



DATREND
Systems Inc.

vPad-CheckTM

**Automated Maintenance & Inspection App
for vPad Safety Analyzers**

App Version 1.97.X

Operating Manual

vPad-CheckTM
*Automated Maintenance and Inspection App
for vPad Safety Analyzers*
Operating Manual

© 2013 - 2019 Datrend Systems Inc.
Unit 130 - 4020 Viking Way
Richmond, BC • CANADA • V6V 2L4
Tel: 800.667.6557 (North America only) or
+1.604.291.7747 • Fax +1.604.294.2355
e-mail: customerservice@datrend.com



Revision	Revision History Description	Date
D	Update for V1.97+ software	2018-Mar-22
E	A1 Automation Added, new Statements	2019-Aug-22

Copyright

Datrend Systems Inc. ("DSI") agrees to a limited copyright release that allows you to reproduce manuals and other printed materials for use in service training programs and other technical publications. If you would like other reproductions or distributions, submit a written request to Datrend Systems Inc.

Unpacking and Inspection

Follow standard receiving practices upon receipt of goods. Check the shipping carton for damage. If damage is found, stop unpacking. Notify the freight carrier and ask for an agent to be present while the goods are unpacked. There are no special unpacking instructions, but be careful not to damage the goods when unpacking. Inspect the goods for physical damage such as bent or broken parts, dents, or scratches.

Claims

Our routine method of shipment is via common carrier. Upon delivery, if physical damage is found, retain all packing materials in their original condition and contact the carrier immediately to file a claim. If the goods are delivered in good physical condition but does not operate within specifications, or if there are any other problems not caused by shipping damage, please contact your local sales representative or DSI immediately.

Standard Terms and Conditions

Refunds & Credits

Please note only serialized products (products labelled with a distinct serial number) and accessories are eligible for partial refund and/or credit. Non-serialized parts and accessory items (cables, carrying cases, auxiliary modules, etc.) are not eligible for return or refund. In order to receive a partial refund/credit, the product must not have been damaged, and must be returned complete (meaning all manuals, cables, accessories, etc.) within 90 days of original purchase and in "as new" and resalable condition. The *Return Procedure* must be followed.

Software which has been activated may not be returned for credit. Use it in Demo mode where possible to ensure suitability for purpose.

Return Procedure

Every product returned for refund/credit must be accompanied by a Return Material Authorization (RMA) number, obtained from Datrend Customer Service. All items being returned must be sent *prepaid* (freight, duty, brokerage, and taxes) to our factory location.

Restocking Charges

Product returned within 30 days of original purchase are subject to a minimum restocking fee of 15%. Product returned in excess of 30 days after purchase, but prior to 90 days, are subject to a minimum restocking fee of 20%. Additional charges for damage and/or missing parts and accessories will be applied to all returns. Product which is not in "as new" and resalable condition, are not eligible for credit return and will be returned to the customer at their expense.

Certification

This instrument was thoroughly tested and inspected and found to meet DSI's specifications when it was shipped from the factory.

Where applicable, calibration measurements are traceable to the National Research Council of Canada (NRC) and/or the National Institute of Standards and Technology (NIST). Devices for which there are no NRC/NIST calibration standards are measured against in-house performance standards using accepted test procedures.

Warranty

Warranty and Product Support

Software Applications (Apps) for use in conjunction with ES601 and vPad Safety Analyzers and other Datrend Products are supplied subject to the Terms and Conditions agreed to by the act of Activating the App. In the event no Activation is required, the Terms and Conditions published on the Datrend Systems Inc. (DSI) website shall prevail. By using the App, you agree to these terms and Conditions. IN NO EVENT SHALL DATREND SYSTEMS INC. BE LIABLE FOR CONSEQUENTIAL DAMAGES.

DSI DISCLAIMS ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR APPLICATION.

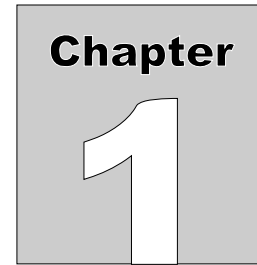
vPad™, vPad-Check™, vPad-ES™, vPad-353™, vPad-623™, vPad-AS™, vPad-A3™, vPad-A6™, vPad-NFPA™, vPad-IN™, vPad-INR™, vPad-RF™, vPad-AIMS™, vPad-Record Manager™, vPad-RM™, vPad-EQM™, vPad-mT™, vPad-Cal™, vPad-XPORT™, Datrend Docs™ and CMX™ are trademarks of Datrend Systems Inc.

Android™ is trademark of Google Inc.

Table of Contents

1	OVERVIEW	1
1.1	Checklists	1
1.1.1	Description	1
1.1.2	Operation	2
1.2	vPad Programming Language	4
2	INSTALLATION	5
3	OPERATION	7
3.1	Startup	7
3.2	Main Menu	8
3.2.1	Checklist Search	9
3.2.2	Update Checklist Files	9
3.2.3	List Editor	10
3.2.4	Settings	13
3.2.4.1	Safety Test Settings	15
3.2.4.2	Security Settings	15
3.2.4.3	Test Record Settings	17
3.2.4.4	Data Input Settings	17
3.2.4.5	Label Printer Settings	17
3.2.5	Hardware Setup for Label Printing	18
3.3	Running a Checklist	21
3.3.1	Equipment Information	22
3.3.2	Prompts	24
3.3.3	Images, Video and Documents	27
3.3.4	Checks	29
3.3.5	Limit Tests	32
3.3.6	Input Requests	33
3.3.7	ECG Tests	34
3.3.8	Electrical Safety Tests and Other Tests	35
3.3.8.1	Automated Safety Tests	35
3.3.8.2	Manual Safety Tests	40
3.3.8.3	ESU Performance Tests	42
3.3.8.4	Patient Monitor Performance Tests	43
3.3.8.5	Test Equipment Database	45
3.3.9	Remote Control	46
3.3.9.1	Remote Device Connection	46
3.3.9.2	Remote Control Steps	49
3.3.10	Optional and Repeatable Steps	52
3.3.11	Checklist Results Preview	53
3.3.12	Creating a Test Record	54
3.3.13	Printing an Inspection Label	60
3.4	Using the Android Keyboard	61

4	V PAD PROGRAMMING LANGUAGE	63
4.1	Checklist Files	63
4.1.1	File Location	63
4.1.2	Filenames	63
4.1.3	File Editing	64
4.2	Checklist Structure	66
4.2.1	Blank Lines	66
4.2.2	Comments	66
4.2.3	Statements	66
4.2.4	Keywords	67
4.2.5	Strings	68
4.2.6	Newline Character Sequence	68
4.2.7	Line Continuation Sequence	69
4.3	prompt Statement	71
4.4	header Statement	73
4.5	show Statement	74
4.6	input Statement	75
4.7	check Statement	76
4.8	limtst Statement	79
4.9	color Statement	82
4.10	power Statement	83
4.11	ecgsim Statement	84
4.12	run Statement	87
4.13	Remote Control Statements	90
4.13.1	xopen Statement	90
4.13.2	xctrl Statement	92
4.13.3	xtest Statement	94
4.14	branch and endbr Statements	95
4.15	begin and loop Statements	99
4.16	IF_ELSE [if_br/else_br/end_if_br] Statements	101
5	UPGRADES AND MAINTENANCE	103



1 Overview

vPad-Check™ is a software application (App) which may be installed on vPad-ES™ and vPad-353™ electrical safety analyzers, and which may be used to automate preventive maintenance (PM) inspections, performance testing and other types of service performed on medical equipment. vPad-Check automation provides these benefits to PM's and similar technical work:

- Reduced job time
- Increased labour efficiency
- Increased consistency of work product - procedures are performed the same way each time, with no missing steps
- Automatic recording of work performed and results obtained - Test Record files provide documented evidence that equipment was tested and was found acceptable
- Traceability through use of supporting apps such as vPad-Cal™ to record ID numbers and calibration status of test equipment used during maintenance work

This chapter provides an overview of vPad-Check and provides an introduction to PM inspection Checklists and the tool used to create them, the vPad Programming Language (vPL).

1.1 Checklists

1.1.1 Description

vPad-Check™ implements procedure automation through **Checklists**. A Checklist may be thought of as an inspection or maintenance procedure broken down into a list of steps which are to be performed in a certain order on the equipment under service (EUS). A step of a Checklist may:

- Display instructions to the technician, ranging from a single line of text to detailed paragraphs
- Display a drawing or photograph of the EUS

- Play a video showing setup or operation of the EUS
- Prompt the technician to inspect an item or feature of the EUS, record details, and assign a "PASS" or "FAIL" to the inspected item
- Prompt the technician to perform a test or a measurement and enter the result, which is then passed or failed automatically according to quantitative criteria included in the Checklist
- Execute an electrical safety test of the EUS automatically
- Execute a sequence of test waveforms automatically (useful for checking patient monitors, defibrillators and other medical equipment incorporating an ECG function)
- Execute performance tests of the EUS automatically, by means of external analyzers and simulators interfaced through serial or wireless connections to the vPad

1.1.2 Operation

To perform a preventive maintenance (PM) inspection or other service procedure on a medical device, the technician begins by selecting the required Checklist from a scrollable\searchable list. In the list, Checklists are typically named according to the type of device being serviced (e.g. anesthesia unit, defibrillator, infusion pump and so on), although it is also possible for there to be Checklists which are specific to particular models (e.g. HP CodeMaster defibrillator), or any other naming convention the user deems appropriate.

Upon selection of a Checklist, the App will scan through the procedure to determine whether or not the Analyzer Base Unit will be needed for inspection or testing of the equipment. If not, the App will start the Checklist immediately without attempting to "**connect**" (i.e., communicate) with the Analyzer Unit. This is convenient for PM inspections of certain devices where no testing of electrical safety or performance is needed; for this type of maintenance work, the vPad Tablet may be removed from the Analyzer and used independently.

If the App determines that the Analyzer will be required to complete a selected Checklist, the App will attempt to connect to the Analyzer before it starts the first step. Depending on Analyzer setup, the connection attempt may be through the tablet's USB Port or alternatively by Bluetooth. Once it has connected with the Analyzer, the App proceeds to the first step of the Checklist. The App then remains connected to the Analyzer for the remainder of the Checklist and for any subsequent Checklists that may be performed, until the App is fully exited to the tablet HOME screen.

After selecting a Checklist and, if necessary, waiting for the App to connect with Analyzer, the technician will then go through the PM procedure one step at a time, pausing to read instructions, review drawings or photographs, inspect items, and enter data as required by the procedure. If the Checklist includes an

electrical safety test, this may be executed automatically by either the vPad-AS™, vPad-NFPAT™, vPad-A3™, or vPad-A6™ software application installed on the tablet. Checklists may also include performance tests, which may be done manually by the technician and then entered into vPad-Check, or automatically by externally-connected test instrumentation. Some Checklists, for example a PM of an anaesthesia machine, may be complex and there is no practical limit as to the number of steps that a Checklist may have.

If necessary, the technician can step backward through the Checklist to repeat certain tests, or to review or re-enter data that was input earlier in the procedure. Steps requiring the technician to perform a pass/fail assessment of an inspection item, or a test or measurement, can be "skipped" by assigning a status of "not applicable". Any procedure step which has been "skipped" or assigned "not applicable" status will be saved and reported as such in the Test Record.

After completing the procedure, a record of the service including all pass/fails, measurements and results will be saved in the vPad filing system. In addition to these results, a Test Record may include equipment information entered by the technician during the procedure, such as make, model, serial number, location and so on. In order to create a Test Record, at minimum an equipment control number or other ID must be entered. Test Record files are named according to this entered ID or control number.

Once the Test Record has been saved, a PM inspection label may then be printed by means of a (optional) label printer connected to the vPad. vPad-Check supports PM label printing with several different printer models. To make use of this feature, the user must have a compatible label printer. Additionally, a Smart Cable adaptor for interfacing the printer to vPad may be required. More information on PM label printing may be found in Sections 3.2.5 and 3.3.13 of this manual. Label printing is enabled in the vPad-Settings app.

1.2 vPad Programming Language

vPad "Checklists" correspond to a set of files which are stored in the vPad-Check directory of the Android tablet. Each Checklist file has a ".**vp1**" file extension and contains a ***script*** which has been written in the vPad Programming Language or "**vPL**".


Each step of a Checklist procedure has a direct one-to-one relationship with a corresponding vPL ***statement*** in the Checklist file. vPL provides statements for displaying instructions, showing photographs and video, accepting data input from the user, running electrical safety tests, interfacing with external test instruments and so on. Although the majority of PM procedures will be written in vPL as a simple list of instructions or tasks to be completed, the language provides for branches and loops which enable technicians to handle different equipment configurations or conditions while an inspection is in progress. vPL is a straightforward language that lends itself to easy creation and update of PM's and other service procedures with minimal learning curve and little or no experience in "computer programming" per se.


Checklist files may be edited with any text editor or word processor. Minor changes to a Checklist may be made using the text editor app installed on the tablet.

Wholesale modification or creation of new Checklists is more efficiently accomplished with a personal computer (PC) and the vPad Integrated Development Environment or **vPad-IDE™**, a software package for Microsoft® Windows® which may be downloaded at no charge from www.datrend.com. When using a PC for Checklist creation or editing, file(s) may be transferred from the PC to the tablet by USB flash drive, WiFi network, or point-to-point Bluetooth. vPad-Check provides an "Update Files" feature for this purpose.

More information on the vPad Programming Language and vPad-IDE may be found in Chapter 4 of this manual.

2 Installation

The vPad-Check App is normally pre-installed on the vPad tablet, but the icon for the App may not have been placed on the Home screen .

Go to the Apps screen by pressing the Apps icon, , which may be found on the favourites task bar at the bottom of the screen, or in the upper right corner of the Home screen (tablet and Android build dependent). Find the vPad-Check icon and press and hold the icon for a few seconds. You will then be able drag and drop the icon onto the Home screen.

In order to use the App it will need to be activated. When the App is started for the first time, the activation screen will be presented. Enter the activation code provided by your dealer to allow the App to run.

In the event that vPad-Check is not on the tablet, contact your dealer and an installation package for the app will be provided via download.

In order to use vPad-Check effectively, you must have at least one software application installed and activated for automated electrical safety testing. The app you choose for electrical safety testing will depend on the standard you prefer or are required to follow:

Electrical Safety Standard	Safety Testing App
AAMI-ES1	vPad-AS™
NFPA 99	vPad-NFPA™
IEC60601-1	vPad-AS™
IEC62353 Edition 1	vPad-A3™
IEC62353 Edition 2	vPad-A6™

3 Operation

This chapter covers operation of the vPad-Check app and Checklist execution.

3.1 Startup


Press the vPad-Check icon, , to start the app. A splash screen will display the product name and the software version for a few seconds.



Figure 1 - vPad-Check startup screen.

Unlike other vPad Apps which are used for electrical safety or performance testing, vPad-Check will *not* attempt to communicate with the Analyzer Base Unit during the initial startup. The App attempts to communicate only when a Checklist chosen by the technician will require use of the Analyzer, or other test devices which are controlled through the Analyzer. This is convenient for preventive maintenance inspections of certain devices where no testing of electrical safety or performance is needed; for this type of maintenance work, the vPad Tablet may be removed from the Analyzer and used independently.

3.2 Main Menu



After start-up, vPad-Check will display the main menu of the app:



Figure 2 - Main menu of vPad-Check.

Some example Checklists are supplied with the app and will appear in the scrollable list as shown above. Example Checklists include PM procedures for typical medical devices as well as some selected models of specific devices. The listing may be sorted in ascending or descending order by selecting the corresponding radio button at upper right on the screen.

To start a test, press and drag the list view up or down until the desired Checklist is found, then tap on the Checklist. Alternatively, you may start a test by entering all or part of a Checklist name in the "Checklist Search" box and pressing **"Search"**. The search is not case sensitive, and will find the closest match to what has been entered. For instance, entering 'be' would take you to 'Bed/Electric' in the sample list. The contents of the search window can be cleared by touching the **"Clear"** or **"Refresh"** button. For other **"Search"** options see the next section.

To return to the Android Home screen, press the  icon on the navigation bar. Note this will only pause the app temporarily and will not actually terminate or close vPad-Check. To fully close the App, press the **'Exit vPad-Check'** button or alternatively the Android "BACK" navigation button  when viewing the main menu.

3.2.1 Checklist Search

To search for a particular Checklist, tap the **'Checklist Search'** box of the main menu (*Figure 2*), enter the first few characters of the Checklist name, then press the **'Search'** button. With the search selection set to **"Starts with"** mode, the List View will move to the nearest matching Checklist, which will then appear highlighted in green. If the full name of a Checklist is entered, and an exact match to the entered name is found, then the Checklist will begin execution automatically. The search is not case sensitive.

"Search" also has the ability to search in a **"Contains"** mode. In this mode, the search will create a smaller list of Checklists which contain the characters entered on the keyboard, and present them in the List View. The list can be further reduced by additional searches.

3.2.2 Update Checklist Files

vPad-Check provides a file-update feature which is intended to facilitate the transfer of Checklist files (files with a **".vpl"** extension) from a personal computer (PC) to the Android tablet. This feature is helpful if you will be using a PC text editor, or the vPad Integrated Development Environment (**vPad-IDE™**), to write your own Checklists (vPad-IDE is free software which you may download from www.datrend.com).

Checklist files may be easily transferred by USB flash drive. Use the file export feature of vPad-IDE to copy files from your PC to the root directory of a USB flash drive, then connect the flash drive to the USB "HOST" port of the tablet. Depending on the tablet model, an adaptor cable provided with the tablet may be required for the flash drive connection (*Figure 3*).

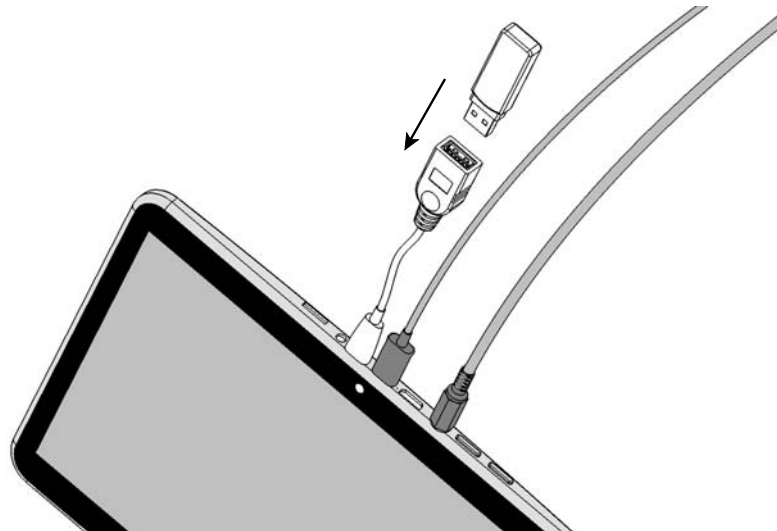


Figure 3 - Typical Connection of USB flash drive to tablet.

After connecting the USB drive, press the **'Update Checklist Files'** button on the main menu of the App (*Figure 2*). vPad-Check will detect the attached USB drive and will then search for ".vp1" files in the root directory of the drive. If any such files are found, they are copied to the vPad-Check directory of the tablet. Files on the tablet which have the same filename as those on the USB drive will be overwritten.

Alternatively, Checklist files may be updated through Bluetooth transfer. If there are no files found on a USB drive when **'Update Checklist Files'** is pressed, then vPad-Check will search for ".vp1" files in **Bluetooth** file transfer folder of the tablet. Through its file export feature, vPad-IDE software enables you to transfer Checklist files from PC to tablet via Bluetooth, should you prefer this method to the USB flash drive method described previously.

An error dialog will be displayed in the event that vPad-Check is unable to find any ".vp1" files in the expected USB or Bluetooth directories:

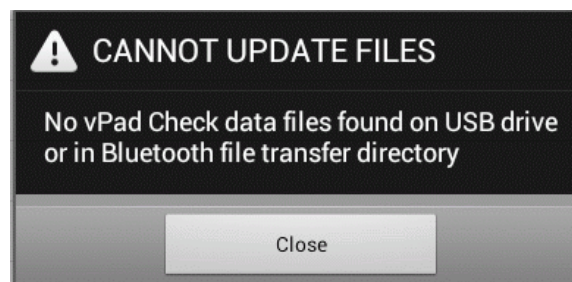


Figure 4 - File update error message.

3.2.3 List Editor

vPad-Check maintains a number of lists which pertain to equipment maintenance and inspection. These lists are used when entering equipment information at the start of a test, or when saving a record of the test results. Lists may be viewed and edited after pressing the **'Edit Lists'** button.

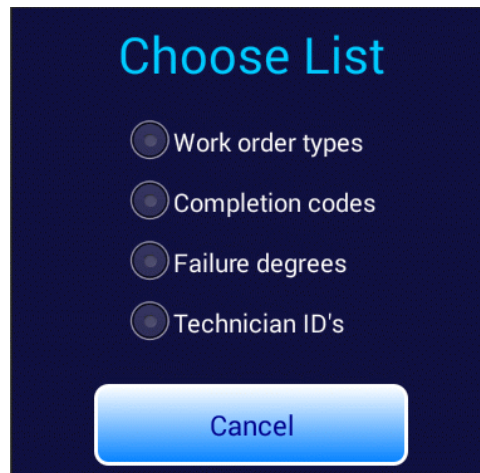


Figure 5 - List selector.

Selecting a list via the associated radiobutton displays the list in an editor window. For example, selecting the **'Technician ID's'** radiobutton displays the editor for technician codes or initials.



Figure 6 - Edit technician list.

To add an item, touch the editor window at the bottom of the list. This will display the soft keyboard. To insert an item in the middle of the list or to edit an existing item, touch the window to place the editor cursor at the desired line. Press **'Save and Exit'** to save edits, or **'Exit'** to return to the main menu without making changes.

Each vPad-Check list is associated with a text file stored in the **Datrend/Database** folder on the tablet. The files and their factory-default contents are:

WorkType.txt	<p>The WorkType.txt file contains codes representing types of work order. Factory default codes are:</p> <p>PM Preventive Maintenance CM Equipment Repair IE Incoming Equipment Inspection EC Environment of Care ED Education EP Elective Procedure LI Liason TC Test code RC Recall (ECRI/SDMA) PJ Project TS Test for AIMS VT Verification Test</p>
TechCode.txt	<p>The TechCode.txt file contains technician IDs or initials. Default technicians are:</p> <p>Tech A Tech B Tech C</p>

ResponseType.txt	<p>The ResponseType.txt file contains codes which may be assigned on completion of a PM inspection or test. Completion codes may be used to automate closure of work orders in certain CMMS systems. Regardless of whether of CMMS is in use or not, completion codes may also be helpful to review of PM outcomes during assessment or audit of the maintenance system. Factory default completion codes include:</p> <pre> INI Incoming Inspection-Pass INF Incoming Inspection-Fail INL Incoming Inspection-Loan INR Incoming Inspection-Repair INE Incoming Inspection-Trial PMP Pm Complete-Pass PMF Pm Complete-Fail PMV Pm Complete-Vendor PMR Pm Complete-With Repair PMU Pm Uncompleted/Could Not Find PMN Pm Uncompleted/In-Use PMO Pm Uncompleted/Other PUV Pm Uncompleted/Vendor ... UPG Upgrade UAB User Abuse USR User Error VEN Vendor Repair Required VIS Visual Inspection VVT Verification Test WOP Waiting on Parts WRP Warranty Repair </pre>
FailureDegree.txt	<p>The FailureDegree.txt file contains degrees or severities of equipment failure in the event a fault is found during a maintenance inspection. These codes may be helpful in developing an AEM program. Factory default failure degrees are:</p> <pre> ---- (no failure) Major Minor </pre>

3.2.4 Settings

Application settings may be configured via the Settings menu (**Figure 7**), which is displayed by pressing

the **Settings** button on the main menu of the App.

