



Common Server ODWEK インストールと構成ガイド



Common Server ODWEK インストールと構成ガイド

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、197 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 i (製品番号 5770-RD1) のバージョン 7、リリース 1 および新しい版で明記されていない限り、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。

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： SC19-2791-00
IBM Content Manager OnDemand for i
Version 7 Release 1
Common Server ODWEK Installation and Configuration Guide

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2010.4

© Copyright International Business Machines Corporation 2001, 2010.

目次

本書について	v	[CONFIGURATION].	19
本書の対象読者	v	[SECURITY]	27
本書の構成	v	[AFP2HTML]	29
前提条件および関連情報	v	[AFP2PDF].	30
WWW で入手可能なその他の情報	vi	[MIMETYPES]	32
前提知識	vi	[ATTACHMENT IMAGES]	37
OnDemand Information Center	vii	[NO HTML]	39
OnDemand のアクセシビリティ情報	vii	[DEFAULT BROWSER]	40
IBM i ナビゲーター	vii	[browser]	47
		[DEBUG]	48
		サンプルの ARSWWW.INI ファイル	49
		次のステップ	51
変更の要約	ix		
新機能	ix	第 4 章 サンプル・アプリケーションの構成	53
コマンドの機能強化	xii	LOGON.HTM	55
以前に追加されて OnDemand バージョン 7.1 で組		CREDIT.HTM	55
み込まれた追加機能	xiii	TEMPLATE.HTM	56
バージョン 7.1 へのアップグレード時の注意事項	xiii	次のステップ	56
第 1 章 製品の概要	1	第 5 章 Web ビューアーのインストール 57	
プログラミング・インターフェースについて	2	概要	57
ビジネスの観点から見た ODWEK Java API のユー		要件	58
ス・ケース	4	インストール	58
お客様の共通ユース・ケース	5	AFP Web Viewer	59
ビューアーについて	6	ユーザー定義ファイルの配布	60
ODWEK の使用	7	AFP Web Viewer ファイルのインストール	61
製品関数	8	サブディレクトリーの追加	61
Add Annotation	8	ユーザー定義ファイルの保管	62
Change Password	8	フォント・ファイルの構成	63
Document Hit List	8	AFP Web Viewer インストール・ファイルの構築	64
Logoff	8	ユーザーのワークステーションへの AFP Web	
Logon	8	Viewer のインストール	64
Retrieve Document	9	AFP フォントのマッピング	64
Search Criteria	9	AFP レポートの表示	65
Server Print Document	9	オーバーレイの表示	66
Update Document	9	Image Web Viewer	66
View Annotations	9	Java 行データ・ビューアー	67
Delete Annotations	9	次のステップ	69
サーバーとデータのセキュリティ	10		
IBM Web Interface for Content Management (WEBi)	11	第 6 章 インストールの検証	71
		CGI プログラムの検証	71
		サブレットの検証	72
		トラブルシューティング	73
		次のステップ	74
第 2 章 インストール・チェックリスト 13			
第 3 章 HTTP サーバーのインストール		付録 A. CGI API リファレンス	75
および構成	15	Add Annotation	76
インストール要件	15	Change Password	79
その他の要件	15	Document Hit List	82
IBM i でのインストール	16	Logoff	87
次のステップ	17		
ARSWWW.INI ファイルの指定	17		
[@SRV@_DEFAULT]	17		
[@SRV@_server].	18		

Logon	89	AFP2WEB Transform 用のオプション	169
Print Document (サーバー)	92	変換された文書の表示	170
Retrieve Document	97	付録 F. AFP から PDF への変換	171
Search Criteria	102	AFP2PDF.INI ファイルの指定	171
Update Document	105	変換された文書の表示	172
View Annotations	107	付録 G. HTTP サーバー構成ファイル 173	
付録 B. Java サブレット・リファレンス	109	HTTP Apache サーバー	173
付録 C. Java API リファレンス.	111	WebSphere Application Server	174
付録 D. Java API プログラミング・ガイド	113	付録 H. 非 HTML 出力	175
クライアント/サーバー・アーキテクチャー	113	区切り文字付き ASCII 出力	175
Java 環境のパッケージ化	113	Logon	176
プログラムのヒント	115	注	176
システム・パラメーターの構成	115	Search Criteria	176
ARSWWW.PROPS ファイルのサンプル	117	注	176
トレースおよび診断情報	117	Document Hit List	177
トレース	117	注	178
例外処理	118	View Annotations	178
定数	119	エラー・メッセージ	179
ODWEK アプリケーションの実行	119	注	179
OnDemand サーバーへの接続	120	付録 I. 各国語サポート.	181
接続の確立	120	DBCS 言語用の ODWEK の構成	181
パスワードの設定と入手	120	ODWEK におけるコード・ページ変換	181
OnDemand サーバーを使用した作業	121	ICU 変換ライブラリー	182
Java API を使用した非デフォルト・ポートへの接続	123	付録 J. 問題判別ツール	185
ODConfig の構築	123	Java ダンプ	187
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	124	IBM Thread and Monitor Dump Analyzer	187
フォルダーの検索	126	Java 診断コマンド	188
SQL ストリングを使用したフォルダーの検索	131	jmap	188
検索の取り消し	135	jstat	189
検索基準のリスト表示	138	HPROF: ヒープ・プロファイラー	189
フォルダーおよびフォルダー情報のリスト表示	142	HAT: Heap Analysis Tool	189
フォルダー基準情報の表示	144	Diagnostic Tool for Java Garbage Collector	189
文書のリストの表示	146	HeapAnalyzer	190
文書の検索	149	HeapRoots	191
文書の印刷	153	付録 K. Apache HTTP Server を使用した CGI のマルチリンガル・サポート機能	193
注に関する情報のリスト表示	155	ソフトウェア前提条件	193
注の追加	157	実装	193
注の削除	159	特記事項.	197
文書の更新	161	商標	198
パスワードの変更	164	索引	201
付録 E. AFP から HTML への変換	167		
AFP2HTML.INI ファイルの形式	167		

本書について

本書には、IBM® Content Manager OnDemand for i バージョン 7 リリース 1 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 i Common Server システムのデータにアクセスできるように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 製品の概要』
- 15 ページの『第 3 章 HTTP サーバーのインストールおよび構成』
- 53 ページの『第 4 章 サンプル・アプリケーションの構成』
- 57 ページの『第 5 章 Web ビューアーのインストール』
- 75 ページの『付録 A. CGI API リファレンス』
- 109 ページの『付録 B. Java サブレット・リファレンス』
- 111 ページの『付録 C. Java API リファレンス』
- 113 ページの『付録 D. Java API プログラミング・ガイド』
- 167 ページの『付録 E. AFP から HTML への変換』
- 171 ページの『付録 F. AFP から PDF への変換』
- 173 ページの『付録 G. HTTP サーバー構成ファイル』
- 175 ページの『付録 H. 非 HTML 出力』
- 181 ページの『付録 I. 各国語サポート』
- 185 ページの『付録 J. 問題判別ツール』
- 193 ページの『付録 K. Apache HTTP Server を使用した CGI のマルチリンガル・サポート機能』

前提条件および関連情報

IBM i の技術情報を検索するには、まず IBM i Information Center をご利用ください。

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

- Web サイト <http://www.ibm.com/systems/i/infocenter/> からアクセスする方法。
- お買い上げの IBM i に付属する CD-ROM からアクセスする方法。

IBM i Information Center SK3T-4091-07

IBM i Information Center の内容は、次のとおりです。

- IBM i インストールおよびアップグレード、データ・マイグレーション、サービスおよびトラブルシューティング、可用性、System i® の統合、System i への接続、データベース、Linux®、WebSphere®、Java™、CL コマンド、システム API、および解説書などを含む、更新された情報および新しい情報。
- IBM i ソフトウェアのトラブルシューティングや構成に役立つ、アドバイザーおよびその他の対話式ツール。

WWW で入手可能なその他の情報

WWW では、さらに多くの IBM i の情報を参照できます。一般的な情報は、IBM i のホーム・ページ (Web サイト: <http://www.ibm.com/systems/i/>) から入手できます。

IBM i の高度な機能に関するワークショップを利用する場合は、Technical Studio (<http://www.redbooks.ibm.com/tstudio/>) にアクセスしてください。

インターネットを利用すれば、世界中どこからでも、IBM i プログラム一時修正 (PTF) を参照、選択、発注、および入手できます。IBM i のインターネット PTF (ダウンロード) および予防サービス計画 (PSP) に関する情報は、インターネット・サイト <http://as400service.ibm.com> から入手できます。

製品資料は、CM OnDemand for i 製品の Web サイト上にあるライブラリー・ページからサポート・ページに移動しました。利用可能なすべての OnDemand for i 製品資料のリストを表示するには、<http://www.ibm.com/software/data/ondemand/400/support.html> に移動します。左側の欄の「Self help」セクション以下で、副見出しの「Learn」を参照してください。

前提知識

ODWEK 用の資料では、インターネット、Web サーバーと Web ブラウザー、TCP/IP、ネットワークング、および OnDemand について理解していることが前提となっています。本書では、読者がハイパーテキスト・マークアップ言語 (HTML)、コモン・ゲートウェイ・インターフェース (CGI)、および Java プログラミングに精通していること、Web ページにコンテンツを提供できること、HTTP サーバー、Java 対応の Web サーバー、および Java アプリケーション・サーバーの構成方法と操作方法を知っていること、および OnDemand サーバーを管理できることを前提としています。

Java AFP2HTML Viewer の使用を計画している場合は、AFP2WEB Transform を入手して、ODWEK アプリケーションが配置されているサーバー上にインストールし、構成する必要があります。AFP2WEB Transform に関する詳細は、IBM 担当員にご相談ください。さらに、AFP2WEB Transform で処理する予定の拡張機能表示

(AFP) の文書や資源にも、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、167 ページの『付録 E. AFP から HTML への変換』を参照してください。

OnDemand から取り出した 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、AFT2PDF Transform を入手して、Web サーバーにインストールし、構成する必要があります。AFP2PDF Transform に関する詳細は、IBM 担当員にご相談ください。さらに、AFP2PDF Transform で処理するために組み込んだ 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、171 ページの『付録 F. AFP から PDF への変換』を参照してください。

OnDemand Information Center

IBM i Information Center (前述) のほかに、OnDemand インフォメーション・センターに必ずアクセスしてください。このセンターは CM OnDemand に関する情報のみに重点を置いています。OnDemand Information Center では、製品情報に、迅速で一元化されたオンライン・アクセスを行えます。このタスク・ベースの文書リポジトリを使用すると、コマンド、エラー・コード、その他の興味のあるトピックについて、製品ライブラリー全体にわたる検索を行うことができます。興味深いページやよく参照するページにブックマークを付けると、以後簡単に参照できるようになります。

OnDemand インフォメーション・センターにアクセスするには、
<http://publib.boulder.ibm.com/infocenter/cmod/v8r4m1//index.jsp> に移動します。

OnDemand のアクセシビリティ情報

本製品でサポートするアクセシビリティ機能に関する詳細情報については、「*IBM Content Manager OnDemand for i: Common Server 管理ガイド*」を参照してください。

IBM i ナビゲーター

IBM i ナビゲーターは、IBM i 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。IBM i ナビゲーターの機能には、タスクでユーザーを導く、システムのナビゲーション、構成、計画機能、およびオンライン・ヘルプがあります。IBM i ナビゲーターを使用すると、サーバーの運用と管理が容易になり、生産性が向上します。IBM i の新規拡張機能への、唯一のユーザー・インターフェースとなっています。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。

IBM i ナビゲーターの詳細については、IBM i Information Center および Web サイト <http://www.ibm.com/servers/eserver/series/navigator/> を参照してください。

変更の要約

「IBM Content Manager OnDemand for i Common Server ODWEK インストールと構成ガイド」のこの版には、新しい技術情報の説明が含まれています。場合によっては、変更が加えられていても変更バーが示さ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。

新機能

- 多種の OnDemand 機能について、プログラム呼び出しに取って代わる新しいコマンドが追加されています。
 - 新しい「OnDemand のインスタンス作成 (CRTINSTOND)」コマンドを、QRLMINST プログラムを呼び出す代わりに使用して、新規の OnDemand インスタンスを作成します。この新しいコマンドには、QRLMINST プログラムで提供されていたものを超える追加パラメーターがあります。このコマンドでは、ポート、自動開始、セキュリティー、および ASP (補助記憶域プール) に関連したパラメーターをコマンドに対して指定することができ、多くの場合に `ars.ini` および `ars.cfg` 構成ファイルを編集する必要がなくなります。プログラム呼び出しインターフェースは、現在サポートされなくなったことに注意してください。コマンド・インターフェースが、バージョン 7.1 でサポートされている唯一のインターフェースです。
 - 新しい「スプール・ファイルのマージ (MRGSPLFOND)」コマンドを、旧 MRGSPLFOND サンプル・コマンド (以前のリリースに同梱されている) または QRLMQMRGF プログラム呼び出しの代わりに使用して、アーカイブする前に複数の小さいスプール・ファイルを 1 つの大きなファイルにマージします。バージョン 7.1 で出荷される新しい MRGSPLFOND コマンドには、新しいパラメーターや拡張されたパラメーターがあり、以前のサンプル・コマンドよりかなり豊富な機能が提供されています。以前のサンプル・コマンドを使用する既存のプログラムはすべて、バージョン 7.1 のパラメーターを使用するように変更する必要があります。
 - 「メディアの移行 (MGRMEDRDAR)」コマンドを、QRLCSFAMMF プログラム呼び出しの代わりに使用して、OnDemand データをあるメディア・タイプから別のメディア・タイプへ移行します。(このコマンドは、OnDemand バージョン 6.1 で使用可能でしたが、プログラム呼び出しが現在はサポートされていないことを確認していただくため、ここにリストしてあります。バージョン 7.1 では、コマンド・インターフェースのみがサポートされます。)
 - 「ONDEMANDのポリシー・レベル日付変更 (CHGPLDOND)」コマンドは、アーカイブ・データの移行ポリシー・レベルの日付を変更する必要がある場合に QRLCASMCLD プログラム呼び出しの代わりに使用します。(このコマンドは、OnDemand バージョン 6.1 で使用可能でしたが、プログラム呼び出しが現在はサポートされていないことを確認していただくため、ここにリストしてあります。バージョン 7.1 では、コマンド・インターフェースのみがサポートされます。)

- 新しい System i ナビゲーター機能が追加されて、OnDemand で使用するネットワーク・ファイル・システム (NFS) ディスク・プールのセットアップに、プログラム呼び出しの代わりに使用できるようになりました。
 - QRLCASMNFS プログラムの呼び出しの代わりに、OnDemand System i ナビゲーター・プラグインの新しいネットワーク・ファイル・システム (NFS) パネルを使用してください。
 - OnDemand で NFS を使用する場合は、「Content Manager OnDemand for i Common Server 管理ガイド」を参照してください。
- 拡張保存管理機能が、OnDemand for i の 7.1 から別料金の機能として使用できるようになりました。これにより、OnDemand のエンド・ユーザー・クライアントには保留と保留解除の機能が提供され、さらに、ARSDOC API の HOLD-ADD および HOLD-RELEASE パラメーターと、ODWEK の ODHold および ODHit Java API が提供されます。保留が解除されるまで保留文書の期限を満了にしないようにすることが可能です。「IBM Content Manager OnDemand for i Common Server 計画とインストール・ガイド」に記載されている ARS.CFG ファイルの新しい ARS_SUPPORT_HOLD の項目を参照してください。

重要な注意: 拡張保存管理を使用する場合は、OnDemand Disk Storage Manager (DSM) が満了処理の完全な制御を行っている必要があります。OnDemand Archive Storage Manager (ASM) あるいは Tivoli® Storage Manager を使用している場合、いずれの記憶域管理機能でもデータを期限切れにする機能を使用不可にする必要があります。例えば、ASM では、拡張保存管理を有効に設定している OnDemand アプリケーション・グループで使用する移行ポリシーの有効期限切れレベルを、使用不可にするか削除するということです。

- 記憶域管理データはすべて、QUSRRDARS ライブラリーから、そのデータが関連付けられている個々のインスタンス・ライブラリーに移動されました。少数のオブジェクトが、QUSRRDARS ライブラリーに残っています。この変更により、QUSRRDARS ライブラリーを独立補助記憶域プール (IASP) へ移動する必要がなくなりました。
- インスタンス・ライブラリーは、ユーザーの補助記憶域プール (ASP 2 から 32 まで) に配置することができます。
- 新しい System Load アプリケーション・グループが追加されました。これには、OnDemand にロードされる各入力ファイルごとに 1 つの項目が入ります。OnDemand クライアントを使用して、どのデータが OnDemand に正常にロードされたかを確認することができます。
- ログ項目などの診断情報を収集するための新しい Java ベースの ARSSUPPORT ユーティリティーが使用可能になりました。このツールは、問題を IBM ソフトウェア・サポートに報告する必要がある場合に便利です。詳細は、「IBM Content Manager OnDemand for i Common Server 管理ガイド」を参照してください。
- Xerces2 Java Parser バージョン 2.6.2 が、このバージョンの Content Manager OnDemand for i に含まれています。このバージョンのパーサーの使用に関する説明は、「IBM Content Manager OnDemand for i Common Server 計画とインストール・ガイド」を参照してください。

- バージョン 7.1 には、さまざまな OnDemand Web Enablement Kit (ODWEK) の変更が組み込まれています。「IBM Content Manager OnDemand for i: Common Server ODWEK インストールと構成ガイド」の中の変更バーで、変更点を確認することができます。
- バッチ管理機能 (ARSXML API) が更新されて、新規に追加された属性、名前が変更された属性、除去された属性があります。例えば、更新と削除の場合の `_ALL` の名前属性値は除去されました。(`_ALL` は、エクスポート操作の場合には、現在もサポートされています。) 詳細は、「IBM Content Manager OnDemand for i Common Server 管理ガイド」を参照してください。
- 新しいサンプル・プログラムが追加され、既存のサンプル・プログラムも更新されています。QRDARS ライブラリーにある QSAMPLES2 ソース・ファイルを参照してください。
- Archive Storage Manager (ASM) ベースの期限満了についての資料が追加されています。この機能により Disk Storage Manager (DSM) を実行する必要がなくなる場合があります。この機能は、OnDemand 6.1 で使用可能でしたが、このトピックに関する資料が「IBM Content Manager OnDemand for i Common Server 管理ガイド」に追加されたことを確認していただくため、ここにリストしてあります。また、既存の OnDemand 記憶域管理機能と新規保存管理機能に関する、前述の拡張保存管理の項目にある「重要な注意」も参照してください。
- AFP2WEB Transform (AFP2PDF および AFP2HTML の両方を含む) は、OnDemand Web Enablement Kit (ODWEK) インターフェースを使用して AFP データが表示できるようにするもので、OnDemand for i バージョン 7.1 の別料金の機能として入手可能になっています。AFP2WEB Transform に関する詳細は、IBM 担当員またはビジネス・パートナーにご相談ください。
- PDF インデクサーは、資源のグループ化と未使用資源の除去をサポートします。資源のグループ化を有効にすると、単一入力ファイル内の文書にまたがる共通資源がグループ化されて、単一オブジェクトとして保管されます。入力ファイルの未使用資源は、索引付けの前に除去することができます。この新しい機能をサポートするために、PDF インデクサーに新しいパラメーター (RESTYPE と REMOVERES) が追加されています。詳しくは、「IBM Content Manager OnDemand for i Common Server 索引付けリファレンス」を参照してください。
- FileNet® P8 プラットフォームとの統合が、バージョン 7.1 で別料金の機能として新規にサポートされるようになりました。これにより、Content Manager OnDemand for i のメタデータを FileNet に送信して、FileNet の Business Process Management (BPM) と FileNet Records Manager (RM) の機能を活用することができます。この Content Federation Services for OnDemand 機能については、「IBM Content Manager OnDemand for i Common Server 計画とインストール・ガイド」内の ARS.CFG ファイルの新しい ARS_SUPPORT_CFSOD の項目と、「IBM Content Manager OnDemand for i Common Server 管理ガイド」の ARSDOC の新しい CFSOD-FED 機能の項目を参照してください。
- OnDemand 機能を制御するさまざまなデータ域に関する資料が、整理されて「IBM Content Manager OnDemand for i Common Server 管理ガイド」に追加されています。

- OnDemand ポストプロセッサ・プログラムの入力ファイルに新しい列ヘッダーが追加されました。この新しいヘッダーは、先頭が「<」文字で、末尾が「>」文字になります。ご使用のポストプロセッサ・プログラムで、この追加部分の互換性を必ずテストしておいてください。

コマンドの機能強化

- 「OnDemand のモニター開始 (STRMONOND)」コマンドに、新しく *DIR2 モニター・タイプが追加されました。これは、現行の *DIR モニター・タイプの場合の .IND ファイルのトリガーではなく、.ARD ファイル (ARSLOAD API と同様) によってトリガーされます。「OnDemand のモニター終了 (ENDMONOND)」コマンドにおける *DIR モニター・タイプは、開始が *DIR モニターとしてか *DIR2 モニターとしてかのいずれかに関係なく、どちらのタイプのモニターも終了させることに注意してください。
- 「OnDemand のモニター開始 (STRMONOND)」コマンドに新しいパラメーターが 2 つ追加されました。サーバー終了 (ENDSVR) パラメーターを使用すると、モニターの終了時にインスタンス・サーバーを終了させることができます。モニターのジョブ名 (JOB) パラメーターを使用すると、モニター・ジョブに使用する名前を指定することができます。
- 「OnDemand への報告書追加 (ADDRPTOND)」、「OnDemand からの報告書印刷 (PRTRPTOND)」、「OnDemand のモニター開始 (STRMONOND)」、「OnDemand からの報告書除去 (RMVRPTOND)」、「OnDemand へのインポート開始 (STRIMPOND)」、「アーカイブ済み記憶域管理の開始 (STRASMOND)」、「ディスク記憶装置管理の開始 (STRDSMOND)」、「スプール・ファイルのマージ (MRGSPLFOND)」、「ポリシー・レベル日付変更 (CHGPLDOND)」および「メディアの移行 (MGRMEDRDAR)」コマンドに 2 つの新しいパラメーターが追加されました。
 - INSTANCE(QUSROND) の代わりに新しいデフォルトである INSTANCE(*DFT) を使用すると、QDFTINST という名前のデータ域からデフォルトのインスタンス名を取り出すことができるため、各コマンドでインスタンス名を明示的に指定する必要がなくなりました。QDFTINST データ域に関して詳しくは、オンライン・ヘルプと「IBM Content Manager OnDemand for i Common Server 管理ガイド」を参照してください。
 - サーバーの開始 (STRSVR) パラメーターを指定すると、コマンドの実行時にインスタンス・サーバーを開始することができます。(STRSVR パラメーターは、CHGPLDOND コマンドには適用できません。) 詳しくは、オンライン・ヘルプを参照してください。
- 「ディスク記憶装置管理の開始 (STRDSMOND)」コマンドに新しく VALIDATE パラメーターが追加されて、すべてのディスク記憶装置ファイルが正しくリンクされて、適切なファイル許可を含むことを確認できます。
- 「OnDemand のモニターの開始 (STRMONOND)」コマンドに、出力キュー・モニターの開始時に指定できる、開始モニターの最大数のパラメーターが、99 から 9 に削減されて、同じ出力キューに対して開始されたモニターの数が多い場合に発生しやすいロッキングの遅延を回避できるようになりました。

以前に追加されて OnDemand バージョン 7.1 で組み込まれた追加機能

- 「TCP/IP サーバー開始 (STRTCPSVR)」コマンドと「TCP/IP サーバー終了 (ENDTCPSVR)」コマンドで、SERVER パラメーターに *ONDMD (OnDemand 用) を指定する際に INSTANCE パラメーターがサポートされるようになりました。開始する特定のインスタンス名を指定するか、あるいは 3 つの特殊値 (*DFT, *ALL, *AUTOSTART) のいずれかを使用して指定できます。インスタンスを開始または終了させる QRLMCTL プログラムの呼び出しは、現在もサポートされていますが、STRTCPSVR コマンドと ENDTCPSVR コマンドの使用を推奨します。詳しくは、「IBM Content Manager OnDemand for i: Common Server 計画とインストール・ガイド」およびオンライン・ヘルプを参照してください。
- 同一ネットワーク上の分散アプリケーション間で情報を共有するためのオープン・インダストリー・スタンダード、Lightweight Directory Access Protocol (LDAP) を使用して、サーバー上の基本的なログイン認証を直接管理することができます。詳しくは、「IBM Content Manager OnDemand for i Common Server 管理ガイド」を参照してください。
- インターネット・プロトコル、バージョン 6 (IPv6) のアドレッシング・フォーマット (これは、TCP/IP 用の IPv4 アドレッシング形式の改訂版です) が Content Manager OnDemand for i のバージョン 7.1 で組み込まれました。詳しくは、「IBM Content Manager OnDemand for i Common Server 計画とインストール・ガイド」を参照してください。
- 「メディアの移行 (MGRMEDRDAR)」コマンドのターゲット (TGT) 宛先として、*ASM を指定できるようになりました。これにより、データを Spool File Archive 方式 (RMC による管理) から Common Server 方式 (ASM による管理) へ移動させることができます。
- PDF インデクサーにより、入力ファイル名の一部を ARSLOAD API を使用してデータを保管する際の索引値として使用できるようになりました。

バージョン 7.1 へのアップグレード時の注意事項

- 旧バージョンの OnDemand からのアップグレードを行う場合、Content Manager OnDemand for i バージョン 7 リリース 1 にアップグレードするためには OnDemand サーバーのバージョン 7.1.2.8 以降を実行している必要があります。現行サーバーのバージョンを判別する方法については、「IBM Content Manager OnDemand for i Common Server 計画とインストール・ガイド」を参照してください。
- Content Manager OnDemand for i のバージョン 7.1 は、バージョン 7.1.2.0 より前の OnDemand クライアント・ソフトウェアをサポートしません。これには、OnDemand Windows® (エンド・ユーザー) クライアント、ODWEK CGI/Servlet/Java API、CICS®, および II4C (eClient) が含まれますが、これらに限定されません。
- OnDemand Administrator クライアントは、OnDemand サーバーと同等以上のバージョンでなければなりません。Content Manager OnDemand for i のバージョン 7.1 の場合、OnDemand Administrator クライアントは、バージョン 8.4.1.3 以降である必要があります。
- バージョン 7.1 にアップグレード後、最初のインスタンスの開始あるいは Archive Storage Management (ASM) プロセスの開始は、通常よりも時間がかかり

ます。ファイル変換および QUSRRDARS からインスタンス・ライブラリーへの
インスタンス固有のデータとオブジェクトの移動があるためです。

– 処理が進行していないのではないかという不安があっても、サーバー・ジョブ
あるいは ASM を終了させないでください。

– ファイル変換と QUSRRDARS からインスタンス・ライブラリーへのデータ移
動の間、状況メッセージがジョブ・ログに書き込まれているので、それを調べ
てジョブが進行していることを確認することができます。

