Effective: July 13 2020
Supersedes: UG-PHD-1040001-201912-019, Ed. 04/01/2020

PHD User Guide

Application Packager

ENGINEERING YOUR SUCCESS.

Parker Hannifin Canada
Electronic Controls Division
1305 Clarence Avenue
Winnipeg, MB R3T 1T4 Canada
office +1 204 452 6776
Fax +1 204 478 1749

http://www.parker.com/ecd

Contents

Publication History

The following table provides an overview of the changes made to this document over the course of its publication history.

Release	Description of Change, Date
Rev. 001	First release of this document, 02/21/2018
Rev. 002	Add Generic CAN note, 06/11/2018
Rev. 003	Add encoder information, update images, 10/23/2018
Rev. 004	USB cable note, update images, 06/04/2019
Rev. 005	Add Command Line Interface section to Chapter 2, 09/05/2019
Rev. 006	Minor text edit to section 2.5.5, update images, 04/29/2020

1. Introduction

These instructions are to be used as a reference tool for the vehicle manufacturer's design, production, and service personnel.
The user of these instructions should have basic knowledge in the handling of electronic equipment.

1.1. Overview

The PHD family of displays are general purpose color displays with capacitive touchscreens suitable for a wide range of industry applications.
This document describes how to configure the Network Connection for a PHD and use the PHD Application Packager to create an Application Package file containing all necessary files needed to execute the application in a Parker PHD display.

1.1.1. Terminology

The abbreviations and acronyms used in this manual are defined in the following table.

Abbreviation	Explanation
PHD	Parker Hannifin Display
.phd	File extension of Application Package file
Crank	Display page graphics creation software, Crank Storyboard
.gapp	File extension of Crank Storyboard embedded engine file
USB	Universal Serial Bus
DLA	Data Link Adapter

1.1.2. Product documentation

The following publications are relevant for users of this product.

- PHD Catalog datasheets MSG33-5021/US thru MSG33-5023/US
- PHD User Guide/Instruction book MSG33-5021-IB/US
- PHD API Reference MSG33-5021-M3/US
- PHD J1939 Stack Generation MSG33-5021-M4/US

All documentation may be found on our web pages, located at www.parker.com/ecd. Contact the manufacturer if there is anything you are not sure about or if you have any questions regarding the product and its handling or maintenance.
The term "manufacturer" refers to Parker-Hannifin Corporation.

2. PHD Application Packager

2.1. Getting started

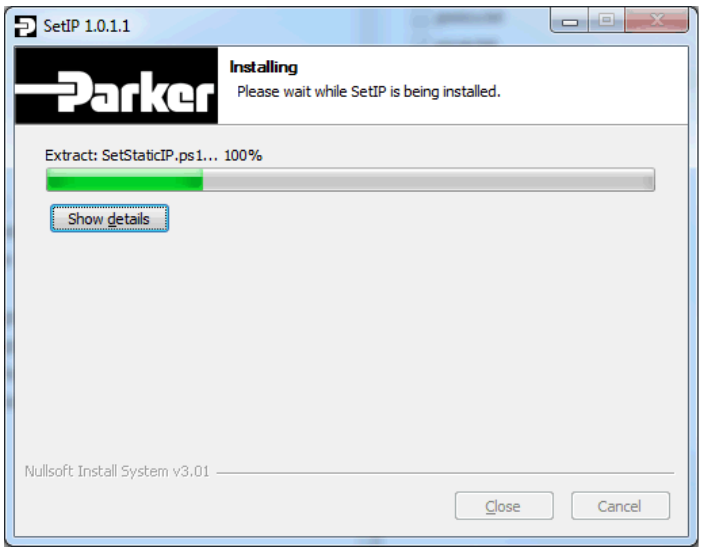
In order for the Parker PHD product interface to be shown in the “Network Connections” window on your computer, the following two steps need to be completed:

- Install the Parker PHD Application Packager, version 1.0 or later.
- Connect the PHD to the computer with a USB cable and power up.

The PHD will auto-connect as a USB device on boot up the first time it powers up. The PHD’s USB mode can be changed to device or host when installing an Application Package file.
Note that faulty USB cables can corrupt the communication between the PHD and laptop. If the cable is faulty, the laptop won’t recognize the PHD device and Windows will show a fault message stating that either an unknown device is connected or the connected device was not recognized. If this occurs double check the USB cable and shielding.
The PHD Application Packager software is used to create an Application Package file consisting of all the files needed to execute the application software on the PHD displays. These files include the Crank files, the CAN configuration files and the I/O configuration files.
In order to create a PHD Application Package file, run the Parker PHD Application Packager Tool.

2.2. Network Interface

During the PHD Application Packager software installation, the User Manual will be installed the same directory chosen for the Packager files. In addition to the User Manual, the application *SetIP.exe* is installed. This application is used to automatically configure the Network settings to work properly with the PHD and the Application Packager tool. SetIP.exe must be run to configure IP addresses and settings for the IP Ports. These settings are automatically configured by running the SetIP application. There are no user defined settings required for this, simply run the application and when the installation progress bar is complete, the settings are installed. After SetIP finishes running, it is necessary to reboot the PHD unit.