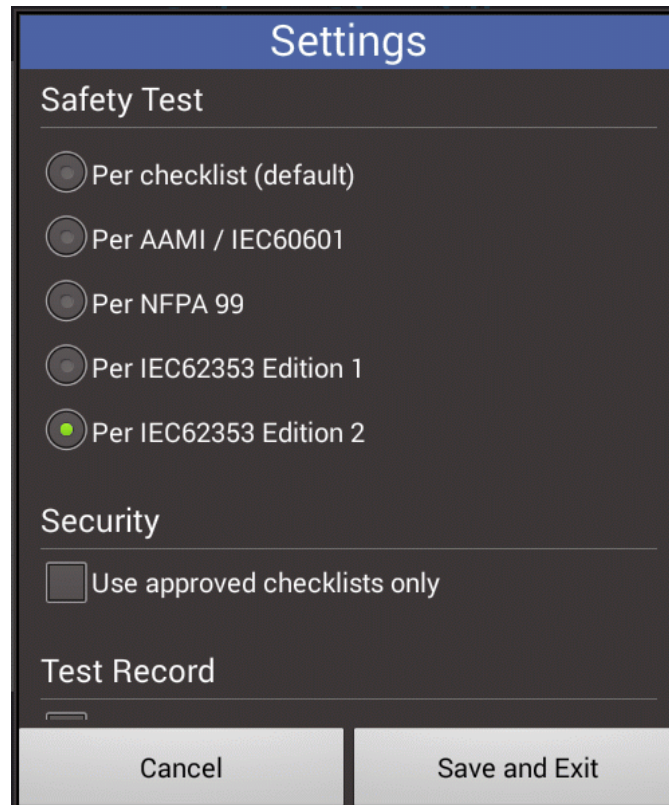


Figure 7 - Settings menu.

The full list of settings may be viewed by touching a blank area on the menu and then dragging the window either up or down to the section of interest. The menu's **Cancel** button dismisses the dialog without making any changes while the **Save and Exit** button saves the displayed settings to permanent memory storage on the tablet. The menu provides settings for safety testing, Checklist security, and Test Record format as described below.

3.2.4.1 Safety Test Settings

A Checklist may include an instruction to execute an electrical safety test by means of an automation app installed on the tablet: vPad-AS™, vPad-NFPA™ or vPad-A3™. The "Safety Test" setting of vPad-Check enables users to optionally override the safety testing app as specified in Checklists with an alternate app to suit configuration of the tablet or user preference.

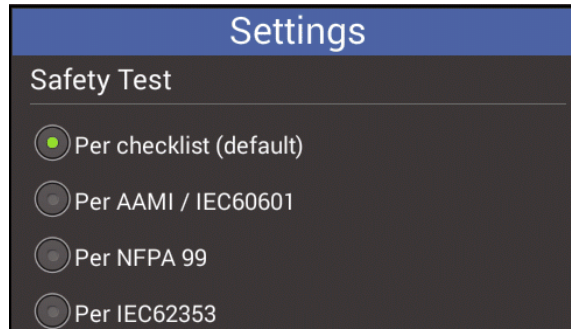


Figure 8 - *Safety test setting.*

The "Safety Test" override is chosen based on a desired testing standard, such as IEC60601. The following table relates the testing standard to the vPad automation app that will be used to perform the test.

Safety Test Setting	Safety Testing App
Per AAMI-ES1 / IEC60601	vPad-AS™
Per NFPA 99	vPad-NFPA™
Per IEC62353 edition 1	vPad-A3™
Per IEC62353 edition 2	vPad-A6™

3.2.4.2 Security Settings

Through the Settings menu, vPad-Check can be enabled to detect and guard against unauthorized changes to inspection and maintenance procedures which have been "approved" by an authorized individual or management committee. The approval is represented by a digital signature which can be optionally inserted into Checklists created by Datrend's **vPad-IDE** software.

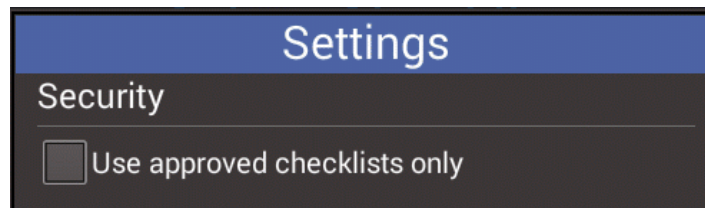


Figure 9 - Security setting.

If the 'Use approved...' box of the Settings is checked, vPad-Check will examine a Checklist for a valid date of approval, authorization individual or body, and digital signature before allowing it to be executed. Checklists which do not contain the required approvals information, or which have been modified since the approval was assigned with the vPad-IDE software, are rejected with an error message when such Checklists are selected from the app's main menu.

Checking the 'Use approved...' box of the app Settings will display a reminder that entry of a password will be required to disable the security in future. Users can opt to *not* enable security by pressing the "NO" button on the CONFIRM dialog.

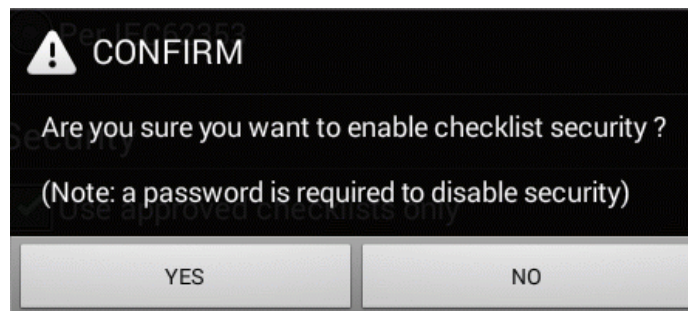


Figure 10 - Security confirmation.

Once the 'Use approved...' box has been checked, users must then enter a security password to uncheck it. The security password is based on a unique ID number which is assigned to each vPad tablet during its manufacture. To see the Tablet ID, run the vPad Installer app on the tablet



and note the ID which is displayed on the app's main menu. Derive the security password for vPad-Check from the first, fifth, ninth, thirteenth and fourteenth significant digits of the ID, excluding dashes. For example, given an ID of **34e3-ff9f-bbd4-d28f**, then the corresponding security password would be: **3fbd2**

3.2.4.3 Test Record Settings

Settings related to the Test Record are:

- | | |
|---|--|
| <i>Show results at end of test</i> | By default, inspection results and measurements acquired during execution of a Checklist are summarized in a "View Data" dialog when the Checklist is completed. Uncheck this box to skip the "View Data" display during a test. |
| <i>Disable signature/date line</i> | By default, Test Record files saved by vPad-Check include a line intended for the technician to sign and date when a hardcopy of the Test Record is printed. If Test Records are to remain in purely electronic form, you may want to check this box to disable the signature/date line in the Record. |

3.2.4.4 Data Input Settings

Technicians can be forced to enter a '**Comment**' response if anything other than **PASS** is selected in a Procedure check statement. The **other** possible selections are Fail, Service, N/A and Info Only. Depending on the setting selected, the technician may be required to enter a minimum of 2, 3, 5 or No characters to explain why the selection was chosen.

3.2.4.5 Label Printer Settings

vPad-Check has the ability to print PM inspection labels after a Test Record file has been saved.

Settings related to label printing are configured with the **vPad Settings** app



on the tablet.

vPad Settings enables you select a printer type; enter a title to be printed at the top of the label; and optionally enable a 2-D or 3-D barcode to be printed on the label, if the selected printer supports such barcodes. For more information, refer to ***MN-129 vPad Settings Operator's Manual***.

3.2.5 Hardware Setup for Label Printing

Once a Test Record has been saved at the end of a test, you have the option of printing an inspection label for application to the equipment under service. This "PM label" will include the ID or control number of the equipment, pass/fail status, the test date, the date at which the next inspection or PM becomes due, and a technician ID.

The vPad system is compatible with label printer models listed in TABLE 3.2-1, as well as other models which may interface with vPad directly through Bluetooth or WiFi. To interface a label printer listed in TABLE 3.2-1 to the vPad, an XBUS Smart Cable or "**XSC**" interface is required; refer to the table for corresponding Datrend part numbers. XSC adaptors are available for wired or wireless interface to the printer, with the wireless option being capable of both wired and wireless operation.

TABLE 3.2-1: Label Printers, including XSC Interface requirement

Printer Make/Model	XSC Interface, Wired (serial only)	XSC Interface, Wireless (Bluetooth & serial)
Dymo LabelWriter SE300 or SE450	7400-100	7400-098
Zebra GC420t	7400-108	7400-107
Brady TLS PC Link	7400-102	7400-101
Brother PT-P750W(wifi)	N/A	N/A
Brother PT-P950NW(blueetooth)	N/A	N/A
Brother PT-P950NW(wifi)	N/A	N/A
Brother QL-720NW(wifi)	N/A	N/A

Figure 11 shows the general setup for a wired connection to a supported label printer through a XSC Smart Cable interface. The figure is a schematic representation and is not drawn to scale. Setup instructions are as follows:

- Using the RJ11-6 cable provided with the XSC interface, connect the "XBUS" Port of the vPad Base Unit to either of the "**XBUS**" inputs of the interface. Note that one of the XBUS inputs will remain available to permit daisy-chaining of additional XSC units for remote device interfacing.
- Connect the serial port of the label printer to the "**DEVICE**" output of the XSC interface.
- Connect the power supply provided with the printer to the printer's power input.

- d. Apply power to the printer when required. The XSC adaptor receives its power from the vPad Base Unit, through the XBUS connection.

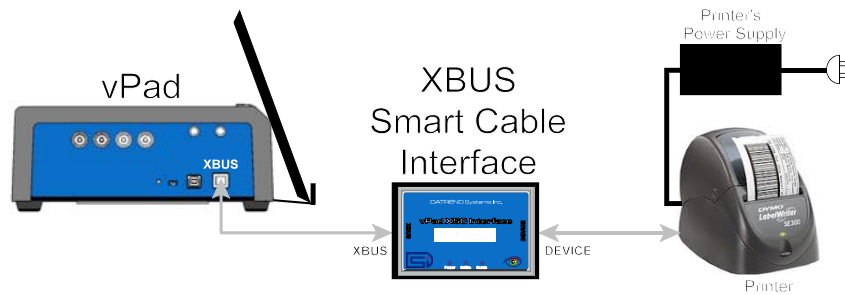


Figure 11 - Schematic showing setup of label printer - wired (serial) interface.

Figure 12 shows the general setup for a wireless connection to a label printer through a XSC Smart Cable that supports wireless communication. The figure is a schematic representation and is not drawn to scale. Communication is based on point-to-point Bluetooth and is possible over distances of up to thirty feet (10 meters).

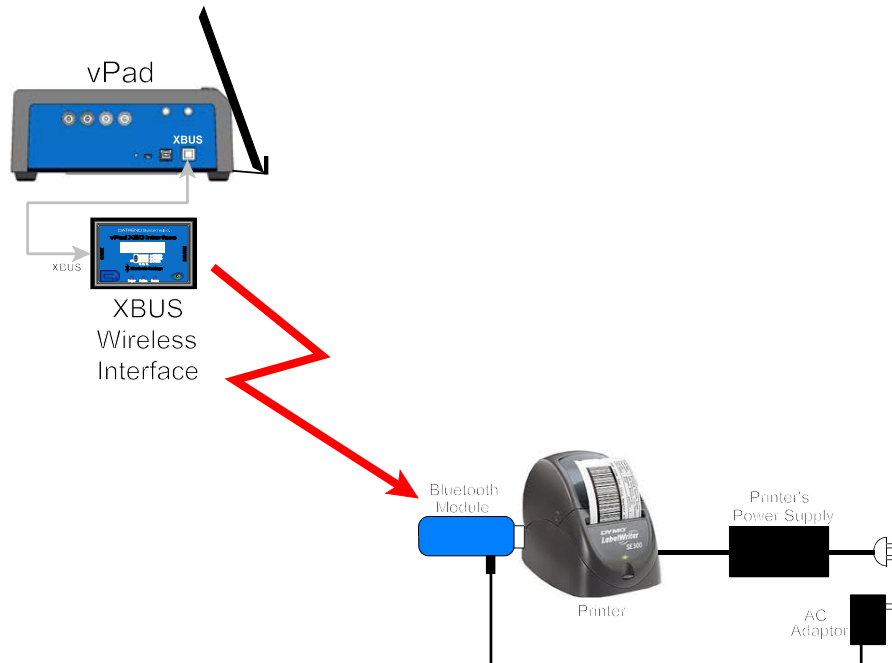



Figure 12 - Schematic showing setup of label printer - wireless (Bluetooth) interface.

Wireless setup instructions are as follows:

- a. Using the RJ11-6 cable provided with the XSC interface, connect the "XBUS" Port of the vPad Base Unit to either of the "**XBUS**" inputs of the interface. Note that one of the XBUS inputs will remain available to permit daisy-chaining of additional XSC units for external device interfacing.
- b. Connect the Bluetooth Module provided with the XSC interface to the serial port of the printer. Note this Bluetooth Module has been configured to communicate specifically with its companion XSC interface.
- c. Connect the AC adaptor provided with the XSC interface to the DC power input of the Bluetooth Module, then connect the adaptor to the AC supply.
- d. Connect the power supply provided with the printer to the printer's power input.
- e. Apply power to the printer when required. The XSC interface receives its power from the vPad Base Unit, through the XBUS connection. When powered, the XSC interface will seek out the Bluetooth Module at the printer, and if the Module is powered up, a point-to-point radio link will be established automatically.
- f. Optionally, a wireless XSC may be switched to wired operation. To do so, disconnect the XSC from the vPad and set the "BTENA" switch of the interface to the "OFF" position. This will disable the Bluetooth connectivity of the XSC. A wired connection to the printer may then be set up as shown in *Figure 11*.

In addition to the label printers identified in TABLE 3.2-1 above, vPad-Check has the ability to print to additional printer models by means of direct communication with the printer through Bluetooth or WiFi.

These alternative printers are selected and set up via the *vPad Settings* app  on the tablet. For more information on printing with these alternative printers, refer to *MN-129 vPad Settings Operator's Manual*.

3.3 Running a Checklist

A test is started by selecting the required Checklist from the main menu of vPad-Check (see **Figure 2**). A Checklist begins by prompting you to enter information about the equipment under service. If vPad-Check determines that the Checklist will require use of the electrical safety analyzer unit, the App will at this point attempt to communicate with the analyzer and verify it is on-line by means of a USB connection, or alternatively by Bluetooth. After this, you will step through the procedure one step at a time, pausing to read instructions, inspect items, and enter data as required by the procedure. Electrical safety tests and device-specific performance tests may also be executed during the procedure. At the end of the Checklist, a Test Record is saved and an inspection label may be optionally printed.

Checklists are made up of individual steps, where a step may be any of the following:

Step Type	Purpose or Function
Prompt	Displays information or instructions.
Show	Displays a drawing, photo or video.
Check	Prompts you to inspect an item and enter an assessment of the item (for example, a 'pass' or a 'fail'). Optionally, test data, comments or information about the item may also be input.
Limit Test	Prompts you to perform a test or a measurement of the equipment under service and enter the result, which is then passed or failed by vPad-Check per requirements included in the Checklist.
Input	Requires you to input specified details which are mandatory, but which are for informational or record-keeping purposes only; entered information will not be assigned a 'pass' or 'fail'.
Colour	Sets the foreground colour of text displayed by prompts, checks, input, and limit tests.
Run vPad App	Launches a vPad app for automated or manual safety testing (vPad-AS; vPad-A3; vPad-NFPA; vPad-ES; vPad-353), ESU testing (vPad-RF), or patient monitor testing (vPad-A1). Additional test device apps may be added in the future.
ECG Test	Controls the vPad ECG simulator; technician is prompted to enter results based on observation of the equipment under test.
Receptacle	Applies power to the equipment under service by means of the vPad test receptacle.
Remote Control	Depending on the step, performs one of the following operations: <ul style="list-style-type: none"> • opens a connection to a remote device through a XBUS Smart Cable (XSC) interface • controls a remote device through an XSC • captures a measurement value or a test result from a remote device through a XSC

Starting with Section 3.3.2, this chapter will provide some illustrative examples of these Step Types for the case of the "Demo Checklist" included with vPad-Check, and will explain operation of the user interface in each case.

3.3.1 Equipment Information

When a Checklist is started, the first screen shown will be for entering information about the equipment:

Figure 13 - Equipment Information menu.

Tap on a text field to display the Android keyboard and enter information. If an external keyboard is connected to the tablet, the Android keyboard is not displayed and data entry is then accepted from the external input device. A barcode scanner can also be used for data entry. Refer to Section 3.4 for detailed instructions on using the Android keyboard.

If the Control Number of the device is known, and an equipment.csv file is available in the vPad-mT folder, pressing the Search button may find the device and populate the remaining fields. If the device is not found, a 'Search Error' dialog box will be displayed. For more information on the equipment.csv file requirements, see the **MN-083b 6100-048 CMX Operator Manual, section 4.2.3** (contact Datrend).

Equipment Information
Checklist: Electrocardiograph/Multi-channel

Control Number: CN1234	Work Order: []	Work Order Type: [Choose]
Serial Number: []	Description: []	
Manufacturer: []	Model: []	
Facility: St. Joseph's Hospital	Location: []	

Tab 1 2 3 4 5 6 7 8 9 0 [Next]

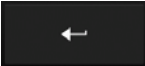
Figure 14

Equipment Information
Checklist: Electrocardiograph/Multi-channel

Control Number: CN1234	Work Order: W15-0538	Work Order Type: PM [Choose]
Serial Number: 4456-00008-TA14	Description: Diagnostic ECG	
Manufacturer: Marquette	Model: MAC VIEW	
Facility: St. Joseph's Hospital	Location: Cardiology	

Tab q w e r t y u i o p [Next]

Figure 15

Pressing ENTER  on the keyboard advances data input to the next text field on the screen.

Alternatively, use  to navigate the text fields, or simply tap a field to jump immediately there.

Note: At a minimum, an equipment control number, or other ID, must be entered in the **Control Number** field in order to save a Test Record.

"**Facility**" and "**Technician ID**" fields are backed up in permanent storage on the Android tablet. These information fields may be modified, but the last values are maintained after power to the vPad is switched off, and are restored on subsequent use of the App.

"**Work Order Type**" and "**Technician ID**" fields may be populated by selecting an item from a preset list. To do so, press the '**Choose**' button to the right of the corresponding field, then touch the desired item in the list.

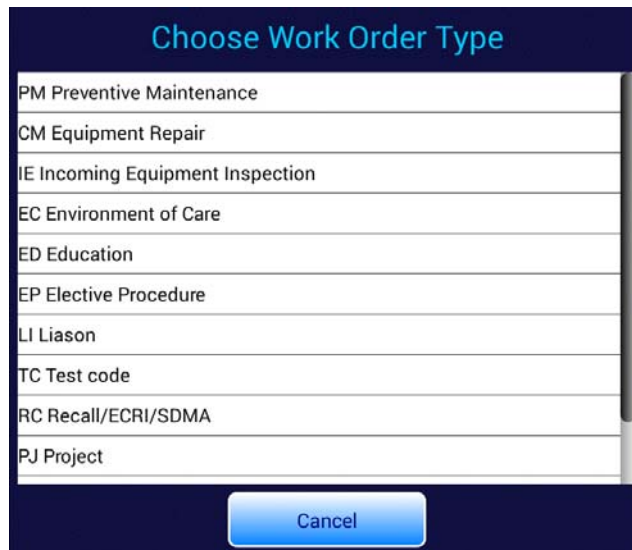



Figure 16 - Choosing a work order type.

Note the work order list and the list of technicians can be customized. Refer to the instructions provided in Section 3.2.3.

When done entering information, press  to dismiss the keyboard and expose the following buttons at the bottom of the menu:

- | | |
|-------------------------|--|
| <i>Escape</i> | Cancel the Checklist and return to the main menu. No test results are saved. |
| <i>Clear</i> | Erases all fields of the equipment information. |
| <i>Next Step</i> | Press to proceed with the next step of the Checklist. |

3.3.2 Prompts

A typical Checklist will include a number of prompt steps. Prompts are used for displaying information or instructions. Different fonts may be used for prompts, depending on the amount of information to be shown.

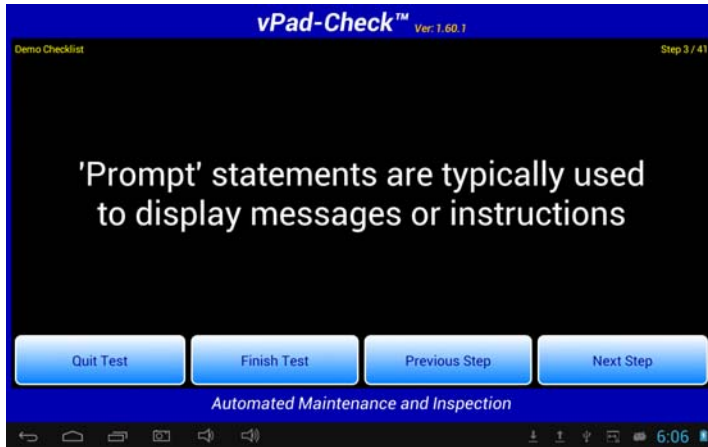


Figure 17 - Prompt with 'normal' font.

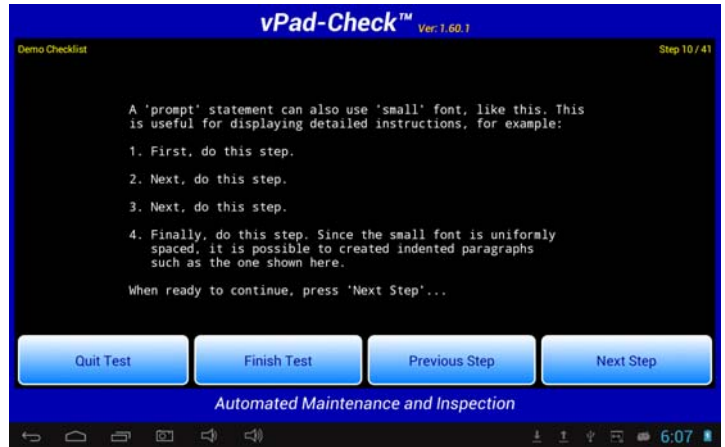


Figure 18 - Prompt with 'small' font.

Four buttons at the bottom of the screen control Checklist navigation. They are:

Quit Test

Press to abort the Checklist in progress and return to the main menu without saving a record of the test. You will be prompted to confirm the abort before the test is exited.

Finish Test

Finish the Checklist at the current step, skipping over any steps which may be remaining in the procedure. You will be prompted to confirm this action before the test is finished.

If you finish a Checklist without completing all steps, you may save a record of the test, however, the record will be assigned an overall result of "FAIL".

This option may be useful when an equipment fault is experienced during a Checklist, and it is not possible at the time to correct the fault and complete the procedure. **'Finish Test'** allows you immediately save a record of the fault, which you can refer to later during corrective maintenance.

Previous Step

Press to go back one step in the Checklist. If data or a test result was recorded for the previous step, it will be displayed and can be modified if desired.

Next Step

Press to proceed with the next step of the Checklist. Depending on the step, this button may be inhibited until requested data has been entered.

A prompt can be configured to have a timeout. In this case, a timer will appear at the top of the screen.



Figure 19 - Prompt with timer.

The Checklist will advance to the next step automatically when the countdown reaches zero. Alternatively, the **'Next Step'** button can be pressed to advance the Checklist before the timeout expires.

3.3.3 Images, Video and Documents

A Checklist may include steps for showing drawings or photographs, for playing previously-recorded video, or for showing pages excerpted from a document. Such steps will cause vPad-Check to display a description with a **'Show Picture'**, **'Play Video'** or **'Show Info'** button below the text.



