

AppleScript Working》

3 – AppleScript Studio で テーブルコントロールを使う

AppleScript Studio でのプログラミングの話題の続きとして、テーブルの使い方を解説しよう。NSTableView というクラスが Cocoa では用意されており、表形式の表示やデータ入力に使えるようになっている。昔の開発環境だと、単純な機能であったが、NSTableView は汎用的なデータの保存のために複雑になっているため、ある程度はどの概念を理解していないと、AppleScript のプログラミングでも理解できないことも多い。今回はテーブルにデータを追加し、データを取り出すというところまでのプログラムの作成方法を解説しよう。

NSTableView とデータソース

よくある RAD 開発システムの場合、テーブルだとたいがい「表と列を指定して値をセットしたり取り出す」ということができるようになっている。しかしながら、Cocoa で用意されたテーブルのコントロールである NSTableView は、さまざまな処理に対応できるのであるが、特に、表に表示するデータの保存についても自由にカスタマイズできるという点が大きな違いだ。もちろん、柔軟性は高いのであるが、逆に言えば、簡単に使うことができないとも言える。

AppleScript Studio は、Cocoa のコンポーネントを使えることが大きな特徴であり、もちろん、テーブルには NSTableView を使う。しかしながら、そのままの状態だと単純に使いにくいこともあるので、データをテーブルなどに保存するためのオブジェクトも用意されている。その使い方さえ知っておけば、よくあるテーブルコントロールと同じように使えるのである。

まず、NSTableView では、データの保存を「データソース (Data Source)」という別のオブジェクトで行う。つまり、テーブル自体にはデータを記憶することはできず、その背後に用意されたデータソースにデータを保存する。Cocoa ではそのデータソースを構築するためのプロトコルとして NSTableDataSource というクラスが定義されている。ここでは、つまりはリクエストがあれば表形式のデータを取り出すなどのメソッドを用意したクラスを定義するためのものになる定義が行われていて、これに従って、プログラマがデータを保存する定義を用意するというのが基本となる。

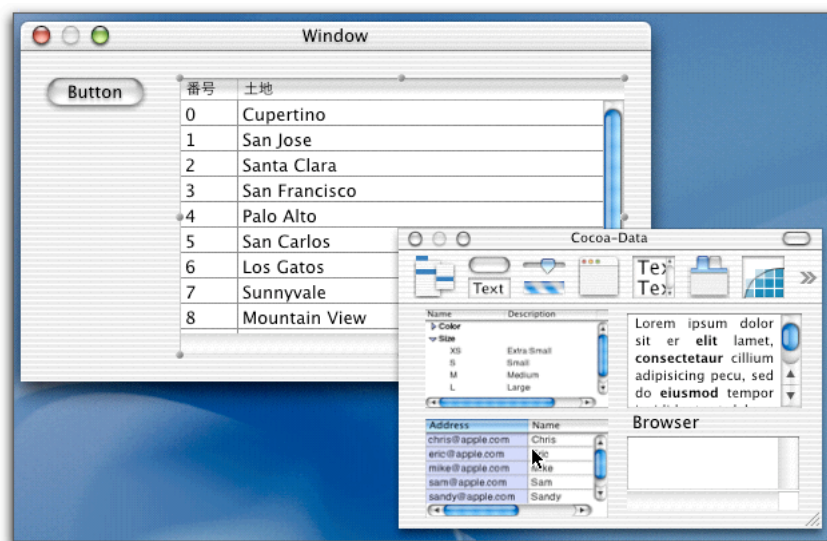
Cocoa では NSOutlineView などでも、同じような手法、つまりデータソースという別のオブジェクトを使う方法を使ってのデータの保存を行っている。AppleScript Studio では、テーブルやアウトラインで使えるデータソースを、オブジェクトとして用意してくれている。ASKDataSource というコンポーネントがある。このコンポーネントを、とりあえず用意しておき、テーブルやアウトラインからこれを使うように設定することで、データの保存ができる。ASKDataSource への読み書きは、AppleScript のプログラムで行える。

