Read Me: Ini File Editor

Contents

[Purpose and Functionality 2](#_Toc494964022)

[Configuration Format 2](#_Toc494964023)

[Required Keys 2](#_Toc494964024)

[Optional Keys 2](#_Toc494964025)

[Table of Required and Optional Keys 2](#_Toc494964026)

[Basic Configuration File Example 3](#_Toc494964027)

[Full Configuration File Example 3](#_Toc494964028)

[Example Result 3](#_Toc494964029)

[Example Setup.ini File – Pre Execution 3](#_Toc494964030)

[Comparison of Setup.ini - Post Execution 4](#_Toc494964031)

[Installation Run Book 4](#_Toc494964032)

[Phases 5](#_Toc494964033)

[Phase I - Confirm Changes to be made 5](#_Toc494964034)

[Phase II – Scan file system 6](#_Toc494964035)

[Phase III – Backup files 6](#_Toc494964036)

[Phase IV – Edit files 6](#_Toc494964037)

[Phase V – Verify Changes 6](#_Toc494964038)

[Forensics 6](#_Toc494964039)

[Developer Contact and Support Info 7](#_Toc494964040)

# Purpose and Functionality

IniFileEditor is a program written in JAVA for maintaining .ini files per the Windows spec located [here](https://en.wikipedia.org/wiki/INI_file). The opportunity to make a generic .ini file editing tool was taken when then the need to migrate a customer to a new IP address for AWS surfaced. The program updates values in a section if present in the file or adds the new property and value if not already present while preserving all other lines in the files (such as comments). Before any changes are made it creates a backup of the original file.

# Configuration Format

Configuration is a prototypical JAVA xml properties file with 2 required keys and 2 optional keys specifying behavior.

## Required Keys

The two required keys are:

1. requiredFileNamePattern
2. requiredSectionName

## Optional Keys

The two optional keys are:

1. unattendedMode
2. restrictSearchToDirectories

## Table of Required and Optional Keys

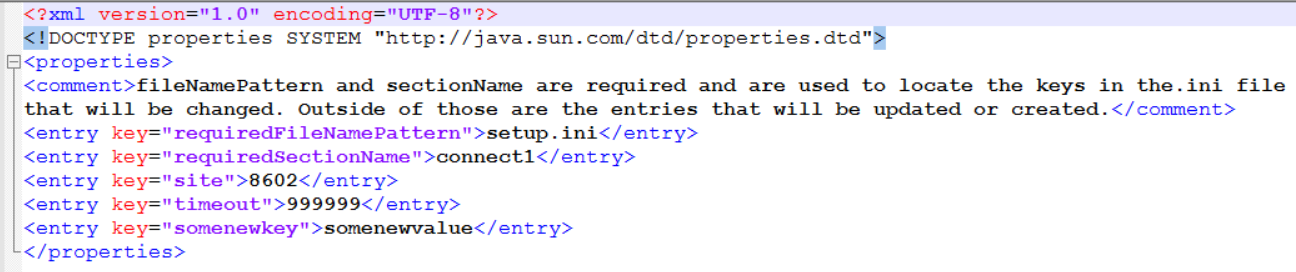
|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | **Purpose** | **Format** | **Required** |
| requiredFileNamePattern | Search pattern for the file. Examples: setup.ini, tomax.ini | Valid Windows format filename without directory | Yes |
| requiredSectionName | The section name without square brackets to search for. Examples: identity, connect1, connect2, general, CreditGateway | Section name in the ini file without square brackets | Yes |
| unattendedMode | Required to run script without user input | Boolean (true or false) | Required to run script without user input. Assumes value of false if not provided. |
| restrictSearchToDirectories | Restricts the search for the file pattern and section to a specific set of directories or single directories. | Semicolon delimited list of directories without trailing backslash. Examples: C:\Tomax;C:\MyFiles | Required to avoid editing all matching files found on the machine |

Beyond that, all other keys are treated as entries in the editing/adding operations with key being a property in the .ini file and value being the value in the .ini file respectively.

## Basic Configuration File Example

Here is an example file which will look in the file system for all files matching the pattern “setup.ini’ and a section called “connect1” (literally “[connect1]” in the file). **Note**: in this ‘minimal configuration’ example that:

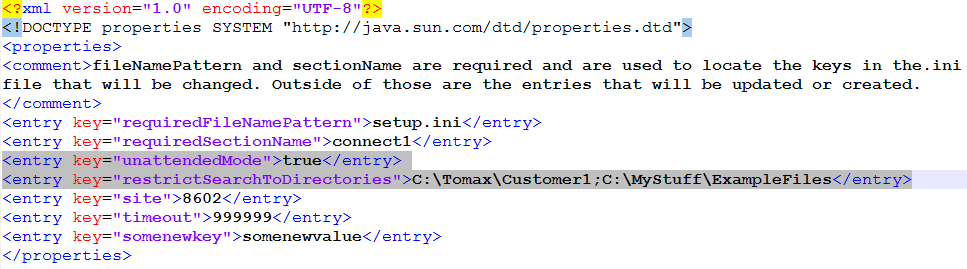
1. unattendedMode is not supplied thus the user is prompted
2. restrictSearchToDirectories is not supplied so the entire machine will be scanned



If qualifying files are found, the site will be created or updated to 8602, along with timeout being set to 999999. In the example, “somenewkey” is a new key so the property-value pair “somenewkey=somenewvalue” will be added to the file in the connect1 section of setup.ini files which have a connect1 section.

