



Table of Contents

1. Test Information2

 1.1. Overview.....2

 1.2. Software Tests (Required).....3

 1.3. Software Tests (Recommended).....7

 1.4. Mechanical Tests (Required)8

 1.5. Mechanical Tests (Recommended)9

 1.6. Electrical Tests (Required)10

 1.7. Electrical Test (Recommended)13

2. History14

List of Acronyms	
API	Application Program Interface
CS	Chip Select
OE	Output Enable
WE	Write Enable

1. Test Information

1.1. Overview

The following are some test tips and a guide to follow during your pre-test effort.

Springboard modules use Handspring's proprietary software APIs to extend the Palm OS®. Software that runs on a Springboard expansion module and uses calls to Handspring's APIs must undergo Springboard software compatibility testing, even if a pre-approved expansion module is used to host the software.

All tests are required unless specified as "Recommended". This means that it is highly recommended that the Developer adhere to these tests, but the results have no bearing on the overall Springboard compatibility.

Modules must support Visor™ handhelds.

- Visor™ Deluxe
- Visor™ Platinum
- Visor™ Prism
- Visor™ Edge
- Any handheld devices supporting the Springboard Expansion Slot.

1.2. Software Tests (Required)

Please note: Any test criteria not met in this section will also require a retest of the electrical test criteria from the subsequent section.

1.2.1 The Springboard™ header exists and is valid.

The test will verify that no errors, crashes, or resets occur when the module is inserted. Visible module applications must launch properly when selected from the Launcher screen in List and Icon View.

1.2.2 The module name and manufacturer name are registered and unique.

The test will verify that the module name is unique and registered with Handspring.

1.2.3 All databases, preferences, or other variable data are written to Card Number 0.

The test will verify all databases, preferences, or other variable data are written to a valid Palm OS card number for the module. The application saves data properly and no crashes, resets, data loss or corruption occurs.

1.2.4 The hsCardAttrCsBase attribute of the HsCardAttrGet() function is used properly.

The test will verify that the hsCardAttrCsBase attribute is not hard coded into the application.

1.2.5 All creator IDs are registered and unique.

The test will verify that the creator ID associated with the application is unique and registered with Palm, Inc. QP will verify this information using the list of creator ID's from Palm™ Developer Support at:

<http://www.palmos.com/dev/tech/palmos/creatorid/>

1.2.6 Verify the application uses Springboard APIs as documented.

Handspring, Inc. requires applications to use the standard publicly documented Springboard API calls. API implementation **must** be consistent with the intent of the documentation.

1.2.7 The Springboard module supports true hot plug and play.

The test will verify that once a module is inserted, all software is available to the user. If the application requires system extensions, patches, shared libraries, interrupt handlers, background threads, etc., the Springboard must contain a Module Setup Application. The creator ID for the Setup Application must be "HsCd". The test will also verify that no crashes, errors, or soft resets occur.

1.2.8 All auto-loading software, e.g. Welcome application, runs correctly upon insertion of the module.

The test will verify that modules containing auto-loading software automatically launches upon insertion of the module. The application(s) should function as expected without crashing or posting any unwanted errors. If the handheld is powered off, the test will also verify that inserting the module automatically powers on the handheld.

1.2.9 The module installs and functions normally when inserted into a handheld that is on.

This test will verify that inserting the Springboard module into a handheld that is powered on does not adversely affect the module or the handheld itself. Once attached, both the handheld and the module should continue function properly.

1.2.10 All elements of Special-Purpose Applications load, execute, and remove properly.

The test will verify that any system extensions, patches, shared libraries, interrupt handlers, background threads, etc. loaded by the module are properly executed. The test will also verify that these elements are properly removed from the handheld internal memory when the module is removed, unless the application is specifically designed to remain on the handheld, e.g. Mail client applications installed by modem modules.

1.2.11 The module properly handles removal when the application is in use.

The Springboard module will be removed while data is being modified or added to the application. In this scenario, the test will verify that the application exits to the Launcher. The module will then be re-inserted. The test will verify that the application launches and continues to function properly. The test will also verify that no data loss, corruption, crashes, errors, or soft resets occur. “Non-application” code must use the HsCardErrTry/HsCardErrCatch macros upon removal.

1.2.12 Communication modules should perform gracefully when the connection is lost and removed then reinserted.

If your module is a communication type device, e.g. pager, modem, phone, etc., this test will verify that the module performs gracefully and displays appropriate error messages when the connection to the service or receiver is lost. The module and the handheld should continue to function properly once the connection has been interrupted. The test will also verify that communications resume properly once the connection has been reinstated.

