Software Deployment at1.7

30003389 – Kyer Potts

Table of Contents

[Table of Contents i](#_Toc25338002)

[Configuration Files 1](#_Toc25338003)

[Using Configuration Files 2](#_Toc25338004)

[Implementing Configuration Files 4](#_Toc25338005)

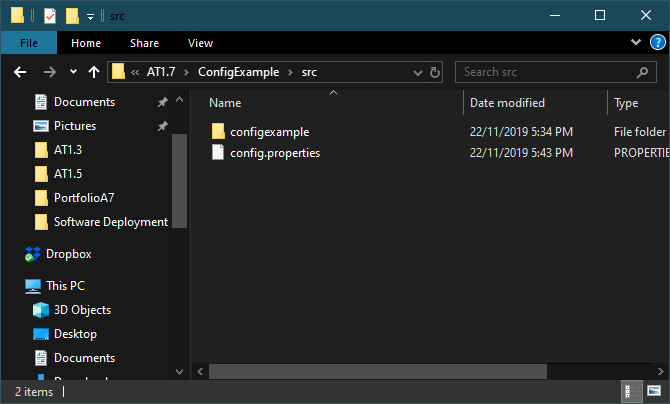
# Configuration Files

Configuration files are used to set certain parameters and initial settings within a program post deployment. The most common of these file types is an INI file, however, the INI file type is not an end all standard and many other types of configuration filer exist. Some of these files allow the user to edit specific settings within a program from a GUI interface within the program itself, or the file can be edited directly through the use of a text editor. The following file types are examples of configuration files:

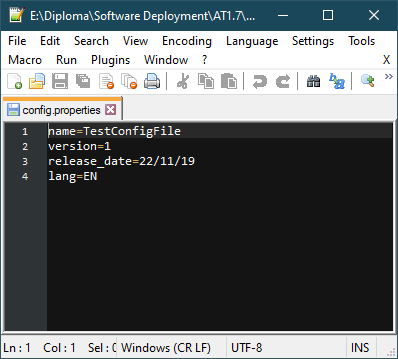
* INI File, a common configuration file format
* .properties, a file extension mainly used in Java
* JSON, with support for complex data types and data structures
* HOCON, a superset of .properties and JSON
* Run commands, which explains the historical origin of the “rc” suffix
* TOML, a formally-specified configuration file format
* YAML, with support for complex data types and structures

# Using Configuration Files

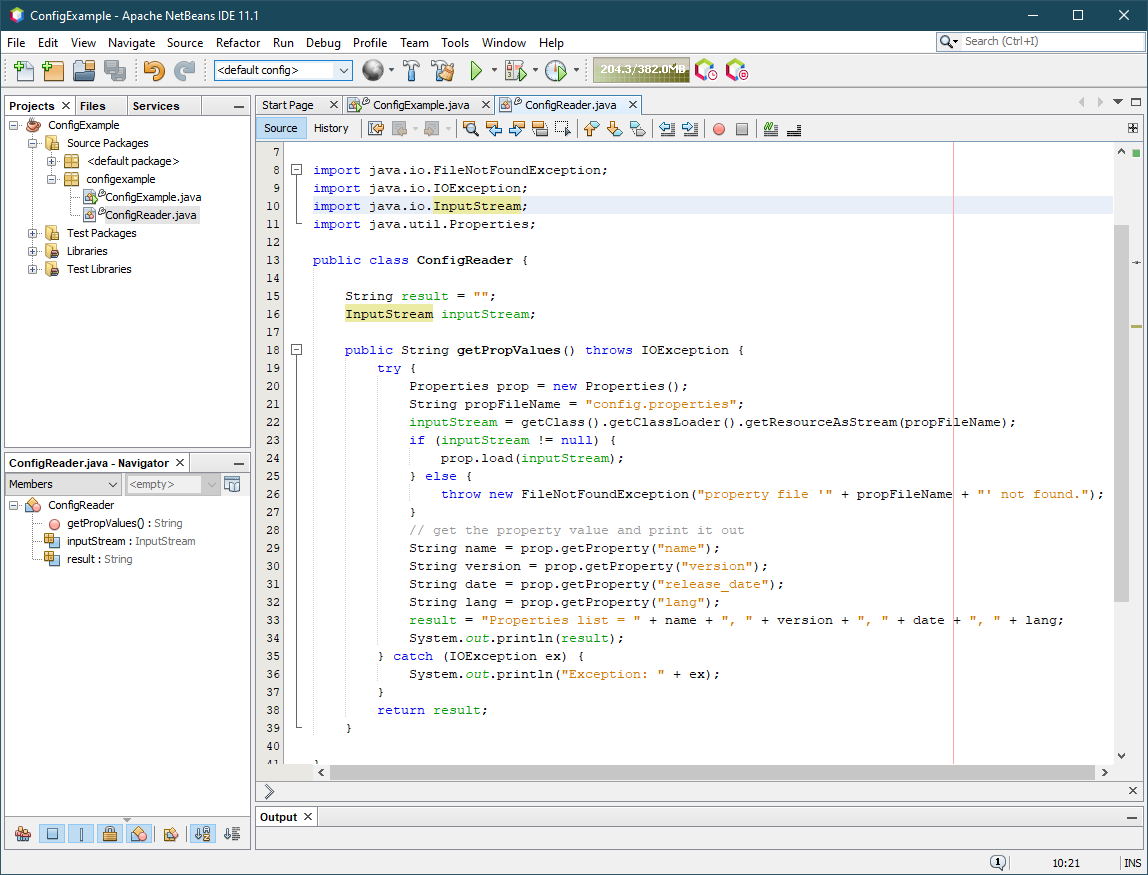
To use configuration files, first the platform of the program must be determined, and the underlying syntax of the language or languages used must be observed and adhered to. A common practice is to import the data of the configuration file with a filestream reader, and then separate the values using code, which then sets the variables in the program with the appropriate information.