Figure 20 - Checklist step for showing a photo or drawing.

The photograph viewer built into vPad-Check supports pinch-zoom and drag gestures.



Figure 21 - Viewing a photograph (JPG).



Figure 22 - Zoomed image.

For video, transport controls (pause/play/rewind/fast forward) may be viewed by touching the screen while the video is playing. You can also drag the slide control to change the playback position.



Figure 23 - Playing a video (MP4).

Document excerpts may also be shown, one page at a time. The document page viewer differs from the photograph viewer in that the displayed page fills the screen horizontally, and is placed within a window that has fast-scrolling enabled. If the page is scrolled down to a particular line and the viewer closed, the view will be restored to the same line if the same page is opened again at a later step in the procedure.

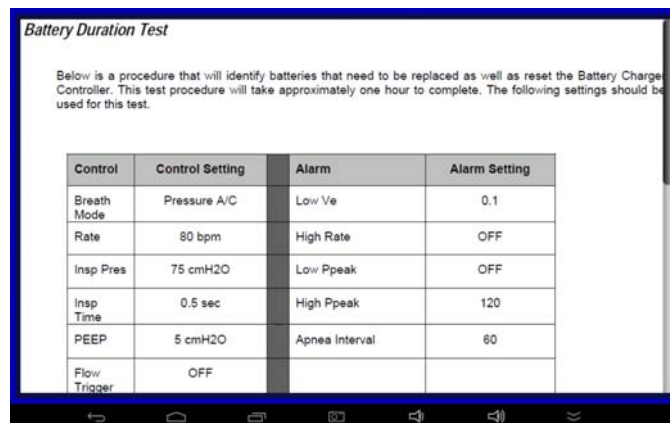


Figure 24 - Viewing a document excerpt.

To exit a picture or video and return to the Checklist, press the Android "BACK" button  .

3.3.4 Checks

A typical Checklist will include a number of steps for checking equipment features such as the chassis, the line cord, equipment controls, indicators, displays and similar items. You will be prompted to check an item and enter the result by means of radio buttons.

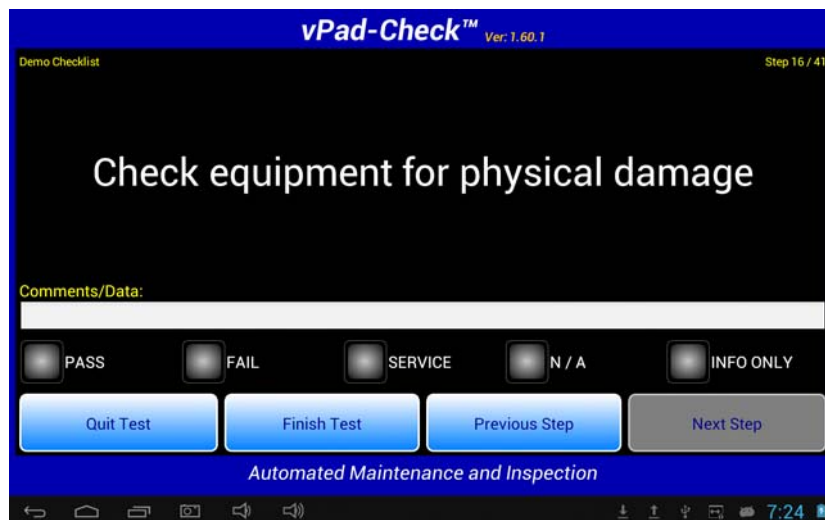


Figure 25 - Pass-fail inspection.

Before selecting a test result, you can optionally enter information by tapping the '**Comments/Data**' box, which will display the soft keyboard.

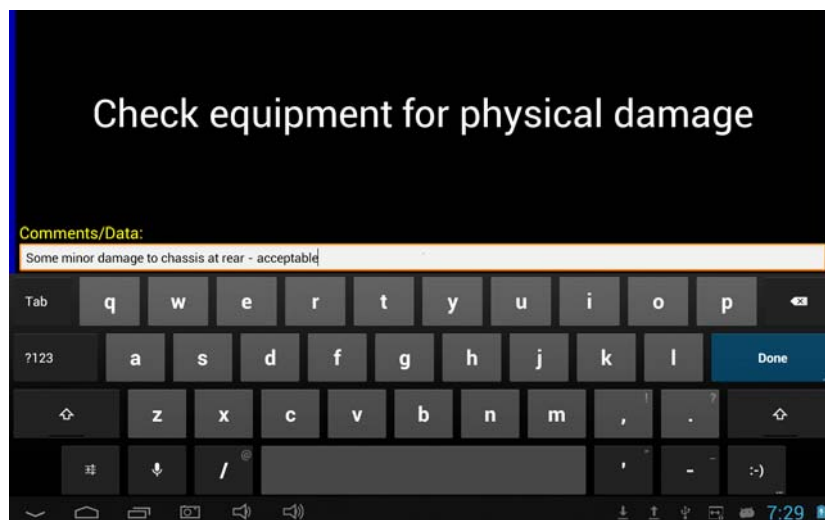


Figure 26 - Entry of comments or data.

The **'Next Step'** button will remain disabled until you select one of these results via a corresponding radio button:

PASS The inspected item meets requirements or is acceptable. When this button is pressed, the "pass" result and entered comments or data are saved and the Checklist advances to the next step automatically.

FAIL The inspected item does not meet requirements or is unacceptable. Selecting this button activates the keyboard and you will be prompted to enter a description of the fault. The Checklist advances to the next step automatically when you press **'Done'** on the keyboard. The **'Next Step'** button will remain disabled until you enter the requested description.

Note that if a "fail" is recorded for an inspection step, the Test Record will be assigned a "FAIL" for the overall test result.

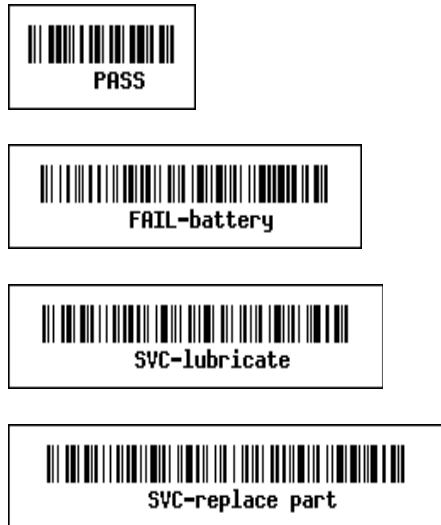
SERVICE The inspected item was serviced or repaired and has been made acceptable. Selecting this button activates the keyboard and you will be prompted to enter a description of the service performed. The Checklist advances to the next step automatically when you press **'Done'** on the keyboard. The **'Next Step'** button will remain disabled until you enter the requested description.

N/A This item is not applicable to the equipment being inspected or tested. Selecting this button activates the keyboard and you will be prompted to enter an explanation as to why the step is not applicable. The Checklist advances to the next step automatically when you press **'Done'** on the keyboard. The **'Next Step'** button will remain disabled until you enter the requested explanation.

INFO ONLY This is a response to a request for information about the equipment, for example, weight, size, version of software, and so on. Selecting this button activates the keyboard and you will be prompted to enter the information if you have not done so already. The Checklist advances to the next step automatically when you press **'Done'** on the keyboard.

If no radiobutton has been pressed, information which is entered in the **'Comments/Data'** box will be checked for the specific keywords **"PASS"**, **"FAIL"** or **"SVC"** (service). If information entered into the box either begins or ends with one of these keywords, the keyword is saved as the result and the checklist then advances to the next step automatically. **'Comments/Data'** information may contain additional text, provided the (case-sensitive) keyword appears at either the start or end of the information. This feature is most useful when data is entered by barcode scanner as an alternative or supplement to the soft keyboard. In this case, a preprinted sheet of barcodes may be used to enable rapid entry of specific inspection and

service data, such as these examples and others:



The **'Previous Step'** button may be used to step backward through a Checklist-in-progress, allowing you to review results you recorded for prior inspection steps. If necessary, a result can be changed by pressing the appropriate button and the Checklist returned to the current step by the **'Next Step'** button.

Through appropriate Checklist programming as described in Chapter 4, a check step can be configured to have alternative buttons to "PASS" and "FAIL", and may display an attached photograph, video or document page by means of an optional **'Info'** button (*Figure 31*).

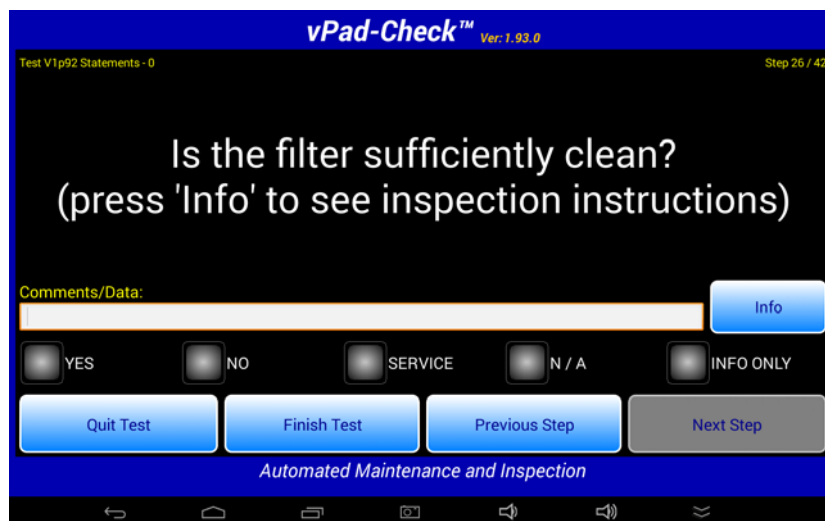


Figure 31 - Check with alternative pass/fail buttons and attached 'info'.

When buttons other than "PASS" and "FAIL" are seen during a test, the leftmost button on the screen will be interpreted as a positive response which is equivalent to a pass, and the button immediately to its right will be interpreted as a negative response which is equivalent to a failure of the inspection step. If a negative response is selected, users will be prompted to enter a rationale.

3.3.5 Limit Tests

A limit test will require you to make a measurement and enter a numerical value, which is then compared to "pass/fail" limits that are in the Checklist. Limit tests are typically used for checking performance of the equipment under service, and for inspection steps where the inspected item is quantitative (e.g., operating temperature, fluid level, battery voltage, etc.)

When prompted by a limit test, make the measurement as instructed then tap the **'Test Data'** box to display the Android keyboard. Enter the measurement result then press **'Done'** on the keyboard.

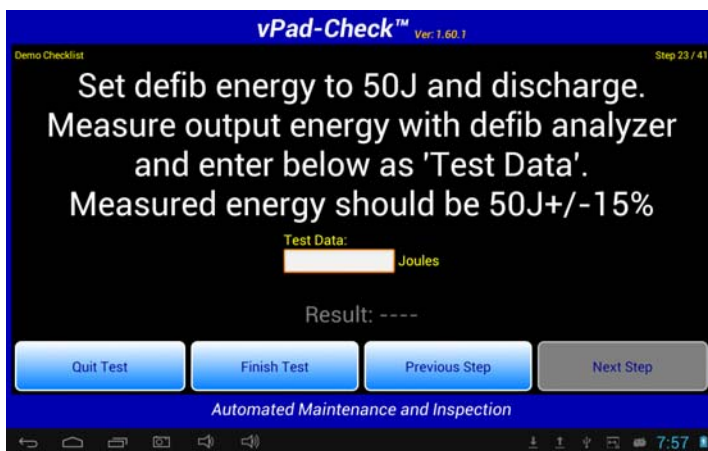


Figure 32 - Limit test.

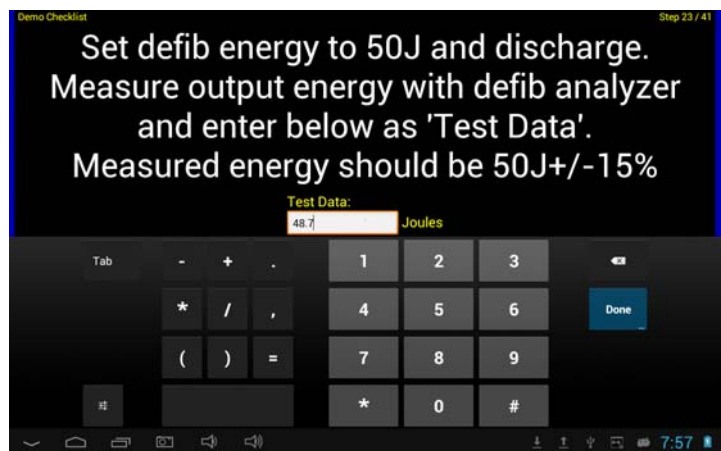


Figure 33 - Entering a measurement value.

The value you entered will be compared to the limits defined in the Checklist and a "PASS" or "FAIL" result will then be shown at the bottom of the screen. If the result is a "PASS", the Checklist will advance to the next step automatically after a short delay.



Figure 34 - Limit test result.

If the result is a "FAIL", you can repeat the measurement and enter a new value, or press **'Next Step'** to save the failing result and continue with the procedure. Note that in this case the overall result of the Checklist will be "FAIL".

3.3.6 Input Requests

Certain steps of a Checklist may request input of specific information. This is for informational or record-keeping purposes only, and the entered information is not assigned a 'pass' or 'fail'. Entry of the requested information is **mandatory** and the Checklist will not be permitted to proceed until a response has been input.

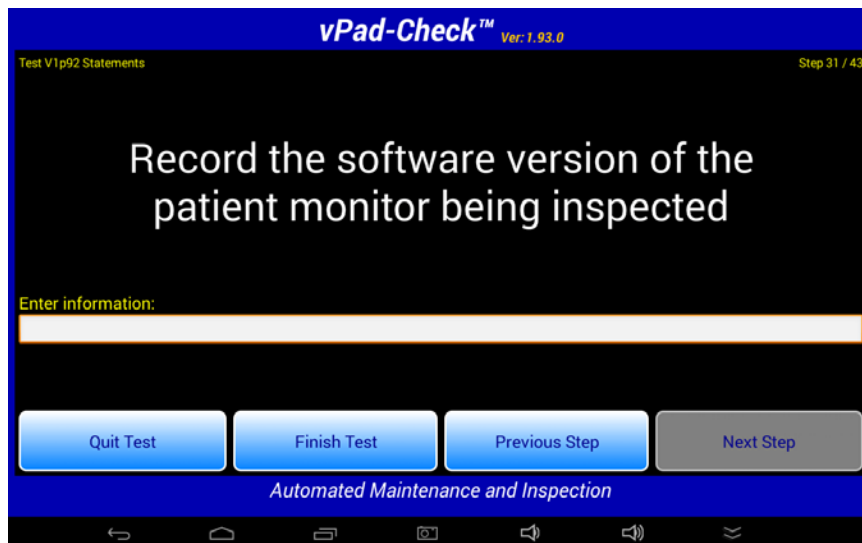
The screenshot shows the vPad-Check™ application interface. At the top, a blue header bar contains the text "vPad-Check™ Ver: 1.93.0". Below the header, the screen is black with white text. It says "Test V1p92 Statements" in the top left and "Step 31 / 43" in the top right. The main instruction in the center is "Record the software version of the patient monitor being inspected". Below this is a yellow text prompt "Enter information:" followed by a long, empty white text input field. At the bottom of the screen, there are four blue buttons: "Quit Test", "Finish Test", "Previous Step", and "Next Step". Below the buttons is a blue bar with the text "Automated Maintenance and Inspection". At the very bottom, there is a black bar with several small, light blue icons for navigation and system functions.

Figure 35 - Input request.

The keyboard will be displayed automatically when an input request is encountered. Enter the information and press **'Done'** to dismiss the keyboard and continue with the next step. The Checklist will remain at the current step until at least 3 characters have been entered into the text field; the Checklist does not confirm validity of the information entered, it only verifies that a minimum number of characters have been typed.

3.3.7 ECG Tests

If a Checklist is intended for a patient monitor or similar medical device, the Checklist may include steps for checking ECG performance. For these steps, the vPad-Check App will control the ECG simulator of the Analyzer Base Unit and you will be observing the corresponding response of the equipment under test. ECG test steps will typically be preceded by 'prompt' steps providing instructions for connecting the vPad ECG simulator to the equipment.

When the vPad ECG simulator has been activated, the simulator waveform will appear on the screen. Some examples of ECG tests are shown below, which are from the "Demo Checklist" provided with vPad-Check.

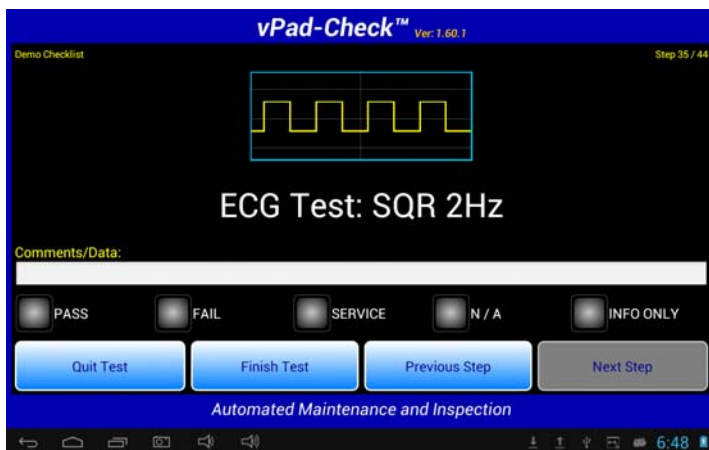


Figure 36 - ECG test - pass/fail with user input.

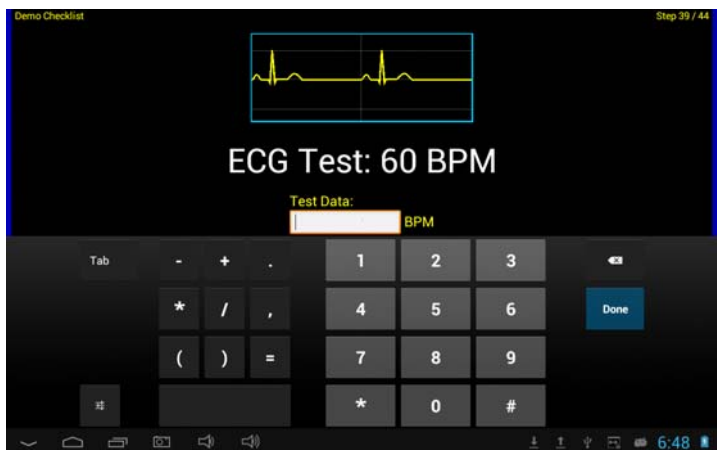


Figure 37 - ECG test - BPM limit test.

Observe the response of the equipment and enter the result requested by the Checklist. Depending on the Checklist, the requested result may be a pass/fail with user data input. For ECG NSR simulations, a limit test may be shown. For additional instructions on operation of these menus, refer to Section 3.3.4 (Checks) or 3.3.5 (Limit Tests).

3.3.8 Electrical Safety Tests and Other Tests

3.3.8.1 Automated Safety Tests

A Checklist may include one or more electrical safety tests of the equipment under service. Checklists typically preface an electrical safety test with a prompt that informs you the test will be started at the next step.

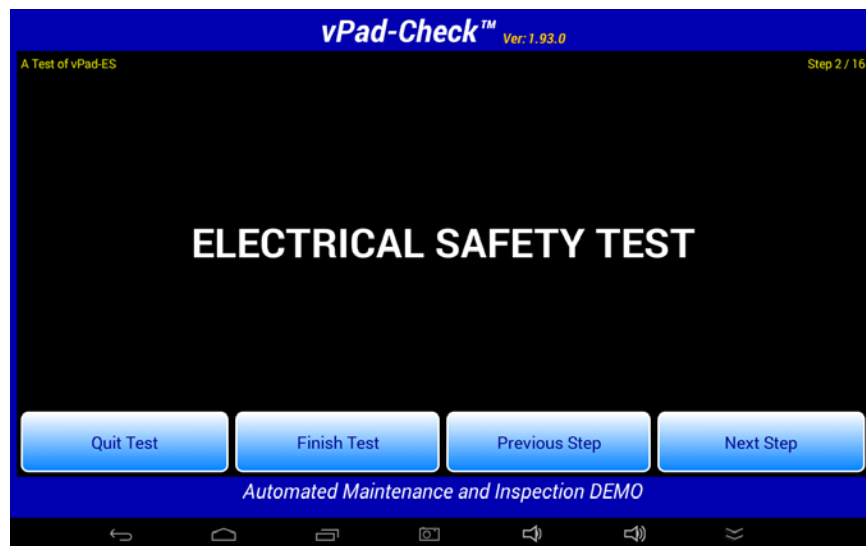


Figure 38 - Prompt for automated safety test sequence.

When a test step for an *automated* safety test is encountered, vPad-Check launches either vPad-AST™, vPad-A3™ or vPad-NFPA™ to perform the test. The App which is used will depend on the "Safety Test" standard specified on the Settings menu (see Section 3.2.4.1).

vPad-Check passes the AutoSequence number defined in the Checklist to the safety testing App. If **Enable Edits** has been checked in the safety testing App, the AutoSequence Editor screen(s) will be shown first, otherwise, the **Editor** will be skipped and the sequence will begin by running the safety measurements defined in the sequence. If **Edits** have been enabled, the safety testing App will begin after a short delay, displaying the test settings for the selected AutoSequence.

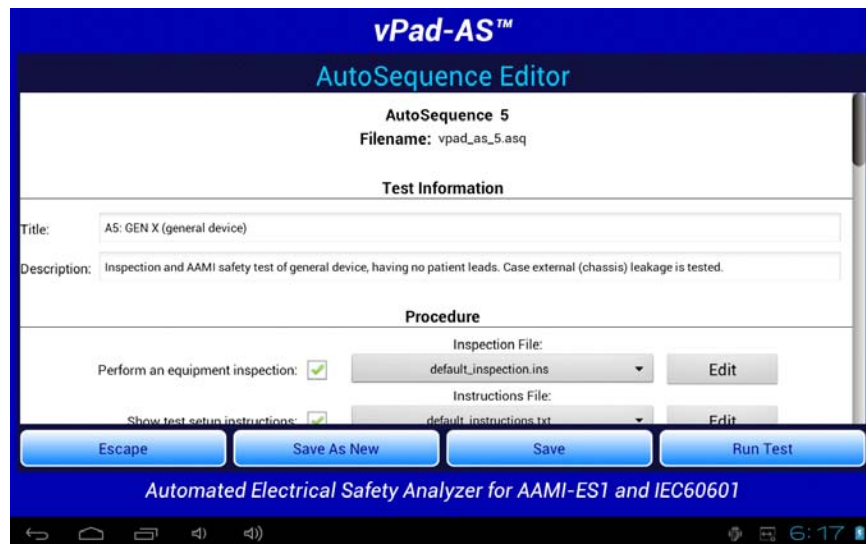


Figure 39 - vPad-Check launches safety testing App.

At this stage, you may review the test settings and make changes if necessary. Start the safety test by pressing the **'Run Test'** button.

If the safety test settings include an equipment inspection, this inspection is bypassed since it is assumed all such inspections will be performed at some other point by the Checklist itself. Typically, the safety test sequence will begin by displaying instructions for connecting the equipment under test to the vPad Analyzer Base Unit. Press **'Next Step'** when ready to proceed with the test.

