デジタルサーチャー講座

検索を科学する

塩田 紳二

第3回 ウィンドウズの検索機能

今回は、ウィンドウズXPに標準で組み込まれている検索機能「インデックスサービス」について解説する。これは、エクスプローラから検索を選んだときにも利用される機能で、あらかじめファイルに関してインデックスを作成しておくというもの。機能的にはGoogle Desktop Searchと同じだが、ファイルやディレクトリーに対してインデックス化の可不可の設定ができるなど、OS組み込みならではメリットもある。

インデックスサービスとは？

ウィンドウズXPに組み込まれている「インデックスサービス」は、もともとは、インデックスサーバーという名称のソフトウェアで、ウィンドウズNT用のIIS（マイクロソフトのウィンドウズ用ウェブサーバーソフト）から全文検索を行うためのサーバー用ソフトウェアとして1996年に登場した。その後、ウィンドウズNT 4.0のオプションパックにインデックスサーバー2.0が搭載され、ウィンドウズ2000からは、インデックスサービスとなってOS標準のモジュールとなった。ただし、バージョン番号は、以前のものを引き継ぎVer.3.0のままだった。

ウィンドウズXPに搭載されているインデックスサービスも同じくVer.3.0だが、ウィンドウズ2000のものに対して、オーディオやビデオなどのメディアファイルが持つプロパティーにも対応するようになっている。

インデックスサービスは、ファイルを対象とした全文検索型の検索エンジンであり、IFilterと呼ばれるコンポーネントを使うことで、さまざまなアプリケーションのファイル形式に対応できる。

標準では、テキストファイル、Officeドキュメント、HTML、MIME（電子メールのメッセージ形式）に対応しており、一部のアプリケーションは、自身のインストール時にIFilterを組み込むものがあるほか、ウェブサイトなどでIFilterを配布しているところもある。

たとえば、アドビは、PDF用のIFilterの配布を行っている※1。これを組み込むことでインデックスサービスを使ってPDFを対象とした検索を行うことが可能になる。ただし、現在配布されているのは、2004年にAcrobat 6.0用に作られたIFilter v6.0で、対応するPDFのバージョンは1.5までのため、最新のAcrobat 7.0で作成されるPDF 1.6には対応していない。

このインデックスサービスは、エクスプローラの検索機能が利用しているほか、ウィンドウズを使ったウェブサーバーで

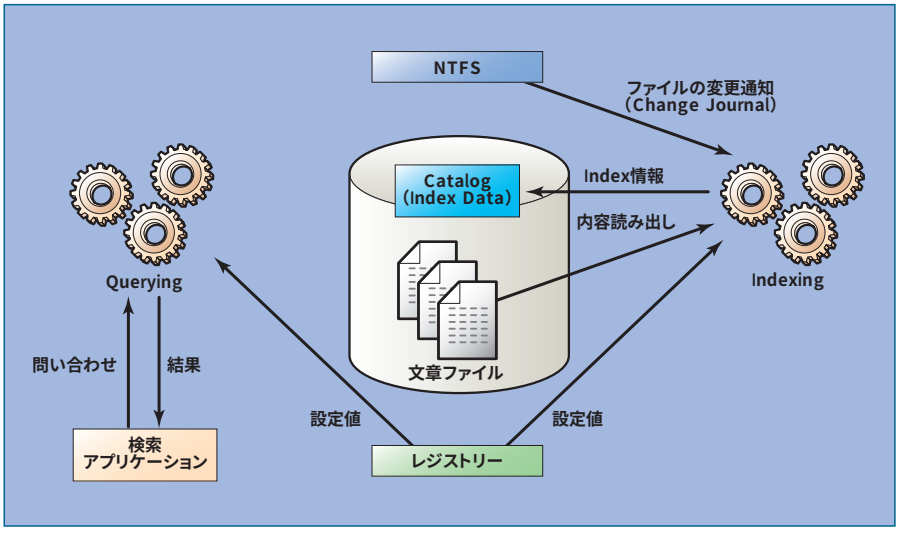

図1 インデックスサービスは、インデックス作成を行うIndexing部と、検索を行うQuery部から構成されている。Indexing部でCatalogを作成し、Query部ではこれを使って検索を行う。

