BestXYZ Processor

Requirements, Design, Implementation, Testing (RDIT)

|  |  |
| --- | --- |
| User Story | Text |
| OUTPUTDATA | The user was able to create an ASCII tabular (spreadsheet) output file containing the following columns:  Time: The GPS Time-Of-Week from the input data, in seconds  UTC Time: The UTC Time-Of-Day, converted from the input data, in seconds  NS: The number of GPS Satellites tracked  E: The E-Axis ECEF Position from the input data, in meters  F: The F-Axis ECEF Position from the input data, in meters  G: The G-Axis ECEF Position from the input data, in meters  DeltaE: The Delta of the E column and the E-Axis of the Antenna Position, in meters  DeltaF: The Delta of the F column and the F-Axis of the Antenna Position, in meters  DeltaG: The Delta of the G column and the G-Axis of the Antenna Position, in meters  RSS: The Root-Sum-Square of the DeltaE, DeltaF, and DeltaG columns |

1. Requirements
   1. The user was able to have the selected input file be processed for Novatel messages
   2. The user was able to create an ASCII tabular output file from the processed Novatel messages.
2. Design
   1. Use a framework to create a main GUI dialog
   2. Add a second thread to perform processing while the GUI runs on the main thread
   3. Add a process button to the main GUI dialog
      1. Clicking the button will perform a series of checks on the selected files
         1. If the checks pass, then the interactive GUI elements (buttons and text entry fields) will be disabled during processing
         2. If the permission checks fail, then a message box will prompt the user with an error message
   4. Create a buffer to read in and process chunks of data from the input file
   5. Write the results into the specified output file
3. Implementation
   1. Create a QT Window
   2. Add a QButton to the GUI
      1. The QButton’s text will be set to “Process”
      2. Using QSpacers place the button underneath both the file selection entry fields and the antenna position selection entry fields
      3. Clicking the button will check that a valid input file is selected
         1. If one is selected, then the file will be checked for READ permissions
         2. If no file is selected, then a QMessageBox will prompt the user with a “read permissions error” dialog
      4. If no output or log files are selected, then they will be generated based on the input file before continuing
   3. Data structures were created to handle Novatel Headers, BestXYZ, Time, and Range messages
   4. Add a QThread to handle message processing without locking the GUI
      1. A ring buffer was used to process data in chunks
         1. A read and write pointer keeps the track of the indexes that contain unprocessed data
         2. The size of the data allocated between the read and write pointers is used to determine when more data should be read from the file
      2. Each BestXYZ message is written to the output file immediately after being processed
   5. When all data has been processed the GUI QButtons and QTextEdits are re-enabled and can be used for another round of processing
4. Testing  
   Test Setup: A computer with the BestXYZ Processor application loaded and containing at least one reference receiver dataset.
   1. Open the BestXYZProcessor application
   2. Type an incorrect file path into input file. For example: “<>><(32kl”
   3. Click the process QButton and verify that an error QMessageBox prompts the user
   4. Repeat steps 4.2 and 4.3 for output and log file QTextEdit entry fields
   5. Select a valid reference receiver dataset input file
   6. Input 0, 0, 0 into the E, F, and G QTextEdit entry fields
   7. Click the process button and verify that all buttons and text entry fields are disabled (greyed out)
   8. Verify that the GUI is not locked up by hovering the mouse over to the upper right hand of the application and checking that the OS specific buttons are still interactive
   9. When the processing is finished, check that the GUI is interactive once more
   10. The data was validated by analyzing the delta E, F, G coordinates and the delta RSS values for consistency
   11. Repeat steps 4.5 through 4.10 but this time load the E, F, and G coordinates in meters from a valid file