StewPot - The Soup Inspector - v1.0b8

This application may be distributed freely, so long as this file ("StewPot Read Me") and the manual ("StewPot Manual") accompany it.

The author, nor MAUi Software, **cannot** be held responsible for any loss while using this package. This is a very powerful Newton soup utility, and **can be used to modify or delete data** in your Newton or the PCMCIA cards you place in it.

Please be sure you back up your current information before attempting to make any changes!

What is a soup, anyway?

Everything you place in your Newton is stored in a "soup". A soup is different from both a flat file (or document), and more flexible that a database, which is why Apple came up with a different term to describe it.

But like a database, a soup has separate **entries**, which can be thought of as records with similar data structure. Take the "Names" soup, for example. Each entry corresponds to a single card and its contents. Each field on the card corresponds to an "element" of the "data structure" of the "record", and its parallel on the Newton is called a "slot". You know that some of the fields on some of the cards aren't filled in, so the Newt doesn't even both to store it--that means, for some entries, some of the slots aren't present.

Unlike personal computers, soups aren't tied to a directory/folder-like structure. This makes it easier in some ways, harder in others.

It's **easier** in that the end-user doesn't have to worry about whether they have a card in their Newt or not in order to jot something down--if a card's not there, the information goes into the soups stored in the Newton itself. If a card is present (and the option to add new items to the card is checked), the information is placed in the soups of the card. During reference, the Newt views the collection of soups in the

Newt and the collection of soups on the currently inserted card as if they were one area.

It’s **harder** in the fact that if the name (or calendar reminder, etc.) isn’t in the Newt soups or the current card, there’s no way for it to know where it is (figuratively speaking) and ask for the card it’s on! For example, let’s say you want to send a mail message to someone, using the MailSaver program (because you want to check for new mail at the same time). The MailSaver program is on one card, your business addresses on another. Unless you put **both** on the same card, you can’t do it!

Whoa! You mean that's why I keep having those messages "Newton still needs the card you removed"!

Well, yes, and that's why if you swap cards you might not find everything you were looking for when you do a find.

Well, why would Apple do such a crazy thing?

Depends on your point of view. If you don't have a lot of information to store, or don't swap cards, this sort of arrangement makes a lot of sense, since you can forget about where things are going. But as more and more accessories become available, you need the ability to place your data where you need it on demand.

Some of the built-in applications, such as the Names and Notes, have a command under the routing slip that allows you to move an item from one storage location (call a "store" in Newt parlance) to the other, but not all. In addition, it would be nice to know, at a glance, where things are.

That's where StewPot comes in.

So what can you do with StewPot?

You can:

• Copy entries in a soup -- or copy the entire soup to a new soup

• Move entries from one store to another (Newt to card, card to Newt) - or move the entire soup from one store to another

• See what public/shareware apps add to your Newt (and remove the stuff if they don't clean up after themselves!)

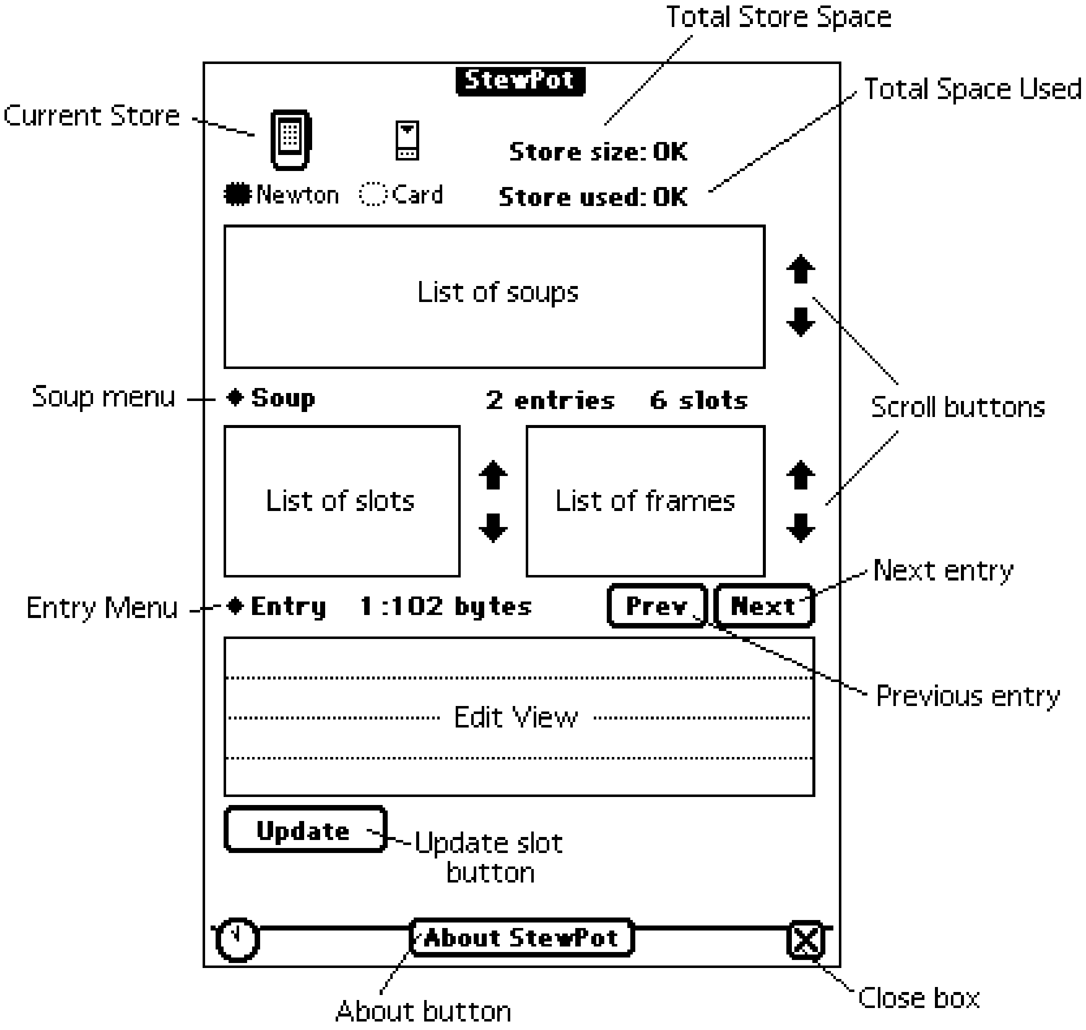
• Create new slots, entries, and soups of your own