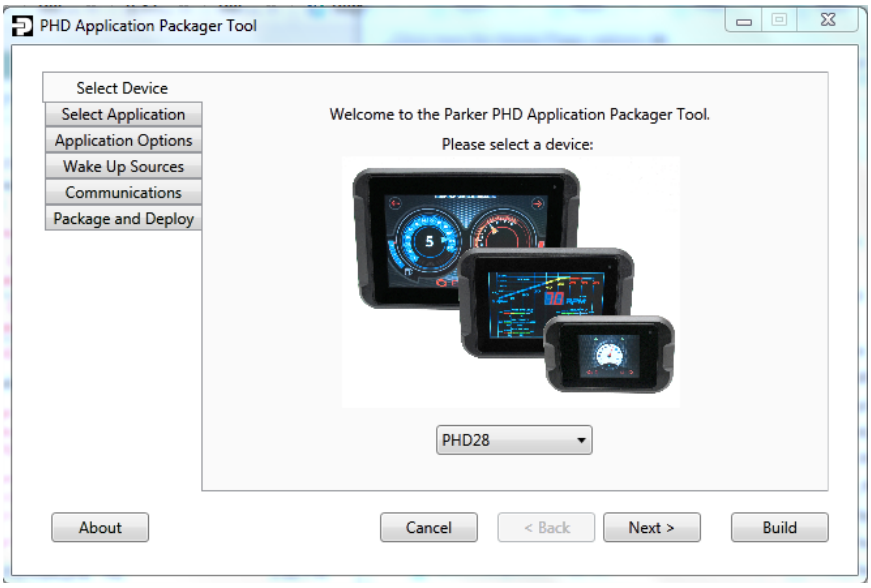
SetIP.exe window

Once it is finished, simply click “Close” to complete the setup.

Note: The SetIP application must be run after the PHD has been plugged into an USB port and powered on. If the PHD is plugged into a different USB port, the SetIP application must be run again to configure that USB port. This also applies when changing to a different version of the PHD. For example, do this if a PHD50 has been programmed for the first time and now a PHD70 is going to be programmed for the first time.

2.3. Application Package file configuration

The configuration of the PHD file consists of four basic steps. Each step allows the user to enter settings, or select the appropriate files to create the complete package for the PHD application. The completed Application Package file has the “.phd” extension. To begin, open the PHD Application Packager.



Start up screen

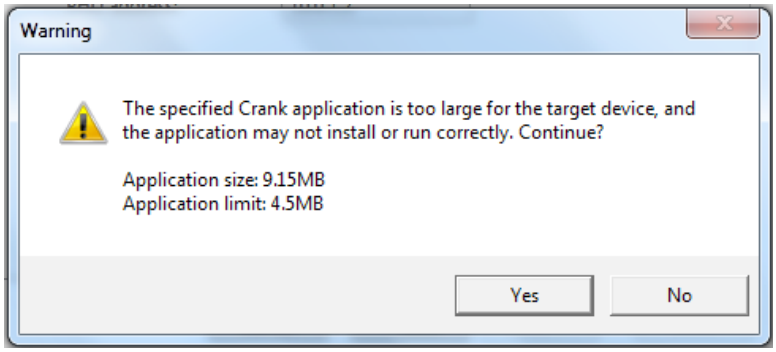
To begin the configuration, select the PHD version you will be programming and then click “Next” to get to the Select Application screen. Selecting the PHD version will enable the correct options for that version.

2.4. Select Application

The *Select Application Screen* is the interface used to select the Crank application file, or a previously built PHD application file. A prebuilt application file is created using the packager and will have the “.phd” extension. If a prebuilt PHD application file is selected, the option to select a Crank application file and the associated setting will be disabled. The crank application file is the Crank Storyboard Embedded Engine file created using the Crank software and is designated with a “.gapp” extension. Please refer to the Crank Software User manual, instructions and training materials on how to create the Crank Storyboard Embedded Engine File.
It is important to take note of the Crank Storyboard Development Environment used to generate the Embedded Engine File (.gapp) and make sure its version matches the Storyboard engine version installed in the PHD display. If the versions don’t match, generally no error message will be generated. However, the new functions included in the more recent Storyboard Development

Environment version won't function in the PHD display.
It is important when building your application to take note of the overall size of the application package. The application package will include all the files in the directory containing the GAPP file, as well as any subdirectories, including the image files, Crank GAPP file, language files, etc. The maximum allowable application size for each PHD type is noted below. In addition, the Packager will check the application size before installation and warn you if it exceeds the limit. Note that if you select to proceed with the download, it may lock up the PHD unit from operating since the application is too large.

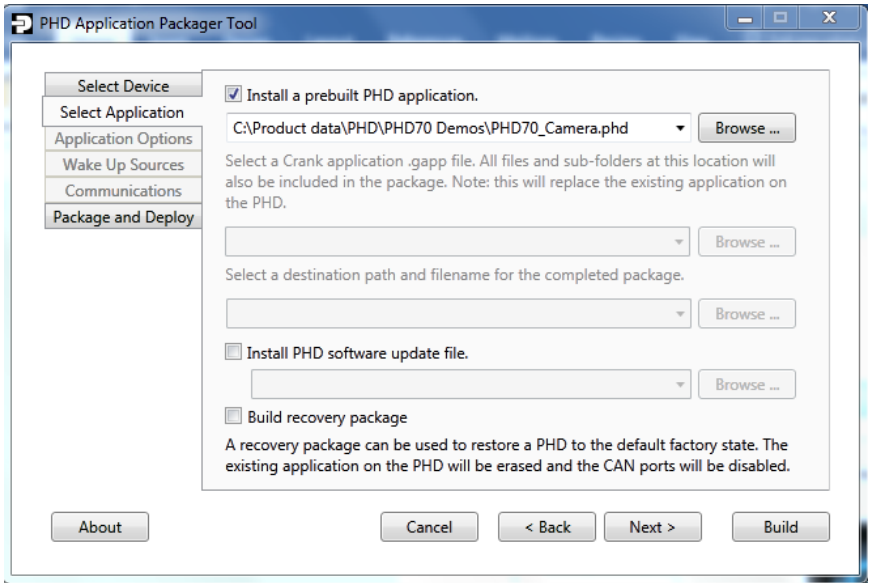
Display Type	Max. Application Size
PHD28	4.5MB
PHD50 and PF2	9.0MB
PHD70 256 MB	143MB
PHD70 1.0 GB	768MB



Application size warning

2.4.1. Destination path

This screen also allows the user to select the desired location for the completed PHD Application Package file, or PHD Package. The PHD Package will be the resulting file created by the PHD Application Packager, to be loaded into the PHD. The PHD Application Package file will be designated with a ".phd" extension.



