2.0 Compatibility Test Plan

The purpose of compatibility testing on the Newton 2.0 Operating System is to find problems that occur on this platform, but not on a 1.x Newton. It is assumed that full testing and bug fixing has already occurred on the 1.x platform.

The following sample test plan uses the CheckBook application, since many of you are already familiar with it. The test plan is somewhat vague so that it can be used as a starting place for testing almost any application.

Passing all of the list of tests provided here does *not* guarantee compatibility with the Newton 2.0 Operating System. Compatibility is the responsibility of the developer and requires complete testing of the application.

Scope of Testing

Complete coverage of the application is required to verify compatibility with 2.0. This includes full functional testing, stress testing, error testing, and boundary testing. The only way to know for sure if everything works is to do complete testing of the application.

Priorities

Although full testing of Checkbook will occur on 2.0, the highest priority is to test areas that are likely to be effected by running on 2.0. The high priority areas are as follows:

<u>Common Problem Areas</u>
Soup & Stores
Install Application
Remove Application
Printing
Faxing
Beaming
Mailing
(Communications, if your application has communications functionality, it would be included here.)
2.0 Friendly

<u>2.0 Features</u>
Make Backdrop
Package Operations
Soup Operations
New Recognizer
Ink Text
Upgrading From 1.x
Communicating With 1.x

If your application relies on undocumented slots, functions, methods or behaviors, these areas should also be included in the list of high priority areas for compatibility testing.

IMPORTANT: If your application interacts with any other built-in applications or features, these are also high priority areas for testing. For example, if your application adds a slot to the Names soup, you should install and use your application, then verify that Names still works correctly.

High Priority Compatibility Test Cases

Be sure that you don't just *do* the test but that you also verify that it worked correctly. For example, if you the test is to beam an item from 1.x to 2.0, make sure that the item on 2.0 is really there and is correct.

Soups & Stores
(Cold reboot the Newton and erase the card between these tests)
with application in internal memory
o create data in internal memory; no card inserted
o delete data
o move data to card
o create data in internal memory; blank card inserted
o delete data
o move data to card
o create data on card
o delete data
o move data to internal
with application on card
o create data in internal memory
o delete data
o move data to card
o create data on card
o delete data
o move data to internal

Install Application
(Use both Newton Package Installer (NPI) and Newton Backup Utility (NBU) to test downloading your application.)
o Download application using NBU from any supported Windows machine
o Download application using NBU from any supported Macintosh
o Download application using NPI from any supported Windows machine
o Download application using NPI from any supported Macintosh
o Download application to internal memory
o Download application to card
o Insert card that contains application
o Insert 1.x card that contains application; mount read-only
o Insert 1.x card that contains application; convert
o Change backdrop application (something besides Notepad); install application

Remove Application
o Scrub application in Extras Drawer.
o Select application in Extras Drawer. Tap Delete in Routing menu.
o Remove card that contains application.

Printing
o print one page
o print multiple pages

- print preview (all pages)
- print w/ each Format
- print w/ cover page
- print w/o cover page
- print now
- print now w/o a printer connected
- print later
    - open item in Outbox
    - preview all pages in Outbox (use Next button)
    - Send to printer from Outbox
    - Delete item from Outbox
    - Duplicate item in Outbox
- print to a serial printer
- print to a network printer
- test any application-specific printing functionality
- print all views that can be printed

Faxing

- fax one page
- fax multiple pages
- fax w/each Format
- fax w/ cover page
- fax w/o cover page
- fax now
- fax later
    - open item in Outbox
    - preview all pages in Outbox (use Next button)
    - Send from Outbox
    - Delete item from Outbox
    - Duplicate item in Outbox
- fax, fine resolution
- fax, not fine resolution
- fax, manually connect
- fax, not manually connect

Beaming

- beam one check, open beam in InBox; Put away
- beam multiple items if the application supports it

Mailing

- Mail one check, Show Text, mail now
- Mail one check, mail later, preview in Outbox, send

Communications

(Since Checkbook does not have communications features, there are no test cases listed here. If your application does have communications features, be sure to test them. In addition to testing the basic communications functionality, also test error conditions, such as lost connections.)

Make Backdrop

o Select Checkbook in Extras Drawer, select “Make Backdrop”

(At a minimum, test the major functionality of the application when it’s the backdrop.)

