NEWTON Q&A: ASK THE LLAMA

Q *I'm trying to remove indexes that I've added to the names soup. But the code to do it is kind of First I have to go through all the indexes to see if my index is in the soup. Then if I find the ind can remove it. Is there an easier way?*

A Yes, there's an easier way. Remember that a call to RemoveIndex will throw an exception if it's passed an invalid index. You can wrap your code in an exception handler and prevent the invali index exception from leaving your code:

```
ExceptionBasedRemoveIndex := func(theSlotsym, theSoupName)
begin
   local theSoup;
   foreach store in GetStores() do
   begin
      theSoup := store:GetSoup(theSoupName);
      if theSoup then
         try
            theSoup:removeIndex(theSlotsym);
         onexception |evt.ex.fr.store| do
            if CurrentException().error <> -48013 then
               ReThrow();
   end;
end
```

This function will remove a particular index (specified by theSlotsym) from a particular soup (specified by theSoupName) on all currently mounted stores. If the index exists, it will be remo otherwise, the exception thrown for trying to remove an invalid index will be caught and ignore If a different exception occurs, it will be rethrown so that other exception handlers or the systen can deal with the exception.

Q *I'm writing a utility that is an auto part. My utility needs preferences, but there's no application add preferences to. Where should I put my preferences?*

A The guidelines for preferences are simple:

- For an any addition that has an icon in the extras drawer, the preferences should be part of t application. Use the info button to access them.
- For something that has no icon in the extras drawer, add your preferences to the system preferences roll.

See the sample "Preefer Madness" on this issue's CD for more information.

Q *When text gets pasted into my paragraph view, that text is highlighted. I want to be able to dete when this happens and then be able to unselect the text. How do I do that?*

A When the text gets added, the viewChangedScript will get called with the slot parameter set to **'text**. You can use the SetHilite message to unhighlight the text. However, the viewChangedS will get called before the underlying implementation of the paragraph view has been changed. T means you need to call SetHilite in a deferred action.

Q *I'm writing a specialized application for a corporate customer. One of the requirements is that th application launch when the Newton is turned on (a "turnkey" application). Is there a way to do with the Newton?*

A You can use the installScript of your application to add a deferred action that opens your application:

```
constant kAppSymbol := '|autoLaunch:PIEDTS|;
installScript := func(partFrame)
begin
   AddDeferredAction(func() GetRoot().(kAppSymbol):Open(), []);
end;
```

This will launch your application whenever the Newton is restarted or a card containing your application is inserted. Note that if the application is closed before the Newton is restarted agair the application will not relaunch. Nor will the user be prevented from accessing other features c the Newton such as Names, Dates, or Extras Drawer; that's a much harder problem.

Q *I've been trying to use the protoRoll and protoRollItem to create a roll browser of my own. Everything works fine until I scroll. For a couple of these items I need to tap the down arrow tv for it to go to the next roll item. I see the scrolling view effect, but it just scrolls to itself. The he slot in each of the roll items has the same value as the height in their viewBounds slot. If I mov roll items around when they're added to the protoRoll (dynamically from my own protos), they fine. How can I fix this?*

A The problem is that one of the protoRollItems in the items array is larger than the protoRoll. If y make the roll browser larger than the largest roll item, all will work fine; otherwise, you have to scroll the roll item twice to move to the next roll item.

Also, since you imply that the entire large roll item is visible, I assume that the protoRoll has vClipping turned off. If you'd had clipping on, you would probably have noticed that the individual roll item was too large.

Q *I'm having some problems with margins when I'm faxing. A fax without a cover page has diff margins than a printed page. The actual viewBounds is the same, but the margins of the fax are different from the viewBounds.*

Also, a fax with a cover page has even different margins. The viewBounds is different, too (20 pixels shorter in height), but that's OK. The problem is that the actual margins when faxed are different from those specified by the viewBounds slot. Is this a known problem?

A Faxes with a cover page have a header line at the top of the fax which takes up those mysteriou pixels. In fact, it might be a bug that faxes without a cover page omit this header, but perhaps th only bug is not documenting that protoPrintFormat (which provides the cover page) also adds t header.

The way to find the correct page bounds is to set the viewBounds of your base print view to tha the parent. The base print view is usually a clView that is a child of a print layout. You can use following code in the viewSetupFormScript of your base print view to set your bounds to those your parent:

```
viewBounds := :Parent():LocalBox();
```

Q *I've got an auto part that installs a template for the formulas roll. On the roll item I've got a protoLabelInputLine for data entry, and a button that I want to use to clear the input line. My buttonClickScript is very simple:*

```
buttonClickScript := func()
begin
   SetValue(myInputLine.entryLine, 'text, "");
end;
```

The first time the button is tapped, the input line gets cleared OK; after that it never seems to wo no matter how I code it. Can you help?

A This is a very subtle problem. The answer will be revealed in stages, so that you to can experien the "Aha!"

Observation 1: When you edit the text in any clParagraphView, no new strings are generated. T existing string is destructively modified (excluding the usual _proto copying, of course).

