Read Me: Ini and Host File Editor

Contents

[Purpose and Functionality 3](#_Toc501553120)

[INI File Maintenance 3](#_Toc501553121)

[Host File Maintenance 3](#_Toc501553122)

[Configuration Format 3](#_Toc501553123)

[Required Key Patterns 3](#_Toc501553124)

[Optional Key Patterns 3](#_Toc501553125)

[Table of Required and Optional Keys Explained 4](#_Toc501553126)

[Basic Configuration File Example 5](#_Toc501553127)

[Full Configuration File Example 5](#_Toc501553128)

[Summary of INI Changes in Full Example File 6](#_Toc501553129)

[Summary of Hosts File Changes in Full Example File 7](#_Toc501553130)

[INI and Host File Example Results from Full Example File 7](#_Toc501553131)

[Setup.ini File 7](#_Toc501553132)

[Pre Execution 7](#_Toc501553133)

[Post Execution Comparison Report (Setup.ini) 8](#_Toc501553134)

[Tomax.ini File 8](#_Toc501553135)

[Pre Execution 8](#_Toc501553136)

[Post Execution Comparison Report (Tomax.ini) 9](#_Toc501553137)

[Hosts File 9](#_Toc501553138)

[Pre Execution 9](#_Toc501553139)

[Post Execution Comparison Report (Hosts) 10](#_Toc501553140)

[Ping Verification Post Execution 10](#_Toc501553141)

[Installation Run Book 11](#_Toc501553142)

[Phases 12](#_Toc501553143)

[Phase I - Confirm Changes to be made (does not apply in unattended mode) 12](#_Toc501553144)

[Phase II – Scan file system 13](#_Toc501553145)

[Phase III – Backup files 13](#_Toc501553146)

[Phase IV – Edit files 13](#_Toc501553147)

[Phase V – Verify Changes 13](#_Toc501553148)

[Forensics 13](#_Toc501553149)

[Documented Full Configuration File Example 15](#_Toc501553150)

[Developer Contact and Support Info 16](#_Toc501553151)

# 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). It can also add or update the entries in a hosts file. The opportunity to make a generic bulk editing tool was taken when then the need to migrate a customer to a new IP addresses for AWS cloud surfaced. The tool’s input is driven by a configuration file.

## INI File Maintenance

The program updates values in an INI file 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 including comments such as leading semicolons. Before any changes are made it creates a backup of the original file.

## Host File Maintenance

With administrator rights it is possible to edit host file entries by inserting new entries, updating the IP addresses of existing entries or removing existing entries.

**Note**: The host file capabilities have been written for execution on a windows machine with regard to the host file meaning the program looks for the *Windows/system32/drivers/etc* directory after locating the operating system drive. Although written in JAVA It has not yet been coded with other operating systems in mind.

# Configuration Format

Configuration is a prototypical JAVA xml properties file with 2 required key patterns and 4 optional keys specifying behavior.

## Required Key Patterns

The required key patterns are:

1. ini + groupName + FileNamePattern
2. ini + groupName + SectionName

…where the plus symbol signifies concatenation, i.e. ‘ini1FileNamePattern’ or ‘ini1SectionName.’

## Optional Key Patterns

The optional key patterns are:

1. unattendedMode
2. ini + groupName + RestrictSearchToDirectories
3. host + groupName + add
4. host + groupName + remove

…where the plus symbol signifies concatenation, i.e. ‘ini2RestrictSearchToDirectories’ or ‘ini3SectionName.’ The entries for hostadd and hostremove require a pipe-delimited entry for the value and use key names like host1add, host2add, host3add, host1remove, etc.