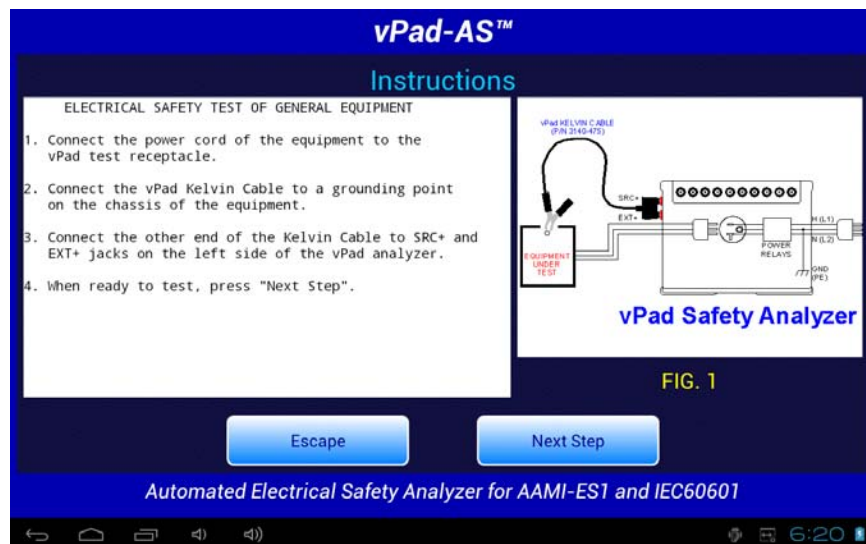



Figure 40 - Safety test setup instructions.

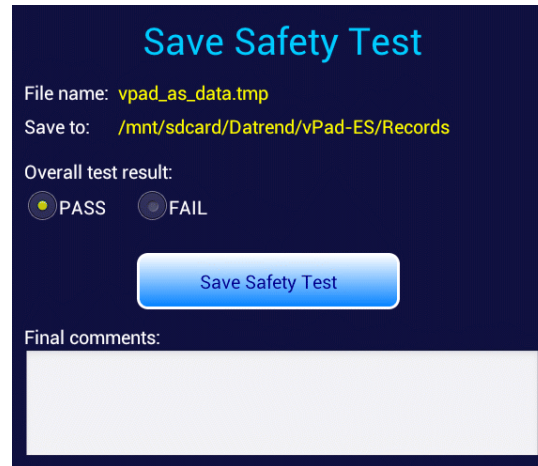
Following the instructions, the safety test will proceed automatically through a number of measurements. Each measurement is compared to a limit which is defined in the safety AutoSequence setup.



Figure 41 - Line cord resistance measurement.

The measurement sequence may stop at certain steps and you may be required to respond to prompts or alarms displayed by the safety testing App. If necessary, refer to **MN-079 (6100-045) vPad-AS Operators Manual**, **MN-080 (6100-046) vPad-A3 Operators Manual** or **MN-100 (6100-084) vPad-NFPA Operators Manual** or for further information on automated electrical safety testing and use of these Apps. Note these operator manuals are provided on the vPad Tablet, and may be viewed with the **Datrend Docs App**  which is on the HOME screen.

After all measurements of the safety test sequence have been completed, you are presented with a menu for saving the acquired results in the Test Record. The overall result of the electrical safety test is shown as "PASS" or "FAIL"; if the result is a "PASS" and you wish to change it to a "FAIL" you may do so by pressing the radio button. If the result is a "FAIL", you cannot modify the overall result. If desired, enter concluding comments about the safety test by tapping the **'Final comments'** box, which will display the Android keyboard. When ready to exit the safety testing App and return to the Checklist, press the **'Save Safety Test'** button.



Save Safety Test

File name: **vpad_as_data.tmp**

Save to: **/mnt/sdcard/Datrend/vPad-ES/Records**

Overall test result:

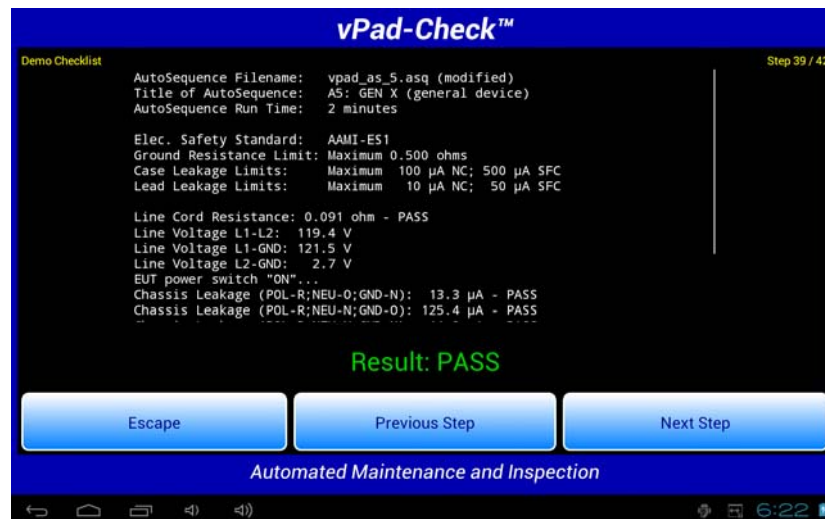
☒ PASS ☐ FAIL

Save Safety Test

Final comments:

Figure 42 - Save safety test results.

After the safety testing App is closed, measurement values and the overall result of the safety test are then displayed by the Checklist. Results are displayed in a scrollable view; to see all values, move the view by press-and-drag gesture. Press **'Next Step'** when ready to proceed to the next step of the Checklist.



vPad-Check™

Demo Checklist Step 39 / 42

AutoSequence Filename: vpad_as_5.asq (modified)
 Title of AutoSequence: A5: GEN X (general device)
 AutoSequence Run Time: 2 minutes

Elec. Safety Standard: AAMI-ES1
 Ground Resistance Limit: Maximum 0.500 ohms
 Case Leakage Limits: Maximum 100 µA NC; 500 µA SFC
 Lead Leakage Limits: Maximum 10 µA NC; 50 µA SFC

Line Cord Resistance: 0.091 ohm - PASS
 Line Voltage L1-L2: 119.4 V
 Line Voltage L1-GND: 121.5 V
 Line Voltage L2-GND: 2.7 V
 EUT power switch "ON"...

Chassis Leakage (POL-R;NEU-O;GND-N): 13.3 µA - PASS
 Chassis Leakage (POL-R;NEU-N;GND-O): 125.4 µA - PASS

Result: PASS

Escape **Previous Step** **Next Step**

Automated Maintenance and Inspection

Figure 43 - Safety test results displayed by vPad-Check.

While a safety test is running, you may press the **'Escape'** button to abort the test. The safety testing App will close and execution will return to vPad-Check. No electrical safety measurements are recorded in this case. Instead, vPad-Check will indicate the safety test has been cancelled, and this cancellation will also be indicated in the Test Record as a "skipped" safety test. If a safety test is skipped, the overall result of the Checklist will be "FAIL".



Figure 44 - Safety test aborted while in-progress.

If you abort a safety test, you can restart the safety test from the beginning by pressing the **'Previous Step'** button, or you can press **'Next Step'** to continue with the Checklist without completing the safety test. To abort the Checklist and discard all results, press **'Quit Test'**.

Many Checklists will include an automated electrical safety test as described in this section. However, it is possible that a Checklist may have more than one safety test, in which case the safety testing App may be launched *several times* over the course of an inspection and maintenance procedure. Checklist procedures for critical medical devices may include two or more safety tests to ensure thorough conformance to a particular standard. Electrical safety requirements for general devices and non-critical equipment are usually satisfied by a single safety test which has been specifically chosen to suit the equipment under service, and most example Checklists provided with vPad-Check are of this type.

3.3.8.2 Manual Safety Tests

Alternatively or in addition to an automated electrical safety test, a Checklist may include one or more steps in which measurements of electrical safety will be performed manually by the technician. Checklists may preface these manual measurements with a prompt providing testing instructions and possibly one or more pass/fail limits.

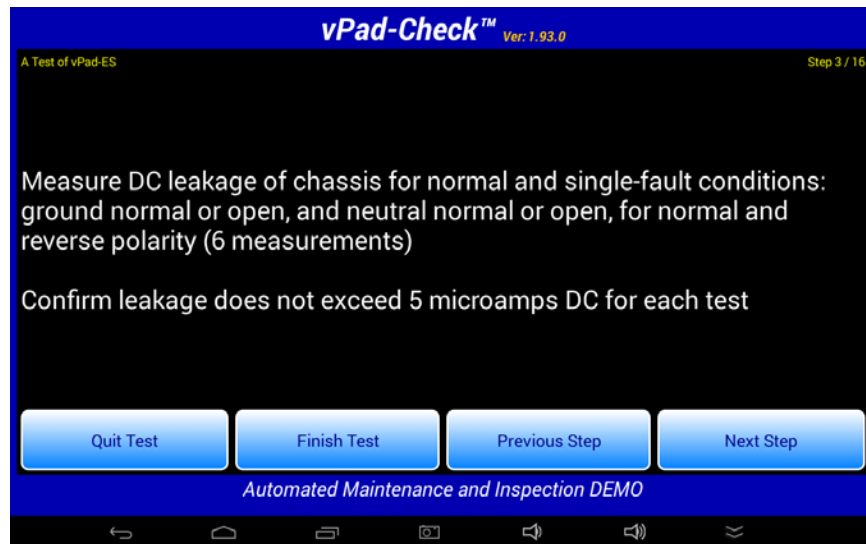


Figure 45 - Prompt for manual safety test.

Manual testing is used in specific cases where automation of the measurements as a sequence is not possible or practical. Examples include: accessible voltage measurements; point-to-point resistance measurements; point-to-point leakage measurements; leakage measurements where a non-standard measuring device is required; and measurements where a manufacturer has specified a pass/fail limit that does not conform to a standard.

When a test step for a *manual* safety test is encountered, vPad-Check launches either vPad-ES™ or vPad-353™ to perform the test. The App which is used is explicitly specified in the Checklist and will *not* depend on the "Safety Test" standard specified on the Settings menu (see Section 3.2.4.1). The vPad-353 App is used when measurements to be performed fall within the scope of the IEC62353 safety standard; vPad-ES is used for all other situations.

vPad-Check will pass the measurement type (voltage; resistance; or leakage) as defined in the Checklist to the safety testing App. When testing leakage currents with vPad-ES, configuration of the measuring device (AAMI; IEC; AC; DC) may be selected via the **'Settings'** button on the leakage menu. Perform measurements as specified in previously-displayed instructions and save each result with the **'Save Data'** button.



Figure 46 - Select safety test measurement.



Figure 47 - Save measurement result to memory.

When all required measurements have been stored, press the **'View Data'** button to recall the recorded values, then press the **'Save Data'** button to prepare the results for return to vPad-Check. On the **'Save Safety Test'** dialog, select either "PASS" or "FAIL" for the overall result with the corresponding radiobutton, then press **'Save Test'** to quit the manual test and return the data to vPad-Check. Note that "PASS" is the default result on entry to the **'Save Safety Test'** dialog; "FAIL" must be chosen by the user manually, when desired.



Figure 48 - View recorded measurement results.



Figure 49 - Select pass/fail and return to vPad-Check.

After the manual testing App has been closed, the recorded measurement values and the overall result as selected by the user are displayed by the Checklist. Results are displayed in a scrollable view; to see all values, move the view by press-and-drag gesture. Press **'Next Step'** when ready to proceed to the next step of the Checklist.



Figure 50 - Results of manual safety test.

Refer to **MN-074 (6100-040) vPad-ES Operators Manual** or **MN-075 (6100-041) vPad-353 Operators Manual** for further information on manual safety testing and use of these Apps.

3.3.8.3 ESU Performance Tests

In addition to safety testing Apps, Checklists may also launch the vPad-RF™ App and thereby execute an AutoSequence test which has been set up for performance testing of an electrosurgical generator (ESU). This requires the vPad-RF App and its associated AutoSequence files to be installed on the same tablet as vPad-Check, and for the vPad safety analyzer unit to be interfaced with the RF analyzer unit of vPad-RF (*Figure 51*) .

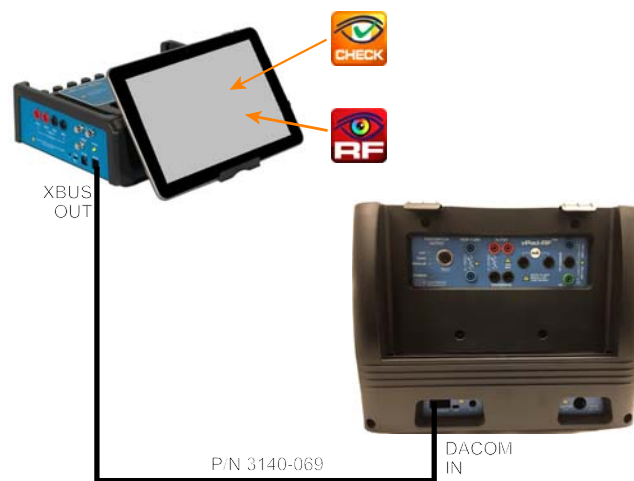


Figure 51 - Safety analyzer interface to vPad-RF™.

This can be accomplished as shown above by using a RJ12-6P6C cable (Datrend part no. **3140-069**) to connect the XBUS output jack on the safety analyzer unit to the DACOM BUS input of the RF analyzer. As an alternative to a hardwired connection, the vPad-RF App may be configured to communicate with the RF Analyzer directly, by means of Bluetooth. Per **Figure 51**, note the only tablet in the system is the one associated with the safety analyzer. The tablet provided with vPad-RF is not required in this application, however, the App and files installed on the vPad-RF tablet must be copied to the safety analyzer's tablet to enable Checklists to access them.

Similar to the way in which vPad-Check may perform electrical safety testing by means of an App such as vPad-AS, vPad-Check can perform ESU testing by means of a vPad-RF AutoSequence and accept results returned by the vPad-RF App for incorporation into the Test Record.



Figure 52 - ESU AutoSequence test in progress.



Figure 53 - ESU performance test results.

Refer to **MN-110 (6100-087) vPad-RF Operators Manual** for further information on use of the vPad-RF ESU Analyzer and AutoSequences in particular.

3.3.8.4 Patient Monitor Performance Tests

Similar to the way in which it interfaces with vPad-RF, vPad-Check can also launch the vPad-A1™ App and thereby execute an AutoSequence which has been set up for performance testing of a patient monitor. This requires the vPad-A1 App and its associated AutoSequence files to be installed on the same tablet as vPad-Check, and for the vPad safety analyzer unit to be interfaced with the Base Module of the vPad-A1 simulator (**Figure 54**).

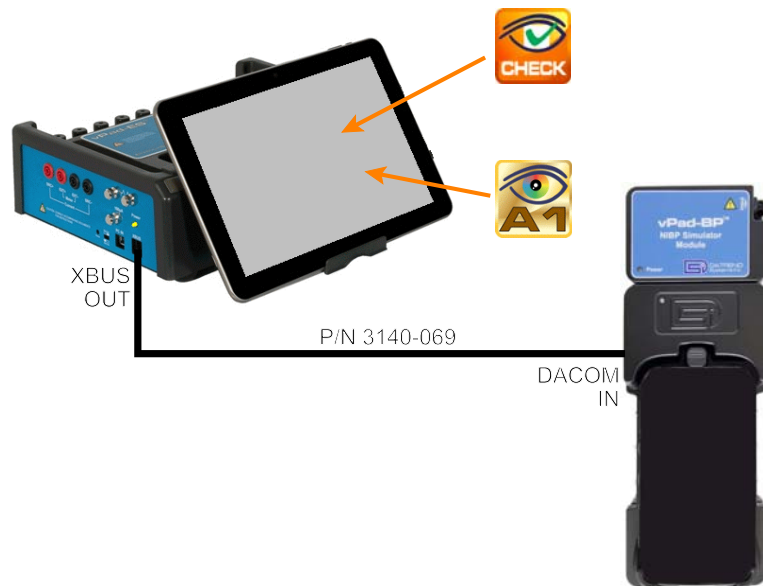


Figure 54 - Safety analyzer interface to vPad-A1™.

This can be accomplished as shown above by using a RJ12-6P6C cable (Datrend part no. **3140-069**) to connect the XBUS output jack on the safety analyzer unit to the DACOM input of the simulator. As an alternative to a hardwired connection, the vPad-A1 App may be configured to communicate with the simulator directly, by means of Bluetooth. Per **Figure 54**, note the only tablet in the system is the one associated with the safety analyzer; the tablet provided with the vPad-A1 is not required in this application, however, the App and files installed on the vPad-A1 tablet must be copied to the safety analyzer's tablet to enable Checklists to access them.

Similar to the way in which vPad-Check interfaces with vPad-RF as previously described, vPad-Check can perform testing of a patient monitor through vPad-A1 and accept results returned by vPad-A1 for incorporation into the Test Record. Refer to **MN-103 (6100-086) vPad-A1 Operators Manual** for further information on use of the vPad-A1 simulator and AutoSequences in particular.

3.3.8.5 Test Equipment Database

Checklists may also launch database Apps such as vPad-Cal™, which enables the user to include information about the test instruments used during the procedure.



Figure 55 - Selecting Test Instruments.

When invoked by vPad-Check, vPad-Cal allows the user to select one or more test instruments for inclusion in the test record file. Selected instruments will appear with a green highlighting. When all required instruments have been selected press the Exit vPad-Cal button and choose the verbose or concise format for inclusion in the test record. **Verbose** format includes all data available for the test instrument; **concise** provides a one line per instrument report, as shown below.



Figure 56 - Test Instrument Record - Concise.

Refer to **MN-088 (6100-080) vPad-Cal Operators Manual** for further information on use of vPad-Cal.

3.3.9 Remote Control

A Checklist may include steps which automatically control an external test instrument such as a defibrillator analyzer or a patient simulator. Through remote control, a Checklist may send messages to a test instrument to configure it for a particular measurement or performance test, and may also acquire measurement values or test results from the instrument which may then be passed or failed. Test data acquired by remote control is combined with other results saved while performing an inspection and maintenance procedure.

This section explains how to set up a test instrument or patient simulator for use with vPad-Check, and provides examples of Checklist steps associated with automated remote control of such test equipment.

3.3.9.1 Remote Device Connection

To connect external test equipment to the vPad, an XBUS Smart Cable or "**XSC**" interface will be required. XSC adaptors are available for wired or wireless interface to test devices, with the wireless option being capable of both wired and wireless operation.

TABLE 3.3-1 lists Datrend part numbers for XSC available at the time of writing of this manual. Additional XSC interfaces are to be released for test products from Datrend systems Inc. as well as existing competitive and legacy test products from Fluke® Biomedical, DNI Nevada, Bio-Tek Instruments and others. Contact Datrend for a current list of supported devices.

TABLE 3.3-1: XSC Interfaces for Test Equipment

XBUS Smart Cable	Wired Option (serial only)	Wireless Option (Bluetooth & serial)
Interface for Datrend AMPS-1, MPS-2 and FMS-3 multi-parameter simulators	7400-104	7400-103
Interface for Datrend Phase3 defibrillator and transcutaneous pacemaker analyzer	7400-109	7400-110
Interface for Fluke Biomedical Impulse 7000 defibrillator & transcutaneous pacer analyzer	7400-113 (USB)	N / A
Interface for IMT Medical PF-300 gas flow analyzer and ventilator tester	7400-111	7400-112
MBOX universal interface	7400-106	7400-105

The "**MBOX**" (part no. 7400-105 or -106) is a universal adaptor which provides a basic interface to almost any test device or patient simulator that can be remotely controlled through a RS-232 serial

port. With MBOX, a Checklist can send control messages to a remote test device and receive data which is sent by the device to vPad. MBOX provides data communication but results acquired from a test instrument through MBOX must be no more than one line of text, must be in English language, and must be manually passed or failed by the user.

Figure 57 shows the general setup for a wired connection to a remote device, in this case a Datrend AMPS-1 patient simulator. The figure is a schematic representation and is not drawn to scale.

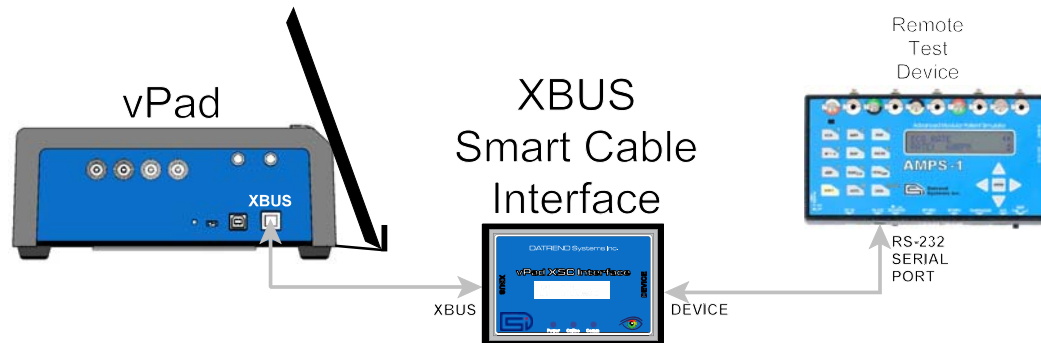


Figure 57 - Schematic showing setup of remote test device - wired (serial) interface.

General setup instructions for wired connections are as follows:

- a. Using the RJ11-6 cable provided with the XSC interface, connect the "XBUS" Port of the vPad Base Unit to either of the "XBUS" inputs of the interface. Note that one of the XBUS inputs will remain available to permit daisy-chaining of additional XSC units for remote device interfacing.
- b. Connect the RS-232 serial port of the test equipment to the "DEVICE" output of the XSC interface.
- c. Apply power to the test equipment when required. The XSC adaptor receives its power from the vPad Base Unit, through the XBUS connection.

Figure 58 shows the general setup for a wireless connection to a remote test device. The figure is a schematic representation and is not drawn to scale. Communication is based on point-to-point Bluetooth and is possible over distances of up to thirty feet (10 meters).

Wireless setup instructions are as follows:

- a. Using the RJ11-6 cable provided with the XSC interface, connect the "XBUS" Port of the vPad Base Unit to either of the "XBUS" inputs of the interface. Note that one of the XBUS inputs will remain available to permit daisy-chaining of additional XSC units for external

device interfacing.

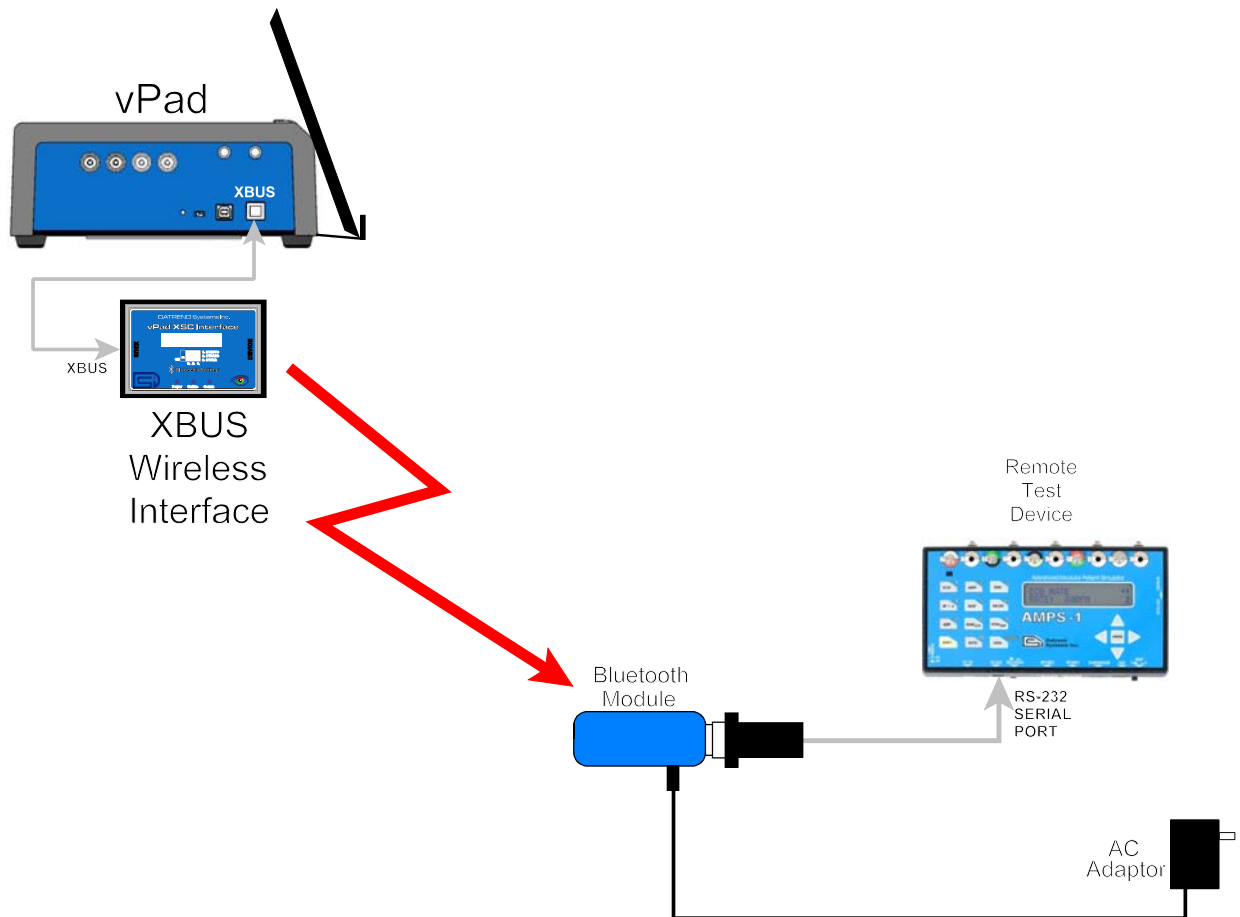


Figure 58 - Schematic showing setup of remote test device - wireless (Bluetooth) interface.

