パラメーター 23

ShowSearchString パラメーター 21

TEMPDIR パラメーター 22

TEMPLATEDIR パラメーター 22

USEEXECUTABLE パラメーター 25, 27

HTTP サーバー・ソフトウェア

インストール 11

httpd.conf ファイル 149

I

Image Web Viewer

インストール 51

構成 59

説明 1

要件 52

IMAGEDIR パラメーター 20

INSTALLDIR パラメーター 25, 26

J

Java API

解説 101

説明 1

ソフトウェア要件 1

プログラミング・ガイド 103

Java アプレット

インストール 51

使用可能 26

説明 1, 5

ディレクトリー 26

要件 52

ラージ・オブジェクトのサポート 24, 146

APPLETCACHEDIR パラメーター 16

Java 仮想マシン 52

Java 行データ・ビューアー

構成 60

ODApplet.jre.path.IE パラメーター 40

ODApplet.jre.path.NN パラメーター 40

ODApplet.jre.version パラメーター 40

ODApplet.version パラメーター 40

Java サブアプレット

解説 99

JFIF 文書

媒体タイプ 30

MIME コンテンツ・タイプ 30

JVM 52

L

LANGUAGE パラメーター 21, 157

Line Data Java アプレット

インストール 51

説明 1, 5

要件 52

APPLETCACHEDIR パラメーター 16

LINEVIEWING パラメーター 38

LOG パラメーター 43, 159

LOGDIR パラメーター 43

Logoff

関数の説明 7

関数呼び出しのサンプル 78

パラメーター 77

API 77

Logon
関数の説明 7
関数呼び出しのサンプル 81
区切り文字付き ASCII 出力 152
パラメーター 79
API 79
LOGON.HTM 47

M

MAXHITS パラメーター 39
MIME コンテンツ・タイプ 27, 113
MIMETYPES セクション 27

N

NLS 19, 21, 157
NOHTML セクション 33
NOLINKS パラメーター 39

O

ODApplet.jre.path.IE パラメーター 40
ODApplet.jre.path.NN パラメーター 40
ODApplet.jre.version パラメーター 40
ODApplet.version パラメーター 40
ODCallback 131
ODCriteria
オペランド 113, 119, 122
検索値 113, 119, 122
名前 113
文書、更新 137
文書の更新 137
ODCriteria.getFixedValues 122
ODCriteria.getName 113
ODCriteria.getOperand 113, 119
ODCriteria.getType 122
ODCriteria.getValidOperands 122
ODCriteria.getValues 122
ODCriteria.setOperand 122, 137
ODCriteria.setSearchValue 113, 137
ODCriteria.setSearchValues 113, 119, 122
ODFolder
アプリケーション・グループ 111
記述 113
基準 113, 119, 122
クローズ 111, 113, 117, 119
検索 113, 117, 119, 122, 126, 128
検索の取り消し 119
名前 113, 126
表示順序 113, 126
文書、印刷 131
文書、検索 128
文書の印刷 131
メッセージ 113

ODFolder (続き)
Retrieve Document 128
ODFolder.close 111, 113, 117, 119, 128
ODFolder.getApplGroups 111
ODFolder.getCriteria 113, 119, 122
ODFolder.getDescription 113
ODFolder.getDisplayOrder 113, 117, 126
ODFolder.getName 113, 126
ODFolder.getNumApplGroups 111
ODFolder.getSearchMessage 113
ODFolder.printDocs 131
ODFolder.retrieve 128
ODFolder.search 113, 117, 119, 126, 128
ODFolder.setApplGroupForSearchWithSQL 117
ODHit
注 133, 135
表示値 113, 117, 126
文書、検索 128
文書、更新 137
文書タイプ 113
文書の更新 137
文書の場所 113
文書リスト 126
Annotations 133, 135
MIME コンテンツ・タイプ 113
Retrieve Document 128
ODHit.addNote 135
ODHit.getDisplayValue 113, 117, 126, 137
ODHit.getDisplayValues 113
ODHit.getDocId 113, 128
ODHit.getDocLocation 113
ODHit.getDocType 113
ODHit.getMimeType 113
ODHit.getNotes 133, 135
ODHit.retrieve 128
ODHit.update 137
ODNote
位置 133
色 133
グループ名 133
時刻 133
テキスト 133
日付 133
ページ 133
ユーザー ID 133
Annotations 133, 135
ODNote.getColor 133
ODNote.getDateTime 133
ODNote.getGroupName 133
ODNote.getOffsetX 133
ODNote.getOffsetY 133
ODNote.getPageNum 133
ODNote.getText 133
ODNote.getUserid 133
ODNote.isOkToCopy 133, 135
ODNote.isPublic 133, 135