は、標準的な全文検索システムとして利用されている。検索ページの拡張子がaspやaspxになっているサイトは大概、インデックスサーバーを利用している。

※1：
http://www.adobe.com/support/downloads/thankyou.jsp?ftpID=2611&fileID=2457

インデックスサービスの構成

インデックスサービスは、大きく、文書をスキャンし、インデックスファイル（カタログ）を作成するIndexing部と、外部からの問い合わせに応じてカタログを検索するQuerying部に分かれている（図1）。インデックスサービスとはいうが、もともとインデックスサーバーといっていたように単純なサービス（ウィンドウズでGUIを持たずにバックグラウンドで動作するプログラム。Unix系でいうデーモン）ではなく、さまざまなモジュールの集合体である。

前者をインデックスサービスの実体であるcisvc.exeが行い、後者は、いくつかのDLLがAPIを実装し、サーチエンジンを制御する。

Indexing部は、NTFS上では、NTFSが持つジャーナル機能を使って、ファイルの作成や削除、変更を検出する（ただし初めてIndexing（インデックス化）が行われる場合には全てのファイルのスキャンを行う）。FAT上などでは、定期的なスキャンによりこうしたファイルを検出する（図2）。

対象となるファイルは、デフォルトまたは組み込まれたIFilterを使い、テキストなどが抽出される（図3）。インデックスサービスでは、単にファイル内のテキストだけでなく、ファイルの更新日時やタイトルなどのメタデータもインデックス化の対象となる。

このフィルターメカニズムが生成するのは、Word Listと呼ばれる単語のリストだ。これは、IFilterが抽出したテキストなどを、ロケール（地域／言語設定）に応じて単語に分解し、さらにノイズワードを排除する。「ノイズワード」とは、数が多く、検索対象を絞ることができない単語のことだ。英語なら冠詞や前置詞などで、日本語ならば「が」や「を」といった助詞などが相当する。単語への分解やノイズワードの定義は、ロケール単位で行われている。

こうしてできたWord Listはカタログに登録され、検索時に検索語から対象ファイルを見つけるために利用される。なお、カタログは標準では各ボリュームにある「\System Volume Information」内にあり、通常はファイルとして見ることはできない。

このWord Listのままでは、データサイズが大きすぎてメモリーなどに置くことができないため、これを圧縮して記録するインデックスを作成する。これには、ShadowとMasterの2つがあり、Word Listは、一定量を超えた段階でいったんShadow Indexにマージされる。その後、1日に1回（レジストリーの設定による）、Shadow IndexとMaster Indexがマージされることになる（図4）。

2段階でマージを行うのは、マージ作業が比較的負荷の高い作業であることと、作業中も検索を可能にするためと思われる。Shadow Indexの段階よりもMaster Indexのほうが、圧縮率が高く、よりコンパクトな状態となっている。

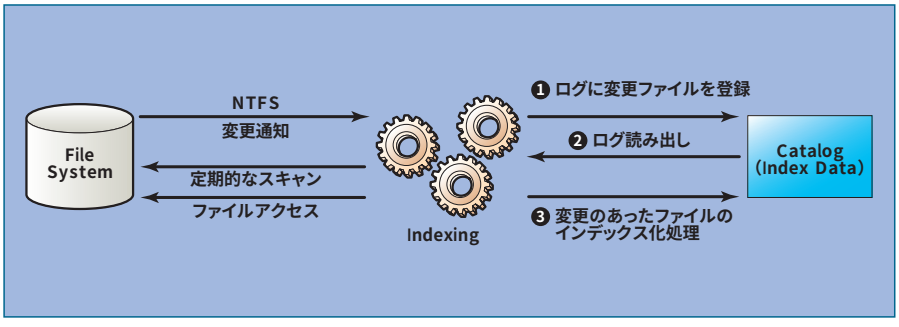

図2　Indexing部は、NTFSからのファイル変更の通知または、ファイルシステムの定期的なスキャンにより変更、追加されたファイルやディレクトリーを検出。これをいったんlogとしてCatalogに登録する。その後、logを元にインデックスの更新を行う。

