Template Toolkit

Technical Documentation

Adding a template set

To add a new set of templates we will need to add the title of the group and the list of different options we need to a set of arrays, then create a map of templates that corresponds with those reasons.

For example if you wanted to add the following:

- ExampleTopic
    - ExampleReason1
    - ExampleReason2

Find the relevant department section in the Templates.ahk file. Each department will have two arrays:

- One array for **"Reasons"** (which will populate the first dropdown).
- One array for **"Templates"** (which will populate the second dropdown based on the selected reason).

For example, if you are adding "ExampleTopic", your task is to update both the "Reasons" array and the corresponding "Templates" array. The reason we need to update both of these, is that updating one will lead to errors when running the program.

In the Templates.ahk file:

- **Step 1**: Add a new item to the end of the “Reasons” array ("ExampleTopic").
    - **Step 2**: Update the Templates array with the list of templates
    - ("ExampleReason1", " ExampleReason2")

After updating these arrays, the first dropdown will show "ExampleTopic" as an option under "Reasons," and selecting this will populate the second dropdown with "ExampleReason1" and "ExampleReason2".

Think of this process in three sections:

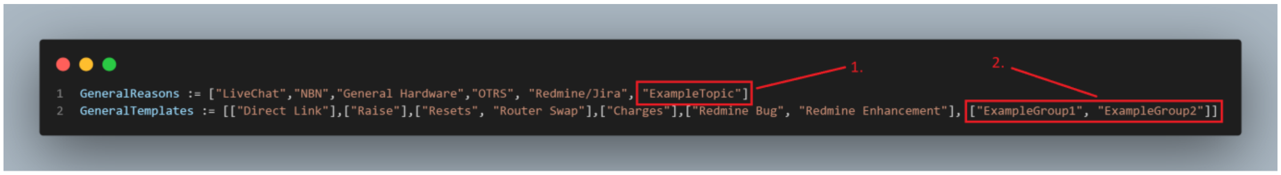
- Dropdown 1: The overall theme or group title of the templates you are adding (e.g., "ExampleTopic").

- Dropdown 2: The different categories of templates under that theme (e.g., "ExampleReason1", " ExampleReason2").

- Templates Box: The individual templates that will be associated with the selected category (you'll add these templates later).

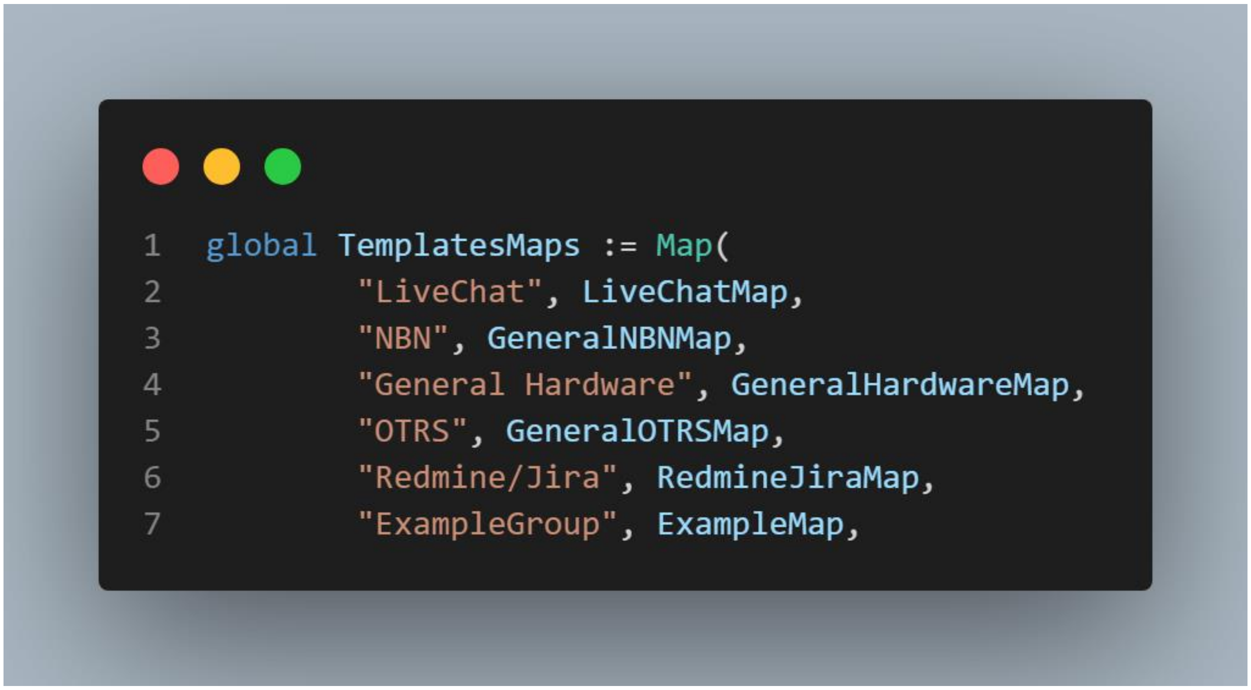
Step 1.
We will add the overall group to the "Reasons" array

