This document was prepared by the staff of Valeport Ltd., the Company, and is the property of the Company, which also owns the copyright therein. All rights conferred by the law of copyright and by virtue of international copyright conventions are reserved to the Company. This document must not be copied, reprinted or reproduced in any material form, either wholly or in part, and the contents of this document, and any method or technique available therefrom, must not be disclosed to any other person whatsoever without the prior written consent of the Company.

Valeport Limited Tel: +44 (0)1803 869292

St Peters Quay Fax: +44 (0)1803 869293

Totnes Email: sales@valeport.co.uk

Devon, TQ9 5EW Web: www.valeport.co.uk

Authorisation:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Position | Name | Date |
| Prepared by | Software Development Engineer | Dom Blount | 11/05/2021 |
| Checked by |  |  |  |
| Approved by |  |  |  |

**Table of Contents**

[1. Introduction 3](#_Toc71549256)

[2. System Requirements 3](#_Toc71549257)

[2.1 Operating System 3](#_Toc71549258)

[2.2 Screen size 3](#_Toc71549259)

[2.3 Hardware 3](#_Toc71549260)

[2.4 Development Enviroment 3](#_Toc71549261)

[3. Icons & Buttons 4](#_Toc71549262)

[3.1 Icons 4](#_Toc71549263)

[3.2 Buttons 4](#_Toc71549264)

[4. Look & Feel 5](#_Toc71549265)

[4.1 Colour 5](#_Toc71549266)

[4.2 Text & Font 6](#_Toc71549267)

[4.3 Title Bar 6](#_Toc71549268)

[4.4 Layout 8](#_Toc71549269)

[4.4.1 Settings & Pop Outs 8](#_Toc71549270)

[4.4.2 Confirmation 8](#_Toc71549271)

[4.4.3 Text Boxes, Combo Boxes & List Boxes 8](#_Toc71549272)

[4.4.4 Check Boxes & Radio Buttons 9](#_Toc71549273)

[4.5 Toolbar 10](#_Toc71549274)

[5. Error Handling & Checking 10](#_Toc71549275)

[6. Releasing Your Application 11](#_Toc71549276)

[6.1 Code Signing Certificate 11](#_Toc71549277)

[6.2 How To Code Sign 12](#_Toc71549278)

[6.3 Uploading Application to Valeport Download 12](#_Toc71549279)

[6.4 Uploading Your Application For Production 13](#_Toc71549280)

[7. Miscellaneous 13](#_Toc71549281)

[7.1 Multiple Windows 13](#_Toc71549282)

# Introduction

This document has been produced to provide guidelines on how to design & create a Valeport application. This should allow software engineers to create applications that will follow the same standards/design principals.

The majority of software developed by Valeport will be Windows Desktop applications. Whilst we also develop applications for mobile devices, due to hardware etc, the size of the different components will be different to that of the Windows application. However, the UI design work for the mobile application has already been done, and any changes should follow the same principles. For simplicity and clarity, this document refers to Windows desktop applications only, but mobile applications should follow the same design principles.

# System Requirements

## Operating System

All applications must be designed to operate on Windows 10 onwards (also see note in introduction).

## Screen size

The minimum screen resolution that applications must be designed to run on is 1024px x 768px.

The application window must be adaptive so when the application is used on various screen resolutions the regions of the forms will expand or contract to best fit the window size accordingly.

The application must also allow for resizing / scaling of the application manually or with the ‘Restore Down’ and ‘Maximise’ buttons in the title bar.

## Hardware

All software must be thoroughly tested on various computers of differing performance capabilities, as the application may function as expected on the development environment / computer, but may struggle on a slower computers. Engineers must test their application on the ‘Test Machine’ before the final release. The R&D laptop is also available for further testing.

## Development Enviroment

The latest stable release of Microsoft Visual Studio (currently 2019) must be used to create Valeport Applications in C#. Customer facing applications should be made using WPF as it allows for more design capabilities than WinForms, however, for internal only applications WinForms is acceptable. Various custom WPF and WinForms components can be found in the [C# repository](http://10.0.1.1:4435/svn/valeport_csharp/trunk), these should be used if and when appropriate.