• ご使用のシステムに定義した OnDemand インスタンス名と対応するジョブ記述
(オブジェクト・タイプ *JOBID) が QUSRRDARS ライブラリーに存在する場合、
そのジョブ記述は、QUSRRDARS からインスタンス・ライブラリーに移動され
ます。OnDemand が、インスタンス・サーバーの開始に使用するジョブ記述を
探さず場合、OnDemand はまず、インスタンス・ライブラリー内のインスタンス名と
同じ名前の *JOBID オブジェクトを探します。それが検出されなかった場合、
OnDemand は、QUSRRDARS ライブラリー内のインスタンス名と同じ名前の
*JOBID オブジェクトを探します。それも検出されなかった場合、OnDemand は
QRDARS ライブラリーにある QOND400 ジョブ記述を使用します。

• バージョン 7.1 において、以下のプログラム呼び出しは、新規コマンドまたは新
規 System i ナビゲーター・インターフェースによって置換されています。

– MRGSPLF サンプル・プログラムと以前の MRGSPLFOND サンプル・プログ
ラムは、新規 MRGSPLFOND コマンドに置き換わりました。

– QRLMINST プログラムは、CRTINSTOND コマンドに置き換わりました。

– QRLCASMNFS プログラムは、System i ナビゲーター・プラグイン・インター
フェースに置き換わりました。

– QRLCSFAMMF プログラムは、MGRMEDRDAR コマンドに置き換わりまし
た。

– QRLCASMCLD プログラムは、CHGPLDOND コマンドに置き換わりました。

• OnDemand コマンド (CMDOND) メニューには、OnDemand for i Common Server
コマンドがすべて組み込まれており、OnDemand バージョン 7.1 で出荷される唯
一の 5250 メニューです。以下の OnDemand 5250 メニューは除去されました。

– CMDRDAR

– ONDEMAND

– RDARS

– RDARSDEF

– RDARSM

– RDARSOBJ

– RDARSRLA

– RDARSRPT

– RDARSUTL

• PDF インデクサーで使用するフォントをインストールした場合、フォントの位置
を確認して、必要に応じてそれらを PDF インデクサーで要求するディレクトリ
ーに移動させてください。詳しくは、「IBM Content Manager OnDemand for i
Common Server 索引付けリファレンス」を参照してください。

- QRDARS ライブラリー内の QPRLR133 プリンター・ファイルは、現在は製品に含まれていません。これは、現在は使用可能ではない Spool File Archive のコンポーネントです。プリンター・ファイルは、バージョン 7.1 のインストール時に除去されています。
- アンマウント・ファイル・システム・プログラム (QRLCASMUFS) は、インスタンス名に *ALL をサポートしなくなりました。すべてのインスタンス固有のファイルが、個々のインスタンス・ライブラリーに含まれるようになったためです。プログラムの呼び出し時に特定のインスタンス名を指定してください。
- OnDemand Web Enablement Kit (ODWEK) CGI/Servlet の最新バージョンを使用する前に、Web Enablement Kit のキャッシュと一時ディレクトリーからすべてのファイルを削除する必要があります。これらのディレクトリーは、arswww.ini ファイル内の CACHEDIR と TEMPDIR 項目で指定されています。
- 既存の Spool File Archive の実装は、それを実行するシステムが 7.1 にアップグレードされる前に、Spool File Archive 環境から Common Server に移行する必要があります。Content Manager OnDemand リリース 5.3 および 5.4 には、Common Server 環境だけでなく、Spool File Archive、AnyStore、Record Archive、および Object Archive のレガシー環境も同様に組み込まれていました。これらの環境は、i5/OS® 5.4 までは、すべて完全にサポートされています。

2006 年 2 月 15 日付けの IBM 製品発表レター (レター番号 DBA06012-0) に記載されているように、5.4 が Spool File Archive、AnyStore、Record Archive、および Object Archive を出荷およびサポートする最終リリースです。Content Manager OnDemand 5.3 から、Spool File Archive 移行ユーティリティーが Content Manager OnDemand ライセンス・プログラム・プロダクトに含まれて使用可能になっています。移行ユーティリティーは、報告書定義および索引を、従来の Spool File Archive 環境から Common Server 環境に移行する機能を提供します。移行ユーティリティーに関する詳細は、「IBM Content Manager OnDemand for iSeries®: Common Server 計画とインストール V5.3」の付録 A を参照してください。既存の Spool File Archive の実装は、システムがバージョン 7.1 にアップグレードされる前に、Spool File Archive 環境から Common Server に移行する必要があります。

第 1 章 製品の概要

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。

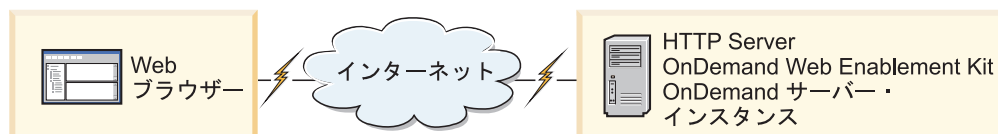


図 1. ODWEK を使用して、OnDemand に保管されているデータへのアクセス

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 i Common Server 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 は、Java バージョン 1.4 以降を必要とします。

- 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、PNG、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 (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) 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、 AFP2WEB 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 HTML 文書に変換されます。管理者は、 AFP2WEB Transform をインストールおよび構成した後、 ARSWWW.INI ファイル (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) 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。

重要: 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 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、固有の ARSWWW.INI ファイル (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) と ODWEK プログラミング・インターフェースがあり、これには他のインスタンスはアクセスできません。 ODWEK プログラミング・インターフェースには、以下の 3 つがあります。

- 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ようにできます。

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。

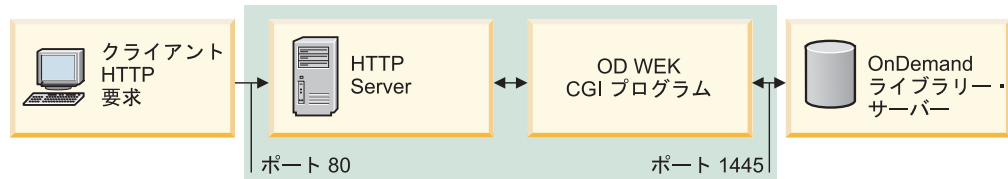


図 2. CGI インターフェースを使用した単一インスタンス

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。

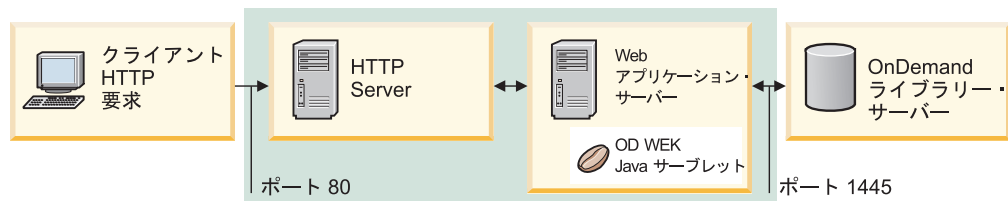


図 3. Java インターフェースを使用した単一インスタンス

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。

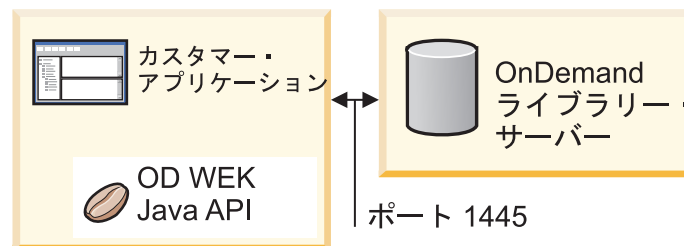


図 4. Java API インターフェースを使用した単一インスタンス

同じシステムで複数の ODWEK インスタンスを構成できます。CGI および Java サブレットの場合、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。Java API の場合は、ODWEK Java API 関数にアクセスする Java コード内に、固有のポート番号が指定されます。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。

図 5 に、複数インスタンス・トポロジーの例を示します。

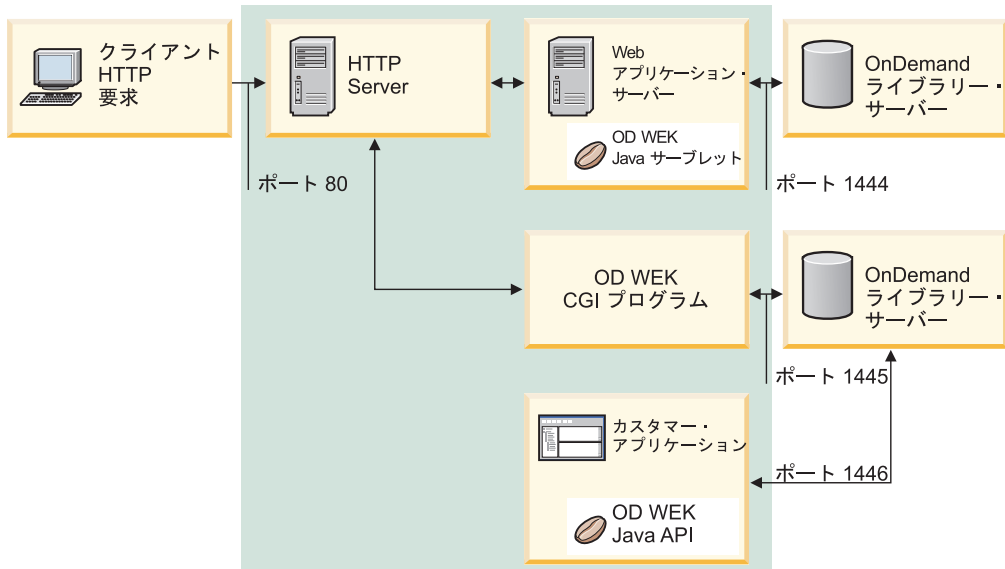


図 5. 複数インスタンス・トポロジー

ビジネスの観点から見た ODWEK Java API のユース・ケース

OnDemand に保管されているデータへのアクセスには、インターネットでのアクセスとイントラネットでのアクセスの 2 つの方法があります。

インターネット経由でデータにアクセスする場合、ユーザー（組織の外部にいるユーザー）は、情報の特定のサブセットに対するアクセス権限が与えられます。例えば、銀行用アプリケーションでは、インターネット・ユーザーは、ログオンして、過去 12 カ月までの最新ステートメントを表示することができます。これは、ユーザーが OnDemand アーカイブに対して持つ唯一のアクセス権限で、ユーザーには、このデータを取得するための非常に限定されたメニューが表示されます。典型的なインターネットの使用法では、OnDemand アーカイブに同時に数万人のユーザーからのアクセスが発生します。OnDemand アーキテクチャーは、使用可能なハードウェアおよびネットワーク資源の限界までシステムを拡張して、ビジネスによる使用量の増加に応じたシステム拡大を可能にします。

イントラネット経由でデータにアクセスする場合、ユーザー（組織で働いているユーザー）は、OnDemand アーカイブにおけるそのユーザー ID のアクセス権に基づいて、さまざまな種類のデータにアクセスする権限が与えられます。ユーザーには、セキュリティー・プロファイルに基づいて保管データへのアクセスを許可する選択基準が与えられます。ODWEK Java API が提供する柔軟性により、組織の特定の要件に一致するようにインターフェースのデザインをカスタマイズすることができます。

お客様の共通ユース・ケース

多くのさまざまなビジネス・セグメントにわたる共通のユース・ケースでは、登録済みのお客様に、表示できる文書の範囲を選択させることができます。金融業界の場合、文書は、事前に定義した月数から選択できる銀行ステートメントの可能性もあります。健康保険会社の場合、この文書は、給付金ステートメントの説明資料である可能性もあります。公益事業の会社の場合、文書は、請求書や送り状の可能性もあります。

お客様は、通常、その会社の Web サイトに登録されているユーザーです。登録プロセスにより、例えば、ユーザー ID とパスワードなどの資格情報が提供されます。会社のアプリケーションにアクセスするには、ユーザーは、その会社の Web アプリケーションによって認証してもらうために、ポータルなどの Web ページから資格情報をサブミットします。ユーザーが認証されると、アプリケーションは、そのユーザーが Web サイトにアクセスしたときに実行できる内容に制限をかけます。

OnDemand では、アクセスを必要とするすべてのユーザーにユーザー ID が定義されている必要があります。管理者は、登録された各インターネット・ユーザーごとにユーザー ID を作成することはできません。ユーザー数は、何千または何百万人に及ぶ可能性があります。したがって、ほとんどのアプリケーションは、特定フォルダーの検索と文書リトリブの許可を持つ各ユーザー ID に代わって検索を実行します。ユーザーが認証されなかったときに、検索値に制約を加えて固有の結果を確保する場合があります。検索するフォルダーに対して完全な許可を持つユーザー ID を 1 つ以上定義することができます。

カスタマーに表示する OnDemand ステートメントを持つものであれば産業に関係なく、カスタマーは次の 2 タイプのトランザクションを実行できます。

ユーザーとの対話なし

1 つ目のトランザクション・タイプは、通常、事前定義されている日付範囲内の、文書リストを表示するアプリケーションを伴います。ユーザーが認証されると、アプリケーションは固有のキーまたはキーの組み合わせを選択し、確実にそのユーザー用の検索結果になるようにして、ユーザーと対話をせずに検索を実行します。

ユーザーとの対話

もう 1 つのトランザクション・タイプは、ユーザーが検索結果リストから文書を選択する操作を伴います。文書のデータ・タイプによって、ユーザーにコンテンツを表示する方法が変わります。例えば、文書が AFP データ・ストリームとして OnDemand に保管されている場合、ユーザー・マシンにローカルでインストールされた AFP ビューアー・プログラムが必要になります。または、表示に適したデータ・ストリームになるようにデータ変換の実行が必要になります。多くの場合、アプリケーションは、ファイルを (例えば、AFP から PDF へ) 変換して、ユーザーへの表示を見やすくします。PDF ビューアーは、インストールが容易ですが、プラグインなどの AFP ビューアーは、Web サービスを提供する会社が入手して管理するには、比較的困難な可能性があります。

このユース・ケースに必要なステップは以下のとおりです。

1. ユーザーは、それぞれの資格情報を使用して Web サイトにサインオンします。Web アプリケーションは、ユーザーを認証して、接続プールから OnDemand 接続を取り出します。
2. Web アプリケーションは、検索する定義済みフォルダーを選択します。
3. Web アプリケーションは、各ユーザーに固有の検索を実行するために、1 つ以上の OnDemand フォルダー・フィールド値を割り当てます。
4. Web アプリケーションは、例えば、アカウント番号と日付範囲などで検索を実行し、検索結果のリストをユーザーに返します。
5. Web アプリケーションは、フォルダーを閉じ、接続を解放して接続プールに戻します。
6. ユーザーは、検索結果リストから表示する文書を選択します。オプションで、文書の取り出しを許可する前に、ユーザーが再度認証されるようにすることもできます。
7. Web アプリケーションは、接続プールから OnDemand 接続を取り出します。
8. Web アプリケーションが文書を取り出し、オプションでデータ形式変更を実行してからそのデータをユーザーに解放します。
9. Web アプリケーションは、OnDemand 接続を解放して接続プールに戻します。

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、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 を使用すると、AFP 文書の表示が可能になります。Image Viewer を使用すると、BMP、GIF、JPEG、PCX、PNG、および TIFF 文書の表示が可能になります。ビューアーは、ブラウザー・ウィンドウに文書を表示できます。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。ユーザーは、ブラウザー・ツールバーにビューアー・ツールバーを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。

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ー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 (CGI および Java サーブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) 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、AFP2WEB 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 (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) 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を含む、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ソフトウェアをワークステーションに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、非定期的な 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。

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

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、75 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。追加した注釈を後で表示すると、注釈テキストのほかに、日時スタンプおよび注釈を作成したユーザーの情報が表示されます。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。)

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。

重要: ODWEK インターフェースで文書検索をする際、索引データ内にパーセント記号 (%) とコロン (:) 文字があると失敗するか、あるいは、予測不能な結果が生じます。

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 許可を持っていないければなりません。(Application Group Access 許可を持っていれば、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。

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。)

Delete Annotations

Delete Annotations 関数により、特定の文書に付加されている注釈を削除することができます。

サーバーとデータのセキュリティー

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ー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ー・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ーを管理する担当者を設けてください。

セキュリティー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」フィールドに平文で表示されます。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。

重要: 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、43 ページの『ENCRYPTURL』を参照してください。

IBM Web Interface for Content Management (WEBi)

IBM Web Interface for Content Management (WEBi) は、ODWEK Java API を使用します。WEBi は、完全な機能を有する Web 2.0 テクノロジー・ベースのクライアントです。専用のブラウザとして、あるいは、1 つ以上のカスタム開発ブラウザの実装に加えた 2 次的ブラウザの実装として、WEBi を使用することができます。 OnDemand サーバーにアクセスする他のすべての Web 実装と同様に、WEBi は、ODWEK Java API 上に構築されます。

第 2 章 インストール・チェックリスト

OnDemand Web Enablement Kit 環境のセットアップでは、一般に、以下のタスクを実行する必要があります。

1. IBM ソフトウェア・サポートに最新の OnDemand 用 PTF の有無を問い合わせます。最新の PTF のリストは、Informational APAR の番号 II14497 で検索できます。
2. IBM ソフトウェア・サポートに最新の IBM i HTTP Server 用グループ PTF の有無を問い合わせます。この HTTP Server のプロダクト番号は 5770-DG1 です。最新の HTTP Sever 用 PTF のリストを表示するには、http://www-912.ibm.com/s_dir/sline003.NSF/GroupPTFs?OpenView&view を参照して該当するグループ PTF の番号をクリックしてください。
3. IBM サポートに最新のデータベース用グループ PTF の有無を問い合わせます。最新の DB2® 用 PTF のリストを表示するには、http://www-912.ibm.com/s_dir/sline003.NSF/GroupPTFs?OpenView&view を参照して、該当するグループ PTF の番号をクリックしてください。
4. 最新の「OnDemand Read This First」資料を、Web ページ <http://www.ibm.com/software/data/ondemand/400/support.html> から入手します。これは、バージョン 7 リリース 1 の他の資料と一緒に「資料」ページで見つけることができます。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。
5. ODWEK の前提条件を確認します。15 ページの『第 3 章 HTTP サーバーのインストールおよび構成』を参照してください。
6. OnDemand ソフトウェアを IBM i サーバーにインストールします。16 ページの『IBM i でのインストール』を参照してください。
7. CGI および Java サブレットの場合、ARSWWW.INI ファイルを構成します。17 ページの『ARSWWW.INI ファイルの指定』を参照してください。
8. Apache HTTP Server を構成します。HTTP サーバー構成ファイルの例については、「付録 G: 173 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。
9. 必要に応じて、QONDADM および QRDARS400 権限リストを更新します。第 3 章の 15 ページの『その他の要件』を参照してください。
10. ワークステーション・ブラウザをセットアップします。これを行うには、以下のタスクを実行します。
 - a. 該当するビューアーのプラグイン・ファイルをダウンロードしてインストールします。57 ページの『第 5 章 Web ビューアーのインストール』を参照してください。
 - b. 最新バージョンの Line Data Viewer アプレットを使用するには、<http://www.java.com> から最新の Java ランタイム環境プラグインをダウンロードしてインストールする必要があります。

- c. ブラウザーがすべての Cookie を受け入れることを確認します。「ツール」>「インターネット オプション」に続いて、「プライバシー」タブを選択します。
- d. Internet Explorer で UTF-8 が選択されていることを確認します。「ツール」>「インターネット オプション」に続いて、「詳細設定」タブを選択し、「常に UTF-8 として URL を送信する」が選択されていることを確認します。
- e. Java ランタイム環境が活動状態であることを確認します。「ツール」>「インターネット オプション」に続いて、「詳細設定」タブを選択し、「Java (Sun)」セクション以下を調べます。「<applet> に Java n vx.y.x を使用 (再起動が必要)」が選択されていることを確認します。パネルに表示されるバージョンは、現在ブラウザにインストールされているバージョンを反映します。

第 3 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。

ODWEK ソフトウェアは、現行バージョンの IBM HTTP Server が稼働している IBM i システム上にインストールする必要があります。また、Java サブレットの使用を計画している場合は、現行バージョンの IBM i Web Application Server (WebSphere) が稼働していることを確認してください。

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 i 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 ソフトウェアをインストールする前に、ご使用のシステムが必要なハードウェアおよびソフトウェア要件を満たしていることを確認してください。要件に関して詳しくは、<http://www.ibm.com/support/docview.wss?rs=152&uid=swg27016180> 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、21 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、21 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。

ODWEK は、エンド・ユーザーのブラウザーが、UTF-8 フォーマットを受け入れることを必要とします。Microsoft® Internet Explorer で、「ツール」>「インターネット オプション」に続いて、「詳細設定」タブを選択します。「ブラウズ」の下で、「常に UTF-8 として URL を送信する」を選択します。

AFP2HTML アプレットの使用を計画している場合は、AFP2WEB Transform 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 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、167 ページの『付録 E. AFP から 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、AFP2PDF Transform 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 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、171 ページの『付録 F. AFP から 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。

IBM i でのインストール

ODWEK のセットアップでは、一般に以下のタスクを実行する必要があります。

1. ODWEK をインストールするには、資料「*IBM Content Manager OnDemand for i Common Server 計画とインストール・ガイド*」(SC88-8272)の指示に従ってください。ライセンス・プログラム番号は 5770RD1、製品オプションは 11 です。

重要: 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)」ウィンドウ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 5770RD1 を入力して「製品オプション (Product Option)」に 11 を入力するか、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 を見つけ、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。

2. 推奨: 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、適用してください。
OnDemand バージョン 7 リリース 1 の PTF の完全なリストについては、Informational APAR II14497 を参照してください。この Informational APAR は、SNDPTFORD コマンドを使用して (PTF 番号に II14497 を指定して) オンラインで注文できます。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。
3. IBM プログラム・プロダクト 5770-DG1 (IBM HTTP Server) に関して使用可能な PTF があれば、すべてロードして適用してください。この製品の最新の PTF は、ODWEK を正しく機能させるために必須です。

次のステップ

IBM i システムで、現行バージョンの IBM HTTP Server が稼働していることを確認してください。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、173 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、現行バージョンの IBM i Web Application Server (WebSphere) が稼働していることを確認してください。WebSphere を構成する必要があります。手順については、IBM WebSphere Application Server Documentation Center (Web サイト <http://www.ibm.com/systems/i/software/websphere/>) 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 ソフトウェアのインストールと HTTP サーバーの構成、および WebSphere の構成 (オプション) が完了したら、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@_QUSROND]
HOST=Sxxxxxx.mynetwork.com
PORT=1450
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、49 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キング・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キング・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ています。
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ています。
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 を定義しておく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。

重要: 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、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。デフォルト値は、1 で、文書に対してキャッシュ記憶機構を使用可能にすることを意味します。文書に対するキャッシュ記憶機構の使用を不可に設定するには、0 (ゼロ) 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 が文書をキャッシュに入れる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・スペースの容量を確認してください (22 ページの『CACHESIZE』を参照)。

重要: 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、キャッシュ記憶機構内の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。デフォルト値は 1024 です。

重要: 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、21 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]
CACHE SIZE=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、コマンド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、コマンド文字で区切ります。

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に、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。

言語	値
ブラジル・ポルトガル語	PTB
カナダ・フランス語	FRC
クロアチア語	HRV
チェコ語	CZE
デンマーク語	DAN
英語	ENU
フィンランド語	FIN
フランス語	FRA
ドイツ語	DEU
ギリシャ語	ELL
オランダ語	NLD
ハンガリー語	HUN
イタリア語	ITA
日本語	JPN
韓国語	KOR
ノルウェー語	NOR
ポーランド語	PLK
ロシア語	RUS
中国語 (簡体字)	CHS
スロバキア語	SKY
スロベニア語	SLO
スペイン語	ESP
スウェーデン語	SVE
中国語 (繁体字)	CHT

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
```

重要: 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、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。

重要: 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 サーバーをコンマ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=QUSROND
```

UPDATETIMESTAMP

サーバーに対する各トランザクションの後で、ODWEK にタイム・スタンプを更新させることを指定する場合に使用します。「非アクティブ・タイムアウト」の値がまだ設定されていない場合、その値は、ユーザーの最近のトランザクション以降に経過した時間と同等になります。その目的は、不要な追加のログオンを防ぐことにあります。各トランザクションの後で ODWEK にタイム・スタンプを更新させることを指定する場合は、UPDATETIMESTAMP パラメーターを 1 に設定します。

重要: 同じユーザー ID が複数のブラウザーでアクティブになっている場合、UPDATETIMESTAMP パラメーターは正しく機能しないことがあります。

UPDATETIMESTAMP 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。

UPDATETIMESTAMP はオプションです。これを指定しないか 0 に設定した場合、REPORTSERVERTIMEOUT を 1 に設定しないと、ODWEK はサーバーに対する各トランザクションの後に、タイム・スタンプを更新しません。「非アクティブ・タイムアウト」の値が設定されている場合、その値は、ユーザーの最近のトランザクション以降ではなく、ユーザーのログオン以降に経過した時間と同等になります。その結果、不要な追加のログオンが行われる可能性があります。

「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。UPDATETIMESTAMP パラメーターと REPORTSERVERTIMEOUT パラメーターは似ています。これらはいずれも、1 に設定した場合、サーバーに対する各トランザクションの後でタイム・スタンプを更新します。相違は、「非アクティブ・タイムアウト」の期間が満了になったときに発生します。REPORTSERVERTIMEOUT は、ユーザーのセッションを終了させ、エラーを報告します。UPDATETIMESTAMP は、ユーザーの新規ログオンを行い、エラーを報告しません。これらはいずれも 1 に設定しないと、タイム・スタンプは更新されず、「非アクティブ・タイムアウト」の値はログオン以降に経過した時間と同等になります。

例:

```
[SECURITY]
UPDATETIMESTAMP=1
```

[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 AFP2WEB Transform 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 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、167 ページの『付録 E. AFP から 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、41 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、97 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。167 ページの『付録 E. AFP から 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。

重要: 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/HTMLbin
```

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、それとも実行可能ファイル (EXE) を使用するかを決定します。

重要: IBM i では、ODWEK は実行可能ファイル (EXE) を使用する必要があります。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 AFP2PDF Transform を入手し、これを HTTP 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。AFP2PDF Transform 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、171 ページの『付録 F. AFP から 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、41 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、97 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。171 ページの『付録 F. AFP から 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。

重要: 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/PDFbin
```

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、それとも実行可能ファイル (EXE) を使用するかを決定します。

重要: IBM i では、ODWEK は実行可能ファイル (EXE) を使用する必要があります。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。ブラウザーは、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 (49 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、PNG、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。

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、41 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]
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、42 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、44 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、ご使用のブラウザーに合った 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
```

PNG

PNG 文書の MIME コンテンツ・タイプを指定します。

注:

1. デフォルトでは、PNG 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/png` に設定され、ワークステーションのオペレーティング・システムで PNG 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]
PNG=image/IBM-OnDemand
```

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、175 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ーターを指定することができます。47 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。

重要: 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 API 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。