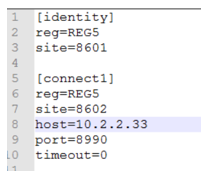
## Full Configuration File Example

Below is a full configuration file example using unattendedMode and restrictSearchToDirectories in grey.



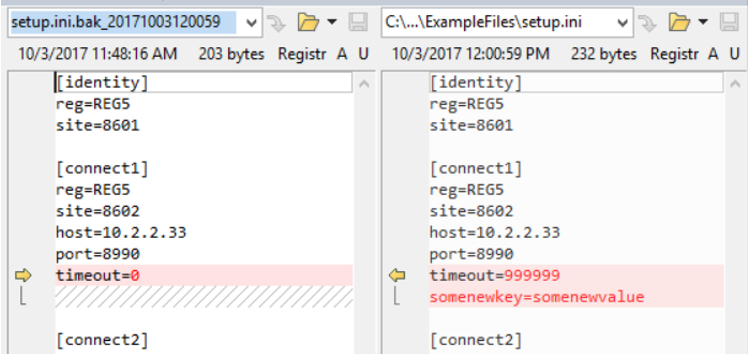
Example Result

## Example Setup.ini File – Pre Execution

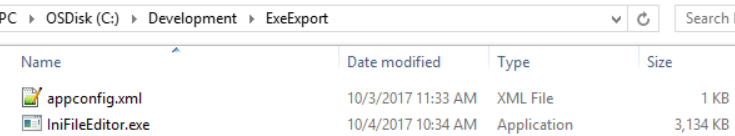
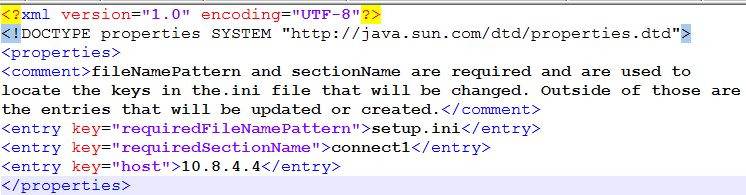
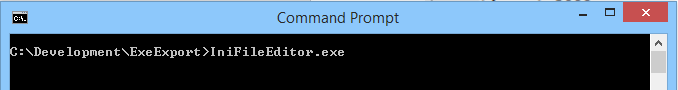
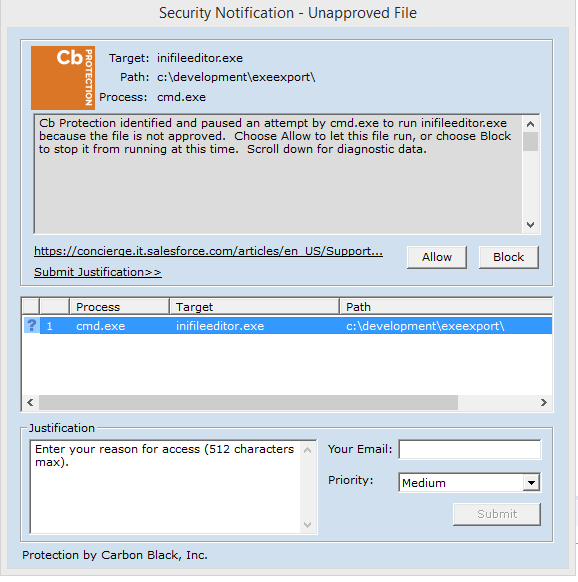
Here is an original file example that we will edit with the configuration mentioned above: 

## Comparison of Setup.ini - Post Execution

Below is a comparison of the before (left) and after (right) post run:



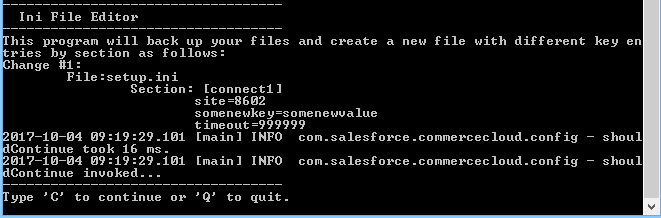
# Installation Run Book

1. Drop the 2 files provided into the same directory. **Note**: appconfig.xml is mandatory.
2. Check that the xml file has the correct edits to the ini file you wish to make. Here is an example showing a change for a property called “host” in the connect1 section of the setup.ini file: 
3. Run the program from a cmd shell using the directory with the IniFileEditor.exe and appconfig.xml files in it like this. **Note**: It may take a moment for the program to start or for security software to warn about the file.
4. If security software or operating system prompts are presented, allow the file to run. Here is an example of what may display as a warning:
5. Allow the program execute. This may take a few minutes depending on the machine capacity. Phases of the execution are outlined in the section “Phases.”
6. There should now be a logs directory. Inspect the log files for the program results.

