TODO: MANAGER

Website: www.mitchcroft.games

Contact: me@mitchcroft.games

Introduction

The purpose of the *TODO: Manager* is to allow developers to manage the elements of code that remain within a project that are yet to be completed. It is an editor extension that scans plain text assets included within the assets, parsing the information via customisable parameters to pick out elements that are of interest.

While intended for assisting in the development of code through the default settings of looking for *TODO* segments, the search parameters are modifiable to the point that they can be used to scan any number of plain text files for your specific needs.

Contents

Features

1.0.0

- **Modifiable search locations**, to refine the directories that will be searched
- **Modifiable ignore locations**, to exclude specific directories from being searched
- **Modifiable file extensions**, to define the types of files you want searched and displayed
- **Customisable search parameters**, for greater control over what you want to be detected within the searched files
- Default, **ready made profile for scanning C# and JavaScript** script files for TODO comments
- **Multi-threaded** support for quick scanning of all the included files within a project, regardless of how large
- **Auto-updating** of the scanned files when the files are changed
- **Caching of file information** to prevent re-scanning of files where it is not required
- Support for **multiple search profile settings**, to support different types of scanning as required
- Supports the **ad hoc inclusion of filters** to parsed sections to allow for greater organisation of your files on-the-go without needing to setup specific categories
- **Search functionality**, to allow for the finding of specific sections of interest
- **Adjustable display window settings**, to achieve the look that works for you

Usage

Setup

To begin searching your assets and organising your work flow, you need to start with a TDMProfile that is used to define what is being searched and how it will be parsed. This is the object that will also hold the cached information about the plain text assets that have been processed. It is the central data object of the package.

To facilitate quick inclusion in a project and setup there is also a default profile that is setup for use with C# and JavaScript files as well as a completely blank profile for individuals to customise what they would like identified.

Both are available through the *Create* menu within the Project panel under the **TODO: Manager sublocation**. Selecting one will create a new profile at the current selection location within the project hierarchy.
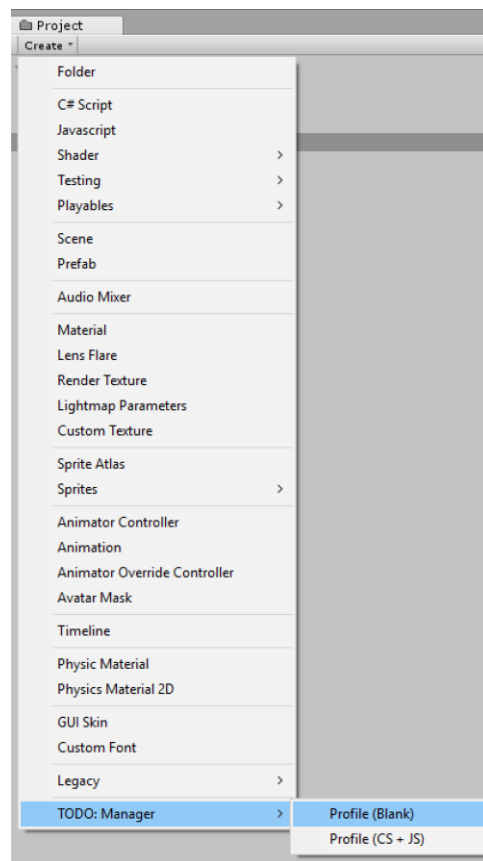

Create a new profile via *Create/TODO: Manager/Profile*

When creating a blank profile, the settings within the profile will need to be defined fully before it can be used. To start quickly, you can create the default C# and JavaScript profile and skip to the TODO: Manager Display Window section of this document to get started. For a breakdown of what the profile settings do, continue reading.

