Settings Utilities

- Abstractions for each of the file/folder related operations are present in `SettingsUtils.cs` .
- To reduce contention between the settings process and runner while trying to access the `settings.json` file of any powertoy, the settings process tries to access the file only when it needs to load the information for the first time. However, there is still no mechanism in place which ensures that both the settings and runner processes do not access the information simultaneously leading to `IOExceptions` .

Utilities

GetSettings<T>(powertoy, filename)

- The GetSettings function tries to read the file in the powertoy settings folder and creates a new file with default configurations if it does not exist.
- Ideally this function should only be called by the `SettingsRepository` which would be accessed only when a powertoy settings object is being loaded for the first time.
- The reason behind ensuring that it is not accessed elsewhere is to avoid contention with the runner during file access.
- Each of the objects which are deserialized using this function must implement the `ISettingsConfig` interface.