

USER'S GUIDE

NEWTONID

VERSION 2.0

FOUNDATION SYSTEMS

© 1997-1998 Foundation Systems. All Rights Reserved.
PO Box 2330
Stanford, CA 94309-2330
<http://www.tow.com/>

Every effort has been made to ensure that the information presented in this manual is accurate. Foundation Systems is not responsible for printing or clerical errors.

Apple, Newton, MessagePad, eMate, the Light bulb logo, and Macintosh are registered trademarks of Apple Computer, Inc., registered in the U.S. and other countries.

Microsoft is a trademark, and Windows is a registered trademark of Microsoft Corp.

Mention of third-party products is for informational purposes and only constitutes neither an endorsement nor a recommendation. Foundation Systems assumes no responsibility with regard to the performance or use of these products.

Contents

1	Getting Started.....	2
2	Creating Your NewtonID	5
3	Registering Products	7
4	Your Registration Codes	8
5	Redoing Your Registration.....	9
6	Frequently Asked Questions	10
Appendix A	NewtonID API.....	12

1. Getting Started

About NewtonID

NewtonID is a framework proposed by Foundation Systems to help simplify and standardize software registration on the Newton.

Newton software developers and Newton users are at a crossroads concerning software registration services.

Developers are looking for a better solution to implementing software registration. Till now, many developers have been relying on a user's owner card to generate and verify software registration codes. And some developers are moving to adopt a hardware serial number-based registration scheme.

Users are looking for a more unobtrusive and hassle-free way to register their applications. Many users are unhappy that their applications are tied to particular owner cards on their Newton. These applications disable themselves if the improper owner card is active—this problem is evident with users having multiple owner cards corresponding to multiple email accounts. Other users are unhappy with hardware serial number registration schemes since this involves a different registration code for each Newton they may own.

NewtonID changes all of this by providing a universal identifier for every Newton user and a flexible architecture for developers. NewtonID avoids the traditional problems associated with multiple owner cards and software registration by providing a standard for identification purposes.

- Universal owner name—No more hassles dealing with software that can't handle multiple owner cards.
- Hardware Serial Number—Displays the serial number of the Newton device, if present.
- Flexibility—NewtonID lets developers choose what kind of registration scheme to use.
- Repository for registration codes—Developers can save their registration codes within NewtonID so users can have convenient access to their software registration codes.

Types of NewtonID Registration

Developers who write NewtonID-Aware applications have three ways with which to generate registration codes:

- NewtonID Owner Name—This uses the name that the user has specified when she first installed NewtonID. This approach to software registration is virtually identical to what many software developers are currently using. This approach does not use the hardware serial number in any way to perform software registration.
- Hardware Serial Number—This uses the Newton's built-in hardware serial number. An application that uses this approach will only work on one Newton. Consequently, users will have to get a new registration code for each Newton they own. With this approach, however, the NewtonID Owner Name can be variable.
- NewtonID Owner Name Hardware Serial Number—This uses both the NewtonID Owner Name and the Newton's hardware serial number for registering software. This approach ties a piece of software not only to the Newton's unique hardware serial number but also to the NewtonID Owner Name of the Newton.

Software developers should make it clear to users what type of registration scheme they are using with their NewtonID-Aware applications. For instance, currently all of Foundation Systems' products make use of the NewtonID Owner Name to generate registration codes.

System Requirements

To use NewtonID on your Newton device, you need the following:

- a Newton device running the Newton 2.0 operating system or later.
- 15 kilobytes of storage space.

Note: There is a version of NewtonID for Newtons running the Newton 1.x operating system. You may find it on Foundation Systems website at: <http://www.tow.com/> The instructions in the proceeding chapters assume that the user is using the version of NewtonID for Newton devices running the Newton 2.x operating system.


Installation

If NewtonID is not already installed on your Newton device, you need to install it into your Newton's memory or onto a PC card. To do this, you use a Newton utility that can install software packages, such as the Newton Connection Utilities or Newton Backup Utility, with a computer running the Mac OS or Windows operating system. Once installed, the NewtonID is automatically stored in the Unfiled folder in the Extras Drawer.

If you are installing NewtonID for the first time, please refer to Chapter 2, "Creating Your NewtonID" for instructions on generating your personal NewtonID.

Opening NewtonID

- 1 Tap Extras.