In the event that use of another programming language is desired it must be approved and signed off on.

# Icons & Buttons

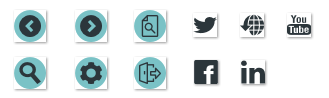
## Icons

Existing items icons for buttons should be resourced from the icons folder (G:\Engineering\Valeport Software\Standardisation Icons).

All icons must have a transparent background, or the background colour of the application.

All icons must be available in (48x48, 32x32, 22x22) pixels.

All icons must be available in black on a transparent background and white on a transparent background.

All future icons must be added to the icons folder (G:\Engineering\Valeport Software\Standardisation Icons).

By repeating to use the same icons for the same / similar functionality will allow users to move to another Valeport branded application and be able to operate it in ease and without confusion.

All new icons must be approved and signed off on by (all) other members of the Software Engineering team and/or the Management.

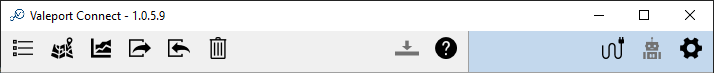
## Buttons

A button (with an icon) of 48px x 48px in size must have an icon that is 32px x 32px in size.

Buttons with no icon must be a minimum of 23px tall and fit the word/text with a minimum 10px margin to the left and right edge of the button.

There must be a minimum margin/gap of 8px between multiple buttons.

The buttons button style must also be set to flat (so only the icon is visible, no border / 3D effect). The icon of a button must be clear enough to indicate to the user that it is a button and what it does when clicked.



Button is 48px in height.

Icon is 32px in height.

Button is 48px in width.

Icon is 32px in width.

Margin between each button of 8px.

Icons folder (G:\Engineering\Valeport Software\Standardisation Icons).

# Look & Feel

## Colour

Valeport Applications should follow a minimalistic colour palette – the colours used must be easy on the eyes as users may use the application for long durations of time.

**Light blue**

RGB value: 192, 216, 237

Hex value: #c3d8ed

**Light grey**

RGB value: 240, 240, 240

Hex value: #f0f0f0

The light grey colour should be used as the main background colour of the application.

Light blue should be used to separate information or to allow a user to more easily differentiate between different segments of information (see example below).

Alternative contrasting versions of the above colours can be used, for example:

RGB value: 115, 198, 215



RGB value: 44, 54, 59



RGB value: 0, 102, 166



RGB value: 229, 236, 246



RGB value: 225, 231, 241



## Text & Font

All Valeport applications must use the **Microsoft Sans Serif** font.

The sizing of text must be appropriate to the application, with a standard size of 8.25pt, this is also the minimum size that can be used.

The title or headers must be a greater size than the main body of text and sub-headers, with a standard and minimum size of 10pt.

**Bold** text can be used to differentiate between information as headings.

## Title Bar

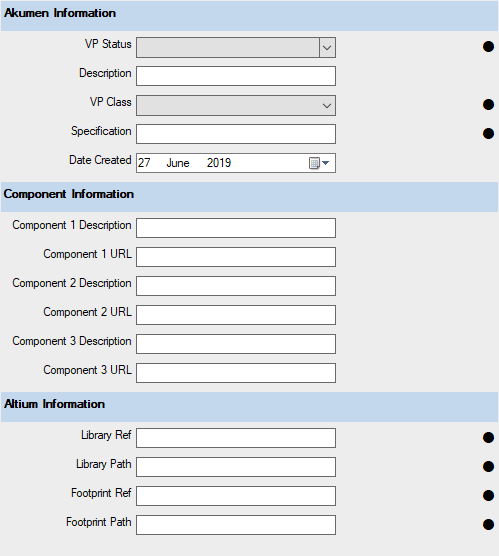
The taskbar of the application must contain the applications main icon, the name of the application and finally the version of the application. See example.



