

## **Collapsible Lists with Web Area**

---

By Joe Resuello, Technical Marketing Engineer, 4D Inc.

Technical Note 09-01

## 要約

---

4D v11 SQL の Web エリアは HTML、JavaScript、CSS の世界への興味深い可能性の扉を開きました。4D HTML タグによるダイナミックな Web テクノロジと統合することで、通常の 4D フォームではできないことが Web エリアでは可能です。このテクニカルノートは Web エリアを使用して折りたたみ可能なリストを実装する方法について説明します。

## はじめに

---

"折りたたみ可能なリスト"と聞けば、皆さんは階層リストを思い浮かべることでしょう。以下は 4D の階層リストの例です:



構造を見てみましょう。このリストはシンプルです。親ノードは"Personal"、"Recreation"、"Work"などの値を持っています。またこれらのノードは子ノードも持っています。親ノード子ノードにかかわらず、一行ごとに一つの値しかないことに留意してください。これは 4D 階層リストの使用です。

Web エリアとダイナミック Web テクノロジを使用すれば、一行に複数の値を持つ親ノードや子ノードを作成できます。HTML レイアウトでは独自のフォーマットやフィールド値の配置が可能です。4D HTML タグを使用すればデータベースのデータを Web エリアに埋め込み、ユーザーにシームレスなユーザーエクスペリエンスを提供できます。Web エリアを 4D フォームの拡張として使用できます。

The screenshot shows a web application interface. At the top, there are navigation buttons: "Previous", "Next", "Delete", "Cancel", and "Accept". Below these is a section titled "Contact" with a "2 of 2" indicator. The contact information is as follows:

Name	Sandra Loh	2	
Address	903 Crispy Lane		
City	San Francisco	Email	Slosh@gmail.net

Below the contact information is a section titled "Appointments for Contact" with a sub-header "Click on a category below to show that type of Appointment." The appointments are listed in a table with a green background for the category headers:

Category	Appointment	Date
Personal	Early Christmas Shopping	11/19/08
	Buy Plasma TV	12/10/08
	Secret Santa drawing	11/13/08
	Watch son's baseball game	12/09/08
Recreation	Trip to Canada	12/13/08
Work	Interview with recruiter	11/2/08

これは添付データベースの Contacts テーブルの入力フォームです。緑の背景のエリアが折りたたみ可能な Web エリアです。"Personal"、"Recreation"、"Work"などのヘッダをクリックすると、そのタイプの Appointments が隠されます。子ノードである Appointments には複数のフィールド値が表示されていることに留意してください。また通常の階層リストに比べ、表示の自由度が高い点にも留意してください。このテクニカルノートでは、Web エリアでこのような機能を作成していきます。

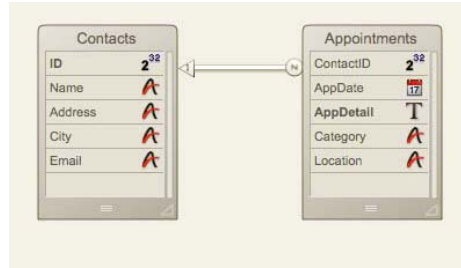
## 概要

Web エリアを作成し、ダイナミックに 4D データをロードするには、以下のステップを行います:

- 4D フォームを作成し Web エリアを配置する
- フォームを Composite モードに設定する
- ローカルに HTML ファイルを作成しロードする
- 入力用の HTML ファイルを定義する

## 4D フォームを作成し Web エリアを配置する

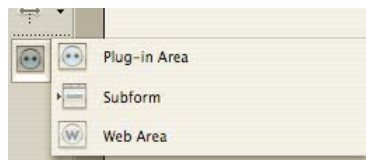
まず最初にやるべきことは 4D フォームに Web エリアを配置することです。この例題には Contacts テーブルがあります。データベースストラクチャは Contacts と Appointments テーブルで構成されていて、1 Contacts レコードに N Appointments レコードがリレートされています。