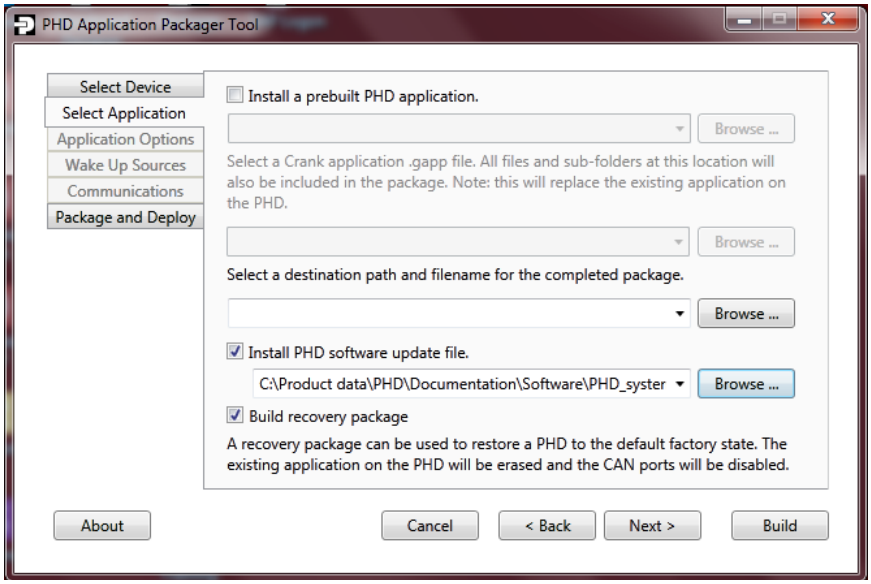
Select Application screen

2.4.2. Software update

Another option on this screen is to select a PHD software update file to be included in the Application Package. A software update file is provided by Parker and can be installed in a PHD to update one or more software components, such as the PHD services (Nexus, J1939) or can update the U-Boot or the kernel.
PHD application and system software update files can be installed either from the PC when the PHD is in Device mode, or from a USB Stick when the PHD is in Host mode. If the application update file is to be installed from a USB stick, it is important to name the file with the following format : i*nstall_phd__name__ .phd*, where *name* is the variable name of the file and is the only portion that can be changed. ***Note the double underscores!*** What is between the sets of double underscores is what will show up on the PHD as the file name on the PHD screen. The PHD will look for all the file names on the USB Stick root directory of this format and show them on a corresponding list on the PHD screen for the user to select which file to load when the application is programmed to do so. If the application update file is to be loaded from a PC to a PHD in Device mode, the file can be any file with the .phd extension.
When installing a system update, in either Device or Host mode, it must be in the following format: PHD_system_update__*name*__.*phd*, where *name* is the variable name of the file and is the only part that can be changed. ***Note the double underscores!*** What is between the sets of double underscores is what the PHD will recognize as the update file. If anything in the file name is changed, other than the *name* string, the update will not be be applied and no warning will be issued.
Also, USB sticks must be formatted in FAT32. Other formats, such as NTFS, are not recognized by the PHD.

2.4.3. Recovery package

The final option on this screen is for building a recovery package. A recovery package will restore the PHD to the factory default application. This can be used if a customer has installed an application on their PHD but the application does not work for some reason. A recovery package does not include a user's application, it can only be used to restore the application originally installed in the PHD when it was programmed at the factory. A recovery file can be loaded either from a PC or from a USB stick. To be loaded from a USB stick, the PHD must have been set to Host mode. To be loaded from a PC, the PHD must have been set to Device mode.
If a recovery file is needed to recover the PHD into a working state during development, select a destination path and filename for the recovery package. For recovery using a PC, the file name can be any .PHD filename, located on the PC. For recovery using a USB stick, the file must be located in the root directory and make certain to name the file *factory_erase.recovery*. Also, check the "Build recovery package" check box. Once the recovery package has been created and stored, copy the file to the root directory of the USB stick. In order to ensure the application package is found it must be named *factory_erase.recovery* on the root directory of the USB stick. When you plug the USB stick into the PHD, it will auto- locate the recovery file, move it from the USB stick to the PHD, decrypt it and install it. The recovery file must be created with the same key as the target PHD.



Build recovery package

It may take several minutes for the recovery file to be copied and installed. After the PHD has found and installed the recovery file, it will reboot in the default minimal application. After this step, either a USB stick or PC can be used to download a more complete application to the PHD. It is important to note that the PHD will still be in the last USB mode it was configured for. The recovery package does not change the USB mode.
To proceed to the Application Options screen, click “Next”, or click “Back” to go back to the Start up screen.

2.5. Application Options

On the *Application Options* screen, there are several selections that affect the operation of the PHD.

