# Overview



# Breakdown

## PB980 (Vent Side) Interface



### Webserver

Express logic has provided a webserver with ThreadX, the operating system on which the PB980 is running.

### LogWebservice

The LogWebservice is intended to provide a restful API, allowing for log data to be retrieved in binary and XML format, containing human readable log entries. For each format, the LogWebservice will be required to embed metadata such as serial number and ventilator software version.

Known Design Challenges

* The LogWebservice will not be able to give a full log in one shot because this will require a significant amount of memory. The restful API will need to support the concept of cursers/ iterators.
* The native PB980 logs are in binary format. Each log has its own format.

### DeviceInfoWebservice

The DeviceInfoWebservice is intended to provide a restful API, allowing for device identifying information to be retrieved in an XML format. This will include information about the hardware, software, firmware, options, and serial numbers.

### Download Trigger Interface

The Download Trigger interface has already been implemented. There is work required to add more security around how a download is triggered. Currently there are some concerns about the weakness or lack of security around triggering a download via the Ethernet. Additionally, the trigger message should contain a hash of the ventilator serial number to ensure that the correct device is connected. Otherwise, the interface risks being retired.

### Download Manifest Security

The download manifest does not add any security to prevent a user from downloading software to the ventilator. It is desirable to require a download key in the manifest prior to allowing a download to proceed.

Known Design Challenges

* The downloader is not able to access the serial number of the vent when in download mode. This is because the serial number is stored on a USB storage device, on the GUI CPU, which is not available to the downloader.
* Even if the serial number makes its way to flash, the corner case, were the GUI/BD CPU is replaced, presents a challenge.

### Download Status

Currently the download status reported by the ventilator only gives the percentage complete per phase of installation per file. It is desirable to also have an overall percentage complete for the whole installation.

## PB980 (PC Side) Interface



### Abstract File System Layer

The Abstract file system layer is intended to allow for a virtual mapping of files based on a URL and a download manifest provided by the laptop agent.

### Download FS Interface

The Download FS Interface provides an interface, allowing for the mappings in the Abstract File System Layer to be configured by a download notification.

### Get Log Http Webclient & Interface

The Get Log Http Webclient & Interface is intended to provide the facilities needed to get log data. Currently there is no infrastructure in place to get logs from the ventilator.

### Get DeviceInfo Http Webclient & Interface

The DeviceInfo Http Webclient & Interface is intended to provide the facilities needed to get device identifying information from the ventilator. Currently there is no infrastructure in place to get device information from the ventilator.

## Laptop Agent Abstraction Layer



### Session Manager & Manager

The Session Manager & Interface is intended to provide a means to establishing a session and logging into the Laptop Agent. Also, the session manager will associate a device to the system and initiate the process of downloading notifications for the device.

The session manager interface will provide an API to login, logoff and create a device. The calling process will have the option of synchronous/asynchronous behavior. The asynchronous behavior will require the calling process to provide a delegate (function pointer and object).

### Inbox

The inbox contains a table of notifications that are available for the device that is associated with the current session. The inbox gets notified of an incoming message by the EventManager.

### Outbox

The Outbox is a container where outgoing messages are placed. Messages are made available to the Message Sender. The Message Sender will serialize and transmit to the message to the laptop agent.

### EventManager

The Eventmanager is intended to track outgoing requests and incoming responses. When a request is made, the calling class registers a callback for the response and sets a timeout. When the response arrives or timeout occurs, the event manager calls the callback notifying the object of the event.

### Message Receiver & Sender

Both the message sender and receiver are dedicated to the task of communicating with the laptop agent.

## User Interface

### Solutions Center



The Solutions Center will provide the functionality required to login, view notifications, start the software update application, Log Viewer application and read release notes. A user will have the ability to double click a download notification to initiate a download.

### Log Viewer



The Log Viewer Application will provide the functionality required to retrieve and view ventilator logs. Export the logs to a file. Load and view an Exported logfile’s.

### Software Update Application



There is already a UI developed for the downloader. The UI was intended only to provide manufacturing a means of downloading software and is considered a temporary implementation.

Below is the current implementation.