- b. Connect the Bluetooth Module provided with the XSC interface to the serial port of the test equipment. Note this Bluetooth Module has been configured to communicate specifically with its companion XSC interface.
- c. Connect the AC adaptor provided with the XSC interface to the DC power input of the Bluetooth Module, then connect the adaptor to the AC supply.
- e. Apply power to the test equipment when required. The XSC interface receives its power from the vPad Base Unit, through the XBUS connection. When powered, the XSC interface will seek out the Bluetooth Module at the remote device, and if the Module is powered up, a

point-to-point radio link will be established automatically.

- f. Optionally, a wireless XSC may be switched to wired operation. To do so, disconnect the XSC from the vPad and set the "BTENA" switch of the interface to the "OFF" position. This will disable the Bluetooth connectivity of the XSC. A wired connection to the test equipment may then be set up as shown in **Figure 58**.

3.3.9.2 Remote Control Steps

A Checklist which includes automated control of a remote test instrument or patient simulator will begin with a step that "opens" a connection to the remote device. This step may be placed anywhere in a Checklist procedure, although it is usually done near the start of the procedure. Establishing communication with the remote device will typically require one or two seconds, during which time the buttons of vPad-Check will be disabled. If connection with the remote device is successful, the Checklist will advance to the next step automatically.

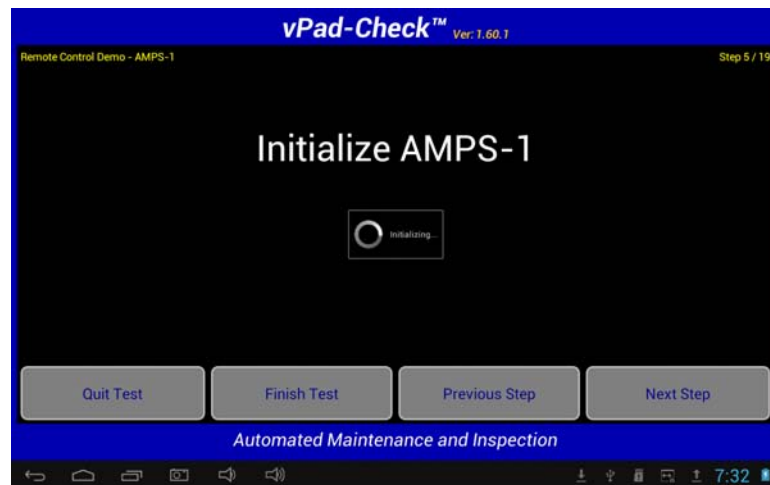


Figure 59 - *Initializing communication with remote test device.*

If the connection cannot be opened, an error message will be displayed. In this case, check physical connections per **Figure 57** or **Figure 58** and verify the test equipment is powered up. Note that certain test devices need to be set to "remote control mode" to enable serial communication; if necessary, refer to instructions provided with the test device. In the event of an communication error, you can correct the problem with the hardware and attempt to reconnect to the device by pressing '**Previous Step**' then '**Next Step**'.

A Checklist may include steps which send messages to a remote test instrument to configure the

device or initiate a measurement procedure. When this type of step is executed, a description of the remote control command is displayed by vPad-Check while the Checklist delays a few seconds, as clocked by a timer which is indicated at the top of the screen. vPad-Check buttons are disabled during transmission of the control command to the remote device. After the delay, the Checklist proceeds to the next step automatically.

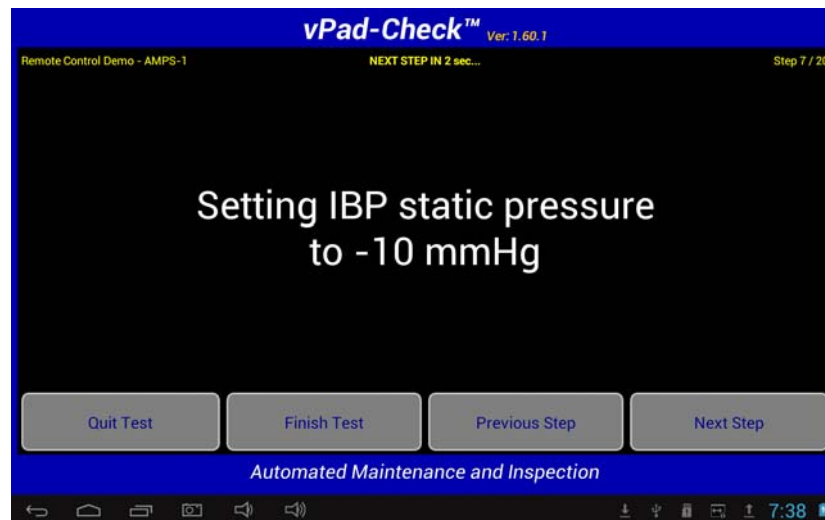


Figure 60 - Sending a control command to a remote test device.

A Checklist may include steps which poll a remote test instrument for measurement data or test results. Data from a test instrument is received and, depending on the test device and XSC adaptor in use, the data may be automatically assigned a pass/fail result or, the user may need to select "PASS" or "FAIL" manually. For received data which is automatically assigned a "PASS" by vPad-Check, the Checklist will advance to the next step automatically. For data which is not automatically assessed, select the appropriate result with the **'PASS'** or **'FAIL'** button.

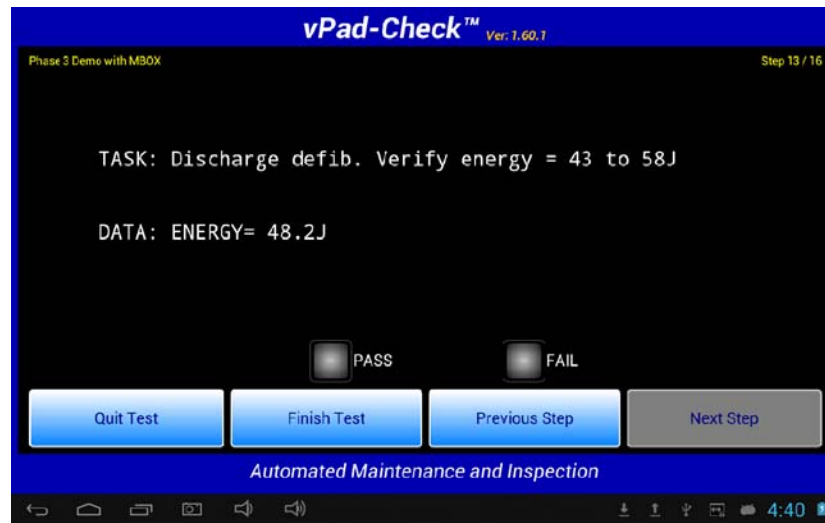


Figure 61 - Data received from remote test device (manual pass/fail).

3.3.10 Optional and Repeatable Steps

Through appropriate Checklist programming as described in Chapter 4, a Checklist may be configured to include optional steps which may be bypassed at the technician's discretion while an inspection procedure is in progress. Similarly, a Checklist may be configured to include a number of steps which can be repeated indefinitely as needed by the technician. Such Checklists are said to contain optional "branches" and/or "loops". Checklist branches and loops are useful in accommodating equipment that may have a variety of possible configurations or options at the time of inspection.

When an optional step or a group of steps is encountered during Checklist execution, the technician will be presented with a question that has a "YES" or "NO" answer.

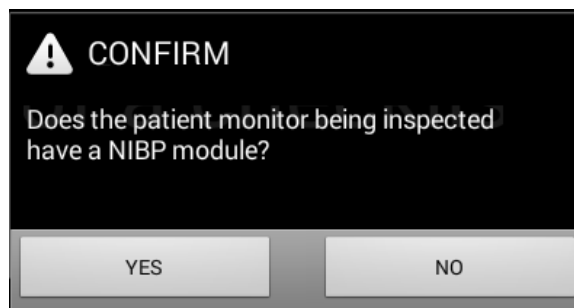


Figure 62 - Request to proceed with optional steps.

The question is phrased such that pressing "YES" will cause the Checklist to continue with the optional step or steps, while pressing "NO" will cause the Checklist to skip over the optional steps. Steps which are skipped in this way will not appear in the Test Record saved at the end of the inspection, and will have no effect on the overall pass/fail result assigned to the equipment.

Another type of optional test is the IF/ELSE branch. Because it may be difficult in this case to form a question that fits well with a Yes/No answer, the button labels may be changed to fit together with a simpler question, as indicated below.

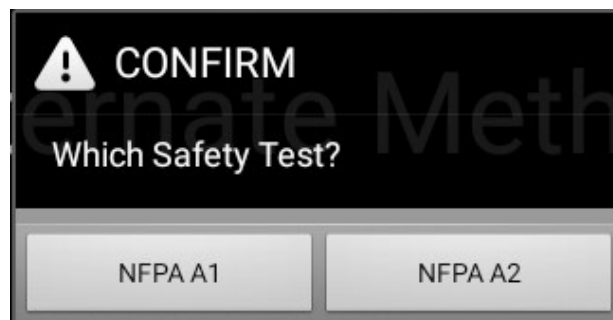


Figure 63 - Either/Or Selection (If/Else)

When a step or group of steps may be optionally repeated, the technician is presented with a similar question having a "YES" or "NO" answer.

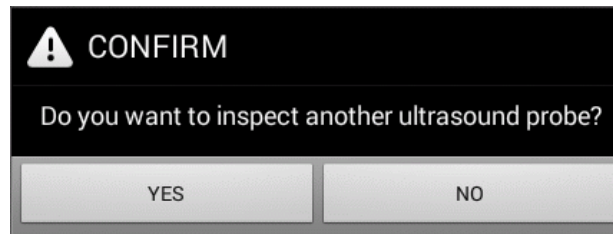


Figure 64 - Request to repeat prior steps.

In this case, the question is phrased such that pressing "YES" will cause the Checklist to repeat a previously completed step or group of steps, while pressing "NO" will cause the Checklist to proceed with the step which follows the question. When steps are repeated, results captured by the repeated steps are added sequentially to the Test Record.

The **'Previous Step'** button functions differently for optional and repeatable steps. When stepping backward through a Checklist, entry into an *optional* group of steps causes the Checklist to leapfrog backward to the start of the optional group. Back-stepping into a *repeatable* group of steps, however, will be blocked at the final step of the last repetition saved to the Test Record; this will prevent review of steps which were performed and saved earlier in the procedure.

Questions to skip or repeat designated steps in a Checklist will not themselves appear in the Test Record. Only those steps which are optionally performed or repeated and which actually save a result (for example, a check, a limit test, an electrical safety test, and so on) will be included in the Test Record.

3.3.11 Checklist Results Preview

When all steps of the Checklist have been completed, a preview of the results may then be displayed. Tasks performed and data recorded during the procedure will be displayed in a ScrollView, as shown in the example below.

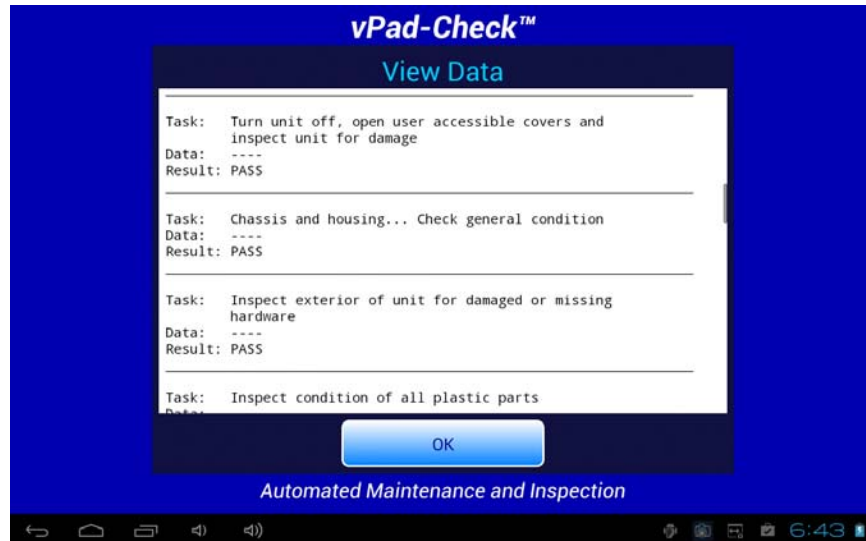


Figure 65 - Checklist results preview.

If necessary, the window can be scrolled up or down using a up/down touch-and-drag gesture. When done, press **'OK'** to proceed to the Create Record menu.

If desired, this step of a Checklist may be disabled via the Settings menu; see Section 3.2.4.3.

3.3.12 Creating a Test Record

When all results have been collected, the final step is to create a record of the test. Reports are stored in text files which have a name beginning with the EUT Control ID. The remainder of the file name includes a date and time stamp. The general format of the file name is:

```
<control-id>_<yyyymmdd>_<hhmmss>_<r>.txt
```

where **<r>** may be:

P	for a passed test, or
F	for a failed test

In the event a Control ID has not yet been entered, a reminder warning will be displayed, as shown in *Figure 66*. Pressing **'Continue'** will return to the Equipment Information screen (see *Figure 13*, page 22). Otherwise, the **'Create Test Record'** screen, *Figure 67*, will be displayed.

The screenshot shows the 'Create Test Record' interface. At the top, the title 'Create Test Record' is displayed in blue. Below it, the 'File name' is '----.txt' and the 'Save to' path is '/storage/emulated/0/Datrend/vPad-Check/Records'. The 'Overall test result' is 'PASS' and the 'Failure degree' is shown as a dropdown menu. The 'Work order type' is 'PM' with a 'Choose' button. The 'Completion code' field is partially obscured by a dialog box. The 'Technician ID' field is also partially obscured. The 'Next test due date' is '2020 Aug 20' with a 'Choose' button. The 'Test report title' is 'Datrend Systems Inc.'. At the bottom, there are 'Cancel' and 'Create Record' buttons, and a 'Final comments' text area. A modal dialog box with a warning icon and the title 'CONTROL NUMBER NOT ENTERED' is centered on the screen. The dialog text reads: 'A record of this test cannot be saved until an equipment control number or ID has been input. Enter the control number or ID using the Equipment Information menu.' and has a 'Continue' button.

Figure 66 - Control ID Reminder Screen.

Once the EUT has been identified by Control Number, it is possible to complete the creation of the Test Record (*Figure 67*). The filename and destination are displayed at the top of the screen and are not user editable.

Create Test Record

File name: 1234_20190820_145306_P.txt

Save to: /storage/emulated/0/Datrend/vPad-Check/Records

Overall test result: PASS Failure degree: ---

Work order type: PM Choose

Completion code: Choose

Technician ID: RWE Choose Labor time: hrs min

Next test due date: 2020 Aug 20 Time Type: RT F Choose

Test report title: vPad Safety Analyzer
Datrend Systems Inc.

Cancel Create Record

Final comments:

Figure 67 - Create a Record.

The Test Record file may be saved anytime by pressing the **'Create Record'** button. The **'Cancel'** button allows you to skip saving the Test Record. If **'Cancel'** is pressed, you will be prompted to confirm the action before the acquired data is discarded.

'Work Order Type' and **'Technician ID'** are typically entered via the Equipment Information menu which is shown at the start of a test (see *Figure 13*, page 22). Alternatively, these fields may be optionally entered or modified here before the Test Record is saved. Note that work order and technician lists displayed by their respective **'Choose'** buttons can be customized; refer to Section 3.2.3, page 10 for instructions.

A **'Completion Code'** may be optionally input into the corresponding field on the Create Record menu, or alternatively selected from a preset list via the associated **'Choose'** button (*Figure 68*). Completion codes are useful for classifying PM outcomes and may be helpful in developing or justifying decisions made in a programme for alternative equipment management ("AEM"). Input of a completion code is also required for some Computerized Maintenance Management Systems (CMMS), which rely on such codes to automate closure of open work orders when vPad test results are uploaded into the system. Note that the list of completion codes can be customized; refer to Section 3.2.3, page 10 for instructions.

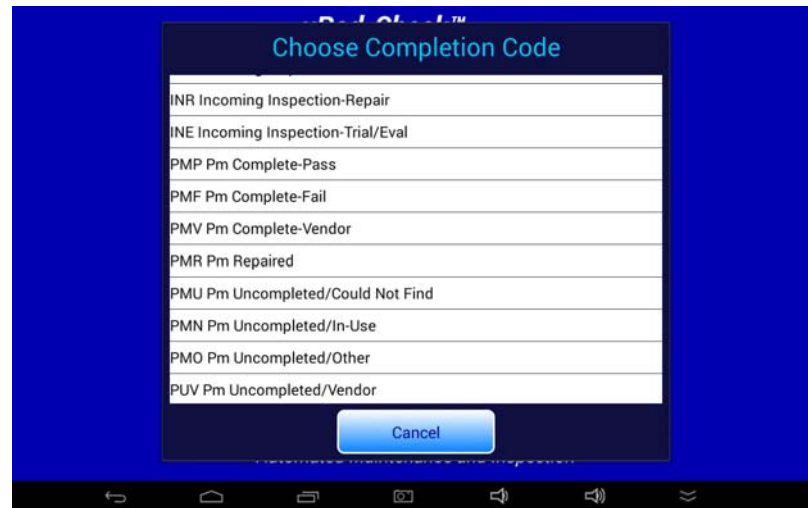


Figure 68 - Select a completion code.

Time spent by the technician/user can be recorded by entering a value into one or both of the **'Labor time:'** fields. Touching either field will result in the display of a numeric keypad to allow data entry, per *Figure 69, Figure 78*. Time may be entered in hours, minutes or a combination; if minutes are input, the two fields will be automatically adjusted as required to obtain a composite time in hours with additional minutes less than sixty. Note the labour time is intended to encompass ALL of the time spent by the user in preparing for and completing the requirements of the work order. The actual time spent using vPad-Check to run a maintenance and inspection procedure is recorded separately in the Test Record under the heading 'Checklist Run Time'. This is the time from the initial selection of the Checklist until the Test Record is created.



Figure 69 - Entering Technician Time.

It is also possible to assign a **Time Type** to the **Labor**. Standard types are Regular Time (RT) and Overtime (OT), but additional time types can be entered as required.

The **'Next test due date'** field shown on the Create Record menu (*Figure 67*) is the date at which the equipment becomes due for its *next* inspection or service. If a due date was entered by the user during input of the equipment information, then this date will be shown in the **'Next test due date'** field. If no due date was entered with the equipment information, then a default due date is automatically generated based on the current date plus the "next test" interval from the Settings menu (see Section 3.2.4.4, 3.2.4.5, page 17). If necessary, the due date shown may be adjusted by touching the field which will display the Android keyboard (*Figure 70*). The **'Next test due date'** is saved in the Test Record, and will appear on the PM inspection label if a label is printed.



Figure 70 - Modifying the default "next test due" date.


During the test, a **FAIL** may have been selected or assigned as a result for one of more steps of the Checklist. If this is true, the **'Overall Test Result:'** assigned to the Record will be **FAIL**. If **SERVICE**, **N/A** or **INFO ONLY** was selected for one or more equipment checks, the overall result will be set to **PASS** provided there are no failed steps in the Record. If the Checklist was finished before all steps were completed, or if an electrical safety test was skipped, the overall result will be **FAIL**. In each of these cases, the **'Failure degree'** (failure severity) drop-down list is enabled and allows you to add information to the Test Record about the equipment problem (*Figure 71*). Capturing information about equipment failures may be helpful for developing or justifying decisions made in programmes for alternative equipment

management ("AEM"). Accordingly, the failure degree list can be customized to suit institutional or regulatory needs; refer to Section 3.2.3, page 10 for instructions.



Figure 71 - Select failure degree when overall result is FAIL.

Test Record files include a Title at the beginning of the file. This may be modified from default "**vPad Safety Analyzer**" after tapping the **'Test report title'** field. Optionally, enter the name of your institution or business, address, phone number and so on. Up to 16 lines may be entered into this text field, of which the first three lines will be shown when the menu is initially displayed (to view or edit additional lines, scroll the window using a press-and-drag gesture). The **'Test report title'** is backed up in permanent memory storage on the Android tablet; the title is maintained after power to the vPad is switched off, and is then restored on subsequent use of the app.

In the event there is other relevant information concerning the inspection, the **'Final Comments'** field provides a means of recording such information. Touch the comments field to display the keyboard, enter the comment(s), then press the  key to dismiss the keyboard.

Once all necessary information concerning the inspection has been entered on the Create Record menu, press the **'Create Record'** button to save the Test Record.

3.3.13 Printing an Inspection Label

If label printing has been enabled via the *vPad Settings* app, after a Test Record is saved a **'Print Label'** menu will then be shown (*Figure 72*).



Figure 72 - Print PM inspection label.

This menu displays information as it will be printed on the inspection label, for example, the equipment ID, test date and the next test due date. Refer to Section 3.2.5 and *MN-129 vPad Settings Operator's Manual* for printer setup instructions. When the printer is ready to print, press the **'Print Label'** button. Multiple labels may be printed if necessary. When done, press **'Exit'** to return to the main menu of vPad-Check.

3.4 Using the Android Keyboard

When you press a text field, the Android keyboard is displayed at the bottom of the screen.

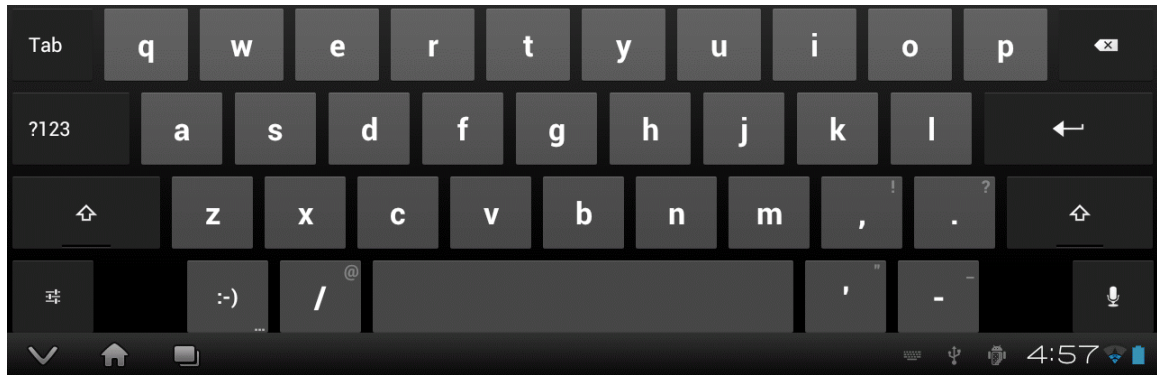


Figure 73 - Android onscreen keyboard.