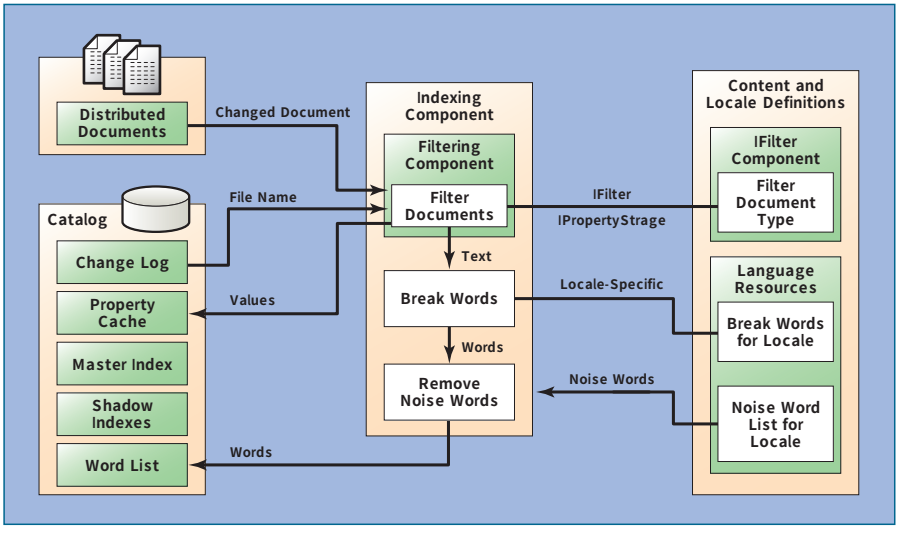

図3　Indexing部は、変更、追加されたファイルをIFilterを使ってテキストに分解する。その後、これを単語に分解し、Noise Wordを取り除いた後、Word ListとしてCatalogに登録する。

IFilter

このIndexingの中核となるのは、IFilterと呼ばれるDLLモジュールである。これは、ファイルからテキストなどを取り出す際に、ファイルタイプに応じて呼びだされる。

インターネットを検索するとさまざまなIFilterが公開されている。とりあえずGoogleなどで「IFilter」と検索してみるといいだろう。MP3やJPEGといったファイル用のIFilterが見つかるはずだ。

IFilterに関しては、「Desktop Search IFilters」※2などに情報がある。

IFilterは、ファイルをチャンク単位に分割する。各チャンクは、テキストかプロパティのどちらか（あるいはIFilter固有データ）になる。プロパティーとは、たとえば、文書名などのメタデータであり、HTMLタグのアトリビュートなどが対応する。

このチャンクに対して、さらにテキストや値（プロパティーのとき）の取得を行わせる。

チャンクからテキストが取り出されるとその後の処理は、インデックスサービス側で行われる。インデックスサービスでは、言語情報（これもプロパティーとしてIFilter経由で取得する）を利用して、テキストをWordに分割する。インデックスサービスでは、これをWord Breakerとよんでいる。

つまり、IFilterは最低限、ファイルからテキスト分解してテキストを抽出できれば、あとの処理は、インデックスサービス側で行ってくれるわけだ。

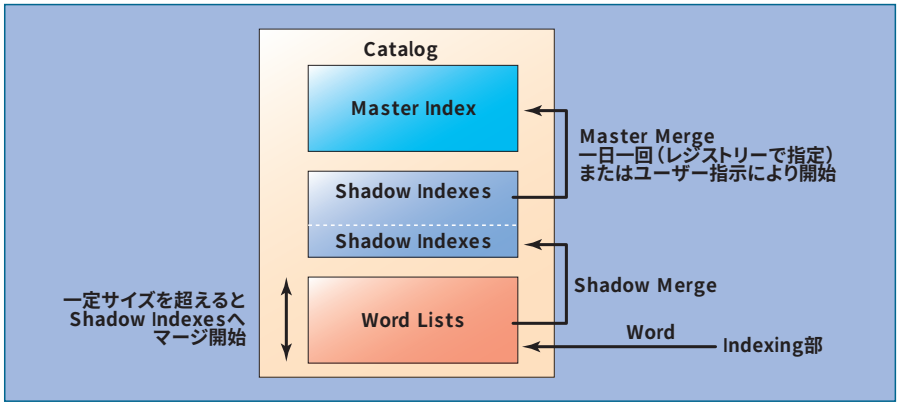

図4　Word Listは、一定サイズを超えた段階で圧縮処理が行われ、Shadow IndexとしてCatalogに登録される。その後、1日に1回といった頻度（レジストリーで指定）で、Shadow Indexは、より圧縮度の高いMaster Indexにマージされる。Word List、Shadow Indexes、Master Indexともに検索時にはインデックス情報として利用される。このような構成になっているのは圧縮処理に時間がかかり、その間も検索を実行できるようにするためである。

