Structural Overview

The following basic functions are common to each of the plugins. They perform some rudimentary operations such as initialization of the plugin, executing the query that has been entered, loading context menu icons, updating settings when configurations are altered in the settings UI, and updating the theme of the icons when the theme changed event is triggered.

IPlugin Interface

Each plugin implements the `IPlugin` interface which comprises of the `Init()` and `Query()` functions.

`Init`

- The `Init()` function initializes the context, storage and settings of each plugin. This is equivalent to a constructor and is the first function to be called in the `Main.cs` file for each plugin.

`Query`

- For every query that the user enters into PT Run, the `PluginManager.cs` executes the `Query()` function in the `Main.cs` file corresponding to each Plugin.

Context Menu Icons

- The `ContextMenus` are loaded for each result based on the type of the result.
- The various types of `ContextMenu` functionalities are:
  - Open containing folder
  - Run as Administrator
  - Open in console
  - Copy path

UpdateSettings

- This function updates the settings of each plugin based on the changes made by the user in the settings UI.
- Eg: To disable drive detection in the indexer plugin, when the user checks or unchecks the drive detection check box, the `UpdateSettings()` function dispatches the changes in the check box to the plugin.

ThemeChanged

- This function is invoked when there is a change in the theme of PT Run.
- It is used to update the `IconPath` for each plugin based on the theme.

Save

- This function saves the configurations of each plugin so that they can be loaded the next time.

Score

- The user query is executed against each of the plugins and the result list view is updated with results from each of the plugins.
- The ordering of the results is based on the `Score` of each Result.
- Each plugin assigns a score to a result based on it's relevance. The results with higher scores are displayed higher in the list view and vice versa.

Plugin settings

Plugin settings that are editable from the settings are stored in `PowerToys Run\settings.json`. In the very first run, those settings are populated from plugin' `plugin.json` file. Unlike Wox we do not support multiple action keywords. Instead, we have `ActionKeyword` and `IsGlobal` options.