

ldc コマンド・リファレンス・ガイド
10g リリース 3 (10.1.3.3.0)
部品番号 : B51311-01

2008 年 11 月

Idc コマンド・リファレンス・ガイド, 10g リリース 3 (10.1.3.3.0)

部品番号: B51311-01

原本名: Idc Command Reference Guide, 10g Release 3 (10.1.3.3.0)

原本協力者: Will Harris, Jean Wilson, Eva Cordes, Rick Petty, Sam White

Copyright © 2007, Oracle. All rights reserved.

制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空、大量輸送、医療あるいはその他の本質的に危険を伴うアプリケーションで使用されることを意図しておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性（**redundancy**）、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle、JD Edwards、PeopleSoft、Siebel は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。その他の名称は、他社の商標の可能性があり得ます。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

目次

第 1 章：概要

このガイドについて	1-2
-----------------	-----

第 2 章：IdcCommand: Java コマンド・ユーティリティ

IdcCommand の設定および実行	2-2
コマンド・ファイル	2-2
コマンド・ファイルの構文	2-3
優先順位	2-4
特殊タグと特殊文字	2-4
構成オプション	2-5
IdcCommand の実行	2-7
ランチャの使用	2-7
引用符付け	2-9
計算される設定	2-9
ランチャの環境変数	2-12
ユーザー・インタフェース	2-13
ランチャの構成	2-14
構成ファイルの例	2-14
リモートでのサービスのコール	2-17

第 3 章：IdcCommandX と IdcCommandUX: ActiveX コマンド・ユーティリティ

IdcCommandUX の概要	3-2
設定	3-2
コール手順	3-2
Visual Basic	3-2
Visual C++	3-3
サービスの実行	3-3
Active Server Page からの IdcCommandUX のコール	3-4

HDA の例	3-4
COM オブジェクトの作成	3-5
接続の初期化	3-5
サービスおよびパラメータの定義	3-5
カスタム・リソースの参照	3-6
サービスの実行	3-6
結果の取得	3-6
SOAP の例	3-7
リソース・インクルードによるコンテンツの形式設定	3-9
リモート・マシンからのコンテンツ・サーバーへの接続	3-11
ASP ページのコーディング	3-11
IdcCommandUX のメソッド	3-16
addExtraHeadersForCommand	3-16
closeServerConnection	3-17
computeNativeFilePath	3-18
computeURL	3-19
computeWebFilePath	3-22
connectToServer	3-23
executeCommand	3-24
executeFileCommand	3-25
forwardRequest	3-26
getLastErrorMessage	3-26
init (非推奨)	3-27
initRemote	3-28

第 4 章：IdcClient OCX コンポーネント

IdcClient OCX の説明	4-2
概要	4-2
イベント、メソッドおよびプロパティ	4-3
OCX イベント	4-3
OCX メソッド	4-4
OCX プロパティ	4-4
IdcClient OCX インタフェース	4-4
IdcClient OCX コントロールの設定	4-5
コンポーネントの設定	4-5
ビジュアル・インタフェースの作成	4-6
IdcClient のイベント	4-17
IntradocBeforeDownload	4-17
IntradocBrowserPost	4-17
IntradocBrowserStateChange	4-18
IntradocRequestProgress	4-18

IntradocServerResponse	4-18
IdcClient のメソッド	4-19
AboutBox	4-20
Back	4-20
CancelRequest	4-21
DoCheckoutLatestRev	4-21
DownloadFile	4-22
DownloadNativeFile	4-23
Drag	4-24
EditDocInfoLatestRev	4-24
Forward	4-25
GoCheckinPage	4-25
Home	4-26
InitiateFileDownload	4-27
InitiatePostCommand	4-27
Move	4-28
Navigate	4-29
NavigateCgiPage	4-29
RefreshBrowser	4-30
SendCommand	4-30
SendPostCommand	4-31
SetFocus	4-31
ShowDMS	4-32
ShowDocInfoLatestRev	4-32
ShowWhatsThis	4-33
StartSearch	4-33
Stop	4-33
UndoCheckout	4-34
ViewDocInfo	4-34
ViewDocInfoLatestRev	4-35
ZOrder	4-35
IdcClient のプロパティ	4-36
ClientControlledContextValue	4-36
HostCgiUrl	4-37
Password	4-37
UseBrowserLoginPrompt	4-37
UseProgressDialog	4-37
UserName	4-38
WorkingDir	4-38

付録 A: サード・パーティ・ライセンス

Apache Software License	A-2
W3C® Software Notice and License	A-2
Zlib License	A-4
General BSD License	A-5
General MIT License	A-5
Unicode License	A-6
その他の帰属	A-7

索引

1

概要

はじめに

このガイドの情報は、Content Server 10gR3に基づいています。この情報は、製品テクノロジーの発達やハードウェアおよびオペレーティング・システムの作成、変更に伴い変更される場合があります。

ブラウザ、Web サーバーおよびオペレーティング・システムの技術的な性質により、サード・パーティ製品のすべてのバージョンおよび機能との互換性を保証することはできません。

この項の内容は、次のとおりです。

❖ [このガイドについて](#) (1-2 ページ)

このガイドについて

このガイドでは、IdcCommand および IdcCommandX ユーティリティを使用して他のアプリケーションからコンテンツ・サーバー・サービスにアクセスする方法について説明します。これらのユーティリティは、サポート・サイトで利用できます。

- ❖ IdcCommand Java コマンド・ユーティリティは、ユーザーによるコンテンツ・サーバー・サービスの実行を可能にするスタンドアロンの Java アプリケーションです。
[第2章「IdcCommand: Java コマンド・ユーティリティ」](#)を参照してください。
- ❖ IdcCommandX および IdcCommandUX は、プログラムによるサービスの実行やファイル・パス情報の取得を可能にする ActiveX コントロールです。IdcCommandX は、コンテンツ・サーバーが使用する標準 IdcCommand サービスの COM ラッパーとして機能します。IdcCommandUX は、IdcCommandX コントロールを更新したもので、マルチバイト言語で機能します。[第3章「IdcCommandX と IdcCommandUX: ActiveX コマンド・ユーティリティ」](#)を参照してください。
- ❖ リモート・コンテンツ・サーバーへの接続および Content Server サービスの実行のために、Object Linking and Embedding Control Extension (OCX) コントロールも提供されます。IdcClient OCX コントロールは、Content Server 内のコンテンツおよびコンテンツ管理機能にアクセスするために Windows Visual Basic 開発環境内で使用します。[第4章「IdcClient OCX コンポーネント」](#)を参照してください。



注意: このガイドの情報は、Content Server 10gR3 に基づいています。この情報は、製品テクノロジーの発達やハードウェアおよびオペレーティング・システムの作成、変更に伴い変更される場合があります。ブラウザ、データベース、Web サーバーおよびオペレーティング・システムの技術的な性質により、サード・パーティ製品のすべてのバージョンおよび機能との互換性を保証することはできません。



注意: このリファレンス・ガイドは、ソフトウェア開発キット (SDK) に含まれます。詳細は、『Getting Started with the Software Developer's Kit (SDK)』を参照してください。



注意: カスタム・コンポーネントでのサービスの使用方法は、『Services Reference Guide』および『Working with Content Server Components』を参照してください。

対象読者





このガイドは、Content Server の機能にアクセスする必要があるアプリケーション開発者を対象としています。このガイドでは、コンテンツ・サーバーの Java コマンド・ユーティリティ、ActiveX コマンド・ユーティリティおよび OCX コンポーネントについて説明します。

また、レイアウト・マネージャ機能を使用して別のインタフェース・ナビゲーションおよび設計を提供するコンテンツ・サーバー管理者も対象としています。

表記規則

このガイドでは次の表記規則を使用します。

- ❖ `<install_dir>/` という表記は、Content Server 製品がインストールされているシステム上の場所を指すために使用されます。
- ❖ スラッシュ (/) は、パス名のディレクトリ・レベルを区切るために使用されます。ディレクトリ名の末尾には常にスラッシュが付きます。
- ❖ 注意、技術ヒント、重要な通知および警告には、次の表記規則が使用されます。

記号	説明
	これは注意です。特に注意が必要な情報を示す場合に使用されます。
	これは技術ヒントです。作業を簡単にするために使用できる情報を示す場合に使用されます。
	これは重要な通知です。必要な手順または情報を示す場合に使用されます。
	これは警告です。データの損失や重大なシステムの問題の原因となる可能性がある情報を示す場合に使用されます。

2

IdcCOMMAND: JAVA コマンド・ユーティリティ

はじめに

IdcCommand ユーティリティは、Content Server サービスを実行するスタンドアロンの Java アプリケーションです。Content Server のブラウザ・インタフェースまたは管理アプレットから実行できるアクションのほとんどは、IdcCommand から実行できます。

プログラムは、サービス・コマンドおよびパラメータを含む[コマンド・ファイル](#) (2-2 ページ) を読み取った後、指定されたサービスをコールします。ログ・ファイルには、コールの実行時間、サービスの実行の成否、実行エラーの有無を記録できます。

この項の内容は、次のとおりです。

- ❖ [IdcCommand の設定および実行](#) (2-2 ページ)
- ❖ [コマンド・ファイル](#) (2-2 ページ)
- ❖ [構成オプション](#) (2-5 ページ)
- ❖ [IdcCommand の実行](#) (2-7 ページ)
- ❖ [ランチャの使用](#) (2-7 ページ)
- ❖ [リモートでのサービスのコール](#) (2-17 ページ)



注意: IdcCommand ユーティリティは、コマンドの成否に関する情報のみを返します。対話型セッションでコンテンツ・サーバーから情報を取得するには、Windows プラットフォームで使用可能な IdcCommandX Java COM ラッパーを使用します。詳細は、[第3章「IdcCommandX と IdcCommandUX: ActiveX コマンド・ユーティリティ」](#)を参照してください。

IdcCOMMAND の設定および実行

IdcCommand を設定するには、次の 2 つの情報を指定する必要があります。

- ❖ [コマンド・ファイル](#) (2-2 ページ) : 実行するサービスおよびサービス・パラメータを指定します。
- ❖ [構成オプション](#) (2-5 ページ) : コマンド・ファイルおよびその他の IdcCommand 情報を指定します。IdcCommand 構成オプションは次の 2 か所に設定できます。
 - 構成ファイルで、次のような名前 / 値ペアを使用して設定します。

```
IdcCommandFile=newfile.hda
IdcCommandUserName=sysadmin
IdcCommandLog=C:/stellent/newlog.txt
ConnectionMode=server
```
 - IdcCommand の実行時にコマンドラインで、次のようなオプション・フラグを指定して設定します。

```
-f newfile.hda -u admin -l C:/stellent/newlog.txt -c server
```



注意: コマンドライン構成オプションは、構成ファイルの設定に優先します。

IdcCommand の実行

IdcCommand は、コマンドラインから実行されます。[構成オプション](#) (2-5 ページ) は、コマンドラインから、または構成ファイルに指定できます。詳細は、2-7 ページの「[IdcCommand の実行](#)」を参照してください。

コマンド・ファイル

コマンド・ファイルは、IdcCommand ユーティリティによって実行されるサービス・コマンドおよびパラメータを定義します。コマンド・ファイルでは、次のルールに従います。

- ❖ [コマンド・ファイルの構文](#) (2-3 ページ)
- ❖ [優先順位](#) (2-4 ページ)
- ❖ [特殊タグと特殊文字](#) (2-4 ページ)

コマンド・ファイルの構文

コマンド・ファイルでは、HDA（ハイパーデータ・ファイル）構文を使用してサービス・コマンドを定義します。

- ❖ 実行される各サービスは、パラメータとともに @Properties LocalData セクションに指定します。
- ❖ 一部のサービスについては、@ResultSet セクションを使用して追加情報を指定します。
- ❖ コマンド・ファイルの 1 つのセクションからのデータは、次のセクションにまたがることはできません。セクションごとに、コマンドのデータ・セットは完結する必要があります。
- ❖ サービス名およびパラメータは、大 / 小文字を区別します。

たとえば、次のコマンド・ファイルは、ADD_USER サービスを実行し、2 人の新規ユーザーの属性を定義します。

```
<?hda version="5.1.1 (build011203)" jcharset=Cp1252 encoding=iso-8859-1?>
```

```
# Add users
@Properties LocalData
IdcService=ADD_USER
dName=jsmith
dUserAuthType=Local
dFullName=Jennifer Smith
dPassword=password
dEmail=email@email.com
@end
@ResultSet UserAttribInfo
2
dUserName
AttributeInfo
jsmith
role,contributor,15
@end
<<EOD>>
@Properties LocalData
IdcService=ADD_USER
dName=pwallek
dUserAuthType=Local
dFullName=Peter Wallek
dPassword=password
dEmail=email@email.com
@end
@ResultSet UserAttribInfo
2
dUserName
AttributeInfo
pwallek
role,contributor,15,account,marketing,7
@end
<<EOD>>
```

優先順位

IdcCommand では、優先順位を使用してコマンド・ファイルの LocalData セクション内に指定された名前 / 値ペア間の競合を解決します。標準の名前 / 値ペアが解析される場合、@Properties LocalData タグ内にあると判断されます。セクションに HDA タグが含まれる場合、標準の名前 / 値ペアは、@Properties LocalData タグ内の名前 / 値ペアに優先します。

たとえば、標準の名前 / 値ペアに foo=x、@Properties LocalData タグ内に foo=y と指定されている場合、名前 / 値ペアの foo=x はタグの外にあるため、優先されます。

特殊タグと特殊文字

次の特殊タグと特殊文字は、コマンド・ファイルで使用できます。

特殊文字	説明
IdcService= <i>service_name</i>	コマンド・ファイルの各セクションには、コールするサービス名を指定する必要があります。
<<EOD>>	データ終了マーカー。コマンド・ファイルには、1 つ以上のセクションをデータ終了マーカーで区切って指定できます。例は、2-3 ページの「 コマンド・ファイルの構文 」を参照してください。
#	行の先頭にある番号記号は、その行がコメントであることを示します。
\	バックスラッシュは、エスケープ文字です。
@Include <i>filename</i>	このタグを使用すると、@Include タグがある場所に別のファイルの内容をインクルードできます。このタグは、HDA ファイル全体あるいは共有の名前 / 値ペアのインクルードに使用できます。このインクルードは、指定されたファイルの内容をそのまま取得して @Include タグの場所に配置します。ファイルでは、必要な回数だけインクルードでき、インクルードされるファイルに他のファイルをインクルードすることも可能です。ただし、循環インクルードはできません。

構成オプション

IdcCommand ユーティリティを実行するには、コマンドラインまたは intradoc.cfg 構成ファイルに次の情報を指定します。

パラメータ	必須	コマンドラインの構文	構成ファイルの構文
コマンド・ファイル (2-5 ページ)	はい	-f name.txt	IdcCommandFile=name.txt
ユーザー (2-5 ページ)	はい	-u sysadmin	IdcCommandUserName=sysadmin
ログ・ファイル (2-6 ページ)	いいえ	-l C:/logs/log.txt	IdcCommandLog=C:/logs/log.txt
接続モード (2-6 ページ)	いいえ	-c auto	ConnectionMode=auto



注意: コマンドライン構成オプションは、構成ファイルの設定に優先します。

コマンド・ファイル

サービス・コマンドおよびパラメータを含むコマンド・ファイルの名前を指定する必要があります。コマンド・ファイル・パラメータには、フルパス (C:/command_files/command.txt など) または相対パスを指定できます。詳細は、2-2 ページの「[コマンド・ファイル](#)」を参照してください。

ユーザー

コンテンツ・サーバーのユーザー名を指定する必要があります。このユーザーには、コールされるサービスを実行するための権限が必要です。

ログ・ファイル

IdcCommand ログ・ファイルのパスおよびファイル名を指定できます。各コマンドが実行されると、メッセージがログ・ファイルに送信されます。ログ・ファイルには、コマンドの実行時間および成功または失敗のステータスが記録されます。ログ・ファイルがすでに存在する場合は、新しいメッセージで上書きされます。ログ・ファイルを使用して、処理情報をユーザーに表示することができます。

- ❖ 実行されたアクションが成功した場合は、「success」メッセージがログ・ファイルに書き込まれます。
- ❖ 実行されたアクションが失敗した場合は、エラー・メッセージがログ・ファイルに書き込まれます。
- ❖ ログ・ファイルが指定されていない場合、情報は画面にのみ表示されます。

接続モード

IdcCommand サービスを実行するための接続モードを指定できます。

接続モード	説明
auto	コンテンツ・サーバーへの接続を試行します。この試行が失敗すると、サービスはスタンドアロン・モードで実行されます。 これは、デフォルトの接続モードです。
server	コンテンツ・サーバーによってのみ、サービスを実行します。
standalone	スタンドアロン・セッションでサービスを実行します。 スタンドアロン・モードで実行できない特定のサービスがあります。一般に、これらのサービスは、サーバーによってバックグラウンド・スレッドで非同期に実行されます。たとえば、検索索引の更新または再構築の際に発生します。

IdcCOMMAND の実行

IdcCommand を実行するには次のようにします。

1. 新しい IdcCommand 作業ディレクトリを作成します。
このディレクトリは、コマンド・ファイルおよび構成ファイルに使用します。
2. 作業ディレクトリに [コマンド・ファイル](#) (2-2 ページ) を作成して、必要なサービス・コマンドを指定します。
3. intradoc.cfg 構成ファイルを <install_dir>/bin/ から作業ディレクトリにコピーします。



重要: IntradocDir または WebBrowserPath の情報を削除しないでください。

4. IdcCommand オプションを作業ディレクトリ内の intradoc.cfg ファイルに追加します。詳細は、2-5 ページの「[構成オプション](#)」を参照してください。

```
IdcCommandFile=newfile.hda
IdcCommandUserName=sysadmin
IdcCommandLog=C:/stellent/newlog.txt
```