Define Profile Settings

The **search criteria** used by the profile is **exposed** to allow individual control over the specifics of what is processed by the package. This section will walk through the specifics of settings up the default C# and JavaScript TODO search profile with an **explanation on how the different sections effect the process:**
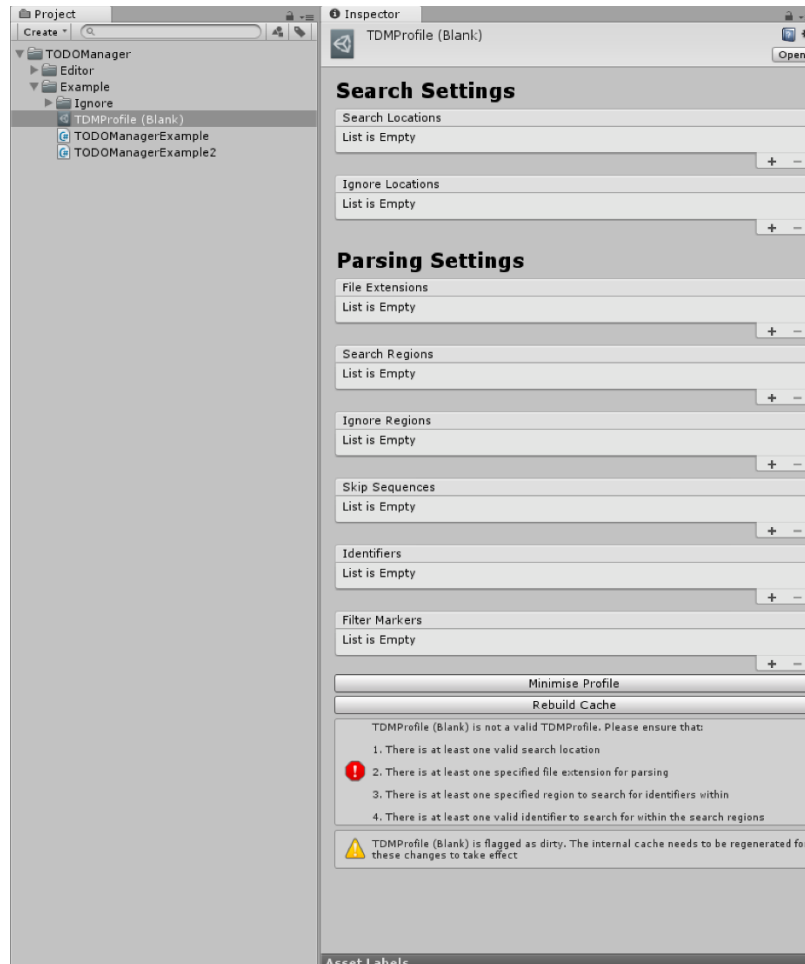
A default, blank profile and the different settings that are modifiable

Warning Messages

There are two primary messages that can be displayed in the Inspector window for a TDMProfile asset (As seen above).

The first message is an **error that displays when there are some problems within the current settings**, causing the profile to be invalid. **While the profile is invalid it will not be usable** with the Display Window and will need to be fixed.

The second is a warning, indicating the profile is currently dirty and as such has an invalid cache. Whenever a setting within the **profile is modified or updated, the profile will become dirty** and need to be re-processed. **Dirty profiles will have their cached information ignored** and all files within the search locations will be processed again. This is done to account for the new settings that need to be applied to the search and parsing process.

Search Locations

Only assets within the projects asset folder are scannable, but the locations within this folder can be specified to best suit your project's needs. Each **profile needs at least one search location** for the profile to be valid. The **search is recursive** from the defined location and will look down the folder hierarchy for additional files (Unless omitted by the ignore locations).
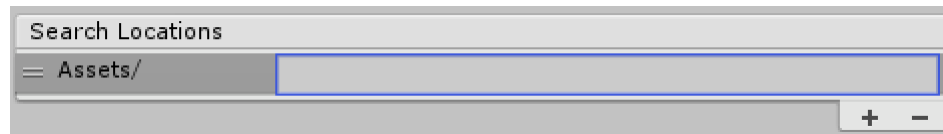