重要: 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。

重要: 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、32 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。

重要: 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、32 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=0
```

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。

重要: 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チェックしてください。詳細については、32 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

ブラウザーに送信される 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、ブラウザーと 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、10 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、1 で、フォルダー記述を表示することを意味します。フォルダー記述を表示しない場合は、0 (ゼロ) を指定してください。このパラメーターが指定されていないか、パラメーターに 1 以外の値が入っている場合には、フォルダー記述は表示されません。

このパラメーターは、ブラウザー・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 (47 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。

重要: 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、32 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 API 関数と Print Document API 関数での、_max_hits パラメーターの値（関数に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 50 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。デフォルト値は 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、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

67 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

67 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

67 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

67 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 1 で、これは、ユーザーが文書リストから文書をサーバー・プリンターに送信できることを意味します。

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。
3. 文書リストから印刷する文書を選択すると、(複数の文書を選択した場合であっても) 選択した最初の文書だけが印刷されます。

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。デフォルト値は P で、これは、ユーザーがサーバー印刷装置を選択できることを意味します。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 (175 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、177 ページの『Document Hit List』 API 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。

重要: 保管場所を表示するには、OnDemand フォルダで 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 1 で、これは注釈が表示できることを意味します。

重要: 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:

- 以下のパラメーターの有効範囲はグローバルです。また、DEFAULT BROWSER セクションの中でだけ指定できます。(これらのパラメーターは、別のブラウザ・セクションに指定した場合には無視されます。)
 - ODApplet.jre.path.IE
 - ODApplet.jre.path.NN
 - ODApplet.jre.version
 - ODApplet.version
- 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。

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。40 ページの『[DEFAULT BROWSER]』を参照してください。

例:

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

```
[Firefox/3.5.1]
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。

重要: 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。

TRACE

ODWEK がトレース・ファイルにメッセージおよびその他のプログラム情報を書き込めるようにします。(トレース・ファイルの名前は ARSWWW.TRACE 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。トレース・レベルを指定する場合、以下のいずれかの値を使用します。

- 0 トレースなし
- 1 エラーのみ
- 2 エラーおよび警告
- 3 エラー、警告、および通知
- 4 すべて

TRACEDIR

TRACE パラメーターを使用してトレースが有効になっている場合に、ODWEK が ARSWWW.TRACE ファイルを書き込む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、トレースが有効な場合、ODWEK はトレース・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]
;Trace:None=0, Error=1, Error+Warn=2, Err+Warn+Info=3, All=4
Trace=4
TraceDir=/QIBM/UserData/OnDemand/www/logs
```

サンプルの ARSWWW.INI ファイル

デフォルトの QUSROND インスタンスのサンプル・インスタンス構成を示します。太字で示される項目は、変更または追加が必要な構成行です。また、場合によっては、隣に括弧で囲んだ太字のイタリック体で注が併記されています。コメント行の行頭にはセミコロンが付いています。このファイルで指定するディレクトリーは、必ず既存のものでなければなりません。ディレクトリーが存在しない場合、ODWEK は失敗します。

```
=====
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; Server Configuration      ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(Note: DEBUG should be turned off unless needed for problem determination.
This will GREATLY affect performance and should not be turned on unless needed.
To turn DEBUG on, the following three lines should be uncommented. Remember to
recomment the lines once problem determination is completed.)
[DEBUG]
;Trace:None=0, Error=1, Error+Warn=2, Err+Warn+Info=3, All=4
;Trace=4
;TraceDir=/QIBM/UserData/OnDemand/www/logs

;[@SRV@_<host alias>]
;HOST=<host name>
;PORT=
;PROTOCOL=

[@SRV@ QUSROND]
HOST= (enter the fully-qualified system name or system TCP/IP address
for the instance to be accessed)
PROTOCOL=0
PORT= (enter the 4-5 digit TCP/IP port address of the instance, for
example port 1450)

[configuration]
CodePage=37 (change to the code page of the instance configured above)
Language=ENU (change to the language code of the instance
configured above)

TemplateDir=/QIBM/UserData/OnDemand/www/SAMPLES
ImageDir=/images
AppletDir=/applet
TempDir=/tmp
(Note: Caching is recommended for performance; however, if the application group
or data changes, the cache files should be deleted.)
```

```

CacheDir=/QIBM/UserData/OnDemand/www/CACHE
CacheSize=1024
CacheMinThreshold=40
CacheMaxThreshold=80
CacheDocs=1
CacheUserIDs=

[security]
ServerAccess=
ReportServerTimeout=0

;[afp2html]
;InstallDir=/QIBM/UserData/OnDemand/www/HTMLbin
;ConfigFile=/QIBM/UserData/OnDemand/www/HTMLbin/afp2html.ini

;[afp2pdf]
;InstallDir=/QIBM/UserData/OnDemand/www/PDFbin
;ConfigFile=/QIBM/UserData/OnDemand/www/PDFbin/afp2pdf.ini
;UseExecutable=0

[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
LINE=application/line
EMAIL=text/html
META=application/unknown
DOC=application/msword
123=application/vnd.lotus-1-2-3
LWP=application/vnd.lotus-wordpro
SAM=application/vnd.lotus-wordpro
PRZ=application/vnd.lotus-freelance
XLS=application/vnd.ms-excel
PPS=application/vnd.ms-powerpoint
PPT=application/vnd.ms-powerpoint
HTML=application/html
HTM=application/htm
(Note: Additional mimetypes may need to be added to handle other
document types. The browser must know what application to call to display
the document.)

[attachment images]
TXT=userText.gif
BMP=userBitMap.gif
GIF=userGIF.gif

[no html]
Begin=[
End=]
Separator=^

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

[default browser]
FOLDERDESC=1      (Set to '1' if you want the folder description
to be displayed)
;AfpViewing=[ascii,html,native,pdf,plugin,xenos]
AfpViewing=plugin (This requires the afplgus.exe or afplgin.exe be
installed on your workstation)

```

```

;LineViewing=[ascii,applet,native]
LineViewing=applet (The Line Data Applet version to be used is shown below on
the ODApplet.version configuration item)
;EmailViewing=[html,native]
EmailViewing=html
NoLinks=1
ViewNotes=1
AddNotes=1
ServerPrint=1
ServerPrinters=P
AutoDocRetrieval=1
MaxHits=200
ShowDocLocation=1
EncryptCookies=0
EncryptURL=0
ODApplet.version=2 (Use version two of the Line Data Viewer Applet)
ODApplet.jre.path.IE=http://www.java.com/en/download/windows_automatic.jsp
(Where to find the Java Runtime Environment (JRE), if not already installed)
ODApplet.jre.version=1.4 (Minimum JRE version required)

```

(The sections below allow you to override the default configuration options for the browser types shown below.)

```

[IE]
NoLinks=0
AddExtension=1
AddFieldsToDocid=1 (may need to be set to 0 to retrieve linedata (*SCS) docs)
AutoDocRetrieval=0
ViewNotes=1
AddNotes=1

```

```

[Mozilla/4.0 (compatible; MSIE 4.0; Windows 95)]
=====

```

次のステップ

ODWEK ソフトウェアのインストールと ARSWWW.INI ファイルの構成が完了したら、次に、サンプル・アプリケーションを構成します。53 ページの『第 4 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 4 章 サンプル・アプリケーションの構成

OnDemand Web Enablement Kit 環境のセットアップでは、一般に、以下のタスクを実行する必要があります。

1. IBM ソフトウェア・サポートに最新の OnDemand 用 PTF の有無を問い合わせます。最新の PTF のリストは、Informational APAR の番号 II14497 で検索できます。
2. IBM ソフトウェア・サポートに最新の IBM i HTTP Server 用グループ PTF の有無を問い合わせます。この HTTP Server のプロダクト番号は 5770-DG1 です。最新の HTTP Sever 用 PTF のリストを表示するには、http://www-912.ibm.com/s_dir/sline003.NSF/GroupPTFs?OpenView&view を参照して該当するグループ PTF の番号をクリックしてください。
3. IBM サポートに最新のデータベース用グループ PTF の有無を問い合わせます。最新の DB2 用 PTF のリストを表示するには、http://www-912.ibm.com/s_dir/sline003.NSF/GroupPTFs?OpenView&view を参照して、該当するグループ PTF の番号をクリックしてください。
4. 最新の「OnDemand Read This First」資料を、Web ページ <http://www.ibm.com/software/data/ondemand/400/support.html> から入手します。これは、バージョン 7 リリース 1 の他の資料と一緒に「資料」ページで見つけることができます。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。
5. ODWEK の前提条件を確認します。15 ページの『第 3 章 HTTP サーバーのインストールおよび構成』を参照してください。
6. OnDemand ソフトウェアを IBM i サーバーにインストールします。16 ページの『IBM i でのインストール』を参照してください。
7. CGI および Java サブレットの場合、ARSWWW.INI ファイルを構成します。17 ページの『ARSWWW.INI ファイルの指定』を参照してください。
8. Apache HTTP Server を構成します。HTTP サーバー構成ファイルの例については、「付録 G: 173 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。
9. 必要に応じて、QONDADM および QRDARS400 権限リストを更新します。第 3 章の 15 ページの『その他の要件』を参照してください。
10. ワークステーション・ブラウザをセットアップします。これを行うには、以下のタスクを実行します。
 - a. 該当するビューアーのプラグイン・ファイルをダウンロードしてインストールします。57 ページの『第 5 章 Web ビューアーのインストール』を参照してください。
 - b. 最新バージョンの Line Data Viewer アプレットを使用するには、<http://www.java.com> から最新の Java ランタイム環境プラグインをダウンロードしてインストールする必要があります。

- c. ブラウザーがすべての Cookie を受け入れることを確認します。「ツール」>「インターネット オプション」に続いて、「プライバシー」タブを選択します。
- d. Internet Explorer で UTF-8 が選択されていることを確認します。「ツール」>「インターネット オプション」に続いて、「詳細設定」タブを選択し、「常に UTF-8 として URL を送信する」が選択されていることを確認します。
- e. Java ランタイム環境が活動状態であることを確認します。「ツール」>「インターネット オプション」に続いて、「詳細設定」タブを選択し、「Java (Sun)」セクション以下を調べます。「<applet> に Java n vx.y.x を使用 (再起動が必要)」が選択されていることを確認します。パネルに表示されるバージョンは、現在ブラウザーにインストールされているバージョンを反映します。

重要: Java API を使用して ODWEK を制御している場合、システム環境のセットアップと、ODWEK アプリケーションの実行について詳しくは、113 ページの『付録 D. Java API プログラミング・ガイド』を参照してください。この章では、ODWEK に付属しているサンプル・アプリケーションを、CGI プログラムや Java サブレットに合わせて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 ライブラリー・サーバーに定義されます。ユーザーがサーバーにログオンすると、ODWEK はその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。「LOGON.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。
- 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。ユーザーに対して、特定のフォルダーに関する検索基準が示されます。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、アプリケーションにコーディングされます。CREDIT.HTM のカスタマイズについては、55 ページの『CREDIT.HTM』を参照してください。
- FCREDIT.HTM。この CREDIT アプリケーションのバージョンでは、HTML フレームの使用をデモンストレーションします。

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。CREDIT.HTM サンプル・アプリケーションのカスタマイズに関する、詳細な説明があります。この説明をガイドとして使用して、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。

重要: IBM では、サンプル・アプリケーションを変更するほか、TEMPLATE.HTM ファイルをお客様の組織に合わせて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、56 ページの『TEMPLATE.HTM』を参照してください。

LOGON.HTM

1. インストール・ディレクトリーから HTTP サーバーの文書ルート・ディレクトリー (例えば、/www/HttpInstanceName/htdocs) 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. サンプルの場合、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="/od/odservlet">
```

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、必要に応じて変更します。
4. ステートメント (行 13) 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。CGI の場合は、「/scripts/arswww.cgi」で置き換えます。
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、フォルダー表示フィールドの名前を含む、コンマで区切られたストリングです。
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 QUSROND 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が許可されてい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 から 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 から 43 をコピーして、その他の各フォルダー検索フィールドについてステップ 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。TEMPLATE.HTM を、ユーザー定義の内容を含むコピーで置き換えてください。ただし、テンプレート・ファイルには、HTML コメント行 <!--A0I#Marker--> を含める必要があります。この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。テンプレート・ファイルは、デフォルトでは、ARSWWW.INI ファイルの TEMPLATEDIR パラメーターで指定されたディレクトリーにあります。詳しくは、27 ページの『TEMPLATEDIR』を参照してください。

次のステップ

サンプル・アプリケーションの構成が完了したら、57 ページの『第 5 章 Web ビューアーのインストール』を参照してください。

第 5 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。

- 行データ文書を表示するには、Line Data アプレットを使用する必要があります。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設定した後、行データ文書の表示を選択すると、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 (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) 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ことを確認してください。
- 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 ビューアーが使用できます。
 - 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 サーバーに AFP2WEB Transform のインストールと構成を行い、ARSWWW.INI ファイル (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) を構成する必要があります。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設定した後、ユーザーが AFP 文書の表示を選択すると、このアプレットがワークステーション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 サーバーに AFP2PDF Transform のインストールと構成を行い、ARSWWW.INI ファイル (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) を構成する必要があります。管理者がこの Transform を使用可能に設定した後、ユーザーが AFP 文書の表示を選択すると、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ています。ユーザーは、ワークステーションに Adobe Acrobat ビューアーをインストールしておく必要があります。

ARSWWW.INI ファイル (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) 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ことを確認してください。

- BMP、GIF、JPEG、PCX、PNG、および TIFF 文書を表示するには、ユーザーは、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ます。それ以外の場合は、これらのタイプの文書を処理できる他のビューアーを使用する必要があります。(例えば、ほとんどのブラウザーに、GIF および JPEG ファイルを表示できる組み込みビューアーが含まれています。) 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 (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) 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ことを確認してください。

注: 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。

OnDemand に 1 つのユーザー ID を定義して、複数のユーザーがその OnDemand にログオンする一方で、各ユーザーが自分自身の情報にのみアクセスできるようにしたい場合には、システムを次のように構成する必要があります。

1. Web アプリケーションの一環として、ログオン検証プロセスを提供します。
2. ログオン検証プロセスは、照会を OnDemand に送信する前に行う必要があります。
3. ログオン成功の結果を使用して、OnDemand にアカウント番号を提供します。
4. ODWEK API を使用して、OnDemand サーバーに SQL 照会を送信します。この SQL 照会には、特定のアカウント番号が含まれていなければなりません。

例えば、ユーザーが自分の会社の「ウェルカム」ホーム・ページを開いたとします。アカウント情報にアクセスするには、ユーザー ID と PIN を入力する必要があります。これらの値は、OnDemand ではなく、会社の Web アプリケーションで検証されます。ログオンに成功したユーザーに対し、Web アプリケーションはアカウント・サマリー・ページを表示します。このページには、アカウント・ステートメントを表示するリンクが掲載されています。「アカウント・ステートメントを見る」をクリックすると、Web アプリケーションにより ODWEK API が呼び出されます。この API には、ログオン・プロセスから派生したアカウント番号を含む SQL 照会も含まれます。API は、OnDemand サーバーに、そのサーバーに対する ODWEK 接続用に作成したユーザー ID とパスワードを使用してログオンし、ユーザーのステートメントを検索して取り出し、その文書をブラウザに送り返します。ブラウザは、ステートメントに含まれているデータのタイプに該当するビューアーを起動します。

要件

ビューアーの要件に関しては、<http://www.ibm.com/support/docview.wss?rs=152&uid=swg27016180> を参照してください。

インストール

重要: 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、60 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。

- 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 18 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。

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。

重要: 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。

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。

重要: 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 62 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、63 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、64 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
```

重要: 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、61 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、61 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。

重要: 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。

重要: 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 レポートの表示

AFP Web Viewer のインストール・ディレクトリーにある FTDPORT2.INI ファイルには、AFP レポートの表示方法に影響を与える、変更可能なパラメーターが含まれています。このセクションでは、これらのパラメーターとその値について説明します。

- 罫線と行

AFP レポートを表示した際に、罫線や行が正しく表示されない場合、表示ドライバーの相違に問題がある可能性があります。別の方法を使用して罫線を表示してください。FTDPORT2.INI ファイルの Misc セクションで、次の行を探します。

```
RuleFix=FALSE
```

この行を、次のように変更します。

```
RuleFix=TRUE
```

- テキスト精度

フォントが正しく置換されず、テキストの位置合わせが正しく行われな場合、特に、「テキスト精度」パラメーターが「文字」に設定されている場合には、レポートが 240 画素メトリックではなく 300 画素メトリックで作成されていることが原因である可能性があります。240Fidelity=FALSE を指定している場合、レポートは 300 画素メトリックを使用して表示されます。240Fidelity=TRUE を指定している場合、レポートは 240 画素メトリックを使用して表示されます。デフォルトは、240 画素メトリックです。

- プリント・ダイアログ・ボックスのデフォルト

プリント・ダイアログ・ボックスが表示される場合、デフォルトでは、レポートの現行ページが印刷されます。FTDPORT2.INI ファイルの設定セクションで PrintAllPages=TRUE を指定すると、このデフォルトを、レポートのすべてのページが印刷されるように変更できます。

- ユーザー定義ページ・サイズ

標準以外のページ・サイズを含むレポートを表示する場合に、2 つのページ・サイズを定義できます。これら 2 つのユーザー定義ページ・サイズは、レポートを表示する際に選択可能な他のページ・サイズのリストに追加されます。2 つのページ・サイズを定義する場合、FTDPORT2.INI ファイルで次の 2 つの行に変更を加えます。

```
PaperSize1=width, length  
PaperSize2=width, length
```

それぞれ、レポート内の各ページの幅と長さを指定します。すべての値は、1/1440 インチの単位でなければなりません。

- ページ・サイズがインチ単位である場合は、その値を 1440 で乗算します。
- ページ・サイズがミリメートル単位である場合、その値を 56.7 で乗算し、結果を最も近い整数に丸めます。

PaperSize1 および PaperSize2 に値を指定しない場合、標準以外のページ・サイズを含むレポートのデフォルトのページ・サイズは、8.5x11 インチになります。

- True Type フォント

True Type フォントを使用するレポートを表示したい場合には、次の手順を実行します。

1. ワークステーションに Adobe Type Manager (ATM) がインストールされている場合には、それを使用不可に設定するか削除してください。Type 1 フォントがインストールされている場合には、それらを削除する必要があります。
2. FTDPOR2.INI ファイルの Misc セクションに、TTONLY=TRUE という行を追加します。

オーバーレイの表示

AFP データ・ストリームの標準 OnDemand エンド・ユーザーのクライアント・ビューがオーバーレイを表示し、ODWEK AFP Web Viewer がオーバーレイを表示しない場合は、AFP Web Viewer でオーバーレイ資源が検出できなかった可能性があります。

オーバーレイが表示されるように AFP Web Viewer を構成するには、FTDPOR2.INI ファイル内で資源ディレクトリーを指定します。ASCII テキスト・エディター (Windows のメモ帳など) を使用してファイルを開き、[Preferences] の下の ResourceDataPath という項目を見つけてください。次に例を示します。

```
[Preferences]
DefaultView=DEFAULT
ViewDataPath=C:\Program Files\IBM\OnDemand AFP Web Viewer\Data
ResourceDataPath=C:\Program Files\IBM\OnDemand AFP Web Viewer\Resource
FontDataPath=C:\Program Files\IBM\OnDemand AFP Web Viewer\Font
```

OnDemand クライアントに使用する ResourceDataPath 項目は、AFP Web Viewer に使用する同一の項目と一致する必要があります。OnDemand クライアントと AFP Web Viewer の両方が、FTDPOR2.INI ファイルを保持する必要があります。

重要: 外部オーバーレイ資源は、AFP 文書と一緒にダウンロードされません。外部の (AFP 文書と同じファイルに保管されていない) 資源は、AFP 文書と一緒にダウンロードする必要があります。外部の資源は、ResourceDataPath パラメーターで指定したディレクトリーに保管する必要があります。

AFP Web Viewer は、ResourceDataPath で指定した資源ディレクトリーにオーバーレイをダウンロードしません。そのため、他の何らかの方法でこの資源をクライアント・ワークステーションにダウンロードできない場合は、資源を含むように AFP データ・ストリームを変更し、AFP 文書と AFP 資源が同じファイルに入るようにする必要があります。

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。

重要: 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

ーシ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

ODLineDataViewer2.jar は、Java 行データ・ビューアーです。

Java 行データ・ビューアーの要件については、<http://www.ibm.com/support/docview.wss?rs=152&uid=swg27016180> を参照してください。

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 Internet Explorer ユーザーのための Java プラグイン・インストール・ファイルのロケーションは、CGI および Java サブレットの場合、ARSWWW.INI ファイルの追加パラメーターにより決定されます。

68 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.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になるかのプレビューは、 http://java.sun.com/getjava/install-windows.html を参照してください。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>	Internet Explorer の場合。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 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。
ODApplet.jre.version	<version>	Internet Explorer の場合。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 (1.4 または 1.4.1_01 など) を入手してください。

| 以下の例は、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。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およびインストールするようプロンプトが出されます。

```
| [DEFAULT_BROWSER]  
| ODApplet.jre.path.IE=http://java.sun.com/getjava/installer.html  
| 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た後に、HTTP サーバーの構成、Web アプ
| リケーション・サーバーの構成、 ARSWWW.INI ファイルの検証、サンプル・アプ
| リケーションの構成、Web ビューアーのインストールを実行したら、ODWEK を使
| 用し始める前にインストール済み環境を検証する必要があります。検証手順につい
| ては、71 ページの『第 6 章 インストールの検証』を参照してください。

第 6 章 インストールの検証

この段階までに、ODWEK の基本インストールのすべてのステップを完了している必要があります。

OnDemand ライブラリー・サーバーにログオンし、フォルダーをオープンすることにより、ODWEK が正しくインストールされているかどうかを検証できます。 CGI プログラムを使用している場合、71 ページの『CGI プログラムの検証』のセクションを参照してください。 Java サブレットを使用している場合、72 ページの『サブレットの検証』のセクションを参照してください。

重要: Java API を使用している場合、113 ページの『付録 D. Java API プログラミング・ガイド』を参照して、システムを構成し、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方法の詳細について確認してください。

CGI プログラムの検証

以下のステップを実行することで、インストールを検証できます。

重要: 開始する前に、HTTP サーバーを再始動し、構成ファイルに加えた変更によってシステムを初期化してください。

1. ARSWWW.INI ファイルの [SVR@_default] セクションで、HOST、PORT、および PROTOCOL の各パラメーターを検証します。ARSWWW.INI ファイルのデフォルト・ロケーションは /QIBM/UserData/OnDemand/www です。マルチリングル・サポート機能を使用している場合は、193 ページの『実装』で詳細を参照してください。

重要: ARSWWW.INI ファイルの PORT パラメーターの値は、OnDemand ライブラリー・サーバーが稼働しているポートの番号であり、IBM HTTP サーバーがクライアントからの要求を listen するポートの番号ではありません。

2. クライアント (ブラウザ) を起動します。
3. ブラウザーのアドレス行で、OnDemand ライブラリー・サーバー、HTTP ポート、およびログオン関数を含む URL を入力します。次に例を示します。

```
http://odserver1.xyz.com:80/logon.htm
```

ここで、*odserver1.xyz.com* は ARSWWW.INI ファイル内の HOST パラメーターの値、80 は HTTP ポートであり、*logon.htm* は ODWEK が呼び出す関数を指定しています。この例では、ODWEK は指定された OnDemand ライブラリー・サーバーにログオンするログオン関数を呼び出します。(logon.htm ファイルは、ODWEK に付属しているサンプル・アプリケーションの 1 つです。サンプル・アプリケーションのデプロイ方法については、53 ページの『第 4 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。)

4. システムが正しく構成されていれば、ODWEK はログオン画面を表示します。

5. ログオン画面が表示されない場合は、73 ページの『トラブルシューティング』を参照してください。
6. ログオン画面で、OnDemand ライブラリー・サーバー上で有効なユーザー ID とパスワードを入力します。「Submit」をクリックすると、「フォルダーのオープン (Open a Folder)」画面に移動します。
7. この時点で、基本インストールは成功しています。ただし、各種の文書を検索して ODWEK に組み込んだすべての変換をテストすることにより、検証プロセスの続行が必要な場合があります。

サーブレットの検証

Web アプリケーション・サーバーを停止および再始動していない場合は、それらの作業を行ってから続行してください。サーブレットの構成の詳細については、<http://www.ibm.com/software/data/ondemand/400/support.html> を表示して「ODWEK」および「WebSphere」を検索してください。

サーブレットが正しく動作することを検証する場合、Web ブラウザーを開始してサーブレットをオープンします。サーブレットのロケーションを指定してください。例えば、<http://server/od/odservlet> (*server* はサーブレットをデプロイするシステムのホスト名、*od* はコンテンツ・ルート、*odservlet* は WebSphere Application Server のサーブレット・マッピング・セット) のように指定します。

「Internet Connection Version x.x.x.x」というテキストを示す Web ページが表示され、引数「_function」が指定されていなかった場合には、デプロイメントは成功しています。

トラブルシューティング

このセクションでは、インストールの検証を試みた場合に生じる共通のエラーについて説明し、これらのエラーに対処できる解決策も示しています。

問題	解決策
ログオン画面が表示されません。	<p>ログオン画面が表示されない場合の主な原因は、logon.htm ファイルのマッピング規則に誤りがあることです。マッピング規則は httpd.conf ファイルに指定されています。HTTP サーバー情報で、指示を参照してください。</p> <p>マッピング規則が正しい場合は、次を行ってください。</p> <ol style="list-style-type: none">1. サンプル・ディレクトリー /QIBM/UserData/OnDemand/www/samples、またはログオン・スクリプトが置かれている任意のディレクトリーに対する許可が正しいことを検証します。2. logon.htm ファイルに対する許可が正しいことを検証します。3. HTTP サーバーが操作可能であることを検証します。 <p>必要な訂正を行い、HTTP または Web アプリケーション・サーバーを再始動して、再度ログオンしてみてください。</p>
エラー 404。ファイルが見つかりません。	<p>ログオン画面は表示されるが、サーバーにログオンしようとするときエラー 404 のメッセージが表示される場合は、httpd.conf ファイルのファイル・マッピングを検証してください。HTTP サーバーの資料で、指示を参照してください。</p> <p>必要な訂正を行い、HTTP または Web アプリケーション・サーバーを再始動して、再度ログオンしてみてください。</p>
エラー 500。サーバー・エラー。	<p>ログオン画面が正常に表示されたが、ログオン情報の入力後にエラー 500 のメッセージを受け取る場合は、HTTP または Web アプリケーション・サーバーのエラー・ログで詳細情報を調べます。ほとんどの場合、構成エラー（例えば、/QIBM/UserData/OnDemand/www 以下にディレクトリーが見つからないなど）が原因です。</p> <p>必要な訂正を行い、HTTP または Web アプリケーション・サーバーを再始動して、再度ログオンしてみてください。</p>

