File Format Transfer (.INI to .CSV)

Software Requirement Specifications (SRS) Document (IEEE Std-830)

Contents

[Introduction 4](#_Toc44511810)

[Purpose 4](#_Toc44511811)

[Scope 4](#_Toc44511812)

[Definitions, Acronyms, and Abbreviations 4](#_Toc44511813)

[Overview 4](#_Toc44511814)

[General Description 5](#_Toc44511815)

[Product Perspective 5](#_Toc44511816)

[Product Functions 5](#_Toc44511817)

[Generate the user interface 5](#_Toc44511818)

[Buttons clickable from the user interface 5](#_Toc44511819)

[“Upload ftmMwMc.ini/ftmMwMd.ini file” button 5](#_Toc44511820)

[Read the input file 5](#_Toc44511821)

[“Transfer Now” button 5](#_Toc44511822)

[Output single CSV file 5](#_Toc44511823)

[“Combine all ftmMwMc/ftmMwMd output files” button 6](#_Toc44511824)

[“Clear all ftmMwMc/ftmMwMd single output CSV files” button 6](#_Toc44511825)

[Exit the program 6](#_Toc44511826)

[User Characteristics 6](#_Toc44511827)

[General Constraints 6](#_Toc44511828)

[Assumptions & Dependencies 7](#_Toc44511829)

[Specific Requirement 7](#_Toc44511830)

[Functional Requirements 7](#_Toc44511831)

[External Interface Requirements 7](#_Toc44511832)

[Performance Requirements 7](#_Toc44511833)

[Speed Requirements 7](#_Toc44511834)

[Safety Critical Requirements 7](#_Toc44511835)

[Precision Requirements 8](#_Toc44511836)

[Reliability and Availability Requirements 8](#_Toc44511837)

[Capacity Requirements 8](#_Toc44511838)

[Software Quality Attributes 8](#_Toc44511839)

[Security 8](#_Toc44511840)

[Maintainability 8](#_Toc44511841)

[Legal Requirements 8](#_Toc44511842)

[Look and Feel Requirements 8](#_Toc44511843)

[Usability Requirements 8](#_Toc44511844)

[Operational Requirements 8](#_Toc44511845)

# Introduction

## Purpose

The purpose of this software requirement specification document is to provide a complete description of requirements to understand the implementation process of the tool, File Format Transfer (.INI to .CSV).

## Scope

File Format Transfer is used to transfer the specific ‘sign’ sections in the configuration files (.INI) to an output CSV file (.CSV). The input files are ‘ftmMwMc.ini’ and ‘ftmMwMd.ini’ specifically. The document entails the specification of the software only. The software is not responsible for any of the physical mechanisms and therefore any requirements relating to said mechanism are not included in this document.

The goal of this tool is to have a script read .INI file input by the user, then output single CSV file, finally combine single CSV files to one combined CSV file.

## Definitions, Acronyms, and Abbreviations

These definitions will be used throughout the requirements and the rest of the document.

**ftmMwMc.ini** – this refers to the input configuration file name ending in .INI, which must be identical in all situations for the software to read.

**ftmMwMd.ini** – this refers to the input configuration file name ending in .INI, which must be identical in all situations for the software to read.

**‘sign’ section** – this refers to section(s) required format transfer, the software reads the data in this/these section(s) and transfers the data as it is.

**CROSSINGLOC\_LON** – this refers to crossing location’s longitude, the column name of the input file which must exist identically for the software to read.

**csv file** – this refers to the file format required for output file, and output file must end with a .csv extension.

## Overview

This document is divided into three sections, including the above introduction section.

In section 2, the general description of the tool is given. The factors that affect the utility and its requirements are described. This is to provide a context for the requirements that are given later. Real-life constraints (especially for the issues detected during testing the software) are illustrated here.

In section 3, the specifications of the software’s requirements are defined. This includes the description of the functionality of the system, as well as the required behaviour. Software quality attributes are introduced.

# General Description



## Product Perspective

The tool is designed to automate format transfer in ‘sign’ sections for two different formatted configuration files (‘ftmMwMd.ini’ and ‘ftmMwMc.ini’). The following functions are listed in the manner of how the tool works.

## Product Functions

### Generate the user interface

The tool must be able to generate a user interface after clicking to open the software.

### Buttons clickable from the user interface

The tool must be able to allow user to click the buttons shown on the user interface.

### “Upload ftmMwMc.ini/ftmMwMd.ini file” button

After clicking on the button, the file dialog interface opens and asks for a filename to open.