Adding a **blank option to the list will scan the entire projects files**, but there can be as many individual search locations as you would like however they need to be a valid folder location.
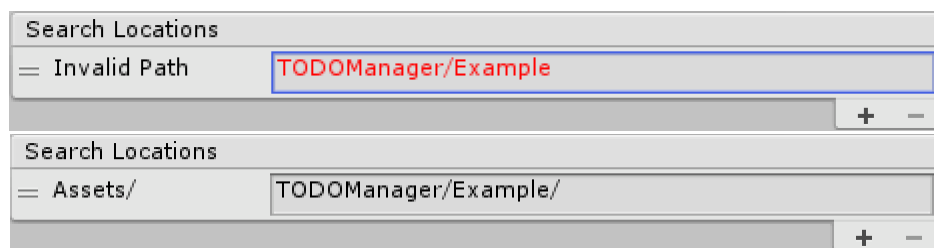
For a file path to be considered valid, it **must be capped by a directory separating character**. As can be seen in the images above, without the trailing slash, the path is coloured red and labelled as invalid. Valid directories will be displayed in the standard font colour and show the *'Assets/'* label at the beginning of the definition. It should be noted that the presence of any **directory control sequences (I.e. './' or '../') will cause the path to be marked as invalid**.

Ignore Locations

The ignore locations can be used to **specify directory locations within search locations that should be omitted from the process**. **Ignore paths can be omitted** if you want to scan everything within the search locations. The **rules for defining ignore locations are the same as they are for the search locations**.
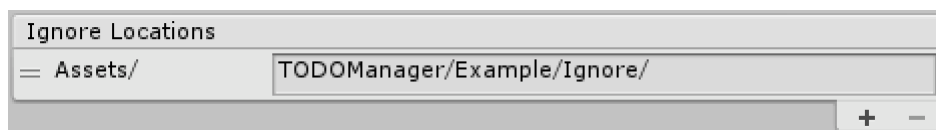
The major difference is that the **ignore locations will only be considered if they are a child directory of one of the search locations** (As seen above).
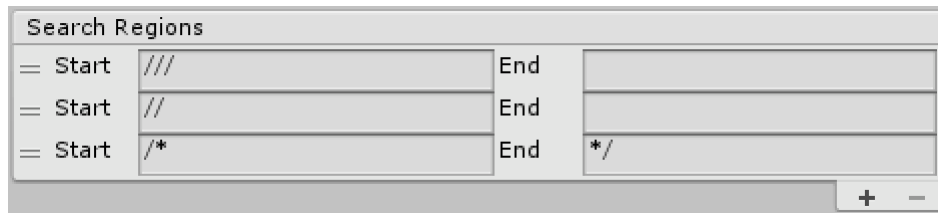
File Extensions
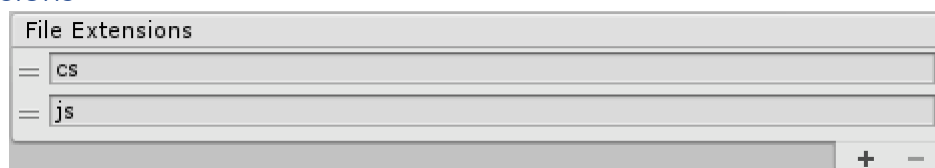
The file extensions **define the type of files that will be parsed** when scanning the assets folder. These should just be the extension characters by themselves, and **the '.' Character will be automatically**

removed from any entered text (See above). There can be as many different file extensions as you require.

Search Regions

These definitions **define the areas within a file that will be searched** for the important information. Not all aspects of a file can be displayed, so this allows for the narrowing down of the specific sections that are to be considered. When looking for TODO comments within C# and JavaScript files, only comment sections need to be considered.
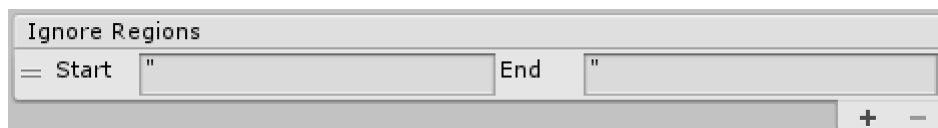
```
Search Regions
═  Start   ///                        End
═  Start   //                         End
═  Start   /*                         End   */
                                              +   −
```