5. <install_dir>/bin ディレクトリに格納されている IdcCommand を実行します。

```
IdcCommand.exe
```

ランチャの使用

ランチャは、Windows 環境でのサービスの管理や Java VM のコマンドライン引数および環境設定の構成に使用されるネイティブの C++ アプリケーションです。

ランチャの動作は、主に、構成ファイルを検出して読み取り、特殊な値を計算した後、構成したコマンドラインで実行可能ファイルを起動することです。構成ファイルでは、Bourne Shell のような置換を使用できます。置換はすべてドル記号 (\$) で始まり、英数字の識別子または中カッコ ({}) で囲まれた式が続きます。

ランチャ実行可能ファイルは、<install_dir>/shared/os/platform/bin/Launcher にインストールされます。UNIX システムでは、Launcher.sh への symlink が bin/ ディレクトリに作成されます。Launcher.sh は、ランチャ実行可能ファイルを実行する Bourne Shell ラッパーです。このラッパーの目的は、プラットフォームに適したバイナリのランチャ実行可能ファイルを見つけることです。ランチャという用語は、ここではネイティブのランチャ実行可能ファイルまたは Bourne Shell スクリプトの Launcher.sh を指すために使用されます。

ランチャまたは **Launcher.sh** への **symlink** は、有効な **intradoc.cfg** 構成ファイルとともにディレクトリ内に存在し、起動される Java クラス・ファイルと同じ名前（大 / 小文字を区別）である必要があります。ランチャはこの名前を使用して環境変数 **STARTUP_CLASS** を設定します。

Windows では、**GetModuleFileName()** をコールしてこの名前を計算します。**UNIX** システムでは、**argv[0]** を調べて計算します。**PLATFORM** 変数は、プラットフォームの **Content Server** 識別子に設定されます。**BIN_DIR** 変数は、ランチャがあるディレクトリに設定されます。

ランチャは、**BIN_DIR** から **intradoc.cfg** というファイルを読み取ります。このファイルには、**IntradocDir** の値が含まれます。**IntradocDir** は、相対パスを解決するためのベース・ディレクトリとして使用されます。このガイドでの非修飾パスはすべて、**IntradocDir** に対して相対的であるとみなされます。**Content Server** の将来のリリースでは、これらの変数名は変更または削除される可能性があります。

intradoc.cfg ファイルに **SharedDir** の値が含まれていない場合、ランチャが **SharedDir** を **\$IntradocDir/shared** に設定します。ランチャが **Windows** サービスを起動する場合は、**IS_SERVICE** を 1 に設定します。また、設定されていない場合は、**PATH_SEPARATOR** をプラットフォームに適した文字に設定します。

次に、ランチャは、次の順序で使用可能な構成ファイルをすべて読み取ります。

1. **\$SharedDir/config/resources/launcher.cfg**
2. **\$SharedDir/config/resources/launcher-local.cfg**
3. **\$BIN_DIR/./config/config.cfg**
4. **\$IntradocDir/config/config.cfg**
5. **\$IntradocDir/config/config-\$PLATFORM.cfg**
6. **\$IntradocDir/config/state.cfg**
7. **\$SharedDir/os/\$PLATFORM/launcher.cfg**
8. **\$SharedDir/os/\$PLATFORM/launcher-local.cfg**
9. **\$BIN_DIR/intradoc.cfg**
10. **\$BIN_DIR/intradoc-\$PLATFORM.cfg**
11. **-cfg** オプションを使用して、コマンドラインに指定されたすべてのファイル

引用符付け

ランチャは、Bourne Shell のような引用符付けルールを使用します。文字列を二重引用符 (") で囲むと、空白をエスケープできます。バックスラッシュ (\) を任意の文字の前に置くと、その文字を指定できます。最終的なコマンドラインが計算されると、ランチャは、そのコマンドラインを引用符なしの空白ごとに区切ります。その結果、各文字列は引用符なしとなり、コマンドの argv 配列のエントリとして使用されます。

計算される設定

構成ファイルを読み取った後、ランチャは変数置換を処理します。一部の変数は、ディレクトリまたはファイルの検証、コマンドライン引数リストの作成あるいは PATH のような変数の構成のために追加の計算を行うことができます。

これらの特殊な計算は、変数に対してそのタイプに基づいて実行されます。変数のタイプを設定するには、前述の構成ファイルのいずれかに `TYPE_variable_name=typename` と設定します。

ランチャの変数タイプを次に示します。

❖ file

- 例:

```
TYPE_PASSWD_FILE=file
PASSWD_FILE_sys5=/etc/passwd
PASSWD_FILE_bsd=/etc/master.passwd
```

このタイプは、ファイルを検索します。*variable_name* の値が既存ファイルへのパスの場合は、その値が保存されます。既存ファイルへのパスではない場合は、*variable_name_* で始まるすべての変数がチェックされます。既存ファイルへのパスである最後の値が *variable_name* の新しい値に使用されます。

この例では、PASSWD_FILE は、/etc/master.passwd が存在する場合には /etc/master に設定されます。そうではなく、/etc/passwd が存在する場合には /etc/passwd に設定されます。それ以外の場合は、PASSWD_FILE は定義されません。

❖ directory

- 例:

```
TYPE_JDK=directory
JDK_java_home=$JAVA_HOME
OS_DIR=$SharedDir/os
DEFAULT_JDK_DIR=$OS_DIR/$PLATFORM
JDK_legacy142=$DEFAULT_JDK_DIR/j2sdk1.4.2_04
JDK_default=$DEFAULT_JDK_DIR/jdk1.5.0_07
```

この例では、JDK は、ディレクトリである最後の JDK_ 変数と同じ値が設定されます。通常、これは、Content Server とともにインストールされた JDK を指します。

JDK_java_home が \$JAVA_HOME を参照していることに注意してください。変数がどの構成ファイルにも定義されておらず、環境に定義されている場合は、環境の値が使用されます。

❖ executable

- 例:

```
TYPE_JAVA_EXE=executable
JAVA_EXE_default=java$EXE_SUFFIX
JAVA_EXE_jdk_default=$JDK/bin/java$EXE_SUFFIX
```

executable タイプは、実行可能ファイルを検索します。このタイプの機能は **file** タイプと非常によく似ていますが、各候補値について \$PATH 内のすべてのディレクトリを調べます。この例では、JAVA_EXE は、JDK 内の **Java** 実行可能ファイルが存在する場合は、そのファイルに設定されます。それ以外の場合は、PATH 内の最初の **Java** 実行可能ファイルに設定されます。

❖ list

- 例:

```
TYPE_JAVA_OPTIONS=list
JAVA_MAX_HEAP_SIZE=384
DEFINE_PREFIX=-D
JAVA_OPTIONS_BIN_DIR=${DEFINE_PREFIX}idc.bin.dir=$BIN_DIR
JAVA_OPTIONS_maxheap=${JAVA_MAX_HEAP_SIZE}-Xmx${JAVA_MAX_HEAP_SIZE}\m}
JAVA_OPTIONS_service=${IS_SERVICE+$JAVA_SERVICE_EXTRA_OPTIONS}
```

list タイプは、実行可能ファイルのオプション・リストを計算します。

variable_name_ で始まる各値が引用符付きのオプションになり、*variable_name* にはリスト全体が設定されます。この例では、JAVA_OPTIONS には次の文字列が設定されます。

```
"-Didc.bin.dir=/intradocdir/bin/" "-Xmx384m".
```

❖ path

- 例:

```
TYPE_JAVA_CLASSPATH=path
JAVA_CLASSPATH_legacy=$CLASSPATH
JAVA_CLASSPATH_orig=$IntradocDir/classes
JAVA_CLASSPATH_unpackaged=$SharedDir/classes
JAVA_CLASSPATH_components=$COMPONENTS_CLASSPATH
JAVA_CLASSPATH_server=$SharedDir/classes/server.zip
JAVA_CLASSPATH_refinery=$SharedDir/classes/idcrefinery.zip
JAVA_CLASSPATH_flexion=$SharedDir/classes/flexionxml.jar
JAVA_CLASSPATH_jspserver=$SharedDir/classes/jspserver.jar
JAVA_CLASSPATH_ldap=$SharedDir/classes/ldapjdk.jar
```

classpath タイプは、パスのような値を計算します。*variable_name_* で始まる各変数の値が、PATH_SEPARATOR の値で区切られて *variable_name* の値に追加されます。この例では、JAVA_CLASSPATH は非常に長いクラスパスに設定されます。

❖ lookupstring

• 例:

```

TYPE_VDK_PLATFORM=lookupstring
PARAMETER_VDK_PLATFORM=${PLATFORM}_${UseVdkLegacySearch+vdk27}
VDK_PLATFORM_aix_vdk27=_rs6k41
VDK_PLATFORM_aix=_rs6k43
VDK_PLATFORM_hpux_vdk27=_hpux11
VDK_PLATFORM_hpux=_hpux11
VDK_PLATFORM_freebsd_vdk27=_ilnx21
VDK_PLATFORM_freebsd=_ilnx21
VDK_PLATFORM_linux_vdk27=_ilnx21
VDK_PLATFORM_linux=_ilnx21
VDK_PLATFORM_solaris_vdk27=_ssol26
VDK_PLATFORM_solaris=_ssol26
VDK_PLATFORM_win32_vdk27=_nti40
VDK_PLATFORM_win32=_nti40

```

lookupstring は、2 番目のパラメータを使用して最終的な値の検索キーを構成します。2 番目のパラメータとは、`$PARAMETER_variable_name` の値です。この値が定義されていない場合は、`variable_name` の現在の値が検索キーとして使用されます。この例では、`PARAMETER_VDK_PLATFORM` は、`UseVdkLegacySearch` の値に応じて、`${PLATFORM}_` または `${PLATFORM}_vdk27` の値となります。

この値は、その後、変数 `VDK_PLATFORM_${PARAMETER_VDK_PLATFORM}` の値の検索に使用されます。検索された値は、引用符を付けられ `VDK_PLATFORM` に割り当てられます。

❖ lookuplist

• 例:

```

TYPE_STARTUP_CLASS=lookuplist
STARTUP_CLASS_version=Installer --version
STARTUP_CLASS_installer=Installer
STARTUP_CLASS_WebLayoutEditor=IntradocApp WebLayout
STARTUP_CLASS_UserAdmin=IntradocApp UserAdmin
STARTUP_CLASS_RepositoryManager=IntradocApp RepositoryManager
STARTUP_CLASS_Archiver=IntradocApp Archiver
STARTUP_CLASS_WorkflowAdmin=IntradocApp Workflow
STARTUP_CLASS_ConfigurationManager=IntradocApp ConfigMan

```

lookuplist は、2 番目のパラメータを使用して最終的な値の検索キーを構成します。2 番目のパラメータとは、`$PARAMETER_variable_name` の値です。この値が定義されていない場合は、`variable_name` の現在の値が検索キーとして使用されます。

lookupstring とは異なり、lookuplist は最終的な値に引用符を付けません。この例では、`STARTUP_CLASS` の現在の値を *version* と仮定します。`STARTUP_CLASS` は、値 `Installer --version` に置き換えられます。

ランチャの環境変数

計算される設定を処理した後に、ランチャは文字列 `EXPORT_` で始まるすべての変数に対して繰り返されます。各変数の値は環境変数の名前として使用されます。つまり、環境変数の名前には `EXPORT_` 変数の後半の値が割り当てられます。たとえば、`EXPORT_IDC_LIBRARY_PATH=LD_LIBRARY_PATH` は、`IDC_LIBRARY_PATH` 変数の値を `LD_LIBRARY_PATH` という名前でエクスポートします。

変数 `JAVA_COMMAND_LINE` は、コマンドラインの取得に使用されます。使用されていないランチャに対するコマンドライン引数はすべて、コマンドラインに追加されます。**UNIX** システムでは、コマンドラインは解析され、引用符は付けられず、`execv` がコールされます。**Windows** では、シャットダウンのミューテックスが作成され、`CreateProcess` がコマンドラインを使用してコールされます。`CreateProcess` はバックスラッシュを付ける処理を元に戻さないため、注意する必要があります。

ランチャをデバッグするための主な方法は、最終的なコマンドの引数の前に `-debug` フラグを追加する方法です。`$BIN_DIR/debug.log` というファイルを作成することもできます。このファイルは、デバッグ・モードをトリガーしてデバッグ出力を格納します。

ランチャは、動作を制御するために設定または使用される次の構成エントリを認識します。これらの構成変数は、**Content Server** の将来のリリースで変更または削除される可能性があります。

- ❖ `IDC_SERVICE_NAME`: サービスの登録、登録解除、起動および停止に使用される **win32** サービスの名前。
- ❖ `IDC_SERVICE_DISPLAY_NAME`: サービスの登録に使用される **win32** の表示名。
- ❖ `IntradocDir`: 相対パス名のベース・ディレクトリ。
- ❖ `IdcBaseDir`: `IntradocDir` の代替名。
- ❖ `SharedDir`: 特に定義されていない場合は、`$IntradocDir/shared` に設定されます。
- ❖ `SHARED_CONFIG_DIR`: 特に設定されていない場合は、`$SharedDir/config` に設定されます。
- ❖ `OS_DIR`: 特に設定されていない場合は、`$SharedDir/os` に設定されます。
- ❖ `PATH_SEPARATOR`: 特に設定されていない場合は、コロン (:) またはセミコロン (;) のいずれかに設定されます。
- ❖ `STARTUP_CLASS`: ランチャ実行可能ファイルの名前に設定されます。
- ❖ `MUTEX_NAME`: **win32** でシャットダウンのミューテックスを作成するために使用される名前。
- ❖ `BEFORE_WIN_SERVICE_START_CMD`: 設定される場合は、**win32** サービスの起動前に実行されるコマンドライン。
- ❖ `UseRedirectedOutput`: 設定される場合は、**win32** 上のランチャに **Java VM** からの出力をファイルにリダイレクトするように指示します。

- ❖ ServiceStartupTimeout: Java プロセスが win32 で正常に起動するのを待機するのに使用されるタイムアウト。



技術ヒント: Launcher.exe を使用して status.dat ファイルを変更し、JVM コマンドラインの値を変更すると、理論上は Java プログラムを Windows サービスとして実行できます。この方法は、通常の使用にはお薦めしませんが、ランチャの様々な構成方法が明らかになります。

ユーザー・インタフェース

ランチャの UI は、起動するアプリケーションとまったく同じです。たとえば、ランチャの名前を IntradocApp に変更した場合、次のコマンドライン引数を指定して Web レイアウト・エディタを起動します。

```
IntradocApp WebLayout
```

これにより、Web レイアウト・エディタはスタンドアロン・アプリケーションとして起動されます。

デフォルトでは、アプリケーションはコンソール出力なしの状態です。ただし、IdcServer、IdcAdmin、IdcCommandX または Installer を起動すると、Java 出力が画面に表示されます。その他の場合はすべて、出力はクリーナ・インタフェースに対して抑止されます。

バッチ・ローダーやリポジトリ・マネージャなど、アプリケーションによっては、アプリケーションからの Java 出力を表示するほうがよいものがあります。ランチャに Java 出力を画面にダンプさせるには、次のように -console フラグを使用します。

```
IntradocApp RepMan -console
```

これで、リポジトリ・マネージャが起動されたコンソールに出力が書き込まれます。

ランチャの名前を IdcServer、BatchLoader、SystemProperties、あるいは追加パラメータを必要としないその他の Java クラスに変更すると、ダブルクリックするだけで起動できます。その他の場合は、ショートカットを使用するとダブルクリックで起動できます。

ランチャの構成

ランチャを使用するには、まず、Launcher.exe ファイルの名前を変更して、起動するクラス・ファイルと同じ名前の実行可能ファイルにします。一般的な例として、IdcServer.exe や IntradocApp.exe があります。



注意: カスタム・アプリケーションを作成する場合、カスタム・ディレクトリを作成し、Launcher.exe の名前を起動するサービスに変更する必要があります。有効な intradoc.cfg ファイルが実行可能ファイルと同じディレクトリに存在する必要があります。必須パラメータは *IntradocDir* のみです。ただし、他のエントリを追加して Java アプリケーションの起動方法を変更できます。

構成ファイルの例

構成ファイルのエントリの例を示します。

```
<?cfg jcharset="Cp1252"?>
#Content Server Directory Variables
IntradocDir=C:/stellent/idcm1/
CLASSPATH=$COMPUTEDCLASSPATH;$SHAREDDir/classes/jtlds.jar
```

ほとんどすべての Content Server アプリケーションを起動するには、これで十分です。Inbound Refinery など、その他のアプリケーションには、クラスパスにクラスを追加する必要があります。このファイルを変更して、別の Java 仮想マシンを使用して Content Server を実行できるようにすることも可能です。

CLASSPATH は、リストされたエントリの順にクラス・ファイルを検索するように指定されています。つまり、ランチャは *<install_dir>/classes/* ディレクトリ全体を検索してから、共有ディレクトリまたは server.zip ファイル内を調べます。これは、Zip ファイルをパッチせずに Java クラスをオーバーロードする場合に適しています。さらに、Java アプリケーションがランチャとの通信に適した API に準拠している場合は、ランチャを使用してインストール、アンインストールおよび Windows サービスとして実行することができます。ランチャを使用して Java アプリケーションを Windows サービスとして実行する方法の詳細は、IdcServer.java または IdcAdmin.java のソース・コードを参照してください。

COMPUTEDCLASSPATH は、ランチャが使用する CLASSPATH にクラス・ファイルを追加するために使用されます。クラス・ファイルを追加するには、このフラグを上書きします。



注意: intradoc.cfg ファイルは、通常、特定のデータベースに対する JDBC ドライバを組み込むようにインストール時に変更されます。別の JDBC ドライバを使用する場合は、ドライバを *<install_dir>/shared/classes* ディレクトリに配置し、intradoc.cfg ファイルを適宜変更します。

