Newland AIDC
SCANNING MADE SIMPLE

NEWLAND

WEB KIOSK

Software

Version: V2.21

User Guide

Disclaimer

Please read the manual carefully before using the product and operate it according to the manual. It is advised that you keep this manual for future reference.

Do not disassemble the device or remove the seal label from the device; doing so will void the product warranty provided by Newland Europe BV.

All pictures in this manual are for reference only, and the actual product may differ.

Regarding product modification and update, Newland Europe BV reserves the right to make changes to any software or hardware to improve reliability, function, or design at any time without notice. The information contained herein is subject to change without prior notice.

The products depicted in this manual may include software copyrighted by Newland Europe BV or a third party. The user, corporation or individual shall not duplicate, in whole or in part, distribute, modify, decompile, disassemble, decode, reverse engineer, rent, transfer, or sublicense such software without prior written consent from the copyright holders.

This manual is copyrighted. No part of this publication may be reproduced, distributed, or used in any form without Newland Europe BV's written permission.

Risk Warning Regarding Unauthorized System Updates:
You should use the Newland-provided tool to update this product's system. Modifying system files by installing a third-party ROM system or using any cracking method may result in product malfunction or data loss and void your warranty.

Newland Europe BV reserves the right to make a final interpretation of the statement above.

Newland Europe BV

Rolweg 25, 4104 AV, Culemborg,
The Netherlands
www.newland-id.com

Newland Europe BV is a subsidiary of Newland Digital Technology Co., Ltd. Our general conditions of Purchase, Sale and Delivery are filed with the Record Office of the Chamber of Commerce of Utrecht, The Netherlands.

K.v.K. H.R. Utrecht / Chamber of Commerce Utrecht: Reg. nr. 17109876

Contents

Introduction

The Newland Web Kiosk is an application that can be used in conjunction with an Ndevor Desktop Profile and provides a full screen protected view of a web application or website. The application supports running multiple web applications at the same time by creating multiple URL shortcuts within the Desktop Profile. Target devices for the Newland Web Kiosk include Newland Android based Mobile Computers, Newland Nquire devices and Nwear mobile computer devices.

Starting with release 2.0 of the Newland Web Kiosk, web applications can take advantage of the integrated scanning API. Through the API, the web application can control the scanner that is integrated into the Newland device. In addition, the data received from the scanner can be captured using the API as well, leading to a controlled way of receiving the scanned data.

The scanning API also includes functions to support scanners connected via Bluetooth to the mobile device via Newland's NHUB application.

Starting with version 2.08 of the Newland Web Kiosk, in addition to the Scanning API, an API to be able to read NFC tags can be used.

In this document you will find all the necessary information to set up Newland's web-kiosk and the full description of the scanning API.

Installation

Add the Newland Web Kiosk APK to your Ndevor instance by using the Application/Application Management screen.

Enable the application in an App Profile that is applicable to your device(s).

Make sure the admin tool is installed on your devices, by enabling "Admin Tool" in your Configuration Profile, Ninstaller tab. Then create or modify your Desktop Profile. In the URL, prefix http or https with "newland" so that in total, the URL will look like:

The “newland” prefix will make the Newland Web Kiosk launch instead of the standard browser (usually Chrome).

Do NOT add the Newland Web Kiosk to the Application section of the Desktop Profile! Only add URL Shortcuts!

Configuration Options

The Web Kiosk can be configured by adding options at the end of the URL. Options can be combined by concatenating them.

Options	***Function:***
:ALLOWZOOM	Will allow the user to use zoom gestures and buttons on the web session. Tapping the HOME button will reset the zoom to its initial state.
:NOBUTTONS	The use of the on-screen buttons is disabled.
:IGNSSLERR	Ignore SSL errors (invalid certificate on the website) Use this option with caution especially when visiting 3rd party websites.
:ALLOWDWN	Will allow the user to download and open files from a web page, i.e. pdf files. The file will be opened with whichever application is associated with the filetype downloaded.
:ALLOWCPP	Will allow the user to use copy/paste logic within input fields. Selection can be made by double-tapping in the text to be selected, or by single tapping on the cursor “tear” as in the example below.  Important note: When the user types in a website url in a text field, and then uses the copy-paste logic, he will be offered the option to open that URL, and the standard browser Chrome will be opened with that URL. The Web-Kiosk is not capable of preventing that from happening, as this is standard Android behavior.
:ALLOWPULL	The Web Kiosk will allow the user to refresh the current page by “swipe-down” gesture from the top of the page. A spinner will be displayed while the page is being refreshed. This option can be used independently from :NOBUTTONS, however, it can also be used in combination, so that the user will still have a way to refresh the page, even when the button bar is disabled.