このリレーションが示す通り、Contacts テーブルの入力フォームに Web エリアを挿入します。これにより Contacts レコードごとの Appointments を、カテゴリごとに、Web エリアに表示させることができます。

## Web エリアを作成する

デザインモードで Contacts テーブルの入力フォームを開き、オブジェクトリストの一番下の項目から Web エリアを選択します。



フォームにエリアを描画します。

A screenshot of a web form titled **Contact**. The form has fields for Name, Address, City, and Email, each with a corresponding label and a text input field. Below the form fields, there is a section titled **Appointments for Contact** with a sub-header **Click on a category below to show that type of Appointment.** Under this section, there is a category labeled **4D\_WA\_Web** with a green globe icon.

プロパティリストで Web エリアのオブジェクト名、変数名を設定します。Web エリアコマンド"WA OPEN URL"を使用して Web エリアにコンテンツをロードする際にここで指定する Web エリアの参照が必要になります。

▼ Objects	
Type	Web Area
Object Name	4D_WA_Web
Variable Name	4D_WA_Web

## フォームを Composite モードに設定する

Web エリアを正しくロードするためには、"4D v11 Addendum r2"に記載されているとおり、フォームが Composite モード用にデザインされているか確認してください:

*"Web エリアを表示させるためには、エリアを"compositing モード"のウィンドウに置かなければなりません。Mac OS のこの内部的なウィンドウ処理モードは、すべての 4D ウィンドウで使用されているわけではありません。*

4D v11 SQL では、"compositing モード"のウィンドウは:

- *Open form window* コマンドで生成されるすべてのウィンドウ
- *Compositing Mode* (定数値 4096) タイプを持つ、*Open window* コマンドで生成されたウィンドウ
- デザインモードで (ボタンで実行された)、プロジェクトフォームを表示するウィンドウです。"

私は *Open form window* を使用するようにしています。以下はファイルメニューから Demo をロードする際に使用されるコードです (ファイル->Start Demo):

```
C_LONGINT(($windowRef_demo)
$windowRef_demo:=Open form window([Contacts];"ContactsInput";Plain window
;Horizontally Centered ;Vertically Centered ;*)

ALL RECORDS([Contacts])
CREATE SET([Contacts];"current")
ORDER BY([Contacts];[Contacts]City)
MODIFY SELECTION([Contacts];*)

CLOSE WINDOW($windowRef_demo)

SET MENU BAR(1)
CLEAR SET("current")
```

*Open form window* コマンドに関する詳細は以下のページを参照してください:

<http://www.4d-japan.com/docs/CM1/CM100675.HTM>

## ローカルに HTML ファイルを作成しロードする

次に行うことは入力 HTML ファイルをロードすることです。ファイルはテンプレートとして動作し、PROCESS HTML TAGS で処理される 4D タグを含んでいます。Mac および PC でファイルにアクセスするための 4D コードを見てみましょう。

ファイルは On Startup データベースメソッドでロードされます。以下がコードです:

```
C_LONGINT($platform_l)
C_TEXT($templatePath_t)
C_TEXT(<>sep;<>4dfolder_t;<>outputPath_t)
C_BLOB(<>templateListHtml_x)

`ファイルパスを構築するためにプラットフォームを取得する
PLATFORM PROPERTIES($platform_l)

<>4dfolder_t:=Get 4D folder(Database Folder )
If ($platform_l=Windows )
    <>sep:="\\"
Else
    <>sep:=":"
End if

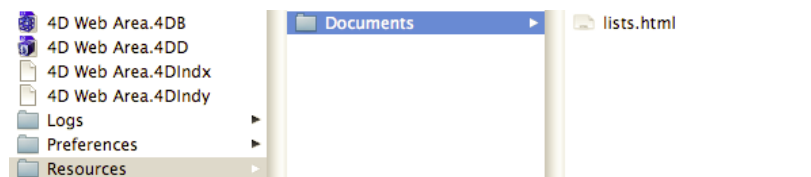
$templatePath_t:=<>4dfolder_t+"Resources"+<>sep+"Documents"+<>sep+"lists.html"
DOCUMENT TO BLOB($templatePath_t;<>templateListHtml_x)

<>outputPath_t:=<>4dfolder_t+"Resources"+<>sep+"Documents"+<>sep+"output.html"
```