たとえば、Windows システム上で IBM の仮想マシンを使用して Content Server を実行するには、コマンドラインは次のようになります。

```
#customized for running IBM's VM
JAVA_EXE=full path
```

カスタム JVM を使用する場合は、使用する Java 実行可能ファイルへのフルパスを使用します。



警告: JVM コマンドラインを上書きしないでください。上書きする場合は、次のコマンドラインで始めます。

```
JvmCommandLine=$JAVA_EXE $JAVA_OPTIONS $JAVA_SERVICE_EXTRA_OPTIONS
$DEFINE_PREFIXjava.endorsed.dirs=$ENDORSEDPATH $APPEND_CLASSPATH "$CLASSPATH" $STARTUPCLASS
```

使用する JVM を変更することにし、その VM にすべての標準 Sun SDK jar ファイルがある場合は、J2SDK 構成エントリを使用して SDK ディレクトリのルート・ディレクトリを再配置する方が、JAVA_EXE を使用して Java 実行可能ファイルの場所を指定する（この方法は IBM VM には適用できません）よりも適しています。

J2SDK 変数は、Sun SDK ライブラリ（tools.jar など）があるディレクトリを変更します。JAVA_EXE エントリを設定せずにこのエントリを変更すると、Java 実行可能ファイルは、J2SDK のパスの %bin ディレクトリ内にあると判断されます。J2SDK のデフォルト値は、...%shared%os%win32%j2sdk1.4.2_04 です。

\$JAVA_OPTIONS の値を上書きするには、\$JAVA_OPTIONS=server または別の同様の値を使用します。

一般的に使用されるコマンドライン・オプションを次に示します。アスタリスク（*）が付けられたオプションは、Windows プラットフォームでのみ使用できます。マークが付けられていないオプションは、Windows または UNIX プラットフォームで使用できます。

オプション	説明
-console	* Java 出力およびエラー・ストリームがコンソールに表示されるように、Windows コンソール・ウィンドウを開いたままにするようランチャに指示します。
-debug	起動時に使用されたパスおよび変数とともに、起動時のエラーを表示します。
-fileDebug	-debug と似ていますが、このオプションは、デバッグ・データを debug.log ファイルにダンプします。通常、Windows サービスをデバッグするために、intradoc.cfg ファイルの JAVA_OPTIONS または JAVA_SERVICE_EXTRA_OPTIONS にのみ設定します。

オプション	説明
-install	* ランチャによって Windows サービスとしてみなされる Java アプリケーションをインストールするために使用されます。
-install_autostart	* -install オプションと似ていますが、このオプションはサーバーの起動時に起動するアプリケーションをインストールします。
-uninstall	* ランチャによって Windows サービスとしてみなされる Java アプリケーションをアンインストールするために使用されます。
-remove	* -uninstall と同じです。
-dependent <i>service-name</i>	<p>* サービス <i>service-name</i> も稼働しているかどうか Windows サービスを依存させます。</p> <p>このコマンドは、サービスごとに依存コールを行う場合に便利です。</p> <p>たとえば、コンテンツ・サーバーを起動する前にデータベースを起動する場合、データベースの起動に依存するようにコンテンツ・サーバーの起動を指定できます。</p>
-dependent <i>user password</i>	<p>* -install とともに使用され、<i>user</i> によって指定されたユーザーの資格証明とパスワード <i>password</i> を使用してサービスをインストールします。</p> <p>このコマンドは、資格証明に関係なくユーザーをチェックしますが、サービスをインストールしないことがあります。自動起動によってサービスが自動的に実行されるように、ユーザーの資格証明をサービスにまで拡張する必要があります。</p> <p>Inbound Refinery などの特定のサービスには、サービスがより高い権限で実行できるようにするために最後のフラグが必要です。ユーザー名は、一般的な Microsoft 形式の DOMAIN¥User にする必要があります。ユーザーがパスワードを変更すると、サービスはログインできなくなるため、実行されません。</p>
-help	ランチャの使用に関して詳細出力を示します。
-version	ランチャのバージョン番号を表示して終了します。

オプション	説明
<code>-asuser user password</code>	* インストール時に、特定の <i>password</i> の指定された <i>user</i> としてサービスをインストールするために使用されます。
<code>-exec pathname</code>	<code>argv[0]</code> 設定を上書きします。 <code>symlink</code> のターゲットはソースを認識しないため、ターゲットの <i>pathname</i> を指定するために <code>Launcher.sh</code> で使用されます。
<code>-cfg configfilename</code>	計算される設定を割り出す前に読み取る、追加の構成ファイルを指定します。
<code>-idcServiceName servicename</code>	* Windows サービスの名前を指定します。このオプションは、(たとえば、インストール・ディレクトリ全体が削除された場合に) <code>-remove</code> とともに使用して、 Content Server のランチャを使用せずに別の Content Server サービスをアンインストールできます。



技術ヒント: クラスパスをカスタマイズして、**Oracle.dll** ファイルをロードするようにシステム・パスを変更するには、次のようにパスを変更します。

```
PATH=$SHARED_DIR%os%win32%lib%;$SHARED_DIR%search%vdk%_nti40%bin;$SHARED_DIR%search%vdk%_nti40%filters;%OLDPATH
```

カスタムの **.dll** をロードする場合は、それらのファイルを
`<install_dir>/shared/os/win32/lib/` ディレクトリに配置します。

リモートでのサービスのコール

リモートでサービスを使用するには、リモート・マシンに次のファイルを配置する必要があります。

- ❖ `<install_dir>/bin/IdcCommand.exe`
- ❖ `<install_dir>/bin/intradoc.cfg` (コンテンツ・サーバー上のファイルと同じ)
- ❖ `<install_dir>/config/config.cfg`

さらに、リモート・マシン上の `config.cfg` ファイルの `#Additional Variables` セクションに、次の構成エントリを定義する必要があります。

- ❖ `IntradocServerPort=4444`
- ❖ `IntradocServerHostName=IP または DNS`

3

IdcCOMMANDX と IdcCOMMANDUX: ACTIVE X コマンド・ユーティリティ

はじめに

IdcCommandUX は、プログラムによるコンテンツ・サーバー・サービスの実行やファイル・パス情報の取得を可能にする ActiveX コントロールです。各コントロールは、コンテンツ・サーバーが使用する標準 IdcCommand サービスの COM ラッパーとして機能します。

IdcCommandUX は、IdcCommandX と同じですが、マルチバイト言語で機能するように更新され、ASP および SOAP への使用が増した関数が多くなっています。

この章では、IdcCommandX より機能性が高い IdcCommandUX について説明します。

この章の内容は、次のとおりです。

- ❖ [IdcCommandUX の概要](#) (3-2 ページ)
- ❖ [Active Server Page からの IdcCommandUX のコール](#) (3-4 ページ)
- ❖ [リソース・インクルードによるコンテンツの形式設定](#) (3-9 ページ)
- ❖ [リモート・マシンからのコンテンツ・サーバーへの接続](#) (3-11 ページ)
- ❖ [IdcCommandUX のメソッド](#) (3-16 ページ)



注意: IdcCommandUX を使用するには、Visual Basic または Visual C++ の開発環境が必要です。

IdcCOMMANDUX の概要

この項の内容は、次のとおりです。

- ❖ [設定](#) (3-2 ページ)
- ❖ [コール手順](#) (3-2 ページ)
- ❖ [サービスの実行](#) (3-3 ページ)

設定

IdcCommandUX を設定するには、IdcCommandUX 設定ファイルを実行します。このファイルは、Content Server の DVD にある Extras/IdcCommandUX/Setup.exe に格納されています。

コール手順

次の手順のいずれかを使用して IdcCommandUX をコールします。

- ❖ [Visual Basic](#) (3-2 ページ)
- ❖ [Visual C++](#) (3-3 ページ)

Visual Basic

Visual Basic 環境から IdcCommandUX をコールするには、次のようにします。

1. IdcCommandUX をコントロールとして Visual Basic プロジェクトに追加します。
2. 次のようにして、コントロールを作成します。

```
Set idcCmd=CreateObject("Idc.CommandUX")
```
3. 関数 [init \(非推奨\)](#) (3-27 ページ) をコールして *UserName* パラメータと *StellentDir* パラメータを定義し、接続を定義および初期化します。

```
Dim idcCmd
idcCmd.init("UserName", "StellentDir")
```

 - *UserName* パラメータは、IdcCommandUX でコールされるサービスを実行するための権限を持つユーザーを指定します。
 - *StellentDir* パラメータは、intradoc.cfg 構成ファイルがあるコンテンツ・サーバー・ディレクトリへの完全パスを指定します。

次に例を示します。

```
Dim idcCmd

idcCmd.initRemote("sysadmin", "c:\stellent\bin")
```

Visual C++

IdcCommandUX コントロールをプロジェクトに追加し、必要な IdcCommandUX クラスをコールします。

サービスの実行

IdcCommandUX を使用してサービスを実行する際、次のことに注意してください。

- ❖ IdcCommandUX は、有効なユーザー名および intradoc.cfg ファイルの場所を使用して初期化する必要があります。
- ❖ 通信に HDA 形式を使用する必要がある関数には、[computeWebFilePath](#) (3-22 ページ)、[computeNativeFilePath](#) (3-18 ページ) および [computeURL](#) (3-19 ページ) があります。HDA 形式の詳細は、『Working with Content Server Components』を参照してください。
- ❖ [executeCommand](#) (3-24 ページ) では、HDA 形式または SOAP コマンドを使用できます。SOAP を使用するには、[init](#) (非推奨) 関数のかわりに、[initRemote](#) (3-28 ページ) 関数を使用する必要があります。
- ❖ IdcCommandUX は、稼働中のコンテンツ・サーバーへの接続を確立しようとします。接続が確立されない場合は、失敗します。
- ❖ 返された HDA 形式の文字列には、`StatusCode` 変数と `StatusMessage` 変数を使用したコマンドの成否に関する情報が含まれます。
 - コマンドが成功した場合、`StatusCode` はゼロ (0)、`StatusMessage` はログイン・メッセージ (「You are logged in as sysadmin」) となります。
 - コマンドが失敗した場合、`StatusCode` は負数 (-1)、`StatusMessage` はエラー・メッセージとなります。



注意: 詳細は、『Idoc スクリプト・リファレンス・ガイド』を参照してください。



技術ヒント: ランチャ (Java プログラムを Windows サービスとして起動できるようにするネイティブ C++ アプリケーション) の使用方法は、2-7 ページの「[ランチャの使用](#)」を参照してください。

ACTIVE SERVER PAGE からの IdcCOMMANDUX のコール

Active Server Page (ASP) からの IdcCommandUX のコールは、次の手順（この項で後述）で構成されます。

1. [COM オブジェクトの作成](#) (3-5 ページ)
2. [接続の初期化](#) (3-5 ページ)
3. [サービスおよびパラメータの定義](#) (3-5 ページ)
4. [オプション - リソース・インクルードによるコンテンツの形式設定](#) (3-9 ページ)
5. [リソース・インクルードによるコンテンツの形式設定](#) (3-9 ページ)
6. [結果の取得](#) (3-6 ページ)

2つの例を示します。1つはHDAを使用し、もう1つはSOAPを使用しています。例では、指定された基準に合致するコンテンツ・サーバーのドキュメントのリストを取得するために必要な手順を示します。各手順の詳細は、HDAの例を参照してください。

HDA の例

```
' Create COM object
Set idcCmd = CreateObject("Idc.CommandUX")
' Initialize the connection to the server
x = idcCmd.initRemote("/stellent/", "socket:localhost:4444", "sysadmin", "false")
' Define the service
cmd = "@Properties LocalData" + Chr(10)
cmd = cmd + "IdcService=GET_SEARCH_RESULTS" + Chr(10)
' Define the service parameters
cmd = cmd + "ResultCount=5" + Chr(10)
cmd = cmd + "SortField=dInDate" + Chr(10)
cmd = cmd + "SortOrder=Desc" + Chr(10)
cmd = cmd + "QueryText=dDocType=research" + Chr(10)
' Reference a custom component
cmd = cmd + "MergeInclude=ASP_SearchResults" + Chr(10)
cmd = cmd + "ClassStyle=home-spotlight" + Chr(10)
cmd = cmd + "@end" + Chr(10)
' Execute the command
results = idcCmd.executeCommand(cmd)
' Retrieve results
Response.Write(results)
```

関連項目

- 詳細は、3-11 ページの「[リモート・マシンからのコンテンツ・サーバーへの接続](#)」を参照してください。

COM オブジェクトの作成

コードの 1 行目で、COM オブジェクトを作成します。

```
' Create COM object
Set idcCmd = CreateObject("Idc.CommandUX")
```

接続の初期化

コンテンツ・サーバーへの接続を初期化するには、`initRemote` 関数をコールします（すべてのパラメータの詳細は、3-28 ページの「[initRemote](#)」を参照してください）。この例では、次のようにしています。

- ❖ `HttpWebRoot` パラメータは、`config/config.cfg` ファイルに定義されている Web ルートの値を指定します。
- ❖ `idcReference` パラメータは、コンテンツ・サーバーへの接続に関する情報を含む文字列を指定します。この文字列は、`"socket"` の後に `IntradocServerHostName` および `IntradocServerPort` を続けて指定します。
- ❖ `idcUser` は、接続する際のユーザーです。
- ❖ `isSoap` パラメータは、リクエストが SOAP XML 形式か HDA 形式かを示すブール値です。この例では、HDA 形式であるため、`FALSE` です。

```
' Initialize the connection to the server
x = idcCmd.initRemote("/stellent/", "socket:localhost:4444", "sysadmin", false)
```

サービスおよびパラメータの定義

サービスおよびパラメータを定義するには、次の行で構成される HDA 形式の文字列を作成します。

```
@Properties LocalData
service
parameters
@end
```



重要： 必須および任意のパラメータは、コールされるサービスによって異なります。詳細は、『[Services Reference Guide](#)』を参照してください。



注意： この例では、オプションのカスタム・コンポーネントの参照後に `@end` 文字列が作成されます。3-9 ページの「[リソース・インクルードによるコンテンツの形式設定](#)」を参照してください。

カスタム・リソースの参照

次のようにして、カスタム・リソースを参照し、パラメータを ASP からのリソース・インクルードに渡すことができます。

- ❖ カスタム・リソース・インクルードを参照するため、MergeInclude パラメータをインクルードの名前に設定します。

この例では、ASP_SearchResults インクルードを使用して ResultSet ではなく HTML として出力形式を設定しています。詳細は、3-9 ページの「[リソース・インクルードによるコンテンツの形式設定](#)」を参照してください。

- ❖ パラメータをリソース・インクルードに渡すため、変数を名前 / 値ペアとして設定します。

この例では、値が home-spotlight の ClassStyle 変数が、ASP_SearchResults インクルードで使用できます。



注意: @end コードは、HDA 形式の文字列の @Properties LocalData セクションを閉じるために必要です。3-5 ページの「[サービスおよびパラメータの定義](#)」を参照してください。

```
' Reference a custom component
cmd = cmd + "MergeInclude=ASP_SearchResults" + Chr(10)
cmd = cmd + "ClassStyle=home-spotlight" + Chr(10)
cmd = cmd + "@end" + Chr(10)
```

サービスの実行

サービスを実行するために、[executeCommand](#) (3-24 ページ) メソッドをコールします。



注意: サービスの実行後に、[closeServerConnection](#) (3-17 ページ) メソッドを使用して、確実に接続を閉じることができます。

```
' Execute the service
results = idcCmd.executeCommand(cmd)
```

結果の取得

結果は、HTML 形式または ResultSet 形式のいずれかです。

この例では、サービス・コールの結果は HTML 形式となります。

```
' Retrieve results
Response.Write(results)
```

SOAP の例

この例では、次のようにしています。

- ❖ GET_SEARCH_RESULTS サービスがコールされます。
- ❖ サービスのパラメータは、フィールド / 値ペアを使用して次のように定義されています。
 - ResultCount パラメータは、返される結果の数を 5 に設定します。
 - SortField パラメータは、返される結果をリリース日でソートします。
 - SortOrder パラメータは、返される結果を降順にします。
 - QueryText パラメータは、"Content Type matches *research*" という問合せ式を定義します。

次の例に示すように、[initRemote](#) (3-28 ページ) 関数を使用して、SOAP 形式のリクエストの場合、isSOAP を TRUE に設定する必要があります。

```
' Create COM object
Set idcCmd = CreateObject("Idc.CommandUX")
' Initialize the connection to the server
x = idcCmd.initRemote("/stellent/ ", "sysadmin",
    "socket:localhost:4444", true)
' Create the SOAP envelope
cmd = cmd & "<?xml version='1.0' encoding='UTF-8'?>" + Chr(10)
cmd = cmd & "<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://
    schemas.xmlsoap.org/soap/envelope/'>" + Chr(10)
cmd = cmd & "<SOAP-ENV:Body>" + Chr(10)
' Define the service
cmd = cmd & "<idc:service xmlns:idc='http://www.stellent.com/
    IdcService/'>" + Chr(10)
cmd = cmd & "IdcService='GET_SEARCH_RESULTS'">" + Chr(10)
' Define the service parameters
cmd = cmd & "<idc:document>" + Chr(10)
cmd = cmd & "<idc:field name='NoHttpHeaders'>1</idc:field>" +
    Chr(10)
cmd = cmd & "<idc:field name='ClientEncoding'>UTF8</idc:field>" +
    Chr(10)
cmd = cmd & "<idc:field name='QueryText'>dDocType
    &lt;matches&gt; research</idc:field>" + Chr(10)
cmd = cmd & "<idc:field name='ResultCount'>5</idc:field>" +
    Chr(10)
cmd = cmd & "<idc:field name='SortOrder'>Desc</idc:field>" +
    Chr(10)
cmd = cmd & "<idc:field name='SortField'>dInDate</idc:field>" +
    Chr(10)
cmd = cmd & "</idc:document>" + Chr(10)
cmd = cmd & "</idc:service>" + Chr(10)
cmd = cmd & "</SOAP-ENV:Body>" + Chr(10)
cmd = cmd & "</SOAP-ENV:Envelope>" + Chr(10)
' End SOAP envelope and execute the command
results = idcCmd.executeCommand(cmd)
' Retrieve results
Response.Write(results)
```