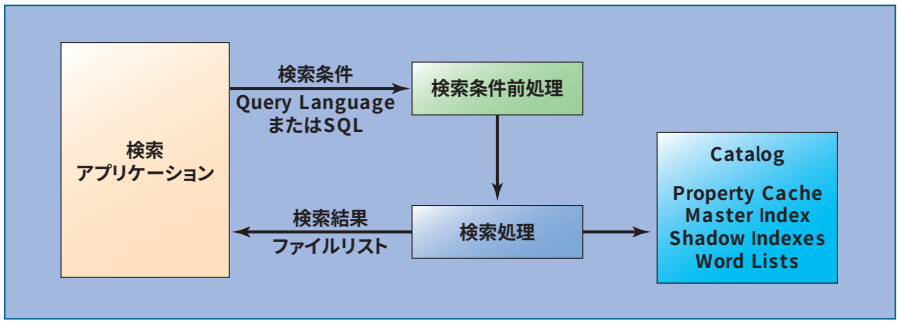

図5　Query部は、外部アプリケーションからAPI経由で検索条件を受け付ける。その後、その検索条件を前処理（オプティマイズ）してから検索を行う。

各文書が持つプロパティについては、カタログ内にキャッシュとしてプロパティ値を保存する。このようにすることで、文書にアクセスせずとも、プロパティ値を得ることができる。

また、このIFilterは、インデックスサーバーだけでなく、SharePoint Portal Server、Exchange、SQL Serverなども利用する。

ある意味、ウィンドウズで、固有のデータ形式を持つファイルを検索する場合の標準的なフィルターといってもいい。

※2：
http://channel9.msdn.com/wiki/default.aspx/Channel9.DesktopSearchIFilters

検索インターフェイス

インデックスサービスは、SQLまたは専用の問い合わせ言語で検索条件を指定する。検索条件を受け取ったら、これを前処理し、その後、Catalogを使って対象となるファイルを見つける。すでにインデックス化が行われているため、検索処理はこのインデックス内から条件を満たすエントリーを探し、対応するファイル（のパス）を得るという処理になる（図5）。

ファイルの更新日時などのプロパティー情報は、事前にCatalogにPropertyキャッシュとして登録されているため、検索ごとにファイルにアクセスする必要はない。

インデックスサービスは、アプリケーション側のインターフェイスとして以下のものを持っている。

- **ISAPI（IISからの呼び出し）**
- **OLE DB Helper/Provider（OLE DB）**
- **ActiveX Data Object（埋め込みオブジェクト）**
- **Query/Admin Helper（OLEオートメション）**

インデックスサービスは、インデックス

サーバーとして登場したときに、IISと組み合わせて使うため、ISAPI（Internet Server Applications Programming Interface）と、C++、Visual Basicなどの言語からの呼び出し、HTML内のスクリプトからの呼び出し（埋め込みオブジェクト）などに対応していた。その後、アプリケーションからデータベースとしてアクセスできるようにしたり、インデックスサービス自体の制御を行うインターフェイス（OLE DB Helper、Admin Helper）を持つことになった。

これらのAPIでは、インデックスサーバー専用の問い合わせ言語（Indexing Service Query Language）もしくはSQLを利用する。これはAPIにより検索条件を指定する言語が違っているということである。SQLだけに統一できないのは、HTML経由などの問い合わせの際に、検索条件を指定し、それを直接インデックスサービスが解釈するからである。

ウェブサイトなどで、検索機能にANDやORのついた検索があるが、このような条件検索を直接実行できるようにしたのがインデックスだった。このため、簡易な検索言語が定義されており、これを入力としてASPファイル内で、検索対象などを指定して簡単にサイト内の全文検索ができるようになっていた。このときの検索言語が一部のインターフェイスで利用できる。