いずれにしても、ユーザインタフェースのコンポーネントと、データの保存のコンポーネントが別々に用意されているということを知っておくのが第一歩だ。

Interface Builder でのテーブルとデータソースの設定

実際に、テーブルをウィンドウに配置したときに、データソースをどのように用意するのかを解説しよう。Project Builder で、AppleScript Application のテンプレートを使ってプロジェクトを用意すると、そこには、MainMenu.nib という nib ファイルがあるのでそこに設定を追加しよう。MainMenu.nib をダブルクリックして開き、そこにある Window にテーブルを配置したとする。テーブルを配置するには、ツールパレットの左から 5 番目のアイコンを選択して、パレットのタイトルを「Cocoa-Data」としておき、そして、そこに表示される左下のものがテーブルのコンポーネントである。

テーブルのコンポーネントと、配置したテーブル



テーブルを配置した後は、「Cupertino」などのデータが表示されるが、これはレイ

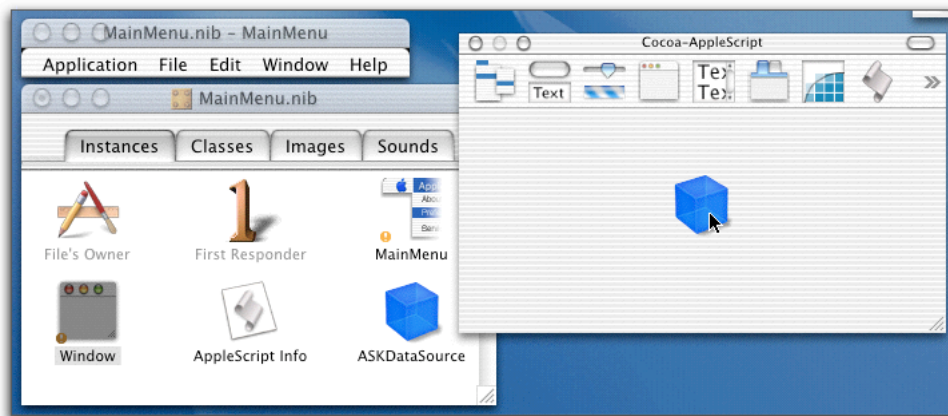
アウトの参考のためだけのもので、実際にこのデータが表示されるわけではない。

この NSTableView のコンポーネントは、少し複雑な構成になっている。単に 1 つのオブジェクトではなく、まず、NSScrollView というスクロールを行うコンポーネントがまずあって、その中に NSTableView が含まれている。だから、単に配置したコンポーネントを選択すると、それは、NSScrollView である。ダブルクリックすると、NSTableView が選択されるのが基本である。Info パレットのタイトルに、選択されているオブジェクトのクラス名がでてくるので、をよく見ながら作業をするというのがコツだ。ここでは、AppleScript の Name として、NSScrollView には scView1、NSTableView には、table1 という名前を付けてある。

それから、列の見出しであるが、プログラムで設定等もできるが、固定的なものだと、最初からつけておくのが一般的だろう。そのとき、コントロールをダブルクリックして、NSTableView の方を選択し、さらに、列をクリックすると、Info パレットには、NSTableColumn というクラス名がでてくる。つまり、テーブルの中の 1 つ 1 つの列が実はさらに別のオブジェクトとして管理されているのである。そして、列見出しの部分をダブルクリックすると、列見出しをキー入力できるようになるので、そこで任意にキータイプすれば良い。