Note: If you don't see the Extras Drawer icon, it means Extras is the backdrop, the default background application. Go to the Extras Drawer by tapping every  you see. For information about the backdrop application, refer to the documentation that came with your Newton device.

- 2 If you don't see the NewtonID icon, tap the folder tab and tap Unfiled Icons.



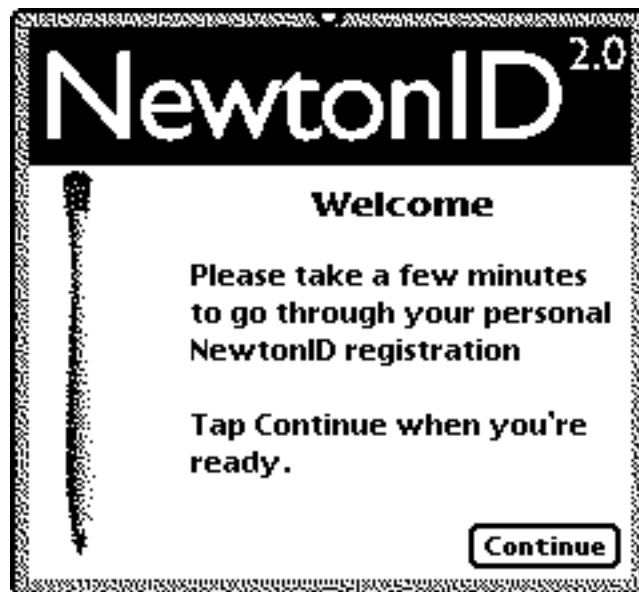
- 3 Tap the NewtonID icon **NewtonID**.

Note: If the NewtonID icon doesn't appear, tap Extras, tap the folder tab, and in the list that appears, tap All Icons. Then tap the NewtonID icon

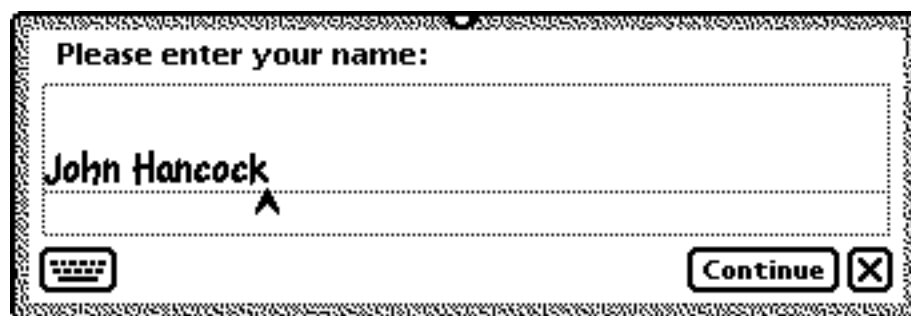
- 4 When you are finished using NewtonID, tap  to quit.

2. Creating Your NewtonID

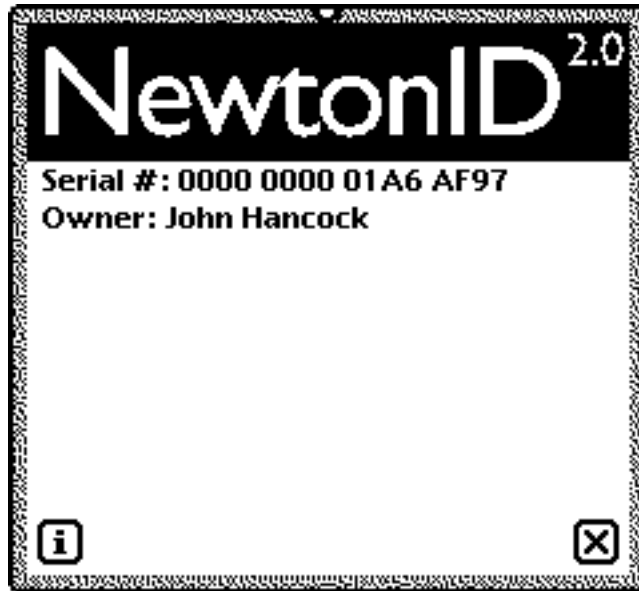
- 1 If NewtonID is being installed for the first time, a dialog will appear, prompting you to create your personal NewtonID.



- 2 Tap Continue. The following dialog appears.