1.2.13 The module does not interfere with the proper operation of the handheld.

The test will verify that any system extensions, patches, shared libraries, interrupt handlers, background threads, etc. or the module itself does not affect the functionality of the handheld or its native applications while inserted or when removed, e.g. Digitizer, responsiveness to taps, Date Book +, beaming, Graffiti, Keyboard, Find, etc.

1.2.14 The application releases any customizations to the handheld when switching applications.

Once connectivity and usage between the handheld and the Springboard module has been established, the test will verify that any custom button mappings, color tables, port redirects, etc. implemented by the application are released and returned to their native state when switching to another application. An IR and HotSync operation will be performed to ensure both ports are released and successful data transfer occurs. Exception: If a button is assigned to launch another application, then the button is not expected to be released.

1.2.15 The application supports contrast control or restores the contrast setting upon exit.

The test will verify when the Contrast Control silkscreen button is activated, a Contrast Control dialog appears and all levels of contrast function as expected. The test will be performed while in the Springboard application. The test will also change contrast levels in a native application then verify the contrast levels are maintained inside the Springboard application and vice-versa. If the contrast control is not accessible from within the Springboard application, then the last user designated contrast setting must be restored upon exiting the application.

1.2.16 If supported, verify that application appropriately handles global find.

If the module application supports Global Find, the test will verify that user-input data can be found within the application. Whole words and beginnings of words will be used. The same find operations will be executed from outside and within the module application. Data must be found within your own application and the native applications. The test will also verify that no crashes, errors, or resets occur during a Find or from launching the module application from the Find results listings.

1.2.17 The module allows the handheld to post alarms, warnings, system sounds and messages.

The test will verify that when in use, the Springboard module does not inhibit alarms, system sounds and messages. The test includes the following scenarios while working with the Springboard module:

- A low battery condition to ensure the correct message posts.
- Option selection sounds (i.e. pen down on button) in menu options and pull down list.
- Tapping on a data entry field before entering Graffiti strokes.
- Date Book and/or Date Book+ alarms

1.2.18 The application handles invalid data during data entry appropriately.

The test will verify that invalid data entry in text fields is handled properly. Upon immediate exit of the field containing the invalid data, the application must either post an error message, a system beep or immediately ignore the data entered. The application must not permit the invalid data to be “saved” or cause a crash. If email is supported, the test will also verify that any invalid data in mail fields (i.e. To:, Subject:) will post an appropriate error message. All invalid data will be tested, including and not limited to invalid usernames, passwords, and characters.

1.2.19 The module functions properly during a soft and hard reset.

With the Springboard module inserted in the handheld, the test will verify that the Springboard module performs gracefully during a soft and hard reset. The application should continue to function properly upon a soft reset. The Springboard module should reinstall upon a hard reset then function as expected.

1.2.20 The module performs gracefully in a limited memory condition.

The test will force a storage full and/or limited memory condition on the handheld then verify that an error message posts when new data is entered. The Handheld and Springboard module is to perform gracefully and continue to function once memory has been freed up.

1.2.21 The application does not exhibit any signs of memory leaks.

The test will execute a cursory test pass through every feature in the application using a debug ROM. The test will be performed at first insertion of the module as well as on a subsequent insertion.

1.2.22 Verify compatibility across all OS revisions and handheld devices that support the Springboard Expansion Slot.

Unless otherwise documented in the product documentation, we assume the latest Handspring Palm OS® revisions and handheld devices that support the Springboard Expansion Slot are supported. The test will verify that the Springboard module is compatible with all supported Visor models, as well as operating systems, and most other Springboard Capable models. A 80/20% rule is applied whereby 80% of the testing will be performed on the most current/popular Visor model(s) supported and the remaining 20% on the other supported devices. The test will also verify that an appropriate message posts when the module is inserted and launched on an unsupported configuration.

1.2.23 Verify the module does not interfere with other 3rd party Springboard modules.

This test will verify that your module does not interfere with, prevent proper functionality of, or have adverse effects on other “Springboard Compatible” 3rd party modules. The 3rd party module(s) will be inserted after your module has been activated and removed. The test will verify that the 3rd party application launches and functions as expected. The test will also verify that no data loss, corruption, crashes, errors, or resets occur. The module(s) chosen will be at the discretion of the test lab. After compatibility testing has completed, if it is found that your module interferes with another module in the future, you may need to provide a patch for users of your module.

1.2.24 Verify the module supports data input from a 3rd party Visor keyboard.

This test is applicable if the Springboard module supports data entry. This test will verify that data entered via a 3rd party Visor keyboard product is accepted and transfers properly into the application.