リソース・インクルードによるコンテンツの形式設定

この項では、IdcCommandUX によって実行されるサービスの出力形式の設定に使用されるカスタム・リソース・インクルードの例を示します。

3-4 ページの「[Active Server Page からの IdcCommandUX のコール](#)」で説明した例では、ASP_SearchResults リソース・インクルードを使用して検索関数の出力形式を設定し、ResultSet ではなく HTML を返します。

```
<@dynamichtml ASP_SearchResults@>
<table border=0>
  <$loop SearchResults$>
    <tr class="site-default">
      <td class="<$ClassStyle$>">
        <a href="<$URL$>" target=new><$dDocTitle$></a><br>
        <$xAbstract$>
      </td>
    </tr>
  <$endloop$>
</table>
<@end@>
```

- ❖ <@dynamichtml ASP_SearchResults@> エントリは、リソース・インクルードの名前を定義します。<@end@> エントリは、リソース定義を終了します。
- ❖ <\$loop SearchResults\$> エントリと <\$endloop\$> エントリの上に定義されるコードは、SearchResults ResultSet の各コンテンツ・アイテムに対して実行されます。この ResultSet には、GET_SEARCH_RESULTS サービスに定義された問合せと合致するドキュメントがすべて含まれます。
- ❖ <td class="<\$ClassStyle\$>"> エントリは、<\$ClassStyle\$> Idoc Script 変数の値を表示します。この例では、API コール時に ClassStyle 値が渡されます。
- ❖ <a href="<\$URL\$>" target=new><\$dDocTitle\$> エントリは、現在のコンテンツ・アイテムのタイトルをファイルへのリンクとして表示します。
- ❖ <\$xAbstract\$> エントリは、現在のコンテンツ・アイテムの概要の値を表示します。

生成され、このリソース・インクルードから Active Server Page に返された HTML の形式は、次のとおりです。

```
<table border=0>
<tr class="site-default">
<td class="home-spotlight">
<a href="/stellent/dir/dir/xyz.htm" target=new>Article 1</a><br>
This is the abstract for Article 1
</td>
<td class="home-spotlight">
<a href="/stellent/dir/dir/xyz.htm" target=new>Article 2</a><br>
This is the abstract for Article 2
</td>
<td class="home-spotlight">
<a href="/stellent/dir/dir/xyz.htm" target=new>Article 3</a><br>
This is the abstract for Article 3
</td>
<td class="home-spotlight">
<a href="/stellent/dir/dir/xyz.htm" target=new>Article 4</a><br>
This is the abstract for Article 4
</td>
<td class="home-spotlight">
<a href="/stellent/dir/dir/xyz.htm" target=new>Article 5</a><br>
This is the abstract for Article 5
</td>
</tr>
</table>
```

この HTML ページをブラウザに表示すると、次のようになります。

<u>Article 1</u>	<u>Article 2</u>	<u>Article 3</u>	<u>Article 4</u>	<u>Article 5</u>
This is the abstract for Article 1	This is the abstract for Article 2	This is the abstract for Article 3	This is the abstract for Article 4	This is the abstract for Article 5

リモート・マシンからのコンテンツ・サーバーへの接続

この項では、Active Server Page から IdcCommandUX を使用して、リモート・マシンからコンテンツ・サーバーへの接続を確立する方法について説明します。次の手順は必須です。

1. [変数の作成](#) (3-13 ページ)
2. [COM オブジェクトの作成](#) (3-13 ページ)
3. [接続の初期化](#) (3-13 ページ)
4. [接続ステータスのリターン](#) (3-13 ページ)
5. [サービスおよびパラメータの定義](#) (3-14 ページ)
6. [サービスの実行](#) (3-15 ページ)
7. [結果の取得](#) (3-15 ページ)

この項の例では、CHECKIN_UNIVERSAL サービスをコールして、リモート・マシンからチェックイン関数を提供します。

ASP ページのコーディング

この項では、リモート・マシンからコンテンツ・サーバーにアクセスするための Active Server Page をコーディングする手順を示します。



重要: このコードでは、エラーの状態をチェックしません。

```

' Create variables
Dim idccommand, sConnect, str
' Create COM object
Set idccommand = Server.CreateObject("idc.CommandUX")
' Initialize the connection to the server
x = idccommand.initRemote ("/stellent/ ", "sysadmin", "socket:localhost:4444", false)
' Return connection status (optional)
sConnect = idccommand.connectToServer
if sConnect then
Response.Write "Connected"
else
Response.Write "Not Connected"
end if
str = "@Properties LocalData" & vbCrLf
' Define the service
str = str + "IdcService=" & "CHECKIN_UNIVERSAL" & vbCrLf
' Define the service parameters
str = str + "doFileCopy=1" & vbCrLf
str = str + "dDocName=RemoteTestCheckin23" & vbCrLf
str = str + "dDocTitle=Test1" & vbCrLf
str = str + "dDocType=ADACCT" & vbCrLf
str = str + "dSecurityGroup=Public" & vbCrLf
str = str + "dDocAuthor=sysadmin" & vbCrLf
str = str + "dDocAccount=" & vbCrLf
str = str + "primaryFile:path=C:/inetpub/Scripts/query2.asp" & vbCrLf
str = str + "@end" & vbCrLf
' Execute the command
res=idccommand.executeCommand(str)
' Return connection status
sClosed = idcCmd.closeServerConnection
if sClosed then
Response.Write "Server connection closed"
else
Response.Write "Failed to close server connection"
end if
' Retrieve results
Response.Write(res)

```


変数の作成

この例のために作成する必要がある変数を次に示します。

- ❖ **idccommand**: COM オブジェクトの名前
- ❖ **sConnect**: コンテンツ・サーバーへの接続のステータス
- ❖ **str**: サービスとそのパラメータを定義する HDA 形式の文字列

```
' Create variables
Dim idccommand, sConnect, str
```

COM オブジェクトの作成

コードの 1 行目で、COM オブジェクトを作成します。

```
' Create COM object
Set idccommand = Server.CreateObject("idc.CommandUX")
```

接続の初期化

コンテンツ・サーバーへの接続を初期化するには、次のようにします。

```
' Initialize the connection to the server
x = idccommand.initRemote ("/stellent/ ", "sysadmin", "socket:localhost:4444", false)
```

接続ステータスのリターン

この例では、[connectToServer](#) (3-23 ページ) メソッドと [closeServerConnection](#) (3-17 ページ) メソッドを使用して、サービスの実行前後の接続ステータス情報を返します。

```

' Return connection status
sConnect = idccommand.connectToServer
if sConnect then
Response.Write "Connected"
else
Response.Write "Not Connected"
end if
...
' Return connection status
sClosed = idcCmd.closeServerConnection
if sClosed then
Response.Write "Server connection closed"
else
Response.Write "Failed to close server connection"
end if

```

サービスおよびパラメータの定義

サービスおよびパラメータを定義するには、次の行で構成される HDA 形式の文字列を作成します。

```

@Properties LocalData
service
parameters
@end

```



重要: 必須および任意のパラメータは、コールされるサービスによって異なります。詳細は、『Services Reference Guide』を参照してください。

この例では、次のようにしています。

- ❖ CHECKIN_UNIVERSAL サービスがコールされます。
- ❖ サービスのパラメータは、フィールド / 値ペアを使用して次のように定義されています。
 - doFileCopy パラメータは、TRUE (1) に設定されます。そのため、チェックインの成功後に、ファイルはハード・ドライブから削除されません。
 - dDocName パラメータは、コンテンツ ID を定義します。
 - dDocTitle パラメータは、タイトルを定義します。
 - dDocType パラメータは、タイプを定義します。
 - dSecurityGroup パラメータは、セキュリティ・グループを定義します。
 - dDocAuthor パラメータは、作成者を定義します。
 - dDocAccount パラメータは、セキュリティ・アカウントを定義します。(アカウントが有効である場合、このパラメータは必須です。)
 - primaryFile パラメータは、ファイルの元の名前と、サーバーから見た場合のファイルの場所への絶対パスを定義します。



重要: 必須パラメータは、コールされるサービスによって異なります。詳細は、『Services Reference Guide』を参照してください。

```
str = "@Properties LocalData" & vbCrLf
' Define the service
str = str + "IdcService=" & "CHECKIN_UNIVERSAL" & vbCrLf
' Define the service parameters
str = str + "doFileCopy=1" & vbCrLf
str = str + "dDocName=RemoteTestCheckin23" & vbCrLf
str = str + "dDocTitle=Test1" & vbCrLf
str = str + "dDocType=ADACCT" & vbCrLf
str = str + "dSecurityGroup=Public" & vbCrLf
str = str + "dDocAuthor=sysadmin" & vbCrLf
str = str + "dDocAccount=" & vbCrLf
str = str + "primaryFile:path=C:/inetpub/Scripts/query2.asp" & vbCrLf
str = str + "@end" & vbCrLf
```

サービスの実行

サービスを実行するために、[executeCommand](#) (3-24 ページ) メソッドをコールします。

```
' Execute the service
res=idccommand.executeCommand(str)
```

結果の取得

この例では、CHECKIN_UNIVERSAL サービス・コールの結果は HTML 形式となります。

```
' Retrieve results
Response.Write(res)
```

IdcCOMMANDUX のメソッド

この項では、次の IdcCommandUX のメソッドについて説明します。

- ❖ [addExtraHeadersForCommand](#) (3-16 ページ)
- ❖ [closeServerConnection](#) (3-17 ページ)
- ❖ [computeNativeFilePath](#) (3-18 ページ)
- ❖ [computeURL](#) (3-19 ページ)
- ❖ [computeWebFilePath](#) (3-22 ページ)
- ❖ [connectToServer](#) (3-23 ページ)
- ❖ [executeCommand](#) (3-24 ページ)
- ❖ [executeFileCommand](#) (3-25 ページ)
- ❖ [forwardRequest](#) (3-26 ページ)
- ❖ [getLastErrorMessage](#) (3-26 ページ)
- ❖ [init](#) (非推奨) (3-27 ページ)
- ❖ [initRemote](#) (3-28 ページ)



重要: すべてのパラメータは、特に指示がないかぎり必須です。

addExtraHeadersForCommand

このコマンドは、別の HTTP に似たヘッダーをコマンドに追加します。

- ❖ セキュリティ上の理由で、一部のパラメータしかヘッダーに渡すことができません。
- ❖ このコマンドの最も一般的な使用法は、EXTERNAL_ROLES および EXTERNAL_ACCOUNTS の値をリクエストに設定することです。
- ❖ 値は、全部で 1 つの文字列とし、改行で区切る必要があります。

例

ASP の例を次に示します。

```
extraHeaders = "EXTERNAL_ROLES=contributor" _  
              + vbCrLf _  
              + "EXTERNAL_ACCOUNTS=my_account"  
idcCmd.addExtraHeadersForCommand(extraHeaders)
```

closeServerConnection

```
Public Sub closeServerConnection()
```

説明

サーバー接続を閉じます。

- ❖ このメソッドは、コールする必要がありません。[executeCommand](#) (3-24 ページ) メソッドが、サービスの実行後に接続を自動的に閉じるためです。接続の状態を管理するのに単に便利のように指定するだけです。

パラメータ

なし

出力

- ❖ 接続が閉じると TRUE を返します。
- ❖ 接続を閉じるのに失敗すると FALSE を返します。

例

この ASP の例では、closeServerConnection メソッドの結果を変数に渡し、if/else 文を使用して接続ステータスのメッセージを返しています。

```
sClosed = idcCmd.closeServerConnection
if sClosed then
Response.Write "Server connection closed"
else
Response.Write "Failed to close server connection"
end if
```

関連項目

- [executeCommand](#) (3-24 ページ)
- [connectToServer](#) (3-23 ページ)

computeNativeFilePath

Public Function computeNativeFilePath(Data As String) As String

説明

HDA 専用の関数。

ネイティブ・ファイルのパスを文字列として返します。

- ❖ 一般に、この関数は、ネイティブ・ファイルを処理して、バルク・ファイルのロードや取得などのアクションを実行するために使用されます。
- ❖ 必須パラメータ (dDocType、dID など) の値を決定するために、DOC_INFO または SEARCH_RESULTS のサービス・コールから返される **ResultSet** を参照できます。
 - DOC_INFO サービスは、前のリビジョンを指定するために使用できます (DOC_INFO は前のリビジョン・ラベルのリストを返します)。
 - SEARCH_RESULTS サービスは、コンテンツ・アイテムの最新のリビジョンを指定するのに十分なデータのみを返します。

パラメータ

- ❖ **Data:** 次のコンテンツ・アイテムを定義する HDA 形式の文字列。
 - **dDocType:** コンテンツ・アイテムのタイプ (ADACCT、FILES など)。
 - **dID:** 生成されたコンテンツ・アイテムのリビジョン ID。
 - **dExtension:** ファイル拡張子 (HCSF、DOC、TXT など)。
 - **dDocAccount:** コンテンツ・アイテムのアカウント。アカウントが有効である場合、このパラメータを定義する必要があります。



注意: コンテンツ ID (dDocName) とコンテンツ・アイテムの内部リビジョン識別子 (dID) を混同しないでください。dID は、特定のリビジョンのコンテンツ・アイテムに対する、生成された参照です。

出力

- ❖ パラメータとして渡される文字列の値として、**NativeFilePath** を定義する文字列を返します。次に例を示します。

NativeFilePath=c:¥stellent¥vault¥adacct¥1.doc

- ❖ StatusCode および StatusMessage を含む HDA 文字列を返します。
 - コマンドが成功した場合、StatusCode はゼロ (0)、StatusMessage はログイン・メッセージ (「You are logged in as sysadmin」) となります。
 - コマンドが失敗した場合、StatusCode は負数 (-1)、StatusMessage はエラー・メッセージとなります。
 - 接続が失敗した場合は、FALSE を返します。

例

HDA 形式の文字列の例を次に示します。

```
String str = "@Properties LocalData\n"+
"dDocType=ADACCT\n"+
"dID=67\n"+
"dExtension=DOC\n"+
"dDocAccount=mainaccount\n"+
"@end\n";
```

computeURL

```
Public Function computeURL(Data As String, IsAbsolute As Boolean) As String
```

説明

HDA 専用の関数。

コンテンツ・アイテムの URL を文字列として返します。

- ❖ 相対 URL または絶対 URL をコンテンツ・サーバーに指定できます。
 - 相対 URL を定義すると、URL はローカル・サーバー上の有効な場所として評価されます。
次に例を示します。
/stellent/groups/Public/documents/FILE/doc.txt
 - 絶対 URL を定義すると、絶対 URL パスが返されます。
次に例を示します。
http://server/stellent/groups/Public/documents/FILE/doc.txt
- ❖ コンテンツ・サーバーのパラメータ (HttpRelativeWebRoot および HttpServerAddress) の値を決定するために、GET_DOC_CONFIG_INFO サービス・コールから返されるプロパティ・データを参照できます。

- ❖ コンテンツ・アイテムの必須パラメータ (dSecurityGroup、dDocType など) の値を決定するために、DOC_INFO または SEARCH_RESULTS のサービス・コールから返される ResultSet を参照できます。
 - DOC_INFO サービスは、前のリビジョンを指定するために使用できます (DOC_INFO は前のリビジョン・ラベルのリストを返します)。
 - SEARCH_RESULTS サービスは、コンテンツ・アイテムの最新のリビジョンを指定するのに十分なデータのみを返します。
- ❖ 特定のリビジョンおよびレンディションの URL を返すには、コンテンツ・アイテムのリビジョン・ラベル (dRevLabel) およびファイル拡張子 (dWebExtension) のエントリを使用します。次に例を示します。


```
dDocName=test10
dRevLabel=2
dWebExtension=pdf
```
- ❖ 最新のリビジョンの URL を返すには、コンテンツ・アイテムのリビジョン・ラベル (dRevLabel) のエントリを省略できます。たとえば、次のようにコンテンツ ID (dDocName) とファイル拡張子 (dWebExtension) だけを定義すると、最新のリビジョンが返されます。


```
dDocName=test11
dWebExtension=html
```

パラメータ

- ❖ Data: 次のコンテンツ・アイテムを定義する HDA 形式の文字列。
 - **HttpRelativeWebRoot:** 相対パスの Web ルート・ディレクトリ (/stellent/ など)。このエントリは、相対 URL の場合は必須、絶対 URL の場合は任意です。
 - **HttpServerAddress:** コンテンツ・サーバーのドメイン名 (testserver17、mycomputer.com など)。(サーバー・アドレスは、http://www.mycomputer.com/ のような完全アドレスではなく、mycomputer.com のような部分 URL として指定します。) このエントリは、絶対 URL の場合は必須、相対 URL の場合は任意です。
 - **dSecurityGroup:** セキュリティ・グループ (Public、Secure など)。
 - **dDocType:** タイプ (ADACCT、FILES など)。
 - **dDocName:** コンテンツ ID (test10、hr_0005467 など)。
 - **dWebExtension:** Web 表示可能ファイルのファイル拡張子 (xml、html、txt など)。
 - **dDocAccount:** コンテンツ・アイテムのアカウント。アカウントが有効である場合、このパラメータを定義する必要があります。
 - **dRevLabel (任意):** コンテンツ・アイテムのリビジョン・ラベル。定義すると、特定のリビジョンが参照されます。
- ❖ IsAbsolute: 絶対 URL アドレスを定義する場合は、TRUE (1) に設定します。