The standard comment sequences that are used are the traditional single line comment '//', Visual Studio XML style region comments '///' and the multi-line comment '/* */' sequences. The first two are simpler in the sense that they are **in effect until they reach the end of the line they are defined on**. For regions like this, **no end sequence is required**, and the search will stop when the end of the line is reached. **For the multi-line comment however, the end sequence can be defined to describe when the region is over.**

Ignore Regions

These definitions work much in the same way that search regions do, with the obvious difference being that **text within these areas is completely skipped over**. This is **useful for omitting elements that may be intentionally defined** as something but would otherwise appear in the display window. For example, given the code snippet:

```
string tutorialText = "Programmers often leave //TODO: Comments for them to return to later!";
```

This text tutorial would be captured in the search otherwise as it follows the criteria (See Identifiers below), but as this is intentional, we would rather that this is ignored.

```
Ignore Regions
═  Start   "                          End   "
                                              +   −
```
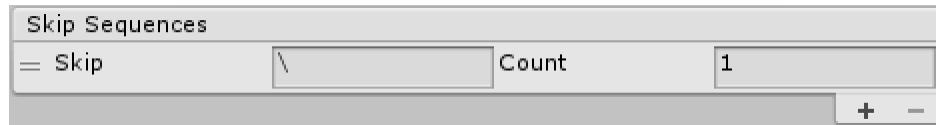
By defining the beginning and ending of a string sequence as an ignore region, the parsing process will skip over it and ignore the comment that is stored within. **Ignore regions can be omitted if you don't wish to exclude anything.**

Skip Sequences

These are used to **describe a sequence of characters (1 or more) that can be used to indicate that an additional number of characters should also be passed over**. Returning to the above tutorial text example, consider if you wanted to add punctuation to the string:

```
string tutorialText = "Programmers often leave \"//TODO: Comments\" for them to return to later!";
```

Without defined skip sequences, the parsing process reaches the '' character that is just before the '//TODO:' sequence, this would indicate to the parsing process that the end of ignore region has been reached and the intentional comment would be included again. To prevent this, skip sequences can be used to omit control characters:
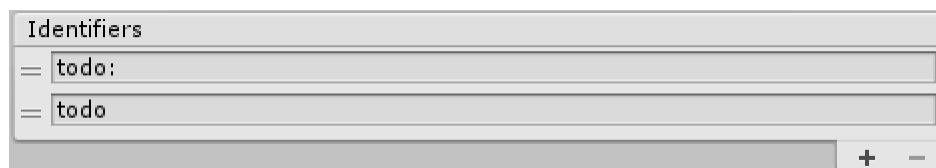
Skip Sequences				
= Skip	\	Count	1	
				+ −

This defines the '\' character as a section that should be skipped. **The parsing process will then consume this character and the specified number of characters after it** (in this case 1) before resuming. It is worth noting that **the sequence of characters can be of any length. Skip sequences can also be omitted if you don't wish to skip anything.**

Identifiers

Identifiers are used to **indicate at the beginning of a search region that the text enclosed within the region should be processed and displayed**. This allows us to prevent every single comment from being displayed and catch only the specific information that we are interested in. In the TODO example, we are only interested in comments that begin with the 'TODO' sequence of characters. To capture this, we can define any number of identifiers to be captured:
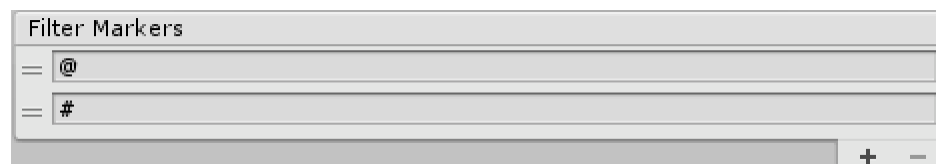
Identifiers	
= todo:	
= todo	
	+ −

When displayed in the editor window, Identifiers are stripped from the sequence of characters that are shown. By including a second identifier with the trailing ':' character, any comments where the TODO was written with a colon will be stripped and removed as well as the 'TODO' section. This means that the following text will show cleanly without the ':' preceding the important aspect.