このコードについては後ほど説明します。

## HTML ファイルをロードする

入力 HTML ファイルはストラクチャの近くに置く必要があります。この例題ではストラクチャと同階層にある Resources フォルダ内に"Documents"フォルダを作成し、その中に HTML ファイルを配置します。



次の節で使用するので、フォルダ名とファイル名を覚えておいてください。

この例題では入力ファイル名は"lists.html"です。これにはリストを表示したり隠したりするための JavaScript コードやスタイルを設定する CSS、4D データをロードするための 4D タグが含まれています。"lists.html"のコードについて知りたければ"入力 HTML ファイルの定義"を参照してください。

このディレクトリへのパスはインタープロセス変数<>outputPath\_tにも格納されます:

```
<>outputPath_t:=<>4dfolder_t+"Resources"+<>sep+"Documents"+<>sep+"output.html"
```

"output.html" HTML ファイルはダイナミックに生成されます。"lists.html"の 4D タグが処理されると結果が"output.html"に格納され、このファイルの内容が Web エリアに表示されます。

## プラットフォームに基づきファイルを取得する

HTML にアクセスするにはファイルパスを知らなければなりません。プラットフォームにより異なるセパレータが使用されているため、ユーザのプラットフォームをテストしてファイルにアクセスする必要があります。

以下はユーザのプラットフォーム (Windows または Mac) をチェックし、HTML ファイルのパスを定義するコードです。

```
`ファイルパスを構築するためにプラットフォームを取得する
PLATFORM PROPERTIES($platform_l)

<>4dfolder_t:=Get 4D folder(Database Folder )
If ($platform_l=Windows )
    <>sep:="\\"
Else
    <>sep:=":"
End if
```

パスはストラクチャファイルから始まってハードコードされていることに留意してください。前の節で示した通り、ここでフォルダ名と HTML ファイル名が使用されています。

もちろんデータベースストラクチャから HTML ファイルへのパスだけでは十分ではありません。Get 4D folder コマンドを使用してデータベースフォルダへの絶対パスを取得します。

```
<>4dfolder_t:=Get 4D folder(Database Folder )
```

Get 4D folder に関する詳細は以下の Web ページを参照してください:

<http://www.4d-japan.com/docs/CMJ/CMJ00485.HTM>

## 変数にファイルをロードする

プラットフォームごとのセパレータを取得したら、後で使用するためにパスを変数に格納します。指定したパスにある入力ファイルは DOCUMENT TO BLOB を使用して BLOB 変数に変換および格納されます。これは後ほど PROCESS HTML TAGS で処理されます。

```
$templatePath_t:=<>4dfolder_t+"Resources"+<>sep+"Documents"+<>sep+"lists.html"  
DOCUMENT TO BLOB($templatePath_t;<>templateListHtml_x)  
  
<>outputPath_t:=<>4dfolder_t+"Resources"+<>sep+"Documents"+<>sep+"output.html"
```

これが On Startup で実行されることです。後ほど処理するためにテンプレート HTML ファイル ("lists.html") の内容を変数に格納し、出力ファイルまでのパスを定義しました。