Version Number

Application Icon

Application Name



Different information fields are separated with a light blue panel.

Headers for each field are in bold.

Main background colour for the application is light grey.

## Layout

### Settings & Pop Outs

Settings screen must pop-out from the main form as another window, the minimum size for this pop-out window is 450px x 620px. Other screens that affect the operation of the software (eg connecting to an instrument) must also pop out.

If multiple windows are required for different options of a single operation (eg Settings may have Software Settings and Instrument Settings) a tab control element must be used to reduce the potential number of pop-out windows on the application (1 pop-out window, with multiple tab control elements).

### Confirmation

An ‘OK’ and ‘Cancel’ and if necessary, ‘Apply’ button must be placed at the bottom right of an input form for consistency.

When an edit is made (even if incomplete) the ‘Apply’ button’s state must be set to ‘enabled’ to indicate to the user that the change needs to be applied. The other tabs state will become ‘locked / disabled’ so the user’s changes can’t be lost if they changed tab/screen.

When the ‘Apply’ or ‘Cancel’ button is clicked updates will be made and all other tabs states will be set to ‘enabled’ and the ‘Apply’ button’s state will be returned to a ‘disabled’ state.

When appropriate the ‘Cancel’ button may have the ability to close the form / window.

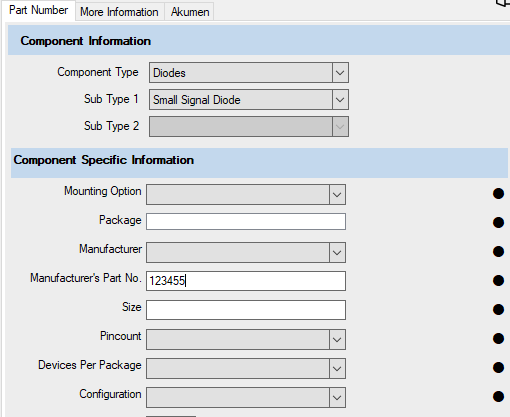
### Text Boxes, Combo Boxes & List Boxes

Entry field elements such as, text boxes, combo boxes and list boxes, must be aligned to the right of the associated label and the element should be a minimum for 175px wide.

The associated labels must be aligned to the right, this will allow for easy recognition of the desired input field.

There must be adequate spacing/margining between each individual input field. A standard of 13px with a minimum of 6px.

Tab control element used to minimize the number of windows.



Text aligned right to left.

Equal spacing between each element (13px).

Appropriate width of element for contained data (200px).

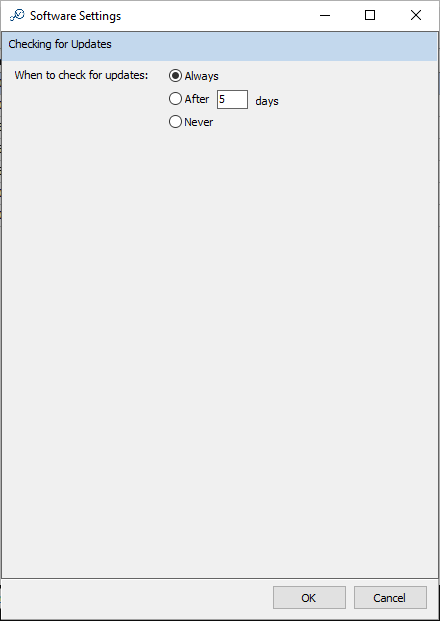
Headers is bold & separated with a light blue panel.

### Check Boxes & Radio Buttons

Check boxes must be aligned ‘top to bottom’. When a greater number of options are available than the windows height, the check box elements should descend with the use of a scroll bar.

When all usable vertical screen space has run out 2 - 3 columns of check boxes can be used to display the large amount of options.

Spacing / margining between each individual field must be 13px.



Blue background used to separate information. Header above, relevant info below.

Text aligned right to left.

Default / desired option selected by default.