Step 2.
Add a new array to the "Templates" group. This needs to be done following the existing format of the file. The new array will look like this ["ExampleReason1", " ExampleReason2"]. This needs to be inside the existing set of square brackets and separated by a comma.



```
GeneralReasons := ["LiveChat","NBN","General Hardware","OTRS", "Redmine/Jira", "ExampleTopic"]
GeneralTemplates := [["Direct Link"],["Raise"],["Resets", "Router Swap"],["Charges"],["Redmine Bug", "Redmine Enhancement"], ["ExampleGroup1", "ExampleGroup2"]]
```

Once this is done we will need to associate the new group with a place to store the template text itself.
This can be done in the section just below the arrays. The left hand side in quotes needs to match the group that you added to the "Reasons" array in the previous step. The map naming conventions in this section is typically the name of the group with Map added to the end of it.

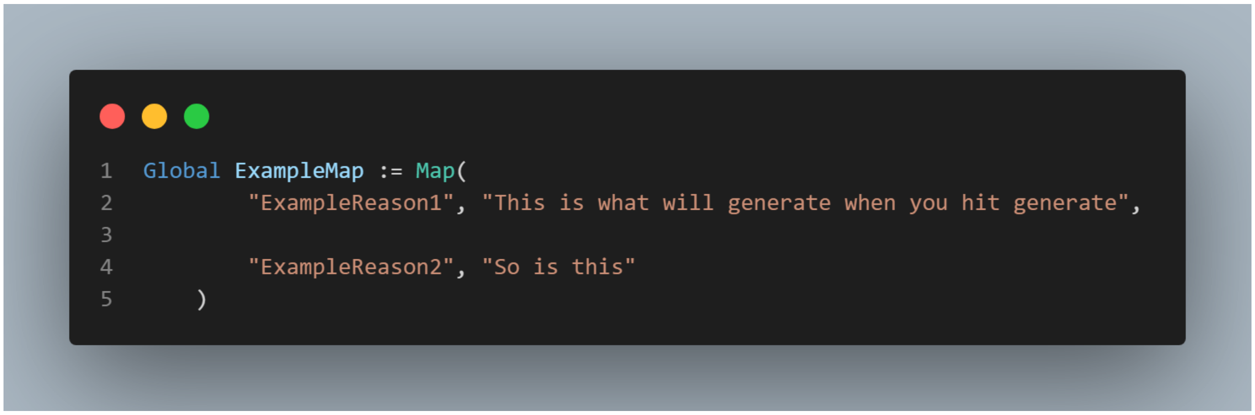


```
global TemplatesMaps := Map(
        "LiveChat", LiveChatMap,
        "NBN", GeneralNBNMap,
        "General Hardware", GeneralHardwareMap,
        "OTRS", GeneralOTRSMap,
        "Redmine/Jira", RedmineJiraMap,
        "ExampleGroup", ExampleMap,
```

Once the “Reason” is associated to a map we need to build the templates map to go along with it.

Step 1.
Scroll down to the list of maps and find the relevant department. This is not technically necessary but helps keep the code readable.

Step 2.
Enter the following code. The map name needs to match the map you listed in the previous step. This is a key-value pair. The first set of quotes on each line contains the text you added to the “Templates” array above and will tell the GUI which template to pull when you select the dropdowns. The second set of quotes on the line contains the actual template text you would like to generate.

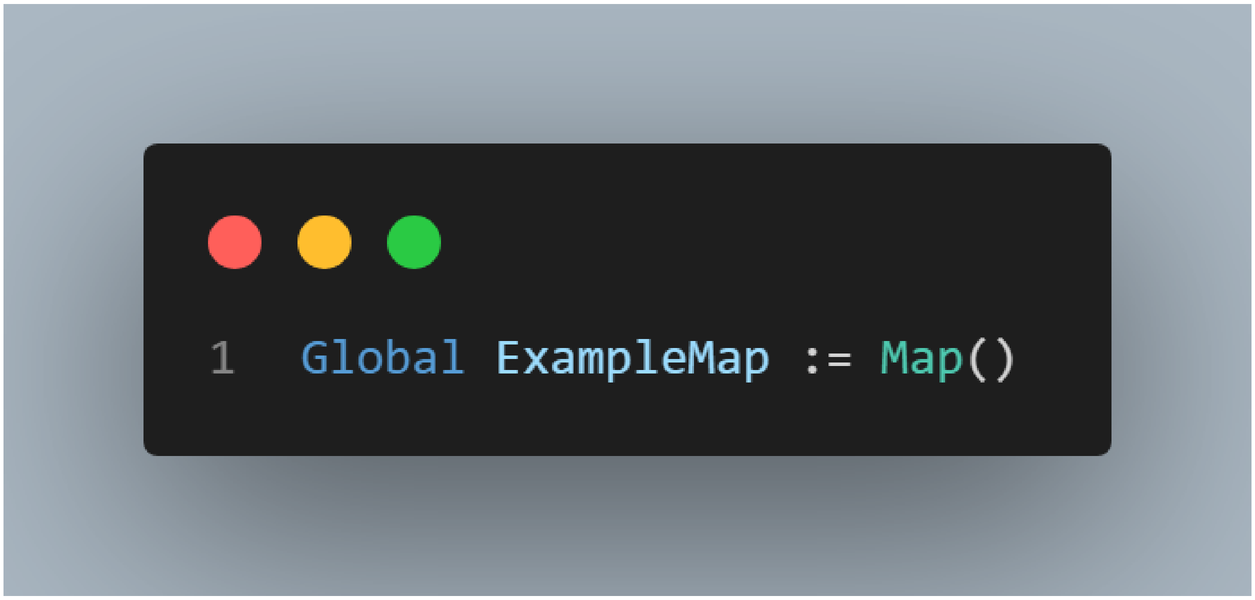


```
Global ExampleMap := Map(
        "ExampleReason1", "This is what will generate when you hit generate",

        "ExampleReason2", "So is this"
    )
```

