

RZToDoList

Ralph Zazula

zazula@next.com

Released to Public Domain - 4/4/95

This application started as a test-bed for a custom NXBrowserCell I developed called **RZBrowserCell**. This object (and its associates) are found in the RZBrowserCell subproject of this application.

RZBrowserCell

This NXBrowserCell subclass provides a simple way to develop browser-based interfaces similar to that found in Mail.app (and RZToDoList!). An RZBrowserCell can display tab-aligned icons

and text in multiple fonts and colors.

To configure a `RZBrowserCell`, you specify the location of tab-stops and the contents of the cell at each stop. For example:

```
id prototypeCell = [[RZBrowserCell alloc] init];
[prototypeCell addFixedTab:4];
[prototypeCell addFixedTab:24];
[prototypeCell addFixedTab:-85];
[browser setCellPrototype:prototypeCell];
```

This configures the browser's prototype Cell to be a `RZBrowserCell` with fixed tab stops at 4 & 24 pixels from the left edge and 85 pixels from the right edge.

When filling the browser (through the standard delegate method) you set the contents of the cell by specifying an icon or text string (with optional font and color attributes) and a tab location, e.g.,:

```
...  
[theCell setImageNamed:"MyIcon" at:0];  
[theCell setText:"MyString" at:1];  
[theCell setText:@"%d items" at:2, [self numItems]];  
...
```

RZToDoList

This is a simple to-do list application with various types of items that can be added: appointments, low-high priority items. The application has a simple heuristic built in for sorting items (item raise in the list as deadlines draw near).

Files can be dragged into the text at the bottom of the panel and associated with each to-do item.

There is also a distributed-objects API (ToDoServer.h) that can

be used to add items from another program (the infrastructure for sending notifications to applications is in place but not implemented).

This application is fairly robust but not what I would call a commercial-quality product. It is mainly intended to show an example of how the RZBrowserCell could be used.

The files

I don't use indent. It's evil. I set tabs and indent to "3" in edit. If a file looks funny to you, try this:

- select all
- Command-Shift-| <yes, vertical bar>
- expand -3

Unfortunately, most of this code is lacking voluminous comments. I didn't originally intend on releasing this code and it'd never happen if I had to go back and re-comment things.

There is some trickery in the RZBrowserCell in order to share lists of text and tab tokens across multiple cells in the browser. The thing to remember is that a NXBrowser may go off freeing cells when you do a [browser loadColumnZero], so I had to make sure that the -free method of RZBrowserCell didn't nuke stuff in another instance - thus, the reason for all the reference counting.

The code for dealing with in-line images in Text is pretty old but works (although there is a bug in that you have to make a selection first before dragging in a file).

Julie Zelenski's ObjectError class comes to the rescue of a few

errant objc_msgSends() that I haven't bothered to track down. So, don't be alarmed if you see console messages like:

```
RZToDoList[9251]: SuperText: does not  
recognize selector -nextText
```

Feel free to compile debug and figure out what's happening. At least with OE in there, the app keeps cruising along after the bogus message is sent...

I hope you find this code useful.

Thanks go to Dan Marusich for, as always, great artwork!

Ralph Zazula
NeXT Computer
Spring '95