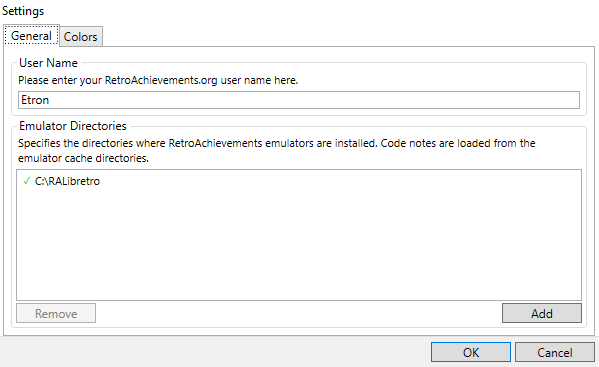
# Tutorial 0

## Getting Started

Before you start scripting you will need to download the newest release of [RATools](<https://github.com/Jamiras/RATools/releases>). The program needs a little setup before you can use it:

1. Open the settings menu by clicking on \*File -> Settings…\*
2. Enter your in name in the User Name textbox
3. Click \*Add\* and browse to the file location of RALibRetro. The icon next to the file location will change to a green checkmark if the emulator was correctly detected.
4. (Optional) Click on the \*Colors\* tab and click on the \*Dark Theme\* button.
5. Press the OK button at the bottom of the screen.

![Screenshot of the settings dialog](RATools\_Settings.PNG)



## Importing Code

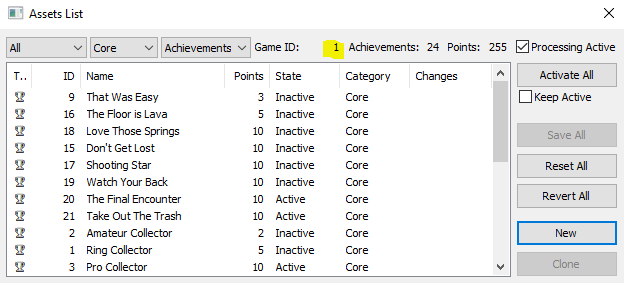
When first learning RAScripts it can be a little daunting to know where to start. One good method of learning how to write scripts is to import existing achievements into RATools. This tutorial will focus on how to import a script into RATools, how to clean up the script if necessary, and how to update changes back in to RALibRetro. There are several cases were you would want to import code with RATools:

* Learning how to code a specific achievement
* Fixing tickets
* Converting existing code to an RAScript
* Creating Rich Presences and/or Leaderboards

If you are just learning RAScripts then it is extremely useful to import a set your familiar with to see how it looks in script form. To import a new script:

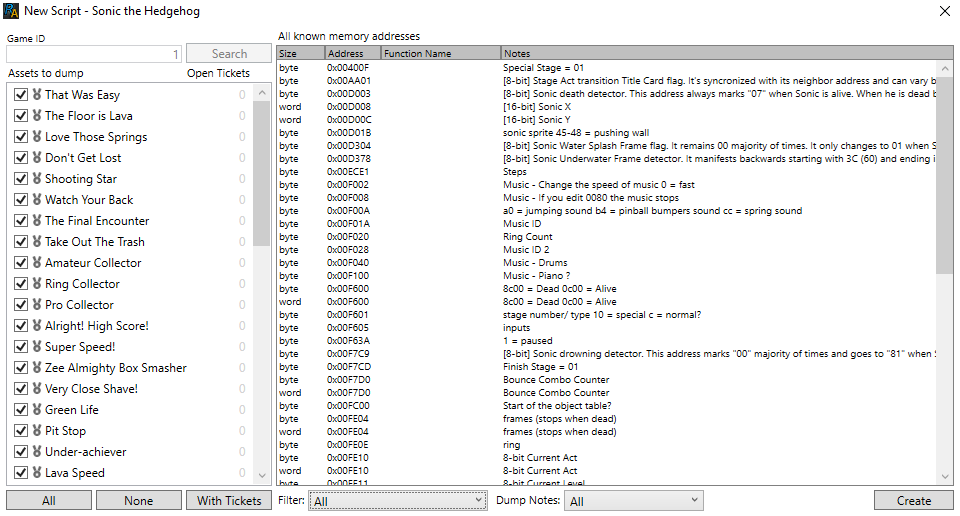
1. Load and play a game in RALibRetro (or any of the RA emulators that support editing achievements). You only need to start playing the game which will trigger downloading the necessary achievement data to the RACache folder.
2. Record the game id number which can be found on the top of the menu item \*RetroAchievements -> Asset List\*. For this example we will be looking Sonic the Hedgehog which is game id 1.