## Toolbar

A toolbar should be used when appropriate to be used to navigate between multiple pages/windows within the application.

When appropriate, the number of pages/windows should be kept to as few as possible.

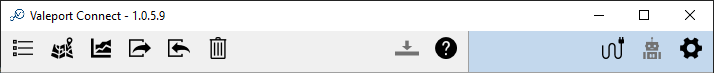
Use of the blue background can be used to separate different information/buttons. The below example separates ‘Instrument Information & Settings’ from the ‘Files & Recorded Data’.

There must be a minimum margin/gap of 8px between multiple buttons.

Icons and buttons on the toolbar must follow the same guidelines as ‘5.2 Buttons’.

The first button, navigates to the initially focused page (home button).

Standardised icons used.



Blue background used to separate information. Instrument info on the right & software info on the left.

Equal spacing between buttons

# Error Handling & Checking

Format, range checking, invalid characters, etc. must be validated when the input field loses focus (on leave of the element).

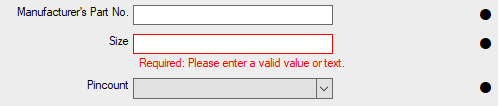
On the event that data in an entry field has been entered incorrectly, or an invalid input is made. The border of the entry field must turn to red. Underneath the entry field an appropriate and detailed error message must be made visible that informs the user why there is an error and how to fix it.

If an error does occur the form must adjust so that the error is visible to the user. For example, on a panel with a scroll bar, scroll the panel to the position of the first error.

The error message displayed should inform the user on the minimum and maximum values that could be entered in the input box, the maximum number of characters, if the information is required (therefore not an empty field), etc.

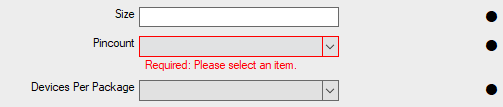
An exception to these rules will be when multiple parameters relate to each other, if all values are valid but do not match or follow different formats, all parameters should be marked as a potential error as it is unclear which parameter is at fault.

Inform the user of required fields.



Reason for the error appearing

Solution to fix the error



Custom WinForms components for text boxes and combo boxes have been made to incorporate this method of error handling. These components can be found in the [C# repository](http://10.0.1.1:4435/svn/valeport_csharp/trunk).

This repository contains multiple elements that can be referenced or imported to a Visual Studio C# Application. This repository will make maintaining and updating applications much easier, as a simple update to the repository will change all apps that reference the element.

# Releasing Your Application

Applications must be tested and signed off on by the individual(s) that has requested the creation of the software.

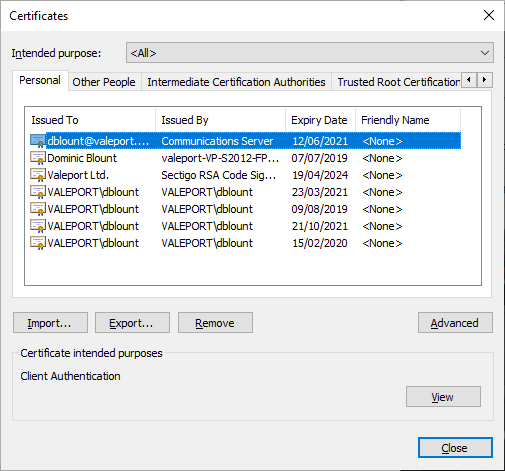
Applications can be released internally and externally using the [Valeport download webpage](https://valeport.download/). If an application is being released externally over this webpage it must be code signed to ensure that the code being sent to the customer is safe and secure.

