;A_TravelAdvisor_CreateIF.rtfd;;¬ Previous Section   ;C_TravelAdvisor_DefineClasses.rtfd;;¬ Next Section

3. Travel Advisor Tutorial

The Design of Travel Advisor

Travel Advisor is much like Currency Converter in its basic design. Like Currency Converter, it's based on the Model-View-Controller paradigm. A controller object (TAController) manages a user interface comprised of Application Kit objects. Also as before, the controller sends a message to the Converter object to get the result of a computation. In other words, the Converter object is reused.

TA_Design.eps ¬

Travel Advisor's view objects, in terms of Model-View-Controller, are all off-the-palette Application Kit objects, so the following discussion concentrates on those parts of the design distinctive to Travel Advisor.

Model Objects

Travel Advisor's design is more interesting and dynamic than Currency Converter's because it must display a unique set of data depending on the country the user selects. To make this possible, the data for each country is stored in a Country object. These objects encapsulate data on a country (in a sense, they're like records in a relational database). The application can manage potentially hundreds of these objects, tracking each without recourse to a ªhardwiredº connection.

Another model object in the application is the instance of the Converter class. This instance does not hold any data, but does provide some specialized behavior.

Controller

The controller object for the application is TAController. Like all controller objects, TAController is responsible for mediating the flow of data between the user interface (the View part of the paradigm) and the model objects that encapsulate that data: the Country objects. Based on user choices in the interface, TAController can find and display the requested Country object; it can also save changes made by users to the appropriate Country object.

What makes this possible is an NSDictionary object (called a *dictionary* from here on). A dictionary is a container that stores objects and permits their retrieval through key-value associations. The key is some identifier paired with an object in the dictionary (the object often holds the identifier as one of its instance variables). To get the object, you send a message to the dictionary using the key as an argument (**objectForKey:**).

```
NSColor *aColor = [aDictionary objectForKey:@°BackgroundColor°];
```

A Country object holds the name of a country as an instance variable; this country name also functions as the dictionary key. When you store a Country object in the dictionary, you also store the country name (in the form of an NSString) as the object's key. Later you retrieve the object by sending the dictionary the message **objectForKey:** with the country name as argument.

Storing Data Source Information

TAController also manages the data source for the table view on the interface. It stores the keys of the dictionary in an array object (NSArray), sorted alphabetically. When the table view requests data, the TAController ªfeedsº it the objects in the array.

Creation of Country Objects

Another important point of design is the manner in which the Country objects are created. Instead of Interface Builder creating them, the TAController object creates Country objects in response to users clicking the Add button.

Delegation and Notification

An essential aspect of design not evident from the diagram are the roles *delegation* and *notification* play. The TAController object is the delegate of the application object and thereby receives messages that enable it to manage the application, which includes tracking the edited status of Country objects, initiating object archival upon application termination, and setting up the application at launch time.

Related Concepts: ;TravelAdvisorConcepts.rtfd;linkMarkername TheCollectionClasses;, The Collection Classes

;TravelAdvisorConcepts.rtfd;GettinginontheAction:DelegationandNotification;¬ Delegation and Notification

Also see ``Implementing the TAController Class'' ;¬;E_TravelAdvisor_ImplemTAController.rtfd;;¬ for a diagram that depicts the data relationships of TAController as data source.