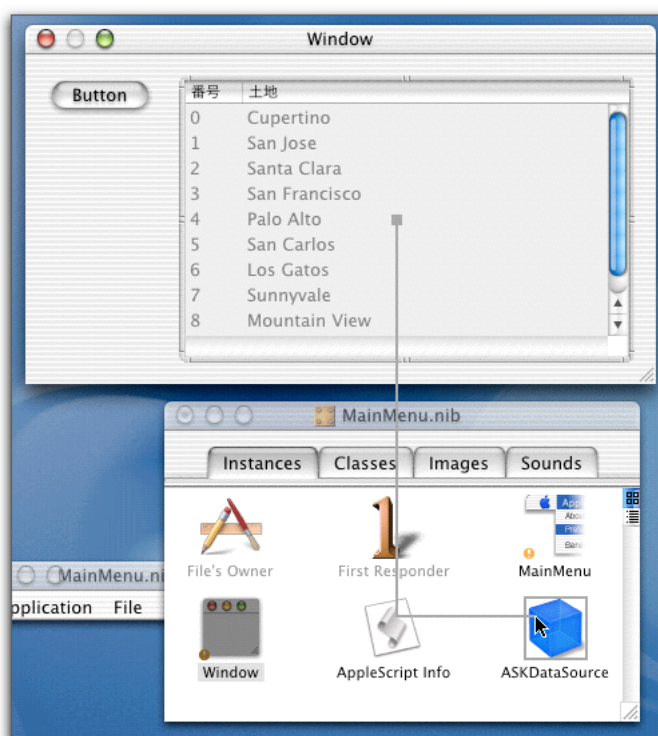
次に、データソースを用意しよう。ツールパレットの左から 8 番目に AppleScript のロゴのアイコンがあり、これをクリックすると、青い立方体のアイコンだけがツールパレットにでてくる。これを単に MainMenu.nib のウィンドウの中にドラッグして、オブジェクトを追加する。ASKDataSource という名前になっているが、名前を変えたりはできないので、とにかくここに作っておくのである。ちなみに、アプリケーションを起動すると、MainMenu.nib ファイルがロードされる nib ファイルであると設定されている。したがって、MainMenu.nib に設定されているオブジェクトのインスタンス化が自動的に行われるので、ここに作った ASKDataSource はオブジェクトとして存在するのである。

データソースのオブジェクトを追加する



こうしてデータソースが確保されるのであるが、テーブルがそのデータソースを使うという設定を、Interface Builder 上で行っておく必要がある。テーブルをダブルクリックして、NSTableView が選択された状態で、テーブルの中のどこからでもいいので、control キーを押しながらドラッグし、ASKDataSource のところまでドラッグ&ドロップを行う。

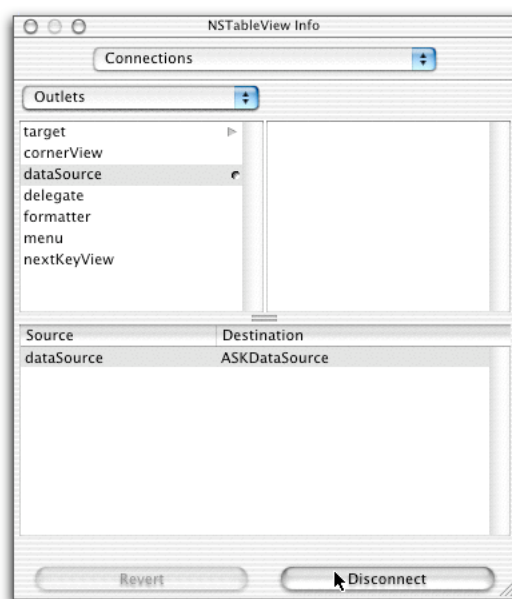
NSTableView と ASKDataSource を結ぶ



すると、Info パレットに、NSTableView の Connections の設定が表示される。ここで、dataSource というアウトレットが選択されていることを確認して、Connection ボタンをクリックする。（WebObjects がインストールされているのなら、左上にあるポップ

アップメニューで、Outlets を選択する。)

dataSource と ASKDataSource を接続する



これで、NSTableView はデータソースを持つことができるようになった。

AppleScript プログラム上でのテーブル

プロジェクトには AppleScriptKit.asdictionary というファイルがあって、用語が参照できるので、さっそくそれを見ながらのプログラミングということになるのだが、余分な情報があって見づらいのも確かだ。いずれにしても、次のようなオブジェクトの階層があることを知っておく必要がある。

まず、テーブルのオブジェクトは、Data View Suite にある Table View クラスである。このクラスがテーブルそのものであるが、テーブルの各列としては、さらに Table Column クラスがあり、このオブジェクトは 1 つのテーブルに複数個存在する。

一方、Table View クラスからは、データソースを示すクラスである Data Source も複数個存在するという定義になっているが、Interface Builder で接続した ASKDataSource を参照するには、「Data Source 1」として、1 つ目のデータソースを指定すればよい。

