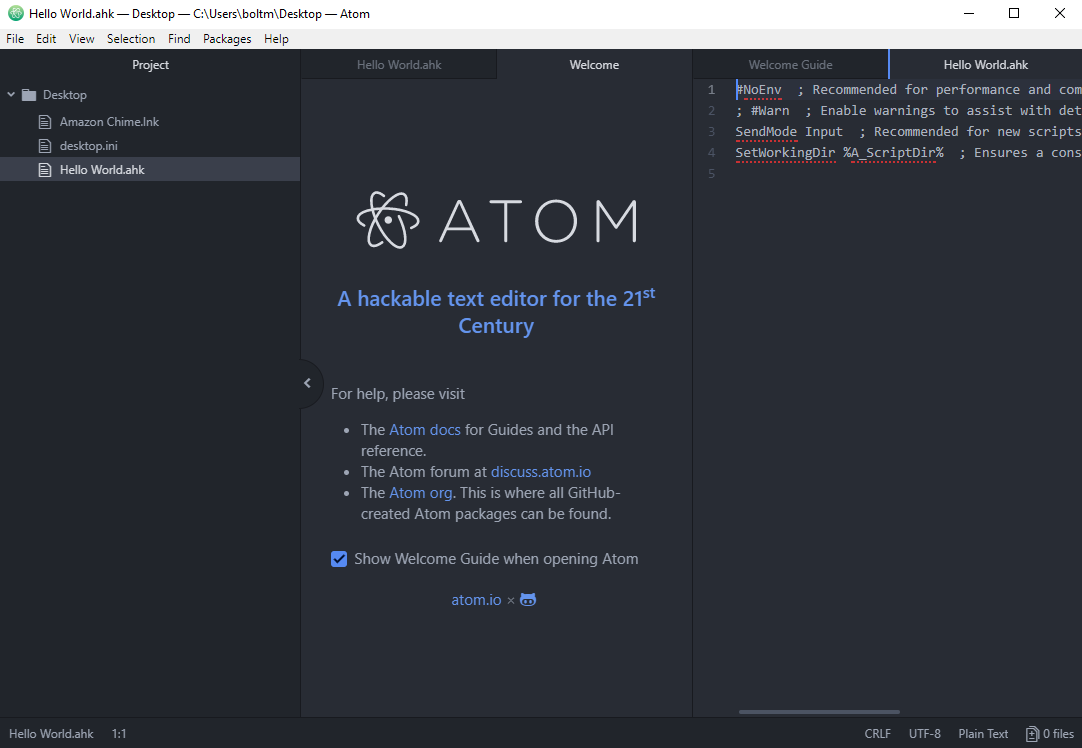
The purpose of this document is to provide insight on how the Alarm Acknowledger (AA) program functions and interacts with Lenel. This document will not explain how the Auto Hot Key (AHK) scripting language is written. Some links that would be beneficial to familiarize yourself with would be the [AHK Webpage](https://autohotkey.com/docs/AutoHotkey.htm) and [Turbo Future](https://turbofuture.com/computers/keyboard-shortcut-keys).

Before we can start learning how AA functions we need to prepare our environment. You can visit the [AHK webpage](https://autohotkey.com/) or the Amazon Software Center to install the application on your computer. Once the application is installed, you are technically able to create a script using Notepad. However, I would highly recommend downloading a text editor as it is easier to keep track of your code. I personally use [Atom](https://atom.io/) to create my scripts but you can any text editor of your choice. For the purpose of this tutorial I’m going to assume you’re using Atom.

Atom Enviroment:



You will notice a couple of tabs at the top that load with Atom, go ahead and close all of them. Go ahead and open AA within Atom and you be able to modify the script.

Now let’s dive into how Alarm Acknowledger is coded. The first interaction the end user has is generally with one of the five GUI’s (Acknowledge GUI, Responder Type / TT's GUI, Update Alarms GUI, Issue Reporting GUI, Help).

These sections of code are separated by multiple dashes with their name at the end of the dashes as demonstrated in the graphic below.



Two of the GUI’s are activated with hot keys.

* Acknowledge GUI = CTRL + 0
* Update GUI = CTRL + 1

While the other three are activated by buttons within the first two GUI’s. I’ve linked the training document to this document if you would like to further dive into how the GUI’s are used from an end user view.