## Table of Required and Optional Keys Explained

|  |  |  |  |
| --- | --- | --- | --- |
| **Key Pattern** | **Purpose** | **Value Format** | **Required** |
| unattendedMode | Required to run script without user input such as when launched by another process | Boolean (true or false) | Required to run script without user input. Assumes value of false if not provided. |
| FileNamePattern e.g. ini1FileNamePattern or ini3FileNamePattern | The INI file to search for and make edits in. To find the setup.ini file in the first group of changes it should be set to <entry key="ini1FileNamePattern">setup.ini</entry> | String in the format of an INI file such as ‘chantest.ini’ | Yes |
| SectionName e.g. ini1SectionName or  Ini2SectionName | The section name without square brackets to search for. Examples: identity, connect1, connect2, general, CreditGateway | Section name in the INI file without square brackets such as connect1 | Yes |
| RestrictSearchToDirectories e.g. ini1RestrictSearchToDirectories | 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. 2 Examples:  1. C:\Tomax;C:\MyFiles  2. D:\MyPlace\stuff;D:\YourPlace\foo\bar | Required to avoid editing all matching files found on the machine |
| hostadd e.g. host1add, host2add, host3add | Adds an entry to the operating system drive’s hosts file in the /Windows/system32/drivers/etc/ directory or if the host name is already present updates the IP address. | Pipe-delimited with IP address first | Required only if you want to modify the hosts file |
| hostremove e.g. host1remove, host3remove | Removes an entry from the operating system drive’s hosts file in the /Windows/system32/drivers/etc/ directory if and only if both the IP address and host match. | Pipe-delimited with IP address first | Required only if you want to remove an entry from the hosts file |

Beyond that, all other keys in a group of INI changes 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.

For example:

<entry key="ini1site">8601</entry>

<entry key="ini1timeout">12121212</entry>

<entry key="ini1somenewkey">somenewvalue4</entry>

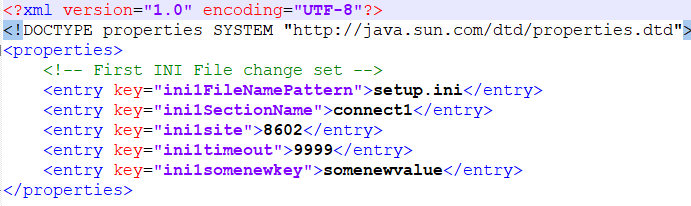
<entry key="ini1somenewkey666">somenewvalue666</entry>

The ini prefix and groupName (in this case “1”) are not actually used in the file. The above example would update or add entries for site, timeout, somenewkey and somenewkey666.

## Basic Configuration File Example

Here is a basic example file which will make only INI changes by looking 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:

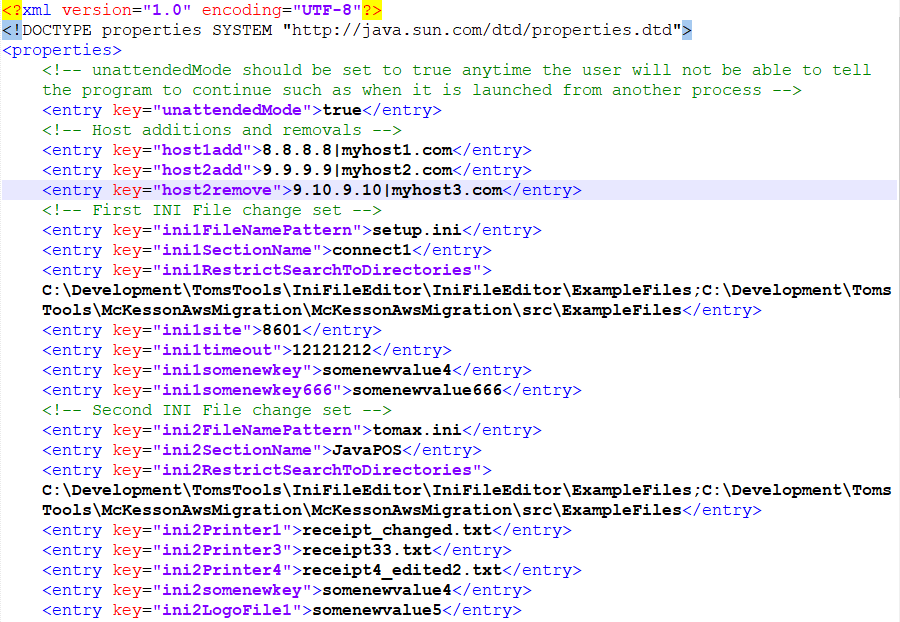
1. unattendedMode is not supplied thus the user is prompted to continue
2. There are no hostadd or hostremove keys present (changing only INI files)
3. RestrictSearchToDirectories is not supplied for the group (machine will be scanned)



So, what changes would be made ibn this minimal example? If qualifying files are found, the site will be created or updated to 8602, along with timeout being set to 9999. 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 were found **and** contain a [connect1] section.