注意: コンテンツ ID (dDocName) とコンテンツ・アイテムの内部リビジョン識別子 (dID) を混同しないでください。dID は、特定のリビジョンのコンテンツ・アイテムに対する、生成された参照です。

出力

- ❖ パラメータとして渡される文字列の値として、URL を定義する文字列を返します。次に例を示します。

URL=http://server/stellent/groups/public/documents/FILE/doc.txt

- ❖ StatusCode および StatusMessage を含む HDA 文字列を返します。
 - コマンドが成功した場合、StatusCode はゼロ (0)、StatusMessage はログイン・メッセージ (「You are logged in as sysadmin」) となります。
 - コマンドが失敗した場合、StatusCode は負数 (-1)、StatusMessage はエラー・メッセージとなります。
 - 接続が失敗した場合は、FALSE を返します。

例

HDA 形式の文字列の例を次に示します。

```
String str = "@Properties LocalData\n"+
  "HttpServerAddress=testserver17\n"+
  "HttpRelativeWebRoot=/stellent/\n"+
  "dDocAccount=mainaccount\n"+
  "dSecurityGroup=Public\n"+
  "dDocType=ADACCT\n"+
  "dDocName=test11\n"+
  "dWebExtension=html\n"+
  "@end\n";
```

computeWebFilePath

Public Function computeWebFilePath(Data As String) As String

説明

HDA 専用の関数。

Web 表示可能ファイルのパスを文字列として返します。

- ❖ 一般に、この関数は、Web 表示可能テキスト・ファイル（XML など）を処理して、バルク・ファイルのロードや取得などのアクションを実行するために使用されます。
- ❖ [computeNativeFilePath](#) (3-18 ページ) のかわりに [computeWebFilePath](#) を使用すると、最新のリビジョンを返すために必要となるのは、特定のリビジョン ID (dID) ではなくコンテンツ ID (dDocName) のみという利点があります。
- ❖ 必須パラメータ (dSecurityGroup、dDocType など) の値を決定するために、DOC_INFO または SEARCH_RESULTS のサービス・コールから返される ResultSet を参照できます。
 - DOC_INFO サービスは、前のリビジョンを指定するために使用できます (DOC_INFO は前のリビジョン・ラベルのリストを返します)。
 - SEARCH_RESULTS サービスは、コンテンツ・アイテムの最新のリビジョンを指定するのに十分なデータのみを返します。

パラメータ

- ❖ Data: 次のコンテンツ・アイテムを定義する HDA 形式の文字列。
 - **dSecurityGroup:** セキュリティ・グループ (Public、Secure など)。
 - **dDocType:** コンテンツ・アイテムのタイプ (ADACCT、FILES など)。
 - **dDocName:** コンテンツ ID (test10、hr_0005467 など)。
 - **dWebExtension:** Web 表示可能ファイルのファイル拡張子 (xml、html、txt など)。
 - **dDocAccount:** コンテンツ・アイテムのアカウント。アカウントが有効である場合、このパラメータを定義する必要があります。



注意: コンテンツ ID (dDocName) とコンテンツ・アイテムの内部リビジョン識別子 (dID) を混同しないでください。dID は、特定のリビジョンのコンテンツ・アイテムに対する、生成された参照です。

出力

- ❖ パラメータとして渡される文字列の値として、WebFilePath を定義する文字列を返します。次に例を示します。

```
WebFilePath=http:¥¥testserver17.stellent.com¥stellent¥groups¥main¥documents¥test.xml
```

- ❖ StatusCode および StatusMessage を含む HDA 文字列を返します。
 - コマンドが成功した場合、StatusCode はゼロ (0)、StatusMessage はログイン・メッセージ (「You are logged in as sysadmin」) となります。
 - コマンドが失敗した場合、StatusCode は負数 (-1)、StatusMessage はエラー・メッセージとなります。
 - 接続が失敗した場合は、FALSE を返します。

例

HDA 形式の文字列の例を次に示します。

```
String str = "@Properties LocalData\n"+
"dDocAccount=mainaccount\n"+
"dSecurityGroup=Public\n"+
"dDocType=ADACCT\n"+
"dDocName=test11\n"+
"dWebExtension=xml\n"+
"@end\n";
```

connectToServer

```
Public Function connectToServer() As Boolean
```

説明

サーバーへの接続を確立します。

- ❖ 接続は、コマンドが実行されるまで開いたままです。コマンドが実行されると、接続は自動的に閉じます。
- ❖ このメソッドは、コールする必要がありません。[executeCommand](#) (3-24 ページ) メソッドが、サービスを実行するために接続を自動的に開くためです。接続の状態を管理するのに単に便利のように指定するだけです。

パラメータ

なし

出力

- ❖ 接続が開くと TRUE を返します。
- ❖ 接続が失敗した場合は、FALSE を返します。

例

この ASP の例では、`connectToServer` メソッドの結果を変数に渡し、`if/else` 文を使用して接続ステータスのメッセージを返しています。

```
sConnect = idcCmd.connectToServer
if sConnect then
Response.Write "Connected"
else
Response.Write "Not Connected"
end if
```

関連項目

- [executeCommand](#) (3-24 ページ)
- [closeServerConnection](#) (3-17 ページ)

executeCommand

```
Public Sub executeCommand(Data As String)
```

説明

コンテンツ・サーバー・サービスを実行します。

- ❖ このメソッドは、接続がすでに確立されているかどうかを `connectToServer` コールを使用して評価します。接続が存在する場合は、オープン接続を使用します。接続が存在しない場合は、接続を確立します。
- ❖ コマンドが完了すると、接続は閉じます。

パラメータ

- ❖ **Data:** IdcService コマンドおよびサービス・パラメータを定義する HDA 形式の文字列。次に例を示します。

```
@Properties LocalData
IdcService=GET_SEARCH_RESULTS
ResultCount=5
SortField=dInDate
SortOrder=Desc
QueryText=dDocType=research
@end@
```

前述の例（3-7 ページの「[SOAP の例](#)」）に示したような SOAP 形式のメッセージにすることもできます。3-28 ページの「[initRemote](#)」も参照してください。

出力

- ❖ 元のリクエストとともに結果を保持する HDA ファイルを表す文字列を返します。
- ❖ StatusCode および StatusMessage を含む HDA 文字列を返します。
 - コマンドが成功した場合、StatusCode はゼロ (0)、StatusMessage はログイン・メッセージ（「You are logged in as sysadmin」）となります。
 - コマンドが失敗した場合、StatusCode は負数 (-1)、StatusMessage はエラー・メッセージとなります。
 - 接続が失敗した場合は、FALSE を返します。
- ❖ SOAP リクエストが送信された場合、返される文字列は SOAP 形式の XML です。

例

この ASP の例では、cmd 変数によって定義されるデータ文字列に指定されたコマンドを実行します。

```
results = idcCmd.executeCommand(cmd)
```

関連項目

- [connectToServer](#) (3-23 ページ)
- [closeServerConnection](#) (3-17 ページ)

executeFileCommand

```
executeFileCommand (requestString)
```

説明

この関数は、サービス・リクエストを実行した後、RAW レスポンスをクライアントにパイプするために使用されます。このコマンドは、executeCommand と同一ですが、Active Server Page (ASP) 以外ではコールできません。

- ❖ コンテンツ・サーバーからのレスポンスは、クライアントのブラウザにリダイレクトされます（これは、ASP で操作できる文字列としてレスポンスが指定される executeCommand によるレスポンスとは異なります）。
- ❖ ASP を解して通してコンテンツ・サーバーからクライアント・ブラウザにバイナリ・ファイルを転送する必要がある GET_FILE および同様のサービスに使用すると便利です。

- ❖ この関数は、リクエスト・パラメータが環境変数として渡されないかぎり、追加のヘッダーを返します。
- ❖ `requestString` は、サービス・リクエストの名前です。
- ❖ 詳細は、3-24 ページの「[executeCommand](#)」を参照してください。

パラメータ

なし

forwardRequest

`forwardRequest()`

説明

この関数は、コンテンツ・サーバーにポストされたマルチパート・フォームを転送するために使用されます。チェックインの実行に使用すると便利です。

パラメータ

なし

getLastErrorMessage

`getLastErrorMessage()`

説明

このメソッドは、通信エラーまたは構成エラーに関する特定のエラーの詳細を取得します。たとえば、接続を確立するために適切なホスト名を指定しなかった場合、このメソッドは接続エラーを返します。リクエストに対する戻り値の一部として **Content Server** がエラーを返す場合、このメソッドは値を返しません。

パラメータ

なし

例

この例では、オブジェクトを作成し、サーバーへの接続を初期化します。

```
Set idcCmd = Server.CreateObject("Idc.CommandUX")
```

```
x = idcCmd.init("sysadmin", "c:\%stellent%\bin")
If x = false Then
y = idcCmd.getLastErrorMessage()
Response.Write(y)
End If
```

init (非推奨)

```
Public Function init(Username As String, StellentDir As String) As Boolean
```

説明

これは非推奨の関数です。 [initRemote](#) (3-28 ページ) を使用してください。

この関数は、コンテンツ・サーバーへの接続を初期化します。

- ❖ この関数は、コンテンツ・サーバーへの接続を確立する、またはコンテンツ・サーバー・サービスを実行する前に、コールする必要があります。

必須パラメータ

- ❖ **Username:** IdcCommandUX コールに指定されたサービスを実行するための権限を持つ有効なコンテンツ・サーバー・ユーザー。
- ❖ **StellentDir:** intradoc.cfg 構成ファイルがあるディレクトリへの完全パス。このパラメータを設定しない場合は、現在の作業ディレクトリが使用されます。

出力

サービスの実行が失敗すると FALSE を返します。

例

この ASP の例では、サーバーへの接続を初期化します。

```
❖ x = idcCmd.init("sysadmin", "c:/stellent_english/bin/")
```

initRemote

```
initRemote(HttpWebRoot, idcReference, idcUser, isSoap)
```

説明

この関数は、コンテンツ・サーバーに接続するようにモジュールを初期化します。次の例のように、最初に idcCmd を宣言する必要があります。

```
Dim idcCmd  
idcCmd.initRemote("stellent", "socket:test204:4444", "sysadmin", "false")
```

必須パラメータ

- ❖ HttpWebRoot: HttpWebRoot の IdocScript 値。
- ❖ idcReference: socket:hostname:port 書式の、コンテンツ・サーバーへの接続方法に関する情報を含む文字列。通常、socket:localhost:4444 です。hostname は IntradocServerHostName と、port は IntradocServerPort と同一である必要があります。
- ❖ idcUser: 接続する際のユーザー。
- ❖ isSoap: リクエストが SOAP XML 形式か HDA 形式かを示すブール値。

4

IdcClient OCX コンポーネント

はじめに

リモート・コンテンツ・サーバーへの接続および Content Server サービスの実行のために、Object Linking and Embedding Control Extension (OCX) コントロールが提供されます。IdcClient OCX コントロールは、Content Server 内のコンテンツおよびコンテンツ管理機能にアクセスするために Windows Visual Basic 開発環境内で使用します。

この項では、IdcClient OCX コントロールと設定手順について説明し、イベント、メソッドおよびプロパティを示します。IdcClient.ocx コントロールは、リモート・コンテンツ・サーバーへの接続と一般的なサーバー機能を実行するために使用されます。

この章の内容は、次のとおりです。

- ❖ [IdcClient OCX の説明](#) (4-2 ページ)
- ❖ [IdcClient OCX コントロールの設定](#) (4-5 ページ)
- ❖ [IdcClient のイベント](#) (4-17 ページ)
- ❖ [IdcClient のメソッド](#) (4-19 ページ)
- ❖ [IdcClient のプロパティ](#) (4-36 ページ)



注意: IdcClient OCX コンポーネントを使用するには、Visual Basic または Visual C++ の開発環境が必要です。

IdcClient OCX の説明

この項では、IdcClient OCX コントロールの概要を説明し、イベント、メソッドおよびプロパティに関する基本情報を示します。また、IdcClient OCX インタフェースについても説明します。

この項の内容は、次のとおりです。

- ❖ [概要](#) (4-2 ページ)
- ❖ [イベント、メソッドおよびプロパティ](#) (4-3 ページ)
- ❖ [IdcClient OCX インタフェース](#) (4-4 ページ)

概要

IdcClient は、サービスの実行やファイル・パス情報の取得といったアクションをプログラムで実行できるようにする ActiveX コントロールです。IdcClient コントロールは、Microsoft Internet Explorer ブラウザのラッパーでもあります。

IdcClient OCX コントロールは、Unicode 規格を使用するように設計され、ほとんどの場合に UTF-8 形式でコンテンツ・サーバーとデータを交換します。Unicode は、1 文字当たり 2 バイト (16 ビット) のストレージを使用し、様々な言語 (英語、日本語、アラビア語など) で使用される文字を表すことができます。英語の ASCII (情報交換用米国標準コード) 文字は 1 バイト (8 ビット) しか必要としないため、ASCII 文字は各 Unicode 文字の上位バイトがゼロで表されます。



重要: IdcClient OCX は、Win9x プラットフォーム上で Unicode アプリケーションを実行できるようにする Microsoft Layer for Unicode 上に構築されます。Win9x プラットフォーム上に IdcClient OCX コントロールを配布する際に、unicows.dll も配布する必要があります。このコンパニオン DLL は、Windows ベースのシステム上には配布できません。



注意: Unicode 規格の詳細は、Web で Unicode Consortium (<http://www.unicode.org/>) を参照してください。

ほとんどの場合、メソッドは、シリアライズされた HDA 形式を通信に使用します。シリアライズされた HDA 形式は、通信に使用される Java メソッドです。返されるシリアライズされた HDA 形式の文字列には、コマンドの成否に関する情報が含まれます。

IdcClient OCX コントロールは、メソッド・コールによって実行できる機能を備えています。メソッドは、アクションを実行し、結果を返すこともあります。情報は、パラメータを使用してメソッドに渡されます。関数には、パラメータをとらないもの、パラメータを1つとるもの、パラメータを複数とるものがあります。たとえば、文字列として渡される2つのパラメータを持つ関数は、次の書式を使用します。

```
Function(Parameter As String, Parameter As String) As String
```



注意：詳細は、『Services Reference Guide』を参照してください。

- ❖ IdcClient OCX を使用すると、サービスを実行するクライアント・アプリケーションを作成できます。OCX コントロールは、コマンドおよびパラメータを含む名前 / 値ペアを取り、指定されたサービスをコールします。実行結果は、コール側プログラムに渡されます。
- ❖ IdcClient OCX には、コマンドを実行するユーザー名およびパスワードが必要です。ユーザーには、コマンドを実行するための適切な権限が必要です。コマンドには、管理アクセス・レベルが必要なものもあれば、書込み権限のみが必要なものもあります。

イベント、メソッドおよびプロパティ

IdcClient OCX コントロールは、リモート・コンテンツ・サーバーへの接続およびサーバー機能の実行に使用されます。この項では、Visual Basic のイベント、メソッドおよびプロパティの基本的な概要について説明します。

OCX イベント

イベントは、ユーザーまたはサーバーがアクションを実行すると実行されます。

次に例を示します。

- ❖ IntradocBrowserPost イベント：ユーザーがフォームをブラウザ内から送信するたびに実行されます。
- ❖ IntradocServerResponse イベント：サーバーがリクエストされたアクションを完了すると実行されます。



注意：詳細は、4-17 ページの「[IdcClient のイベント](#)」を参照してください。

OCX メソッド

Visual Basic 標準コントロールは、すべての Visual Basic 開発環境に共通のメソッドを備えています。また、IdcClient OCX コントロールは、この特殊なコントロールに固有のプライベート・メソッドを備えています。これらのメソッドは、特性の設定ではなく、アクションの実行または初期化に使用されます。

次に例を示します。

- ❖ AboutBox() メソッド: 製品バージョン情報を示す「About」ボックスを起動します。
- ❖ GoCheckinPage メソッド: 新しいコンテンツ・アイテムまたはコンテンツ・アイテムのリビジョンをチェックインします。



注意: 詳細は、4-19 ページの「[IdcClient のメソッド](#)」を参照してください。

OCX プロパティ

プロパティは、オブジェクトを記述またはフォーマットし、コードにより、あるいは Visual Basic 開発環境でプロパティ・ウィンドウを使用して変更できます。プロパティは、オブジェクトの基本的な特性を記述します。

次に例を示します。

- ❖ UserName プロパティ: 割り当てられたユーザー名を指定します。
- ❖ WorkingDir プロパティ: ダウンロードされたファイルが格納される場所を指定します。



注意: 詳細は、4-36 ページの「[IdcClient のプロパティ](#)」を参照してください。

IdcClient OCX インタフェース

IdcClient OCX コントロールは、Content Server 内のコンテンツおよびコンテンツ管理機能にアクセスするために Windows Visual Basic 開発環境内で使用します。OCS 統合は、ビジュアル開発環境でサービスをコールする、またはリモート・コンテンツ・サーバーに接続するように設計されています。

ほとんどの場合、メソッドは、シリアルライズされた HDA 形式を通信に使用します。返されるシリアルライズされた HDA 形式の文字列には、コマンドの成否に関する情報が含まれます。障害が発生した場合、StatusCode は負数となり、StatusMessage はエラーを示します。返される HDA に StatusCode パラメータが含まれていない場合、サービス・コールは成功しました。

IdcCLIENT OCX コントロールの設定

この項では、IdcClient OCX コンポーネントの設定に必要な手順について説明します。
また、Microsoft Visual Basic 開発環境でのビジュアル・インタフェースの作成方法についても説明します。

この項の内容は、次のとおりです。

- ❖ [コンポーネントの設定](#) (4-5 ページ)
- ❖ [ビジュアル・インタフェースの作成](#) (4-6 ページ)