次のステップ

このセクションでは、ODWEK のインストール完了後に必要とする可能性のある情報へのロードマップを示しています。これには、本書で取り上げているオプション構成タスクのリスト、ODWEK クライアント・ソフトウェアの配布や AFP フォントの処理を担当する管理者を対象とする情報、ODWEK にビジネス・アプリケーションを統合する必要があるプログラマーを対象とする情報、および問題判別に関するいくつかのヒントが含まれます。

以下のセクションでは、オプションの構成タスクに関する情報を提供しています。

- 167 ページの『付録 E. AFP から HTML への変換』
- 171 ページの『付録 F. AFP から PDF への変換』

以下のセクションでは、管理者を対象とする情報を提供しています。

- 64 ページの『AFP フォントのマッピング』
- 59 ページの『AFP Web Viewer』
- 66 ページの『Image Web Viewer』
- 67 ページの『Java 行データ・ビューアー』

以下のセクションでは、プログラマーを対象とする情報を提供しています。

- 75 ページの『付録 A. CGI API リファレンス』
- 109 ページの『付録 B. Java サブレット・リファレンス』
- 111 ページの『付録 C. Java API リファレンス』
- 113 ページの『付録 D. Java API プログラミング・ガイド』
- 175 ページの『付録 H. 非 HTML 出力』

問題判別のツールやヒントに関する情報については、185 ページの『付録 J. 問題判別ツール』を参照してください。

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。

重要: 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となりません。(Access 許可を持っていないと、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
<code>_function=addnote</code>	注釈を追加します。
<code>_server=value</code>	OnDemand サーバーの名前。
<code>_user=value</code>	OnDemand ユーザー ID。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となりません。(Application Group Access 許可を持っていないと、注釈を追加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>_perm=value</code>	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。 <code>_group</code> 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。
<code>_group=groupName</code>	<code>_perm</code> 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。
<code>_copy=value</code>	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 <code>off</code> 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 <code>on</code> 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。
<code>_text=value</code>	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 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、ADDNOTE.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、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、175 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 (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>
<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 CODEPAGE 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、175 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
<code>_function=dochitlist</code>	検索基準に一致する文書のリストを表示します。
<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>	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>folder field name2=value</code>	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。
<code>folder field nameOP=value</code>	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
<code>_display_fields=value[,value,...]</code>	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。
<code>_sort_field=value[,value,...]</code>	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、 <code>_sort_field=Account,Account+Balance,Date</code> 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 がサポートする日付形式の詳細については、「IBM Content Manager OnDemand for i: Common Server 管理ガイド」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、175 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。 <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>_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。)

表 5. Document Hit List 関数 (続き)

名前=値	目的
<code>_cgibin=program</code>	<p>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。</p> <p><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 /QIBM/Proddata/OnDemand/www/bin ディレクトリーから CGI プログラムを取り出します。</p>
<code>_or=value</code>	<p>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。</p>
<code>_logoff=1</code>	<p>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。</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`
`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、175 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 がサポートする日付形式の詳細については、「IBM Content Manager OnDemand for i: Common Server 管理ガイド」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、175 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となりません。(Access 許可を持っていると、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=printDocuments</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となりません。(Application Group Access 許可を持っていると、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
<p><code>_html=value</code></p>	<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>
<p><code>_nohtml=value</code></p>	<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、175 ページの『付録 H. 非 HTML 出力』を参照してください。</p>
<p><code>_docids=documentIDList</code></p>	<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>重要: 文書 ID の数が 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。</p>
<p><code>_port=value</code></p>	<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>
<p><code>_codepage=value</code></p>	<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>重要: 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=printDocuments  
&_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・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、175 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
<p>_afp=value</p>	<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>重要: _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、32 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>
<p>_email=value</p>	<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>重要: _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、32 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。 重要: <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、32 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 がサポートする日付形式の詳細については、「 <i>IBM Content Manager OnDemand for i: Common Server 管理ガイド</i> 」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、175 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 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、UPDATE.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、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、175 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、175 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 サブレット・リファレンス

Java サブレットは、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ブレットを使用します。

Java サブレットは、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、75 ページの『付録 A. CGI API リファレンス』を参照してください。

| Java サブレット構成の例については、<http://www-306.ibm.com/software/data/ondemand/400/support.html> を表示して「ODWEK」および「WebSphere」を検索して参照してください。OnDemand の新しいリリースにアップグレードする場合は、サンプル文書に記述されているとおりに EAR ファイルを更新して CLASS と JAR ファイルが更新されるようにする必要があります。

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。

資料を表示するには、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/api` 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しておく必要があります。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。
- 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

このセクションに示す例のソース・コードの多くは、ご使用の IBM i サーバー上の /QIBM/ProdData/OnDemand/www/api ディレクトリー内の ODApiSamples.zip という名前の zip ファイルに含まれています。

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.ODApplication

このクラスは、OnDemand アプリケーションを表します。ODApplication オブジェクトのインスタンスでは、アプリケーション開発者が、OnDemand アプリケーションに指定された情報にアクセスできます。

com.ibm.edms.od.ODApplicationGroup

このクラスは、OnDemand アプリケーション・グループを表します。ODApplicationGroup オブジェクトのインスタンスでは、アプリケーション開発者が、OnDemand アプリケーション・グループに指定された情報にアクセスできます。ODApplicationGroup オブジェクトのインスタンスは、ODServer.openFolder() メソッドを使用することで、ODServer オブジェクトのインスタンスから生成されます。

com.ibm.edms.od.ODApplicationGroupFields

このクラスは、OnDemand アプリケーション・グループ・フィールドを表します。ここには、アプリケーション・グループ・フィールドの情報が含まれます。

警告: すべての ODWEK オブジェクトに対するアクセスは、単一スレッド環境でのみ行ってください。

com.ibm.edms.od.ODCabinet

このクラスは、OnDemand フォルダのコレクションを表します。キャビネットは、管理者が OnDemand サーバー上に定義および管理します。

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.ODConfig

ODConfig Java オブジェクトは、システム・パラメーターの構成に使用する優先方式です。詳細については、115 ページの『システム・パラメーターの構成』を参照してください。

com.ibm.edms.od.ODConstant

ODConstant は、パブリック・インターフェースです。

com.ibm.edms.od.ODCriteria

OnDemand フォルダ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.ODHitProperties

このクラスを使用して、ヒットの OnDemand 内部プロパティ値を取得します。

com.ibm.edms.od.ODHold

このクラスは、OnDemand 保留の定義を表します。このオブジェクトは、(ODServer.getHolds()) への呼び出しが正常に終了した場合に戻されます。

重要: すべての ODWEK オブジェクトへのアクセスは単一スレッド環境で行う必要があります。

com.ibm.edms.od.ODLogicalView

このクラスは、OnDemand 論理ビューを表します。

com.ibm.edms.od.ODNamedQuery

このクラスは、OnDemand 名前付き照会を表します。名前付き照会の詳細を含み、既存の名前付き照会を検索するシステムを有効にし、新規の名前付き照会を OnDemand サーバーに保管します。

com.ibm.edms.od.ODNamedQueryCriteria

このクラスは、OnDemand 名前付き照会の基準を表します。名前付き照会に保管される検索基準の詳細を含みます。

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。

com.ibm.edms.od.ODUser

このクラスは、OnDemand ユーザーを表します。このクラスから、OnDemand サーバーに保管される、住所や電話番号などのユーザー情報を収集できます。

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.props ファイルをユーザー定義のランタイム・ディレクトリーにコピーします。
2. 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。119 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。

The Java API 共有ライブラリー (サービス・プログラム) は ARS3WAPI64 であり、ライブラリー内にあります。

システム・パラメーターの構成

ODConfig Java オブジェクトは、システム・パラメーターの構成に使用する優先方式です。

動作中の ODServer インスタンスには、8 つのシステム・パラメーターが必要です。ODConfig デフォルト・コンストラクターを使用して、以下のように、これらのパラメーターをそのデフォルト値に設定します。

```

<pre>
    try{
        ODConfig cfg = new ODConfig();
        ODServer srvr = new ODServer(cfg);
        srvr.initialize("MyCustomApp");
        cfg.printConfig();
    }
    catch(ODException e) {
        System.out.println("Exception " + e);
    }
}
</pre>

```

このサンプル・コードは、次のデフォルト・パラメーターを構成します。

```

<pre>
AfpViewing    PLUGIN
LineViewing   APPLET
MaxHits       200
AppletDir     /applets
Language      ENU
TempDir       The temp path as defined by the Java System.getProperty("java.io.tmpdir") method.
TraceDir      The temp path as defined by the Java System.getProperty("java.io.tmpdir") method.
TraceLevel    0
</pre>

```

これらのパラメーターについては、17 ページの『ARSWWW.INI ファイルの指定』を参照してください。

TraceLevel パラメーターは、ODWEK 問題判別に使用されるトレース・レベルを反映します。トレースの詳細については、117 ページの『トレース』を参照してください。

これらのパラメーターは、次のサンプル・コードを使用して明示的に設定することもできます。このサンプルでは、異なる ODConfig コンストラクターを使用しています。

```

<pre>
    try{
        ODConfig cfg = new ODConfig(ODConstant.PLUGIN, //AfpViewer
                                    ODConstant.APPLET, //LineViewer
                                    null,                //MetaViewer
                                    500,                //MaxHits
                                    "c:\\applets",       //AppletDir
                                    "ENU",              //Language
                                    "c:\\temp",          //TempDir
                                    "c:\\temp\\trace",    //TraceDir
                                    1);                 //TraceLevel

        ODServer srvr = new ODServer(cfg);
        srvr.initialize("MyCustomApp");
        cfg.printConfig();
    }
    catch(ODException e){
        System.out.println("Exception " + e);
    }
}
</pre>

```

このコンストラクターでは、デフォルト値としてゼロまたはヌル値のパラメーターを設定しています。

重要: このオブジェクトには、構成中以外に、パラメーターを設定するメソッドはありません。このオブジェクトは、構成後には変更できません。

ARSWWW.PROPS ファイルのサンプル

以下のサンプルは、Java API の場合に ARSWWW.INI ファイルの代わりに使用される ARSWWW.PROPS ファイルを示しています。

```
#AfpViewing=[ascii,html,native,pdf,plugin]
AfpViewing=plugin

#LineViewing=[ascii,applet,native]
LineViewing=applet

MaxHits=50
Language=ENU
TempDir=/QIBM/UserData/OnDemand/www/tmp

#Trace:None=0, Error=1, Error+Warn=2, Err+Warn+Info=3,All=4
Trace=4
TraceDir=/QIBM/UserData/OnDemand/www/logs

AppletVersion=2
AppletJREVersion=
AppletJREDownloadPathIE=
AppletJREDownloadPathNN=
AppletJREDownloadPathNN=

AppletDir=/QIBM/UserData/OnDemand/www/applets
AppletCacheDir=
AppletImageDir=

AppletFitWidth=
AppletServerSearch=
CompressMemSize=
DocMemSize=
ShowAllFolderCriteria=
AddFieldsToDocid=1
ServerAccessList=

#Afp2HtmlInstallDir=/QIBM/UserData/OnDemand/www/binhtml
#Afp2HtmlConfigFile=/QIBM/UserData/OnDemand/www/binhtml/afp2web.ini
#Afp2PdfInstallDir=/QIBM/UserData/OnDemand/www/binpdf
#Afp2PdfConfigFile=/QIBM/UserData/OnDemand/www/binpdf/afp2pdf.ini
```

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

ODWEK トレースは、ARSWWW.PROPS ファイルに指定されたトレース・ディレクトリ内にある ARSWWW.TRACE ファイルにトレース・ステートメントを書き込みます。ODWEK トレースは、問題判別を支援することを目的としています。ODWEK トレースが有効になっている場合、トレースの形式を問わず、パフォーマンスの低下が生じます。

定期的に、手動でトレース・ファイルを消去する必要があります。トレース・ファイルは循環せず、十分なファイル・スペースを必要とします。

ODWEK トレースを有効にするには、次のステップを実行します。

1. ODCConfig デフォルト・コンストラクターを変更します。115 ページの『システム・パラメーターの構成』の手順に従ってください。

2. 以下を反映するように、次のデバッグ・スタンザを変更します。

```
;Trace=None=0, Error=1, Error+Warn=2, Err+Warn+Info=3, All=4
Trace=4
TraceDir=/QIBM/UserData/OnDemand/www/logs
```

ヒント: トレース・パラメーターを使用すると、トレースを各種のレベルに設定できます。 ODWEK の問題をトラブルシューティングする際には、トレース・レベルを最高レベルに設定してください (IBM ソフトウェア・サポートによる指定がある場合を除く)。一方、定常状態にある ODWEK アプリケーションをモニターする場合には、トレースを比較的低いレベルに設定できます。例えば、Trace=1 に設定すると、オーバーヘッドを最小限に抑えて、エラー状態のみ警報を出します。 ODWEK の旧リリースの ARSWWW.PROPS ファイルを使用している場合は、古いデバッグ・セクションを削除してください。複数のデバッグ・セクションがあると、ODWEK トレースが行えない場合があります。

3. TraceDir パラメーターが参照するディレクトリーに、arswww.trace ファイルが作成されます。
4. ARSWWW.PROPS ファイルに加えた変更を有効にするため、ODWEK アプリケーションを再始動する必要があります。
5. 問題をトラブルシューティングするためにトレースを有効にする場合、トレースが有効になってから問題を再現して、ARSWWW.TRACE ファイルを IBM サポートに送付してください。

ARSWWW.PROPS 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 arswwww.trace ファイルにトレース情報を書き込みます。

```
TraceDir=/QIBM/UserData/OnDemand/www/logs
```

重要: 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。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、185 ページの表 16 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、117 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 ( ODEException e )
{
    System.out.println( "ODEException: " + 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、111 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. アプリケーションをコンパイルおよび実行する前に、以下のようにエクスポート
をセットアップしておく必要があります。

```
export LIBPATH=/QSYS.LIB/QRDARS.LIB
export PATH=/usr/bin:/your_program_path
export -s CLASSPATH=/QIBM/ProdData/OnDemand/www/api/ODApi.jar:/your_program_path
```

2. 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[ ] )
    {
        :
        :
        :
    }
}
```

3. 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。

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。111 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( "AppName" );

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

char directory = "/QIBM/UserData/OnDemand/www";
int port = Integer.parseInt('1450');

odServer.logon( "LIBSRVR1", "ADMIN", "PASSWD", CONNECT_TYPE_TCPIP, port, directory );
.
.
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、121 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.initialize( "AppName" );

odServer.setServerName("LIBSRVR1");
odServer.setUserId("ADMIN");
odServer.setPassword("PASSWD");

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

char directory = "/QIBM/UserData/OnDemand/www";
int port = Integer.parseInt('1450');

odServer.logon( odServer.getServerName( ),
                odServer.getUserId( ),
                odServer.getPassword( ),
                ODConstant.CONNECT_TYPE_TCPIP,
                port,
                directory );
.
```

```
.  
.  
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、『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、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`
- `getPassword`
- `getPort`
- `getServerName`
- `getUserId`
- `setConnectType`
- `setPassword`
- `setPort`
- `setServerName`
- `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
- ポート
- ユーザー ID
- パスワード
- 構成ディレクトリー (`arswww.props` ファイルのロケーション)

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 < 5 )  
        {
```

```

        System.out.println( "usage: java TcServerMisc <server> <port> <userid> <password> <config dir>" );
        return;
    }

    try
    {
        //-----
        // Set the stage
        //-----

        System.out.println( "Testcase TcServerMisc started." );
        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
        //-----
        ODConfig odConfig = BuildODConfig.build(argv[4] + "/arswww.props");
        if(odConfig == null)
            System.out.println("BuildODConfig Failed.");

        else
        {
            odServer = new ODServer(odConfig );
            odServer.initialize( "TcListFolders.java" );
            odServer.setServerName( argv[0] );
            odServer.setUserId( argv[2] );
            odServer.setPassword( argv[3] );
            odServer.setConnectType(ODConstant.CONNECT_TYPE_TCPIP);
            odServer.setPort(Integer.parseInt(argv[1]));
            System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
            odServer.logon( );

            //-----
            // Test miscellaneous methods
            //-----
            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 );

            System.out.println( "Setting connect type to ODConstant.CONNECT_TYPE_TCPIP..." );
            odServer.setConnectType( ODConstant.CONNECT_TYPE_TCPIP );

            System.out.println( "Setting install directory to /QIBM/ProdData/OnDemand/www ..." );
            odServer.setInstallDir( "/QIBM/ProdData/OnDemand/www" );

            //-----
            // Cleanup
            //-----

            System.out.println( "Logging off..." );

            odServer.logoff( );
            odServer.terminate( );
            System.out.println( "" );

            System.out.println( "-----" );
            System.out.println( "" );
            System.out.println( "Testcase TcServerMisc 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;
    }
}

```

Java API を使用した非デフォルト・ポートへの接続

場合によっては、ODWEK Java API を使用してデフォルト以外のポートにアクセスする必要があります。例えば、OnDemand サーバー上に、デフォルト・ポートを使用するインスタンスと、それ以外のポートを使用するインスタンスの 2 つのインスタンスがある場合が考えられます。システムを正しく構成していないと、Java プログラムの実行時に、「instance2 サーバーへの接続を確立できません。(A connection cannot be established to the instance2 server.)」というエラーを受け取ることになります。

この調整を行うには、ログオンの直前に、Java ソース内で `ODServer.setPort()` メソッドを使用します。次に、サーバー名に、ホスト名を (インスタンスの別名の代わりに) 指定します。メソッドでは、インスタンスを解決するために、`ARSWWW.PROPS` ファイルのその部分を使用しません。

ODConfig の構築

以下の例は、`BuildODConfig` と呼ばれる Java プログラムを参照します。この例は、Java コードに構成パラメーターを設定するプログラムの構築のためのガイドラインになります。

構成パラメーターを設定するための `ODConfig` の使用例:

```

import java.io.FileInputStream;
import java.util.Properties;

import com.ibm.edms.od.ODConfig;

public class BuildODConfig{

    static public ODConfig build(String propsFile){
        ODConfig odConfig = null;
        try{
            FileInputStream fileinput = new FileInputStream(propsFile);
            Properties props = new Properties();
            props.load(fileinput);
            String AfpV = (String)props.get("AfpViewing");
            String LineV = (String) props.get("LineViewing");
            String MetaV = (String) props.get("MetaViewing");
            Long maxHits = Long.parseLong((String)props.get("MaxHits"));
            String appletDir = (String)props.get("AppletDir");
            String lang = (String)props.get("Language");
            String tempDir = (String)props.get("TempDir");
            String traceDir = (String)props.get("TraceDir");
            int traceLvl = Integer.parseInt((String)props.get("Trace"));

```

```

        props.remove("AfpViewing");
        props.remove("LineViewing");
        props.remove("MetaViewing");
        props.remove("MaxHits");
        props.remove("AppletDir");
        props.remove("Language");
        props.remove("TempDir");
        props.remove("TraceDir");
        props.remove("Trace");

        odConfig = new ODConfig(AfpV,
            LineV,
            MetaV,
            maxHits,
            appletDir,
            lang,
            tempDir,
            traceDir,
            traceLvl,
            props);
    }
    catch ( Exception e2 )
    {
        System.out.println( "exception: " + e2 );
        System.out.println( "Cause: " + e2.getCause());
        e2.printStackTrace();
    }
    return odConfig;
}
}

```

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`
- `getApplGroupNames`
- `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`
- `getApplicationGroup`
- `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
- ポート
- ユーザー ID
- パスワード
- フォルダ名
- 構成ディレクトリー (`arswww.props` ファイルのロケーション)

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 TcApplGrp
{
    public static void main ( String argv[ ] )
    {
        ODServer odServer;
        ODFolder odFolder;
        String[ ] appls;
        int j;
        long agid=0;
        char conntype = 'T';
        String directory = "";

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

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "Testcase TcApplGrp started." );
            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( " Get application group by name and display" );
            System.out.println( " Get application group by application group id and display" );
            System.out.println( "" );
            System.out.println( "-----" );
            System.out.println( "" );

            //-----
            // Logon to the specified server
            //-----
            ODConfig odConfig = BuildODConfig.build(argv[5] + "/arswww.props");

            if (odConfig != null)
            {
                odServer = new ODServer(odConfig );
                odServer.initialize( "TcApplGrp.java" );
                System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
                odServer.logon( argv[0], argv[2], argv[3], conntype, Integer.parseInt(argv[1]), directory );

                //-----
                // Open the specified folder
                //-----
                System.out.println( "Opening " + argv[4] + " folder..." );
                odFolder = odServer.openFolder( argv[4] );

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

                System.out.println("\nGet the Application Group by name: " + agname);
                ODApplicationGroup odAG1 = odServer.getApplicationGroup(agname);
                System.out.println("Application Group Name = " + odAG1.getName());
                System.out.println("Application Group Description = " + odAG1.getDescription());
                System.out.println("Application Group ID = " + odAG1.getId());
                System.out.println("Mapped Applications:");
                appls = odAG1.getApplicationNames();
                for (int i = 0; i < appls.length; i++)
                    System.out.println( " " + appls[i] );
                agid = odAG1.getId();
                System.out.println("Application Group ID = " + agid);

                //-----
                // Cleanup
                //-----
                odFolder.close( );
                odServer.logoff( );
                odServer.terminate( );
            }
            else
            {
                System.out.println( "" );
                System.out.println( "Testcase TcApplGrp failed -" );
                System.out.println( " ODConfig could not be initialized. " );
                System.out.println( " Probable cause: " );
                System.out.println( " File arswww.props is not found in directory " + argv[5] );
                System.out.println( "" );
            }
        }

        System.out.println("Re-Logon and attempt to get the Application Group using Application Group ID saved above.");
        odConfig = BuildODConfig.build(argv[5] + "/arswww.props");
    }
}

```

```

if (odConfig != null)
{
    odServer = new ODServer(odConfig );
    odServer.initialize( "TcApplGrp.java" );
    System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
    odServer.logon( argv[0], argv[2], argv[3], comntype, Integer.parseInt(argv[1]), directory );
    System.out.println( "Opening " + argv[4] + " folder..." );
    odFolder = odServer.openFolder( argv[4] );

    System.out.println("\nGet the Application Group for id " + agid);
    ODApplicationGroup odAG = odServer.getApplicationGroup(agid);
    System.out.println("Application Group Name = " + odAG.getName());
    System.out.println("Application Group Description = " + odAG.getDescription());
    System.out.println("Application Group ID = " + odAG.getId());
    System.out.println("Mapped Applications:");
    appls = odAG.getApplicationNames();
    for (int i = 0; i < appls.length; i++)
        System.out.println("    " + appls[i]);

    //-----
    // Cleanup
    //-----
    odFolder.close( );
    odServer.logoff( );
    odServer.terminate( );
    System.out.println( " " );
    System.out.println( "-----" );
    System.out.println( " " );
    System.out.println( "Testcase TcApplGrp completed. - analyze results if required" );
    System.out.println( " " );
}
else
{
    System.out.println( " " );
    System.out.println( "Testcase TcApplGrp failed -" );
    System.out.println( " ODCConfig could not be initialized. " );
    System.out.println( " Probable cause: " );
    System.out.println( " File arswww.props is not found in directory " + argv[5] );
    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`
- `setMaxHits`

- search
- getSearchMessage
- close

この例には、以下の ODCriteria メソッドを示します。

- getName
- setOperator
- 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
- 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
- ポート
- ユーザー ID
- パスワード
- フォルダー名
- 最大ヒット
- 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.props ファイルのロケーション)

制約事項: ヒット回数は、arswww.props 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 server, userid, password, folder, directory;
String crit = "", operator = "", value1 = "", value2 = "";
String header, line1, line2, hit_value, useable_value;
boolean mismatch_detected, use_default_values;
int j, k, opr;
int port, maxHits;

//-----
// If too few parameters, display syntax and get out
//-----
if ( argv.length < 11 && argv.length != 7 )
{
    System.out.println( "usage: java TcSearch <server> <port> <userid> <password> <folder> <MaxHits (-1 for default)> " );
    System.out.println( "          <criteria> <opr> <value1> <value2> <config dir> " );
    System.out.println( "          or, to use default search criteria " );
    System.out.println( "java TcSearch <server> <port> <userid> <password> <folder> <MaxHits (-1 for default)> <config dir> " );
    return;
}

try
{
    //-----
    // Set the stage
    //-----
    System.out.println( "Testcase TcSearch started." );
    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( "  If specified:" );
    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.props 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
    //-----
    use_default_values = argv.length == 7;
    server = argv[0];
    port = Integer.parseInt(argv[1]);
    userid = argv[2];
    password = argv[3];
    folder = argv[4];
    maxHits = Integer.parseInt(argv[5]);
    if ( use_default_values )
        directory = argv[6];
    else
    {
        crit = argv[6];
        operator = argv[7];
        value1 = argv[8];
        value2 = argv[9];
        directory = argv[10];
    }

    ODConfig odConfig = BuildODConfig.build(directory + "/arswww.props");
    if(odConfig == null)
    {
        System.out.println( "" );
        System.out.println( "Testcase TcListFoldersByCrit failed -" );
        System.out.println( "  ODConfig could not be initialized. " );
        System.out.println( "  Probable cause: " );
        System.out.println( "  File arswwww.props is not found in directory " + directory );
        System.out.println( "" );
    }
    else
    {
        odServer = new ODServer(odConfig );
        odServer.initialize( "TcSearch.java" );
        String localdir = "";
        System.out.println( "Logging on to " + server + " server with user " + userid + "..." );
        odServer.logon( server, userid, password, ODConstant.CONNECT_TYPE_TCPIP, port, localdir);

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

        System.out.println( "Name=" + odFolder.getName( ) + " Desc=" + odFolder.getDescription( ) + "" );

        //-----
        // If we are not using the default search values:
        //-----
        if ( !use_default_values )
        {
            //-----
            // Find the requested criteria
            //-----
            System.out.println( "Getting " + crit + " criteria..." );

```

```

odCrit = odFolder.getCriteria( crit );
if ( odCrit == null )
    System.out.println( " *** " + crit + " criteria does not exist - NullPointerException will be reported" );

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

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

//-----
// Set the search values
//-----
if ( opr == ODConstant.OPBetween || opr == ODConstant.OPNotBetween )
{
    odCrit.setSearchValues( value1, value2 );
    System.out.println( " " + odCrit.getName( ) + " " + getOperatorName( opr ) + " " + value1 + " and " + value2 );
}
else
{
    odCrit.setSearchValue( value1 );
    System.out.println( " " + odCrit.getName( ) + " " + getOperatorName( opr ) + " " + value1 );
}
}
//-----
// Set Max Hits limit if specified
//-----
if ( maxHits != -1 )
{
    System.out.println( "Setting MaxHITS to " + maxHits + "..." );
    odFolder.setMaxHits( maxHits );
}
else
    System.out.println( "No MaxHits passed in. Will use the MaxHits in arswww.props/ODConfig, or " );
    System.out.println( " the Folder defined max, which ever is less." );