## Full Configuration File Example

Below is a full configuration file example using unattendedMode and RestrictSearchToDirectories along with hostadd and hostremove tagging whose execution which will be referenced further below.



### Summary of INI Changes in Full Example File

Regarding the INI changes, if qualifying files are found they will be executed against each group. Remember that group name talked about earlier? It can be as simple as 1 or 2 or longer if needed.

The tags commented under “First INI File change set” will result in the following applied to any **setup.ini** file found within the search restriction which contains a **[connect1]** section:

* **site** will be created or updated to 8601
* **timeout** will be created or set to 12121212
* **somenewkey** is a new key so the property-value pair “somenewkey=somenewvalue4” will be added
* **somenewkey666** is a new key so the property-value pair “somenewkey666=somenewvalue666” will be added

The tags commented under “Second INI File change set” will result in the following applied to any **tomax.ini** file found within the search restriction which contains a **[JavaPOS]** section:

* **Printer1** will be created or updated to receipt\_changed.txt
* **Printer3** will be created or updated to receipt33.txt
* **Printer4** will be created or updated to receipt4\_edited2.txt
* **somenewkey** is a new key so the property-value pair “somenewkey=somenewvalue4” will be added
* **LogoFile1** is a new key (since technically there is a commented one present) so the property-value pair “LogoFile1=somenewvalue5” will be added

### Summary of Hosts File Changes in Full Example File

Regarding host file changes, if the program is run with administrator rights:

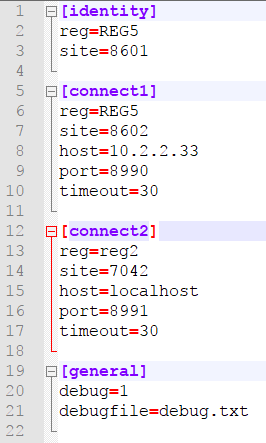
* A host entry will be added for **myhost1.com** with IP address 8.8.8.8
* A host entry will be added for **myhost2.com** with IP address 9.9.9.9
* Any host entry which matches exactly the hostname **myhost3.com and 9.10.9.10** will be removed from the host file

## INI and Host File Example Results from Full Example File

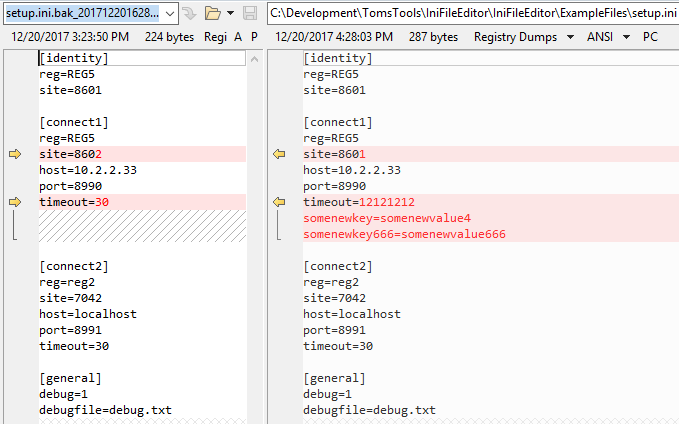
So does this thing actually work? An example showing pre/post execution which spells out the changes in comparison reports follows.