1.2.25 Documentation must be accurate and consistent in functionality with the module.

The test will use the Table of Contents from the User's Manual as a guide or a high level checklist for testing. All major functions of the module (hardware and software) will be tested against the documentation for consistency and accuracy. Not all contents of the documentation will be tested and the test will not include combination testing. Help screens within the application are also considered part of the documentation and must be accurate and consistent with the functionality of the application.

1.2.26 Documentation bitmaps or screen shots must be accurate and consistent with the module.

The test will use the User's Manual as a guide to ensure the screen shots are consistent with the hardware and software. If the documentation is not the final version, the Developer must inform QP.

1.2.27 Documentation must include modifications to native functionality of the handheld.

Per the design of the application, the test will use the User's Manual as a guide to ensure that all modifications to the normal functionality of the handheld are documented. For example, if the buttons are reassigned to do something specific within the application like game controls.

1.2.28 Verify contact information is provided in the User Documentation.

The test will verify that contact and/or technical support information (phone number, email address, and/or web page) is provided with the User Documentation.

1.3. Software Tests (Recommended)

1.3.1 The application name and icon displays properly from the Launcher.

The test will verify that the name and icon for all applications contained on the Springboard module (including the Welcome application) display properly from the Launcher screen in Icon and List modes. The icon should be unique to the application (not the default Palm OS icon).

1.3.2 If color and/or grayscale is implemented, the screen content is readable on handhelds that support color and grayscale.

The test will verify that colorized and grayscale screen content is readable on handhelds that support color, black and white, and grayscale. Care should be taken to avoid visually confusing control selection, list scrolling, and text highlighting. All elements should provide good contrast for readability. The test will also verify that all images of various supported resolutions (if applicable) and active windows are drawn, refreshed, and displayed properly.

1.3.3 Verify that application doesn't change or obscure behavior of silk-screened icons.

The test will verify that all dialogs and screens in the module application do not go below the area of the silk-screened icons (Applications, Menu, Calculator, Find). The UI must fit within the viewing area. The application must not override the Launcher silk-screened icon at any time. A user of the must be able to select the Launcher silk-screened icon while the application is deployed.

1.4. Mechanical Tests (Required)

1.4.1 The module does not place any undue pressure or harm to the case of the handheld.

This test will verify there is no undue pressure or element exhibited against the Handheld such that the handheld will be marred, scratched, or broken by attaching or removing the Springboard module from the Handheld.

1.4.2 Only a reasonable amount of insertion and extraction force is required to insert or remove the module.

This test will manually verify an excessive amount of human force is not needed to insert or extract the module from all of the supported handheld models.

1.4.3 The Springboard module should not exhibit any dangerously sharp edges.

1.4.4 The Springboard module must include the “keying” features.

The correct “keying” features (slots) must be present on both the left and right sides of the module. The test will also verify that the keying features are sturdy, and do not warp, split, or bend with a reasonable amount of pressure added.

1.4.5 The Springboard module must meet the Module Inspection tolerances.

Download the Module Inspection specifications at:

<http://www.handspring.com/developers/Devkit2/ModuleInspection.zip>

1.5. Mechanical Tests (Recommended)

1.5.6 The Springboard module should not intrude or obscure access physically on the cradle, battery-door, IR port, stylus holder, hard & silk-screen buttons, or display screen.

1.5.7 The Springboard module should not contain parts that are easily broken off with a reasonable amount of force.

1.6. Electrical Tests (Required)

Please note: Any test criteria not met in this section will also require a retest of the software test criteria from the preceding Software section.

1.6.1 Under no circumstances should the module drain the batteries and/or backup capacitor to such a degree that the handheld computer cannot maintain the integrity of its internal RAM.

1.6.2 The module and its application fail gracefully in the event of sudden power loss to the module.

This test is applicable for modules that require a power supply or batteries. The power supply or batteries will be removed while the module is inserted, activated, and the application launched. The test will verify that the module does not cause any data corruption or adverse affects to the handheld due to sudden power loss. The test will also verify that the module is able to recover properly and continues to function properly once adequate power has been restored.

1.6.3 Battery modules provide user feedback on battery levels.

This test is only applicable for battery modules. The test will verify that a mechanism is provided to give the user feedback on the battery level, e.g. icon indicator, low battery message, etc. The test will also verify that the indicator properly reflects the current battery level.

1.6.4 Under low battery conditions on the handheld, module, or both, no data loss or data corruption occurs.

The test will verify that when the handheld, module, or both the handheld and module are simultaneously under a “forced low battery” condition, that no data loss or data corruption occurs. The test will also verify that the Springboard module continues to function properly once adequate power has been restored.