コンポーネントの設定

次の手順に従って、Microsoft Visual Basic 開発環境で IdcClient OCX コンポーネントを設定します。

1. 新しいプロジェクトを作成します。
2. 「**Project**」 → 「**Components**」を選択します。
3. システムの IdcClient.ocx ファイルを参照して、「**Open**」をクリックします。
IdcClient モジュールが「**Component Controls**」リストに追加されます。
4. IdcClient ActiveX Control モジュールのチェック・ボックスが選択されていることを確認し、「**OK**」をクリックします。
IdcClient OCX コントロールが、コントロールのリストに入ります。
5. (オプション) Visual Basic 開発環境を使用してユーザー独自のビジュアル・インタフェースを構築できます。あるいは、4-6 ページの「[ビジュアル・インタフェースの作成](#)」に記載されている手順に従って基本的なビジュアル・インタフェースを構築できます。

ビジュアル・インタフェースの作成

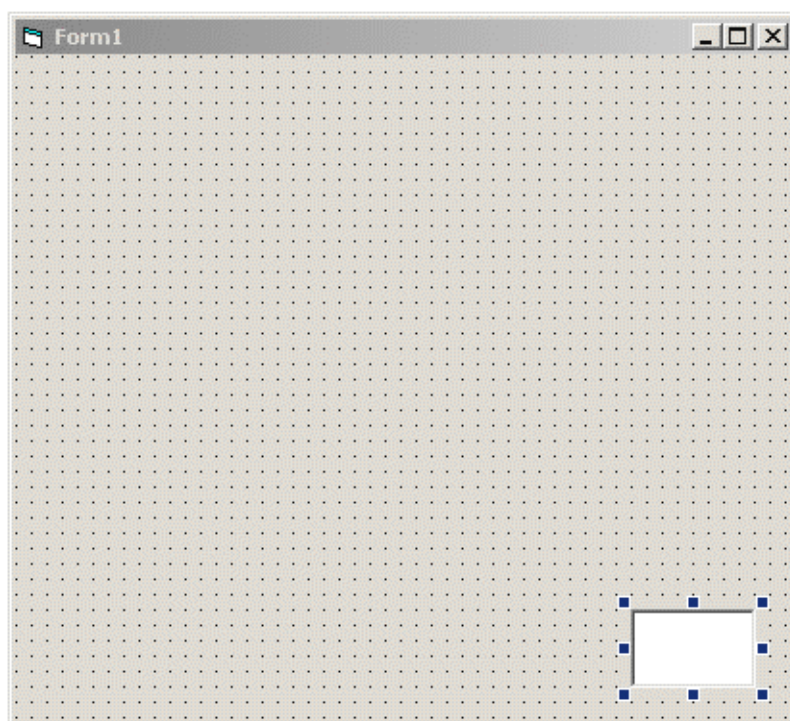


注意： Visual Basic プロジェクトが作成され、IdcClient OCX コントロールがコントロールのリストに入っていることを前提とします。詳細は、4-5 ページの「[コンポーネントの設定](#)」を参照してください。

次の手順に従って、基本的なビジュアル・インタフェースを構築します。

1. コントロールを選択して Visual Basic フォーム上に描画します（図 4-1 を参照）。

図 4-1 Visual Basic フォーム上に描画された OCX コントロール



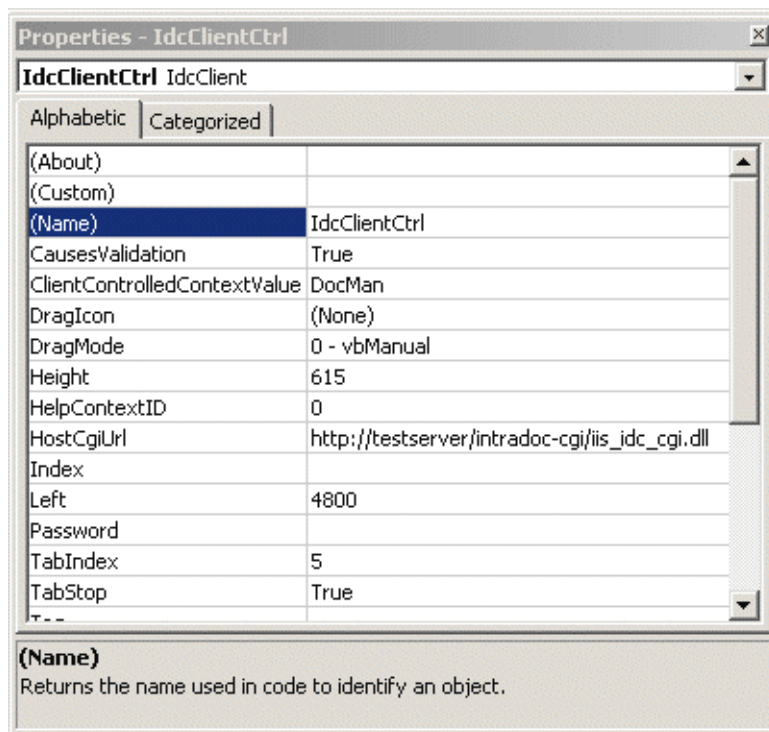
2. Properties ウィンドウのドロップダウン・リストから、IdcClient OCX コントロールを選択します（Properties ウィンドウが現在表示されていない場合は、メイン・メニューから「View」→「Properties Window」を選択します）。

3. IdcClient OCX コントロールの名前を IdcClientCtrl に変更します。
4. 特定のインスタンスの iss_idc_cgi.dll を参照するように HostCgiUrl を定義します (図 4-2 を参照)。

次に例を示します。

`http://testserver/intradoc-cgi/iss_idc_cgi.dll`

図 4-2 編集後の IdcClient プロパティ

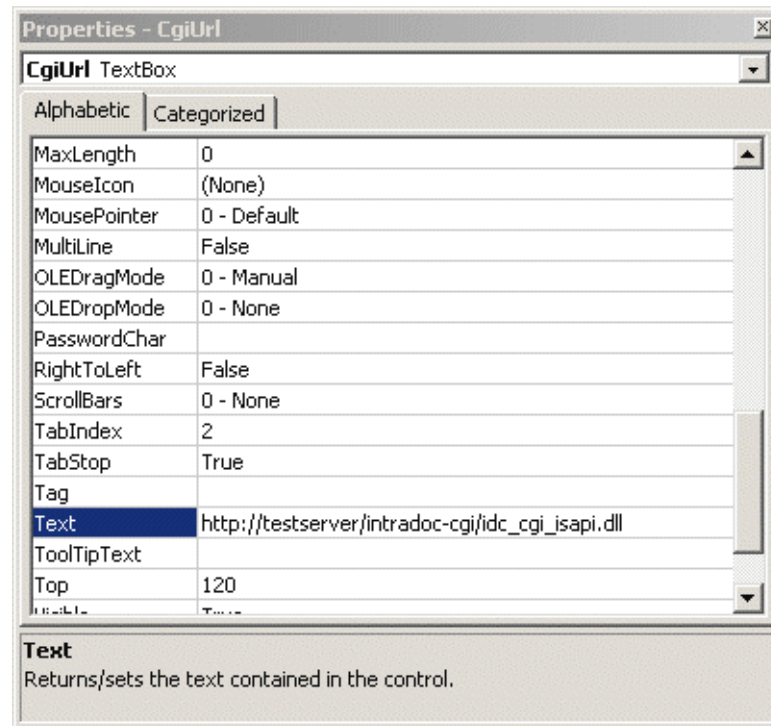


5. フォーム上に、テキストボックスを描画して名前を CgiUrl とします。
6. 「Text」フィールドに、表示されるテキストとして HostCgiUrl 値を入力します (図 4-3 を参照)。

次に例を示します。

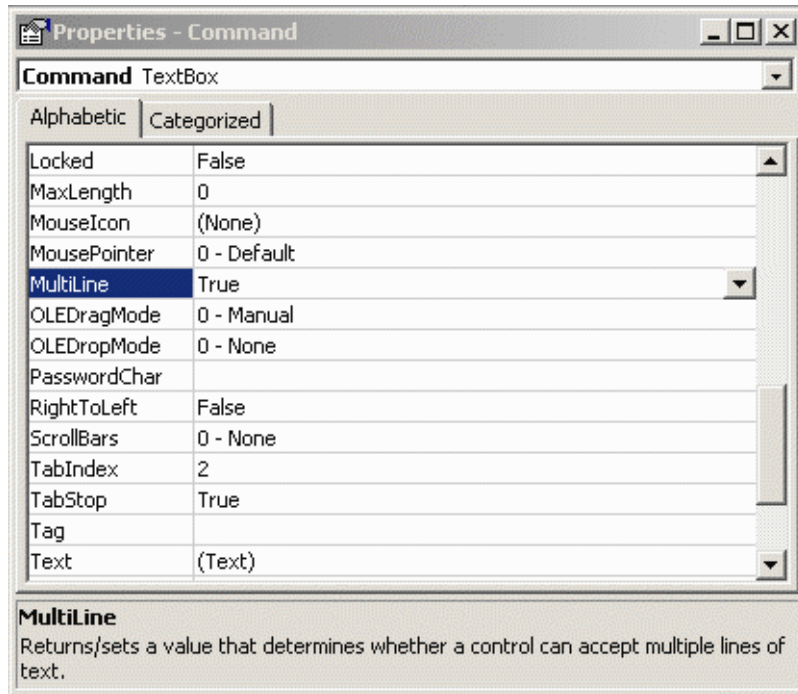
`http://testserver/intradoc-cgi/iss_idc_cgi.dll`

図 4-3 編集後の CgiUrl テキストボックスのプロパティ



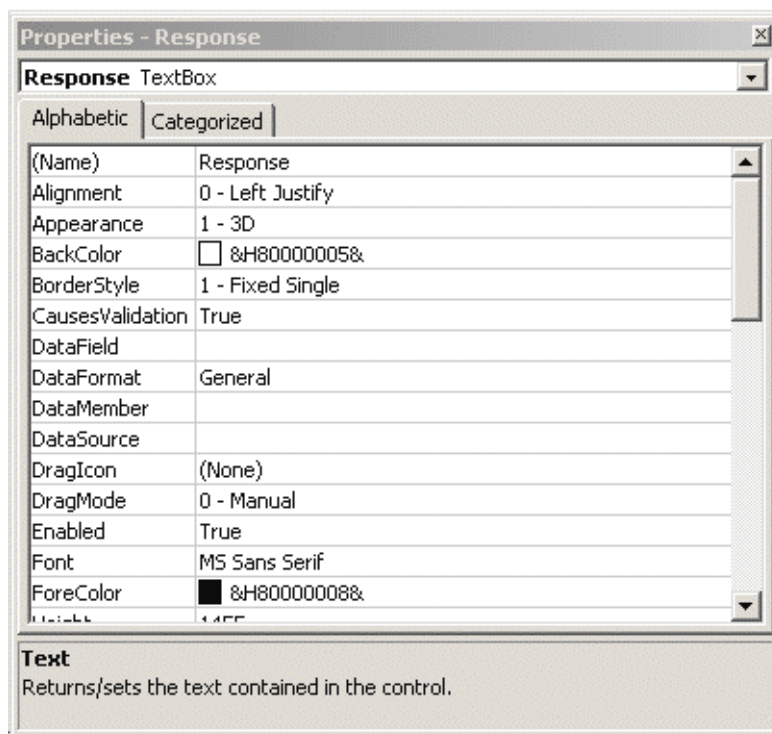
7. フォーム上に、テキストボックスを描画して名前を **Command** とします。
8. 「Text」 フィールドの入力を消去（空白のままに）して、「MultiLine」を **True** に設定します（図 4-4 を参照）。

図 4-4 編集後の Command テキストボックスのプロパティ



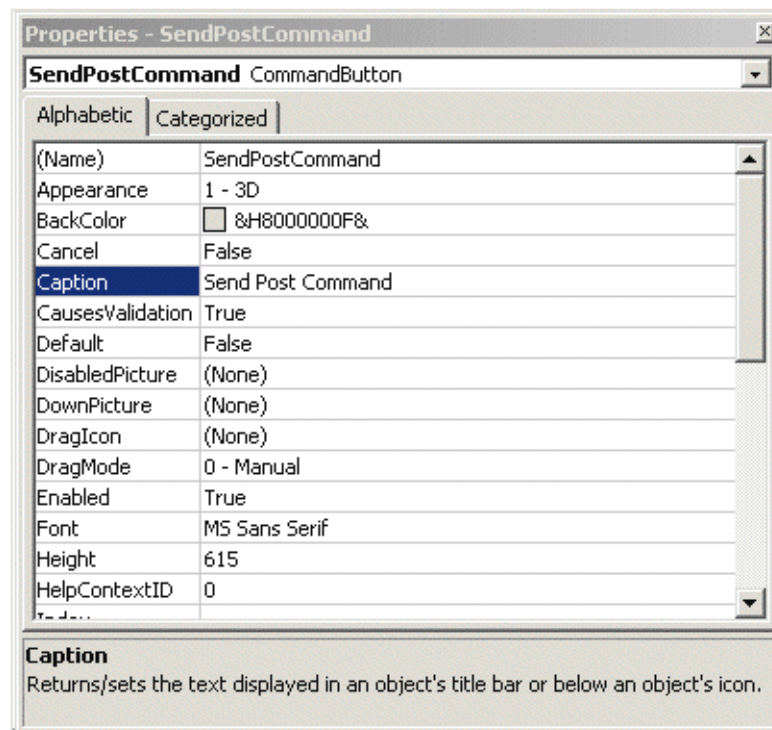
9. フォーム上に、テキストボックスを描画して名前を **Response** とします。
10. 「Text」 フィールドの入力を消去（空白のままに） します。

図 4-5 編集後の Response テキストボックスのプロパティ



11. フォーム上に、ボタンを描画して名前を `SendPostCommand` とします。
12. 「Caption」 フィールドに、表示されるテキストとして「Send Post Command」と入力します（図 4-6 を参照）。

図 4-6 編集後の `SendPostCommand` コマンドボタンのプロパティ



13. フォームで、「View」 → 「Code」を選択します。

14. ドロップダウン・リストから「SendPostCommand」、「Click」の順に選択し、次のアクションを実行するようにコードを変更します（図 4-7 を参照）。

- ❖ Host Cgi Url を設定する。
- ❖ コマンドを発行する。
- ❖ （オプション）LF を CRLF に置換し、編集コントロールでの表示を読みやすくなるようにする。
- ❖ レスポンスを表示する。

次に例を示します。

```
Dim R As String
IdcClientCtrl.HostCgiUrl = CgiUrl.Text
R = IdcClientCtrl.1.SendPostCommand(Command.Text)
R = Replace(R, vbLf, vbCrLf)
Response.Text = R
```

図 4-7 編集後の SendPostCommand_Click コード

```
Private Sub SendPostCommand_Click()
    Dim R As String

    ' Set the Host CGI Url
    IdcClientCtrl.HostCgiUrl = CgiUrl.Text

    ' Issue the command
    R = IdcClientCtrl.SendPostCommand(Command.Text)

    ' Optional--replace LF with CRLF here
    R = Replace(R, vbLf, vbCrLf)

    ' Display the response
    Response.Text = R

End Sub
```

15. ドロップダウン・リストから「Form」、「Load」の順に選択し、次の行を追加してコンテンツ・サーバーのログイン・プロンプトを設定します（図 4-8 を参照）。

```
IdcClientCtrl.UseBrowserLoginPrompt = True
IdcClientCtrl.UseProgressDialog = True
```

図 4-8 編集後の Form_Load コード

```
Private Sub Form_Load()
    IdcClientCtrl.UseBrowserLoginPrompt = True
    IdcClientCtrl.UseProgressDialog = True
End Sub
```

16. (オプション) Cgi Url、Command、Response など、適切な説明的ラベルを追加します (図 4-9 を参照)。

図 4-9 説明的ラベルが付いたビジュアル・インタフェース

The screenshot shows a Windows-style window titled "Form1". Inside the window, there are three labeled input areas and one button. The first area is labeled "Cgi Url:" and contains a text box with the URL "http://testserver/intradoc-cgi/idc.cgi_isapi.dll". Below this is a larger area labeled "Command" which is an empty text box. Below that is another large area labeled "Response" which is also an empty text box. At the bottom of the form is a button labeled "Send Post Command". The entire form has a dotted grid background, and the window has standard minimize, maximize, and close buttons in the title bar.

17. 「Run」 → 「Start」 を選択してビジュアル・インタフェースをテストします。

図 4-10 完成したビジュアル・インタフェース

The image shows a Windows-style window titled "Form1". Inside the window, there are three main sections: "Cgi Url:" with a text box containing "http://testserver/intradoc-cgi/idc_cgi_isapi.dll", "Command" with a large empty text area, and "Response" with a large empty text area that has scrollbars on the right and bottom. At the bottom of the window, there is a button labeled "Send Post Command" and a small empty square box to its right.

18. 「Command」 フィールドに書式設定したコマンドを入力します（図 4-11 を参照）。

たとえば、次のコマンドではユーザーを追加します。

```
@Properties LocalData  
IdcService=ADD_USER  
dName=user99  
dUserAuthType=Local  
@end
```



注意：ADD_USER サービスの詳細は、『Services Reference Guide』を参照してください。

図 4-11 コマンドが定義されたビジュアル・インタフェース

Form1

Cgi Url:

Command:

Response:

19. 「**Send Post Command**」 ボタンをクリックしてコマンドを実行します。返された結果が「**Response**」フィールドに表示されます（図 4-12 を参照）。

図 4-12 返された結果が表示されたビジュアル・インタフェース

The screenshot shows a window titled 'Form1' with three main sections: 'Cgi Url', 'Command', and 'Response'. The 'Cgi Url' field contains 'http://testserver/intradoc-cgi/idc_cgi_isapi.dll'. The 'Command' field contains the following text: '@Properties LocalData', 'IdcService=ADD_USER', 'dName=user99', 'dUserAuthType=Local', and '@end'. The 'Response' field contains the following text: '<?hda version="6.2 (021115)" jcharset=UTF8 encoding=utf-', '@Properties LocalData', 'refreshSubjects=', 'blFieldTypes=StatusMessage message,dUserArriveDate dat', and 'refreshMonikers='. Below the 'Response' field is a 'Send Post Command' button.

