CUSV Fault Management Tool Requirements

Designed by Grey Files and Sahit Kadthala

0 – Overview

The CUSV Fault Management Tool is intended to allow the user to view the set of faults, failsafes, interrupts, and system notes associated with a specific software release. The embedded table will be populated with the items that are stored for the given software version. Additional functionality to modify the set of items is also provided if the given software release is not marked as read only. This tool also serves to allow for better access to stored fault information with features like searching and filtering.

1 – View Items for a Software Release

The user shall be able to select a software release and a category of items (faults, failsafes, interrupts, or system notes) from all that are stored in the system by selecting their desired release, and category from the dropdowns. This will display the items and all corresponding data to the user for that software release.

## 1.1 - Filter Items for Keywords

The user shall be able to add up to five filters to filter out a subset of items from the current display. Each filter will be a simple text string that will be filtered for in the specified field. A filter might look something like “’Enumeration Name’ contains ‘transmission’”. In that case, the only items that would be displayed on the table would be items that contain the string “transmission” in the “Enumeration Name” column (case in-sensitive).

## 1.2 – Search Items for Keywords

The user shall be able to enter a word or phrase that will be used in a search of all columns to filter items from the current display. This global search will work like a filter but would apply to all columns.

## 1.3 – User Customizable Views

The user shall be able to select the set of columns that will be displayed on the table. They should be able to save the current set of columns (referred to here as a view) locally and select that view in the future. Views for the different categories of items (faults, failsafes, interlocks, and system notes) only apply to the category of item that they were created for. This means, for example, that when viewing faults, the user would see a different set of saved views than when they are viewing interlocks.

2 - Edit Items in Unreleased Software Release

When viewing a software release marked “unreleased,” the user shall be able to select an existing item in the table and edit information within that item. The user shall also be able to create new items and fill in their fields or delete items completely. Pressing save when editing or adding a fault will create a pop-up to confirm that the changes should be sent to the database.

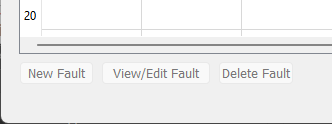
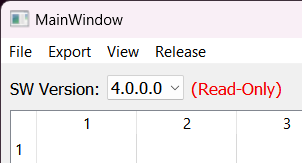
## 2.1 - Confirm Edits to Fault Information

When selecting the “save” button to update the database with the edited or new fault information, the user shall be presented with a list of all the changes that will be sent to the database and asked to confirm the changes.

3 – Mark Software Release as "Released” (Read-Only)

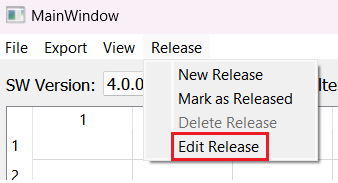
When viewing a software release marked “unreleased,” the user shall be able to mark the release as “read only”. There will be a popup message that will ask for confirmation and selecting “Yes” will lock all items and prevent any further changes to any items. Marking a release as “read only” would occur when the user selects this option from the release menu in the toolbar.

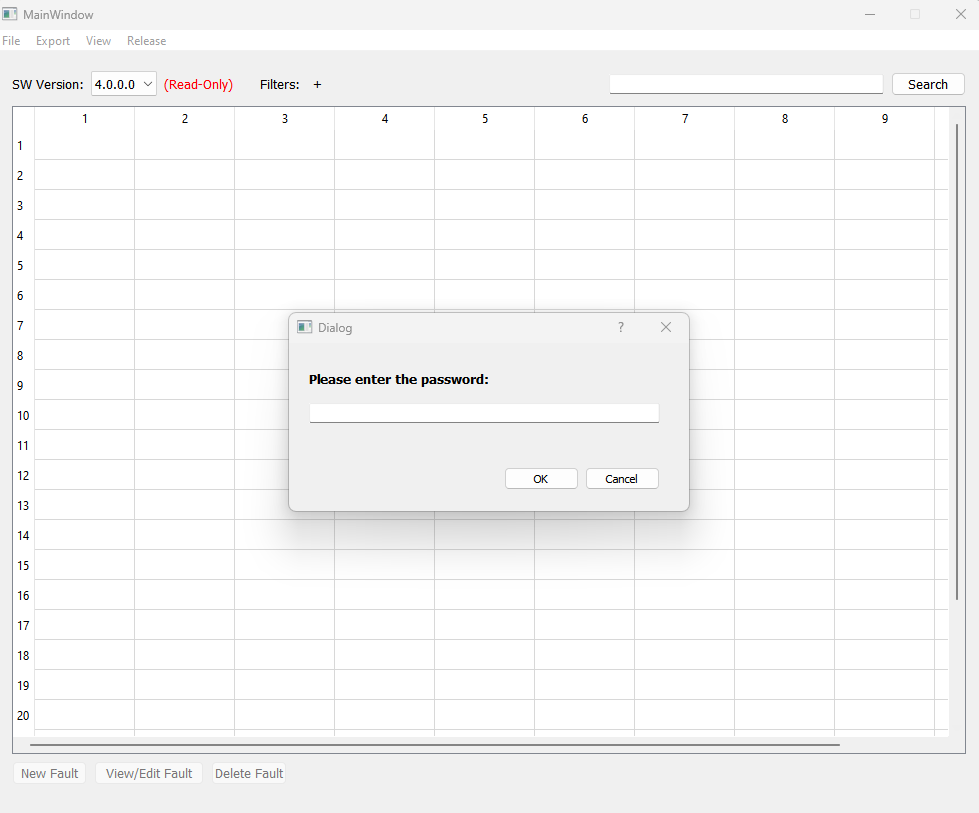
The buttons to add new items as well as edit existing items will be disabled if the user selects to view an already released software release, enforcing the idea of preventing additional changes.