- 3 Enter your name and tap Continue.
- 4 A slip will appear, asking you to confirm the NewtonID Owner Name you just entered.
- 5 Tap OK.




Congratulations! You've just established your NewtonID for your Newton. Your NewtonID contains important information when registering NewtonID-Aware applications:

- **NewtonID Owner Name:** "John Hancock" This name is often used to generate the registration code for NewtonID-Aware applications. It may be used in conjunction with your Newton's Hardware Serial Number (if present) to generate registration codes as well.
- **Hardware Serial Number:** "0000 0000 01A6 AF97" Every Newton beginning with the MessagePad 2000 and the eMate 300 have a hardware serial number assigned to it. The HW Serial Number may be used by some NewtonID-Aware programs to generate registration codes and may also be used in conjunction with your NewtonID Owner Name to generate codes.

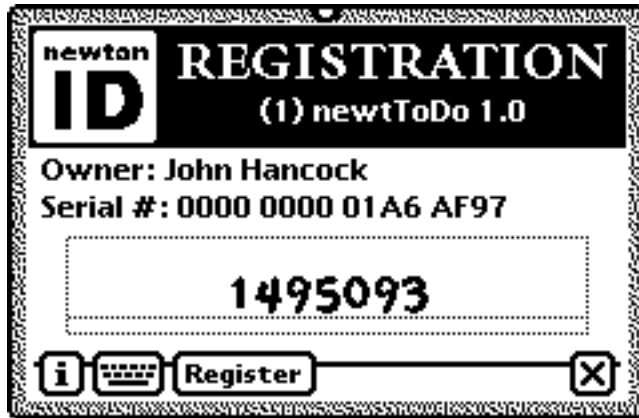
3. Registering Products

To register NewtonID-Aware applications, follow these steps:

- 1 Open the application to be registered.
- 2 Tap the Information button  and in the list that appears, tap Register.




- 3 The NewtonID Registration View appears. Enter the registration code provided to you by software publisher in the input view.



- 4 Tap Register.
- 5 A slip will appear, telling you if the registration code you entered was valid or invalid.

Note: If the registration code is invalid, verify that the NewtonID Owner Name you sent to the software publisher is the same as the name displayed in the NewtonID Registration View.


- 6 Tap  to close the NewtonID Registration View.
- 7 Depending on the application being registered, you may have to open and close the app or restart your Newton for the application to properly register itself.

Note: Other NewtonID-Aware applications may have different procedures for registering than the one listed here. Please refer to the documentation that came with your application for additional information and instructions on registering.

4. Your Registration Codes



Whenever you register a NewtonID-Aware application, it will store its registration code on your Newton. You can view these registration codes with NewtonID, and you also have them exported as a note to the Notepad application

Viewing Your Registration Codes

- 1 Open NewtonID.
- 2 The NewtonID view will appear. Any registration codes you have will be displayed in the window below your NewtonID Owner Name.
- 3 Tap  when you are done.




Exporting Your Registration Codes

- 1 Open NewtonID.
- 2 Tap the Information button .
- 3 In the list that appears, tap Export registration codes.
- 4 A slip will appear, telling you that the registration codes have been exported to the Notepad.
- 5 Tap  to close NewtonID.

There will be a new Notepad note listing the registration codes for your NewtonID-Aware applications.

5. Redoing Your Registration

You can change your NewtonID by erasing your current NewtonID information and creating a new NewtonID. Any registration codes you may have for NewtonID-aware products will be erased along with your old NewtonID information.

- 1 Open NewtonID.
- 2 Tap the Information button .
- 3 In the list that appears, tap Redo registration.
- 4 A slip will appear, confirming your decision to redo your NewtonID registration.
- 5 Tap OK.

Your original NewtonID information, along with any registration codes you have will be deleted. NewtonID will now begin the installation process again.

- 6 Follow the steps outlined in Chapter 2, “Creating Your NewtonID”.

6. Frequently Asked Questions

This section describes common questions and problems that you may encounter when using NewtonID and NewtonID-Aware products. If, after reviewing the suggested solutions and answers, you still cannot solve the problem or remain confused, contact Foundation Systems at: support@tow.com.

Can I delete the NewtonID application after I have installed it?

Yes. After you have installed and created your NewtonID, you can do any of the following to the NewtonID application:

- Keep it—Keep the NewtonID application installed on your system.
- Delete—Delete the NewtonID application by scrubbing its icon away in Extras.
- Freeze—Freeze the NewtonID application. Select NewtonID in Extras, Tap the Routing button and in the list that appears, tap Freeze.