コマンドの検証

20. Web ブラウザで、コンテンツ・サーバーに管理者としてログインします。
21. 「**Administration**」 → 「**Admin Applets**」 を選択します。
22. 「**User Admin**」 リンクをクリックします。アプレットが起動し、追加されたユーザー（user99 など）が表示されます。

IdcCLIENT のイベント

イベントは、ユーザーまたはサーバーがアクションを実行すると実行されます。
IdcClient OCX イベントを次に示します。

- ❖ [IntradocBeforeDownload](#) (4-17 ページ)
- ❖ [IntradocBrowserPost](#) (4-17 ページ)
- ❖ [IntradocBrowserStateChange](#) (4-18 ページ)
- ❖ [IntradocRequestProgress](#) (4-18 ページ)
- ❖ [IntradocServerResponse](#) (4-18 ページ)

IntradocBeforeDownload

ファイルのダウンロード前に実行されます。

- ❖ ダウンロード前に必要なサーバーのアクションおよび更新を初期化します。

パラメータ

このイベントは、次のパラメータを渡します。

- ❖ `ByVal params As String`
- ❖ `cancelDownload As Boolean`

IntradocBrowserPost

フォームがブラウザ内から送信されるたびに実行されます。

パラメータ

このイベントは、次のパラメータを渡します。

- ❖ `ByVal url As String`
- ❖ `ByVal params As String`
- ❖ `cancelPost As Boolean`

IntradocBrowserStateChange

ブラウザの状態が変わるたびに実行されます。

パラメータ

このイベントは、次のパラメータを渡します。

- ❖ ByVal browserStateItem As String
- ❖ ByVal enabled As Boolean

IntradocRequestProgress

サーバーからの進行状況レポートの送信のリクエストを実行します。このイベントが発生するのは、メソッドのコール後のみです。

パラメータ

このイベントは、次のパラメータを渡します。

- ❖ ByVal statusData As String
- ❖ ByVal isDone As Boolean

IntradocServerResponse

サーバーがリクエストされたアクションを完了した後に実行されます。たとえば、ファイルのダウンロード後などです。このイベントは、サーバーからのレスポンスである HDA エンコード・データを処理します。このイベントが発生するのは、アクションがブラウザで実行される場合のみです。

パラメータ

このイベントは、次のパラメータを渡します。

- ❖ ByVal response As String

IdcCLIENT のメソッド

次の IdcClient OCX メソッドを使用できます。

- | | |
|---|---|
| ❖ AboutBox (4-20 ページ) | ❖ NavigateCgiPage (4-29 ページ) |
| ❖ Back (4-20 ページ) | ❖ RefreshBrowser (4-30 ページ) |
| ❖ CancelRequest (4-21 ページ) * | ❖ SendCommand (4-30 ページ) * |
| ❖ DoCheckoutLatestRev (4-21 ページ) | ❖ SendPostCommand (4-31 ページ) * |
| ❖ DownloadFile (4-22 ページ) | ❖ SetFocus (4-31 ページ) |
| ❖ DownloadNativeFile (4-23 ページ) | ❖ ShowDMS (4-32 ページ) |
| ❖ Drag (4-24 ページ) | ❖ ShowDocInfoLatestRev (4-32 ページ) |
| ❖ EditDocInfoLatestRev (4-24 ページ) | ❖ ShowWhatsThis (4-33 ページ) |
| ❖ Forward (4-25 ページ) | ❖ StartSearch (4-33 ページ) |
| ❖ GoCheckinPage (4-25 ページ) | ❖ Stop (4-33 ページ) |
| ❖ Home (4-26 ページ) | ❖ UndoCheckout (4-34 ページ) |
| ❖ InitiateFileDownload (4-27 ページ) * | ❖ ViewDocInfo (4-34 ページ) |
| ❖ InitiatePostCommand (4-27 ページ) * | ❖ ViewDocInfoLatestRev (4-35 ページ) |
| ❖ Move (4-28 ページ) | ❖ ZOrder (4-35 ページ) |
| ❖ Navigate (4-29 ページ) | |

アスタリスク (*) が付けられたメソッドは、ブラウザ・アクティビティとは関係なく、値を返すメソッドです。



重要: すべてのパラメータは、特に指示がないかぎり必須です。

AboutBox

Sub AboutBox()

説明

製品のバージョン情報を示す「About」ボックスを起動します。

- ❖ このメソッドは、製品の「About」ボックスを表示します。
- ❖ コールを実行できない場合、FALSE を返します。

パラメータ

なし

Back

Sub Back()

説明

前の HTML ページを表示します。

- ❖ 前画面に戻ります。
- ❖ ユーザーに表示する前の HTML ページをキャッシュされた情報から取得します。

パラメータ

なし

CancelRequest

Function CancelRequest() As Boolean

説明

このメソッドは、現在アクティブなリクエストを取り消します。リクエストを取り消すことができない場合、または現在アクティブなリクエストがない場合は、FALSE を返します。

パラメータ

なし

出力

次のブール値を返します。

- ❖ リクエストが取り消された場合は、TRUE を返します。
- ❖ 取消しリクエストを実行できない場合は、FALSE を返します。

DoCheckoutLatestRev

Sub DoCheckoutLatestRev(docName As String, curID As String)

説明

コンテンツ・アイテムの最新のリビジョンをチェックアウトまたはロックします。

- ❖ コンテンツ・アイテム名およびバージョン・ラベルから、コンテンツ・アイテムの最新のリビジョンをチェックアウトします。
- ❖ IntradocServerResponse イベントを実行します。イベントは、メソッドの発生前に実行されます。詳細は、4-17 ページの「[IdcClient のイベント](#)」を参照してください。



注意: curID は、コンテンツ・アイテムのバージョン・ラベルです。生成されたコンテンツ・アイテムのリビジョン ID ではありません。

この関数は、次の情報を返します。

- ❖ dID および dDocName を含むシリアルライズされた HDA
- ❖ 最新のリビジョンがチェックアウトできない、またはシステムで見つからない場合、FALSE
- ❖ パラメータとして渡されたデータ

パラメータ

- ❖ docName: ユーザーが割り当てたコンテンツ・アイテム名。
- ❖ curID: 最新のリビジョンに対する一意の識別子。任意です。

DownloadFile

```
Function DownloadFile(command As String, filename As String) As String
```

説明

定義されたファイルをダウンロードします。

- ❖ 現在関連付けられているコマンドおよびファイル・タイプから、変換後ファイルのファイル・ダウンロードを実行します (DownloadNativeFile と比較)。
- ❖ IntradocBeforeDownload イベントを実行します。イベントは、メソッドの発生前に実行されます。詳細は、4-17 ページの「[IdcClient のイベント](#)」を参照してください。

この関数は、次の情報を返します。

- ❖ ステータス・コードおよびステータス・メソッドを含むシリアル化された HDA
- ❖ パラメータとして渡されたデータ
- ❖ 指定されたファイルをダウンロードできない場合、FALSE

パラメータ

- ❖ command: 現在関連付けられているコマンド。
- ❖ filename: ファイル形式。これは、PDF、HTM、その他のサポートされる形式などのファイル・タイプです。

DownloadNativeFile

```
Function DownloadNativeFile(id As String, docName As String, filename As String) As String
```

説明

定義されたネイティブ・ファイルをダウンロードします。

- ❖ コンテンツ・アイテムのリビジョン ID、コンテンツ・アイテム名およびファイル・タイプから、ネイティブ・ファイルのファイル・ダウンロードを実行します (DownloadFile と比較)。
- ❖ IntradocBeforeDownload イベントを実行します。イベントは、メソッドの発生前に実行されます。詳細は、4-17 ページの「[IdcClient のイベント](#)」を参照してください。



注意: id は、生成されたコンテンツ・アイテムのリビジョン ID です。コンテンツ・アイテムのバージョン・ラベルではありません。

この関数は、次の情報を返します。

- ❖ dID および dDocName を含むシリアル化された HDA
- ❖ パラメータとして渡されたデータ
- ❖ 指定されたファイルをダウンロードできない場合、FALSE

パラメータ

- ❖ id: 最新のリビジョンに対する一意の識別子。
- ❖ docName: ユーザーが割り当てたコンテンツ・アイテム名。
- ❖ filename: ファイル形式。これは、DOC、PDF、その他のサポートされる形式などのファイル・タイプです。

Drag

```
Sub Drag([nAction])
```

説明

ドラッグ操作を開始、終了または取り消します。

- ❖ Drag メソッドは、標準コントロールの実装と同様に処理されます。
- ❖ 詳細は、Visual Basic の API リファレンスを参照してください。

パラメータ

- ❖ nAction: 実行するアクションを指定します。nAction を省略すると、nAction は 1 に設定されます。

Drag メソッドの設定は次のとおりです。

- ❖ 0: ドラッグ操作を取り消します。コントロールの元の位置に戻します。
- ❖ 1: (デフォルト) コントロールのドラッグを開始します。
- ❖ 2: ドラッグを終了します。つまり、コントロールをドロップします。

EditDocInfoLatestRev

```
Sub EditDocInfoLatestRev(docName As String, curID As String, activateAction As String)
```

説明

最新のリビジョンに関するコンテンツ・アイテム情報を編集します。

- ❖ ODMA 関連です。
- ❖ コンテンツ・アイテム名、バージョン・レベルおよび現在アクティブなリクエストされたアクションから、最新のリビジョンに関するコンテンツ・アイテム情報を編集します。
- ❖ 最新のリビジョンに関するコンテンツ・アイテム情報を編集できない、またはシステムで見つからない場合、FALSE を返します。



注意: curID は、コンテンツ・アイテムのバージョン・ラベルです。生成されたコンテンツ・アイテムのリビジョン ID ではありません。

パラメータ

- ❖ curID: 最新のリビジョンに対する一意の識別子。
- ❖ activateAction: ODMActivate に渡されます。これは、Idoc Script として使用されます。任意です。
- ❖ docName: ユーザーが割り当てたコンテンツ・アイテム名。任意です。

Forward

Sub Forward()

説明

次の HTML ページを表示します。

- ❖ 次画面に移動します。
- ❖ ユーザーに表示する次のHTMLページに関するキャッシュされた情報を取得します。

パラメータ

なし

GoCheckinPage

Sub GoCheckinPage(id As String, docName As String, isNew As Boolean, params As String)

説明

新しいコンテンツ・アイテムまたはコンテンツ・アイテムのリビジョンをチェックインします。

- ❖ コンテンツ・アイテムのリビジョン ID およびコンテンツ・アイテム名から、新しいコンテンツ・アイテムまたはコンテンツ・アイテムのリビジョンをチェックインします。
- ❖ このメソッドは、コンテンツ・アイテムのチェックイン・ページを開き、一意のコンテンツ・アイテム識別子、ユーザーが割り当てたコンテンツ・アイテム名および割り当てられたコンテンツ・アイテムのパラメータを関連するテキスト・フィールドに入力します。また、新しいコンテンツ・アイテムなのか、リビジョンなのかも指定します。



注意: id は、生成されたコンテンツ・アイテムのリビジョン ID です。コンテンツ・アイテムのバージョン・ラベルではありません。

この関数は、次の情報を返します。

- ❖ 指定されたファイルをチェックインできない場合、FALSE
- ❖ dID および dDocName を含むシリアル化された HDA
- ❖ パラメータとして渡されたデータ

パラメータ（すべて任意）

- ❖ id: 最新のリビジョンに対する一意の識別子。
- ❖ docName: ユーザーが割り当てたコンテンツ・アイテム名。
- ❖ IsNew: チェックインされるコンテンツ・アイテムが新しいコンテンツ・アイテムなのか、リビジョンなのかを定義します。
 - TRUE の場合、新しい一意のコンテンツ・アイテムのバージョン・ラベルが割り当てられます。
 - デフォルトは TRUE です。
- ❖ params: 「Check In」 ページを事前入力するパラメータ。

Home

Sub Home ()

説明

定義されたホームページに戻ります。

- ❖ ホーム画面に移動します。
- ❖ HTML ページ・リクエストを実行し、定義されたホームページをユーザーに表示します。

パラメータ

なし

InitiateFileDownload

```
Function InitiateFileDownload(command As String, filename As String) As String
```

説明

ファイル・ダウンロードを開始します。

- ❖ 現在関連付けられているコマンドおよびファイル・タイプから、ファイル・ダウンロードを開始します。このメソッドは、コンテンツ・アイテムの特定のレンディション、最新のリビジョンまたは最新リリースのリビジョンのファイル・ダウンロードを開始します。
- ❖ IntradocServerResponse イベントを実行します。イベントは、メソッドの発生前に実行されます。
- ❖ 詳細は、4-17 ページの「[IdcClient のイベント](#)」を参照してください。

パラメータ

- ❖ **command:** 現在関連付けられているコマンド。
- ❖ **filename:** ファイル形式。これは、PDF、HTM、その他のサポートされる形式などのファイル・タイプです。

出力

- ❖ リクエストされた情報を含むシリアル化された HDA を返します。
- ❖ パラメータとして渡されたデータを返します。

InitiatePostCommand

```
Function InitiatePostCommand(postData As String) As String
```

説明

post コマンドを開始します。

- ❖ サービス・コールを開始します。割り当てられた post データから、post コマンドを開始します。
- ❖ IntradocServerResponse イベントを実行します。イベントは、メソッドの発生前に実行されます。詳細は、4-17 ページの「[IdcClient のイベント](#)」を参照してください。

パラメータ

- ❖ postData: サービス・コマンドおよび必要なサービス・パラメータを含むシリアルライズされた HDA

出力

- ❖ リクエストされた情報を含むシリアルライズされた HDA を返します。
- ❖ StatusCode および StatusMessage を返します。
 - 障害が発生した場合、StatusCode は負数となり、StatusMessage はエラーを示します。
 - 返される HDA に StatusCode パラメータが含まれていない場合、サービス・コールは成功しました。

Move

Sub Move(Left As Single, [Top], [Width], [Height])

説明

オブジェクトを移動します。

- ❖ Move メソッドは、標準コントロールの実装と同様に処理されます。
- ❖ 詳細は、Visual Basic の API リファレンスを参照してください。

パラメータ

- ❖ nLeft: オブジェクトの左端の水平座標を指定します。これは、単精度値です。
- ❖ nTop: オブジェクトの上端の垂直座標を指定します。これは、単精度値です。
- ❖ nWidth: オブジェクトの新しい幅を指定します。これは、単精度値です。
- ❖ nHeight: オブジェクトの新しい高さを指定します。これは、単精度値です。

Navigate

Sub Navigate(url As String)

説明

URL パスを計算します。

- ❖ 指定された完全な URL から、URL をシリアルライズされた HDA から計算し、文字列として値を返します。

この関数は、次の情報を返します。

- ❖ リクエストされた情報を含むシリアルライズされた HDA
- ❖ パラメータとして渡されたデータ

パラメータ

- ❖ url: 完全な URL パス

NavigateCgiPage

Sub NavigateCgiPage(params As String)

説明

CGI パスを計算します。

- ❖ 定義されたコンテンツ・アイテムのパラメータから、CGI パスをシリアルライズされた HDA から計算し、文字列として値を返します。

パラメータ

- ❖ params: 割り当てられたコンテンツ・アイテムのパラメータ

RefreshBrowser

```
Sub RefreshBrowser()
```

説明

ブラウザをリフレッシュします。

- ❖ このメソッドは、Web ブラウザをリフレッシュし、動的情報を更新します。

パラメータ

なし

SendCommand

```
Function SendCommand(params As String) As String
```

説明

サービス・リクエストをコンテンツ・サーバーに発行します。

- ❖ 定義されたコンテンツ・アイテムのパラメータから、コンテンツ・アイテムの処理に関連するコンテンツ・サーバーからサービスを実行します。

パラメータ

- ❖ params: CGI URL エンコードのパラメータ

出力

- ❖ リクエストされた情報を含むシリアル化された HDA を返します。
- ❖ パラメータとして渡されたデータを返します。

SendPostCommand

Function SendPostCommand(postData As String) As String

説明

post コマンドを送信します。

- ❖ サービス・コールを実行します。
- ❖ IntradocBrowserPost イベントを実行します。イベントは、メソッドの発生前に実行されます。詳細は、4-17 ページの「[IdcClient のイベント](#)」を参照してください。

パラメータ

- ❖ postData: サービス・コマンドおよび必要なサービス・パラメータを含むシリアル化された HDA

出力

- ❖ リクエストされた情報を含むシリアル化された HDA を返します。
- ❖ StatusCode および StatusMessage を返します。
 - 障害が発生した場合、StatusCode は負数となり、StatusMessage はエラーを示します。
 - 返される HDA に StatusCode パラメータが含まれていない場合、サービス・コールは成功しました。

SetFocus

Sub SetFocus()

説明

コントロールにフォーカスを割り当てます。

- ❖ SetFocus メソッドは、標準コントロールの実装と同様に処理されます。
- ❖ 詳細は、Visual Basic の API リファレンスを参照してください。

パラメータ

なし

ShowDMS

Sub ShowDMS()

説明

コンテンツ・マネージャと関連付けられている HTML ページを開きます。

- ❖ ODMA 関連です。
- ❖ コンテンツ・マネージャのアクセス・ページをブラウザに表示します。

パラメータ

なし

ShowDocInfoLatestRev

Sub ShowDocInfoLatestRev(docName As String, curID As String, activateAction As String)