The onscreen keyboard is the default method used by Android to enter information. If you have connected an alternative input device to the tablet (e.g. Bluetooth/USB keyboard or barcode scanner), then the Android keyboard will not appear when a text field is pressed and data entry will then be from the external input device. To switch back to the default keyboard, disconnect the external input device.

The text field which is currently accepting data from the Android keyboard will be highlighted:

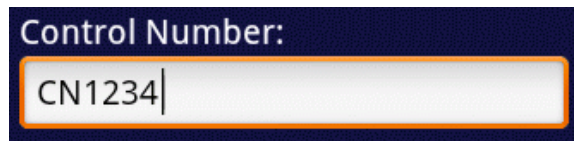
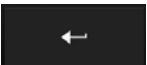
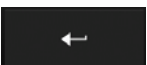

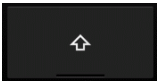



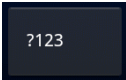
Figure 74 - Single-line text field.




Android text fields may be either single- or multi-line. A practically unlimited amount of text may be typed for either single- or multi-line fields.

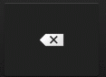
Pressing ENTER  (or **Next**) for a single-line text field causes data entry to jump to the next text field on the screen.

Pressing ENTER  on a multi-line text field inserts a new line. To move from a multi-line field to a different text field on the screen, tap on the desired field.

Press  to navigate forward through multiple text fields on a screen. Press   to navigate backward. You can also jump to a particular field on a screen simply by tapping the field.

Switch the Android keyboard from alpha to numeric modes by pressing .

In alpha mode, to activate Caps Lock press and hold the Shift key  or double-tap the Shift key so that the indicator lights up: . Once activated, you can press  to switch Caps Lock off.

To correct a mistake in typing, tap the text field to the right of the error. This will activate the Android edit cursor. Erase with backspace/correct  then retype.

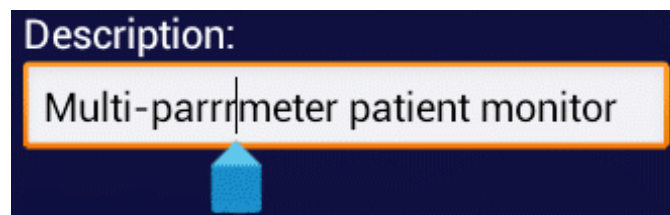



Figure 75 - Error correction with edit cursor.

When done entering text, the Android keyboard may be removed from the screen by pressing  at bottom left.

Note that as an alternative the default Android keyboard described above, one or more additional keyboards can be installed on the vPad Tablet as "Apps". Alternative keyboards may be better suited for certain types of data entry and may include support for external input devices such as barcode scanners or hardware keyboards. Keyboard apps for Android may be downloaded from the Google Play Store (<https://play.google.com/store>).

4 vPad Programming Language

vPad "Checklists" correspond to a set of files which are stored in the vPad-Check directory of the Android tablet. A Checklist file contains a *computer program* which has been written in the vPad Programming Language or "**vPL**". Each step of a Checklist procedure has a direct one-to-one relationship with a corresponding vPL *statement* in the Checklist file. vPL statements can display instructions, show photographs and video, accept data input from the user, run electrical safety tests, communicate with external test instruments and perform other functions.

This chapter specifies the vPad Programming Language and explains how vPL may be used to create Checklist procedures.

4.1 Checklist Files

4.1.1 File Location

Checklist files are stored in this directory of the Android tablet:

<storage_media>/Datrend/vPad-Check

The **<storage_media>** refers to the memory installed in the Android tablet and its name will depend on the make and model of the tablet. For most tablets, this is typically **mnt/sdcard**.

4.1.2 Filenames

Checklist files have a "**.vpl**" file extension. The filename will usually refer to the medical equipment under service, for example:

Defibrillator.vpl
Electrocardiograph.vpl
Incubator.vpl
Ventilator.vpl

Filenames may refer to medical equipment generically, as above, or they may refer to specific device manufacturers and model numbers.

A filename may also describe a particular type of maintenance procedure, for example:

How to Replace Zoll AED Battery.vpl

As in the above example, filenames may include whitespace characters and they can also include comma characters. Here are a few more examples showing use of whitespace and commas in filenames.

Monitor, CO2.vpl
Monitor, ECG.vpl
Monitor, Patient, Multi-parameter.vpl
Monitor, NIBP.vpl
Monitor, SpO2.vpl
Pump, Enteral Feeding.vpl
Pump, IV Infusion.vpl
Pump, Suction.vpl

Since the Checklist selector menu of vPad-Check displays the Checklist files alphabetically, the above is a useful way of naming Checklists which will make it easier for a user to search for a procedure of interest.

4.1.3 File Editing

Checklist files are text files that may be edited with any text editor or word processor.

You can use the text editor app installed on the Android tablet to edit Checklist files. This may be appropriate when small changes to a file are needed. Alternatively, you can copy vPad-Check file(s) to a personal computer (PC), make changes, and copy the file(s) back to the tablet. vPad-Check provides an "Update Files" feature to help with transferring files from the PC to the tablet (see Section 3.2.2).

You may find it easier to write Checklists on a PC with a *source code editor*. A source code editor is a text editor program designed specifically for editing source code of computer programs by programmers. Source code editors provide *syntax highlighting* which makes computer programs easier to read. EditPlus (www.editplus.com) is an example of a source code editor with user-definable syntax settings. For example, EditPlus may be configured to display vPL keywords in blue; comments in green; text in grey; and arguments in red¹.

¹ Requires installation of **vpl.stx** syntax file for EditPlus, available from Datrend Systems. See contact information in Chapter 5.

vPad-Check OPERATORS MANUAL

The definitive tool for creation, modification and ongoing management of Checklists is the vPad Integrated Development Environment, or **vPad-IDE™**. vPad-IDE is a complete, installable software package for Microsoft® Windows® systems which may be downloaded for free from Datrend Systems at www.datrend.com

With vPad-IDE software installed on your Windows PC (**Figure 77**), you are able to create Checklists for PM inspections and other maintenance tasks by writing them in vPL with the **Script Editor**, or by visually designing each step of the Checklist with the graphical **Step Builder**. While under development, Checklists can be tested with the IDE Step Builder or with an Android Emulator that runs on the PC. When ready, Checklists can be deployed to vPad Tablets through a number of export options, including distribution through the Internet. vPad-IDE can convert scripts to printable documents in PDF or MS Word® compatible RTF format. Optional document control features provide an approvals or sign-off process, as well as document security through an embedded digital signature. The software is supplied with an extensive library of generic PM templates to which you may add device-specific inspections and tests to satisfy OEM or institutional requirements².

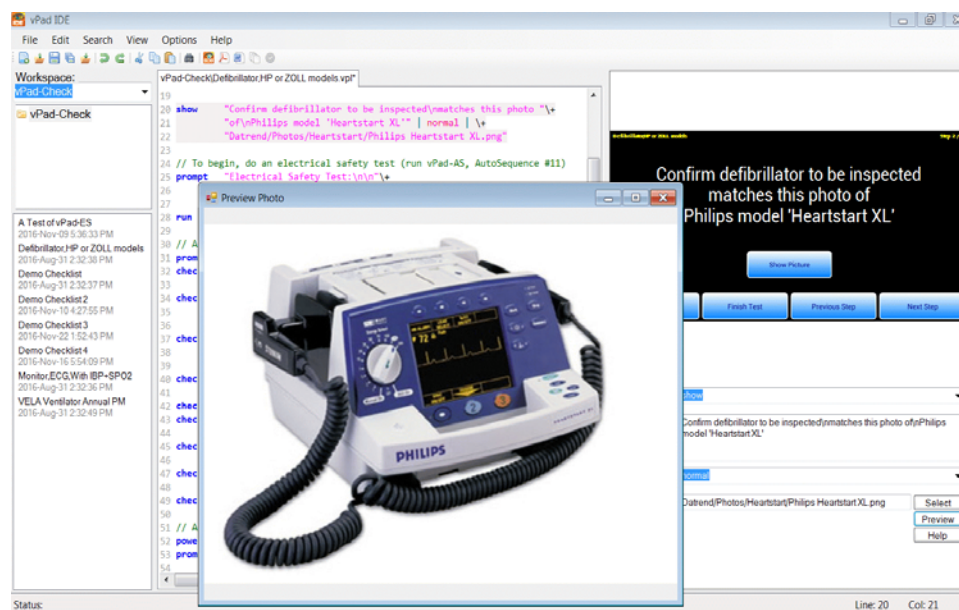


Figure 77 - Editing a Checklist with Datrend's vPad-IDE software

To obtain vPad-IDE software, simply go to www.datrend.com and navigate to the vPad Products > vPad-IDE for Windows® page. Click the "download" button and enter the requested information. You will be sent an e-mail with a link to download the software.

² Checklist procedures provided with vPad-IDE are not warranted to follow any manufacturer's or agency's recommended practise, and are provided for guidance purposes only.

4.2 Checklist Structure

4.2.1 Blank Lines

Blank lines may be placed anywhere in a Checklist program and will be ignored by vPad-Check when the Checklist is executed. Blank lines are useful for separating a Checklist into logical sections.

4.2.2 Comments

Comments may be placed within a Checklist program to explain what the program does or how it works. Any line that begins with a double-slash `//` is considered a comment by vPad-Check and will be ignored when the Checklist is executed. A comment must be placed on its own line; a comment can not be placed inside of a vPL statement or on the same line as a statement.

```
// This is an example of a comment
// And here is another comment following the first comment
```

4.2.3 Statements

Each step of a Checklist procedure has a direct one-to-one relationship with a corresponding vPL *statement* in the Checklist.

A vPL statement begins with a **keyword**. The keyword is then followed by a list of one or more **arguments**, where the arguments are separated by the pipe character (`|`):

```
keyword      <argument_1> | [argument_2] | [argument_3] ...3
```

There must be at least one whitespace between the keyword and the first argument. Arguments following the first may be optional and will depend on the keyword. The use of whitespace on either side of a pipe character is optional but helps to make the program more readable.

³ In this manual, `< >` brackets are used to indicate the required or mandatory arguments of a vPL statement, and `[]` brackets are used to indicate optional arguments.

4.2.4 Keywords

The following keywords are defined in the vPL language:

Keyword	Application
prompt	Displays information or instructions to the user
header	Similar to a prompt , displays text to the user but also places the text in the Test Record file to provide a sectional heading
show	Displays a drawing (PNG), photo (JPG) or video (MP4 or 3GP)
input	Requests mandatory input of information as specified by displayed instructions
check	Pass/fail inspection with optional user input; may include attached image, video or document excerpt, or access to a text file list of selectable values
limtst	Limit test; for quantitative inspections or performance tests where measurements are evaluated against set limits
color	Sets the foreground colour of text as displayed by prompt , show , check , input and limtst statements
power	Switches the vPad test receptacle ON or OFF
ecgsim	Controls the vPad ECG simulator
run	Runs an electrical safety AutoSequence (via vPad-AS, vPad-A3 or vPad-NFPA app), a manual safety test (via vPad-ES or vPad-353 app), an ESU performance test (via vPad-RF app) or patient monitor performance test (via vPad-A1 app)
xopen	Opens a connection to a remote test instrument or a patient simulator via an XBUS Smart Cable interface (XSC)
xctrl	Controls a remote test device through a XSC interface
xtest	Runs a test or measurement procedure with a remote test device via an XSC interface
branch	Marks the beginning of an optional group of steps within a Checklist
endbr	Marks the end of an optional group of steps within a Checklist
begin	Marks the beginning of an optionally repeatable group of steps within a Checklist
loop	Marks the end of an optionally repeatable group of steps within a Checklist
if_br	Marks the beginning of an IF/ELSE/ENDIF construct within a Checklist
else_br	Marks the end of the IF section and the beginning of the ELSE option

end_br	Marks the end of the IF section of code
---------------	---

Keywords may be written in either upper or lower case letters. The use of lower case is a convention.

4.2.5 Strings

Following the keyword, the first argument of a vPL statement may be a *string* of characters. A string is a message that will be displayed by vPad-Check when the statement is executed.

A string may be surrounded by double-quotation marks. Enclosing strings with double-quotes is optional in vPL, but it is a convention to do so in other computer languages and it will make the strings appear highlighted in a different colour when using the vPad-IDE software (see *Figure 77*).

If double-quotes are used inside of a string, then they will be removed before the message is displayed by vPad-Check. If it is necessary to emphasize a part of a message, use the single-quote character in strings.

Here are some examples of simple, single-line strings:

```
"Check fuses... ratings OK?"
```

```
"Inspect inspiratory and expiratory flow valves"
```

```
"Set switch to 'OFF' position"
```

In the last example, note the use of single-quotes to emphasize 'OFF'.

4.2.6 Newline Character Sequence

A line break may be placed inside of a string by using the `\n` character sequence. This is done to achieve a certain appearance or arrangement of the text when the message is displayed by vPad-Check.

Here is an example of a string which includes some newline characters:

```
"Fan(s) and filter(s)...\n\nCheck condition,\nclean and lubricate as required"
```

By default, vPad-Check will center-justify strings for display, unless a style is specified which overrides the default. If the above string was displayed with default center-justification, the three newlines `\n` will cause the message to appear as follows:

```

Fan(s) and filter(s)...

    Check condition,
clean and lubricate as required

```

4.2.7 Line Continuation Sequence

Many statements will be short enough to be placed on a single line in the Script Editor of vPad-IDE. Some statements, however, may have a very long string, a large number of arguments, or both.

When a statement is too long to easily fit on a single line on the editor's screen, the statement may be broken up into several lines using the `\+` character sequence, which is placed at the end of each partial line of the statement. The `\+` sequence is used only to make the Checklist program more human-readable and it has no effect on how the statement will be executed by vPad-Check.

When a statement has a long string that has been formatted into paragraphs by the `\n` sequence, the `\+` sequence may be placed at the end of each line in the paragraph to make the statement more readable. Note it is the `\n` which controls the formatting of the displayed message, not the `\+`.

Here is an example of a **limtst** statement, illustrating the use of the `\+` sequence:

```

limtst  "Set defib to 50J & discharge.\n"\+
        "Measure output energy with defib analyzer\n"\+
        "and enter as 'Test Data'.\n"\+
        "Measured energy should be 50J+/-15%\n"\+
        | Joules \+
        | target \+
        | 50 \+
        | 15 \+
        | percent

```

This example shows a `\+` continuation sequence after each line of the string argument, and also following each additional argument after the string.

This statement can be rewritten with fewer line continuation marks. The following statement will produce exactly the same result from vPad-Check as the one shown above.

Example:

```
limtst "Set defib to 50J & discharge.\nMeasure output energy with defib analyzer\n"\n      "and enter as 'Test Data'. \nMeasured energy should be 50J+/-15%" | Joules \n\n      | target | 50 | 15 | percent
```

The statement could also be written with no `\+` sequences at all. In that case, the statement will almost certainly extend outside the margins of a page as displayed by the IDE's Script Editor, but it would still be an executable statement as far as vPad-Check is concerned.

4.3 prompt Statement

Syntax: **prompt** **<string>** | [**style**] | [**timeout**]

UI Reference: Section 3.3.2

Description: The **prompt** statement displays a text string in a specified style. If the optional timeout is specified, the Checklist will automatically proceed to the next step when the timeout expires. **prompt** statements will *not* appear in the Test Record file saved at the end of an inspection.

The string is the only required argument. The style is optional and if not specified a default style of **normal** will be used. When included, the optional timeout is specified as an integer number of seconds (1, 2, 3...). If a timeout is specified, then the style must also be specified.

Styles are defined in the table below. All styles will automatically word wrap to make the string fit in the designated space on the vPad display. The newline character sequence **\n** may be used in strings to override the default word wrapping.

Style	Font			Paragraph Justification	Left/Right Margins
	Rel. Size	Colour	Character Spacing		
normal (default)	60	white*	Proportional	Centered	0 pixels
bold	60-bold	white*	Proportional	Centered	0 pixels
red	60-bold	red	Proportional	Centered	0 pixels
medium	38	white*	Proportional	Left	0 pixels
mmono	32	white*	Uniform	Left	90 pixels
small	22	white*	Uniform	Left	200 pixels

* Default font colour for these styles is white. Text colour can be modified with the '**color**' statement. See Section 4.9.

Additional styles are:

- bell** This has the same appearance as **normal** style, and adds a bell sound which is played when the prompt is shown.
- alert** This has the same appearance as **red** style, and adds an alert tone which is played when the prompt is shown.

Examples:

```
prompt "Review manufacturer's instructions"
```

```
prompt "BEGIN EQUIPMENT INSPECTION" | bold
```

```
prompt "WARNING: HIGH VOLTAGE!" | red
```

```
prompt "ATTENTION!\nDisconnect hose from PORT A now." | bell
```

```
prompt "The equipment must reach operating temperature before "\+
"testing. Switch the power 'ON' and allow about 15 "\+
"minutes for the temperature to stabilize. The green "\+
"indicator lamp on the equipment will be lit once the "\+
"operating temperature has been reached.\n\n"+
"Press 'Next Step' when the equipment is ready..." | medium
```

```
prompt "UPSTREAM (Proximal) OCCLUSION SENSOR TEST:\n\n"+
"1. Press the ON/OFF key to turn the IV pump ON. Select\n"+
"   any Care Area and press OK.\n\n"+
"2. From the Drug List select BASIC and then press OK.\n\n"+
"3. Select Primary Bag and press OK.\n\n"+
"4. For Mode select mL/hr and press OK.\n\n"+
"5. Set the Rate to 100mL/hr and press OK.\n\n"+
"6. Set VTBI mL to 13.3 mL and press OK.\n\n"+
"When ready to continue, press 'Next Step'..." | small
```

```
prompt "Delay 20 seconds. Please wait..." | normal | 20
```


4.4 header Statement

Syntax: **header** **<string>** | **[style]**

UI Reference: Section 3.3.2

Description: The **header** statement displays a text string in a specified font **[style]**. This statement functions similarly to a **prompt** statement (see Section 4.3), but with the following exceptions:

- a. it is not possible to specify a **[timeout]** for a **header** ; and (more importantly),
- b. the **header's** **<string>** argument is saved to the Test Record file when the corresponding step is executed, thereby creating a sectional heading at that point within the test results. Accordingly, it is preferable to use upper-case letters for a **header's** **<string>**, although any arbitrary text (including numerals) may be used.

Examples: **header** "1. VISUAL INSPECTIONS" | **bold**
 header "2. EQUIPMENT PERFORMANCE TESTS" | **bold**

4.5 show Statement

Syntax: **show** <string> | <style> | <filepath> | [doc]

UI Reference: Section 3.3.3

Description: The **show** statement displays a text string in a specified style, with a "**Show Picture**", "**Play Video**" or "**Show Info**" button beneath the text. When the button is pressed, the photograph, video, or document page specified by <filepath> is opened and shown on the tablet.

All arguments are required, except for [doc] which is optional. Text styles are the same as those specified for the **prompt** statement (see Section 4.3). The **doc** argument is added at the end to enable the document page viewer as functionally described in Section 3.3.3; it is otherwise omitted for the photographic viewer or the video player.

The filepath includes the filename of the image or video and all directories required to specify the path, up to but *not* including the name of the storage media (i.e. , the filepath will exclude '**mnt/sdcard**'). Similar to a string, the filepath may be enclosed in double-quotes but this would be for code readability only and is optional.

Supported file types are **.png** and **.jpg** for photographs, **.mp4** and **.3gp** for video, and **.png** for document excerpts or pages. The file type determines whether the button displayed by vPad-Check is "**Show Picture**", "**Play Video**" or "**Show Info**".

Examples:

```
show "Verify the equipment being serviced\n"\+
    "matches this photo of the defibrillator" | normal | \+
    "Datrend/vPad-ES/Instructions/hp_defib_1.jpg"

show "Play this video to see how to remove\n"\+
    "the rear panel of the equipment" | normal | \+
    "DCIM/Camera/VID_20131022_143820.mp4"

show "Read these instructions for\n"\+
    "the battery test procedure" | normal | \+
    "Datrend/Photos/VELA/VELA_pm_sl_battery.png" | doc
```

4.6 input Statement

Syntax: **input** <string> | [**style**]

UI Reference: Section 3.3.6

Description: The **input** statement displays a text string along with a keyboard and a text box for user input of *mandatory* data or comments.

This statement saves the entered information to the Test Record. Contents of the comments/data text box must be at least 3 characters/digits long and are saved as the test data for the step. The <string> of the statement is also saved, to give the recorded data and result some context. The Result is automatically assigned as "INFO ONLY".

The [**style**] of the text may be any of the styles supported by the **prompt** statement (see Section 4.3).

Examples: **input** "Record the software version number"

 input "Enter the lot number of the disposable pads"

 input "Enter the part number of the cable"

4.7 check Statement

Mode 1: Show Image

Syntax: **check** <string> | [style] | [pass_text] | \+
 [fail_text] | [image_filename] | [doc]

UI Reference: Section 3.3.4

Description: The **check** statement displays a text string along with radiobuttons and a text box for user input of data or comments. If an **[image_filename]** is specified, the Comments text box on the vPad screen will be shortened, and an "Info" button will then be displayed at the extreme right (see *Figure 31*, page 31).

Except for the <string>, all other arguments of the statement are optional. As required, arguments after the <string> are progressively specified moving from left to right, with no unspecified arguments between the <string> and the final argument. If "blank" or "empty" arguments are placed between the <string> and the last argument furthest to the right, unpredictable behaviour of the statement may result.

This statement saves information to the Test Record. The radiobutton which is pressed is recorded as the test result for the Checklist step. All radiobuttons, with the exception of the Fail button, are considered a Pass. All radiobuttons, with the exception of the Pass button, require at least 3 characters/digits of input in the Comments text box to explain 'why' the button was selected. Contents of the comments/data text box, if any, are saved as the test data for the step. The <string> of the statement is also saved, to give the recorded data and result some context.

Default labels associated with the PASS and FAIL buttons on the vPad screen are "PASS" and "FAIL", respectively. These buttons may be optionally assigned alternative labelling via the **[pass_text]** and **[fail_text]** arguments respectively. These arguments must be chosen such that the **[pass_text]** is a positive response which implies a pass (e.g. "YES", "COMPLETE", "ACCEPTED", etc), and the **[fail_text]** is a negative response which implies a failure (e.g. "NO", "INCOMPLETE", "REJECTED", etc).

The optional **[image_filename]** can be a **jpg**, **png** or **mp4** file. A complete path to the file must be specified, *excluding* the name of the storage media (e.g. **mnt/sdcard**). Typically, images will be stored in the **Datrend/Photos** folder on the tablet. Images may be placed in a subfolder which is named according to the Checklist, or perhaps the make

and/or model of equipment that will be inspected with the Checklist. Alternatively, images may be organized in equipment subfolders in a **Datrend/vPad-Check/Show** folder.

If an **[image_filename]** is specified, the optional **doc** argument may be added at the end to enable the document page viewer; it is otherwise omitted to display **[image_filename]** in the photographic viewer (**jpg** or **png** file), or the video player (**mp4** file).

