

XMLデータベースの
検索速度を最速にする

第2回

データ検索表示の小技

PROJECT KySS

<http://www.projectkyss.net/>

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
 - .NET Framework 1.1
 - Internet Explorer 6.0
 - XML

Level



Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥NETARCHO2ディレクトリに収録しています。

¥POSTALGUIDE_ALLOWPAGING
サンプル1

¥POSTALGUIDE_DATALISTPAGING
サンプル2

¥POSTALGUIDE_CONNECT
サンプル3

¥POSTALGUIDE_STRBUILDER
サンプル4

¥STRBUILDER_HOWTO
記事末コラムのサンプル

¥POSTALAREACLASS
都道府県表示用クラス(サンプル1~4で共通利用)

¥DATA
サンプルで利用するXMLファイル

*) サンプル動作環境における注意

本稿のサンプルは、巨大なファイルサイズのXML文書を扱っているため、マシンのスペックによっては動作が不安定になる場合があります。

サンプルデータは、CSV形式のフリーデータをもとに、XSL変換によって作成しています。市町村合併などにより、現行の郵便番号とはデータが異なる場合があります。

（ 今回のテストの 目的 ）

今回は、構造の異なる3種類のXMLファイル(10MB~18MBの郵便番号データ)を用意し、XMLDocumentクラスや、XPathDocumentクラスとXPathNavigatorクラスを使ったサンプルを作成して、「XMLファイルの読み込み」「検索」「ページングに要するタイム」を競いました。

今回は、以下の2つの視点から、さらなるタイムの短縮を狙います。

- ・ページング方法を工夫することによるタイムの短縮
- ・検索した都道府県名や市郡町村名をひとつながりの「住所」として表示する場合に、文字列の連結方法を工夫することによるタイムの短縮

今回使用するXMLファイルは、約12万件の郵便データを記録した、フラットな構造です(リスト1)。ファイルサイズは約18MB、要素ノードの総数は599,436個です。

（ ページングのタイム を短縮する ）

リスト1の約12万件のデータを、ページングの都度、読み込むのでは時間がかかります。そこで、「都道府県」の指定を必須条件として、表示させるデータを絞り込んでみます。初めてプログラムを実行したときは全データを読み込むので多少の時間を要しますが、ページングの際には、都道府県で絞り込んだデータを読み込むため、タイムを短縮できそうです。

「都道府県」の指定を必須とするには、条件が指定されたことを検証する必要があります。そこで、ビジュアルコード(.aspxファイル)の中で、検証コントロールを使います。

図1のように、DropDownListにデータの入力を検証するRequiredFieldValidatorコントロールを指定し、この項目の選択を必須としておきます。コントロールのプロパティで、ErrorMessage属性の値に、コントロールにエラーが発生したときに表示させるメッセージを指定します。既定値は「Required

リスト1：フラットな階層のXML文書 (postalGuide_basic.xml)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<新郵便番号>
<データ>
<都道府県>神奈川県</都道府県>
<市郡町村>横浜市西区</市郡町村>
<町域>みなとみらいクイーンズタワーA (19階)</町域>
<郵便番号>2206019</郵便番号>
</データ>
(略：以下<データ>繰り返し)
</新郵便番号>
```

FieldValidator」ですが、ここでは、注意を促す文字として「※」を指定しています。

ControlToValidate属性には、検証対象となるコントロールのIDを指定します。この属性は必須で、値を指定しなければ実行時にエラーが発生します。

なお、サンプルでは、DropDownList1で選択された項目を、一度TextBoxコントロールに代入してから、その値を検証しています。そこで、図1のように、RequiredFieldValidatorのControlToValidate属性では、DropDownListコントロールのIDであるDropDownList1ではなく、TextBoxコントロールのIDである「Textbox3」を選択しています。

DataGridを使う SAMPLE 1

postalGuide_AllowPaging¥postalGuide_AllowPaging.aspx

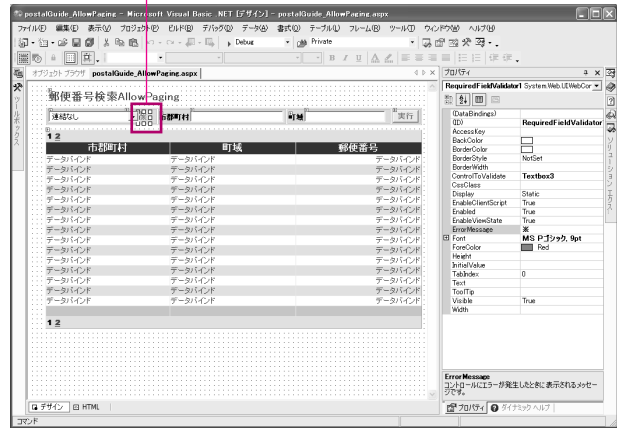
まず、DataGridコントロールで通常のページング方法を試してみます。前回紹介した、DOMをサポートするXml Documentクラスのメソッドやプロパティを使います。

「都道府県」に「北海道」を指定し、「市郡町村」「町域」は未入力のままで「実行」ボタンをクリックすると、図2のような表示結果になります。ページ番号を表示させるには、DataGridコントロールのAllowPaging属性に「True」を指定してから、PagerStyle属性のMode属性に「NumericPages」を指定します。Mode属性の既定値は「NextPrev」なので、そのままでは“<”と“>”の記号で表示されてしまいます。

■ DataShow プロシージャ

それでは検索表示処理の主要部分のコードを見てゆきましょう。

図1：フォームデザイナーで、検証コントロールを設定する RequiredFieldValidator コントロールを配置



* ControlToValidate属性に「Textbox3」を指定している。また、このサンプルのように、一度TextBoxコントロールに代入する方法をとる場合は、TextBoxコントロールのVisibleプロパティをFalseに設定してはいけない。TextBoxコントロールを非表示にするのではなく、表示されている状態で、DropDownListコントロールの背後に隠れるように配置する。

図2：「北海道」の全データを表示させ、8ページ目に移動している



各検索キーに合致するノードを取得するDataShowプロシージャの処理では、ノードを取得するためにSelectNodesメソッドとXPathのcontains関数を使用しています。selectPref、selectCity、selectAreaの3つの変数には、ユーザーが指定した、「都道府県」「市郡町村」「町域」が格納されています。次のコードでは、各検索キーに合致する「市郡町村」のノードリストが返されます^[注1]。

注1) 「contains(String1,String2)」で、String1の中にString2が含まれている場合にTrueが返されます。