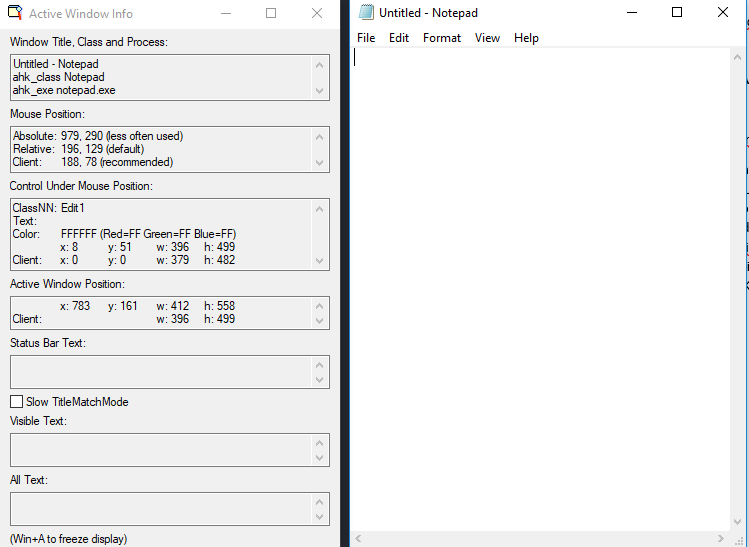
Some default functions I have running in the script are as follows:

* #SingleInstance force
  + This makes it so there can only be one version of AA running at a time. Without this the script will load multiple instances of itself and create conflict.
* SetNumLockState, AlwaysOn
  + As most of the keys are tied to the numpad I’ve removed the ability to turn the numpad off.
* SetKeyDelay, 50
  + This function allows AA to interact better with Lenel as the program will move slower.

Two major issues AA faces when trying to interact with Lenel are as follows:

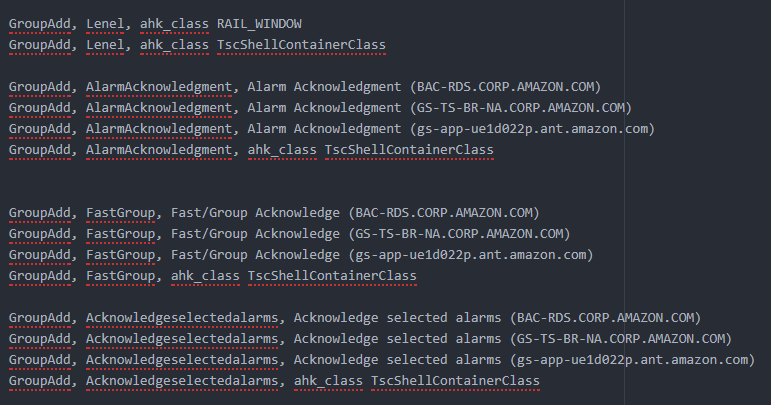
Lenel is running through a remote desktop. This means that the computer running Lenel and the computer running AA are separate PC’s. If the computer running Lenel lags or temporarily refuses to receive input, the computer running AA or sending the input will not know and continue running the code. Luckily there are ways to solve this. Most Lenel windows have specific naming conventions and using certain functions (WinWait, IfWinNotActive, WinActivate) you can have AA wait before continuing. One good way to get the window name of Lenel windows is to use the “Active Window Info (Window Spy)” program. This program will install as part of the AHK installation package.

Active Window Info Program:



As outlined in the picture above, you can see the window’s Title, Class, and Process. Unfortunately, knowing the Lenel window name and knowing how to wait for specific windows to appear using the WinWait, IfWinNotActive, and WinActivate functions does not solve all of our interaction issues.

There are different types of ways that you can connect to Lenel and different Lenel servers that all have their own unique Titles. In order to know which window you need to wait for, you can’t be too specific so the “Title” is not usable. You’re going to have to use the “Class” categorizer. In order to make the program accessible on all versions of Lenel but still wait for specific windows you will need to create groups then use the WinWait, IfWinNotActive, and WinActivate functions specifically looking for those groups. Outlined below are the current groups I have created within AA that work well identifying the required Lenel winssdows.

Groups:

Now that most of the Lenel interaction is covered I’ll explain to you some things you might notice when reviewing the code. Ferst off let’s go over this line.



You will see this line all over the code as it is a known issue with AHK. If the code ends without this line being present there is a small percentage change that a key will be held down. There is no way to undo this once the bug has occurred and it is highly inconvenient. This line fixes that so I have added it to all possible endings.