Observation 2: During the compile cycle, the Newton Toolkit will turn all your strings into constants. Contrast this to using braces to construct a frame. As an illustration, assume you hav these three methods:

```
Method1 := func()
begin
   return {slot1: "also string"};
end;

Method2 := func()
begin
```

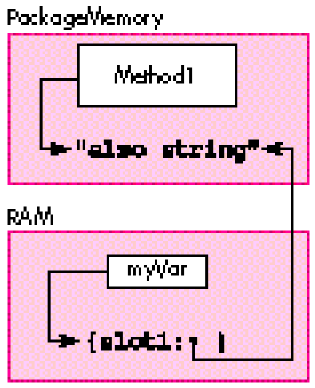
```
   return '{slot2: "also also string"};
end;

Method3 := func()
begin
   return "a string";
end;
```

The braces specify a frame constructor. Each time you call Method1 it will return a reference to
newly allocated frame, though not different contents. For example, when the following is execu

```
myVar := call Method1() with ();
```

here’s what you get in memory:



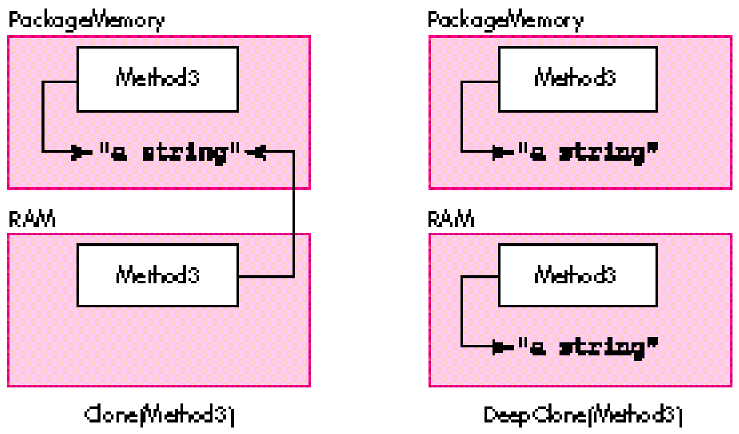
On the other hand, Method2 quotes the frame, which makes it a quoted constant. In other word
each time you call Method2 it will return a reference to the same frame. And Method3 does
something else altogether: In the Newton Toolkit, a string is treated like a quoted frame (or arra
It’s a constant object, so each time you call Method3 it will return a reference to the same string
Note that this means that in both Method1 and Method2 the slot in each frame will reference the
same string. Diagrams that show what happens in memory when each of these three methods is
executed are provided on this issue’s CD along with this column.

Observation 3: When you call SetValue, you’re actually copying the reference to the empty strin
from your buttonClickScript into the text slot of the entry line. You might think this would caus
error, because the string constant can’t be modified. But clParagraphViews are smart: if the stri
can’t be modified (that is, if it’s read-only), a copy is made.

Observation 4: I checked in the inspector, and your buttonClickScript is not read-only. This me
that the string constant "" in that script is also not read-only.

Observation 5: To prevent the grip of death on a card, you would need to call EnsureInternal on
your formula roll entry. This effectively makes a copy of the entire template, including constan

the NewtonScript heap. The following illustration contrasts a Clone with a DeepClone (which i
what EnsureInternal uses). Note that the DeepClone creates a new read/write copy of the string



Conclusion: You press the Clear Data button once. This sets the reference of the input line strin
point to the string constant in your buttonClickScript. Since the string constant is no longer reac
only, changing the input line string destructively modifies the string constant. You may think th
this would lead to a bus error or worse, but thanks to NewtonScript, it works as it should. The
next time you press the Clear Data button, the input line string reference gets replaced with a
reference to the now modified string constant.

The solution is to change the SetValue call to

```
SetValue(dataItem.entryLine, 'text, Clone(""));
```

This will make a copy of the string constant and return a reference to the copy.

Q *Just recently I came into possession of a sword. It was handed to me by a lady in a lake whose
was clad in the finest shimmering samite. I figure with this sign of divine providence I should t
to wield supreme executive power. What do you think?*

A Fortunately, strange women in ponds has not been used as a basis of a system of government s
the Dark Ages. These days supreme executive power derives from a mandate from the masses,
from a farcical aquatic ceremony. If I claimed to be President just because some aquatic gymnas
threw a sword at me, I'd be locked up for sure.

The llama is
the unofficial mascot of the Developer Technical Support group in
Apple's Personal Interactive Electronics (PIE) division. Send your
Newton-related questions to NewtonMail DRLLAMA or AppleLink
DR.LLAMA. The first time we use a question from you, we'll send
you a T-shirt.

Thanks

to Don Gummow and our PIE Partners for the questions used in this column, and to jXopher Bell, Bob Ebert, Mike Engber, Kent Sandvik, Jim Schram, and Maurice Sharp for the answers.

Have more questions?

Need more answers? Take a look at PIE Developer Info on AppleLink.