2.5.1. Display orientation

The first option is the *Display Orientation*. It is important to note that this selection determines how the screen is drawn on the PHD. This is a different setting than the Screen Orientation used in the Crank Storyboard Development Environment configuration. The display orientation selection box applies to the application, it does not apply to the Splash Screen image. The Splash Screen image loads before the application itself.
The Crank configuration determines the graphical layout and screen dimensions. It is important to consider the intended orientation both when designing the Crank Storyboard screens, as well as when creating the PHD Application Package.

2.5.2. Crank sbengine options

The “Crank sbengine options” text field allows the user to enter command line parameters for sbengine when it is run on the PHD. These command line parameters are provided by Crank and will configure the various options for Crank features on the display itself. A list of the valid command line parameters are in the Crank Storyboard help file. These parameters are the same parameters that can be entered when running the Crank application from the Command Line execution on the PC.
For example, by adding “-ogesture,mode=single” to the Crank sbengine options, the Packager Tool

will then include those parameters in the app.sh file it generates to launch the Crank application on the PHD itself. The app.sh default command would look like:
sbengine <crank_app_name.gapp> &
But with the addition of the Crank sbengine option, it would then look like:
sbengine -ogesture,mode=single <crank_app_name.gapp> &

2.5.3. Keypad configuration

The *Keypad Configuration* allows for the configuration of the GPIO to interface with various matrix style keypads. This setting determines the I/O configuration settings, and loads the correct files into the package for the selected matrix style keypad. The keypad options available are loaded using a configuration file for each desired type of matrix keypad. These files are XML files that must be stored in the "Keypad" folder, located under the parent folder where the PHD_Packager.exe software is installed.

2.5.4. Encoder configuration

The *Encoder Configuration* allows for the configuration of the GPIO to interface with various encoders styles. This setting determines the I/O configuration settings, and loads the correct files into the package for the selected encoder signals. Encoders are only supported on the PHD50 and PHD70, so this option is grayed out if the PHD28 is selected.

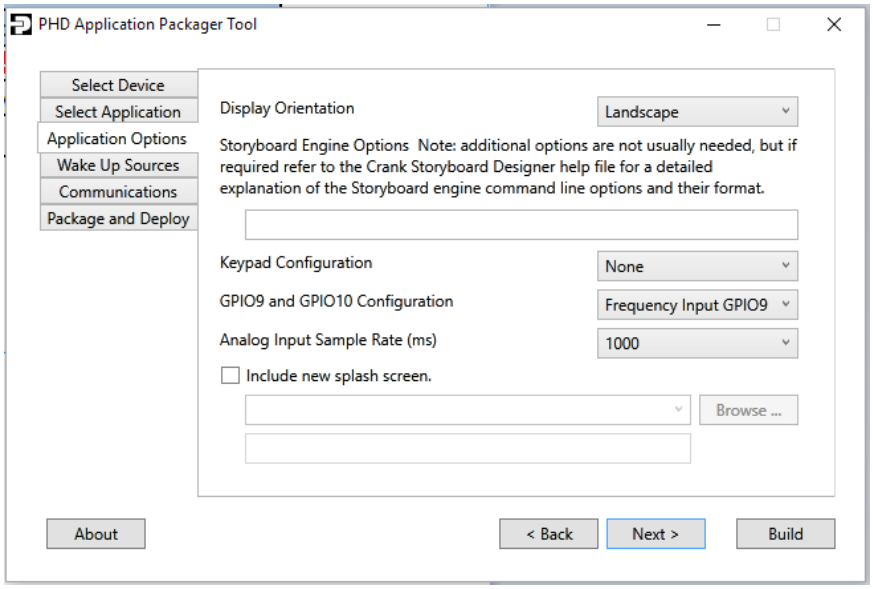
2.5.5. Analog input sample rate

This screen has a feature to set the analog input sample rate. The sample rate is in milliseconds and the valid range is from 1000 to 25 ms. A faster sample rate requires more processing by the PHD which may affect how the application performs. If the frequency input is selected for the PHD50, the analog sample rate will determine the update rate for the frequency value.

2.5.6. Splash screen

The *Splash Screen* image can also be configured in this screen. For the PHD28 and PHD50, this image must be a 16-bit RGB color .bmp (bitmap) file type. The PHD70 requires 24-bit RGB color. The image size should match or be smaller than the screen of the PHD (based on pixel count). The size and color depth of the selected image is shown below the image name. It is important to check that these values match the required values for the target display. If the selected image file and color depth aren't compatible with the selected PHD, then the values will appear in red below the file name. This is the Splash Screen image that appears when the PHD is powering up, and is separate from any screen included in the Crank Storyboard engine.
For each PHD, the LCD screen has a native orientation. By default, the splash screen image will be orientated to the native orientation. If the PHD will be mounted in a different orientation, then the corresponding splash screen image must be adjusted accordingly, e.g., the PHD28 LCD is native portrait orientation; if a landscape orientation is needed, the splash screen image must be created to be rotated at 90 or 270 degrees.