説明

最新のリリースに関するコンテンツ・アイテム情報を表示します。

- ❖ コンテンツ・アイテム名、バージョン・ラベルおよび実行するアクションから、最新のリリースに関するコンテンツ・アイテム情報をブラウザ・コントロールに表示します。



注意: curID は、コンテンツ・アイテムのバージョン・ラベルです。生成されたコンテンツ・アイテムのリリース ID ではありません。

パラメータ

- ❖ docName: ユーザーが割り当てたコンテンツ・アイテム名。
- ❖ curID: 最新のリリースに対する一意の識別子。任意です。
- ❖ activateAction: 現在アクティブなリクエストされたアクション。任意です。

ShowWhatsThis

Sub ShowWhatsThis()

説明

WhatsThisHelpID プロパティを持つオブジェクトに指定された「What's This」ヘルプ・トピックを表示します。

- ❖ ShowWhatsThis メソッドは、標準コントロールの実装と同様に処理されます。
- ❖ 詳細は、Visual Basic の API リファレンスを参照してください。

パラメータ

- ❖ Object: 「What's This」ヘルプ・トピックを表示するオブジェクトを指定します。

StartSearch

Sub StartSearch()

説明

問合せページをブラウザ・コントロールに表示します。

- ❖ ブラウザ操作を実行します。

パラメータ

なし

Stop

Sub Stop()

説明

ブラウザを停止します。

- ❖ このメソッドは、ブラウザでの情報のロードを停止または取り消します。

パラメータ

なし

UndoCheckout

Sub UndoCheckout(docName As String, curID As String)

説明

このサービスは、コンテンツ・アイテムのチェックアウトを元に戻します。

- ❖ コンテンツ・アイテム名およびバージョン・ラベルから、コンテンツ・アイテムをシステムで検索し、チェックアウトを元に戻そうとします。コンテンツ・アイテムがシステムに存在しない場合、コンテンツ・アイテムがチェックアウトされていない場合、またはチェックアウトを元に戻すのに十分な権限がユーザーにない場合、サービスは失敗します。
- ❖ IntradocServerResponse イベントを実行します。イベントは、メソッドの発生前に実行されます。
- ❖ 詳細は、4-17 ページの「[IdcClient のイベント](#)」を参照してください。



注意: curID は、コンテンツ・アイテムのバージョン・ラベルです。生成されたコンテンツ・アイテムのリビジョン ID ではありません。

パラメータ

- ❖ curID: 最新のリビジョンに対する一意の識別子。
- ❖ docName: ユーザーが割り当てたコンテンツ・アイテム名。任意です。

ViewDocInfo

Sub ViewDocInfo(id As String)

説明

コンテンツ・アイテム情報ページに移動し、コンテンツ・アイテム情報をブラウザに表示します。

- ❖ ブラウザ操作を実行します。
- ❖ コンテンツ・アイテムのリビジョン ID から、コンテンツ・アイテム情報をブラウザに表示します。



注意: id は、生成されたコンテンツ・アイテムのリビジョン ID です。コンテンツ・アイテムのバージョン・ラベルではありません。

パラメータ

- ❖ id: 最新のリリースに対する一意の識別子

ViewDocInfoLatestRev

```
Sub ViewDocInfoLatestRev(docName As String, curID As String)
```

説明

コンテンツ・アイテム情報ページに移動し、最新のリリースに関するコンテンツ・アイテム情報を表示します。

- ❖ コンテンツ・アイテム名およびバージョン・ラベルから、最新のリリースに関するコンテンツ・アイテム情報を表示します。



注意: curID は、コンテンツ・アイテムのバージョン・ラベルです。生成されたコンテンツ・アイテムのリリース ID ではありません。

この関数は、次の情報を返します。

- ❖ dID および dDocName を含むシリアル化された HDA
- ❖ パラメータとして渡されたデータ

パラメータ

- ❖ docName: ユーザーが割り当てたコンテンツ・アイテム名
- ❖ curID: 最新のリリースに対する一意の識別子

ZOrder

```
Sub ZOrder([Position])
```

説明

指定されたフォームまたはコントロールをグラフィック・レベル内の z-order の前または後に配置します。

- ❖ ZOrder メソッドは、標準コントロールの実装と同様に処理されます。
- ❖ 詳細は、Visual Basic の API リファレンスを参照してください。

パラメータ

- ❖ **nOrder**: 他のオブジェクトを基準としたオブジェクトの位置を示す整数を指定します。nOrder を省略すると、設定は 0（ゼロ）です。

zOrder メソッドの設定は次のとおりです。

- ❖ 0: (デフォルト) オブジェクトは **z-order** の前に配置されます。
- ❖ 1: オブジェクトは **z-order** の後に配置されます。

IdcClient のプロパティ

各データ・アイテムまたは属性は、Visual Basic ではプロパティとして実装されます。プロパティは、Visual Basic 開発環境内ではオブジェクトの **public** インタフェースを通して公開されます。これらの属性を使用すると、要素をさらに詳しく記述できます。

IdcClient OCX プロパティは次のとおりです。

- ❖ [ClientControlledContextValue](#) (4-36 ページ)
- ❖ [HostCgiUrl](#) (4-37 ページ)
- ❖ [Password](#) (4-37 ページ)
- ❖ [UseBrowserLoginPrompt](#) (4-37 ページ)
- ❖ [UseProgressDialog](#) (4-37 ページ)
- ❖ [UserName](#) (4-38 ページ)
- ❖ [WorkingDir](#) (4-38 ページ)

ClientControlledContextValue

ユーザー指定のコンテキスト値を指定します。この値は、コンテンツ・サーバーによって配信される Web ページで、変数 **ClientControlled** として Idoc Script で使用できるようになります。

- ❖ 値を文字列として返します。
- ❖ パラメータをとりません。

HostCgiUrl

ホスト CGI bin の完全な URL パスを指定します。

- ❖ 値を文字列として返します。
- ❖ パラメータをとりません。

Password

割り当てられたユーザー・パスワードを指定します。

- ❖ 値を文字列として返します。
- ❖ パラメータをとりません。

UseBrowserLoginPrompt

ブラウザのログイン・プロンプトを使用できるようにします。ユーザー認証のダイアログ・ボックスを表示するかどうかを定義します。

- ❖ TRUE に設定すると、ユーザー認証のダイアログ・ボックスが開きます。
- ❖ デフォルトは TRUE です。

次のブール値を返します。

- ❖ ログインが成功した場合は、TRUE を返します。
- ❖ ログインが拒否された場合は、FALSE を返します。

UseProgressDialog

ユーザー進行状況ダイアログを使用できるようにします。ユーザー進行状況のダイアログ・ボックスを表示するかどうかを定義します。

- ❖ TRUE に設定すると、ユーザー進行状況のダイアログ・ボックスが開きます。
- ❖ デフォルトは TRUE です。

次のブール値を返します。

- ❖ アクションが成完了した場合は、TRUE を返します。
- ❖ アクションが失敗した場合は、FALSE を返します。

UserName

割り当てられたユーザー名を指定します。

- ❖ 値を文字列として返します。
- ❖ パラメータをとりません。

WorkingDir

作業ディレクトリをフルパスとして指定します。これは、ダウンロードされたファイルが格納される場所です。

- ❖ 値を文字列として返します。
- ❖ パラメータをとりません。



サード・パーティ・ライセンス

概要

この付録では、この製品に含まれるすべてのサード・パーティ製品のサード・パーティ・ライセンスを示します。

- ❖ [Apache Software License](#) (A-2 ページ)
- ❖ [W3C® Software Notice and License](#) (A-2 ページ)
- ❖ [Zlib License](#) (A-4 ページ)
- ❖ [General BSD License](#) (A-5 ページ)
- ❖ [General MIT License](#) (A-5 ページ)
- ❖ [Unicode License](#) (A-6 ページ)
- ❖ [その他の帰属](#) (A-7 ページ)

APACHE SOFTWARE LICENSE

- * Copyright 1999-2004 The Apache Software Foundation.
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

W3C® SOFTWARE NOTICE AND LICENSE

- * Copyright © 1994-2000 World Wide Web Consortium,
- * (Massachusetts Institute of Technology, Institut National de
- * Recherche en Informatique et en Automatique, Keio University).
- * All Rights Reserved. <http://www.w3.org/Consortium/Legal/>
- *
- * This W3C work (including software, documents, or other related items) is
- * being provided by the copyright holders under the following license. By
- * obtaining, using and/or copying this work, you (the licensee) agree that
- * you have read, understood, and will comply with the following terms and
- * conditions:
- *
- * Permission to use, copy, modify, and distribute this software and its
- * documentation, with or without modification, for any purpose and without
- * fee or royalty is hereby granted, provided that you include the following
- * on ALL copies of the software and documentation or portions thereof,
- * including modifications, that you make:
- *
- * 1. The full text of this NOTICE in a location viewable to users of the
- * redistributed or derivative work.

*
* 2. Any pre-existing intellectual property disclaimers, notices, or terms
* and conditions.If none exist, a short notice of the following form
* (hypertext is preferred, text is permitted) should be used within the
* body of any redistributed or derivative code: "Copyright ©
* [\$date-of-software] World Wide Web Consortium, (Massachusetts
* Institute of Technology, Institut National de Recherche en
* Informatique et en Automatique, Keio University).All Rights
* Reserved. <http://www.w3.org/Consortium/Legal/>"
*
* 3. Notice of any changes or modifications to the W3C files, including the
* date changes were made.(We recommend you provide URIs to the location
* from which the code is derived.)
*
* THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS
* MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
* NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR
* PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE
* ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
*
* COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
* CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR
* DOCUMENTATION.
*
* The name and trademarks of copyright holders may NOT be used in advertising
* or publicity pertaining to the software without specific, written prior
* permission.Title to copyright in this software and any associated
* documentation will at all times remain with copyright holders.
*

ZLIB LICENSE

* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.3, July 18th, 2005

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org

Mark Adler madler@alumni.caltech.edu

GENERAL BSD LICENSE

Copyright (c) 1998, Regents of the University of California

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

"Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

"Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

"Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GENERAL MIT LICENSE

Copyright (c) 1998, Regents of the Massachusetts Institute of Technology

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UNICODE LICENSE

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories <http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>, and <http://www.unicode.org/cldr/data/>. Unicode Software includes any source code published in the Unicode Standard or under the directories <http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>, and <http://www.unicode.org/cldr/data/>.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2006 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Unicode and the Unicode logo are trademarks of Unicode, Inc., and may be registered in some jurisdictions. All other trademarks and registered trademarks mentioned herein are the property of their respective owners

その他の帰属

Adobe, Acrobat, and the Acrobat Logo are registered trademarks of Adobe Systems Incorporated.

FAST Instream is a trademark of Fast Search and Transfer ASA.

HP-UX is a registered trademark of Hewlett-Packard Company.

IBM, Informix, and DB2 are registered trademarks of IBM Corporation.

Jaws PDF Library is a registered trademark of Global Graphics Software Ltd.

Kofax is a registered trademark, and Ascent and Ascent Capture are trademarks of Kofax Image Products.

Linux is a registered trademark of Linus Torvalds.

Mac is a registered trademark, and Safari is a trademark of Apple Computer, Inc.

Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation.

MrSID is property of LizardTech, Inc. It is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.

Oracle is a registered trademark of Oracle Corporation.

Portions Copyright © 1994-1997 LEAD Technologies, Inc. All rights reserved.

Portions Copyright © 1990-1998 Handmade Software, Inc. All rights reserved.

Portions Copyright © 1988, 1997 Aladdin Enterprises. All rights reserved.

Portions Copyright © 1997 Soft Horizons. All rights reserved.

Portions Copyright © 1995-1999 LizardTech, Inc. All rights reserved.

Red Hat is a registered trademark of Red Hat, Inc.

Sun is a registered trademark, and Sun ONE, Solaris, iPlanet and Java are trademarks of Sun Microsystems, Inc.

Sybase is a registered trademark of Sybase, Inc.

UNIX is a registered trademark of The Open Group.

Verity is a registered trademark of Autonomy Corporation plc

索引

記号

-c connection_mode
 auto, 2-6
 server, 2-6
 standalone, 2-6

A

AboutBox, 4-20
Active Server Page
 IdcCommandUX のコール, 3-4
Active Server Page からの IdcCommandUX のコール,
 3-4
ActiveX コマンド・ユーティリティ, 「メソッド」を参
 照, 3-16
ASP
 IdcCommandUX のコール, 3-4

B

Back, 4-20

C

CancelRequest, 4-21
ClientControlledContextValue, 4-36
closeServerConnection(), 3-17
computeNativeFilePath(Data As String) as string, 3-18
computeURL(Data As String, IsAbsolute As Boolean) as
 string, 3-19
computeWebFilePath(Data As String) as string, 3-22
connectToServer() as boolean, 3-23
 戻り値, 3-23

D

DoCheckoutLatestRev, 4-21
DownloadFile, 4-22
DownloadNativeFile, 4-23
Drag, 4-24

E

EditDocInfoLatestRev, 4-24
EOD, 2-4
executeCommand (Data As String), 3-24
 パラメータ
 Data, 3-24
 戻り値, 3-24

F

Forward, 4-25

G

GoCheckinPage, 4-25

H

HostCgiUrl, 4-37

I

IdcClient ActiveX Control, 4-5
IdcClient OCX コンポーネント, 4-1
IdcClient のイベント, 4-17
IdcClient のプロパティ, 4-36
IdcClient のメソッド, 4-19

IdcCommand

- リポジトリ・サーバーのコマンド・ユーティリティ , 2-7
 - コマンド・ファイルの構文 , 2-2
 - コマンドライン・オプション , 2-2
 - ランチャの使用 , 2-17
 - リモートでのサービスのコール , 2-7
- IdcCommandUX - リポジトリ・サーバーの ActiveX コマンド・ユーティリティ
 - メソッド , 3-16
- IdcCommandUX と ASP, 3-4
- IdcService= (コマンド・ファイルの構文のタグ) , 2-4
- init as boolean (*IntrdocDir As String*)
 - パラメータ
 - UserName, 3-27
- init as boolean (*StellentDir As String*)
 - パラメータ
 - StellentDir, 3-27
- InitiateFileDownload, 4-27
- InitiatePostCommand, 4-27
- IntrdocBeforeDownload, 4-17
- IntrdocBrowserPost, 4-17
- IntrdocBrowserStateChange, 4-18
- IntrdocDir As String (*init as boolean*), 3-27
- IntrdocRequestProgress, 4-18
- IntrdocServerResponse, 4-18

M

- Microsoft Visual Basic, 4-5
- Move, 4-28

N

- Navigate, 4-29
- NavigateCgiPage, 4-29

O

- OCX コンポーネント , 4-1

P

- Password, 4-37

R

- RefreshBrowser, 4-30

S

- SendCommand, 4-30
- SendPostCommand, 4-31
- SetFocus, 4-31
- ShowDocInfoLatestRev, 4-32
- ShowWhatsThis, 4-33
- StartSearch, 4-33
- Stop, 4-33

U

- UndoCheckout, 4-34
- UseBrowserLoginPrompt, 4-37
- UseProgressDialog, 4-37
- UserName, 4-38

V

- ViewDocInfo, 4-34
- ViewDocInfoLatestRev, 4-35
- Visual Basic, 4-5

W

- WorkingDir, 4-38

Z

- ZOrder, 4-35

い

イベント

- IntrdocBeforeDownload, 4-17
- IntrdocBrowserPost, 4-17
- IntrdocBrowserStateChange, 4-18
- IntrdocRequestProgress, 4-18
- IntrdocServerResponse, 4-18

か

概要

- 対象読者 , 1-3
- 表記規則 , 1-3

こ

コマンド・ファイルの構文, 2-2
 特殊タグ
 IdcService=, 2-4
 特殊文字
 \, 2-4
 #, 2-4
 EOD, 2-4
 優先順位, 2-4
 コマンドライン・オプション, 2-2

さ

作成および実行, IdcCommand パラメータ, 2-7

そ

属性

ClientControlledContextValue, 4-36
 HostCgiUrl, 4-37
 Password, 4-37
 UseBrowserLoginPrompt, 4-37
 UseProgressDialog, 4-37
 UserName, 4-38
 WorkingDir, 4-38

め

メソッド

AboutBox, 4-20
 Back, 4-20
 CancelRequest, 4-21
 DoCheckoutLatestRev, 4-21
 DownloadFile, 4-22
 DownloadNativeFile, 4-23
 Drag, 4-24
 EditDocInfoLatestRev, 4-24
 Forward, 4-25
 GoCheckinPage, 4-25

InitiateFileDownload, 4-27
 InitiatePostCommand, 4-27
 Move, 4-28
 Navigate, 4-29
 NavigateCgiPage, 4-29
 RefreshBrowser, 4-30
 SendCommand, 4-30
 SendPostCommand, 4-31
 SetFocus, 4-31
 ShowDocInfoLatestRev, 4-32
 ShowWhatsThis, 4-33
 StartSearch, 4-33
 Stop, 4-33
 UndoCheckout, 4-34
 ViewDocInfo, 4-34
 ViewDocInfoLatestRev, 4-35
 ZOrder, 4-35

メソッド (ActiveX コマンド・ユーティリティ), 3-16
 closeServerConnection(), 3-17
 computeNativeFilePath(Data As String) as string,
 3-18
 computeURL(Data As String, IsAbsolute As Boolean)
 as string, 3-19
 computeWebFilePath(Data As String) as string, 3-22
 connectToServer() as boolean, 3-23
 executeCommand (Data As String), 3-24
 init as boolean (IntradocDir As String), 3-27

ゆ

優先順位, 2-4

ら

ランチャの使用, 2-17

り

リモートでのサービスのコール, 2-7