## Setup.ini File

### Pre Execution

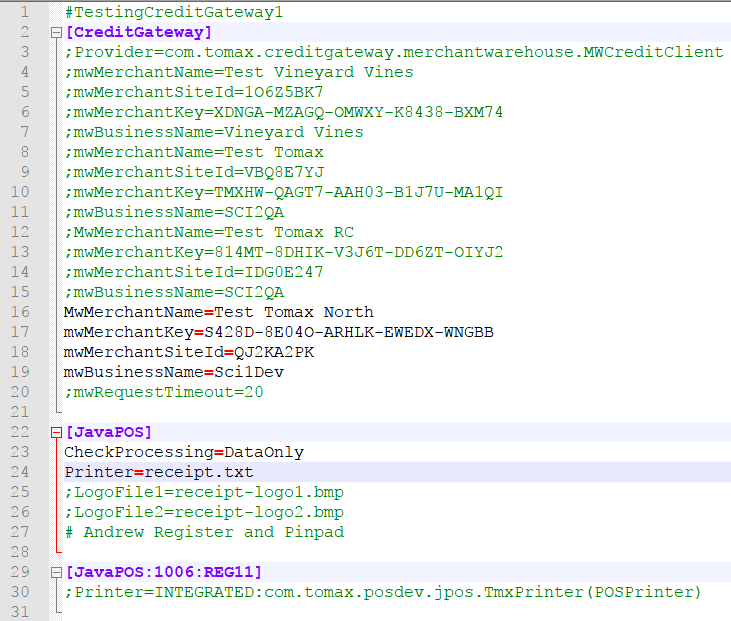


### Post Execution Comparison Report (Setup.ini)

Below is a comparison of the before (left) and after (right) post run which is also using the backup of the setup.ini file: 

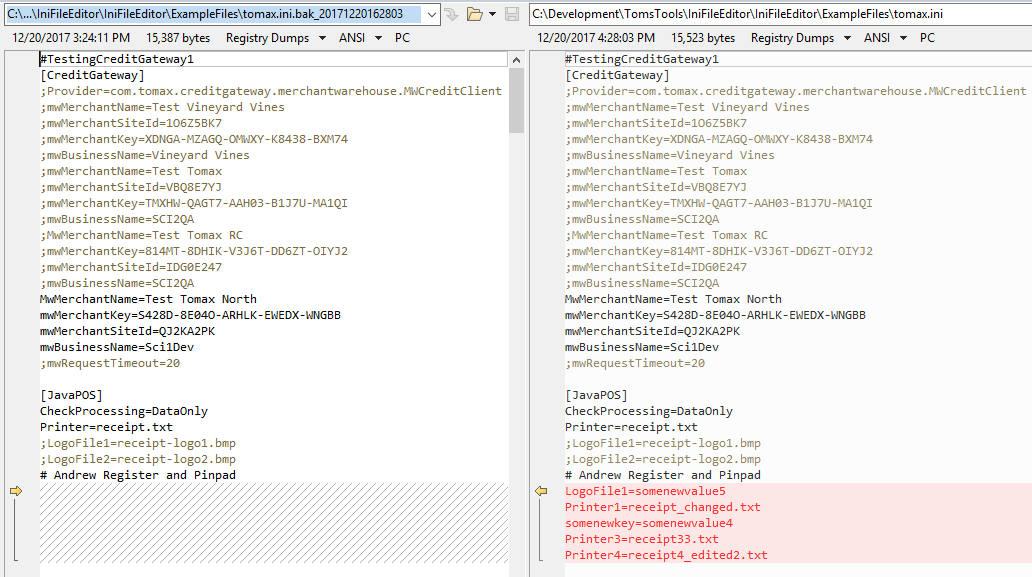
## Tomax.ini File

### Pre Execution



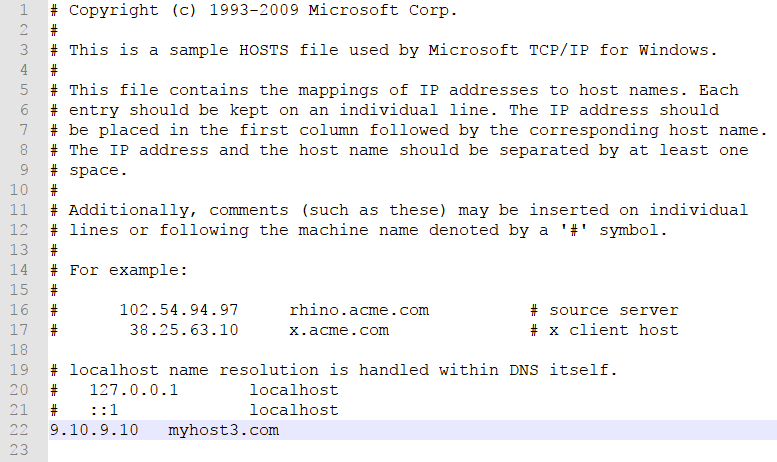
### Post Execution Comparison Report (Tomax.ini)