Filter Markers

Filters are **used for ad hoc organisation of extracted sections** that are displayed. By adding these sections within an identified region, **the text that follows them will be extracted and used as a label** that can be **used to filter the elements that are displayed** (See Filters in the display window).

| Filter Markers	
= @	
= #	
	+ −

Given the above defined markers, the code comment below will create two labels that contain the comment. This is especially useful for **organising a large quantity of comments in more complex projects.**

```
//TODO #High importance, @Person add the field values required for the object
```

Control Buttons

The control buttons allow for the manual application of the processes that are applied to the profiles when they are displayed in the window.

| Minimise Profile |
| Rebuild Cache |
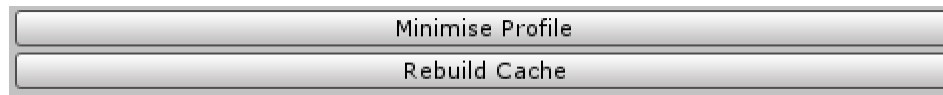
- **Minimise Profile** – Removes all values from the profile that are invalid or have no impact on the processing of files (E.g. duplicate entries)
- **Rebuild Cache** – This will process the profile, scanning the required files and updating the internal cache information (Note: processing the profile via this method will not be multi-threaded)

TODO: Manager Display Window

The display window is used to show the sections that have been identified by the profile. It can be found on the **toolbar** through *Window/TODO: Manager* or by pressing the **keyboard shortcut CTRL + SHIFT + T. There can be any number of TODO: Manager display windows open.** This can be useful if you have more then one profile that you are actively using and need to access all values at once.

Additionally, there are two different rendering modes based on the desired positioning and size of the window. **If wider then it is tall, the window will display in landscape mode. If the window is taller than it is wide, then it will enter a more vertical friendly arrangement** for the information. The following screen captures where taken in landscape mode.

Profile Selection

Before the display can be opened, a valid profile must be selected for use. When the window is opened **the first TDMProfile asset found in the project will be assigned to the use field**. This can be useful if you want to quickly progress and only have one profile. I**f you have multiple to choose from then the profile field will allow you to search the available options**, or you can drag your desired profile into the field. **If the chosen profile is invalid, an error message will be shown** prompting you to go to the profiles settings and fix what errors there are.
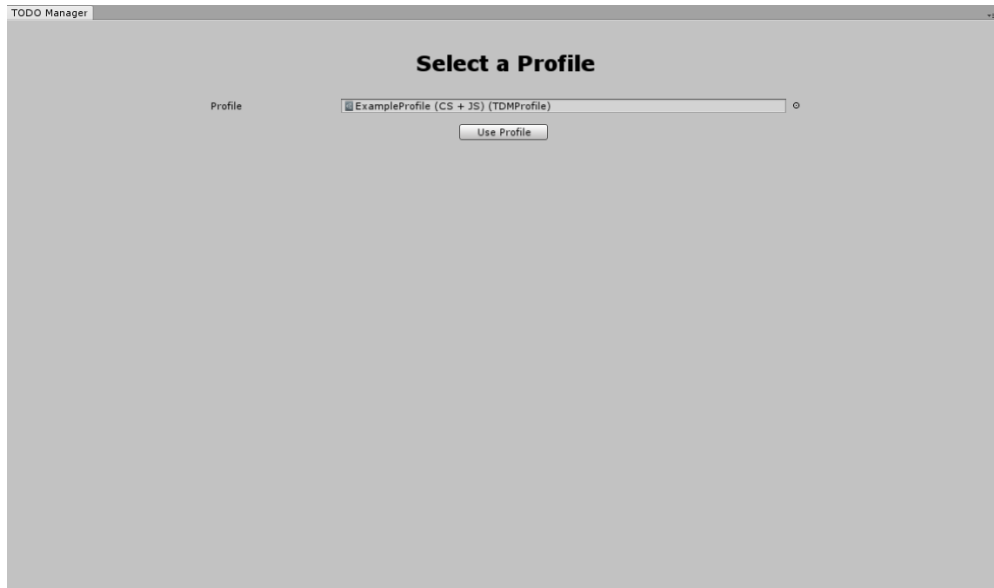
Main Display