ODNote.setGroupName 135
ODNote.setText 135
ODServer
アプリケーション名 109
検索の取り消し 119
サーバー 109
サーバーへの接続 108
サーバー・プリンター 131
接続先 109
接続タイプ 109
切断 109
パスワード 109, 140
パスワード設定 140
パスワードの設定と入手 109
パスワード変更 140
フォルダー、オープン 128
フォルダー記述 125
フォルダーのオープン 122, 128
フォルダー名 125
プリンター 131
文書、検索 128
ポート 109
ユーザー ID 109
ユーザー ID の設定と入手 109
ローカル・ディレクトリー 109
Retrieve Document 128
ODServer.cancel 119
ODServer.changePassword 140
ODServer.getConnectType 109
ODServer.getFolderNames 125
ODServer.getFoldersDescription 125
ODServer.getLocalDir 109
ODServer.getNumFolders 125
ODServer.getPassword 109
ODServer.getPort 109
ODServer.getServerName 109
ODServer.getServerPrinters 131
ODServer.getUserId 109
ODServer.logoff 108, 109
ODServer.logon 108, 109
ODServer.openFolder 122, 128
ODServer.retrieve 128
ODServer.setApplicationName 109
ODServer.setConnectType 109
ODServer.setLocalDir 109
ODServer.setPassword 109
ODServer.setPort 109
ODServer.setServer 109
ODServer.setUserId 109
ODServer.terminate 108, 109
ODWEK ソフトウェア
インストール 11
OnDemand Internet Connection
説明 1
OnDemand サーバー・オプション
デフォルト 13

OnDemand サーバー・オプション (続き)
パラメーター 14
HOST パラメーター 15
PORT パラメーター 14, 15
PROTOCOL パラメーター 14, 15
@SRV@_DEFAULT セクション 13
@SRV@_server セクション 14
OS/400
インストール 12

P

PCX 文書
媒体タイプ 31
MIME コンテンツ・タイプ 31
PDF 文書
媒体タイプ 31
MIME コンテンツ・タイプ 31
PORT パラメーター 14, 15
POST メソッド 8
Print Document
関数の説明 7
関数呼び出しのサンプル 86
パラメーター 82
API 82
Java API 131
PROTOCOL パラメーター 14, 15

R

REPORTSERVERTIMEOUT パラメーター 23
Retrieve Document
関数の説明 7
関数呼び出しのサンプル 91
パラメーター 87
API 87

S

Search Criteria
関数の説明 7
関数呼び出しのサンプル 94
区切り文字付き ASCII 出力 152
パラメーター 92
API 92
Java API 113, 117, 122
SQL ストリング 117
SECURITY セクション 23
SEPARATOR パラメーター 34
SERVERACCESS パラメーター 23
SERVERPRINT パラメーター 40
SERVERPRINTERS パラメーター 41
SHOWDOCLOCATION パラメーター 41
ShowSearchString パラメーター 21
SQL 検索ストリング、Java API を使用した 117

T

TCP/IP 通信プロトコル 14, 15
TEMPDIR パラメーター 22
TEMPLATEDIR パラメーター 22
TEMPLATE.HTM 49
TIFF 文書
 媒体タイプ 31
 MIME コンテンツ・タイプ 31
TrueType フォント
 AFP フォントのマッピング 58
TXT 添付ファイル 33

U

Update Document
 関数の説明 7
 関数呼び出しのサンプル 96
 パラメーター 95
 API 95
 Java API 137
USEEXECUTABLE パラメーター 25, 27

V

View Annotations
 関数の説明 7
 関数呼び出しのサンプル 98
 区切り文字付き ASCII 出力 154
 パラメーター 97
 API 97
VIEWNOTES パラメーター 41

W

Web アプリケーション
 サンプル 47
Web サーバー・オプション
 言語 157
 コード・ページ 157
 DBCS 157
 NLS 157
Web ページ
 サンプル 47

[特殊文字]

@SRV@_DEFAULT セクション 13
@SRV@_server セクション 14



プログラム番号: 5722-RD1

Printed in Japan

SC88-4013-01



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12