次に Web エリアがロードされるたびに何が起こるかを見てみましょう。

## Web エリアを表示する

HTML テンプレートの準備ができていますので、Contacts テーブルの異なるレコードごとに Web エリアが読み込まれるたびにダイナミックなコンテンツを表示できます。以下はフォームメソッドのコードです:

```
C_BLOB($webblob)  
  
Case of  
  : (Form event=On Load )  
  
    、 HTML テンプレートの BLOB を処理する  
    PROCESS HTML TAGS(<>templateListHtml_x;$webblob)  
  
    、 このファイルは以下の場所にある /Resources/Docusments/output.html  
    BLOB TO DOCUMENT(<>outputPath_t;$webblob)  
    WA OPEN URL(*;"4D_WA_Web";<>outputPath_t)  
End case
```

このコードについては後ほど詳しく説明します。

このフォームがロードされるたびに (On Startup で取得した) HTML テンプレートが PROCESS HTML TAGS を使用して処理されます。PROCESS HTML TAGS は結果を \$webblob BLOB に格納します。

PROCESS HTML TAGS に関する詳細は以下の Web ページを参照してください:

<http://www.4d-japan.com/docs/CM1/CM100816.HTM>

結果の BLOB は外部 HTML ファイルとして "output.html" という名前で BLOB TO DOCUMENT コマンドを使用して保存されます。"output.html" ファイルへのパスは <>outputPath\_t 変数に格納されています。



次は最終ステップです。"output.html"を Web エリアに表示します。

## WA OPEN URL コマンド

"WA OPEN URL"コマンドは HTML ファイルの内容を Web エリアに表示させるために使用する Web エリア専用のコマンドです。引数として Web エリアのオブジェクト名または変数名と、内容を定義した HTML ファイルのパスを渡します。

```
WA OPEN URL(*;"4D_WA_Web";<>outputPath_t)
```

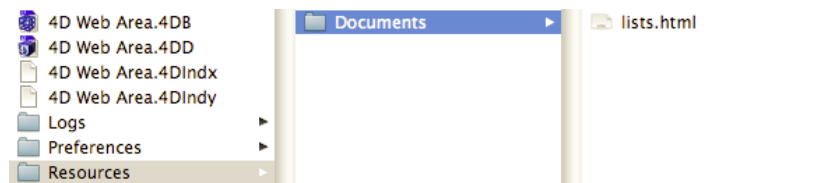
WA OPEN URL に関する詳細は以下の Web ページを参照してください:

<http://www.4d-japan.com/docs/CMJ/CMJ01020.HTM>

## 入力 HTML ファイルについて

この機能の実装はすべて HTML テンプレートファイル ("lists.html") に記述されていて、これにより Web エリアは 4D データを動的に扱うことができます。4D HTML タグにより、情報へのアクセスは非常に簡単です。

lists.html ファイルは以下の場所にあります:



## JavaScript 関数と折りたたみ可能なエリア

一つの問題を解決するための方法は複数あるものです。HTML でダイナミックな動作をさせる方法もいろいろあります。同じことを行うために異なる方法を実行する数多くのスクリプトがあります。それゆえここで紹介するエリアの展開や折りたたみを行う JavaScript コードは、この方法が唯一というわけではありません。

以下は渡された<div>要素を表示したり非表示にしたりする"toggleDisplay(id)" JavaScript 関数です。このコードは CSS プロパティ.style.display を使用して要素の表示/非表示を切り替えます。

```
function toggledisplay (id){
  if (document.getElementById) {
    var obj = document.getElementById(id);
    obj.style.display = (obj.style.display=='block'? 'none': 'block');
  }
}
```

この関数は'Personal'などのカテゴリヘッダ div 要素がクリックされるたびに呼び出されます。

JavaScript で表示/非表示を切り替える方法は複数あります。この例題のコードはたったの6行です。しかしスライドアニメーションやフェード効果などを付加できるより堅牢なコードもあります。さまざまな方法を試すことをお勧めします。