続いて、Data Source クラスを見るが、いろいろとややこしい感じがするだろう。ここで、考え方として、データソースをデータベースのテーブルのように考えればよい。つまり、データソースには複数のカラムがあるというのが 1 つの見方だがこの場合のカラムはデータではなくテーブル構造を示すものだ。そして、データ自体は、まず行

に保持されるから、データソースには複数の行があるということになる。そして、その行に対して、カラムに対応した複数のセルがあるということになる。1 対多の関係を→で示すと、次のようになる。

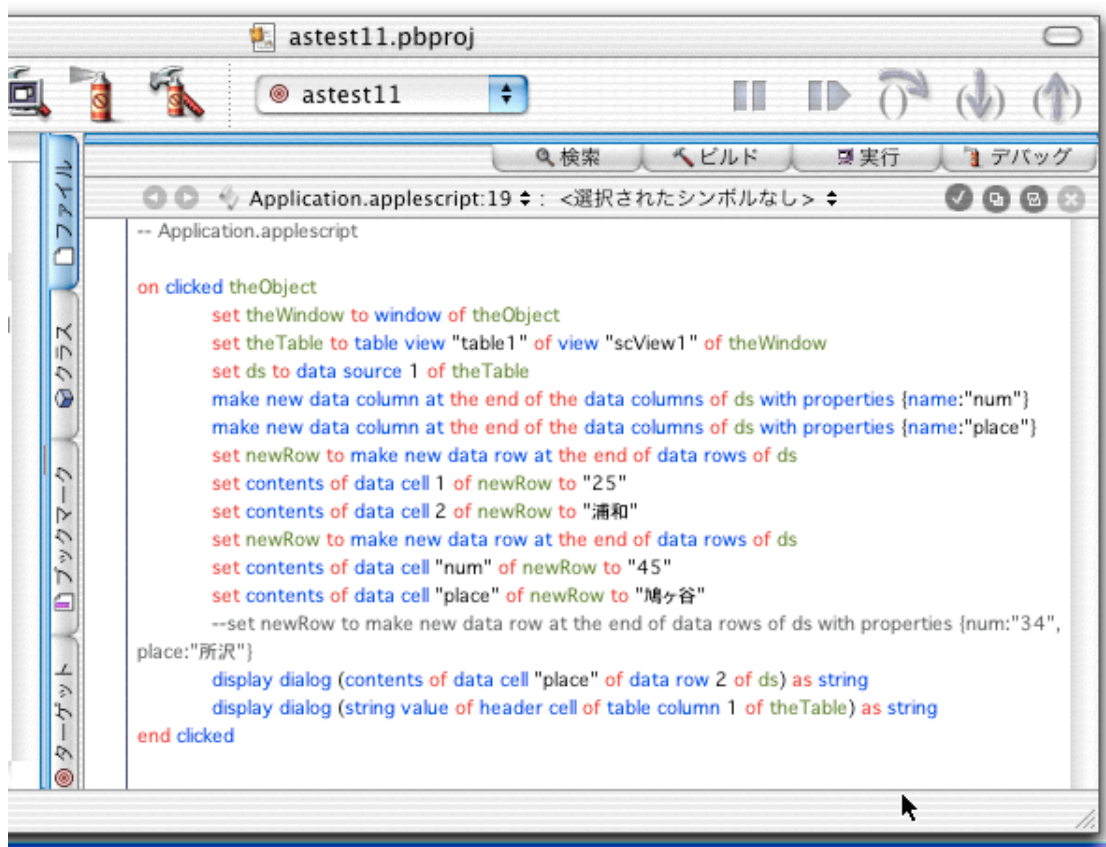
テーブル定義：Data Source → Data Column

記録データ：Data Source → Data Row → Data Cell

そして、セルにあるデータそのものは、Data Cell クラスの contents プロパティで参照することができる。

ここで、とりあえずプログラムをみていただきたい。前記のユーザインタフェースで、ボタンをクリックしたときに、on clicked ハンドラが呼び出される。プログラムは後でテキストでも示す。