Below is a comparison of the before (left) and after (right) post run which is also using the backup of the **tomax.ini** file:



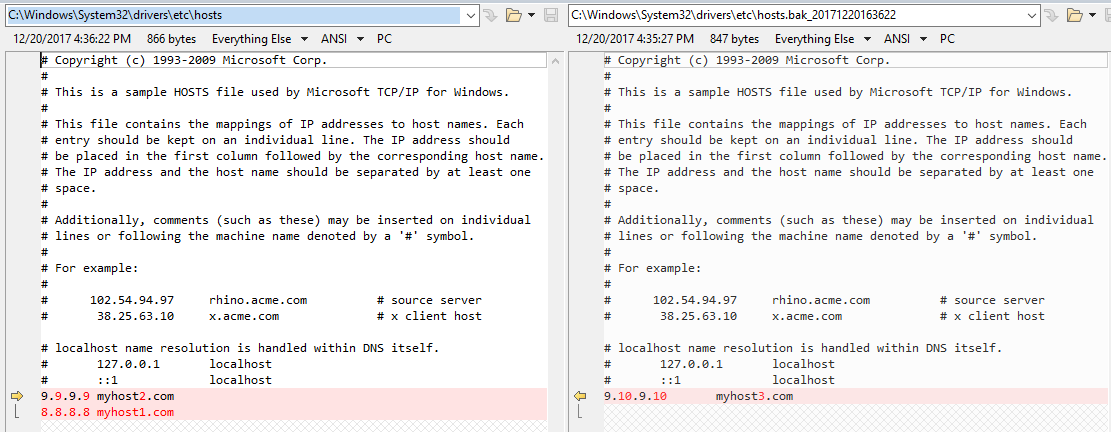
### Hosts File

### Pre Execution



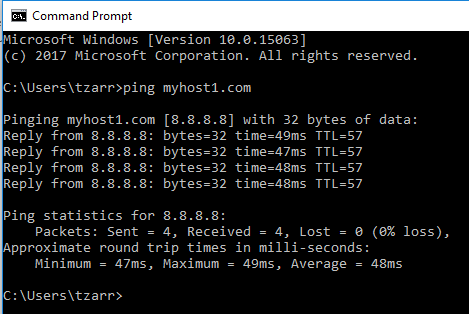
### Post Execution Comparison Report (Hosts)

Below is a comparison of the before (left) and after (right) post run which is also using the backup of the hosts file:



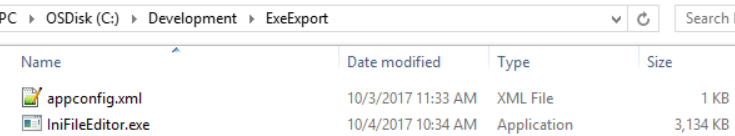
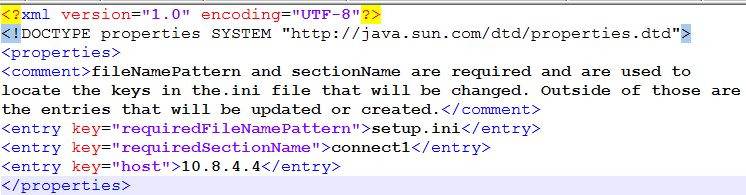
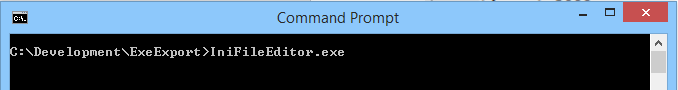
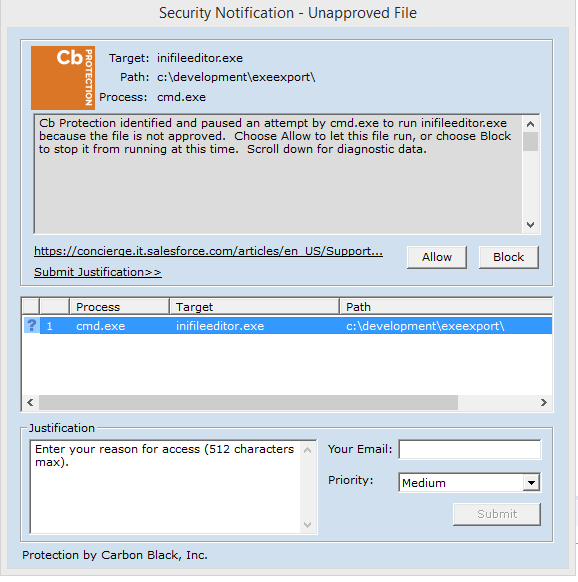
## Ping Verification Post Execution