## 4D HTML タグ

以下はテンプレートファイルの body 部です。Appointments リストを表示するためにどのように 4D HTML タグが使用されているかを示しています。タグは Appointments のカテゴリごとにリストの項目を生成します。

```
<!--#4DSCRIPT/W_Appointments-->
<!--#4DLOOP appointmentCategory_at-->

<!--#4DSCRIPT/W_LoadAppByCategory-->

<div class="header" onclick="toggledisplay('<!--#4DVAR (Record
Number([Appointments]))-->'); return false"><!--#4DVAR [Appointments]Category--
></div>
<div id="<!--#4DVAR (Record Number([Appointments]))-->" width="560" border="1"
cellspacing="0" style="display: block;">

<!--#4DLOOP [Appointments]-->
  <div style="border-bottom:1px solid #efefef;">
    <div class="detail"><!--#4DVAR [Appointments]AppDetail--></div><br>
    <div>
      <div class="location"><!--#4DVAR [Appointments]Location--></div>
      <div class="date"><!--#4DVAR [Appointments]AppDate--></div><br>
    </div>
  </div>
<!--#4DENDLOOP-->

</div>

<!--#4DENDLOOP-->
```

まず#4DSCRIPT タグを使用して'**W\_Appointments**'メソッドを呼び出しています。このメソッドは Contacts フォームに表示されているカレントユーザに関連する Appointments をクエリします。そして Category フィールドの DISTINCT VALUE 値を取り出して appointmentCategory\_at 配列に格納します。以下は'**W\_Appointments**'のコードです:

```

C_TEXT($1)
ARRAY TEXT(appointmentCategory_at;0)
QUERY([Appointments];[Appointments]ContactID=[Contacts]ID)
DISTINCT VALUES([Appointments]Category;appointmentCategory_at)

```

最初の#4DLoop では Category 値ごとにループを行うために配列 appointmentCategory\_at が指定されています。

カテゴリーごとに、#4DSRIPT を使用して'W\_LoadAppByCategory'メソッドを呼び出します。このメソッドは現在のカテゴリーに一致する Appointments のセレクションを作成します。以下は'W\_LoadAppByCategory'のコードです。

```

C_TEXT($1)
QUERY([Appointments];[Appointments]ContactID=[Contacts]ID;*)
QUERY([Appointments];[Appointments]Category=appointmentCategory_at{appointmentCategory_at})

```

最後の#4DLOOP [Appointments]では、セレクションのレコードごとにループを行い、#4DVAR で情報を表示します。

4D HTML タグに関する情報は以下の Web ページを参照してください:

<http://www.4d-japan.com/docs/CMJ/CMJ02070.HTM>

これで入力 HTML に記載されたテンプレートを使用し、ダイナミックな情報がスタイルやレイアウトの自由度の高いエリアに表示されるようになりました。

Personal	
<b>Early Christmas Shopping</b>	
Oakland	11/19/08
<b>Buy Plasma TV</b>	
Marin	12/10/08
<b>Secret Santa drawing</b>	
Oakland	11/13/08
<b>Watch son's baseball game</b>	
San Jose	12/8/08

情報が複数行にわたりカスタマイズされた方法で表示されていることに注目してください。スタイル設定は HTML と CSS で行われています。そして JavaScript を使用し、カテゴリーヘッダをクリックするとカテゴリーの情報の表示/非表示が切り替わります。

## まとめ

このテクニカルノートでは、Web エリアに表示される折りたたみ可能なリストの作成方法を示しました。4D タグを使用することでダイナミックに 4D データを取得でき、HTML と CSS を使用してルック&フィ

ールをカスタマイズできます。これによりエリアが 4D フォームの拡張のように動作します。Web エリアの作成からローカル HTML の Web エリアへのリンク、そして 4D HTML タグを使用してダイナミックに HTML ファイルのコンテンツを定義する方法を見ました。短くて簡単な JavaScript 関数を使用してリストエリアを展開したり折りたたんだりできました。このようにして Web エリアを使用すると、以前の 4D ではなしえなかったことが可能になります。