Examples:

```

check  "Chassis and housing...\n\nCheck general condition"

check  "Check power cord, plug and strain relief"\+
      | normal | "COMPLETED"

check  "Is the equipment securely attached via the wall " \+
      | "mount?" | red | "YES" | "NO"

check  "Test the ventilator's alarms and audible signals.\n\n"\+
      | "Press the 'Info' button for detailed instructions." \+
      | medium | "PASS" | "FAIL" \+
      | "Datrend/Photos/Ventilator/vent_alarm_service.png" | doc

```

Mode 2: Response Lists

Syntax: **check** <string> | [style] | \+
 [pass_text] | [fail_text] | [list_filename]

UI Reference: Section 3.3.4

Description: The check statement displays a text string along with radiobuttons and a text box for user input of data or comments. If a [list_filename] is specified, the Comments text box on the vPad screen will be shortened, and a "List" button will then be displayed at the extreme right.

Except for the <string> and <list_filename>, all other arguments of the statement are optional. As required, arguments after the <string> are progressively specified moving from left to right, with no unspecified arguments between the <string> and the final argument. If "blank" or "empty" arguments are placed between the <string> and the last argument furthest to the right, unpredictable behaviour of the statement may result.

This statement saves information to the Test Record. The radiobutton which is pressed is recorded as the test result for the Checklist step. All radiobuttons, with the exception of the Fail button, are considered a Pass. All radiobuttons, with the exception of the Pass button, require at least 3 characters/digits of input (in the default mode) in the Comments text box to explain 'why' the button was selected. The number of characters can be varied from 0 to 5 in the Settings selection. Contents of the comments/data text box, if any, are saved as the test data for the step. The <string> of the statement is also saved as the Task, to give the recorded data and result some context.

Default labels associated with the PASS and FAIL buttons on the vPad screen are "PASS" and "FAIL", respectively. These buttons may be optionally assigned alternative labelling via the [pass_text] and [fail_text] arguments respectively. These arguments must be chosen such that the [pass_text] is a positive response which implies a pass (e.g. "YES", "COMPLETE", "ACCEPTED", etc), and the [fail_text] is a negative response which implies a failure (e.g. "NO", "INCOMPLETE", "REJECTED", etc).

The optional [list_filename] is a simple ANSI text file. A complete path to the file must be specified, **excluding** the name of the storage media (e.g. mnt/sdcard). Typically, list files will be stored in the Datrend/Lists folder on the tablet. List files may be placed in a subfolder which is named according to the topic of the list, or perhaps the make and/or model of equipment that will be inspected with the Checklist. Alternatively, list files may be organized in equipment subfolders in a Datrend/vPad-Check/Lists folder.

List files provide a convenient way of entering data into the Comments field from a pre-determined list of options. As an example, this could be a list of replacement batteries with part numbers, or a list of generic parts, or a list of accessory items normally replaced. Selection from the list can be made multiple times, with the selections appending to the Comments. The Comments line can be cleared completely using a press and hold (approx 2 sec) in the Comments field.

Examples: `check "Test battery output,\n replace if necessary. | normal |
"REPLACED" | "FAIL" | "Datrend/Database/CMMS_GENERIC
Tables/Batteries_Monitors.lst"`

When appending entries from a List file, it can be inconvenient to keep going back to the keyboard to enter delimiting text such as ` , ' . To assist in creating a readable Comment of two or more List entries, there is a **'delimiter'** input box. When creating the Comment string, open the List, select an entry (you will go back to the main check window), touch on the delimiter box (to the right of the Comments label), open the List, etc.

The delimiter can be changed by touch and holding the delimiter box. Another input box and the keyboard will appear. Enter the new delimiter string (eg. `: , ' , `; ' , etc.) and finish the entry by pressing **Done** (or the equivalent) on the keyboard. The delimiter box will

now show the new text to be used as a delimiter.



Figure 78 - Using List Files

4.8 limtst Statement

Syntax:

```
limtst <string> | <UOM> | <comparison> | <X> | [Y] | [tol_type]
```

UI Reference: Section 3.3.5

Description: This statement encodes a "limit test", in which a user is prompted to input numerical data such as a measurement or a test result, and the entered value is then compared to one or two boundary values to obtain a result of "PASS" (measurement within limits) or "FAIL" (measurement outside of limits).

Arguments of the limit test statement are:

<string>	This is a required argument. This message is intended to provide instruction to the user as to the quantity to be measured or tested and then entered into the "Test Data" field of the UI.
-----------------------	---

<UOM>	This is the units of measure and it a required argument. This string is displayed to the right of the "Test Data" field and provides further information as to the quantity being measured or tested.								
<comparison>	This is a required argument which specifies the type of comparison performed by the limit test. The defined values for this parameter are: <table> <tr> <td>min</td><td>Compare entered data to a minimum threshold. If the value entered by the user is greater than or equal to threshold X, the test result is assigned a "PASS". If the entered data is less than X, the test result is "FAIL".</td></tr> <tr> <td>max</td><td>Compare entered data to a maximum threshold. If the value entered by the user is less than or equal to threshold X, the test result is assigned a "PASS". If the entered data exceeds X, the test result is "FAIL".</td></tr> <tr> <td>range</td><td>Compare entered data to an acceptance range. If the value entered by the user falls within the range set by lower limit X and upper limit Y, the test result is assigned a "PASS". If the value falls outside the range of X to Y, the test result is "FAIL".</td></tr> <tr> <td>target</td><td>Compare entered data to an acceptance range which is defined by an expected value or "target" X and a +/- tolerance Y. If the value entered by the user falls within the range defined by target X and tolerance Y, the test result is assigned a "PASS". If the value falls outside the range defined by X and Y, the test result is "FAIL".</td></tr> </table>	min	Compare entered data to a minimum threshold. If the value entered by the user is greater than or equal to threshold X , the test result is assigned a "PASS". If the entered data is less than X , the test result is "FAIL".	max	Compare entered data to a maximum threshold. If the value entered by the user is less than or equal to threshold X , the test result is assigned a "PASS". If the entered data exceeds X , the test result is "FAIL".	range	Compare entered data to an acceptance range. If the value entered by the user falls within the range set by lower limit X and upper limit Y , the test result is assigned a "PASS". If the value falls outside the range of X to Y , the test result is "FAIL".	target	Compare entered data to an acceptance range which is defined by an expected value or "target" X and a +/- tolerance Y . If the value entered by the user falls within the range defined by target X and tolerance Y , the test result is assigned a "PASS". If the value falls outside the range defined by X and Y , the test result is "FAIL".
min	Compare entered data to a minimum threshold. If the value entered by the user is greater than or equal to threshold X , the test result is assigned a "PASS". If the entered data is less than X , the test result is "FAIL".								
max	Compare entered data to a maximum threshold. If the value entered by the user is less than or equal to threshold X , the test result is assigned a "PASS". If the entered data exceeds X , the test result is "FAIL".								
range	Compare entered data to an acceptance range. If the value entered by the user falls within the range set by lower limit X and upper limit Y , the test result is assigned a "PASS". If the value falls outside the range of X to Y , the test result is "FAIL".								
target	Compare entered data to an acceptance range which is defined by an expected value or "target" X and a +/- tolerance Y . If the value entered by the user falls within the range defined by target X and tolerance Y , the test result is assigned a "PASS". If the value falls outside the range defined by X and Y , the test result is "FAIL".								
<X>	This is a decimal number and it is a required argument for all limit tests. According to the comparison selected, the X value will set the threshold for a max or min test, the lower bound for a range test, or the expected value of the entered data for a target test.								
[Y]	This is a decimal number and it is required only for range or target tests. According to the comparison selected, the Y value will set the upper bound for a range test or the +/- tolerance for a target test.								
[tol_type]	This argument selects the tolerance "type" and is required only for target tests. The argument has two defined settings, as follows: <table> <tr> <td>units</td><td>The tolerance Y is represented in the same units of measure</td></tr> </table>	units	The tolerance Y is represented in the same units of measure						
units	The tolerance Y is represented in the same units of measure								

as the target value **X**. In this case, the acceptance range for entered data will span **(X - Y)** to **(X + Y)**.

percent The tolerance **Y** is a percentage of the target value **X**. In this case, the acceptance range for entered data will span **(X - dX)** to **(X + dX)**, where **dX = (Y * X)/100**.

The **limtst** statement saves information to the Test Record. The value entered by the user is saved along with the "PASS" or "FAIL" result of the limit comparison. The **<string>** of the statement is also saved, to give the recorded data and result some context.

Examples:

```
limtst "Check battery voltage exceeds 10V" | volts | min | 10

limtst "Test occlusion pressure alarm\n"\+
      "of infusion device.\n\n"\+
      "Check pressure does not exceed 17.5 psi"\+
      | psi | max | 17.5

limtst "Measure power output.\nCheck power = 200 to 400 uW"\+
      | microwatts | range | 200.0 | 400.0

limtst "Check ECG ratemeter reads 60 +/- 2 BPM"\+
      | BPM | target | 60 | 2 | units

limtst "Discharge defib at 100J.\nCheck energy = 100J +/- 15%\n"\+
      | Joules | target | 100 | 15 | percent
```

4.9 color Statement

Syntax: **color** <rgb_setting>

Description: Sets the foreground colour of text as displayed by **prompt**, **check**, **limtst** or **show** statements. **rgb_setting** is a 24-bit number in hexadecimal notation which specifies the relative mix of red, green and blue in the overall colour. The argument begins with a '#' character, followed by red, green and blue values with each component respectively ranging from 00 (black) to FF (maximum intensity).

By default, Checklists display text in white (**rgb_setting** = #FFFFFF). When required, the default colour can be changed by preceding a **prompt** or other statement with a **color** statement. The colour setting remains in effect until changed by a subsequent **color** statement.

Examples:

```

color  #FF0000
prompt "This text will appear in bright red"

color  #00FF00
prompt "This text will appear in bright green"

color  #0000FF
prompt "This text will appear in bright blue"

color  #FFFF00
prompt "This text will appear in bright yellow"

color  #F8F8A0
prompt "This text will appear in a softer yellow"

color  #FFC700
prompt "This text will appear in orange"

color  #0080FF
prompt "This text will appear in a medium blue"

// Reset text colour to default (white)
color  #FFFFFF

```

4.10 power Statement

Syntax: **power** **<state>**

UI Reference: This statement produces no display indication on the Android tablet. The state of the test receptacle can be determined by observing the indicator lamp next to the receptacle.

Description: Controls the state of the test receptacle of the vPad Analyzer Base Unit. The **state** of the receptacle may be set to either **on** or **off** respectively.

Examples: **// Apply power to equipment under test**
 power on

// Power down the equipment under test
 power off

4.11 ecgsim Statement

Syntax: **ecgsim** <wave_number> | [bpm] | [delta_bpm]

UI Reference: Section 3.3.7

Description: The **ecgsim** statement controls the vPad ECG simulator and provides a choice of user interface for data entry and test result selection.

Arguments of the **ecgsim** statement are:

<wave_number> This argument is required. It is an integer value (1, 2, 3... 30) which represents the waveform to be generated by the ECG simulator according to the table below.

No.	Waveform	Description
1	SQR 2Hz	Square wave, 2 Hz, 1 mV p-p
2	TRI 2Hz	Triangle wave, 2 Hz, 1mV p-p
3	PUL 4sec	'DC' pulse, +1 mV for 4 sec, 0 mV for 4 sec
4	PUL 30ppm	Pulse, +1mV for 63 msec, rate = 30 ppm
5	PUL 60ppm	Pulse, +1mV for 63 msec, rate = 60 ppm
6	SQR 1 KHz	Square wave, 1 KHz, 1 mV p-p
7	SIN 0.5 Hz	Sine wave, 0.5 Hz, 1 mV p-p
8	SIN 10 Hz	Sine wave, 10 Hz, 1 mV p-p
9	SIN 40 Hz	Sine wave, 40 Hz, 1 mV p-p
10	SIN 50 Hz	Sine wave, 50 Hz, 1 mV p-p
11	SIN 60 Hz	Sine wave, 60 Hz, 1 mV p-p
12	SIN 100 Hz	Sine wave, 100 Hz, 1 mV p-p
13	CMRR SQR 2Hz	Square wave, 2 Hz, 10mV common mode
14	CMRR PUL 4sec	DC pulse, 10mV common mode
15	CMRR SIN 0.5Hz	Sine wave, 0.5 Hz, 10mV common mode
16	CMRR SIN 50Hz	Sine wave, 50 Hz, 10mV common mode

No.	Waveform	Description
17	CMRR SIN60Hz	Sine wave, 60 Hz, 10mV common mode
18	CMRR SQR 1KHz	Square wave, 1KHz, 10mV common mode
19	NSR 30 BPM	ECG normal sinus rhythm, 30 BPM
20	NSR 60 BPM	ECG normal sinus rhythm, 60 BPM
21	NSR 120 BPM	ECG normal sinus rhythm, 120 BPM
22	NSR 180 BPM	ECG normal sinus rhythm, 180 BPM
23	NSR 240 BPM	ECG normal sinus rhythm, 240 BPM
24	NSR 300 BPM	ECG normal sinus rhythm, 300 BPM
25	VFIB	Arrhythmia, ventricular fibrillation
26	AFIB	Arrhythmia, atrial fibrillation
27	SVT	Arrhythmia, supraventricular tachycardia
28	VTACH	Arrhythmia, ventricular tachycardia
29	PVC	Arrhythmia, pre-ventricular contraction
30	ASYS	Arrhythmia, asystole, random baseline

[bpm] This argument can be specified only for the NSR simulations #19 through #24 only. It enables a limit test which implicitly uses the BPM rate of the specified simulation as the target value (see *Figure 37*, *Figure 47*, *Figure 49*, *Figure 53*, page 34, 41, 43). If **bpm** is specified for simulations #1 to #18, or #25 to #30, the argument will be ignored.

[delta_bpm] Sets the pass/fail tolerance for **bpm** mode. If not specified, a default tolerance of ± 2 BPM is applied.

The **ecgsim** statement saves information to the Test Record. The ECG waveform, the pass/fail result and comments or data entered by the user are recorded.

Examples: // Set simulator to SQR 2Hz, pass/fail test
 ecgsim 1

 // Set simulator to AFIB, pass/fail test
 ecgsim 26

 // Set simulator to ECG 60BPM, enable BPM rate test with
 // default tolerance of +/- 2 BPM
 ecgsim 20 | bpm

 // Set simulator to ECG 180BPM, enable BPM rate test with
 // tolerance of +/- 3 BPM
 ecgsim 22 | bpm | 3

4.12 run Statement

Syntax: **run <app_name> | [test]**

UI Reference: Section 3.3.8

Description: The **run** statement launches a vPad App from a Checklist. The App returns its results as a formatted text file, which is then copied into the vPad-Check Test Record.

In most cases, both arguments of the **run** statement will need to be specified. The arguments are:

<app_name> This is the vPad application to be launched by vPad-Check. This argument may be **vPad-AS**, **vPad-A3**, **vPad-A6**, **vPad-NFPA**, **vPad-ES**, **vPad-353**, **vPad-623**, **vPad-RF**, **vPad-A1** or **vPad-Cal**. Future vPad Apps may be callable from vPad-Check via the **run** statement, and the same syntax will generally apply to such future Apps.

[test] For **vPad-AS**, **vPad-A3**, **vPad-A6** or **vPad-NFPA** Apps, the **[test]** is an AutoSequence number (1, 2, 3...) to be executed by the App. If not specified, then AutoSequence #1 will be executed by default.

For the **vPad-ES** App, the **[test]** is a manual test type, which may be any one of the following:

Voltage
Load current
Resistance
Case leakage
Lead leakage
ECG simulator

If not specified, then the voltage test will be performed by default.

For **vPad-353** or **vPad-623** Apps, the **[test]** is a manual test type, which may be any one of the following:

Voltage
Load current
PE resistance
Insulation resistance
Leakage current
ECG simulator

If not specified, then the voltage test will be performed by default.

For the **vPad-RF** App, the **[test]** is the name of an existing RFA AutoSequence file on the tablet. The path to file is not to be included, however, the **.rfa** file extension must be specified. A file must always be specified as there is no default.

For the **vPad-A1** App, the **[test]** is either the name of an existing XML AutoSequence file on the tablet; or, a tab in the app at which to start testing. For an AutoSequence, the path to file is not to be included, however, the **.xml** file extension must be specified. A file must always be specified as there is no default. Alternately, when pointing to a tab, **[test]** is 'perform: <tab>', where valid tabs are main, ecg, resp, temp, spo2, ibp, nibp.

The **[test]** argument is not used when launching **vPad-Cal**.

Future Apps may potentially use this argument to represent something other than an AutoSequence number, a filename or a tab, and will be specified when such Apps become available.

Examples:

```
// Do a safety test with vPad-AS; run autosequence #4
run  vPad-AS | 4

// Do a safety test with vPad-A6; run autosequence #16
run  vPad-A6 | 16

// Do a safety test with vPad-NFPA; run autosequence #7
run  vPad-NFPA | 7

// Do a case (chassis) leakage test with vPad-ES
run  vPad-ES | Case leakage

// Do an insulation resistance test with vPad-623
run  vPad-623 | Insulation resistance
```


vPad-Check OPERATORS MANUAL

```
// Do a Conmed 5000 PM inspection and test with vPad-RF
run  vPad-RF | ConMed System 5000 PM Inspection.rfa

// Do a Philips MP40 performance test with vPad-A1
run  vPad-A1 | Philips Intellivue MP40.xml

// Do a SPO2 performance test with vPad-A1
run  vPad-A1 | perform:spo2

// Get test equipment information using vPad-Cal
run  vPad-Cal
```

4.13 Remote Control Statements

4.13.1 xopen Statement

Syntax: **xopen** <target_device> | <address> | [**setup_delay**]

UI Reference: Section 3.3.9.2

Description: The **xopen** statement opens a connection to a remote test instrument by means of a corresponding XBUS Smart Cable (XSC) interface.

This statement must be the first remote control statement of a Checklist and as such the **xopen** statement will proceed any **xctrl** or **xtest** statements that are used. If a Checklist includes a second **xopen** statement, it will effectively close the first connection and open a new one; in this way a Checklist may interface with multiple testing devices within the same procedure, although with only one device at a time.

Each XSC adaptor is designed to interface vPad to a specific **target_device**. An exception is the MBOX interface (part no. 7400-105 (wireless) or -106 (serial)) which may be used for basic send/receive communication with any test instrument that supports remote control operation through a serial port.

The arguments of the **xopen** statement are:

target_device This is the name or ID of a test device to be controlled by the Checklist. The name is arbitrary, but it is required and must be specified in the statement. Typically, **target_device** will be the model of the test instrument or simulator which is under control, or a nickname or other ID which corresponds to the remote device. Since this argument is really a string, it can be optionally enclosed in double-quotes for better readability with a source code editor.

address This is a single alphabetic character (A, B, C... Z, a, b, c... z) which specifies the XBUS address of the XSC interface. The **address** character may be found in user documentation which is provided with the XSC interface. This is a required argument.

[**setup_delay**] This argument is optional and corresponds to the maximum amount of time needed for the XSC to set up the communication interface on the

target device side. This delay is needed primarily for wireless interfaces which may require slightly longer to initialize, depending on distance and radio environment. The **xopen** statement waits until the **setup_delay** expires before confirming that the XSC interface and remote test device are on-line and ready to continue with the test. The **setup_delay** is in seconds and may be set to any decimal number greater than zero. For most applications, a delay of 1 to 3 seconds will be appropriate; refer to user documentation provided with the XSC adaptor if necessary. If not specified, a default delay of 2 seconds is used.

Examples:

```
// Example #1
// Open a connection to AMPS-1 simulator via XSC interface 7400-104
// Use default setup delay of 2 seconds
xopen  AMPS-1 | A

// Example #2
// Open a connection to MBOX universal adaptor 7400-106
xopen  MBOX | M | 1

// Example #3
// Open a connection to Phase3 defib tester via Phase3 XSC interface
xopen  Phase3 | P

// Example #4
// Open a connection to Impulse 4000 defib tester
// Wireless adaptor is in use, so allow a little longer for setup delay
xopen  Impulse 4000 | i | 3.5

// Example #5
// Open a connection to Phase3, using a MBOX for comm interface
xopen  Phase3 | M | 1
```

4.13.2 xctrl Statement

Syntax: **xctrl** <string> | <command> | [cmd_delay]

UI Reference: Section 3.3.9.2

Description: The **xctrl** statement causes a remote test instrument to execute one or more operations under the control of a corresponding XBUS Smart Cable (XSC) interface. If the XSC interface is MBOX, the **xctrl** statement does not control the remote device but is instead used to configure the MBOX itself. After the command is sent, the Checklist proceeds to the next step automatically. The communication connection to the XSC must be initialized by an **xopen** statement before **xctrl** is used.

Arguments of the **xctrl** statement are:

<string> This is a required argument. This message is displayed on the screen while data communications are in progress. Usually, this string will be a description of the effect the **<command>** argument will have on the remote device which is being controlled.

<command> This is a required argument and its use will depend on whether MBOX or a device-specific XSC is being used to interface with the remote test instrument.

If MBOX is in use, the **<command>** configures the communication connection from MBOX to the test device (for example, setting the baud rate, character transmission delay and other serial port settings). In this case, the **<command>** is *not* sent to the test device but is used by MBOX itself. Refer to MBOX user manual for further details.

For other device-specific XSC, the **<command>** causes one or more control messages to be sent from the XSC to the test device to configure the device or cause the device to perform a specific function. It is possible the **<command>** may be passed directly to the test device without modification, but this is usually not the case. Refer to the user manual for the particular XSC for more information.

[cmd_delay] This is an optional argument which causes the Checklist to delay before automatically proceeding with the next step, thereby allowing a remote test device some time to execute the required operation(s). The delay is also useful in enabling the user to confirm correct operation of the test device, by means of the descriptive **<string>** which is shown on the screen during data communications. The **cmd_delay** is in seconds and may be set to any integer number greater than zero (1, 2, 3...). For many applications, a delay of 2 seconds will be suitable; refer to user documentation provided with the XSC adaptor if necessary. If not specified, a default delay of 2 seconds is used.

Examples:

```
// Example #1
// Make AMPS-1 simulator produce a static IBP level of -10 mmHg
// XSC adaptor is use is part no. 7400-104 for AMPS interface
// The control command is "161"
// Use default command delay of 2 seconds
xctrl  "Setting IBP static pressure\nto -10 mmHg..." | 161
```

```
// Example #2
// XSC adaptor is use is MBOX part no. 7400-106
// Set the baud rate of the MBOX serial port to 2400 baud
xctrl  "Set baud rate to 2400..." | }br1{ | 1
// Set MBOX character transmission delay to 100 msec
xctrl  "Set TX delay to 100msec..." | }td4{ | 1
// MBOX is now set up to communicate with remote test device
```

4.13.3 xtest Statement

Syntax: **xtest** <string> | <command> | [pass_delay]

UI Reference: Section 3.3.9.2

Description: The **xtest** statement causes a remote test instrument to execute a test procedure or a measurement function under the control of a corresponding XBUS Smart Cable (XSC) interface. If the XSC interface is MBOX, the **xtest** statement is used to send control messages to the test device and to receive measurement data sent by the device. The communication connection to the XSC must be initialized by an **xopen** statement before **xtest** is used.

Arguments of the **xtest** statement are:

<string> This is a required argument. This message is displayed on the screen while the remote test device is performing the requested test or measurement function. Usually, this string will be a description of the test or measurement that is initiated through the **<command>** argument.

<command> This is a required argument and its use will depend on whether MBOX or a device-specific XSC is being used to interface with the remote test instrument.

If MBOX is in use, the **<command>** is parsed by MBOX into a *control message* and a *protocol parameter*. The control message is sent by MBOX to the test device. Depending on the protocol extracted from the **<command>**, MBOX may wait until the test device returns a response, or alternatively MBOX may signal the Checklist to advance to the next step automatically. In the case where test data is returned by the remote device, the data is displayed and is then passed or failed manually by the user (see on page). Refer to MBOX user documentation for further information and application notes.