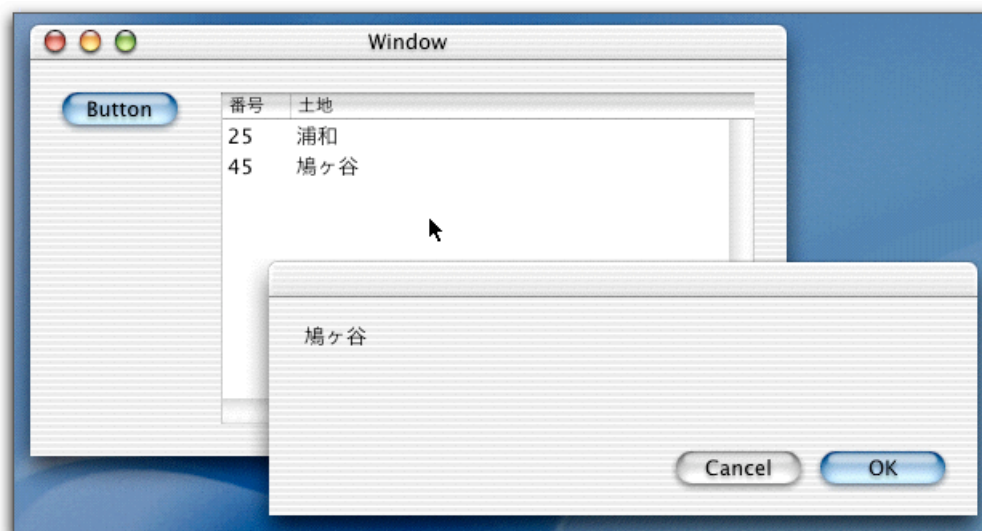
作成したプログラム



実際の動きは次のようになる。プログラムを実行すると、まずはウィンドウが表示され、ボタンとテーブルが表示されるが、テーブルの中は何もデータはない。見出しだけ、Interface Builder で設定した通りになっている。

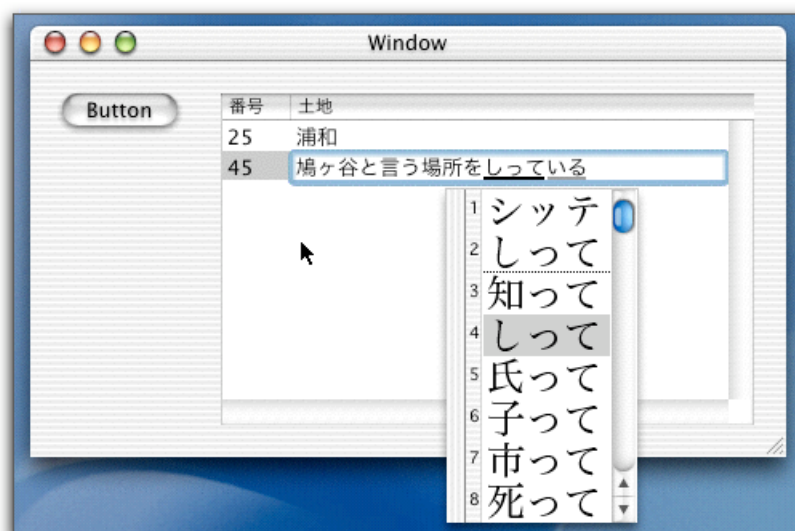
そこで、ボタンをクリックすると、テーブルにデータが入力される。また、ダイアログボックスが表示されるが、2 行目の 2 列目のデータが取り出されて、ダイアログボックスで表示されているわけだ。さらに、その後に、1 列目の列見出しの文字列もダイアログボックスで表示される。

ボタンをクリックするとテーブルにデータがセットされる



さらに、テーブルのデータは、ダブルクリックするとセル単位で編集できるようになっている。編集をできるかできないかは、Interface Builder での設定で可能となっている。

テーブルではテキストの編集もできる



プログラムをチェックしていこう。まず、ここでクリックしたボタンの clicked ハンド

うだから、引数の theObject はボタンを参照している。従って、以下のようにすれば、変数の theWindow は、ウインドウそのものを参照する。

```
set theWindow to window of theObject
```