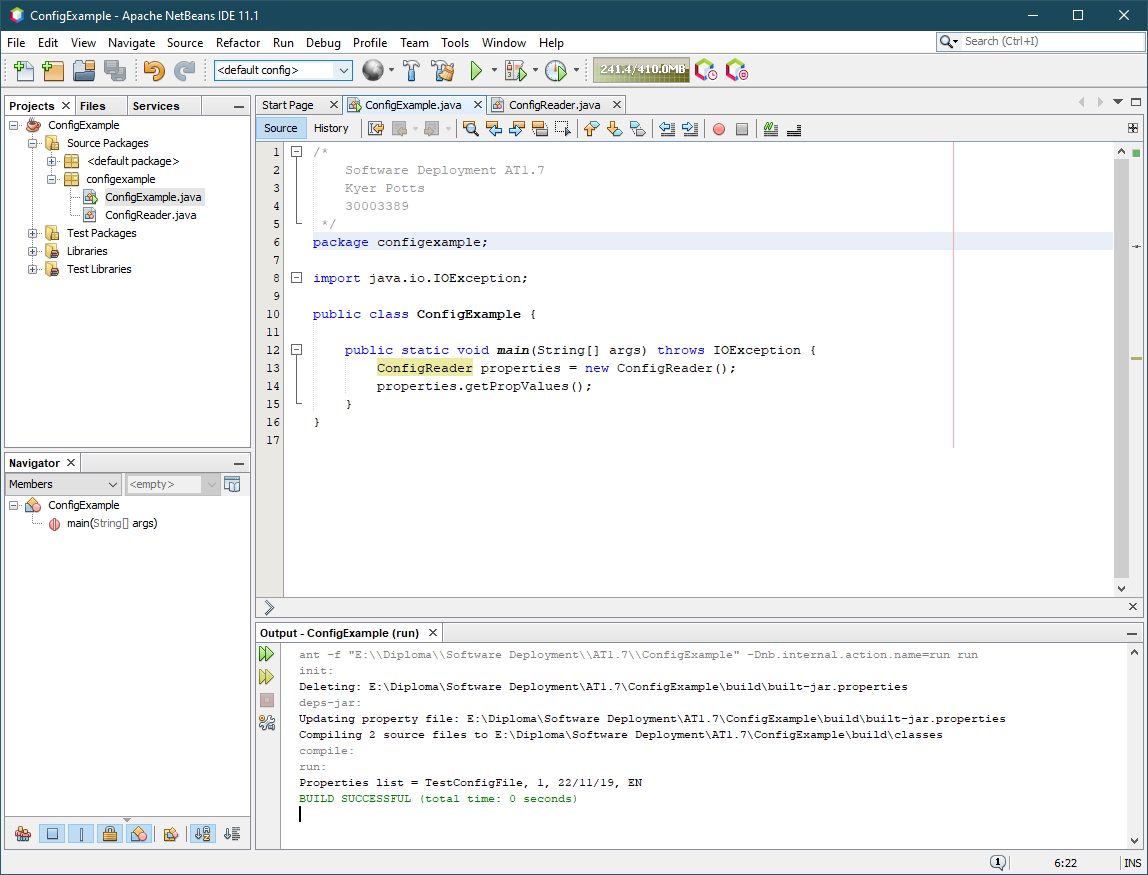
Here I have place the config file within the src files of the project. This will allow the program to easily find the file with the streamreader.



I created a new class with the following code snippet to enable to program to locate and read the config file, then store the information contained within the config file as a concatenated string.



The main class then instantiates the config reader class and runs the getPropValues() method to print out the information contained within the config file.



Seen above, the output of the config file is “TestConfigFile, 1, 22/11/19, EN”

# Implementing Configuration Files

Using the above method, configuration settings can be imported directly into variables during runtime, allowing users to dynamically update the information contained within variables and change the behaviour of the program. There are many different implementations of configuration files, from altering graphical settings within video games, to changing font designs within text editors. Depending on the program being built, the programmer will need to identify the most optimal path to include configuration settings within their application.