:AUTRFRx	The Web Kiosk will automatically refresh the current page after x minutes of inactivity. Example :AUTRFR1 will refresh the currently displayed page when the user has not performed any action on the device for one minute.
:NFCTAB	When an NFC tag was read, while the current webpage has no implementation of the nlNFC API, the Web Kiosk will enter the NFC data in the current field and will submit a TAB key to the application, to proceed to next input field.
:NFCENTER	When an NFC tag was read, while the current webpage has no implementation of the nlNFC API, the Web Kiosk will enter the NFC data in the current field and will try to find and execute a submit button on the current page. If no submit button could be found, the Web Kiosk will submit an ENTER button to try to submit the page/form.
:AUTOFOCUS	When a webpage is loaded, that has no fields that have the autofocus tag, the AUTOFOCUS parameter will cause the Web Kiosk to automatically set focus on the first input field found on the page. This feature is useful in applications where no scanning API was implemented in the host application. The feature will allow the user to immediately input data via keyboard, scanner or NFC, without having to tap the cursor into the first field.

Example URL:

newlandhttps://www.newland-id.com:ALLOWZOOM:NOBUTTONS

Note:

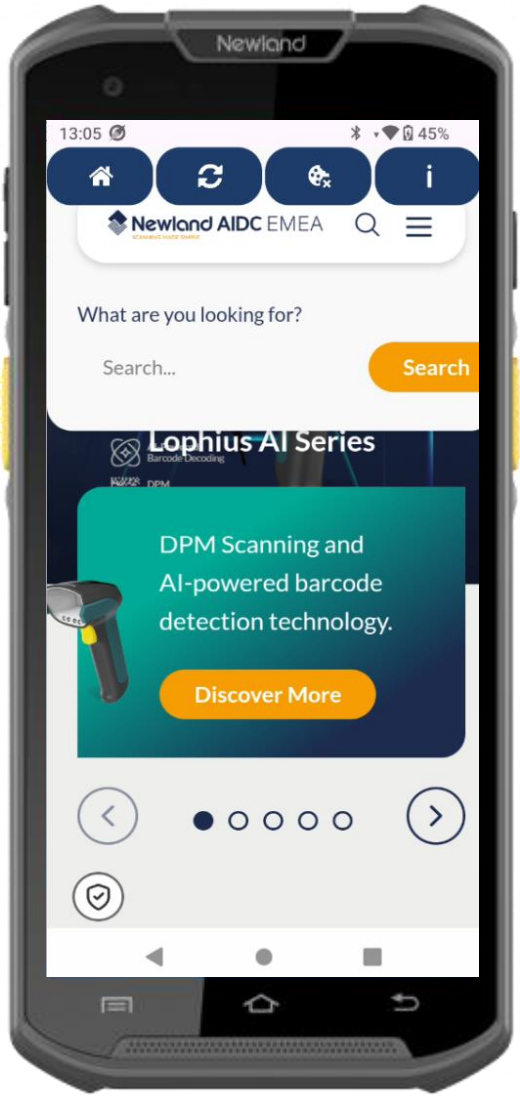
From Web Kiosk version 2.18, the options are case-insensitive.

Use

When the Newland Web Kiosk is running, it will show the website in full screen.

A button bar can be launched by swiping from the left edge of the screen to the right, unless it is disabled with the :NOBUTTONS option.

The button bar will appear in the top of the screen for two seconds during which the buttons can be tapped.



Returns the user to the initial URL and resets zoom

Refreshes the currently displayed URL

Clears cookies and browser cache, then executes home function(s)

Information. Shows application version information and Newland disclaimer.

Scanning & NFC API

Release 2.0 of the Newland Web Kiosk integrates an API that allows web applications to control the scanner and to receive the scanned data in a controlled manner. The integration consists of script functions to control the scanner, that can be called directly from the web application and a script function to receive the scanned data. This function must be implemented in the web-host application and will be called from the Web Kiosk when a barcode is scanned. The scanner needs to be configured to output its data via the API output method and all the corresponding Broadcast Output settings must be configured correctly to allow the scanning API in the Web Kiosk to work. A function is available that will set up the scanner settings correctly.

Receiving Scanned or NFC Data

When the scanner has been used to scan a barcode, the Web Kiosk will call back the nlBarcode function which needs to be present in the script section of the host's web page. When a NFC tag is scanned, the Web Kiosk will call back nlNFC. Each page on the web application needs to contain these functions to execute the appropriate actions for that page to process the scanned data. When working with NHub and Bluetooth scanners, nlNHubResult is called to pass results from NHub commands.