PHD unit	Native orientation	Color format	Color depth	Maximum size in native orientation
PHD28	Portrait	RGB	16 bpp	240 x 320
PHD50	Landscape	RGB	16 bpp	800 x 480
PHD70	Landscape	RGB	24 bpp	800 x 480

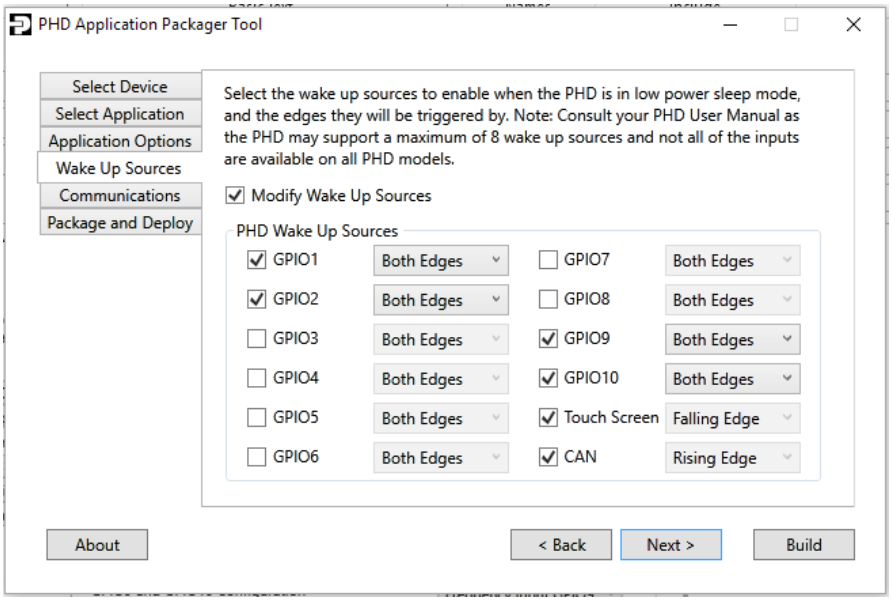


Application Options screen

To proceed to the *Wake Up Sources* screen, click "Next".

2.5.7. Wake Up Sources

In this screen, settings are provided to select which sources can be used to wake up the PHD from its low power sleep mode. This is an important setting to allow for the proper operation of the PHD displays. Consideration must be given not only to the sources that are available on the target PHD display, but also the sources that interact with the rest of the application. For example, if GPIO4 is selected as a wake-up source, but GPIO4 is attached to an input that is not powered during the wake-up cycle, the PHD will not wake up as anticipated.
When selecting the GPIO as a wake up source, you can select to trigger on either the 'rising edge' or the 'falling edge' of the signal, or both the rising and the falling edge. This allows for various sleep and wake modes to be programmed. For example, a door switch that would wake the PHD when a door was closed and then opened, or opened and then closed, or just opened, or just closed. If the GPIO is an analog signal, the rising edge would need to exceed 4 Vdc to be a rising edge trigger or fall below 1 Vdc to be a falling edge trigger.
If CAN is selected as a wake source, there is a small delay as the PHD wakes, so the CAN message that wakes the PHD will not be received. Care should be taken in the application software to ensure correct processing of CAN messages when the PHD wakes.
If a wake-up source is checked that isn't available on the target PHD, then that functionality won't work in the application, and no error message will be generated. In order to make selections, the "Modify Wake Up Sources" tick box must be checked to enable the check boxes for the wake-up sources. If "Modify Wake Up Sources" is unchecked, the wake up sources will not be changed regardless of which Wake Up Sources are checked.

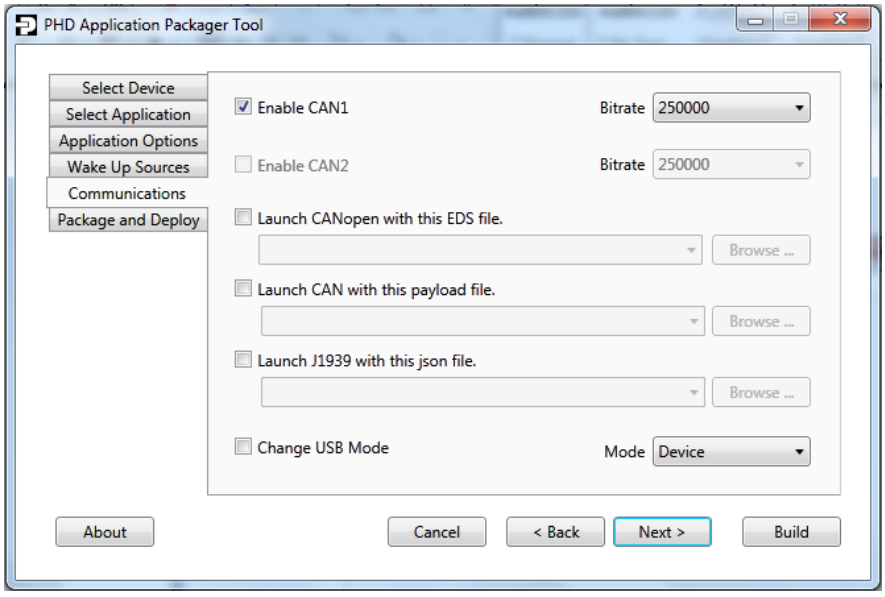


Wake Up Sources screen

To proceed to the *Communications* screen, click "Next".

2.6. Communications settings

In the *Communications screen*, the settings to enable the CAN ports as well as to configure the corresponding bit rates are shown. The option to enable CAN 2 will be grayed out if the selected PHD doesn't support the second CAN port.
Also configured in this screen is the type of CAN stack. PHD displays can support Generic CAN, CANopen (Legacy) and SAE J1939 CAN stacks. It is important to make sure that the target PHD Display contains the correct CAN stack option for the CAN stack launch file selected in this screen. If the CAN stack launch file selection doesn't match the PHD target's CAN stack option, the CAN messaging will not function properly. Care must be taken to ensure the launch file selection matches the CAN stack type in the target PHD Display. Note that when using the Generic CAN option, all CAN data will be returned and care must be taken to make sure the application isn't overloaded when attempting to process the CAN message events.
The last configuration on this screen is to configure the USB mode. This can be set to "Device" or "Host". The "Device" Setting would be for connecting to a PC and the "Host" Setting would be for connecting to a USB memory stick, or similar device. It is important to note that when the created PHD Package file is downloaded, the USB Device mode in the PHD may be changed by the file, so that it no longer communicates using the same mode as before. For example, if the PHD configuration is set to USB Host mode in this screen, it will not communicate with a PC after the PHD Package file is loaded and executed in the PHD Display. To communicate with the PHD Display, and set its USB mode back to "Device" a new PHD file would need to be created, copied to a USB memory stick and then loaded into the PHD Display.
If the PHD has been set to Device mode, it will look for a PHD application package file (*.phd) on a USB stick when the stick is inserted in the USB port.