![Screenshot of finding the game ID in the assets list](Assets\_List.PNG)



1. Open RATools and click on the menu item \*File -> New Script\*
2. Enter the Game ID number in the top left and press the \*\*Search\*\* button
3. All of the achievements, rich presence, and leaderboards will show up on the left panel and all of the game notes will show up on the right.
4. Add a checkmark next to the achievements you want to import and select how you want to filter code notes. When in doubt use All for the both the Filter and Dump Notes so that all of the code notes will be included in the new script.
5. Press the \*\*Create\*\* button to have RATools generate the script

![Screenshot of creating a new script](New\_Script.PNG)



## Cleaning up Code

When you import a new script, RATools will generate scripts with the memory addresses and values. Below is an example of what the achievement “Amateur Collector” looks like when imported.

```

achievement(

title = "Amateur Collector", description = "Collect 20 Rings", points = 2,

id = 2, badge = "00199", published = "2012-11-02 12:03:12 AM", modified = "2021-03-01 5:48:45 PM",

trigger = word(0x00FE20) >= 20 && word(0x00FFF0) == 0 && byte(0x00FFFB) == 0 && byte(0x00F601) == 12

)

```

To make the script easier to understand and maintain it is recommended that you replace the memory address and constant values with variables. By using the find/replace function (Ctrl+H) in RATools you can replace all of the instances of the same memory with variables quickly. For this example we will replace:

* word(0x00FE20) -> Rings
* word(0x00FFF0) -> DemoMode
* word(0x00FFFB) -> DebugMode
* byte(0x00F601) -> StageType
* 12 -> SpecialStage

```

Rings = word(0x00FE20)

DemoMode = word(0x00FFF0)

DebugMode = word(0x00FFFB)

StageType = byte(0x00F601)

SpecialStage = 12

achievement(

title = "Amateur Collector", description = "Collect 20 Rings", points = 2,

id = 2, badge = "00199", published = "2012-11-02 12:03:12 AM", modified = "2021-03-01 5:48:45 PM",

trigger = Rings >= 20 && DemoMode == 0 && DebugMode == 0 && StageType == SpecialStage

)

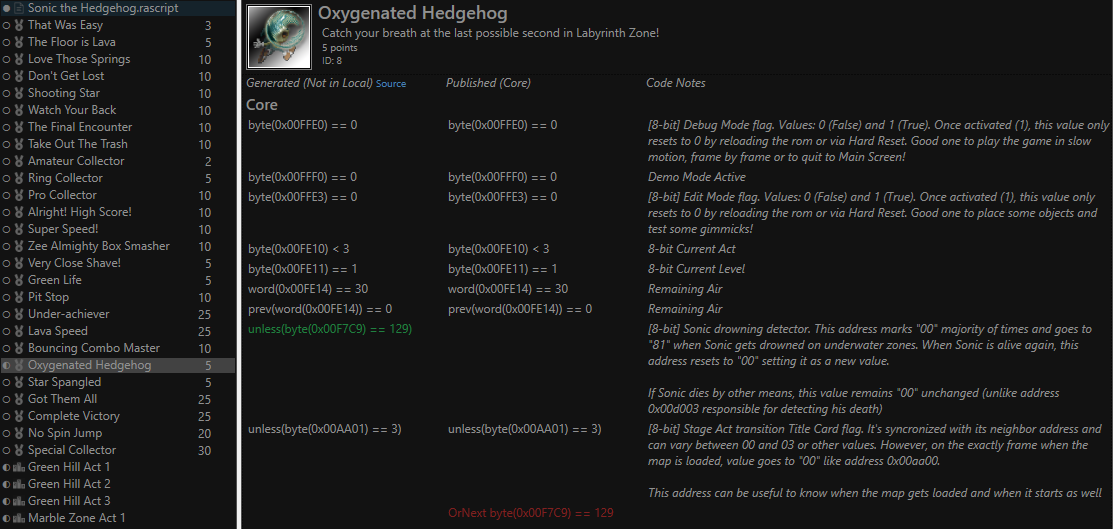
```

Notice how you can easily read the above function and understand what is going on. By using the find and replace method any achievement that uses the memory word(0x00FE20) will now reference Rings instead. The next step would be to use functions for similar types of achievements. For example, the above ring collecting achievement has a 100 and 200 rings variation where the only difference is number of rings to collect. There is an example of how to make a function for this type of achievement in [Example 1A](../01\_Memory/Example\_1A.md).

## Comparing Code

If you wish to see how the script changes you made compare to what’s in the core you can select the achievement from the list on the left panel. A filled in circle next to achievement indicates that it is new and does not exist in core, a half circle indicates the achievement has changed from the core, and a hollow circle means that the script matches the core. Lines added by script will be colored green and lines remove will be colored red. To get back to the code view click on the script name at the top left. Note that RATools has some optimizations built in so it will automatically improve existing code. The example below shows such a case where the optimizations has made the code cleaner although no changes were made in the script.

![Screenshot of comparing the script with the code in the Core](Code\_Compare.PNG)



## Updating Code

Once you are ready to test you can export the updated code back into RALibRetro with \*Edit -> Update Local\*. Note that the script must be error free to proceed with the update. Select all of the achievements you wish to update and press the \*OK\* button. You new updates will appear next time you open RALibRetro. If RALibRetro was open during the update you’ll need to select the achievements that were updated and press the “Reset” button to reload them from the local cache.