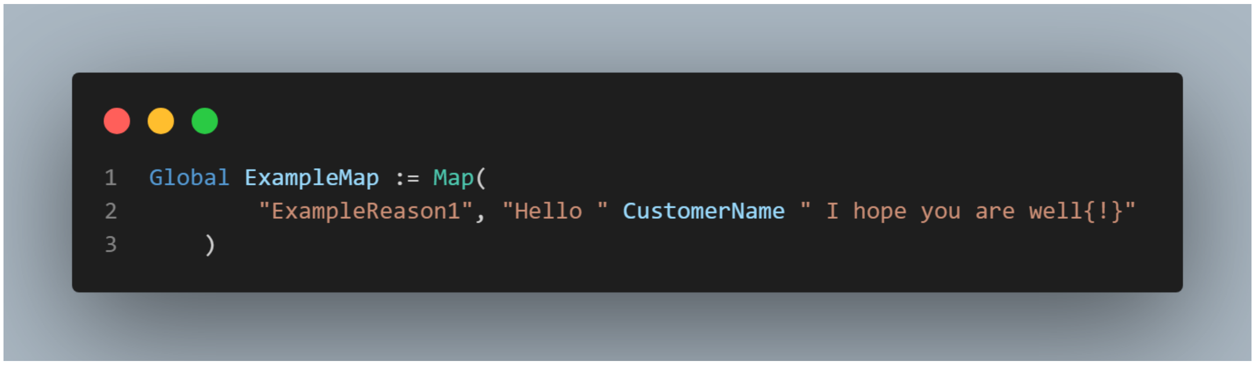
Finally, we need to make sure the main script can work out where this newly created map actually is. If we had all the code in one file this would be easy, but to keep things readable the code is split into multiple files.

Go into the main toolkit file, name BuddyToolkit or AussieToolkit, and add the following line to the list of global variables right at the top of the file



```
Global ExampleMap := Map()
```

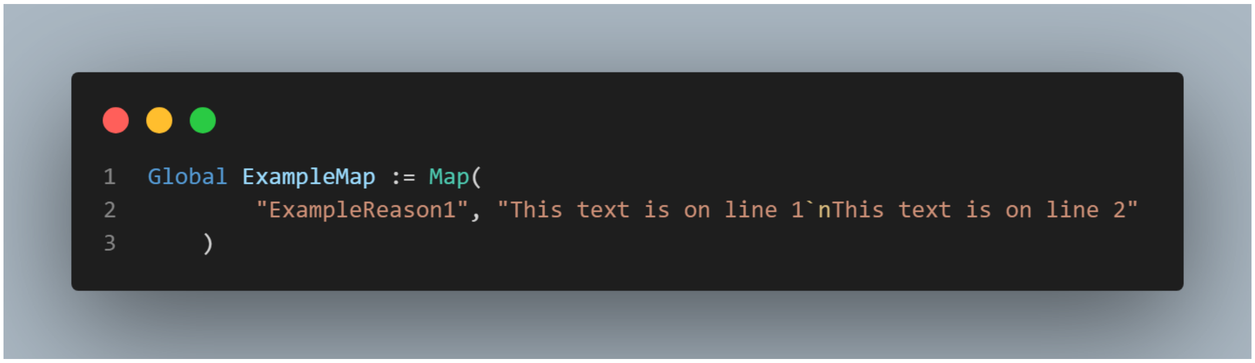
If you would like to include the customer's name, or any other variable contained in the other sections of the script you need to break out of the double quotes in the template text. Below is an example:



```
Global ExampleMap := Map(
        "ExampleReason1", "Hello " CustomerName " I hope you are well{!}"
    )
```

Note that in this example I used an exclamation mark. AHKv2 will attempt to process this as code even though it is in a string. You can avoid this by containing most symbols in curly braces.

If you wish to have line breaks in your templates, you can use the following characters “`n” as below:



```
Global ExampleMap := Map(
        "ExampleReason1", "This text is on line 1`nThis text is on line 2"
    )
```