Communications screen

When the settings are complete, click “Next” to proceed to the *Package and Deploy* options screen.

2.7. Package and Deploy settings

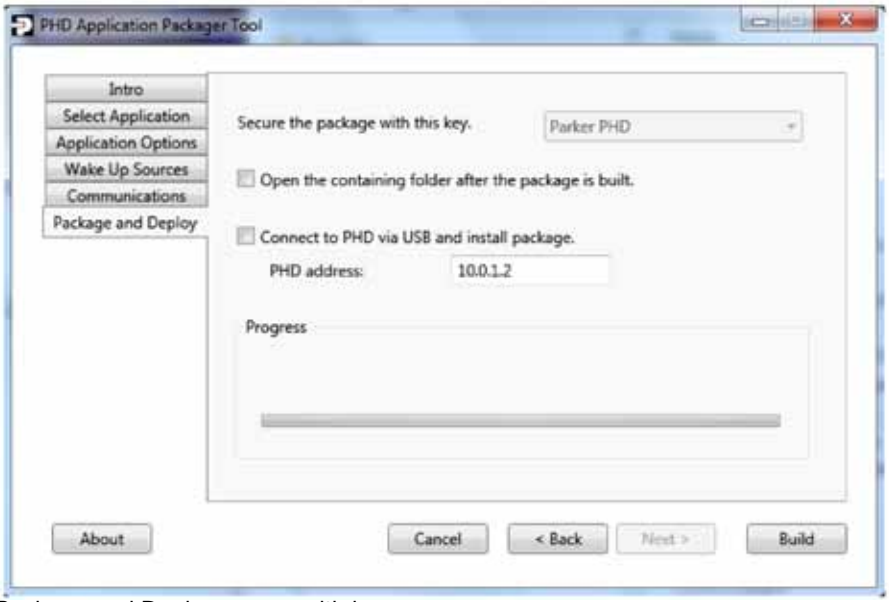
In the *Package and Deploy* screen, the file you are about to create can be password protected with a key file that locks PHD displays to certain PHD Package files.

2.7.1. Encryption key

The PHD application packages are secured to a set of PHDs using a key that is asymmetrically encrypted. This means an encryption key is used when the application package is created to encrypt the package contents. If the application packager does not have an encryption key targeting a particular PHD, the application package cannot be decrypted by that PHD. A package encrypted with a specific encryption key cannot be decrypted or installed on a PHD unless it contains the corresponding decryption key. It is also important to note that once an application package is installed on a PHD, it cannot be retrieved from that PHD, regardless of the key. The default encryption key is the Parker PHD key.
To request a key, please fill in the “Key Request Form” and submit it per the instructions on the form. However, if Parker has already issued a key for an OEM, we can only issue keys to the authorized representative of that OEM. If you need additional keys for an OEM which Parker has already issued a key to, please contact the authorized person at the OEM.
When the key file is delivered, it will come as an executable file named “Key Config - xxxxx.exe”. To install the key, simply run the executable file and it will place the corresponding key file into the correct PHD Packager directory.

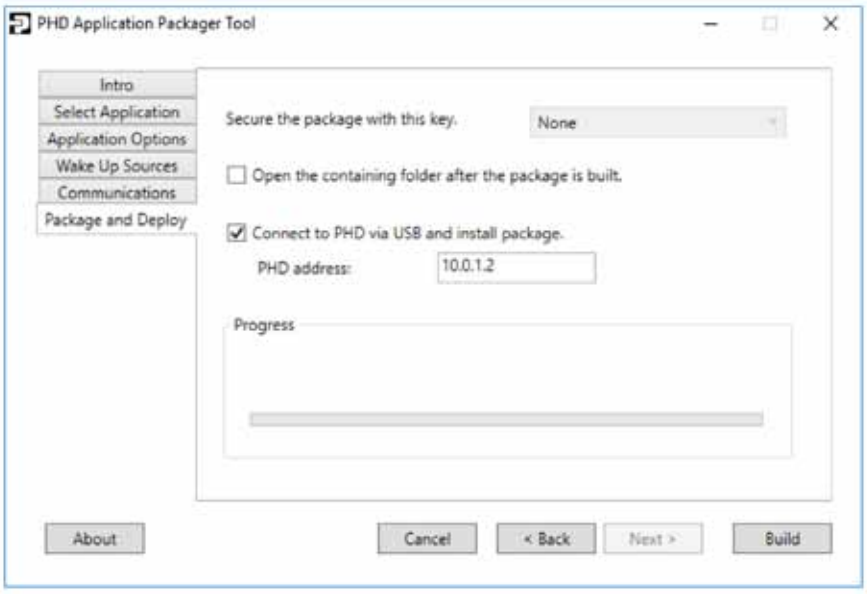
Key Config screen

After the key has been installed, the name of the key should display in the Key field on the Package and Deploy tab.