## 3.1 - Unlock a Release Marked “read only”

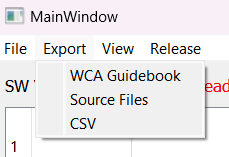
The user shall be able to select the option to make an edit to the items in a release already marked “Read-Only.” When the “Edit Release” option is selected, the user will be prompted for a password. If the password authentication is successful, then the buttons to add, edit, and delete faults will be enabled, and the software release will return to an unreleased state. To return the release to “read only”, the user would then select to “Mark as Released”.





4 – Export Item List for a Software Release

When viewing the items for a software release, the user shall be able to export the list of items in one of several formats.



## 4.1 - Export Item List in CSV

The user will be able to select the “CSV” menu option which will export a CSV file containing all data of the selected category from the selected software release.

## 4.2 - Export Fault List in Source Files

The user will be able to select the “Source Files” option and a file which will export that file correctly formatted to contain the proper fault information in the correct source code format:

1. VLC – FaultConstants.h
2. DSD – FaultStringLib.h/cpp
3. Fault reset configuration for VLC Central/HandlerFactory.xml

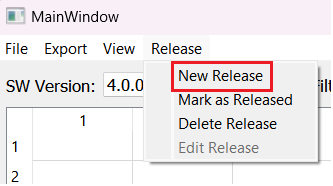
These source files are based on the system as designed right now; the final version may export different source files if the codebase changes to form a central location for all fault information

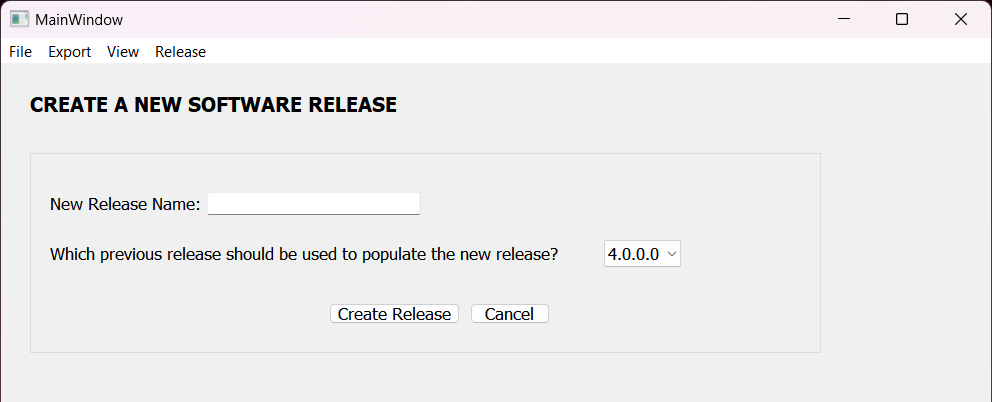
## 4.3 - Export Fault List in WCA Guidebook

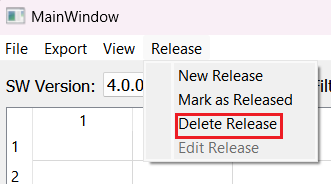
The user will be able to select the “WCA Guidebook” option and a payload which will export the correctly formatted WCA Guidebook xml file that includes all faults that are in the software release currently being viewed.

5 – Create/Delete Unreleased Software Release

The user shall be able to create a new software release in the system. The user will provide the name (number) for the release, and which existing release it shall inherit its initial items from. Similarly, the user shall be able to delete any unreleased software release and the option will be disabled when viewing an already released version.







Yet to Add

1. Enter user information for database connection
2. View all/no columns
3. View rows compact/expanded

Constraints

* Application should be written in C++ using Qt
* Database/storage medium should support offline access/export