

.NET
Architecture
Forum.NET Compact
Frameworkを使った
モバイルアプリケーション
構築法

第4回

Webアプリケーションとの相互利用

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 Framework1.1
.NET Compact Framework1.0
(SP1またはSP2)
Pocket PCを使用したPDA

Level



Samples

・本稿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¥NETARCH01ディレクトリに収録しています。

¥POCKETPC

今回解説するMakingRainmanCardプロジェクト(ASP.NET Webアプリケーションと画像)と、生成されたXMLを利用するモバイルアプリケーション(CardSelectプロジェクト)

(アプリケーションを 作成する前に)

PDAの進化はめざましく、2~3年前のパソコンに匹敵するスペックになっています。これからの時代は、ハードウェアのスペックよりもむしろ、目的や使用場所によって、パソコンとPDAを使いわけようになりそうです。

たとえば、この連載で解説しているアプリケーションでいえば、

①パソコンでWebアプリケーションにアクセスして、提供されている画像を得たり、画像情報の組み込みや設定を行なう。画像情報はXMLデータとして記録しておき、設定値もXML形式の設定ファイルとして保存する。

*) 今回のアプリケーションの動作確認に使ったPocket PCは、「HP iPAQ Pocket PC」です。

*) 今回使用しているシンボル画像は、大森直也氏によるフリー画像です。

*) 前回から取り上げているプログラムは、筆者が愛媛大学総合情報メディアセンター村田健史氏助教授率いるプロジェクトにて開発中の福祉ツール「RAINMAN Toolkit (フリーウェア)」を本誌用にリメイクしたものです。同ツールは3月号の付録CD-ROMに収録しています。また、http://sp.cs.ehime-u.ac.jp/rainman/rainman_top.htmでも配布しています。

*) Windows Mobile 2003 for Pocket (PocketPC 2003) 最新版のベースはWindows CE.NET 4.2 OSです。

②モバイルアプリケーションのXMLファイルを、パソコン側で作成/保存したXMLファイルで置き換える。

という使い分けが考えられます。

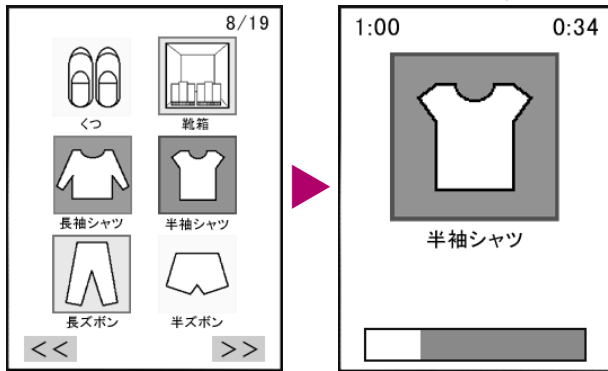
今回は、第2回目で少し触れたように、モバイルアプリケーションで使うXML形式の設定ファイルを、ASP.NETによるWebアプリケーションで作成してみます。

(画面遷移と動作)

図1に表示している画像のファイル名やタイトルの情報は、リスト1のようなXMLファイルに記録されています。このXMLデータを読み込んで画像を表示させているため、XMLデータを変更すれば、もちろん表示も変わります。

今回は、リスト1のXMLファイルを生成するASP.NET Webアプリケーションを作成します。その際、XMLファイルを単純に生成するだけでなく、指定したノードを削除する機能や、同一階層の要素をソートする機能も付けてみ

図1：ページングしつつ画像を選択（左）タイマーの表現(右)