標準状態だとIFilterがほとんどインストールされていない状態なので、実際のところ、インデックスサービスにはあまり利点がない。しかし、利用頻度の高いファイルを対象とするIFilterを入手できれば、検索はエクスプローラから直接行えるため、場合によっては、本連載の第1回で解説したGoogle Desktop Search（以下、GDS）よりも便利なことがある。一番の違いは、ファイルの日付や拡張子、ディレクトリーなどを指定して検索が可能なことだ。こうした条件設定により、キーワード自体では絞りきれない検索であっても、候補をかなり絞ることが可能だ。

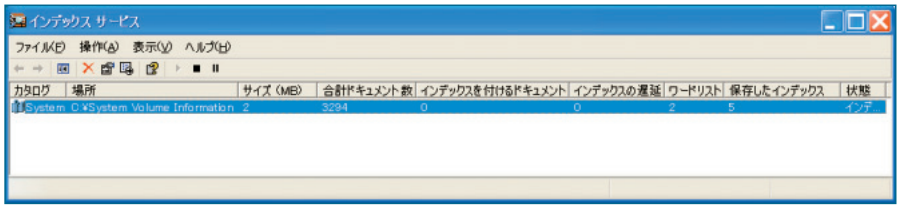

図6　ウィンドウズXPにあるインデックスサービスの管理ツール（管理コンソール）。ここからインデックスサービスの起動やMaster Indexへのマージなどを指示できる。

ウィンドウズ内での検索

このインデックスサービスは、エクスプローラの「検索」機能、標準では、犬が出てくるあの検索機能で利用されている。ただし、そのためには、インデックスサービスを有効にし、対象となるファイルに対してインデックス化を許可するフラグを設定しておく必要がある。

この検索機能は、インデックスサービスがオフであっても、その場で対象ファイルの検索を行うようになっており、インデックスサービスを止めていても時間はかかるものの検索機能は利用できる。

もう1つあるユーザーインターフェイスは、管理コンソール（図6）で、これは次のプログラムを実行する。

```
C:\WINDOWS\system32\ciadv.msc
```

ここでは、インデックスサービスの起動などに加え、カタログのクリアや前述のMaster Mergeなどを強制的に行わせることが可能だ。

現時点では、ユーザーが利用できるインデックスサービスのユーザーインターフェイスはこの程度だが、IIS上でASP（Active Server Pages）を使って検索フォームを作成することはできる。

次世代ウィンドウズでも利用

マイクロソフトが発表したデスクトップサーチや、次世代ウィンドウズ（コード名「Longhorn」）に搭載される検索機能は、このインデックスサービスをベースにしたものになるという。

標準状態では、GDSほど対象ファイルが多くないのが欠点だが、英語版が提供されている「MSN Search Toolber with Windows Desktop Search」では、200種類以上のファイル形式に対応している。年内には、日本語にも対応するとのことなので、これが登場すれば、もう少し使いやすいものになるだろう。

前述したが、インデックスサービスでは、メタデータ（プロパティー）を取り扱うことができ、検索パスを制限することも可能である。このため、ローカルディスク上のすべてを検索結果としてしまうGDSよりも使いやすい場合がある。

たとえば、自身で作成した文書を特定のフォルダー以下に入れていて、その他のディレクトリーに検索の対象にもするが、インターネットなどからダウンロードしたPDFファイルが大量にある場合など、GDSでは、検索結果が大量すぎて、なかなか目的のファイルが見つけにくいことがある。（GDSのプラグインソフトGoogle Desktop Extremeなどである程度は対応可能。http://desktop.google.com/plugins.html参照）

ドキュメントなども整備されていて、アプリケーション側でファイル検索機能を実装しなくとも、インデックスサービスを呼び出すだけで検索機能を付加できる。

こうしたメリットがありながら、いままであまり活用されてこなかったのは、IFilterの提供など、マイクロソフト自身があまり手間を掛けてなかったのが理由だろう。

［インターネットマガジン　バックナンバーアーカイブ］ ご利用上の注意

このPDFファイルは、株式会社インプレスR&D（株式会社インプレスから分割）が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン　バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

http://i.impressRD.jp/bn

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容（技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど）は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者（執筆者、写真の撮影者、イラストの作成者、編集部など）が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元（株式会社インプレスR&D）、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D（当時は株式会社インプレス）と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接的および間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先
株式会社インプレスR&D
All-in-One INTERNET magazine 編集部
im-info@impress.co.jp