For other device-specific XSC, the **<command>** causes one or more control messages to be sent from the XSC to the test device to make the device to perform a specific test or measurement. The XSC then monitors the status of the test device as the test or measurement is in progress. Once test data are obtained from the remote device, the XSC

will return a result to the Checklist which may include pass/fail information. In the event the test result is assigned a "PASS" by the XSC, the Checklist may optionally advance to the next step automatically. Refer to the user manual for the particular XSC for more information.

[pass_delay] This argument is optional and is usable only with XSC interfaces that provide automatic pass/fail of test results obtained from a remote device. This argument has no effect in MBOX interface applications.

For XSC which pass or fail test results automatically, a time delay may be specified by this argument which takes effect when the test result is a "PASS". In this case, the Checklist will advance to the next step automatically after the **pass_delay** expires. The intent of the delay is to provide sufficient time for the user to recognize that the result obtained from the test device has been passed. The **pass_delay** is in seconds and may be set to any integer number greater than zero (1, 2, 3...). Typically, a delay of about 5 seconds should be adequate. If the test result is a "FAIL", or if no **pass_delay** is specified in the **xtest** statement, then the user must press the "Next Step" button to make the Checklist advance.

The **xtest** statement will save information to the Test Record if a test result is obtained from the remote test device. The data obtained from the device and the pass/fail result are recorded. The **<string>** of the **xtest** statement is also saved, to give the recorded information some context.

Examples:

```
// Example #1:
// Check output energy of a defibrillator at 100 Joules, using a test instrument
// Smart Cable in use is XSC interface for Datrend Systems Phase3 analyzer
// The Smart Cable will automatically pass or fail the energy result from Phase3
prompt  "Set defib energy to 100J\nand charge."
xtest   "Discharge defib.\nVerify energy is 100J+/-15%" | DE,1,100,15 | 5

// Example #2:
// Check output energy of a defibrillator at 100 Joules, using a test instrument
// Smart Cable in use is a MBOX, connected to a Datrend Systems Phase3 analyzer
// The user must pass or fail the energy measurement data returned by Phase3
prompt  "Set defib energy to 100J\nand charge."
xtest   "Discharge defib.\nVerify energy is 85J to 115J" | DEHT -999
```

4.14 branch and endbr Statements

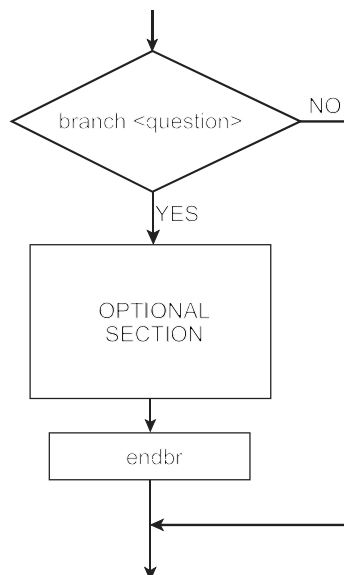
Syntax: **branch** <string> | [**label**]
 endbr [**label**]

UI Reference: Section 3.3.10

Description: **branch** and **endbr** statements work together to define an optional step or group of steps within a Checklist, where a group may include any number of statements of the kind specified in Sections 4.3 through 4.13. The **branch** statement is placed before the optional steps, and the corresponding **endbr** statement is placed after them.

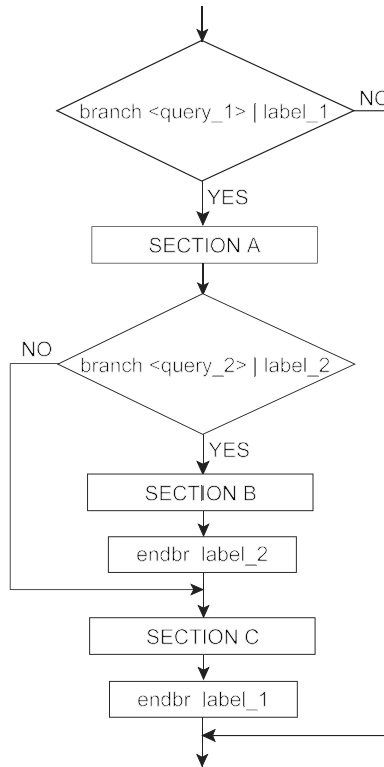
The <string> argument of the **branch** statement is intentionally chosen to be question having a "yes" or "no" answer, for example: **"Do you want to test the xxxx?"** When this question appears, users answer it by pressing either a "YES" or "NO" button. If the user responds "YES" to the question, execution proceeds with the statement immediately following the **branch**; if the user responds "NO", execution jumps forward to the matching **endbr** of the optional section, skipping all steps in between.

The [**label**] argument is optional. If not specified in the **branch**, and if the "NO" button is pressed in response to the question, then the Checklist is searched to find the first **endbr** which follows the **branch**, and execution then jumps to the first statement after the **endbr**. This type of execution may be represented by the following flow chart:



A [**label**] may be any word or phrase, and is typically chosen to be a mnemonic that

represents the optional section. If a **[label]** has been specified in the **branch**, and if the "NO" button is pressed in response to the question, then the Checklist will be searched to find the next **endbr** having a matching **[label]**. Execution will then jump to the next statement after the matching **endbr**. Labels can be used to "nest" branches within one another, enabling arbitrarily complex program flows within a Checklist. This flowchart illustrates the concept of labelled branching:



In response to a "NO" keypress from the user, branches always jump *forward* to a subsequent step in the Checklist; it is not possible to branch backward to a prior step. Instead, "reverse" jumps in program flow may be implemented with **begin** and **loop**, as described in the following section.

Example: The following Checklist excerpt illustrates how **branch** and **endbr** may be used in combination with other vPL statements. The example pertains to a ventilator which may or may not be supplied with an oxygen tank connected to it.

```
branch "Does the equipment have an oxygen tank attached?" | O2_TANK
branch "Is the valve on the tank set to 'ON'?" | VALVE
check  "Set valve to 'OFF' position before disconnecting O2 tank"\+
      | normal | "DONE"
endbr  VALVE
prompt "Disconnect oxygen tank from equipment"
check  "Inspect oxygen fitting on tank"
check  "Inspect oxygen fitting on equipment"
prompt "Move oxygen tank to a safe location"
endbr  O2_TANK
prompt "Equipment is ready for further inspections"
```

4.15 begin and loop Statements

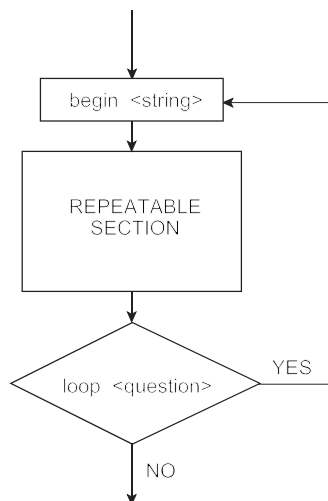
Syntax: **begin** <string> | [style]
 loop <string>

UI Reference: Section 3.3.10

Description: **begin** and **loop** statements work together to define a repeatable step or group of steps within a Checklist, where a group may include any number of statements of the kind specified in Sections 4.3 through 4.13. The **begin** statement is placed at the beginning of the repeatable group of steps, and the corresponding **loop** statement is placed at the end. The steps within the repeatable group are performed at least once, and the **loop** statement displays a dialog which provides the opportunity to repeat them.

The **begin** statement functions similarly to a **header**; when the statement is executed, the statement's <string> argument is copied to the Test Record at the current step.

The <string> argument of the **loop** statement is intentionally chosen to be question having a "yes" or "no" answer, for example: **"Do you want to repeat the xxxx test?"** When this question appears, users answer it by pressing either a "YES" or "NO" button. If the user responds "YES" to the question, execution jumps backward to the preceding **begin** statement; if the user responds "NO", execution continues with the statement following the **loop**. The program flow may be represented by the following flow chart:



Unlike branches, Checklist loops cannot be nested. A Checklist, however, may include any number of separate loops. One or more branches, nested or otherwise, can also be contained *inside* a loop.

Example: The following Checklist excerpt illustrates how **begin** and **loop** may be used in combination with other vPL statements. The example pertains to an inspection for an ultrasonic transesophageal echocardiograph, where the equipment may be supplied with at least one probe, or possibly more than one.

```
begin      "ULTRASONIC TEE PROBE INSPECTION" | bold
check      "Check TEE probe tip for physical damage"
check      "Check TEE probe connector for damage or bent pins"
prompt     "Perform leakage test (TEE leakage tester required)"
limtst     "Measure TEE probe leakage.\n\n"+
           "Confirm leakage is less than 400 uA" | uA | max | 400
loop      "Inspect another ultrasound TEE probe?"
prompt     "Clean and dry the TEE probe(s) and return to service"
```

4.16 IF_ELSE [if_br/else_br/end_if_br] Statements

Syntax: **if_br** <string> | [**yes_text**] | [**no_text**] | [**label**]
 <action1>
 else_br [**label**]
 <action2>
 end_if_br [**label**]

UI Reference: Section 3.3.10

Description: **if_br**(anch), **else_br**(anch) and **end_if_br**(anch) statements work together to define two groups of steps within a Checklist, where a group may include any number of statements of the kind specified in Sections 4.3 through 4.13. The **if_br** statement is placed before the first group of steps (see <action1> above), the **else_br** is placed before the second group of steps (see <action2> above), and the corresponding **end_if_br** statement is placed at the end of the second group of steps.

The <string> argument of the branch statement is intentionally chosen to be a question having a "yes" or "no" answer, for example: "Test version 1.03 firmware - press Yes?\n Else test newer firmware - press No" When this question appears, users answer it by pressing either a "YES" or "NO" button. If the user responds "YES" to the question, execution proceeds with the statement immediately following the **if_br**; if the user responds "NO", execution jumps forward to the matching **else_br** section, skipping all the **if_br** steps in between.

Default labels associated with the Yes and No buttons on the vPad screen are "Yes" and "No", respectively. These buttons may be optionally assigned alternative labelling via the [**yes_text**] and [**no_text**] arguments respectively. These arguments must be chosen such that the [**yes_text**] is a response which logically points to the **if_br** section of the code (e.g. "YES", "Standard Ed.1", "Version A", etc), and the [**no_text**] is a response which logically points to the **else_br** section of the code (e.g. "NO", "Standard Ed.2", "Version B", etc).

A [**label**] may be any word or phrase, and is typically chosen to be a mnemonic that is descriptive of the section. If a [**label**] has been specified in the branch, and if the "NO" button is pressed in response to the question, then the Checklist will be searched to find the next **else_br** having a matching [**label**]. Execution will then jump to the next statement after the matching **else_br**. Labels can be used to "nest" branches within one another, enabling arbitrarily complex program flows within a Checklist.

In response to a "NO" keypress from the user, branches always jump forward to a subsequent step in the Checklist; it is not possible to branch backward to a prior step. Instead, "reverse" jumps in program flow may be implemented with **begin** and **loop**, as

described previously.

Example: The following Checklist excerpt illustrates how `if_br`, `else_br`, and `end_if_br` may be used in combination with other vPL statements. The example pertains to a monitor which may or may not have a grounded power cord, so the safety test can change.

```
if_br  "Does the equipment power plug have a ground pin?" | | | PE_TEST
branch "Does the device have Applied Parts?" | AP_TEST
check  "Connect all Applied Parts to appropriate terminals on \+
the safety analyzer" | normal | "DONE"
endbr  AP_TEST
run vPad-NFPA | A1

else_br | PE_TEST
prompt "Provide a ground plane for device to sit on"
branch "Does the device have Applied Parts?" | AP_TEST
check  "Connect all Applied Parts to appropriate terminals on \+
the safety analyzer" | normal | "DONE"
endbr  AP_TEST
run vPad-NFPA | A2

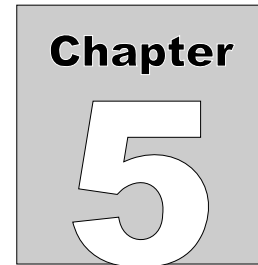
end_if_br | PE_TEST
check  "Inspect the chassis"
check  "Inspect accessories"
prompt "Equipment is ready for further inspections"
```

Alternately,

```
if_br  "Which Safety Test?" | "NFPA A1" | "NFPA A2" | PE_TEST
branch "Does the device have Applied Parts?" | AP_TEST
check  "Connect all Applied Parts to appropriate terminals on \+
the safety analyzer" | normal | "DONE"
endbr  AP_TEST
run vPad-NFPA | A1

else_br | PE_TEST
prompt "Provide a ground plane for device to sit on"
branch "Does the device have Applied Parts?" | AP_TEST
check  "Connect all Applied Parts to appropriate terminals on \+
the safety analyzer" | normal | "DONE"
endbr  AP_TEST
run vPad-NFPA | A2

end_if_br | PE_TEST
check  "Inspect the chassis"
check  "Inspect accessories"
prompt "Equipment is ready for further inspections"
```



5 Upgrades and Maintenance

Software Applications (Apps) may be upgraded from time-to-time to provide enhanced features or improvements. Contact Datrend or your local dealer for information on these upgrades.

In some cases, the changes made to an App may be to correct operational issues that have come to the attention of DSI. These shall generally be referred to as maintenance upgrades. In other cases, the changes may be to improve performance or add features which would enhance the overall functionality of the App. The decision to apply a charge to any upgrade shall be solely at the discretion of DSI.

Datrend Systems Inc.
Unit 130 - 4020 Viking Way
Richmond, BC • CANADA • V6V 2L4
Tel: 800.667.6557 (North America only) or
+1.604.291.7747 • Fax +1.604.294.2355
e-mail: customerservice@datrend.com
www.datrend.com

APPENDIX A. Example Checklists

Below is a listing of the "Demo Checklist" which is included with the vPad-Check App. The Demo Checklist illustrates how some of the more important keywords of the vPad Programming Language (including **prompt**, **check** and others) are used. Many other Checklist examples are provided with the vPad-Check App. Checklists may be viewed or modified with any word processor or text editor, including the text editor installed on the Android tablet. *vPad-IDE* software, available for download at no charge from www.datrend.com, provides a full-featured environment for creating, modifying and managing Checklist programs.

```
// FILE: Demo Checklist.vpl
// Date: April 23 2015
// Version: 2.0
//
// A demo checklist providing examples of VPL statements, including
// "prompt", "check", "limtst" and others.

prompt  "DEMONSTRATION\nCHECKLIST" | bold

prompt  "A procedure for equipment inspection and/or maintenance can be "\+
        "implemented as a vPad 'Checklist'.\n\n"\+
        "A Checklist is made up of steps which are called 'statements'.\n\n"\+
        "This demo shows some examples of the statements which are available "\+
        "for creating Checklists, and how these work..." | medium

prompt  "'Prompt' statements are typically used\n\n"\+
        "to display messages or instructions"

prompt  "Prompts can be displayed\n\n"\+
        "in BOLD font, like this..." | bold

prompt  "...or in RED font\n\n(useful for warnings)" | red

prompt  "Prompts can be accompanied\n\n"\+
        "by a 'bell' sound..." | bell

prompt  "...and important messages can be\n\n"\+
        "emphasized with an 'alert' sound" | alert

prompt  "The 'medium' font, shown here, can be used in those situations "\+
        "where there is a need to display somewhat more information "\+
        "than will easily fit on the screen using the 'normal' font." | medium

prompt  "This is the 'mmono' font. It is slightly smaller than 'medium' "\+
        "font and it is uniformly-spaced. This is useful if you want to "\+
        "display some information in a tabular layout, for example, like "\+
        "this:\n\n\n"\+
        "          Column 1          Column 2\n\n"\+
        "          Item 1           Item A\n\n"
```

vPad-Check OPERATORS MANUAL

```

"                Item 2                Item B\n"\n"+
"                Item 3                Item C\n" | mmono

prompt "A 'prompt' statement can also use 'small' font, like this. This is "\n"+
"       useful for displaying detailed instructions, for example:\n\n"\n"+
"1. First, do this step.\n\n"\n"+
"2. Next, do this step.\n\n"\n"+
"3. Next, do this step.\n\n"\n"+
"4. Finally, do this step. Since the small font is uniformly\n"\n"+
"   spaced, it is possible to created indented paragraphs\n"\n"+
"   such as the one shown here.\n\n"\n"+
"When ready to continue, press 'Next Step'..." | small

color #80FF80
prompt "Prompts default to displaying\ntext in white.\n\n"\n"+

"       This default can be changed with a\n'color' statement, as shown here. "

color #FFFFFF
prompt "A timer can be attached to a prompt, as shown above. When the timer counts "\n"+
"down to zero, the checklist will automatically proceed to the next step. "\n"+
"If you do not want to wait for the timer to reach zero, you can press the "\n"+
"'Next Step' button anytime.\n\n"\n"+
"Prompts with a timer can be used to flash a quick message, or make a checklist "\n"+
"delay until some other process, such as a test, has finished. Either way, a "\n"+
"timer is useful for adding more automation to your checklists."\n"+
| medium | 30

check "The 'check' statement is used for checking or inspecting items.\n\n"\n"+
"Follow the displayed instruction and press a radiobutton to enter your answer. "\n"+
"You will not be allowed to proceed until a 'PASS', 'FAIL' or other result "\n"+
"has been entered. If you do not choose 'PASS', you will be prompted to give "\n"+
"Press 'Next Step' to see some examples of 'check' statements..." | medium

check "Check equipment for physical damage"

check "Check line cord of equipment"

check "Check power plug of equipment"

check "Check indicators & displays"

check "Check fuses & enter ratings"

check "Enter the software version\nof the patient monitor"

prompt "The 'limtst' statement is used for tests which will involve measurements.\n\n"\n"+
"You enter the measurement into a 'Test Data' box, then you press the "\n"+
"'Check Limits' button. The data you entered is compared to upper and/or lower "\n"+
"limits which are defined in the 'limtst' statement. The measurement is then "\n"+
"assigned a pass or fail, based on the limit(s) which have been set.\n\n"\n"+
"Press 'Next Step' to see a 'limtst' example for a defib..." | medium

limtst "Set defib energy to 50J and discharge.\n"\n"+
"Measure output energy with defib analyzer\n"\n"+
"and enter below as 'Test Data'. \n"\n"+
"Measured energy should be 50J+/-15%" | Joules | target | 50 | 15 | percent

prompt "The message on the previous screen has a lot of text. But, less "\n"+
"information can be displayed if the user knows what they should do "\n"+
"when asked to perform a test and then enter the measurement result.\n\n"\n"+
"For example, press 'Next Step' to see a simpler (and more typical) "\n"+

```

vPad-Check OPERATORS MANUAL

```
"example of how to apply the 'limtst' statement..." | medium

limtst "Measure power output.\n\nCheck power = 200 to 400 uW"\n+
      | microwatts | range | 200 | 400

show "The 'show' statement can be used to\n"\n+
     "display a photograph..." | normal |\n+
     "Datrend/vPad-ES/Instructions/hp_defib_1.jpg"

show "The 'show' statement can also be\n"\n+
     "used to play a video..." | normal \n+
     "Datrend/vPad-ES/Instructions/diagnostic_ecg_video.mp4"

show "A smaller font can be used for the message displayed above "\n+
     "the 'Show' button, in the event there is a need to provide "\n+
     "a detailed explanation for the image or video that will be "\n+
     "displayed..." | medium |\n+
     "Datrend/vPad-ES/Instructions/hp_defib_3_ecg.jpg"

prompt "The next step will demonstrate the\n"\n+
       "'power' statement, which will set\n"\n+
       "the test receptacle ON"

power on

prompt "The next few steps will demonstrate operation of the ECG "\n+
       "simulator, via the 'ecgsim' statement.\n\n"\n+
       "The first example test will be a SQR 2Hz performance wave, with "\n+
       "optional test result input via pass/fail buttons..." | medium

ecgsim 1

prompt "The next example is a AFIB arrhythmia..." | medium
ecgsim 26

prompt "For NSR waveforms, a tolerance in BPM can be optionally specified "\n+
       "for the rate, which is predefined for the waveform chosen. Press "\n+
       "'Next Step' to see this example..." | medium
ecgsim 20 | bpm | 3

prompt "Next, this demo will perform a\n"\n+
       "quick safety test by running\n"\n+
       "the vPad-AS application..."

run vPad-AS | 5

prompt "Other statements can remotely control external test instruments and "\n+
       "download test results directly to vPad.\n\n"\n+
       "To make use of this capability, you will need an accessory 'Smart Cable' "\n+
       "to connect the external tester or simulator of interest to vPad.\n\n"\n+
       "To see an example of how this works, try the 'Remote Control Demo - AMPS-1' "\n+
       "checklist." | medium

prompt "That's the end of the show.\n\nSee you next time!"

// Make sure the test receptacle is OFF
power off

// End of script
```

Below is a listing of "Remote Control Demo - AMPS-1". This example illustrates how some of the remote control statements of vPL are used.

vPad-Check OPERATORS MANUAL

```
// File: Phase 3 Demo with MBOX.vpl
// Date: Oct. 25 2013
// Rev.: 1.0
//
// Demo checklist using a Phase 3 with MBOX interface for defib
// energy and other measurements. Energies tested are 5, 10, 50, 100,
// 200 and maximum energy (charge time test). Pass/fail is by manual
// entry, based on displayed acceptance ranges (15% tolerances).
// Cardioversion/sync is also tested.

prompt "DEMO OF PHASE 3\nREMOTE CONTROL VIA MBOX" | bold

prompt "Connect Phase 3 serial port to vPad\n\"+\nvia MBOX adaptor (P/N 7400-106).\"

prompt "Apply power to Phase 3"

prompt "Set Phase 3 to REMOTE mode"

prompt "During this procedure, if you need to cancel a test that has \"+\nbeen started on the Phase 3, remember to press the CANCEL button \"+\nat the lower right corner of the Phase 3's LCD display" | medium

prompt "Connect paddle plates to Phase 3\n\"+\nor use hands-free cable"

// Reset MBOX interface to defaults (9600 baud; no delay; EOC = CR)
xopen "Phase 3" | M | 1

// Set Phase 3 ECG simulator to Coarse VFIB
// Command sent to Phase 3 is "ECGDJ10"
// Protocol = -2 causes this checklist to go to the next step
// automatically when Phase 3 returns "*" handshake response
xtest "Setting ECG to Coarse VFIB..." | ECGDJ10 -2

// Apply power to defib under test
power on

// Prompt user to set the defib energy for the test
prompt "Set defib to 2 Joules"

// Get Phase 3 to perform defib energy test, low range, return data = text
// "DELT" command returns a string like this: "ENERGY=ddd.dJ"
xtest "Discharge defib.\nVerify energy = 1.7 to 2.3J" | DELT

// Repeat for 10J energy
prompt "Set defib to 10 Joules"
xtest "Discharge defib.\nVerify energy = 8.5 to 11.5J" | DELT

// Repeat for 50J energy (high range test)
prompt "Set defib to 50 Joules"
xtest "Discharge defib.\nVerify energy = 43 to 58J" | DEHT

// Repeat for 100J energy (high range test)
prompt "Set defib to 100 Joules"
xtest "Discharge defib.\nVerify energy = 85 to 115J" | DEHT

// Repeat for 200J energy (high range test)
prompt "Set defib to 200 Joules"
```

vPad-Check OPERATORS MANUAL

```
xtest    "Discharge defib.\nVerify energy = 170 to 230J" | DEHT

// Charge time test at max energy (high range test)
prompt   "Set defib to run on battery power, and select maximum energy"
xtest    "Charge up & discharge.\nCheck charge time < 10 sec" | DCHT

// Set Phase 3 ECG simulator to Coarse AFIB
xtest    "Setting ECG to Coarse AFIB..." | ECGDA10 -2

// Cardioversion test (high range test; sync to Q-wave)
prompt   "Set energy to 100J and\nenable SYNC"
xtest    "Discharge and check\nSYNC is 20 to 64 msec" | DSHQT

prompt   "End of Phase 3 demo"

// Power down the equipment under test
power    off

// End of checklist
```

vPad-Check OPERATORS MANUAL