### Read the input file

The tool must be able to read the selected input ‘ftmMwMc.ini’ or ‘ftmMwMd.ini’ file and the data in the corresponding sections.

### “Transfer Now” button

After clicking on the button, the script starts to read the ‘sign’ sections and construct them in a data frame finally format the data in CSV file.

### Output single CSV file

The tool must be able to generate the single CSV file with the corresponding naming format “OUTPUT-ftmMwMc as of *CURRENTTIME*.csv” or “OUTPUT-ftmMwMd as of *CURRENTTIME*.csv”.

### “Combine all ftmMwMc/ftmMwMd output files” button

After clicking on the button, the script combines all the single output files to one combined CSV output file with the corresponding naming format “Combined-OUTPUT-ftmMwMc as of *CURRENTTIME*.csv” or “Combined-OUTPUT-ftmMwMd as of *CURRENTTIME*.csv”.

### “Clear all ftmMwMc/ftmMwMd single output CSV files” button

After clicking on the button, the script clears all single output CSV files with the same naming pattern (‘OUTPUT -ftmMwMc as of \*.csv’ or ‘OUTPUT-ftmMwMd as of\*.csv’).

### Exit the program

After clicking on the “close” button on the top right of the frame, the program terminates and exits.

## User Characteristics

Users of the tool (Linear Distance Calculator) are described below:

**General Tool Users** – they are the main users. They shall ensure the input file formats correctly, and the input file is expected to be a configuration file ending in .INI. It is expected that most of them are familiar with use of applications and can read and understand English.

**Software Technician/Maintenance** – They will be responsible to read and understand the content of this document. It is expected that the software technician will receive training on how to operate the tool and update the tool if requested. When updating is required, this specification requirements document shall also be up to date.

## General Constraints

**Input file format** – The tool must read a configuration file to be parsed later. The input file must have the section names “[sign]” in ‘ftmMwMc.ini’ and “[signX]” in ‘ftmMwMd.ini’. ‘ftmMwMc.ini’ must have 24 options under “[sign]” section.

**Zero sign** – If the sign count shows as ‘0’ in the input file, all columns return ‘NaN’ value (blank in the cell) except FTM number.

**FTM number** – FTM number is extracted from the path of the input file if exists. If FTM number is unable to be identified, the column “FTM” returns ‘N/A’.

Figure 2. Multiple selections of highways.

## Assumptions & Dependencies

The requirements for the utility described in this SRS are based on the following assumptions:

The existing operating system is compatible with the tool.

The environment settings are well set and some kind of interface is ready communicate with the tool.

An individual directory/path is used to store all the information.

The input file is stored in the same directory as the tool.

The file names have no conflict.

# Specific Requirement

## Functional Requirements

1. Except for the interactions between user and interface, the tool must require minimum or no user interaction with the core code.
2. The tool must be able to read the user’s input and therefore read the configuration file.
3. The tool must be able to output error message when action is unable to perform.
4. The tool must output single CSV file per one input file with the data in “[sign]” sections.
5. The tool must allow “Upload”, “Transfer”, “Combine” and “Clear” buttons to have the corresponding performance.
6. The tool allows combining single output CSV files to one combined CSV file.
7. The tool allows clearing all single output CSV files at once.
8. If the user closes the user interface, the application terminates.

## External Interface Requirements

1. The application shall be able to communicate with user.

## Performance Requirements

### Speed Requirements

After the user uploads the input configuration file, the application shall generate the single CSV file in the backend without any error in a timely manner. If the file is unbale to load, a popup window shows the error message, “Unable to load the source file!”.

### Safety Critical Requirements

The tool must comply with all applicable data processing tool safety laws.

### Precision Requirements

The data shall be identical as the input configuration file. Only the format changes.

### Reliability and Availability Requirements

Not applicable

### Capacity Requirements

Not applicable

## Software Quality Attributes

### Security

All user data including the input file shall be kept locally, as well as the output file.

### Maintainability

The application is to be easily modifiable; the code is heavily commented.

### Legal Requirements

1. The application shall comply with all relevant information/data privacy acts.
2. The application shall comply with all relevant transportation data laws.

### Look and Feel Requirements

1. The application shall be user-friendly.

### Usability Requirements

1. The tool must be simple for a profession in able condition to understand and use all its features.
2. The system must meet all software accessibility standards enforced by the Government of Canada.

### Operational Requirements

1. The tool should allow a variety of PC operating systems.
2. The tool needs a Python 3 environment.