NFC "Wedge"

In case of NFC tags read, when the Web Kiosk does not find a handler for the NFC data (nlNFC function) in the currently displayed webpage, the Web Kiosk will insert the NFC tag data in the field on the webpage that currently has focus. The Web Kiosk will add a TAB or ENTER character to the NFC data, if parameters :NFCTAB or :NFCENTER were provided. (See configuration parameters overview earlier in this user guide.

nlBarcode

nlBarcode (barcode, type, typename, status)		
Parameter	*Type*	*Description*
Source	String	INTERNAL: From internal scanner XX:XX:XX:XX:XX:XX: Mac address of Bluetooth scanner connected via NHub
Barcode	String	Scanned barcode in ASCII format.
Type	String	Symbology ID number. Possible values are listed in the appendix in the Newland Android Mobile Computer API Handbook.
Typename	String	Symbology name, i.e. CODE128. Possible values are listed in the appendix mentioned above.
Status	String	"ok" after a successful scan, or "fail" after a scanning timeout or otherwise failed scan operation. "Fail" will only happen when the scanner was initialized with the option "Send output on failed scan attempt". Not used when receiving from NHub connected scanners.

nlNHubResult

nlNHubRestult (result, status, device)		
Parameter	*Type*	*Description*
result	String	Result of NHub command
status	Integer	Status of NHub command result
device	String	BT Mac address of the scanner

nlNFC

nlNFC (source, data)		
Parameter	*Type*	*Description*
source	String	Always "NFC"
Data	String	NFC tag UID, in hexadecimal format

Controlling and configuring the scanner

Functions are provided to control the scanning process.

Scanner Power

nlScan.Power(scannerPower)		
Controls scanner power. This function can be used to control whether the user can use the scanner or not.		
Parameter	*Type*	*Values*
scannerPower	Integer	0=Off 1=On

Scanner Initialization

nlScan.Initialize(sendFailBroadcast)		
Set the scanner up to work with the scanning API. Puts output mode to API and configures the scanner's broadcast output parameters. It is highly recommended to call this function one time from the web application's landing page.		
Parameter	*Type*	*Values*
senFailBroadcast	Integer	0=Do not send output on fail 1=Send output on failed scan attempt

Scan Mode

nlScan.Scanmode(mode)		
Sets the scanner's output mode. For the API to work correctly, this mode must be set to 3. The scanner Initialization function sets the mode to 3 already. This function can be used to set the scanner back to regular wedge mode upon ending the web application, or in case the web application needs to use pages that do not support Newland's Web Kiosk scanning API.		
Parameter	*Type*	*Values*
Mode	Integer	1=Fill in EditText directly 2=Simulate keystroke 3=Output via API

Trigger

nlScan.Trigger(timeout)		
Triggers the scanner to start scanning.		
Parameter	*Type*	*Values*
Timeout	Integer	Set the scanner's timeout in seconds.

Triggermode

nlScan.Triggermode(mode)		
Sets the mode of triggering the scanner.		
Parameter	*Type*	*Values*
mode	Integer	0=Level mode 1=Continuous mode 2=Pulse mode 4=Delay mode

Timeout

nlScan.Timeout(timeout)		
Sets the timeout of the scanner, after which a scan will fail.		
Parameter	*Type*	*Values*
Timeout	long	Timeout in milliseconds. Default: 3000.

Interval

nlScan.Interval(interval)		
Sets the timeout between decode sessions.		
Parameter	*Type*	*Values*
Timeout	long	Interval in milliseconds. Default: 50.

Repeatinterval (aka re-read delay)

nlScan.Repeatinterval(interval)		
For continuous scanning, sets the interval in which the same barcode can be scanned again.		
Parameter	*Type*	*Values*
Timeout	long	Interval in milliseconds.

Stop Scan

nlScan.Stop()		
Stops a current scanning action.		
Parameter	*Type*	*Values*
n/a		

Beeper

nlScan.Beep(beep)		
Sets the good-read beeper.		
Parameter	*Type*	*Values*
Beep	Integer	0=Beeper off 1=Beeper on

LED

nlScan.LED(led)		
Sets the good-read LED indicator.		
Parameter	*Type*	*Values*
Beep	Integer	0=LED off 1=LED on

Symbology Configuration

nlScan.BarcodeConfig(CodeID, Property, Value)		
Configures the scanner's symbology settings. For all possible settings please refer to the manual "Newland Android PDA Symbologies Configuration Handbook".		
Parameter	*Type*	*Values*
CodeID	String	Symbology ID in text, i.e. CODE128
Property	String	The parameter to be configured
Value	String	The value to be assigned to the parameter

NHub Send Command

nlScan.NHubSend(Property, Value)		
Sends a command to the NHub Bluetooth Scanner interface. This function allows us to send configuration and other commands to the Bluetooth scanner connected to the device via NHub. Refer to "Programming_Guide_Based_on_Newland_Unified_Commands_Set" for a full overview of scanner command parameters.		
Parameter	*Type*	*Values*
Property	String	The parameter to be configured 'CMD' for sending commands to the BT connected scanner
Value	String	The value of the command to be sent to the BT connected scanner. i.e. "SCNTRG1" to trigger the BT scanner

Device API

The device API provides functions for web applications to obtain device information.