• Modify the contents of first and second level entry slots

• Remove a single entry from a soup, or all of them

• Remove a soup entirely from a store

Figure 1 is a snapshot of the StewPot display.



Store size: OK

Store used: OK

2 entries 6 slots

Entry 1 :102 bytes

Figure 1: The StewPot Display

When you start StewPot, it defaults the current store to the Newt’s internal soups, then checks for the presence/absence of a PCMCIA storage card. If one is present, and unlocked, the icon next to the one for the Newton looks like this: If it’s present and the card is locked, it looks like this: If there’s no card inserted, the icon will look like this:

Tapping the “Newton” or “Card” buttons switches the location of the store being looked at, and the current list of soups in that store will be shown in the list of soups. The store size and store used statistics will be updated.

Selecting the name of a soup will do three things: (1) selects the first entry of the soup, (2) updates the entry number and size of the entry, and (3) displays the slots of the first entry in the list of slots.

Selecting one of the slots will do one of two things:

• If the item is a unary object (that is, the slot doesn’t contain a frame or array), the value of the slot for this entry is shown in the edit view

• If the item is a frame or array, the items are shown in the list of frames. Tap an item in the frame list, and if it’s unary, the value is shown in the edit view.

To move through the entries of a soup (if there are more than one), use the “Next” and “Prev” buttons.

Okay, so what else is new? I’ve seen this stuff before!

Where StewPot differs from other soup utilities is in providing direct access to soup, entry, and slot functions, through the use of the “Soup” and “Entry” menus.

The Soup Menu

Note: In all cases, if the operation doesn’t make sense to do if you have a locked card, or no card is present, then it won’t be performed.

Copy

Prompts for the name of a new soup to be created on the opposite store, and **copies** all of the selected soup’s entries into the new soup.

Create

Prompts for the name of a soup, and its primary index, separated by a blank. StewPot makes sure the soup doesn’t already exist in the currently selected store, then creates it.

Move

Moves the entire soup from the current store to the other one (i.e., if the soup is on the card, it’s copied to the Newt and then removed from the card.

Remove

Deletes the soup (and all its entries) from the current store.

The Entry Menu

Note: In all cases, if the operation doesn't make sense to do if you have a locked card, or no card is present, then it won't be performed.

Add Slot

Creates a new slot with the specified name for the currently selected entry **only**. Initially, the new slot's value is nil.

Copy

Copies the current entry and moves the view to point to the copy. The new entry is normally placed at the end of all the entries, and is given a unique id.

Create

Prompts for the names of slots to be placed into a new entry. All slots of the new entry are set to nil.

Find

Prompts for a string to search for, and either moves the view to that entry, or returns a message stating the string was not found.

GoTo

Prompts for an entry number to move the view to. If the entry number is not found, or outside the range of entry values, a message is displayed.

Move

Copies the current entry to the same named soup on the other store, then deletes the entry from the current soup. If the other store doesn't have a soup with the same name as the current soup, or if the entry cannot be created on the other store, nothing is done.

Remove

Deletes the current entry, after a confirmation. The view is moved to the next entry in the soup (if one is present).

Remove All

Deletes all of the entries from the current soup, after a confirmation.

Remove Slot

The currently selected slot is removed from the current entry **only**.

Is that it?

Yep, that’s it.

Well, I could talk about what’s actually in all those soups you see out there, that it, what the information means, but that manual’s already been written--it’s the

Newton Programmer’s Guide that comes with NTK. Suffice to say, here are some general guidelines about possible consequences of messing in soups:

- **Don’t** mess with the System soup entries whose “tag” slot contains “userConfiguration”, “Calibration”, “LetterWeights”, “UserDictionary”, and “Packages”. These contain your Newton system and handwriting preferences. Remember, other applications may store preferences, etc. as separate entries in the System soup.

- **Don’t** delete the Calendar, Calendar Notes, Directory, InBox, Names, Notes, OutBox, Repeat Meetings, Repeat Notes, System, or To do soups on the Newton’s internal store. This will cause the built-in applications to mess up in a big way.

- **Don’t** add new slots to any of the above soups, as they won’t show up under the applications. You **can** remove slots that exist, but they will be added back if you store things in them later from the application.