次にテーブルをどうすれば参照できるかだが、NSScrollView の中に NSTableView があったことを思い出してほしい。そうすれば、以下のようにすると、theTable がテーブルを参照し、ds がそのテーブルのデータソースを参照できるようになる。

```
set theTable to table view "table1" of view "scrollView1" of theWindow
```

```
set ds to data source 1 of theTable
```

初期状態では、データソースにはカラムが一切定義されていないため、行の追加はできない。フィールド定義されていないテーブルのようなものである。そこで、データソースに、新たにカラムを作成するが、make コマンドを使う。つまり、データソースの中の data column の集合の最後に、新しく data column を作る。Name というプロパティは、その列の名前である。列の名前は、テーブルに表示された列名と異なってもかまわないが、データソースを操作するときには、この名前を使うことができる。(2 行に分割される部分は「ツ」と記載するが、実際には 1 行で入力してもらいたい。)

```
make new data column at the end of the data columns of ds ツ
```

```
with properties {name:"num"}
```

```
make new data column at the end of the data columns of ds ツ
```

```
with properties {name:"place"}
```

これでカラムが定義できたので、続いて行を作ってデータを保存しておく。行についても、同様に、データソースの行の集合の最後に新しく data row を make コマンドで作る。ここで、作った行への参照を newRow という変数におさめておく。そして、その新しい行に含まれている data cell オブジェクトを参照し、その contents プロパティに対して文字列を設定することで、実際にテーブルにその文字が見えるようになるという具合である。


```
set newRow to make new data row at the end of data rows of ds
set contents of data cell 1 of newRow to "25"
set contents of data cell 2 of newRow to "浦和"
set newRow to make new data row at the end of data rows of ds
set contents of data cell "num" of newRow to "45"
set contents of data cell "place" of newRow to "鳩ヶ谷"
```

data cell については、番号で指定してもいいが、data column の title プロパティの値を利用して、どのセルかと言うことを指定することもできる。上記のプログラムは 2 つのバリエーションを示しているので、プログラムと、テーブルでのデータ設定結果を見比べてもらいたい。

ところが、ここで、次のようなプログラムは動かないものかと若干は期待したけど、これはだめであった。やはり、data row がさらに複数の data cell というオブジェクトにつながっているため、直接書き込むことはできないのである。

```
make new data row at the end of data rows of ds ツ
with properties {num:"34", place:"所沢"}
```

逆にデータソースからデータを取り出す場合も、まったく同様である。たとえば、2 行目の 2 列目のデータを取り出して、ダイアログボックスに表示するのは、次の部分である。

```
display dialog (contents of data cell "place" of data row 2 of ds) as string
```

また、テーブルに見えている列の名前は、table column オブジェクトを参照して、その header cell を参照し、さらにその string value プロパティを参照することで、得ることができる。このあたりは、オブジェクト階層が深い。

```
display dialog (string value of header cell of table column 1 of theTable) as string
```

列幅については、header cell の cell size プロパティを取得することでわかる。{ 30, 17 } のようなリスト形式で得られ、単位はドット数である。1 項目目が幅なので、

item 1 of cell size of header cell of table column 1 fo theTable

のようにすれば、列幅が分かるということだ。

他に知りたいものとなると、まずは行数となると思われる。行数は、count コマンドで数えればいいので、変数 ds がデータソースを参照しているとすれば、

set numRows to count (data rows of ds)

とすれよい。

テーブルの中で選択されている行や列を知るには、table view クラスの selected row や selected column プロパティを使う。これだと、何行目が選択されているのかが分かるが、1 行目あるいは 1 列目は、いずれのプロパティも「1」になる。これらのプロパティが 0 の場合には選択されていないことになる。たとえば、2 行目を選択すると、行全体は選択されているが、特定の列は選択されていないので、selected row は 2 であるが、selected column は 0 となっている。なお、列を選択できないように Attributes を設定すると、selected column プロパティの取得時にはエラーが出る。

データソースとテーブルが分離していることと、データソース内部でのデータ構造を理解することで、テーブルは自由に扱えるようになるだろう。

(この項、以上)