//-----
// Search the folder
//-----
System.out.println( " Searching " + folder + ( use_default_values ? " using default values " : "" ) + "..." );

long startTime = System.currentTimeMillis();
hits = odFolder.search();
long estimatedTime = System.currentTimeMillis() - startTime;
System.out.println( "Elapsed Search Time: " + estimatedTime + " ms" );
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 TcSearch 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 ODCConstant.OPEqual:
            str = "Equals";
            break;
        case ODCConstant.OPNotEqual:
            str = "Not Equal";
            break;
        case ODCConstant.OPLessThan:
            str = "Less Than";
            break;
        case ODCConstant.OPLessThanEqual:
            str = "Less Than or Equal";
            break;
        case ODCConstant.OPGreaterThan:
            str = "Greater Than";
            break;
        case ODCConstant.OPGreaterThanEqual:
            str = "Greater Than or Equal";
            break;
        case ODCConstant.OPIn:
            str = "In";
            break;
        case ODCConstant.OPNotIn:
            str = "Not In";
            break;
        case ODCConstant.OPLike:
            str = "Like";
            break;
        case ODCConstant.OPNotLike:
            str = "Not Like";
            break;
        case ODCConstant.OPBetween:
            str = "Between";
            break;
        case ODCConstant.OPNotBetween:
            str = "Not Between";
            break;

        default:
            str = "Operator unknown";
            break;
    }

    return str;
}

static String getDocTypeString( char type )
{
    String str;

    switch( type )
    {
        case ODCConstant.FileTypeAFP:
            str = "AFP";
            break;
        case ODCConstant.FileTypeBMP:
            str = "BMP";
            break;
        case ODCConstant.FileTypeEMAIL:
            str = "EMAIL";
            break;
        case ODCConstant.FileTypeGIF:

```

```

        str = "GIF";
        break;
    case ODCConstant.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.FileTypeSCS:
        str = "SCS";
        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。

- getName
- getDescription
- getDisplayOrder
- setAppListGroupForSearchWithSQL
- setMaxHits
- search
- close

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
- 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
- ポート
- ユーザー ID
- パスワード
- フォルダー名
- SQL ストリング
- 最大ヒット
- 構成ディレクトリー (arswww.props ファイルのロケーション)
- 日付 1 (オプション)
- 日付 2 (オプション)
- 日付形式 (オプション)
- アプリケーション・グループ (オプション)

SQL ストリングを使用したフォルダーの検索の例:

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

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

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 8 )
        {
            System.out.println( "usage: java TcSQLSearch <server> <port> <userid> <password> <folder> <SQL> );
            System.out.println( "      <MaxHits (-1 to use default)> <config dir>" );
            System.out.println( "      <opt begin Date> <opt end Date> <opt Date Format> <opt ApplGrpName>" );
            return;
        }
        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "Testcase TCSQLSearch started." );
            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( "  Set the application group (if specified)" );
            System.out.println( "  Search the folder using specified Sql" );
            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.props 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
//-----
ODConfig odConfig = BuildODConfig.build(argv[7] + "/arswww.props");
if(odConfig == null)
    System.out.println("BuildODConfig Failed.");
else
{
    odServer = new ODServer(odConfig );
    odServer.initialize( "TcSQLSearch.java" );
    System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
    String directory = "";
    odServer.logon( argv[0], argv[2], argv[3], ODConstant.CONNECT_TYPE_TCPIP, Integer.parseInt(argv[1]), directory);

    //-----
    // Open the specified folder and find the requested criteria
    //-----
    System.out.println( "Opening " + argv[4] + " folder..." );
    odFolder = odServer.openFolder( argv[4] );
    System.out.println( "Name=" + odFolder.getName( ) + " Desc=" + odFolder.getDescription( ) + "" );
    System.out.println( "Getting " + argv[5] + " criteria..." );

    //-----
    // Search with Application Group limiters if specified
    //-----
    if(argv.length == 11 )
    {
        System.out.println("**Search is limited to Application Group " + argv[9]);
        odFolder.setApplGroupForSearchWithSQL(argv[10]);
    }
    //Set MaxHits
    int maxHits = Integer.parseInt(argv[6]);
    if (maxHits != -1)
    {
        System.out.println("Setting MaxHITS to " + maxHits + ".");
        odFolder.setMaxHits(maxHits);
    }
    else
        System.out.println("No MaxHits passed in. Will use the MaxHits in arswwww.props/ODConfig, or ");
        System.out.println(" the Folder defined max, which ever is less.");

    //-----
    // Search with date limiters if specified
    //-----
    if(argv.length >= 10 ) //call search(sql,date1,date2,date format)
    {
        String date1 = argv[8];
        if(date1.compareTo(" ") != 0) //Only go in here if date was actually spec'd
        {
            System.out.println("Searching " + argv[4] + " with Date Range limits of " + argv[8] + " and " + argv[9]);
            System.out.println("Date format is " + argv[10]);

            hits = odFolder.search(argv[5],
                argv[8], //date1
                (argv.length > 10) ? argv[9] : " ", //date2
                (argv.length >= 11) ? argv[10] : " "); //date format
        }
        else
        {
            //-----
            // Search the folder
            //-----
            System.out.println( " Searching " + argv[4] + "with SQL " + argv[5] + "..." );
            hits = odFolder.search(argv[5] );
        }
    }
    else if(argv.length > 8 && argv.length < 11 ) //call search(sql,date1,date2)
    {
        String date1 = argv[8];
        if(date1.compareTo(" ") != 0) //Only go in here if date was actually spec'd
        {
            System.out.println("Searching " + argv[4] + " with Date Range limits of " + argv[8] + " and " + argv[9]);

            hits = odFolder.search(argv[5],
                argv[8], //date1
                (argv.length > 10) ? "" : argv[9]); //date2
        }
        else //call search(sql)
        {
            //-----
            // Search the folder
            //-----
            System.out.println( " Searching " + argv[4] + "with SQL " + argv[5] + "..." );
            hits = odFolder.search(argv[5] );
        }
    }
    else //call search(sql)
    {
        //-----
        // Search the folder
        //-----
        System.out.println( " Searching " + argv[4] + "with SQL " + argv[5] + "..." );
        hits = odFolder.search(argv[5] );
    }
}

//-----
// Display the hits
//-----

```

```

System.out.println(" Number of Hits Found = " + hits.size());
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 TcSQLSearch 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 ODCConstant.OPBetween:
        str = "Between";
    break;
    case ODCConstant.OPNotBetween:
        str = "Not Between";
    break;
    default:
        str = "Operator unknown";
    break;
    }
    return str;
}
static String getDocTypeString( char type )
{
    String str;

    switch( type )
    {
    case ODCConstant.FileTypeAFP:
        str = "AFP";
    break;
    case ODCConstant.FileTypeBMP:
        str = "BMP";
    break;
    case ODCConstant.FileTypeEMAIL:
        str = "EMAIL";
    break;
    case ODCConstant.FileTypeGIF:
        str = "GIF";
    break;
    case ODCConstant.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.FileTypeSCS:
        str = "SCS";
    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;
}
}

```

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。

- getName
- getCriteria
- search
- close

この例には、以下の ODCriteria メソッドを示します。

- setOperator
- 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
- ポート
- 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.props ファイルのロケーション)

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 TcCancelSearch
{
    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 < 6 )
{
    System.out.println( "usage: java TcCancelSearch <server> <port> <userid> <password> <folder> <config dir>" );
    return;
}
try
{
    //-----
    // Set the stage
    //-----
    System.out.println( "Testcase TcCancelSearch started." );
    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 1970-2005" );
    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 .1 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.props 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
    //-----
    ODConfig odConfig = BuildODConfig.build(argv[5] + "/arswww.props");
    if(odConfig == null)
        System.out.println("BuildODConfig Failed.");
    else
    {
        odServer = new ODServer(odConfig);
        odServer.initialize( "TcCancelSearch.java" );
        char conntype = 'T';
        String directory = "";
        System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
        odServer.logon( argv[0], argv[2], argv[3], conntype, Integer.parseInt(argv[1]), directory);

        //-----
        // Open the specified folder and display its name and description
        //-----
        System.out.println( "Opening " + argv[4] + "..." );
        odFolder = odServer.openFolder( argv[4] );
        odCrit = odFolder.getCriteria( "Report Date" );
        odCrit.setOperator( ODConstant.OPBetween );
        odCrit.setSearchValues( "01/01/1970", "01/01/2005" );

        //-----
        // 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)..." );
        System.out.println( "  Searching " + odFolder.getName() + "..." );
        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 .1 seconds..." );
        ( Thread.currentThread( ) ).sleep( 100 );
        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( "Testcase TcCancelSearch 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 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。

- getName
- getOperator
- getValidOperators
- getType
- getDefaultFmt
- getDisplayFmt
- getDisplayFmtQual
- getMaxEntryChars
- getMaxDisplayChars
- getMinSearchValue
- getMaxSearchValue
- getDBFieldNames
- getDBFieldMask
- getSearchValues
- getFixedValues
- isUpdateable
- isRequired
- isDefaultValueAvailable
- isDefaultValueFixed

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
- getDescription
- getNumApplGroups
- getApplGroupNames
- getNumCriteria
- 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
- ポート
- 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.props ファイルのロケーション)

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;
        String[] value_vec;
        String[] search_values, fixed_values, database_fields;
        Object[] appl_grps;
        int[] valid_oprs;
        int j, k, opr;
        char field_type;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 6 )
        {
            System.out.println( "usage: java TcListCriteria <server> <port> <userid> <password> <folder> <config dir>" );
            return;
        }
        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "Testcase TcListCriteria started." );
            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 and names of folder application groups" );
            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( "   Required, DefaultValueAvailable, and DefaultValueFixed (true/false) flags" );
            System.out.println( "   Maximum Entry Characters" );
            System.out.println( "   Maximum Display Characters" );
            System.out.println( "   Number and names of database fields" );
            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( "" );

            ODCConfig odConfig = BuildODConfig.build(argv[5] + "/arswww.props");
            if(odConfig == null)
            {
                System.out.println( "" );
                System.out.println( "Testcase TcListCriteria failed -" );
                System.out.println( " ODCConfig could not be initialized. " );
                System.out.println( " Probable cause: " );
                System.out.println( " File arswww.props is not found in directory " + argv[5] );
                System.out.println( "" );
            }
            else
            {
                odServer = new ODServer(odConfig );
                odServer.initialize( "TcListCriteria.java" );
                System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
                char conntype = 'T';
                String directory = "";
                odServer.logon( argv[0], argv[2], argv[3], conntype, Integer.parseInt(argv[1]), directory);

                //-----
                // Open the specified folder and display its name and description
                //-----
                System.out.println( "Opening " + argv[4] + " folder..." );
                odFolder = odServer.openFolder( argv[4] );
                System.out.println( "Name=" + odFolder.getName( ) + " Desc=" + odFolder.getDescription( ) + "" );

                //-----
            }
        }
    }
}
```

```

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

//-----
// For each folder criteria,
//-----
System.out.println( "There are " + odFolder.getNumCriteria( ) + " 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.getDefaultOperator( );
    System.out.println( " Default operator: " );
    System.out.println( " " + getOperatorName( opr ) );

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

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

    //-----
    // Display field format for Date
    //-----
    if ( odCrit.getDefaultFmt( ) != null )
    {
        System.out.println( " Date Format:" );
        System.out.println( " Default Display Fmt " + odCrit.getDefaultFmt( ) );
        System.out.println( " Display Fmt " + odCrit.getDisplayFmt( ) );
    }
    System.out.println( " DisplayFmtQualifier " + odCrit.getDisplayFmtQual( ) );

    //-----
    // Display field id mask info
    //-----
    System.out.println( " FieldIdMask:" );
    System.out.println( " Required=" + odCrit.isRequired( ) +
        " Default=" + odCrit.isDefaultValueAvailable( ) +
        " Fixed Default=" + odCrit.isDefaultValueFixed( ) );

    //-----
    // Display max entry chars
    //-----
    System.out.println( " MaxEntryChars:" );
    System.out.println( " " + odCrit.getMaxEntryChars( ) );

    //-----
    // Display max display chars
    //-----
    System.out.println( " MaxDisplayChars:" );
    System.out.println( " " + odCrit.getMaxDisplayChars( ) );

    //-----
    // Display min search values
    //-----
    System.out.println( " MinSearchValue:" );
    System.out.println( " " + odCrit.getMinSearchValue( ) );

    //-----
    // Display max search values
    //-----
    System.out.println( " MaxSearchValue:" );
    System.out.println( " " + odCrit.getMaxSearchValue( ) );

    //-----
    // Display the database field names
    //-----
    System.out.println( " Database fields:" );
    for ( j = 0; j < appl_grps.length; j++ )
    {
        database_fields = odCrit.getDBFieldNames( (String)appl_grps[j] );
        if(database_fields == null)
            System.out.println("No DBFields Defined");
        else{
            System.out.println( " " + database_fields.length + " mapping for ApplGroup " + (String)appl_grps[j] + "':" );
            for ( k = 0; k < database_fields.length; k++ )
            {
                System.out.println( " " + "DB Field Name = " + ( database_fields[k].equals( "" ) ? ) );
                System.out.println( " [Empty string]" : database_fields[k] );
                long fieldMask = odCrit.getDBFieldMask( (String)appl_grps[j], database_fields[k] );
                System.out.println( " mask: " + fieldMask );

                System.out.println( " ApplID Field = " +
                    ((fieldMask & ODConstant.OD_FLDMSK_APPL)==0
                    ? false : true ));

                System.out.println( " Field Can Be Updated? = " +

```



```

        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
- ポート
- ユーザー ID
- パスワード
- 構成ディレクトリー (arswww.props ファイルのロケーション)

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;
        String folder_desc;
        int num_folders;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 5 )

```



```

    {
        System.out.println( "usage: java TcListFolders <server> <port> <userid> <password> <config dir>" );
        return;
    }
}

try
{
    //-----
    // Set the stage
    //-----

    System.out.println( "Testcase TcListFolders started." );
    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
    //-----
    ODConfig odConfig = BuildODConfig.build(argv[4] + "/arswww.props");
    if(odConfig == null)
    {
        System.out.println( "" );
        System.out.println( "Testcase TcListFolders failed -" );
        System.out.println( "  ODConfig could not be initialized. " );
        System.out.println( "  Probable cause: " );
        System.out.println( "  File arswww.props is not found in directory " + argv[4] );
        System.out.println( "" );
    }
    else
    {
        odServer = new ODServer(odConfig);
        odServer.initialize( "TcListFolders.java" );
        System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
        char conntype = 'T';
        String directory = "";
        odServer.logon( argv[0], argv[2], argv[3], conntype, Integer.parseInt(argv[1]), directory);

        //-----
        // Display the number of folders available.
        //-----
        num_folders = odServer.getNumFolders( );

        System.out.println( "" );
        System.out.println( "There are " + num_folders + " folders available to " + argv[2] + " 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 TcListFolders 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( );
}
}
}

```

フォルダー基準情報の表示

次の例では、ODServer メソッドを使用して指定サーバー上にフォルダーを開き、そのフォルダー名と記述を表示し、フォルダー基準情報を印刷します。フォルダー基準には、以下が含まれます。

- 名前
- デフォルト演算子
- 有効な演算子
- フィールド・タイプ
- デフォルト値
- 固定値

フォルダー基準情報を表示する場合、必ず以下が守られるようにしてください。

- 演算子に `Unknown operator` を示すものがない。
- フィールド・タイプに `Unknown type` を示すものがない。
- 各メソッドでデフォルト値が同じである。
- ODWEK API によって表示されるフォルダー基準情報が、OnDemand クライアントによって表示されるものと一致している。

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
- ポート
- ユーザー ID
- パスワード
- フォルダー基準
- 構成ディレクトリー (`arswww.props` が含まれる)

フォルダー基準情報表示の例:

```
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;
public class TcListFoldersByCrit
{
    public static void main ( String argv[] )
    {
        ODServer odServer;
        Enumeration folders_enum;
        String folder_name;
        String folder_desc;
        int num_folders;
        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 6 )
        {
            System.out.println( "usage: java TcListFoldersByCrit <server> <port> <userid> <password> <Folder Criteria> <config dir>" );
            return;
        }
        try
        {
            //-----
```

```

// Set the stage
//-----
System.out.println( "Testcase TcListFoldersByCrit started." );
System.out.println( "This testcase should:" );
System.out.println( "  Log onto the server." );
System.out.println( "  Display the total number of folders available for the user." );
System.out.println( "  Display the number of folders available for the user with criteria." );
System.out.println( "  Display one line for each folder retrieved from search criteria, " );
System.out.println( "    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
//-----
//-----
ODConfig odConfig = BuildODConfig.build(argv[5] + "/arswww.props");
if(odConfig == null)
{
    System.out.println( "" );
    System.out.println( "Testcase TcListFoldersByCrit failed -" );
    System.out.println( "  ODConfig could not be initialized. " );
    System.out.println( "  Probable cause: " );
    System.out.println( "  File arswww.props is not found in directory " + argv[5] );
    System.out.println( "" );
}
else
{
    odServer = new ODServer(odConfig );
    odServer.initialize( "TcListFolders.java" );
    System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
    char conntype = 'T';

    String directory = "";
    odServer.logon( argv[0], argv[2], argv[3], conntype, Integer.parseInt(argv[1]), directory);

    //-----
    // Display the total number of folders available.
    //-----
    num_folders = odServer.getNumFolders( );

    System.out.println( "" );
    System.out.println( "There are " + num_folders + " folders available to user " + argv[2] );
    System.out.println( "  on server " + argv[0] + "." );

    //-----
    // Display the number of folders available with search criteria.
    //-----
    num_folders = odServer.getNumFolders(argv[4]);

    System.out.println( "" );
    System.out.println( "There are " + num_folders + " folders with criteria " + argv[4] );
    System.out.println( "  available to user " + argv[2] + " on server " + argv[0] + "." );

    //-----
    // Display the folder names and descriptions
    //-----
    int i = 1;

    // get a limited folder list based on search criteria

    System.out.println( "" );
    System.out.println( "List folders found using search criteria... " + argv[4] + ":" );
    System.out.println( "" );

    i = 1;
    for (Enumeration folders_enum2 = odServer.getFolderNames(argv[4]); folders_enum2.hasMoreElements( ); )
    {
        folder_name = (String)folders_enum2.nextElement( );
        folder_desc = odServer.getFolderDescription( folder_name );

        System.out.println(i++ + " " + folder_name + " --- " + folder_desc );
    }

    //-----
    // Cleanup
    //-----
    odServer.logoff( );
    odServer.terminate( );

    System.out.println( "" );
    System.out.println( "-----" );
    System.out.println( "" );
    System.out.println( "Testcase TcListFoldersByCrit completed - " );
    System.out.println( "  compare results to Windows Client if required" );
    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( );
}
}
}
}

```

文書のリストの表示

以下の例では、ODFolder、ODHit、および ODCriteria 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。

- getDisplayOrder
- getFolderSortOrder
- getCriteria
- getSortLocation
- setSortLocation
- 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 ODCriteria メソッドを示します。

- getName
- setSortOrder
- getAscending
- setAscending

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
- ポート
- ユーザー ID
- パスワード
- フォルダー名
- 第 1 基準 (主要なソート・キー)
- 第 1 基準の昇順(1) または降順(0)
- 第 2 基準 (重要度のやや低いソート・キー)
- 第 2 基準の昇順(1) または降順(0)
- 第 3 基準 (重要度の最も低いソート・キー)
- 第 3 基準の昇順(1) または降順(0)
- 構成ディレクトリー (arswww.props 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;
        ODCriteria odCrit;
        Enumeration crit_enum;
        String server, userid, password, folder, crit1, crit2, crit3;
        boolean ascend1, ascend2, ascend3;
        //-----
        // If too few parameters, display syntax and get out
        //-----
if ( argv.length < 12 )
{
    System.out.println( "usage: java TcSortedHitlist <server> <port> <userid> <password> <folder> <criteria> " +
        " <ascending1> <criteria2> <ascending2> <criteria3> <ascending3> <config dir>" );
    return;
}
try
{
    //-----
    // Set the stage
    //-----
    System.out.println( "Testcase TcSortedHitlist started." );
    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 default sort order");
    System.out.println(" Leave sort enablement disabled by default");
    System.out.println(" Search the folder using the default criteria and display the Hitlist ");
    System.out.println(" Display the default sort order");
    System.out.println(" Enable sort");
    System.out.println(" Search the folder using the default criteria and display the Hitlist ");
    System.out.println(" Set the sort order specified on the command line");
    System.out.println(" Disable sort");
    System.out.println(" Display the new sort order");
    System.out.println(" Search the folder using the default criteria and display the Hitlist ");
    System.out.println(" Enable sort");
    System.out.println(" Search the folder using the default criteria and display the Hitlist ");
    System.out.println("");
    System.out.println("-----");
    System.out.println("");

    //-----
    // Logon to the server
    //-----
    server = argv[0];
    userid = argv[2];
    password = argv[3];
    folder = argv[4];
    crit1 = argv[5];
    ascend1 = argv[6].equals( "0" ) ?false:true;
    crit2 = argv[7];
    ascend2 = argv[8].equals( "0" ) ?false:true;
    crit3 = argv[9];
    ascend3 = argv[10].equals( "0" ) ?false:true;
    ODCConfig odConfig = BuildODConfig.build(argv[11] + "/arswww.props");
    if(odConfig == null)
        System.out.println("BuildODConfig Failed.");
    else
    {
        odServer = new ODServer(odConfig );
        odServer.initialize( "TcSortedHitlist.java" );
        String directory = "";
        System.out.println( "Logging on to " + server + " server with user " + userid + "..." );
        odServer.logon( server, userid, password, ODCConstant.CONNECT_TYPE_TCPIP, Integer.parseInt(argv[11]), directory);

        //-----
        // Open the folder and display default folder sort order
        //-----
        System.out.println("Opening " + folder + "...");
        odFolder = odServer.openFolder(folder);
        System.out.println("Sort Location = " + odFolder.getSortLocation());
        displayFolderSortOrder(odFolder);
        //-----
        // Search the folder with sort disabled by default
        //-----
        System.out.println("With sort disabled by default:");
        System.out.println(" Searching with SORT=NONE...");
        searchAndListHits(odFolder);
        //-----
        // Search the folder after enabling sort
        //-----
        odFolder.setSortLocation(ODConstant.OD_SORT_LOCATION_MIDTIER);
        displayFolderSortOrder(odFolder);
        System.out.println("With Mid-tier sort enabled: USING Default Folder sort settings");
        System.out.println("Sort Location = " + odFolder.getSortLocation());
        System.out.println(" Searching folder with SORT=M and default criteria...");
        searchAndListHits(odFolder);
        //-----
        // Create a new sort order and display it
        //-----
        System.out.println("/n/nSetting specified sort order...");
        System.out
        .println("-----");
        for (crit_enum = odFolder.getCriteria(); crit_enum.hasMoreElements(); )
        {
            odCrit = (ODCriteria) crit_enum.nextElement();
            odCrit.setSortOrder(0);
        }
        if (!crit1.equals("-"))
        {
            odCrit = odFolder.getCriteria(crit1);
            odCrit.setSortOrder(1);
            odCrit.setAscending(ascend1);
        }
        if (!crit2.equals("-"))

```

```

        {
            odCrit = odFolder.getCriteria(crit2);
            odCrit.setSortOrder(2);
            odCrit.setAscending(ascend2);
        }
        if (!crit3.equals("-"))
        {
            odCrit = odFolder.getCriteria(crit3);
            odCrit.setSortOrder(3);
            odCrit.setAscending(ascend3);
        }
        displayFolderSortOrder(odFolder);

        //-----
        // Search the folder after disabling sort
        //-----
        odFolder.setSortLocation(ODConstant.OD_SORT_LOCATION_NONE);
        System.out.println("With sort disabled:");
        System.out.println(" Searching folder with setSortLocation(NONE) ...");
        System.out.println("Sort Location = " + odFolder.getSortLocation());
        searchAndListHits(odFolder);
        //-----
        // Search the folder after enabling sort
        //-----
        odFolder.setSortLocation(ODConstant.OD_SORT_LOCATION_MIDTIER);
        displayFolderSortOrder(odFolder);
        System.out
        .println(" Searching folder with setSortLocation(MIDTIER) and CmdLine sort order (default criteria)...");
        System.out.println("Sort Location = " + odFolder.getSortLocation());
        searchAndListHits(odFolder);
        //-----
        // Search the folder after enabling Server sort
        //-----
        odFolder.setSortLocation(ODConstant.OD_SORT_LOCATION_SERVER);
        displayFolderSortOrder(odFolder);
        System.out
        .println(" Searching folder with setSortLocation(SERVER) and CmdLine sort order (default criteria)...");
        System.out.println("Sort Location = " + odFolder.getSortLocation());
        searchAndListHits(odFolder);
        //-----
        // Cleanup
        //-----
        odFolder.close();
        odServer.logoff();
        odServer.terminate();
        System.out.println("");
        System.out.println("-----");
        System.out.println("");
        System.out.println("Testcase TcSortedHitlist completed - Ensure that:");
        System.out.println(" 1. The default sort order is the same as that shown by the");
        System.out.println("    Windows Client");
        System.out.println(" 2. The 1st hitlist is not sorted");
        System.out.println(" 3. The 2nd hitlist is sorted according to the default order");
        System.out.println(" 4. The new sort order is correct for the command line");
        System.out.println(" 5. The 3rd hitlist is not sorted");
        System.out.println(" 6. The 4th hitlist is sorted according to the specified order");
        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 void searchAndListHits( ODFolder odFolder )
{
    ODHit odHit;
    Vector hits;
    String[ ] display_crit;
    String value;
    int j, k;
    try
    {
        System.out.println( " Searching folder with default criteria..." );
        hits = odFolder.search();
        System.out.println(" Number of hits: " + hits.size());
        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);
            }
        }
    }
    catch ( ODEException 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();
    }
}
static void displayFolderSortOrder( ODFolder odFolder )
{
    ODCriteria odCrit;
    ArrayList sort_order;
    int j, order;
    try
    {
        sort_order = odFolder.getFolderSortOrder();
        System.out.println("Sort Order:");
        for ( order = -1; order < 100; order++ )
        {
            for( j = 0; j < sort_order.size( ); j++ )
            {
                odCrit = (ODCriteria) sort_order.get(j);
                if (odCrit.getSortOrder() == order)
                    System.out.println(" "
                        + order
                        + ". "
                        + odCrit.getName()
                        + (order <= 0 ? "" : " ("
                            + (odCrit.getAscending() ? "ascending" : "descending")
                            + ")"));
            }
        }
    }
    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
- Logoff
- 取り消し
- 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。

- getViewMimeType
- getDocType
- getDocument
- getResources
- 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
- ポート
- ユーザー ID
- パスワード
- フォルダ名
- 構成ディレクトリー (arswww.props ファイルのロケーション)