Device Model

model = nlScan.NGetModel()		
Returns the model's name of the device on which the application is running.		
Parameter	*Type*	*Values*
model	String	The device model name

Serial Number

serial = nlScan.NGetSerial()		
Returns the serial number of the device on which the application is running.		
Parameter	*Type*	*Values*
serial	String	Device serial number

Device Build Number

build = nlScan.NGetBuild()		
Returns the firmware build number of the device on which the application is running.		
Parameter	*Type*	*Values*
build	String	The device firmware build number

Device Name

devicename = nlScan.NGetDeviceName()		
Returns the device name of the device on which the application is running.		
Parameter	*Type*	*Values*
devicename	String	Device name

NQuire API

The NQuire API provides functions to control specific functions of the Nquire series products. These APIs will only have effect when running the Web Kiosk on NQuire devices and will be ignored on other devices.

nlScan.nqGPIO

The NQuire API provides function to control the output signal from the GPIO ports. This API will only have effect when running the Web Kiosk on NQuire devices.

nlScan.nqGPIO(port, status)		
Controls the output signal from the GPIO ports on NQuire devices		
Parameter	*Type*	*Values*
port	String	Select the port to be switched “A” or “B”
Status	Boolean	True=On, False=off

nlScan.nqLOGO

The NQuire API provides function to control the output signal from the GPIO ports. This API will only have effect when running the Web Kiosk on NQuire devices.

nlScan.nqLOGO(color)		
Controls the color of the Newland logo LEDs.		
Parameter	*Type*	*Values*
Color	String	RGB (results in white) blue red blue_breathing red_breathing off

Sample HTML code

```
<!DOCTYPE html>
<html>
<head>
<title>Newland Barcode API Demo</title>
</head>
<body>
<h2>Newland Barcode API demo</h2>
<button onclick="nlScan.Power(0)">Disable Scanner</button>
<button onclick="nlScan.Power(1)">Enable Scanner</button>
<button onclick="nlScan.Trigger(2)">Trigger Internal Scanner</button>
<button onclick="nlScan.NHubSend('CMD', 'SCNTRG1')">Trigger BT Scanner</button>
<button onclick="nlScan.Stop()">Stop</button><br>
<br>
<label>Barcode<br><input type="text" id="barcode" /></label> <br><br>
<label>Symbologytype<br><input type="text" id="type" /></label> <br><br>
<label>Symbologyname<br><input type="text" id="typename" /></label> <br><br>
<label>Scanstate<br><input type="text" id="status" /></label> <br><br>
<script>
// This function prevents buttons from gaining focus causing them to repeat itself
document.querySelectorAll('button').forEach(function(button) {
  button.addEventListener('mousedown', function(e) {
    e.preventDefault();
  });
});
// This function will be called from the Web Kiosk when barcode input is received
// For demo we'll fill in the data in the appropriate input fields
function nlBarcode(barcode, barcodetype, barcodetypename, scanstate) {
            document.getElementById('barcode').value = barcode;
            document.getElementById('type').value = barcodetype;
            document.getElementById('typename').value = barcodetypename;
            document.getElementById('status').value = scanstate;
};
// Initialize the scanner settings the way we need
function initScanner() {
  nlScan.Power(1); // Enable scanner
  nlScan.Initialize(1);          // Initialize the scanner
  nlScan.Beep(1);                // Enable good-read beep
  nlScan.LED(1);                 // Enable good-read LED
  nlScan.TriggerMode(0);     // Set trigger mode to "level"
  nlScan.BarcodeConfig('CODE128', 'Minlen', '5');// Set Code128 min length 5
  nlScan.BarcodeConfig('CODE128', 'Maxlen', '25');// Set Code128 max length 25
```

```
  nlScan.BarcodeConfig('QRCODE', 'Enable', '1');// Enable QR code reading
};
// Run automatically when the HTML page finishes loading
window.onload = function() {
  initScanner();
};
</script>
</body>
</html>
```

SCANNING MADE SIMPLE

Newland AIDC EMEA
+31 (0) 345 87 00 33
info@newland-id.com

Rolweg 25
4104 AV Culemborg
The Netherlands