Package and Deploy screen with key

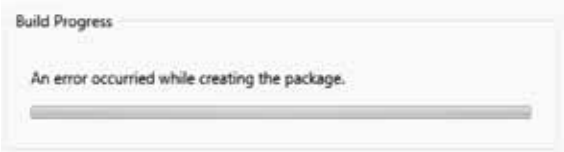
There is also a check box that allows for transferring a package file to a PHD connected to the user's computer with a USB cable. If enabled, the package file will be transferred to the PHD and installed after it has been built. The status of the package install will be shown to the user after the update process completes.



Package and Deploy screen with automatic install checked

2.7.2. Build the file

To complete the PHD Package, click the “Build” button and the PHD Package file will be completed and stored with the name you entered and in the directory you specified as the “destination path and filename” on the Select Application screen.
If an error occurs during the build process, this is most likely caused by the build routine not being able to find one or more of the selected files. This could be caused by a file name being changed, or a file being moved from one of the selected directories.



Build error message

Once the PHD Package file has successfully been configured and created, it can be loaded directly from the PC to the target PHD Display, or copied to a USB memory stick to be loaded. As noted previously, it is important to make sure the USB mode of the target PHD Display is set for the correct type (Host or Device) to enable communication with either a PC or USB memory stick.

2.7.3. Command Line Execution

The PHD Packager has a command line format that can be executed inside scripts, or in the Windows command line for automated builds in a build server or highly repeatable builds.
To create a package, configure the settings within the various tabs in order starting with the “Select Device” tab. A package will be created at the desired location containing the application as well as various other files. This package may then be placed on a USB flash drive to install to a display running in USB host mode. Another file will also be created in the same directory as the crank application called “.packinglist”. This is an XML file that contains the settings set within packager, and allows the same configuration to be loaded upon choosing the same crank application in the future.

During development, it's much more convenient to install the package directly to the display over a USB connection. To do so, follow the instructions under "How do I add software to the unit?" in W482. Then, either build a package with the option "Connect to PHD via USB and install package" enabled to install a newly created package or select "Install a prebuilt PHD application" under "Select Application" to install an existing package to a display.

2.7.4. Command Line Interface

PHD Packager may operate in a "headless" mode, allowing package build and transfer operations to be performed in a batch environment. To set up an application package to be built on the command line, a packinglist should be created for the package. The easiest way to create a packinglist is to build the package using the Packager GUI, in which the options to use can all be set using GUI controls. When created using the GUI version, the resulting .packinglist file will be in the directory as the .gapp file selected to build with. Also, the .packinglist file can also be created manually with a text editor (see "Packinglist Format").

2.7.4.1. Usage

Usage of Packager's command line interface is as follows:

```
PHD_Packager -build -c CrankApp -o Output [BuildOptions] |
             -install PackagePath [-a IPAddress] |
             -h
BuildOptions: [-p Packinglist][-u SystemUpdate][-s SplashScreen]
              [-cos CosConfig][-can CANConfig][-j J1939Config]
              [-i [-a IPAddress]]
```

The following table describes each argument:

Argument	Description
-build/--BuildPackage	Build a new package file
-install/--InstallPrebuiltPackage	Install a prebuilt package to a display
-h/--Help	Displays usage and argument descriptions
-c/--CrankApp	The Crank application (.gapp file) to build the package with
-o/--Output	Path the completed package will be saved to
-p/--Packinglist	Path to the package's packinglist file (default is directory of Crank application)
-u/--SystemUpdate	Path to a system update file to include in the completed package
-s/--SplashScreen	Path to a splash screen file to include in the completed package
-cos/--CosConfig	Path to a cos configuration (.eds) file to start cos with
-can/--CANConfig	Path to a CAN configuration (.json) file to start CAN with
-j/--J1939Config	Path to a J1939 configuration (.json) file to start J1939 with
-i/--InstallPackage	Install package to display after build is complete
-a/--IPAddress	The IP address to use when transferring a package to a PHD (default is 10.0.1.2)

2.7.4.2. Packinglist Format

The following is the structure of the PHD package's packinglist file. Note that indentation indicates element ownership, and that entries in italics indicate attributes rather than child elements.

Root
Description: Root element of all other packinglist elements.
Value: Child elements.

Application
Description: Required input/output paths for a package.
Value: Child elements.

SourceFilename
Description: The absolute path to the crank application to be packed.
Value: String.

OutputFilename
Description: The absolute path to the completed package.
Value: String.

AppOptions
Description: General application settings.
Value: Child elements.

DeviceType
Description: The type of display the package will be installed to.
Value: String ("PHD28", "PHD50", "PF2", "PHD70_256MB", or "PHD70_1GB").

PortraitOrientation
Description: Rotates the crank application for portrait orientated displays.
Value: Boolean ("true" or "false").

EncoderType
Description: Type of encoder to use.
Value: String ("X4", "X1", or "None")

KeypadFile
Description: Name of keypad file corresponding to the keypad to be connected (without the ".xml" extension).
Value: String.

AnalogSampleRate
Description: Period to use to poll analog inputs (in microseconds).
Value: Integer (>=10000, <2^31)

SBengineOptions
Description: Additional arguments to run storyboard engine with.
Value: String.

SplashScreen
Description: Absolute path to a custom splash screen to display when the display boots (Note: Reference Packager for bitmap requirements).
Value: String.

Enabled
Description: Adds file specified in value of "SplashScreen" to package.
Value: Boolean ("true" or "false").

SystemFile
Description: Absolute path to a system update package to install along with the crank application.
Value: String.

Enabled
Description: Adds file specified in value of "SystemFile" to package.
Value: Boolean ("true" or "false").