You can verify that your host entry is working by executing the following in the cmd shell:



Note: For a valid ping response you must have a route to the host (VPN connection, etc.)

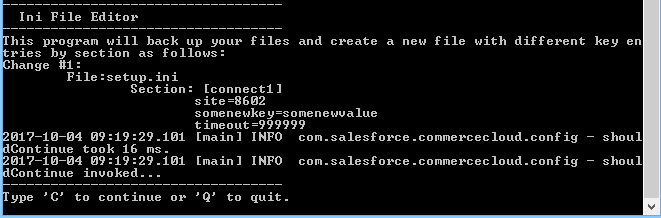
# Installation Run Book

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 want 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 (does not apply in unattended mode)

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:

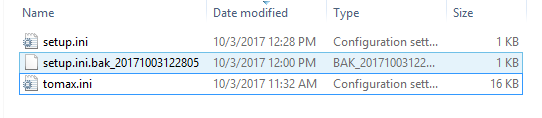


## Phase II – Scan file system

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

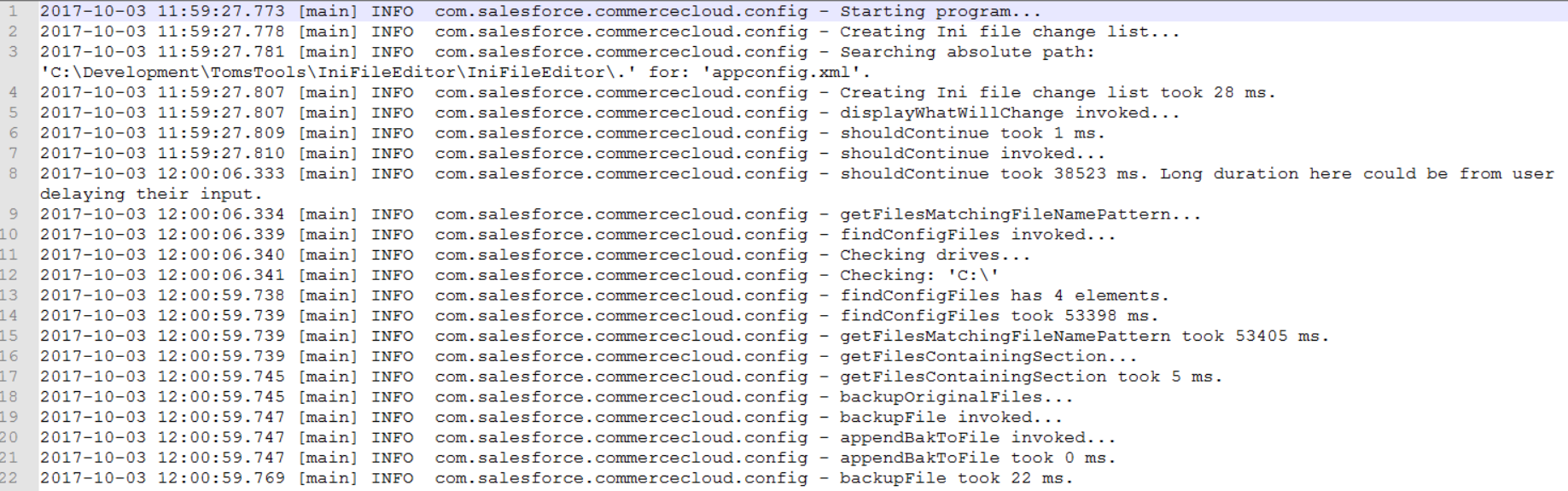
In this phase the files are re-read from disk to make sure the changes are present.