This screen is broken down into three different areas that are used to control the display of information that was identified when processing the active profile:
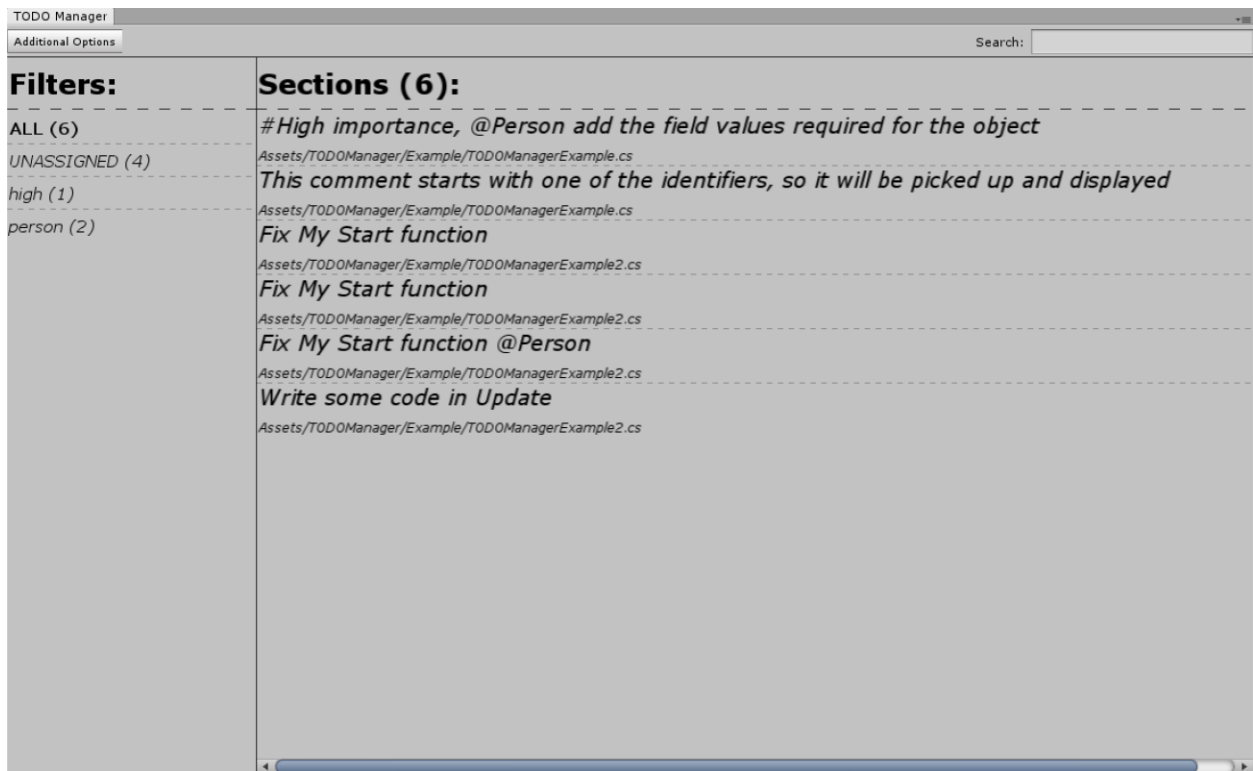
1. Filters

The filters section **displays the various ad hoc labels that were found** in the processed file sections. There are **two default categories** that are included in the menu:

- **ALL** – The master option that displays all the sections that were identified in the profile at once. This is the default option and can be used to get a quick glance at everything that is currently in the project.
- **UNASSIGNED** – This option is only present if there are identified sections that have no filters applied to them. This can be used to quickly identify elements that may need to be returned to for classification or to allocate them filters after the fact.