1.6.5 The module must not assert IRQ when LOWBAT* is asserted

The test will verify that the module does not attempt to wake up the Handheld during a low battery condition.

1.6.6 The maximum operating current must not exceed 100 mA at 3.0 – 3.6V.

The test will verify that all types of Springboard modules, including modules that require a power supply or batteries, do not exceed or cause the handheld to exceed 100 mA at 3.0 – 3.6V.

1.6.7 The maximum current to the module VDOCK pin must not exceed 500mA at 4.75 – 6.2V.

The test will verify that all types of Springboard modules, including modules that do not use VDOCK, do not exceed or cause the handheld to exceed 500 mA at 4.75 – 6.2V.

1.6.8 If the module provides its own charging supply, the module must not inhibit the handheld from charging.

The test will verify that both the module and handheld device can properly charge simultaneously. The module must allow the VDOCK specification of 500mA at 4.75 – 6.2V to be maintained to the module and that the charging circuit within the handheld receives a current of 1.5A at 4.75 – 6.2V. The test will also verify that the module does not cause damage or adverse effects to the handheld device while charging.

1.6.9 Verify the module meets the following Read Cycle specifications:

In the Springboard Read Cycle timing specification, the “CPU expecting data after CSx* asserted” should not be longer than 300ns as specified. The Springboard Read Cycle timing can be found in the “Springboard Development Guide” under Electrical Specifications.

<http://www.handspring.com/developers/documentation.jhtml>

Commentary: The remaining timing specifications are driven by the Handspring device based on a 50pF (maximum) load and is not configured by the module. These timing specifications are generally considered during the module design phase to ensure that selected components are compatible with Springboard timing. By contrast, the module developer programs the “CPU expecting data after CSx* asserted” time specified above through several mechanisms. First, the Access Time ROM token in the Springboard header is read and used to program the access time when the module is inserted. Second, the API HsCardAttrSet() can be used with the hsCardAttrAccessTime selector to dynamically set the access time. Note that when the module is first inserted the system will automatically configure the access time to the longest possible duration (based on the CPU capabilities). Once the Springboard header is read, the access time is adjusted accordingly.

1.6.10 Verify the module meets the following Write Cycle specifications:

In the Springboard Write Cycle timing specification, the “CSx* pulse width” should not be longer than 300ns as specified. The Springboard Write Cycle timing can be found in the “Springboard Development Guide” under Electrical Specifications.

<http://www.handspring.com/developers/documentation.jhtml>

Commentary: The remaining timing specifications are driven by the Handspring device based on a 50pF (maximum) load and is not configured by the module. These timing specifications are generally considered during the module design phase to ensure that selected components are compatible with Springboard timing. By contrast, the module developer programs the “CSx* pulse width” specified above through several mechanisms. First, the Access Time ROM token in the Springboard header is read and used to program the access time when the module is inserted. Second, the API HsCardAttrSet() can be used with the hsCardAttrAccessTime selector to dynamically set the access time. Note that when the module is first inserted the system will automatically configure the access time to the longest possible duration (based on the CPU capabilities). Once the Springboard header is read, the access time is adjusted accordingly.

1.6.11 Upon insertion, the module must not prevent power ramp up time and a minimum 25 ms power-on reset time.

When the Card Detect* pins are asserted, VCC slowly ramps up. The Springboard specification calls for a minimum of 25ms where Reset* is asserted and VCC is fully ramped up. Details of Reset Timing can be found in the “Springboard Development Guide” under Electrical Specifications.

<http://www.handspring.com/developers/documentation.jhtml>

The test will be performed at first insertion of the module as well as on a subsequent insertion.

1.7. Electrical Test (Recommended)

1.7.1 The standby current in VCC when LOWBAT* is asserted should be as low as possible and between 10 - 100 μ A.

Note: This test may be Required in the future.

1.7.2 The module's current draw while the handheld is in sleep mode will be reported.

2. History

Date	Revision #	Description of changes
15 June 2001	1.01	Correction: Section 1.6.9 and 1.6.10 have been modified. The “CPU expecting data after CSx* asserted” (read cycle) and “CSx* pulse width” (write cycle) are the only valid tests for Read and Write Cycle timing verification.
		Correction: Section 1.2.1 has been modified. The Springboard header must exist.
		Correction: Section 1.2.2 has been modified. Both the module name and the manufacture name must be registered and unique.
		Clarification: Section 1.6.11 has clarifies Reset Timing behavior.
16 Apr 2001	1.0	Initial release.

Handspring™, Visor™, Springboard™, and the Handspring and Springboard logos are trademarks or registered trademarks of Handspring, Inc. © 2001 Handspring, Inc.