# Phases

## Phase I - Confirm Changes to be made

By providing some configuration values in an XML file and running the script, the user will be told about the changes and then asked if they want to continue:

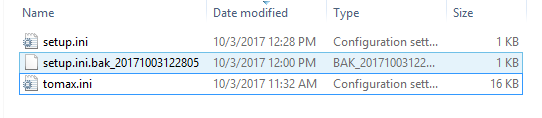


## Phase II – Scan file system

In this phase all drives of the system are scanned looking for a matching file pattern and section; in our example case: “setup.ini” and “connect1.” Files which match these conditions are pulled in for editing. **NOTE**: It could take a few minutes for this process to complete.

## Phase III – Backup files

The original files are backed up with a timestamp in mask format “YYYYMMddHHmmss” or more plainly: 4-digit year 2-digit month 2-digit day 2-digit hour 2-digit minute 2-digit second. Here is a folder with a backup of the setup.ini file:



## Phase IV – Edit files

In this phase the edits and commits are made to the .ini files based on the configuration entries.

## Phase V – Verify Changes

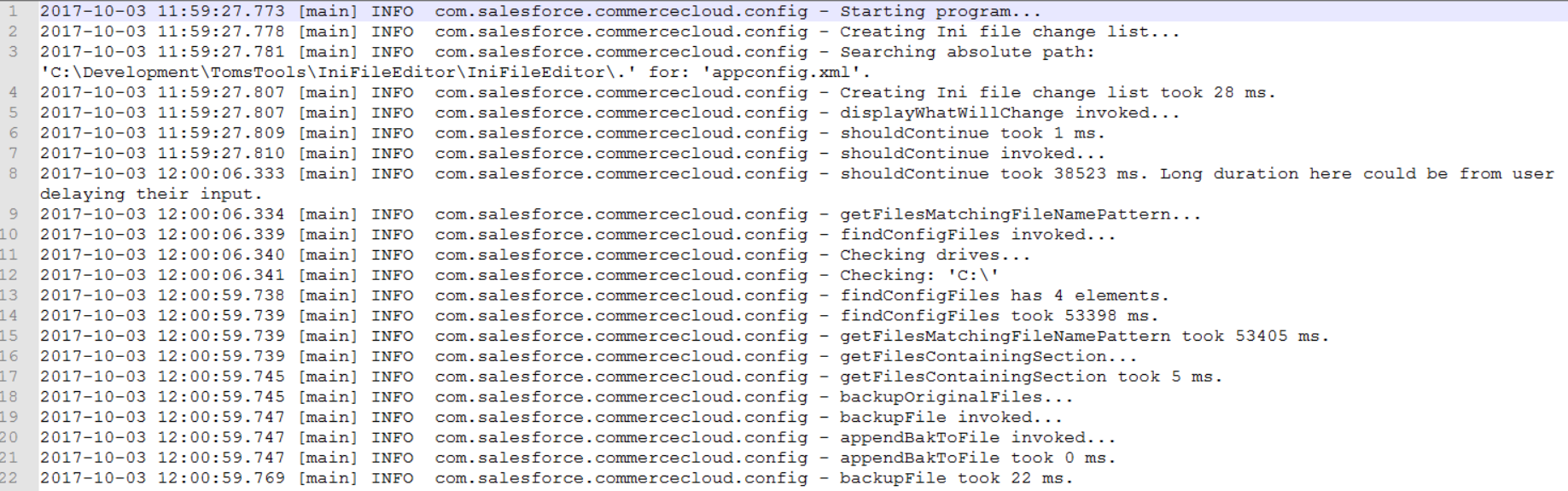
In this phase the files are re-read from disk to make sure the changes are present.

# Forensics

Logging is done with the JAVA library log4j2. *A directory called logs relative to the executable will be created along with the empty files if not provided already*. Logging of nearly everything is done to the console window as the program executes. There are also 2 log files provided:

|  |  |
| --- | --- |
| **IniFileEditor-all.log** | Contains the key milestones that the program runs through |
| **IniFileEditor-error.log** | Contains any errors handled during execution |

Here is an example of the IniFileEditor-all.log file:



Absence of data in the IniFileEditor-error.log (0 kb file size) indicates that the configuration was done appropriately and the run was successful. Contact the developer with any other errors. **Note**: If you encounter SAX parser errors, check the formatting of your appconfig.xml file.

# Developer Contact and Support Info

​Thomas (Tom) Zarr

Technical Architect | Salesforce Commerce Cloud

224 S. 200 W., Salt Lake City, UT 84101

Mobile: [+1 801-971-1217](tel:(801)%20971-1217)