## Code Signing Certificate

Ensure that the SSL Certificate (supplied by Sectigo) is installed on your machine, you can verify this on Internet Explorer:

Setting > Internet Options > Content > Certificates

Here you should see a Sectigo Certificate that expires 19/04/2024



## How To Code Sign

Launch ‘Command Prompt’ or ‘Developer Command Prompt for VS 2019’ and paste this command, adjusting the file location to be in line with your application:

signtool sign /tr http://timestamp.comodoca.com /td sha256 /fd sha256 /a "C:\valeport\_csharp\projects\EnviroLOG\vpEnviroLOGSetup.exe"

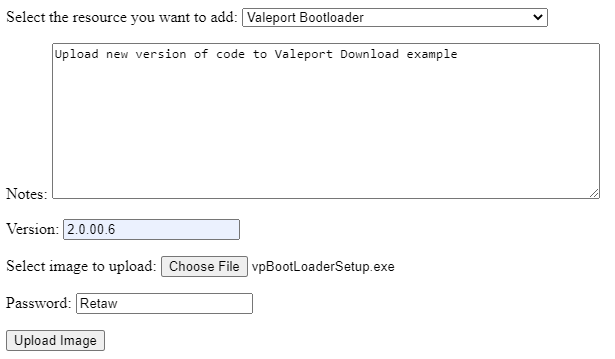
Alternatively, you can add the path to your Environment Options and open Command Prompt at the files location:

signtool sign /tr http://timestamp.comodoca.com /td sha256 /fd sha256 /a vpEnviroLOGSetup.exe

Ensure that you receive the ‘Done Adding Additional Store, Successfully signed’ message before continuing.

## Uploading to the Valeport Download Page

Once your application has been code signed, navigate to: <https://valeport.download/doupload.php>, select your desired resource, fill in the version changing in the notes box & the new version number. Finally, select the executable & use password ‘Retaw’ to upload the file correctly.



Note… there is an on-going error/bug with the Stored Procedure when performing this process when not connected to the Valeport network (i.e. when working from home), so ensure that you are either on the Valeport network or ensure that the version has been uploaded with the correct information.

### Creating A Firmware ZIP

When releasing new versions of Firmware, they must be in the ZIP format used by the Valeport Bootloader application. The ZIP file contains a hex and configure file.

The hex file can be found in the Software Design, Atmel folder under the Instrument Type folder, for example, a SWiFT’s folder would be here, G:\Production\0650\software design\Atmel.

Rename the hex file, to remove the ‘\_defragmented’ text.

Finally, a configure file is needed, named: ‘Bootload2.cfg’.

[General]

ThisHex=0650735D4.hex

ThisInstrument=SWiFT

PreviousVersCmd=#014

PreviousVersMin=0650735A9

PreviousVersMax=0650735D4

;[StoreSettings]

;#041=#042

[OtherCmds]

cmd999=#0028

cmd010=#168;20

The configure file must consist of this text, edited to match the hex files name, the correct min & max versions numbers and the instrument type. Additionally, other commands and store settings can be used, however, these should be left blank unless necessary.

These 2 files can then be converted to a zip file, matching the name of the hex file.

### Notes For Firmware

When uploading new versions of Firmware for the Valeport Bootloader application, ensure that the ‘Notes’ section details the reason for the Firmware change. A brief synopsis detailing the changes can be created using notes in the ‘Mod Notes’ document, as well as, by consulting the Firmware Engineer that pushed the update.

For example, a new version of SWiFT Firmware may be released, named ‘0650735D4’. To find the changes, this file name can be broken down into data, so we can work out the instrument type and repository folder:

0650 = Instrument Id

735 = Type

D4 = Version Number

Using these parameters the ‘Mod Notes’ document needed, would be ‘0650735SM.doc’, which can be located here: G:\Production\0650\software design\mod notes

## Uploading Your Application For Production

All customer facing applications must be made available to production, as they will provide the latest version of the application to the customer on a USB stick.

Code Signed executable but be added to G:\Production\Valeport Software\Installers

Once the new version is added to the G:\ drive, ensure that the previous version is removed. You must then email Valeport Updates to inform all relevant parties of the changes.

# Miscellaneous

## Multiple Windows

If there are multiple windows contained within the applications main window, they should be separated clearly with a white background behind each window that has a margin of at least 3px on every side.