Clicking on the different filter options changes which sections are currently being displayed. Clicking on a section while **holding CTRL will combine the filters that are applied** and show all sections with the selected filters within the section's menu. The **number of sections that contain the filter label are displayed next to them within the brackets**. This list will grow with the number of filters that are added in the sections that are processed.

2. Sections

The sections menu displays all the extracted sections that are defined by the active selected filters and the current search string. These **sections can be selected by clicking on them, and the file can be opened to the location within the file by clicking again on a selected section.** This can be useful for quickly jumping to a specific section and continuing work. Within the sections menu there will be a segment of the section text (See Display Settings for more) as well as the file that the section is located in. The current number of displayed sections is shown in the brackets next to the section header.

3. Command Bar

The command bar offers some additional options that can be used to perform different functions:

- **Search** – On the top-right is the search field which can be used to further refine the sections that are displayed. The text within this field is searched for within the extracted section's text and its file path.
- **Additional Options** – This is a dropdown menu of multiple other options
  - **Profile Selection** – This option returns the display window back to the profile selection screen
  - **Display Settings** – Transitions to the Display Settings menu that can be used to change the visuals of the main display (See Display Settings)
  - **Re-process Assets** – Force the profile to update with the current sections within the profiles elected files. This is useful if Auto-Initialise is not turned on (See Extras)
  - **Find Active Profile** – This selects the active TDMProfile asset within the project window
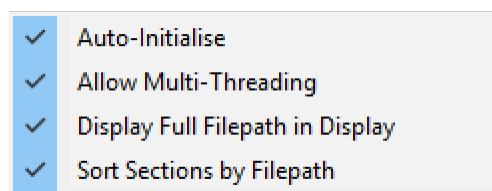
Display Settings

This menu allows for the modification of the display values of the main screen. These values are saved for will persist across instances of the TODO: Manager Display Window:

- **Filter/Section Size Scale** – This allows for the applying of a scale to the visual elements. Lowering the scale will shrink the main display items, allowing more to be shown on screen. Raising the scale will enlarge the text and show less items.
- **Section Comment Length** – This will adjust the number of characters that are shown in the display area for the sections. Large section extracts may slow down the rendering and restricting it can help with performance issues. A value of -1 will remove the character limit from what is displayed.

Extras

By right-clicking on the window during any state (Profile Selection, Main Display or Display Settings), you will bring up the extra's menu with toggleable settings:

- **Auto-Initialise** – When this is enabled, if the project assembly is reloaded then the profile will automatically be re-processed to include any updates that may have been made to the project files.

- **Allow Multi-Threading** – If this is enabled (and the computer supports multi-threading) the identified files will be processed in parallel among the available processors. In large projects this can help speed up the processing time.
- **Display Full File Path in Display** – If this is enabled, in the Section area of the Main Display the file that the section is in displays as the full file path. If disabled, only the files name and extension will be displayed. With multiple files of the same name, this can be useful for telling them apart.
- **Sort Sections by File Path** – If this is enabled then the displayed sections will be sorted based on the file they are located in. Disabling this will list them in the order that they were found in.

In-Asset Files

With this profile now setup for **finding TODO statements within script files**, we can start **adding these comments to the scripts within the search location and have them start showing up within the display** window. For example, we can create various comments that can be returned to later:

```
void Start() {
    /// TODO Fix My Start function
    //todo Fix My Start function
    //Todo Fix My Start function @Person
    /*
        TODO: @Person needs to determine what needs to be generated on Start. This is of #high importance
    */
}
```

These comments will then be translated into the result that can be seen within the display window

Closing

If you are having any issues with this package or have any feature suggestions, please contact me at
me@mitchcroft.games