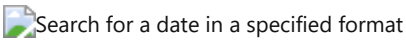
'Time and Date' plugin

The 'Time and Date' plugin shows the date and time in different formats. For the date and time formats the plugin uses the culture setting in Windows, if the format is not commonly defined. The user can search for the system date/time or a custom date/time. The value of each result can be copied to clipboard.

Query examples:

- Format: `time`
- Date/time: `10:30 AM`
- Format and date/time: `Week number::10/10/2022`

 'Time and Date' plugin

 Search for a date in a specified format

Formats

Available formats

Remarks

- The following formats requires a prefix in the query:
  - Unix Timestamp: `u`
  - Windows file time: `ft`
- On invalid number inputs we show a warning that tells the user which prefixes are allowed/required.

List of available formats

The following formats are currently available:

Format	Example (Based on default settings)	As result	As input
Time	5:10 PM	x	x
Date	3/5/2022	x	x
Now	3/5/2022 5:10 PM	x	x
Time UTC	4:10 PM	x	x
Now UTC	3/5/2022 4:10 PM	x	x
Unix Timestamp	1646496622	x	x
Hour	10	x	
Minute	30	x	
Second	45	x	
Millisecond	678	x	

Day (Week day)	Saturday	x	
Day of the week	6	x	
Day of the month	5	x	
Day of the year	64	x	
Week of the month	1	x	
Week of the year (Calendar week, Week number)	10	x	
Month	March	x	
Month of the year	3	x	
Month and day	March 7	x	x
Year	2022	x	
Era	AD	x	
Era abbreviation	A	x	
Month and year	March 2022	x	x
Windows file time (Int64 number)	637820976123938199	x	x
Universal time format: YYYY-MM-DD hh:mm:ss	2022-03-05 16:20:12Z	x	x
ISO 8601	2022-03-05T17:23:04	x	x
ISO 8601 UTC	2022-03-05T16:23:04	x	x
ISO 8601 with time zone	2022-03-05T17:23:04+01:00	x	x
ISO 8601 UTC with time zone	2022-03-05T16:23:04Z	x	x
RFC1123	Sat, 05 Mar 2022 16:23:04 GMT	x	x

Add new formats

- To add a new formats you have to add them to the method `GetList()` of the `AvailableResultsList` class.
  - Please add the new formats in the second range. The first one is reserved for the three main formats (Time, Date, Now).
- After adding the new formats you have to update the Unit Tests!

Optional plugin settings

- The optional plugin settings are implemented via the `ISettingProvider` interface from `Wox.Plugin` project.
- All available settings for the plugin are defined in the `TimeDateSettings` class of the plugin. The settings can be accessed everywhere in the plugin code via the static class instance

`TimeDateSettings.Instance`.

- We have the following settings that the user can configure to change the behavior of the plugin:

Key	Default value	Name/Description
`OnlyDateTimeNowGlobal`	`true`	Show only 'Time', 'Date', and 'Now' result on global queries
`TimeWithSeconds`	`false`	Show time with seconds (Applies to 'Time' and 'Now' result)
`DateWithWeekday`	`false`	Show date with weekday and name of month (Applies to 'Date' and 'Now' result)
`HideNumberMessageOnGlobalQuery`	`false`	Hide 'Invalid number input' error message on global queries

Classes

`AvailableResult.cs`

- Each instance of the `AvailableResult` class represents a time/date result/format that the user can search for.
- The results/formats are defined in the `AvailableResultsList` class.

`AvailableResultsList.cs`

- The `AvailableResultsList` class contains the list of available formats/results in its method `GetList()`.

`ResultHelper.cs`

- The `ResultHelper` class contains methods for some of the result features (tool tip, copy to clipboard) and the error result on incorrect number input.
- And it contains the `SelectStringFromResources()` method for getting the resource strings based on the user input.
  - The method has a parameter for the `stringId` which is the name of the string in the resource file. By default the word `Now` is automatically added at the end to get the string for a system time/date search.
  - If a different/custom string is needed for a system time/date search the parameter `stringIdNow` can be used to override the default behavior of the method.
  - If only a string for the system time/date search is required, you can set `stringId` to `string.Empty` and only `stringIdNow` to a valid string id.

`TimeAndDateHelper.cs`

- The `TimeAndDateHelper` class contains methods to format/convert date and time formats/strings.

`TimeDateSettings.cs`

- The `TimeDateSettings` class provides access to all optional plugin settings.

- The class has a static property called `Instance` that holds an instance of the class itself. This allows us to access the settings from everywhere in the plugin code without having additional parameters in our methods.

`SearchController.cs`

- The `SearchController` encapsulates the methods needed to search and find matches.

Search

Tags

- We compare the user input with the label of each results. If it doesn't match we search the tags of the result too.
- For each result two tag strings are defined. One for a search with system time/date and one for a search with a custom time/date. Most of the results (except the era results) are using one of the generic tag lists: Date, Time or Format
- The selection of the tag (for "system time/date" or "custom time/date") is happening at search time in the `AvailableResultsList.cs` class.
- The different tags in a list are split by the `;` character.

Score

- The plugin uses `FuzzyMatching` to get the matching formats, if the user searches for a specific format. The score is set based on the `FuzzySearch` result.
- To achieve a better balance between sub title matches and tag matches the score of tag matches is divided by two.

Match requirements for global queries

On global queries the high score returned by `FuzzySearch` has negative impacts on the user experience and the search results priority/order of other plugins. To mitigate this we defined some matching requirements:

- If the query is a word of the following conjunction list, we don't return any results: for, and, nor, but, or, so
  - We don't have 'yet' (synonym of 'now') on the list, because this could block results in some languages.
- The first word of the query has to be a full match with a word in the label or tag list.
- For both requirements we compare case-insensitive.

Unit Tests

We have a Unit Test project that executes various test to ensure that the plugin works as expected.

`TimeDateResultTests.cs`

- The `TimeDateResultTests.cs` class contains tests to validate that the time and date values are correctly formatted/calculated.
- That we can execute the tests at any time on any machine, we use a specified date/time value and set the thread culture always to `en-us` while executing the tests.
- Some tests contain checks that calculate the expected result at runtime instead of using an expected value written fix in the code. This is done to get valid results on every machine at any time.

`ImageTests.cs`

- The `ImageTests.cs` class contains tests to validate that each result shows the expected and correct image.
- That we can execute the tests at any time on any machine, we set the thread culture always to `en-us` while executing the tests.

`PluginSettingsTests.cs`

- The `PluginSettingsTests.cs` class contains tests to validate that all settings exist and that they have the correct default values.

`QueryTests.cs`

- The `QueryTests.cs` class contains tests to validate that the user gets the correct results when searching.,
- That we can execute the tests at any time on any machine, we set the thread culture always to `en-us` while executing the tests.

`StringParserTests.cs`

- The `StringParserTests.cs` class contains tests to validate that the typed string gets converted correctly into a `DateTime` object.
- That we can execute the tests at any time on any machine, we set the thread culture always to `en-us` while executing the tests.