# 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:

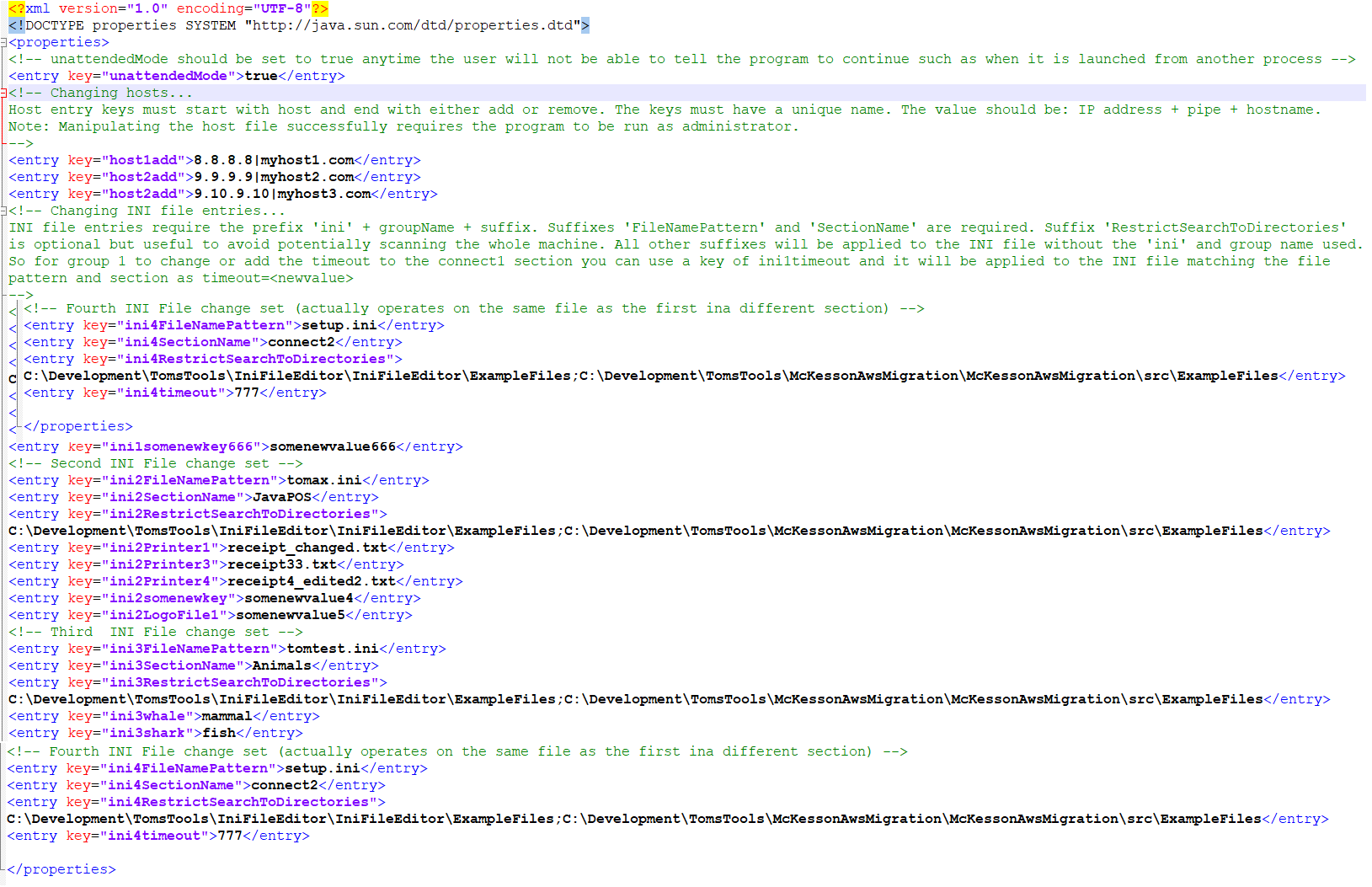
|  |  |
| --- | --- |
| **IniFileEditor-all.log** | Contains the key milestones that the program runs through |
| **IniFileEditor-error.log** | Contains any errors handled during execution |

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. Below is a fully documented example file.

# Documented Full Configuration File Example



# 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)