Note: You may have to install a third-party application such as StandAlone's Freeze Utilities or ICS's NewtCase Utilities to activate freezing on your Newton device.

I don't like the idea of using the Hardware Serial Number to generate my registration codes!

- NewtonID does not perform the registration of software. NewtonID simply provides a Universal Owner Name and displays the hardware serial number of your Newton device.
- Software companies that produce NewtonID-Aware applications determine what kind of registration scheme their products will use to generate registration codes:
 1. NewtonID Owner Name only.
 2. Hardware Serial Number only.
 3. NewtonID Owner Name and Hardware Serial Number.
- Currently, all applications produced by Foundation Systems make use of just the NewtonID Owner Name for generating registration numbers.

Why not use the Owner Info data to generate registration codes? NewtonID seems to be a superfluous solution!

NewtonID provides a solution to the common problem of relying on Owner Cards to generate registration codes:

- Universal Owner Card—There's only one "owner name" to worry about.
- Standard—More and more developers are switching to NewtonID to handle part of their software registration scheme.
- Switching between owner cards won't cause NewtonID-Aware applications to become expired, as many third-party applications that rely on a specific owner name do.

Appendix A: NewtonID API

This section describes the NewtonID Application Programming Interface. You can use the information provided here to make your new and existing programs into NewtonID-Aware applications.

Types of NewtonID Registration

There are three ways you can use NewtonID to help you handle software registrations. Registration codes can be based on three pieces of information provided by NewtonID.

- 1 NewtonID Owner Name—Use the name registered with NewtonID (i.e. "John Doe")
- 2 Hardware Serial Number—Use the hardware serial number of the Newton (i.e. "0000 0000 A9F4 67BC")
- 3 NewtonID Owner Name and Hardware Serial Number—Use both the NewtonID Owner Name and Hardware Serial Number to generate registration codes.

Constants used by NewtonID

The following are constants used by the NewtonID API. The constant `kNewtonIDAppSymbol` is used to identify the NewtonID system soup entry. The constant `kRegistrationType` is used by your application to identify what type of NewtonID Registration is being used (see above). The constants `kRejectMsg` and `kAcceptMsg` are used by the function `kVerificationFunc` to alert the user when a valid or invalid registration code is inputted.

```
constant kNewtonIDAppSymbol := '|NewtonID:NewtonID|';
constant kRegistrationType := 1;
constant kAcceptMsg := "Registration code accepted!";
constant kRejectMsg := "Registration code invalid! Please try again.";
```

kGetNewtonIDFunc

call `kGetNewtonIDFunc` with `()`

Returns a frame containing the NewtonID Owner Name and the Hardware Serial Number from the Newton, or `nil` if NewtonID has not been installed on the Newton. This function returns a frame with the following slots:

Slot description

<code>name</code>	A string that is the NewtonID Owner Name.
<code>serialnumber</code>	A string that is the string representation of the Hardware Serial Number of the Newton device, or <code>nil</code> if the Newton device does not have a serial number. This includes any Newton prior to the MessagePad 2000 and the eMate 300.

```
DefConst('kGetNewtonIDFunc, func() begin
  local newtonID := GetAppPrefs(kNewtonIDAppSymbol, {});

  if newtonID.name then begin
    local serialnumber;

    try serialnumber := call ROM_GetSerialNumber with ()
    onexception |evt.ex| do serialnumber := nil;

    if serialnumber then serialnumber := StrHexDump(serialnumber, 2);

    return { name: newtonID.name, serialnumber: serialnumber };
  end;
  else return nil;
end);
```

Important

The following functions, `kMyPasswordFunc` and `kMyVerificationFunc`, are sample functions used to illustrate how you would incorporate NewtonID into an application. They are by no means meant to be the sole functions used in implementing an application that is NewtonID-Aware, but they do act as excellent starting points.

kMyPasswordFunc

call `kMyPasswordFunc` with (name, serialnumber, regcode, regtype)

Verifies a registration code, given the NewtonID Owner Name, the Hardware Serial Number, and the type of registration being handled.

You may implement this function in any way you find most convenient: NewtonScript, compiled NewtonScript, or C++. You may modify this function prototype as much as you want to as it is here just to show you an easy way to implement a verification function. You may want to have different registration algorithms for different values of `regtype`.