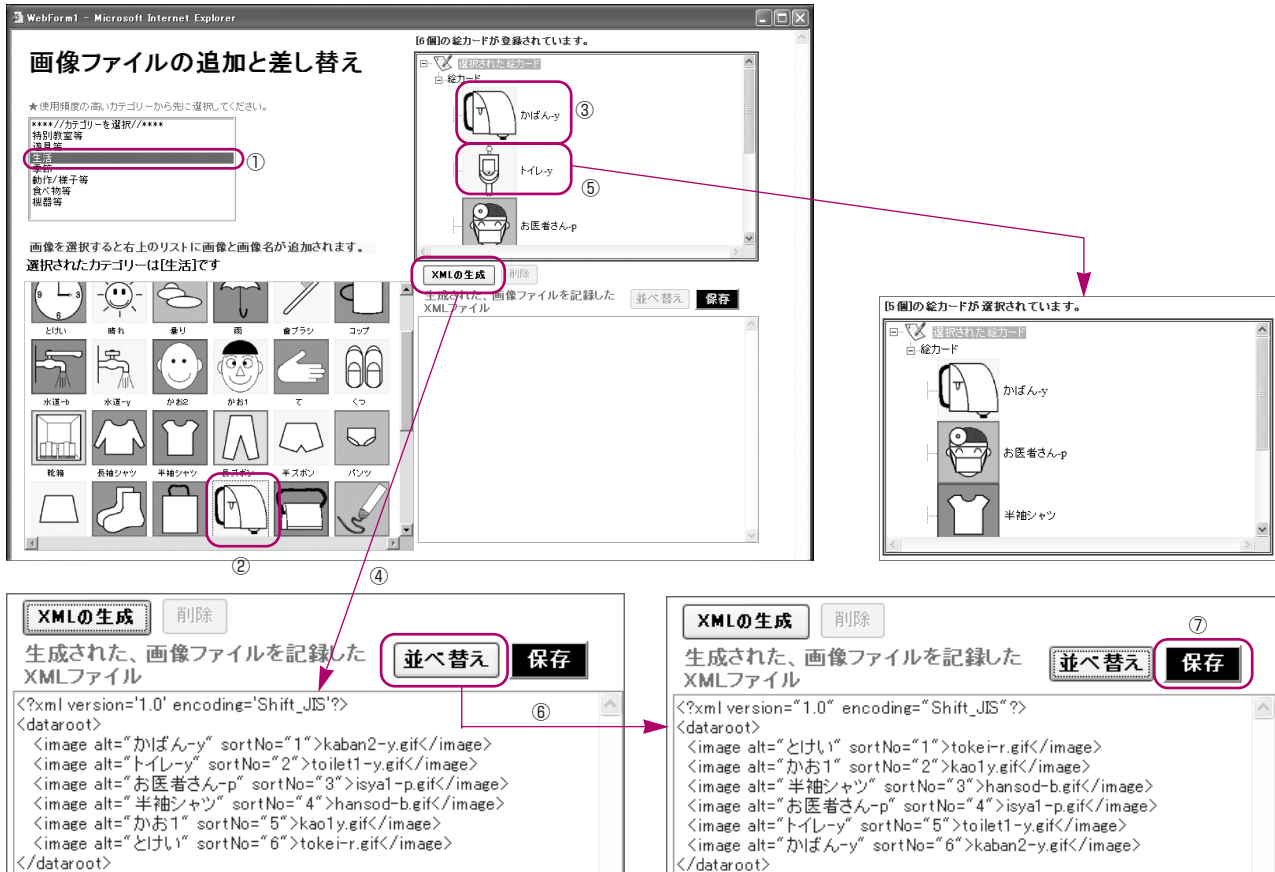


ます（図2）。

それぞれの流れを説明しましょう。

- ① Webアプリケーションにアクセスし、「カテゴリー」を選択すると、下のボックスにカテゴリーに対応する画像が表示されます。

図2：今回作成するWebアプリケーションの画面遷移



リスト1：生成するXMLファイル（xmldata/rainmanCard.xml）

```
<?xml version="1.0" encoding="Shift_JIS"?>
<dataroot>
<image alt="音楽室" sortNo="1">ongaku-r.gif</image>
<image alt="調理室" sortNo="2">chouri-g.gif</image>
<image alt="理科室" sortNo="3">rika-b.gif</image>
～以下<image></image>要素繰り返し～
</dataroot>
```