ModifyWakeupSources
Description: Modify states of wakeup sources to match settings in the “WakeupSources” element.
Value: Boolean (“true” or “false”).
WakeupSources
Description: A collection of wakeup sources and their states.
Value: Child elements.
Source
Description: A collection of properties describing a single wakeup source.
Value: None.
Name
Description: Name of wakeup source.
Value: String (“GPIOX” where “X” is an integer from 1 to 10, “Touch”, “CAN1”, or “CAN2”).
Enabled
Description: Determines if the display should wake on the given source. Note: Wake on GPIO9 and GPIO10 is unsupported.
Value: Boolean (“true” or “false”).
Edge
Description: The type of signal edge that must be observed to wake on the given source. Note: “Falling” is the only valid edge for “Touch”, and “Rising” is the only valid edge for “CAN1” and “CAN2”.
Value: String (“Rising”, “Falling”, or “Both”).
SleepModeTimeout
Description: Amount of time the display waits in standby mode before shutting down (in seconds, PF2 only).
Value: Integer (>0, <2^31).
Communications
Description: CAN/USB settings.
Value: Child elements.
CAN1
Description: Bitrate of CAN bus connected to CAN1 (in bits/second).
Value: Integer (125000, 250000, or 500000).
Enabled
Description: Determines if the display should enable CAN1.
Value: Boolean (“true” or “false”).
CAN2
Description: Bitrate of CAN bus connected to CAN2 (in bits/second).
Value: Integer (125000, 250000, or 500000).
Enabled
Description: Determines if the display should enable CAN2.
Value: Boolean (“true” or “false”).
CANopen
Description: Absolute path to a CANopen configuration file to run CANopen with.
Value: String.
Enabled
Description: Determines if the display should run CANopen.
Value: Boolean (“true” or “false”).
GenericCAN
Description: Absolute path to a CAN configuration file to run CAN with.
Value: String.

Enabled
Description: Determines if the display should run generic CAN.
Value: Boolean (“true” or “false”).

J1939
Description: Absolute path to a J1939 configuration file to run J1939 with.
Value: String.

Enabled
Description: Determines if the display should run J1939.
Value: Boolean (“true” or “false”).

USB
Description: USB options.
Value: Child elements.

ChangeMode
Description: Change the USB mode of the target upon installation.
Value: Boolean (“true” or “false”).

HostEnabled
Description: Enable USB host mode on the target. Otherwise, enable USB device mode.
Value: Boolean (“true” or “false”).

Build
Description: Package build options.
Value: Child elements.

KeyName
Description: Name of encryption key to secure the package with.
Value: String.

2.7.4.3. Packinglist Example

```
<?xml version=”1.0” encoding=”utf-8”?>
<!--This file was created by PHD Packager Tool version 1.6.0.69-->
<Root>
  <Application>
    <SourceFilename>C:\Example\PHD\PHD50
Demos\PHD50_Demo2\PHD50_Demo\PHD50_Demo.gapp</SourceFilename>
    <OutputFilename>C:\Example\PHD\PHD50
Demos\PHD50_Demo2\PHD50_demo2.phd</OutputFilename>
  </Application>
  <AppOptions>
    <DeviceType>PHD50</DeviceType>
    <Orientation>false</Orientation>
    <EncoderType>None</EncoderType>
    <KeypadFile>None</KeypadFile>
    <AnalogSampleRate>1000000</AnalogSampleRate>
    <SBengineOptions></SBengineOptions>
    <SplashScreen Enabled=”false”>C:\Example\PHD\PHD Image
Library\Image Library\Splash Screen Examples\PHD50\Parker Splash
PHD50.bmp</SplashScreen>
    <SystemFile
Enabled=”true”>C:\Example\PHD\Updates\PHD_system_update__54_Park
er__.tar.bz2.phd.gpg</SystemFile>
```

```
    <WakeupSources Enabled="false"
Edges="23068672">1024</WakeupSources>
    <SleepModeTimeout>7200</SleepModeTimeout>
  </AppOptions>
  <Communications>
    <CAN1 Enabled="false">250000</CAN1>
    <CAN2 Enabled="false">250000</CAN2>
    <CANopen Enabled="false" />
    <GenericCAN Enabled="false" />
   <J1939
Enabled="false">C:\Examples\PHD\CAN\j1939\j1939.json</J1939>
    <USB>
      <ChangeMode>false</ChangeMode>
      <HostEnabled>false</HostEnabled>
    </USB>
  </Communications>
  <Build>
    <KeyName>Parker PHD</KeyName>
    <OpenFolder>open</OpenFolder>
  </Build>
</Root>
```

2.7.4.4. Jenkins Setup Example

When configuring Jenkins to build an application package, consider the target devices it will be installed to. It may be necessary to have a different packinglist (and hence different packages) for each device model. Once the different packinglists have been created, run PHD Packager multiple times with the same crank application and different packinglists to generate packages for the intended displays.
The following is an example of a PHD package build process:

```
set packager="C:\Program Files (x86)\Parker\ECD\PHD Packager\PHD_
Packager.exe"
cd phd50_test_app
%packager% -build -c phd50_test_app.gapp -p .packinglist_Parker_PHD50
-o ..\PHD50_test_app_Parker_PHD50.phd
```

MSG33-5021-M2/US 04/29/2020

Parker Hannifin Canada
Electronic Controls Division
1305 Clarence Avenue
Winnipeg, MB, R3T 1T4 Canada
phone 204 452 6776
fax 204 478 1749
www.parker.com/ecd

Your Local Authorized Parker Distributor