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[ ] res_from_server;
        byte[ ] data_from_folder;
        int j;
        int getDocResSize =0;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 6 )
        {
            System.out.println( "usage: java TcRetrieve <server> <port> <userid> <password> <folder> <config dir>" );
            return;
        }
        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "Testcase TcRetrieve started." );
            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 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
            //-----
            ODConfig odConfig = BuildODConfig.build(argv[5] + "/arswww.props");
            if(odConfig == null)
            {
                System.out.println( "" );
                System.out.println( "Testcase TcRetrieve failed -" );
                System.out.println( " ODConfig could not be initialized. " );
                System.out.println( " Probable cause: " );
                System.out.println( " File arswww.props is not found in directory " + argv[5] );
                System.out.println( "" );
            }
            else
            {
                odServer = new ODServer(odConfig );
                odServer.initialize( "TcDeleteDocs.java" );
                String directory = "";
                odServer.logon( argv[0], argv[2], argv[3], ODConstant.CONNECT_TYPE_TCPIP, Integer.parseInt(argv[1]), directory);

                //-----
                // Open the specified folder and search with the default criteria
                //-----
            }
        }
    }
}
```



```

System.out.println( "Opening " + argv[4] + " folder..." );
odServer.cancel();
odFolder = odServer.openFolder( argv[4] );
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 );
    hit_to_retrieve = new Vector( );
    hit_to_retrieve.addElement( odHit );

    System.out.println( " View Mime Type = " + odHit.getViewMimeType());
    System.out.println( " Doc Type = " + odHit.getDocType());
    System.out.println( "Retrieving data from first hit using ODHit.retrieve..." );
    data_from_hit = odHit.retrieve(ODConstant.NATIVE);
    FileOutputStream fos1 = new FileOutputStream("1hitR.out");
    fos1.write(data_from_hit);
    fos1.close();

    System.out.println( "Retrieving data from first hit using ODHit.getDocument/getResources..." );
    data_from_server = odHit.getDocument();
    FileOutputStream fos2 = new FileOutputStream("getDoc.out");
    fos2.write(data_from_server);
    fos2.close();
    if(odHit.getDocType() == ODConstant.FileTypeAFP)
    {
        try{
            res_from_server = odHit.getResources();
            FileOutputStream fos8 = new FileOutputStream("getRes.afp");
            fos8.write(res_from_server);
            fos8.close();
            getDocResSize = data_from_server.length + res_from_server.length;
        }
        catch(OException E)
        {
            System.out.println("There are no resources for this document.");
            getDocResSize = data_from_server.length;
        }
    }
    else
        getDocResSize = data_from_server.length ;

    System.out.println( "Retrieving data from first hit using ODFolder.retrieve (uses callback method)..." );
    callback = new Tcallback( );
    odFolder.retrieve( hit_to_retrieve, callback );
    data_from_folder = callback.getData( );

    if ( data_from_folder == null )
    {
        data_from_folder = new byte[0];
        System.out.println( "Callback function not called during ODFolder.retrieve" );
    }
    else
    {
        FileOutputStream fos = new FileOutputStream("3folderR.out");
        fos.write(data_from_folder);
        fos.close();
    }
    System.out.println( "Length of data from:" );
    System.out.println( " ODHit.retrieve=" + data_from_hit.length );
    System.out.println( " ODHit.getDocument (+getResource) =" + getDocResSize );
    System.out.println( " ODFolder.retrieve=" + data_from_folder.length );
    if ( data_from_hit.length == getDocResSize )
    {
        System.out.println( "ODHit vs. ODHit.getDocument: 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 match" );
            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. ODHit.getDocument: 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 TcRetrieve completed - analyze the result of the comparisons" );
        System.out.println( "" );
        System.out.println( "If the arwww.props file specifies 'native' for the viewer 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
- printDocuments
- 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
- ポート
- 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.props ファイルのロケーション)

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;
        Vector hit_to_print;
        String [] printers;
        String printer_name;
        boolean match;
        int j;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 7 )
        {
            System.out.println( "usage: java TcPrintHit <server> <port> <userid> <password> <folder> <printer> <config dir>" );
            return;
        }

        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "Testcase TcPrintHit started." );
            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
//-----
ODConfig odConfig = BuildODConfig.build(argv[6] + "/arswww.props");
if(odConfig == null)
{
    System.out.println( "" );
    System.out.println( "Testcase TcListFoldersByCrit failed -" );
    System.out.println( " ODConfig could not be initialized. " );
    System.out.println( " Probable cause: " );
    System.out.println( " File arswwww.props is not found in directory " + argv[6] );
System.out.println( "" );
}
else
{
    odServer = new ODServer(odConfig );
    odServer.initialize( "TcPrintHit.java" );
    char conntype = 'T';
    String directory = "";
    System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
    odServer.logon( argv[0], argv[2], argv[3], conntype, Integer.parseInt(argv[1]), directory);

//-----
// If any server printers are available on the server
//-----
System.out.println( "Retrieving list of server printers..." );
printer_name = argv[5];
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[4] + " folder..." );
odFolder = odServer.openFolder( argv[4] );
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.printDocuments( hit_to_print, printer_name,2);
}
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 TcPrintHit 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。

- getNoteStatus
- 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
- ポート
- 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.props ファイルのロケーション)

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;
String str;
int j, exist;
//-----
// If too few parameters, display syntax and get out
//-----
if ( argv.length < 6 )
{
    System.out.println( "usage: java TcListNotes <server> <port> <userid> <password> <folder> <config dir>" );
    return;
}
try
{
    //-----
    // Set the stage
    //-----
    System.out.println( "Testcase TcListNotes started." );
    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 note status for the first hit");
    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
    //-----
    ODConfig odConfig = BuildODConfig.build(argv[5] + "/arswww.props");
    if (odConfig == null)
    {
        System.out.println( "" );
        System.out.println( "Testcase TcListNotes failed -" );
        System.out.println( " ODConfig could not be initialized. " );
        System.out.println( " Probable cause: " );
        System.out.println( " File arswwww.props is not found in directory " + argv[5] );
        System.out.println( "" );
    }
    else
    {
        odServer = new ODServer(odConfig);
        odServer.initialize("TcListNotes.java");
        System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
        char conntype = 'T';
        String directory = "";
        odServer.logon( argv[0], argv[2], argv[3], conntype, Integer.parseInt(argv[1]), directory);
        //-----
        // Open the specified folder and search with the default criteria
        //-----
        System.out.println( "Opening " + argv[4] + " folder..." );
        odFolder = odServer.openFolder( argv[4] );
        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);
            System.out.println(" For the first hit:");
            System.out.println(" Note status is: "
                + odHit.getNoteStatus());
            notes = odHit.getNotes();
            if (notes.size() > 0)
                System.out.println(" There is(are) " + notes.size() + " note(s)");
            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());
                System.out.println(" Text=" + odNote.getText());
            }
        }
        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 TcListNotes completed - Ensure that the information" );
        System.out.println(" is the same as shown by the Windows Client");
        System.out.println("");
    }
}
catch ( ODException 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 getExistenceName( int code )
    {
        String str;
        switch( code )
        {
            case ODConstant.NoteStatusUnknown:
                str = "UNKNOWN";
                break;
            case ODConstant.NoteStatusYes:
                str = "YES";
                break;
            case ODConstant.NoteStatusNo:
                str = "NO";
                break;
            case ODConstant.NoteStatusError:
                str = "ERROR";
                break;
            default:
                str = "UNDEFINED VALUE";
        }
        return str;
    }
}

```

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`

この例には、以下の `ODNote` メソッドを示します。

- `getGroupName`
- `getText`

- isOkToCopy
- isPublic
- setGroupName
- setOkToCopy
- setPublic
- setText

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
- ポート
- 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.props ファイルのロケーション)

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;
        Vector notes;
        int j;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 7 )
        {
            System.out.println( "usage: java TcAddNote <server> <port> <userid> <password> <folder> <note text> <config dir>" );
            return;
        }
        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "Testcase TcAddNote started." );
            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
            //-----
            ODConfig odConfig = BuildODConfig.build(argv[6] + "/arswww.props");

            if (odConfig != null)
            {
                odServer = new ODServer(odConfig );
                odServer.initialize( "TcAddNote.java" );
                System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
                char conntype = 'T';
                String directory = "";
                odServer.logon( argv[0], argv[2], argv[3], conntype, Integer.parseInt(argv[1]), directory);
                //-----
                // Open the specified folder and search with the default criteria
                //-----
                System.out.println( "Opening " + argv[4] + " folder..." );
                odFolder = odServer.openFolder( argv[4] );
                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 );
    System.out.println("Working with DocID " + odHit.getDocId());
    notes = odHit.getNotes( );
    if(notes.size() > 0)
    System.out.println(" There are " + notes.size( ) + " notes for the first hit" );

    odNote = new ODNote( );
    odNote.setText( argv[5] );
    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(" UserId=" + argv[2] );
    System.out.println(" Page=1" );
    System.out.println(" Color=" + 'Y' );
    System.out.println(" Group=" + odNote.getGroupName( ) );
    System.out.println(" Offset=(0,0)" );
    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 TcAddNote completed. - Ensure that the new note was correctly" );
System.out.println( " added by displaying it with the Windows Client" );
System.out.println( "" );
}
else
{
    System.out.println( "" );
    System.out.println( "Testcase TcAddNote failed -" );
    System.out.println( " ODConfig could not be initialized. " );
    System.out.println( " Probable cause: " );
    System.out.println( " File arswww.props is not found in directory " + argv[6] );
    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( );
}
}
}

```

注の削除

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、最初の注を削除します。

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。

- search
- getHits
- close

この例には、以下の ODHit メソッドを示します。

- getDocId
- getNotes
- deleteNote

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
- ポート
- 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.props が含まれる)

注釈の削除の例:

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

public class TcDeleteNote
{
    public static void main ( String argv[ ] )
    {
        ODServer odServer;
        ODFolder odFolder;
        ODHit odHit;
        ODNote odNote;
        Vector hits;
        Vector notes;
        int j, numNotes1 = 0, numNotes2 = 0;

        //-----
        // If too few parameters, display syntax and get out
        //-----
        if ( argv.length < 6 )
        {
            System.out.println( "usage: java TcDeleteNote <server> <port> <userid> <password> <folder> <config dir>" );
            return;
        }
        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "Testcase TcDeleteNote started." );
            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( "  Delete first note" );
            System.out.println( "  " );
            System.out.println( "-----" );
            System.out.println( "  " );

            //-----
            // Logon to specified server
            //-----
            ODConfig odConfig = BuildODConfig.build(argv[5] + "/arswww.props");

            if (odConfig != null)
            {
                odServer = new ODServer(odConfig );
                odServer.initialize( "TcDeleteNote.java" );
                System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
                char conntype = 'T';
                String directory = "";
                odServer.logon( argv[0], argv[2], argv[3], conntype, Integer.parseInt(argv[1]), directory);
                //-----
                // Open the specified folder and search with the default criteria
                //-----
                System.out.println( "Opening " + argv[4] + " folder..." );
                odFolder = odServer.openFolder( argv[4] );
                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);
System.out.println("Working with DocID " + odHit.getDocId());
notes = odHit.getNotes();
numNotes1 = notes.size();
System.out.println(" There are " + numNotes1 + " notes for the first hit");

if (numNotes1 > 0)
{
odNote = (ODNote)notes.elementAt(0);
System.out.println(" Delete in the first note...");
odHit.deleteNote(odNote);

// destroy hits
hits.removeAllElements();

// search again
System.out.println("Searching second time with difault criteria...");
odFolder.search();
hits = odFolder.getHits();
System.out.println(" Number of hits: " + hits.size());

// Retrieve notes again
if (hits.size() > 0)
{
odHit = (ODHit)hits.elementAt(0);
System.out.println("Working with DocID " + odHit.getDocId());
notes = odHit.getNotes();
numNotes2 = notes.size();
System.out.println(" There are " + numNotes2 + "notes for the first hit");
}
else
System.out.println("No document - cannot list notes");
}
else
System.out.println("No document - cannot list notes");

if ((numNotes1 - numNotes2) == 1)
System.out.println("\nSuccess!");
else if (numNotes1 == 0)
System.out.println("\nSuccess!");
else
System.out.println("\nFailed!");
}
else
System.out.println("\nNo hits found");

//-----
// Cleanup
//-----
odFolder.close( );
odServer.logoff( );
odServer.terminate( );
System.out.println( " " );
System.out.println( "-----" );
System.out.println( " " );
System.out.println( "Testcase TcDeleteNote completed. - Ensure that the new note was correctly" );
System.out.println( " added by displaying it with the Windows Client" );
System.out.println( " " );
}
else
{
System.out.println( " " );
System.out.println( "Testcase TcDeleteNote failed -" );
System.out.println( " ODConfig could not be initialized. " );
System.out.println( " Probable cause: " );
System.out.println( " File arswww.props is not found in directory " + argv[5] );
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。

- getFolderSortOrder
- getDisplayOrder
- getCriteria
- search
- close

この例には、以下の ODCriteria メソッドを示します。

- setOperator
- setSearchValues
- setSearchValue

この例には、以下の ODHit メソッドを示します。

- getDocId
- getDisplayValue
- updateValuesForHit

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
- ポート
- ユーザー ID
- パスワード
- フォルダー名
- 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 (arswww.props 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, odCrit2;
```

```

ODHit odHit;
Hashtable hash;
Vector hits;
String[] display_crit;
String line;
String crit1;
String crit2;
String value1;
String value2;
String new_value;
int j;

//-----
// If too few parameters, display syntax and get out
//-----
if ( argv.length < 10 )
{
    System.out.println( "usage: java TcUpdate <server> <port> <userid> <password> <folder>" );
    System.out.println( "    <criteria1> <value1> <criteria2> <value2> <new value2> <config dir>" );
}
return;
}
try
{
    System.out.println( "Testcase TcUpdate started." );
    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( "" );

    //-----
    //Setup ODConfig object using defaults and initialize ODServer.
    //-----
    ODConfig odConfig = BuildODConfig.build(argv[10] + "/arswww.props");
    if(odConfig == null)
    {
        System.out.println( "" );
        System.out.println( "Testcase TcUpdate failed -" );
        System.out.println( " ODConfig could not be initialized. " );
        System.out.println( " Probable cause: " );
        System.out.println( " File arswww.props is not found in directory " + argv[10] );
        System.out.println( "" );
    }
    else
    {
        odServer = new ODServer(odConfig);
        odServer.initialize( "TcUpdate.java");

        //-----
        // Logon to specified server
        //-----
        System.out.println( "Logging on to " + argv[0] + " server with user " + argv[2] + "..." );
        String directory = "";
        odServer.logon( argv[0], argv[2], argv[3], ODConstant.CONNECT_TYPE_TCPIP, Integer.parseInt(argv[1]), directory);

        //-----
        // Open the specified folder and set the requested criteria
        //-----
        crit1 = argv[5];
        crit2 = argv[7];
        value1 = argv[6];
        value2 = argv[8];
        new_value = argv[9];
        System.out.println( "Opening " + argv[4] + " folder..." );
        odFolder = odServer.openFolder( argv[4] );

        ArrayList allCrit = odFolder.getFolderSortOrder();
        /*Clear all default search values*/
        for (int a =0; a < allCrit.size(); a++)
        {
            odCrit = (ODCriteria)allCrit.get(a);
            odCrit.setSearchValues("", "");
        }
        odCrit = odFolder.getCriteria( crit1 );
        if(odCrit == null)
        System.out.println("ERROR: Search Criteria Not Found [" +crit1+"];");
        else
        {
            odCrit.setOperator( ODConstant.OPEqual );
            odCrit.setSearchValue( value1 );
        }
        odCrit2 = odFolder.getCriteria( crit2 );
        if(odCrit2 == null)
        System.out.println("ERROR: Search Criteria Not Found [" +crit2+"];");
        else
        {
            odCrit2.setOperator( ODConstant.OPEqual );
            odCrit2.setSearchValue( value2 );
        }
        //-----
        // Search the folder
        //-----
        if (odCrit != null && odCrit2 != null)

```

```

{
    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:" );
        odHit = (ODHit)hits.elementAt( 0 );
        System.out.println( " DOCID = "+ odHit.getDocId( ) );
        line = " ";
        display_crit = odFolder.getDisplayOrder( );
        for( j = 0; j < display_crit.length; j++ )
            line = line + display_crit[j] + " ";
        System.out.println( line );
        line = " ";
        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.updateValuesForHit( hash );
        System.out.println("New display values as set in ODHit");
        odHit = (ODHit)hits.elementAt( 0 );
        line = " ";
        System.out.println( " DOCID = "+ odHit.getDocId( ) );
        display_crit = odFolder.getDisplayOrder( );
        for( j = 0; j < display_crit.length; j++ )
            line = line + display_crit[j] + " ";
        System.out.println( line );
        line = " ";
        for ( j = 0; j < display_crit.length; j++ )
            line = line + odHit.getDisplayValue( display_crit[j] ) + " ";
        System.out.println( line );
    }
    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 TcUpdate 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
- getPassword
- changePassword
- setUserId
- setPassword
- setServerName
- 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
- ポート
- ユーザー ID
- パスワード
- 新規パスワード
- 初回ログオン
- 構成ディレクトリー (arswww.props ファイルのロケーション)

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 < 7 )
        {
            System.out.println( "usage: java TcChangePassword <server> <port> <userid> <password> " );
            System.out.println( "                                     <new password> <initial logon> <config dir>" );
            return;
        }
        try
        {
            //-----
            // Set the stage
            //-----
            System.out.println( "Testcase TcChangePassword started." );
            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[2];
            original_password = argv[3];
            new_password = argv[4];
            char conntype = 'T';
            String directory = "";
            int logon_first= Integer.parseInt(argv[5]);

            ODConfig odConfig = BuildODConfig.build(argv[6] + "/arswww.props");
            if(odConfig == null)
            {
                System.out.println( "" );
                System.out.println( "Testcase TcGetFolders failed -" );
                System.out.println( " ODConfig could not be initialized. " );
                System.out.println( " Probable cause: " );
                System.out.println( " File arswww.props is not found in directory " + argv[6] );
                System.out.println( "" );
            }
            else
            {
                odServer = new ODServer(odConfig );
                odServer.initialize( "TcChangePassword.java" );
            }
        }
    }
}
```

```

if(logon_first == 1)
{
    //-----
    // Logon to the server using the original password
    //-----
    System.out.println( "Logging on to " + server + " using original password..." );
    odServer.logon( server, userid, original_password, conntype, Integer.parseInt(argv[1]), directory);
}
else
{
    System.out.println("Do NOT logon, but set the pre-req fields of UID,PWD,and Server");
    odServer.setUserId(userid);
    odServer.setPassword(original_password);
    odServer.setServerName(server);
}
//-----
// Change to the new password and logoff
//-----
System.out.println( "Changing to new password..." );
odServer.changePassword( new_password );
System.out.println( "Current ODServer.password is " + odServer.getPassword() );
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..." );
odServer.logon( server, userid, new_password, conntype, Integer.parseInt(argv[1]), directory );

//-----
// Change back to the original password and logoff
//-----
System.out.println( "Changing back to original password..." );
odServer.changePassword( original_password );
System.out.println( "Current ODServer.password is " + odServer.getPassword() );
System.out.println( "Logging off..." );
odServer.logoff( );

//-----
// Cleanup
//-----
odServer.terminate( );
System.out.println( " " );
System.out.println( "-----" );
System.out.println( " " );
System.out.println( "Testcase TcChangePassword 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 から 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 間変換プロセスは、AFP2WEB Transform を必要とします。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 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。

重要: 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、29 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

重要: AFP2HTML アプレットで文書を変換するには、ARSWWW.INI ファイル (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) の DEFAULT BROWSER セクション (または、他のブラウザ・セクション) に 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、41 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、97 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があります。(29 ページの『CONFIGFILE』を参照してください。)

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。169 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 13 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 13. 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。

重要: 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 から 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。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 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。

重要: 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、31 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

重要: 文書を変換するには、ARSWWW.INI ファイル (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) の DEFAULT BROWSER セクション (または、他のブラウザ・セクション) に AFPVIEWING=PDF パラメーターを指定する必要もあります。詳細については、41 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、97 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、31 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ビューアー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サンプルを示します。

- HTTP Apache サーバー
- WebSphere Application Server

重要: 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

以下は、HTTP サーバー構成ファイルのサンプルです。(HTTP 管理クライアントを使用して、最初に HTTP サーバー構成を作成しておく必要があります。)このサンプル構成は、インスタンス QUSROND や、HTTP サーバー ODAPACHE を対象としています。太字で示される構成項目は、ODWEK に必須であるか、ODWEK に関連しています。構成の注は、括弧で囲まれた太字のイタリック体で示されます。これらの注は、構成ファイルには入力しないでください。コメント行の行頭には、# が付いています、この構成ファイルが配置されているサーバー上の IFS のファイル名およびパスは、/www/odapache/conf/httpd.conf です。

注:

1. 「オリジナルの」HTTP サーバーは、現在サポートされていません。
2. ODWEK は、エンド・ユーザーのブラウザーが、UTF-8 フォーマットを受け入れることを必要とします。Microsoft Internet Explorer で、「ツール」>「インターネットオプション」に続いて、「詳細設定」タブを選択します。「ブラウズ」の下で、「常に UTF-8 として URL を送信する」を選択します。

```
=====
Listen *:ppppp    (Set to the port that the browser uses to connect to ODWEK.
The default TCP/IP port number is 80.)
DocumentRoot /www/odapache/htdocs
ServerRoot /www/odapache
DefaultType text/plain
Options -ExecCGI -FollowSymLinks -SymLinksIfOwnerMatch -Includes-IncludesNoExec
-Indexes -MultiViews
ErrorLog logs/error_log
LogLevel Warn
DirectoryIndex index.html
HostNameLookups off
RuleCaseSense OFF
LimitRequestBody 102400
ServerName f.q.h.n (Enter the fully-qualified host name or the TCP/IP address of
the server running ODWEK.)
UseCanonicalName Off
DefaultFsCCSID 0037 (This should be the CCSID of the OnDemand instance.)
DefaultNetCCSID 1208
CGIConvMode EBCDIC (For DBCS languages, change this to EBCDIC_JCD)
ScriptLogLength 200
ScriptLog /www/odapache/logs/cgi_log
Alias /logon /www/odapache/htdocs/oda_logon.html
                                     (The above line points to your logon script name and location.)
AliasMatch ^/images/(.*)$ /www/odapache/htdocs/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
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%(Cookie)n \"%r\" %t" cookie
LogFormat "%{User-agent}i" agent
LogFormat "%{Referer}i -> %U" referer
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log combined
SetEnvIf "User-Agent" "Mozilla/2" nokeepalive
SetEnvIf "User-Agent" "JDK/1\0" force-response-1.0
SetEnvIf "User-Agent" "Java/1\0" force-response-1.0
SetEnvIf "User-Agent" "RealPlayer 4\0" force-response-1.0
SetEnvIf "User-Agent" "MSIE 4\0b2;" nokeepalive
SetEnvIf "User-Agent" "MSIE 4\0b2;" force-response-1.0

# Root directory access authority
<Directory />
Order Deny,Allow
    Deny From all
    Options None
    Options +ExecCGI
    AllowOverride NoneLimit
<Except GET HEAD OPTIONS TRACE POST PUT>
</LimitExcept>
</Directory>

# Directory access for the HTTP server directory documents
<Directory /www/odapache/htdocs>
Order Allow,Deny
    Allow From all
</Directory>

# Directory access to the ProdData ..... line data applets directory
<Directory /QIBM/ProdData/OnDemand/www/applets/>
Order Allow,Deny
    Allow From all
</Directory>

# Directory access to the ProdData .... images directory
<Directory /QIBM/ProdData/OnDemand/www/images>
Order Allow,Deny
    Allow From all
Options +FollowSymLinks
</Directory>

# Directory access to ProdData .... samples directory
<Directory /QIBM/ProdData/OnDemand/www/samples/>
Order Allow,Deny
    Allow From all
Options +FollowSymLinks
</Directory>

# Directory access to the HTTP server instance directory
<Directory /www/odapache/>
Order Allow,Deny
    Allow From all
</Directory>
# Directory access to the QRDARS library
<Directory /QSYS.LIB/QRDARS.LIB>
Order Allow,Deny
    Allow From all
</Directory>

=====

```

WebSphere Application Server

WebSphere 構成ファイルのサンプルは、IBM Content Manager OnDemand for i サポートの Web ページ (<http://www.ibm.com/software/data/ondemand/400/support.html>) から入手できます。この情報を見つけるには、ODWEK WebSphere 構成で検索してください。

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。

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、出力レコードのセットです。これらのレコードには、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が含まれます。

- 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など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 ([)、および右大括弧文字 (]) 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、次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、39 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。この名前は、二重引用符で囲みません。
3. スtring folderDesc はフォルダーの説明です。この説明は、二重引用符で囲みません。

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。この検索基準は、二重引用符で囲みません。
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 14 を参照してください。

表 14. 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。この値は、二重引用符で囲みません。
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 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、エラー発生時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

DBCS 言語用の ODWEK の構成

このセクションは、管理者が DBCS 言語用に ODWEK を構成する場合に役立つ情報を記述します。

CGI および Java サブレットの場合は ARSWWW.INI ファイルの CODEPAGE および LANGUAGE パラメーター、また、Java API の場合は ARSWWW.PROPS ファイルの 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 15 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ます。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。

表 15. DBCS 言語、コード・ページ、およびロケール

国または地域	LANGUAGE=	CODEPAGE=	ロケール
中国 (PRC)	CHS	1388	ZH_CN
日本	JPN	5035	JA_5035
韓国	KOR	933	KO_KR
台湾	CHT	937	ZH_TW

ODWEK におけるコード・ページ変換

標準の OnDemand クライアントと比べると、ODWEK はコード・ページの変換時に異なる動作をします。ODWEK は、中間層のシステムであるので、必ず追加のプレゼンテーション層があります。ほとんどの場合、その層はブラウザーになりますが、ODWEK API を使用するスタンドアロンの Java アプリケーションになる場合もあります。

ODWEK は、内部的には UTF-8 で実行され、その結果、すべての索引とアノテーション・データは UTF-16 (データが TCP/IP で送信される場合のフォーマット) から UTF-8 に変換されます。

ODWEK を使用する Java アプリケーションを実装する場合、ODWEK API との間で受け渡す情報を確実に正しく処理するようにしてください。Java は、内部的には UTF-16 Unicode で動作するので、ODWEK API 関数から戻されるデータは無視することができます。Java が、UTF-8 からの変換を処理します。ODWEK メソッドにストリングを受け渡す場合、Java は UTF-16 であるストリング変数のみをサポートするので追加のタスクを実行する必要はありません。UTF-8 への変換は、ODWEK のネイティブ・サブルーチンにより実行されます。

UTF-8 への内部変換は行いますが、ODWEK は索引に対して他の変換を一切行いません。したがって、ODWEK API を介して受け取る索引データを表示するクライアントは UTF-8 Unicode データを処理する必要があります。ODWEK ベースの Java アプリケーションを使用して、索引データを外部のアプリケーションに送信する場合、それらのアプリケーションで確実に Unicode データを処理できるようにするか、または、そのデータを手動で変換する必要があります。索引またはアノテーション・データをファイルに保存する場合にも同様の配慮が必要です。明示的な変換を実行しない場合、データは Unicode データ・ストリームとして書き込まれます。通常の Web ブラウザーは UTF-8 データの表示および送信が可能のため、Web アプリケーションの実装時に、これは問題にはなりません。