This function should return `true` if the registration code is valid, `nil` otherwise.

<code>name</code>	A string that is the NewtonID Owner Name.
<code>serialnumber</code>	A string that is the string representation of the Hardware Serial Number of the Newton device, or <code>nil</code> if the Newton device does not have a serial number. This includes any Newton prior to the MessagePad 2000 and the eMate 300.
<code>regcode</code>	An integer or a string that is the registration code inputted by the user.
<code>regtype</code>	An integer from 1 to 3 specifying the type of NewtonID Registration is being performed: (1) NewtonID Owner Name, (2) Hardware Serial Number, or (3) Both (1) and (2).

```
DefConst('kMyPasswordFunc, func(name, serialnumber, regcode, regtype) begin
    if regtype = 1 then begin
        // Level 1—NewtonID Owner Name.
        // Continue with your verification code, returning
        // true if the regcode is valid.

    end;
    else if regtype = 2 then begin
        // Level 2—Hardware Serial Number.
        // Continue with your verification code, returning
        // true if the regcode is valid.

    end;
    else if regtype = 3 then begin
        // Level 3—NewtonID Owner Name and HW Serial Number.
        // Continue with your verification code, returning
        // true if the regcode is valid.

    end;
    return nil;
end);
```

kMyVerificationFunc

call `kMyVerificationFunc` with `(code, registering)`

Verifies a given registration code. Developers can use this function as a wrapper function for `kMyPasswordFunc`. It first verifies that NewtonID has been installed on the Newton. If so, the function determines whether we are verifying an existing registration code or if the user is trying to register the program based on the value of `registering`. We pass the registration code, along with the NewtonID Owner Name, the Serial Number, and the registration type to the function `kMyPasswordFunc`.

If the password is valid, and if we are registering the application, we save the user's registration code within the NewtonID soup entry under the application symbol of the app being registered.

This function should return `'registered` if the registration code is valid, `nil` if the registration code is invalid, or `'noNewtonID` if the NewtonID information is not installed.

`code` An integer or string that is the registration code.

`registering` A Boolean, whether or not we are verifying a registration code or trying to register an application. If the `true`, we take the value of `code` and pass it to `kMyPasswordFunc`. Otherwise, we take the registration code from the NewtonID soup entry, or `nil` if it doesn't exist, and pass it to `kMyPasswordFunc` for registration code verification.

```
DefConst('kMyVerificationFunc , func(code, registering) begin
    local NewtonID := call kGetNewtonIDFunc with ();
    local NewtonIDSoupEntry := GetAppPrefs(kNewtonIDAppSymbol, {});
    local regCode, serialnumber;

    if NewtonID then begin
        if registering then regCode := code;
        else regCode := NewtonIDSoupEntry.(kAppSymbol);

        if call kMyPasswordFunc with (NewtonID.name,
            NewtonID.serialnumber, regcode,
            kRegistrationType) then begin
            if registering then begin
                NewtonIDSoupEntry.(kAppSymbol) := regcode;
                EntryFlushXmit(NewtonIDSoupEntry, nil);
                GetRoot():Notify(kNotifyAlert, kAppName,
                    kAcceptMsg);
            end;
            return 'registered;
        end;
        else begin
            if registering then
                GetRoot():Notify(kNotifyAlert, kAppName, kRejectMsg);
            return nil;
        end;
    end;
    else return 'noNewtonID;
end);
```


NewtonID Soup Format

This section describes the format of the NewtonID system soup entry. The NewtonID soup entry is referenced under the symbol, '|NewtonID:NewtonID|' and can be accessed with the following line of code:

```
local newtonID := GetAppPrefs('|NewtonID:NewtonID|', {});
```

Slot description

name A string that is the NewtonID Owner Name

In addition, applications may save their registration codes within the NewtonID soup entry. These registration codes will be stored under the appSymbols of the applications for which they were registered. For instance, a sample NewtonID soup entry might look like the following:

```
{
    name: "John Hancock",
    |WriteHere:ATOW|: 12345678        // Registration code for WriteHere
    |HyperNewt:ATOW|: 45678901       // Registration code for HyperNewt
    tag: "NewtonID:NewtonID",        // Reserved. Do not modify.
    _uniqueID: 10,                    // Reserved. Do not modify.
    _modTime: 49439289,              // Reserved. Do not modify.
}
```