- o New check, fill it out
- o Scroll up, scroll down
- o Overview, scroll through all checks
- o Delete a Check
- o Duplicate a Check
- o File A Check

o Select some other application and make it the backdrop; launch CheckBook.

Package Operations (Select Checkbook in Extras Drawer, then…)

o File icon on card, launch application
o File icon on internal, launch application
o File icon into a different folder (e..g Setup), launch application
o Beam Icon, put away, launch application
o Mail icon, launch application
o Delete icon

Soup Operations

o Verify that soup name is understandable to user.
o Tap icon. Will # of items make sense to the user?
o Tap icon. Delete. Launch application.

Select soup icon then…

- o File on card (should get slip saying it can’t be filed)
- o File on internal (should get slip saying it can’t be filed)
- o Make Backdrop (should get slip saying it’s the wrong kind of icon)
- o Delete; Launch application.

New Recognizer

o Enter text with “Printed (separate letters)” selected
o Enter text with “Cursive (connected letters)” selected

Ink Text

(Ink Text is not supported by CheckBook. If your application allows the user to enter ink text, verify that it is handled correctly.)

Upgrading From 1.x

o On 1.x: Install application on card; create data on card. On 2.0: Mount card read-only; verify that application launches and data is there.
o On 1.x: Install application on card; create data on card. On 2.0, convert card; verify that application launches and data is there.
o On 1.x: Install application in internal memory; create data in internal memory; backup to a card. On 2.0: Mount card read-only; restore application and data; verify that application launches and data is there.

- On 1.x: Install application in internal memory; create data in internal memory; backup to a card. On 2.0: Convert card; restore application and data; verify that application launches and data is there.
- On 1.x: Install application in internal memory; create data in internal memory; backup with NBU. On 2.0: Restore application and data; verify that application launches and data is there.

If your application changes other application soups you should also launch and use the effected application. For example, if your application adds a slot to the Names soup, you should verify that Names works correctly after each “1.x Upgrade” process.

Communicating With 1.x

- Beam 1.x to 2.0. Put away.
- Beam 2.0 to 1.x. Put away.
- E-Mail data 1.x to 2.0.
- E-Mail data 2.0 to 1.x.
- Fax 1.x to 2.0.

2.0 Friendly

Install application; create a new check and fill it in; duplicate a check; delete a check (use applications’s major functionality), then…

- Open Names
- Open Dates
- Open Extras
- Open To Do
- Open Preferences
- Open any other 3rd Party application
- Find everywhere
- Assist (e.g. “Call Anderson”)
- Write something in the NotePad using recognition
- Install application on a 1.x Newton; create a new check and fill it in; duplicate a check; delete a check (use applications’s major functionality), backup 1.x Newton; then…
- Restore 1.x backup to 2.0 Newton. Then…
    - Open Names
    - Open Dates
    - Open Extras
    - Open To Do
    - Open Preferences
    - Open any other 3rd Party application
    - Find everywhere
    - Assist (e.g. “Call Anderson”)
    - Write something in the NotePad using recognition
- Backup application and associated data using NBU on any supported Windows machine from a 2.0 unit and restore to a different 2.0 unit, are the data and preferences restored properly?

o Backup application and associated data using NBU on any supported Macintosh from a 2.0 unit and restore to a different 2.0 unit, are the data and preferences restored properly?

Functionality Testing

The Checkbook application doesn't have much functionality that wasn't already covered in the High Priority Test Cases. The areas that still need to be covered are:

New
Navigation
    CheckView -> Overview
    Overview ->Checkview
    Scrolling
Opening/Closing
Folder Tab
Clock Glance
Intelligent Assistant
Find
Undo

This section is completely application dependent. Specific test cases are not provided because they will be different for every application. All functionality of the application should be tested on 2.0.

Stress Testing

Stress testing is also very application specific. The following is a list of suggestions for stress testing.

- Low memory

  Low storage space

  - Almost full internal store
  - Almost full card

  Locked card.

  - Application on locked card
  - Application data on locked card
  - Application & data on locked card
  - Application and data in internal memory; locked card inserted
- Test with other applications installed
- Boundary conditions. Test with the maximum and minimum amounts of data, the highest and lowest numbers allowed, and any other boundary conditions that might apply to your application.
- Error conditions. Test data that is not allowed by your application. Try to do things that are not supposed to work, and make sure the application fails gracefully.