文書データの場合、この変換を異なる方法で処理することができます。未加工のネイティブ文書データを要求すると、ODWEK は変更を加えていない形式でデータを戻します。すなわち、アーカイブされたときと同じコード・ページで戻します。アプレットを使用してその行データを表示するように要求すると、ODWEK は UTF-8 の ASCII データをアプレットに送信しますが、アプレット呼び出しコードを含む標準の HTML コードのみが戻されます。ASCII 変換を要求すると、ODWEK は、オリジナルの AFP または行データ文書を UTF-8 ASCII 変換した表記にして戻します。他のほとんどの文書タイプについて、ODWEK は OnDemand Windows クライアントと同じ動作をします。すなわち、データをネイティブ・フォーマットでそのまま受け渡します。

ICU 変換ライブラリー

OnDemand は、1 つのコード・ページから別のコード・ページへの変換またはマッピングを行う必要がある場合があります。この変換またはマッピングは、UNICODE 国際コンポーネント (ICU) と呼ばれる標準コンポーネントにより実行されます。

UNICODE 国際コンポーネント (ICU) とは、IBM および他の会社により開発されたオープン・ソース・プロジェクトです。これは、Java および C で使用可能なライブラリーで、国際化対応の目的で使用されます。ICU は、異なるコード・ページ間の文字変換などのサービスや、言語に依存する照合、検索、正規化、およびローケルの情報を提供しています。

以前のバージョンの OnDemand は、ICONV ライブラリー、すなわち別の文字セット変換エンジンを使用していました。OnDemand では、バージョン 7.1.2.1 から ICU を使用しています。ICONV は、UNIX[®] 環境で主に配布されている文字エン

コード・ライブラリーです。例えば、これは、ほとんどの Linux ディストリビューションで GNU C ライブラリーに含まれています。

OnDemand は、さまざまなロケーションで、コード・ページ変換とテキスト操作のために ICU を使用します。

各コンポーネントは、異なるユース・ケースに ICU を使用しますが、すべてのコンポーネントでコード・ページ変換を行います。これにより、OnDemand インフラストラクチャーの他の部分との通信が可能になります。

- OnDemand サーバーは、索引およびアノテーションの通信用に ICU 機能を使用します。すべての TCP/IP トラフィックは UTF-16 ですが、索引データは、データベース・コード・ページのフォーマットです。OnDemand サーバーは、ICU を使用してデータベースまたはインスタンスのコード・ページと UTF-16 の間のデータ変換を行います。
- ODWEK では、UTF-16 の TCP/IP データを UTF-8 に変換するために ICU 機能が必要です。また、行データ Java アプレットで行データを使用しても、データは元のコード・ページから、アプレットが使用する UTF-8 に変換されます。他の文書は、コード・ページを変更されずに受け渡されます。
- 標準の OnDemand クライアントは、ICU 機能を使用して AFP と行データを変換し、内部ビューアーで正しく表示できるようにします。データは、ローカルの Windows コード・ページに変換されます。
- AFP プラグインは、OnDemand クライアント内で AFP ビューアーと同様に動作するため、AFP データをローカルの Windows コード・ページに変換するための独自の ICU ライブラリーを含んでいます。

付録 J. 問題判別ツール

表 16 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 16. 問題判別ツール

ツール	目的	使用可能にする方法
HTML Output	ODWEK がブラウザに戻す HTML のコピーを保管します。	ブラウザの「ファイル」メニューから「名前をつけて保存」を選択します。

表 16. 問題判別ツール (続き)

ツール	目的	使用可能にする方法
Content Manager OnDemand Web Enablement Kit シス テム・トレース	アクセス情報、エラー、およびサーバー情報を保管します。	以下のタスクを実行します。 1. ARSWWW.INI ファイル (CGI および Java サブレットの場合) または ARSWWW.PROPS ファイル (Java API の場合) の DEBUG セクションで、必要なトレース・レベルに応じて TRACE パラメーターを 1、2、3、または 4 に設定します。Java API の場合、適切な ODConfig コンストラクターを使用してこれを指定してください。ODWEK が生成するトレース・ファイルは、ARSWWW.TRACE という名前で、TRACEDIR 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/LOGS です。) 重要: CGI および Java サブレットで、これを指定する場合には、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。HTTP サーバーごとに、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があります。 重要: 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。
AFP Web Viewer Trace Facility	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。	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。 [Misc] ViewTraceFile=d:%temp%afpplgin.log Trace=TRUE 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。

表 16. 問題判別ツール (続き)

ツール	目的	使用可能にする方法
OnDemand System Log	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。	以下のタスクを実行します。 <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。

Java ダンプ

Java プロセスの実行時に、一部の Java 仮想マシン (JVM) が予想通りに反応せず、長時間あるいは JVM がシャットダウンするまでフリーズしていると思われる場合があります。こうした問題の根本原因を判別するのは容易ではありません。

Java プロセスが応答しないときをトリガーに javacore を起動することで、JVM および特定の時点で捉えた Java アプリケーションに関連する診断情報を収集することができるとは限りません。例えば、オペレーティング・システム、アプリケーション環境、スレッド、ネイティブ・スタック、ロック、および、メモリーなどに関する情報が収集できます。収集できる正確な内容は、アプリケーションが実行されているプラットフォームによって異なります。

一部のプラットフォームおよび場合により、javacore は「jvaddump」と呼ばれています。javacore を作成するコードは、JVM に含まれています。これは、環境変数とランタイム・スイッチを使用することで制御できます。デフォルトで、javacore は JVM が予期しない終了をしたときに起動されます。また、javacore は、特定の信号を JVM に送信することでもトリガーできます。javacore あるいは jvaddump は、Sun Solaris JVM にもありますが、javacore の内容の多くは、IBM によって追加されるため、IBM JVM のみに存在すると言えます。

IBM Thread and Monitor Dump Analyzer for Java は、javacore を分析し、モニター・ロックとスレッド・アクティビティーを診断して、ハングやデッドロック、資源競合あるいはモニター・ボトルネックの根本原因を特定することができます。

IBM Thread and Monitor Dump Analyzer

このテクノロジーは、各スレッドを分析して、診断情報を提供します。例えば、現行スレッド情報、javacore を引き起こした信号、Java ヒープ情報 (Java ヒープの最大サイズ、Java ヒープの初期サイズ、ガーベッジ・コレクターのカウンター、割り振り失敗カウンター、Java ヒープの空きサイズ、Java ヒープの割り振り済みサイズ)、実行可能なスレッド数、合計スレッド数、ロックされているモニター数、およびデッドロック情報などを提供します。

さらに、IBM Thread and Monitor Dump Analyzer for Java Technology は、ヒューリスティック分析エンジンに基づいて、Java ヒープ・クラスターの推奨サイズ (IBM SDK バージョン 1.4.2 およびバージョン 1.3.1 SR7 以降にのみ適用可能) を提示します。

IBM Thread and Monitor Dump Analyzer for Java は、各 javacore を比較して、スレッドのプロセス ID 情報、最初の javacore のタイム・スタンプ、最後の javacore のタイム・スタンプ、1 分間あたりのガーベッジ・コレクション数、1 分間あたりの割り振り失敗数、最初の javacore から最後の javacore までの時間、ハングと思われる現象の発生回数、ハングと思われる現象のリストを提供します。

また、このテクノロジーは、javacore 内のすべてのモニター情報を比較して、デッドロックと資源競合またはモニター・ボトルネックを検出します。

IBM Thread and Monitor Dump Analyzer に関して詳しくは、<http://www.alphaworks.ibm.com/tech/jca> (英語) を参照してください。

Java 診断コマンド

jmap

jmap は、Java Development Kit (JDK) の Solaris オペレーティング環境および Linux のリリースに含まれているコマンド行ユーティリティです (Windows のリリースには含まれていません)。このユーティリティは、実行中の JVM またはコア・ファイルのメモリー関連の統計情報を印刷します。コマンド行オプションを指定しないで jmap を使用すると、Solaris の pmap ユーティリティの出力と類似した、ロード済み共有オブジェクトのリストをプリントします。詳しい情報が必要な場合は、-heap、-histo、または -permstat オプションを使用してください。

-heap -heap オプションを使用すると、ガーベッジ・コレクターの名前、アルゴリズム固有の詳細情報 (例えば、並列ガーベッジ・コレクションに使用されるスレッド数、ヒープ構成情報、およびヒープ使用サマリーなど) を含む情報が取得できます。

-histo -histo オプションを使用すると、クラス全体のヒープのヒストグラムを取得できます。各クラスごとに、ヒープ内のインスタンス数、それらのオブジェクトで消費されるメモリーの合計容量 (バイト単位)、およびクラスの完全修飾名が印刷されます。ヒストグラムは、ヒープがどのように使用されているかを理解する場合に有効です。

-permstat

例えば、Java Server Pages や Web コンテナなど、多数のクラスを動的に生成してロードするアプリケーションの場合、永続世代領域のサイズを構成することが重要になる場合があります。アプリケーションがロードするクラスが多過ぎると、OutOfMemoryError 例外がスローされます。jmap コマンドに -permstat オプションを使用すると、永続世代領域にあるオブジェクトの統計情報を取得できます。

jstat

jstat ユーティリティーは、HotSpot JVM の組み込み計測機能を使用して、実行中のアプリケーションのパフォーマンスや資源消費量に関する情報を提供します。この jstat ユーティリティーを使用すると、パフォーマンス問題、特に、ヒープ・サイズ設定やガーベッジ・コレクションに関連した問題を診断することができます。数多くのオプションの中には、ガーベッジ・コレクションの動作やさまざまな世代領域の容量や使用量に関する統計情報を印刷できるものもあります。

HPROF: ヒープ・プロファイラー

HPROF は、JDK バージョン 5.0 に同梱されている単純なプロファイラー・エージェントです。これは、動的にリンクされるライブラリーで、Java Virtual Machine Tools Interface (JVM TI) を使用して JVM とのインターフェースを取ります。プロファイル情報をファイルまたはソケットに ASCII あるいはバイナリー・フォーマットで書き込みます。プロファイラーのフロントエンド・ツールを使用すれば、この情報をさらに処理することができます。

HPROF は、CPU 使用量、ヒープ割り振り統計情報、およびモニター競合プロファイルを表示します。さらに、完全なヒープ・ダンプを出力し、Java 仮想マシン内のすべてのモニターおよびスレッドの状態を報告します。HPROF は、パフォーマンス、ロック競合、メモリー・リーク、その他の問題の分析を行う際に有効です。

HAT: Heap Analysis Tool

Heap Analysis Tool (HAT) は、非意図的オブジェクト保持のデバッグに役立ちます。この非意図的オブジェクト保持というのは、もはや必要ないオブジェクトですが、実行中のオブジェクトから何らかのパスによって参照されているため、ライブ状態のまま保たれているオブジェクトを説明する用語です。HAT は、HPROF を使用して生成されるヒープ・スナップショットでそのオブジェクトのトポロジーを参照できる便利な方法を提供します。このツールにより、例えば、「ルートセットからこのオブジェクトまでのすべての参照パスを表示する」などの数多くの照会を使用することができます。

Diagnostic Tool for Java Garbage Collector

このテクノロジーは、IBM Java 仮想マシン (JVM) の使用時にガーベッジ・コレクターに影響するパラメーターの最適化のための診断ツールです。

Java 上で実行されるアプリケーションは、IBM の Java Development Kit およびランタイム環境で記憶域管理機能として働く、ガーベッジ・コレクションに Java ヒープを使用します。

Java のエンタープライズ・アプリケーションあるいはスタンドアロンのアプリケーションにおけるガーベッジ・コレクターの活動を反映したデータを分析することは、JVM 下で実行するタスクの最適化には不可欠です。例えば、ガーベッジ・コレクション循環の頻度、ガーベッジ・コレクションのさまざまなフェーズで要した時間、プロセスに必要なヒープ・メモリーの量、ガーベッジ・コレクション発生の元になる割り振り失敗の特性、および望ましくないスタック・オーバーフローの存在などの問題を考慮に入れて Java アプリケーションのパラメーターの最適化とボトルネックの予防を行う必要があります。

Diagnostic Tool for Java Garbage Collectorは、「詳細な」ガーベッジ・コレクションの出力を読み取り、テキストおよび図形(グラフ)による視覚化を行い、関連統計を生成して、JVM 下で実行中のアプリケーションに対するガーベッジ・コレクションの特性を調査する場合に役立ちます。このツールは、WebSphere Application Server によってホストされる、アクセス量が非常に多いエンタープライズ・アプリケーションのガーベッジ・コレクター・アクティビティを検討する場合に特に適しています。このツールには、同時に 2 つ以上のファイルをロードできる「複数ファイル分析」方式があるため、1 つの WebSphere クラスタ内にある複数のアプリケーション・サーバーの動作を比較することが可能です。

Diagnostic Tool for Java Garbage Collector バージョン 1.3 には、IBM JVM バージョン 1.5.0、IBM JVM バージョン 1.4.2、および IBM JVM バージョン 1.2.2 用の 3 つの組み込みパーサー、ツールの機能評価用サンプル・ファイル、および詳細な資料が同梱されています。各パーサーは、他のバージョンの IBM JVM によって作成されたデータの解析も行える可能性が高く、例えば、JVM バージョン 1.4.2 のパーサーは、JVM バージョン 1.3.x でも適切に動作します。サポートされていないバージョンの IBM JVM については、GCParser インターフェースを実装して Java の別のパーサーをコーディングできるように、詳細な資料がツールに組み込まれています。資料には、新規にパーサーをコーディングする際に必要な前提やルールが詳細に記述されています。

詳しい情報の参照、および製品のダウンロードは、<http://www.alphaworks.ibm.com/tech/gcdiag> (英語) にアクセスしてください。

HeapAnalyzer

HeapAnalyzer は、ヒューリスティック検索エンジンと Java アプリケーションの Java ヒープ・ダンプの分析によって、可能性のある Java ヒープ・リーク領域を検出します。

Java ヒープ領域は、オブジェクト、配列、およびクラスを定義します。ガーベッジ・コレクターがヒープ内に記憶域を割り振ると、あるオブジェクトに対する参照が、アクティブ状態の JVM のどこかに存在している限りそのオブジェクトは有効な状態が続きます。すなわち、オブジェクトは、到達可能です。オブジェクトがアクティブ状態から参照されなくなると、それはガーベッジになり、再利用のためにレクラメーション処理が可能になります。このレクラメーションが発生すると、ガーベッジ・コレクターは、実行可能な終了処理を行い、オブジェクトに関連付けられているすべての内部 JVM 資源がその資源プールに確実に戻されるようになります。Java ヒープ・ダンプは、特定の時刻における Java ヒープのスナップショットです。

HeapAnalyzer は、Java ヒープ・ダンプを解析し、有向グラフを作成し、それを有向ツリーに変換し、ヒューリスティック検索エンジンを実行することで、Java ヒープ・ダンプの分析をします。

以下に HeapAnalyzer の機能の例を示します。

- Java ヒープのリークではないかと思われる現象のリスト
- kCluster のサイズの推奨
- 割り振り済みオブジェクト/クラス/配列の間のギャップのリスト

- Java オブジェクト/クラス/配列の検索エンジン
- タイプ名別のオブジェクト/クラス/配列のリスト
- オブジェクト名別のオブジェクト/クラス/配列のリスト
- アドレス別のオブジェクト/クラス/配列のリスト
- サイズ別のオブジェクト/クラス/配列のリスト
- 子のサイズ別のオブジェクト/クラス/配列のリスト
- 子の数によるオブジェクト/クラス/配列のリスト
- 頻度別のオブジェクト/クラス/配列のリスト
- サイズ別の使用可能なヒープ・スペースのリスト
- Java ヒープ・ダンプのツリー・ビュー
- 処理済み Java ヒープ・ダンプのロード/保存

HeapRoots

HeapRoots は、ヒープ・ダンプの分析により Java アプリケーションのメモリー・リンクをデバッグするツールです。

Java 仮想マシン (JVM) は、(ヒープと呼ばれる) ランタイム・データ領域を保守して、すべてのクラス・インスタンスと配列オブジェクトの割り振りを行います。オブジェクトのヒープ・ストレージは、ガーベッジ・コレクターとして知られる記憶域管理システムによって自動的に再利用されます。アプリケーションが、ガーベッジ・コレクターによって使用可能にされる領域より多くのヒープ・スペースを要求すると、JVM は `OutOfMemoryError` をスローします。

HeapRoots は、JVM のガーベッジ・コレクションで収集されたヒープ内のオブジェクトに関する情報を含んだ (通常テキスト・ファイルの) ヒープ・ダンプを分析します。

一部の IBM VM (IBM Developer Kits for Windows Java Edition) は、オンデマンドで、ヒープ・ダンプを生成することができます。すなわち、メモリー不足状態をトリガーにしてヒープ・ダンプを起動することもできます。

HeapRoots は、これらのヒープ・ダンプ・ファイルをロードし、データ分析用のコマンドを提供します。これらのコマンドは、データに対してアルゴリズムを実行したり、あるいは、データに関する情報の照会を行います。HeapRoots には、コマンド行の対話式インターフェースがあり、そこにコマンドを入力して結果を表示させることができます。分析の例には、次のものがあります。

- 個々のオブジェクトの検索またはフィルタリング
- さまざまなタイプのオブジェクトのサマリーまたは作表
- ヒープ・アドレス・スペースに関する統計 (例えば、オブジェクト間のギャップなど)
- オブジェクトに対する内向きおよび外向きの参照
- 2 オブジェクト間のパス
- 参照をたどることによるソース・オブジェクトまたはルート・オブジェクトからのヒープの探索
- 1 オブジェクトにより到達可能なオブジェクトの計算

|

- 1 オブジェクトにより有効に保持されているオブジェクトの計算

付録 K. Apache HTTP Server を使用した CGI のマルチリンガル・サポート機能

OnDemand Web Enablement Kit (ODWEK) Common Gateway Interface (CGI) プログラムに、マルチリンガル・サポート機能が追加されました。Apache HTTP Server では、マルチリンガル・サポート機能により、単一の IBM i サーバー上で、異なる言語で実行される OnDemand Common Server インスタンスをサポートできるようになりました。

ソフトウェア前提条件

以下は、CGI に対する ODWEK マルチリンガル・サポート機能に必要なソフトウェアです。

- OnDemand Common Server 環境。

実装

重要: 以下の情報については、画面取りも含めた同様のバージョンが、OnDemand Support Web ページ (<http://www.ibm.com/software/data/ondemand/400/support.html>) で「ODWEK multilingual」というストリングを検索することにより参照できます。

マルチリンガル・サポート機能を実装するには、以下のステップを実行します。後出の例では、CCSID 1147 を使用して、フランス語の HTTP サーバーを作成します。

1. 新規の Apache HTTP Server を作成します。
 - a. `http://i_hostname.company.com:2001/HTTPAdmin` (`i_hostname.company.com` は使用する IBM i サーバーの名前) で HTTP 管理インターフェースをオープンします。
 - b. 「HTTP サーバーの作成 (Create HTTP Server)」をクリックします。
 - c. 新規 HTTP サーバーの名前と説明を指定して、「次へ」をクリックします。
 - d. 新規 HTTP サーバーのルート・ディレクトリーを指定して、「次へ」をクリックします。
 - e. 新規 HTTP サーバーの文書ルート・ディレクトリーを指定して、「次へ」をクリックします。
 - f. 新規 HTTP サーバーが listen するポートの番号を指定します。(このポートは、システム上の他の TCP/IP サービスが使用するものであってはなりません。) 「次へ」をクリックします。
 - g. アクセス・ログを保管するかどうかを指定します。
 - h. ログ・ファイルを保管する日数を指定します。
 - i. 設定を確認し、「完了 (Finish)」をクリックして新規 HTTP サーバーを作成します。(いずれかの設定を変更する場合は、「戻る (Back)」をクリックして前の画面に戻ります。)

2. 次のようにプログラムを実行して ODWEK ディレクトリーを作成します。

```
CALL PGM(QRDARS/QRLMINSTW) PARM('ccsid')
```

例えば、次のコマンドは、

```
CALL PGM(QRDARS/QRLMINSTW) PARM('1147')
```

/QIBM/UserData/OnDemand/www/1147 ディレクトリーを作成します。このディレクトリーは、以下の項目を含みます。

- キャッシュ
- ログ
- tmp
- アプレット
- イメージ
- サンプル
- arswwww.ini

3. 新規 HTTP サーバー用の httpd.conf ファイルを更新して、以下を行います。

- a. CCSID を、アクセスされる OnDemand インスタンスと同じ CCSID にします。(例えば、 OnDemand インスタンスが 1147 という CCSID を使用する場合、httpd.conf ファイルの DefaultFsCCSID は 1147 でなければなりません。)

- b. httpd.conf ファイルに、ODWEK が必要とするその他の更新を行います。

以下の例では、HTTP サーバーの CCSID は 1147 です。HTTP サーバーのルート・ディレクトリーは、/www/odfra です。ODWEK サポート用に追加する行は、コメントで区切られています。太字のイタリック体で強調表示されているテキストは、ご使用の環境に合わせて変更する必要があります。

```
# Configuration originally created by Create HTTP Server wizard on Thu Oct 26 09:31:27 EDT 2006
Listen *:2114
DocumentRoot /www/odfra/htdocs
Options -ExecCGI -FollowSymLinks -SymLinksIfOwnerMatch -Includes -IncludesNoExec -Indexes -MultiViews
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%{Cookie}n \"%r\" %t" cookie
LogFormat "%{User-agent}i" agent
LogFormat "%{Referer}i -> %l" referer
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log combined
LogMaint logs/access_log 7 0
LogMaint logs/error_log 7 0
# Added Lines for ODWEK
DefaultFsCCSID 1147
DefaultNetCCSID 1208
CGIConvMode EBCDIC
# End Added Lines for ODWEK
SetEnvIf "User-Agent" "Mozilla/2" nokeepalive
SetEnvIf "User-Agent" "JDK/1\." force-response-1.0
SetEnvIf "User-Agent" "Java/1\." force-response-1.0
SetEnvIf "User-Agent" "RealPlayer 4\." force-response-1.0
SetEnvIf "User-Agent" "MSIE 4\." nokeepalive
SetEnvIf "User-Agent" "MSIE 4\." force-response-1.0
# Added Lines for ODWEK
LimitRequestBody 102400
ServerName iseries_hostname.company.com
UseCanonicalName Off
AccessFileName .htaccess
ScriptLogLength 200
ScriptLog /www/odfra/logs/cgi_log
Alias /logon /QIBM/UserData/OnDemand/www/1147/samples/logon.htm
AliasMatch ~/images/(.*)$ /QIBM/UserData/OnDemand/www/1147/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/ARS3WCG1.PGM
ScriptAliasMatch ~/scripts/arswww\.cgi/(.*)$ /QSYS.LIB/QRDARS.LIB/ARS3WCG1.PGM
AddType www/unknown cab
AddType www/unknown jar
# End Added Lines for ODWEK
<Directory />
Order Deny,Allow
Deny From all
# Added Lines for ODWEK
```

```

Options None
Options +ExecCGI
<LimitExcept GET HEAD OPTIONS TRACE POST PUT>
</LimitExcept>
# End Added Lines for ODWEK
</Directory>
<Directory /www/odfra/htdocs>
Order Allow,Deny
Allow From all
</Directory>
# Added Lines for ODWEK
<Directory /QIBM/ProdData/OnDemand/www/applets/>
Order allow,deny
Allow from ALL
</Directory><Directory /QIBM/ProdData/OnDemand/www/1147/samples/>
Order allow,deny
Allow from All
Options +FollowSymLinks
<Directory><Directory /QIBM/ProdData/OnDemand/www/1147/images/>
Order allow,deny
Allow from All
Options +FollowSymLinks
</Directory>
</Directory><Directory /QIBM/ProdData/OnDemand/www/1147/>
Order allow,deny
Allow from All
Options +FollowSymLinks
</Directory>
<Directory /www/odfra/>
Order allow,deny
Allow from All
</Directory>
<Directory /QSYS.LIB/QRDARS.LIB>
Order allow,deny
Allow from ALL
Options +ExecCGI
</Directory>
# End Added Lines for ODWEK

```

4. /QIBM/UserData/OnDemand/www/ccsid ディレクトリーの ARSWWW.INI ファイルを編集します。

```
edtf '/QIBM/UserData/OnDemand/www/ccsid/arswww.ini'
```

ここで、*ccsid* は必須の CCSID です。次に例を示します。

```
edtf '/QIBM/UserData/OnDemand/www/1147/arswww.ini'
```

必要に応じて、以下の行を変更します。

```

TraceDir=/QIBM/UserData/OnDemand/www/ccsid/logs
Language=lang
CodePage=ccsid
TemplateDir=/QIBM/UserData/OnDemand/www/ccsid/samples
CacheDir=/QIBM/UserData/OnDemand/www/ccsid/CACHE

```

次に例を示します。

```

TraceDir=/QIBM/UserData/OnDemand/www/1147/logs
Language=FRA
CodePage=1147
TemplateDir=/QIBM/UserData/OnDemand/www/1147/samples
CacheDir=/QIBM/UserData/OnDemand/www/1147/CACHE

```

以下の行を、この HTTP サーバーがアクセスするインスタンスを指定するように変更します。

```

[@SRV@_QUSROND]
HOST=Sxxxxxxx
PROTOCOL=0
PORT=1450

```

次に例を示します。

```

[@SRV@_ONDFRA]
HOST=i_hostname.company.com
PROTOCOL=0
PORT=1514

```

ここで、*i_hostname.company.com* は、ご使用の IBM i システムの名前です。

5. 必要に応じて、ARSWWW.INI ファイルの他の構成セクションを変更します。
6. logon.htm ファイルを、デフォルト・インスタンス名を指定するように編集します。

```
edtf '/QIBM/UserData/OnDemand/www/ccsid/samples/logon.htm'
```

次に例を示します。

```
edtf '/QIBM/UserData/OnDemand/www/1147/samples/logon.htm'
```

以下を変更します。

```
<b>Server Name:</b><input type=text name=_server value=QUSROND>
```

次に例を示します。

```
<b>Server Name:</b><input type=text name=_server value=ONDFRA>
```

7. 必要に応じて、logon.htm ファイルの他の構成セクションを変更します。
8. 新規 HTTP サーバーを始動します。
9. 構成をテストします。
 - a. 次のログオン・ページを開きます。

```
http://iseries_hostname.company.com:port_number/logon
```

次に例を示します。

```
http://rdr400m.raleigh.ibm.com:2114/logon
```

- b. ログオンします。
- c. フォルダをオープンします。
- d. 検索します。
- e. 必要に応じて、次のアーカイブ・データを表示します。
 - 行データ
 - AFPDS
 - イメージ
 - PDF

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。IBM 製品、プログラムまたはサービスに代えて、IBM の知的所有権を侵害することのない機能的に同等の製品、プログラムまたはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒242-8502
神奈川県大和市下鶴間1623番14号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

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 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。サンプル・プログラムは、現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、このサンプル・プログラムの使用から生ずるいかなる損害に対しても、一切の責任を負いません。

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、IBM ロゴ、および ibm.com[®] は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe、Acrobat、Portable Document Format (PDF)、および PostScript[®] は、Adobe Systems Incorporated の米国およびその他の国におよび商標または登録商標です。

Intel® および Pentium® は Intel Corporation の米国およびその他の国における商標です。

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です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、および Windows NT® は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は Open Group の米国およびその他の国における登録商標です。

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 121

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 20

インストール 57

使用可能 31

説明 1

ディレクトリー 31

要件 58

ラージ・オブジェクトのサポート 29, 170

APPLETCACHEDIR パラメーター 20

暗号化 43

一時記憶域 26

一時的作業ディレクトリー 26

イメージ・ディレクトリー 24

印刷

サーバー 46

Java API 153

インストール

アプレット 57

カスタマイズ 60

チェックリスト 13

プラグイン 57

ユーザー定義ファイル 60

ユーザー・ワークステーション 57

要件 15

AFP Web Viewer 57

AFP2HTML Java アプレット 57

ARSWWW.INI ファイル 17

HTTP サーバー 15

IBM i 16

Image Web Viewer 57

Java アプレット 57

Line Data Java アプレット 57

ODWEK ソフトウェア 15

エラー 117, 185

エラー・メッセージ

区切り文字付き ASCII 出力 179

オペランド、Java API 126

[カ行]

解説

サーブレット 109

API 75, 109, 111

CGI API 75

Java API 111

Java サーブレット 109

概要 1

各国語サポート 181

関数

Add Annotation 8

Annotations 8, 9

Change Password 8

delete annotations 9

Document Hit List 8

Logoff 8

Logon 8

Print Document 9

Retrieve Document 9

Search Criteria 9

Server Print Document 9

Update Document 9

View Annotations 9

キャッシュ記憶機構 21, 22, 23

キャッシュ文書 21

キャッシュ・サイズ 22

キャッシュ・ディレクトリー 21

行データ文書

媒体タイプ 35

表示 44

変換 44

MIME コンテンツ・タイプ 35

行データ・ビューアー

構成 67

ODApplet.jre.path.IE パラメーター 45

ODApplet.jre.path.NN パラメーター 45

ODApplet.jre.version パラメーター 45

ODApplet.version パラメーター 46

許可 15

区切り文字 39

区切り文字付き ASCII 出力

エラー・メッセージ 179

区切り文字 39

形式 175

メッセージ 179

Annotations 178

Document Hit List 177

Logon 176

OnDemand が生成する 175

Search Criteria 176

区切り文字付き ASCII 出力 (続き)
View Annotations 178
クラス 113
権限 15
言語 25, 181
検索、フォルダーの 126, 131, 135, 138, 146
検索、文書の 149
検索値、Java API 126, 131
コード・ページ 23, 181
構成
AFP2HTML 構成ファイル 167
AFP2PDF 構成ファイル 171
ARSWWW.INI ファイル 17
HTTP サーバー 15
ODWEK ソフトウェア 15

[サ行]

サーバー
Java API 120, 121
サーバー印刷
関数の説明 9
関数呼び出しのサンプル 96
使用可能 46
パラメーター 92
API 92
Java API 153
サーバーへの接続 120
サーバー・アクセス・リスト 28
サーバー・セキュリティー 10, 27
サーブレット
解説 109
最大ヒット 44
サンプル・アプリケーション 53
サンプル・コード 113
出力区切り文字 39
準備、OnDemand Internet Connection を使用するための 1
照会結果 44
診断情報 117
セキュリティー 10, 27, 43
接続、サーバーへの 121
接続タイプ
Java API 121
切断、サーバーからの 121
設定、パスワードの 164
説明、OnDemand Internet Connection の 1

[タ行]

タイムアウト 27
注 41, 47
通信プロトコル 18, 19
データ・セキュリティー 10
ディレクトリー許可 15
デバッグ・セクション 48

デフォルト演算子、Java API 144
デフォルト・ブラウザー・オプション 40
添付ファイル 37, 38
トランザクション
タイム・スタンプ 28
取り消し、検索の 135
取り出し
文書 42
トレース、問題の 185
トレースおよび診断情報 117

[ハ行]

パスワード
Java API 120, 121, 164
パッケージ階層、Java 113
パラメーター
ADDEXTENSION 40
ADDFIELDSTODOCID 40
ADDNOTES 41
AFP2HTML セクション 29
AFP2PDF セクション 30
AFPVIEWING 41
APPLETCACHEDIR 20
APPLETDIR 20
ATTACHMENT IMAGES セクション 37
AUTODOCRETRIEVAL 42
BEGIN 39
CACHEDIR 21
CACHEDOCS 21
CACHEMAXTHRESHOLD 22
CACHEMINTHRESHOLD 22
CACHESIZE 22
CACHEUSERIDS 23
CODEPAGE 23
CONFIGFILE 29, 31
CONFIGURATION セクション 19
DOCSIZE 24
EMAILVIEWING 42
ENCRYPTCOOKIES 43
ENCRYPTURL 43
END 39
FOLDERDESC 44
HOST 19
IMAGEDIR 24
INSTALLDIR 30, 31
LANGUAGE 25
LINEVIEWING 44
LOG 186
MAXHITS 44
NOLINKS 45
ODApplet.jre.path.IE 45
ODApplet.jre.path.NN 45
ODApplet.version 46
PORT 18, 19
PROTOCOL 18, 19

パラメーター (続き)

- REPORTSERVERTIMEOUT 27
- SECURITY セクション 27
- SEPARATOR 39
- SERVERACCESS 28
- SERVERPRINT 46
- SERVERPRINTERS 46
- SHOWDOCLOCATION 46
- ShowSearchString 26
- TEMPDIR 26
- TEMPLATEDIR 27
- TRACE 48
- TRACEDIR 49
- USEEXECUTABLE 30, 32
- VIEWNOTES 47
- @SRV@_DEFAULT セクション 17
- @SRV@_server セクション 18
- 非 HTML 出力 39, 175
- 非アクティブ・タイムアウト 27
 - およびタイム・スタンプ 28
- 表示、文書の場所の 46
- 表示値、Java API 126
- 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 124
- フォルダー、Java API を使用した検索 126, 131, 135, 138, 146
- フォルダー記述、Java API 142
- フォルダー基準、Java API 144
- フォルダー内のアプリケーション・グループ
 - Java API 124
- フォルダー名、Java API 142
- フォント
 - マッピング 64
 - AFP 64
 - TrueType 64
- ブラウザー
 - サポートされている 58
- ブラウザー・オプション
 - browser セクション 47
 - DEFAULT BROWSER セクション 40
- プラグイン
 - インストール 57
 - 説明 1
- プログラミング・ガイド
 - API 113
 - Java API 113
- プロトコル 18, 19
- 文書
 - キャッシュ記憶機構 21
 - 行データ 44
 - 取り出し 42
 - 媒体タイプ 32
 - 表示 41, 42, 44
 - 変換 41, 42, 44
 - リンク 45
 - AFP 41

文書 (続き)

- EMAIL 42
 - Java API を使用した印刷 153
 - Java API を使用した更新 161
 - MIME コンテンツ・タイプ 32
- 文書、Java API 146, 149
- 文書タイプ、Java API 126
- 文書の場所 46
- ヘルプ 185
- 変更、パスワードの 164
- ポート
 - Java API 121
- ポート番号 18, 19
- ホスト名 19
- 本書について 1

[マ行]

- マッピング、AFP フォントの 64
- メソッド属性、フォーム・タグの 10
- メッセージ 25
 - 区切り文字付き ASCII 出力 179
- メディア・タイプ/サブタイプ 32
- 問題判別 185

[ヤ行]

- ユーザー ID
 - キャッシュ記憶機構 23
 - Java API 120, 121
- ユーザー定義ファイル
 - インストール 60
- 要件
 - キャッシュ記憶機構 15
 - サーバー 15
 - 文書キャッシュ 15
 - AFP2HTML Java アプレット 16
 - AFP2PDF Transform 16
 - AFP2WEB Transform 16
 - HTTP サーバー 15
 - Java API 1

[ラ行]

- ラージ・オブジェクト 29, 170
- リンク 45
- 例 113
- 例外処理 118
- ローカル・ディレクトリー
 - Java API 121
- ログイン 48, 49, 186
- ログ・ファイル 48, 49, 186

[数字]

2 バイト文字セット言語 181

A

Add Annotation

- 関数の説明 8
- 関数呼び出しのサンプル 78
- パラメーター 76
- API 76

ADDEXTENSION パラメーター 40

ADDFIELDSTODOCID パラメーター 40

ADDNOTES パラメーター 41

AFP Web Viewer

- インストール 57
- インストールのカスタマイズ 60
- 構成 59
- 説明 1
- フォント 64
- マッピング、AFP フォントの 64
- ユーザー定義ファイル 60
- ユーザー定義ファイルのインストール 60
- 要件 58
- AFP フォント 64

AFP フォント

- マッピング 64

AFP 文書

- 媒体タイプ 34
- 表示 41
- 変換 41
- MIME コンテンツ・タイプ 34

AFP2HTML Java アプレット

- インストール 57
- 説明 1, 7
- 要件 16, 58
- ラージ・オブジェクトのサポート 29, 170
- APPLETCACHEDIR パラメーター 20

AFP2HTML 構成ファイル 167

AFP2HTML セクション 29

AFP2PDF Java アプレット

- 使用可能 31
- ディレクトリー 31

AFP2PDF Transform

- インストール 16
- 構成 171
- 使用可能 30

AFP2PDF 構成ファイル 171

AFP2PDF セクション 30

AFP2WEB Transform

- インストール 16
- 構成 167
- 使用可能 29

AFPVIEWING パラメーター 41

- Annotations 41, 47
- 関数の説明 8, 9

Annotations (続き)

- 関数呼び出しのサンプル 78, 108
- 区切り文字付き ASCII 出力 178
- パラメーター 76, 107
- API 76, 107
- Java API 155, 157, 159

Apache HTTP server

- 構成ファイル 173

API

- 解説 75, 109, 111
- クラス 113
- サーバー印刷 92
- サンプル・コード 113
- 診断情報 117
- トレースおよび診断情報 117
- パッケージ化 113
- プログラミング・ガイド 113
- 例 113
- 例外処理 118
- Add Annotation 76
- Annotations 76, 107
- CGI API リファレンス 75
- Change Password 79
- Document Hit List 82
- Java API プログラミング・ガイド 113
- Java API リファレンス 111
- Logoff 87
- Logon 89
- Print Document 92
- Retrieve Document 97
- Search Criteria 102
- Update Document 105
- View Annotations 107

APPLETCACHEDIR パラメーター 20

APPLETDIR パラメーター 20

ARSWWW.INI ファイル

- 構成 17
- 指定 17
- デバッグ・セクション 48
- ブラウザ・オプション 47
- ADDEXTENSION パラメーター 40
- ADDFIELDSTODOCID パラメーター 40
- ADDNOTES パラメーター 41
- AFP2HTML セクション 29
- AFP2PDF Transform 30
- AFP2PDF セクション 30
- AFP2WEB Transform 29
- AFPVIEWING パラメーター 41
- APPLET パラメーター 20
- APPLETCACHEDIR パラメーター 20
- ATTACHMENT IMAGES セクション 37
- AUTODOCRETRIEVAL パラメーター 42
- BEGIN パラメーター 39
- browser セクション 47
- CACHEDIR パラメーター 21
- CACHEDOCS パラメーター 21

ARSWWW.INI ファイル (続き)

- CACHEMAXTHRESHOLD パラメーター 22
- CACHEMINTHRESHOLD パラメーター 22
- CACHESIZE パラメーター 22
- CACHEUSERIDS パラメーター 23
- CODEPAGE パラメーター 23
- CONFIGFILE パラメーター 29, 31
- CONFIGURATION セクション 19
- DEFAULT BROWSER セクション 40
- DOCSIZE パラメーター 24
- EMAILVIEWING パラメーター 42
- ENCRYPTCOOKIES パラメーター 43
- ENCRYPTURL パラメーター 43
- END パラメーター 39
- FOLDERDESC パラメーター 44
- HOST パラメーター 19
- IMAGEDIR パラメーター 24
- INSTALLDIR パラメーター 30, 31
- LANGUAGE パラメーター 25
- LINEVIEWING パラメーター 44
- LOG パラメーター 186
- MAXHITS パラメーター 44
- MIMETYPES セクション 32
- NOHTML セクション 39
- NOLINKS パラメーター 45
- ODApplet.jre.path.IE パラメーター 45
- ODApplet.jre.path.NN パラメーター 45
- ODApplet.jre.version パラメーター 45
- ODApplet.version パラメーター 46
- PORT パラメーター 18, 19
- PROTOCOL パラメーター 18, 19
- REPORTSERVERTIMEOUT パラメーター 27
- SECURITY セクション 27
- SEPARATOR パラメーター 39
- SERVERACCESS パラメーター 28
- SERVERPRINT パラメーター 46
- SERVERPRINTERS パラメーター 46
- SHOWDOCLOCATION パラメーター 46
- ShowSearchString パラメーター 26
- TEMPDIR パラメーター 26
- TEMPLATEDIR パラメーター 27
- TRACE パラメーター 48
- TRACEDIR パラメーター 49
- USEEXECUTABLE パラメーター 30, 32
- VIEWNOTES パラメーター 47
- @SRV@_DEFAULT セクション 17
- @SRV@_server セクション 18

ARSWWW.PROPS 117

ASCII 出力

- エラー・メッセージ 179
- 形式 175
- メッセージ 179
- Annotations 178
- Document Hit List 177
- Logon 176
- OnDemand が生成する 175

ASCII 出力 (続き)

- Search Criteria 176
- View Annotations 178

ATTACHMENT IMAGES セクション 37

AUTODOCRETRIEVAL パラメーター 42

B

BEGIN パラメーター 39

BMP 添付ファイル 38

BMP 文書

- 媒体タイプ 34
- MIME コンテンツ・タイプ 34

browser セクション 47

C

CACHEDIR パラメーター 21

CACHEDOCS パラメーター 21

CACHEMAXTHRESHOLD パラメーター 22

CACHEMINTHRESHOLD パラメーター 22

CACHESIZE パラメーター 22

CACHEUSERIDS パラメーター 23

CGI API

- 解説 75

Change Password

- 関数の説明 8
- 関数呼び出しのサンプル 81
- パラメーター 79
- API 79

CODEPAGE パラメーター 23, 181

CONFIGFILE パラメーター 29, 31

CONFIGURATION セクション 19

Cookies 43

- 要件 58

D

DBCS 181

DEFAULT BROWSER セクション 40

delete annotations

- 関数の説明 9

DOCSIZE パラメーター 24

Document Hit List

- 関数の説明 8
- 関数呼び出しのサンプル 86
- 区切り文字付き ASCII 出力 177
- パラメーター 82
- API 82
- Java API 126, 131, 146, 149

E

EMAIL 文書

- 媒体タイプ 35

EMAIL 文書 (続き)

表示 42

変換 42

MIME コンテンツ・タイプ 35

EMAILVIEWING パラメーター 42

ENCRYPTCOOKIES パラメーター 43

ENCRYPTURL パラメーター 43

END パラメーター 39

F

FOLDERDESC パラメーター 44

G

GET メソッド 10

GIF 添付ファイル 38

GIF 文書

媒体タイプ 34

MIME コンテンツ・タイプ 34

H

HOST パラメーター 19

HTTP サーバー

構成ファイル 173

httpd.conf ファイル 173

HTTP サーバー・オプション

デバッグ 48

デフォルト・ブラウザ 40

ブラウザ 40, 47

AFP2HTML セクション 29

AFP2PDF Transform 30

AFP2PDF セクション 30

AFP2WEB Transform 29

APPLETDIR パラメーター 20

ATTACHMENT IMAGES セクション 37

BEGIN パラメーター 39

CACHEDIR パラメーター 21

CACHEDOCS パラメーター 21

CACHEMAXTHRESHOLD パラメーター 22

CACHEMINTHRESHOLD パラメーター 22

CACHESIZE パラメーター 22

CACHEUSERIDS パラメーター 23

CODEPAGE パラメーター 23

CONFIGFILE パラメーター 29, 31

CONFIGURATION セクション 19

END パラメーター 39

IMAGEDIR パラメーター 24

INSTALLDIR パラメーター 30, 31

LANGUAGE パラメーター 25

MIMETYPES セクション 32

NOHTML セクション 39

REPORTSERVERTIMEOUT パラメーター 27

SECURITY セクション 27

HTTP サーバー・オプション (続き)

SEPARATOR パラメーター 39

SERVERACCESS パラメーター 28

ShowSearchString パラメーター 26

TEMPDIR パラメーター 26

TEMPLATEDIR パラメーター 27

USEEXECUTABLE パラメーター 30, 32

HTTP サーバー・ソフトウェア

インストール 15

httpd.conf ファイル 173

I

IBM i

インストール 16

IBM Thread and Monitor Dump Analyzer 187

Image Web Viewer

インストール 57

構成 66

説明 1

要件 58

IMAGEDIR パラメーター 24

INSTALLDIR パラメーター 30, 31

J

Java API

解説 111

説明 1

ソフトウェア要件 1

プログラミング・ガイド 113

Java アプレット

インストール 57

使用可能 31

説明 1, 6, 7

ディレクトリー 31

要件 58

ラージ・オブジェクトのサポート 29, 170

APPLETCACHEDIR パラメーター 20

Java 行データ・ビューアー

構成 67

ODApplet.jre.path.IE パラメーター 45

ODApplet.jre.path.NN パラメーター 45

ODApplet.jre.version パラメーター 45

ODApplet.version パラメーター 46

Java サブレット

解説 109

Java 診断コマンド 188

Diagnostic tool for Java garbage collector 189

HAT: Heap Analysis Tool 189

HeapAnalyzer 190

HeapRoots 191

HPROF: ヒープ・プロファイラー 189

jmap

-heap 188

Java 診断コマンド (続き)

jmap (続き)

-histo 188

-permstat 188

jstat 189

Java ダンプ 187

JFIF 文書

媒体タイプ 35

MIME コンテンツ・タイプ 35

L

LANGUAGE パラメーター 25, 181

Line Data Java アプレット

インストール 57

説明 1, 6

要件 58

APPLETCACHEDIR パラメーター 20

LINEVIEWING パラメーター 44

LOG パラメーター 186

Logoff

関数の説明 8

関数呼び出しのサンプル 88

パラメーター 87

API 87

Logon

関数の説明 8

関数呼び出しのサンプル 91

区切り文字付き ASCII 出力 176

パラメーター 89

API 89

M

MAXHITS パラメーター 44

MIME コンテンツ・タイプ 32, 126

MIMETYPES セクション 32

N

NLS 23, 25, 181

NOHTML セクション 39

NOLINKS パラメーター 45

O

ODApplet.jre.path.IE パラメーター 45

ODApplet.jre.path.NN パラメーター 45

ODApplet.jre.version パラメーター 45

ODApplet.version パラメーター 46

ODCallback 152

ODConfig 123

ODCriteria

オペランド 126, 135, 138

検索値 126, 135, 138

ODCriteria (続き)

名前 126

文書、更新 161

文書の更新 161

ODCriteria.getAscending 146

ODCriteria.getDBFieldMask 138

ODCriteria.getDBFieldNames 138

ODCriteria.getDefaultFmt 138

ODCriteria.getDisplayFmt 138

ODCriteria.getDisplayFmtQual 138

ODCriteria.getMaxDisplayChars 138

ODCriteria.getMaxEntryChars 138

ODCriteria.getMaxSearchValue 138

ODCriteria.getMinSearchValue 138

ODCriteria.getName 126, 138, 146

ODCriteria.getOperator 126, 135, 138

ODCriteria.getSearchValues 138

ODCriteria.getValidOperators 138

ODCriteria.isDefaultValueAvailable 138

ODCriteria.isDefaultValueFixed 138

ODCriteria.isRequired 138

ODCriteria.isUpdateable 138

ODCriteria.setAscending 146

ODCriteria.setFixedValues 138

ODCriteria.setOperator 161

ODCriteria.setSearchValue 126, 161

ODCriteria.setSearchValues 126, 135, 161

ODCriteria.setSortOrder 146

ODCriteria.Type 138

ODFolder

アプリケーション・グループ 124

記述 126

基準 126, 135, 138

クローズ 124, 126, 131, 135

検索 126, 131, 135, 138, 146, 149

取り消し、検索の 135

名前 126, 146

表示順序 126, 146

文書、印刷 153

文書、検索 149

文書の印刷 153

メッセージ 126

Retrieve Document 149

ODFolder.close 124, 126, 131, 135, 138, 149

ODFolder.getAppGroupNames 124, 138

ODFolder.getCriteria 126, 135, 138

ODFolder.getDescription 126, 131, 138

ODFolder.getDisplayOrder 126, 131, 146

ODFolder.getName 126, 131, 135, 138, 146

ODFolder.getNumAppGroups 124, 138

ODFolder.getNumCriteria 138

ODFolder.getSearchMessage 126

ODFolder.getSortLocation 146

ODFolder.printDocuments 153

ODFolder.retrieve 149

ODFolder.search 126, 131, 135, 146, 149

ODFolder.setApplGroupForSearchWithSQL 131
 ODFolder.setMaxHits 126, 131
 ODFolder.setSortLocation 146
 ODHHit
 注 155, 157
 表示値 126, 131, 146
 文書、検索 149
 文書、更新 161
 文書タイプ 126
 文書の更新 161
 文書の場所 126
 文書リスト 146
 Annotations 155, 157
 MIME コンテンツ・タイプ 126
 Retrieve Document 149
 ODHHit.addNote 157
 ODHHit.getDisplayValue 126, 131, 146, 161
 ODHHit.getDisplayValues 126, 131
 ODHHit.getDocId 126, 131, 149, 161
 ODHHit.getDocLocation 126, 131
 ODHHit.getDocType 126, 131
 ODHHit.getMimeType 126, 131
 ODHHit.getNotes 155, 157
 ODHHit.getNoteStatus 155
 ODHHit.retrieve 149
 ODHHit.updateValuesForHit 161
 ODNNote
 位置 155
 色 155
 グループ名 155
 時刻 155
 テキスト 155
 日付 155
 ページ 155
 ユーザー ID 155
 Annotations 155, 157
 ODNNote.getColor 155
 ODNNote.getDateTime 155
 ODNNote.getGroupName 155, 157
 ODNNote.getOffsetX 155
 ODNNote.getOffsetY 155
 ODNNote.getPageNum 155
 ODNNote.getText 155, 157
 ODNNote.getUserid 155
 ODNNote.isOkToCopy 155, 157
 ODNNote.isPublic 155, 157
 ODNNote.setGroupName 157
 ODNNote.setOkToCopy 157
 ODNNote.setPublic 157
 ODNNote.setText 157
 ODServer
 アプリケーション名 121
 サーバー 120, 121
 サーバーへの接続 120
 サーバー・プリンター 153
 接続先 121
 ODServer (続き)
 接続タイプ 121
 切断 121
 設定、パスワードの 164
 取り消し、検索の 135
 パスワード 120, 121, 164
 パスワードの設定と入手 120
 フォルダー、オープン 149
 フォルダー記述 142
 フォルダーのオープン 138, 149
 フォルダー名 142
 プリンター 153
 文書、検索 149
 変更、パスワードの 164
 ポート 121
 ユーザー ID 120, 121
 ユーザー ID の設定と入手 120
 ローカル・ディレクトリー 121
 Retrieve Document 149
 ODServer.cancel 135
 ODServer.changePassword 164
 ODServer.getConnectType 121
 ODServer.getFolderNames 142
 ODServer.getFoldersDescription 142
 ODServer.getNumFolders 142
 ODServer.getPassword 120, 121, 164
 ODServer.getPort 121
 ODServer.getServerName 120, 121
 ODServer.getServerPrinters 153
 ODServer.getUserId 120, 121
 ODServer.initialize 121
 ODServer.logoff 120, 121
 ODServer.logon 120, 121
 ODServer.openFolder 138, 149
 ODServer.ServerName 164
 ODServer.setConnectType 121
 ODServer.setPassword 120, 121, 164
 ODServer.setPort 121
 ODServer.setServerName 120, 121
 ODServer.setUserId 120, 121, 164
 ODServer.terminate 120, 121
 ODWEK ソフトウェア
 インストール 15
 OnDemand Internet Connection
 説明 1
 OnDemand サーバー・オプション
 デフォルト 17
 パラメーター 18
 HOST パラメーター 19
 PORT パラメーター 18, 19
 PROTOCOL パラメーター 18, 19
 @SRV@_DEFAULT セクション 17
 @SRV@_server セクション 18

P

PCX 文書
 媒体タイプ 36
 MIME コンテンツ・タイプ 36
PDF 文書
 媒体タイプ 36
 MIME コンテンツ・タイプ 36
PNG 文書
 媒体タイプ 36
 MIME コンテンツ・タイプ 36
PORT パラメーター 18, 19
POST メソッド 10
Print Document
 関数の説明 9
 関数呼び出しのサンプル 96
 パラメーター 92
 API 92
 Java API 153
PROTOCOL パラメーター 18, 19

R

REPORTSERVERTIMEOUT パラメーター 27
Retrieve Document
 関数の説明 9
 関数呼び出しのサンプル 101
 パラメーター 97
 API 97

S

Search Criteria
 関数の説明 9
 関数呼び出しのサンプル 104
 区切り文字付き ASCII 出力 176
 パラメーター 102
 API 102
 Java API 126, 131, 138
 SQL ストリング 131
SECURITY セクション 27
SEPARATOR パラメーター 39
SERVERACCESS パラメーター 28
SERVERPRINT パラメーター 46
SERVERPRINTERS パラメーター 46
SHOWDOCLOCATION パラメーター 46
ShowSearchString パラメーター 26
SQL 検索ストリング、Java API を使用した 131

T

TCP/IP 通信プロトコル 18, 19
TEMPDIR パラメーター 26
TEMPLATEDIR パラメーター 27

TIFF 文書
 媒体タイプ 37
 MIME コンテンツ・タイプ 37
TRACE パラメーター 48
TRACEDIR パラメーター 49
TrueType フォント
 AFP フォントのマッピング 64
TXT 添付ファイル 38

U

Update Document
 関数の説明 9
 関数呼び出しのサンプル 106
 パラメーター 105
 API 105
 Java API 161
UPDATETIMESTAMP セクション 28
USEEXECUTABLE パラメーター 30, 32

V

View Annotations
 関数の説明 9
 関数呼び出しのサンプル 108
 区切り文字付き ASCII 出力 178
 パラメーター 107
 API 107
VIEWNOTES パラメーター 47

W

Web アプリケーション
 サンプル 53
Web サーバー・オプション
 言語 181
 コード・ページ 181
 DBCS 181
 NLS 181
Web ページ
 サンプル 53

[特殊文字]

@SRV@_DEFAULT セクション 17
@SRV@_server セクション 18



プログラム番号: 5770-RD1

Printed in U.S.A.

SC88-8273-00



日本アイ・ビー・エム株式会社

〒103-8510 東京都中央区日本橋箱崎町19-21