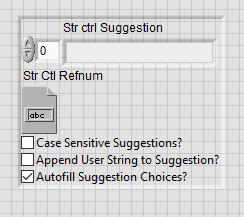
## High Level Requirements

1. Only few\necessary simple API’s will be exposed to the developer to drop in their block diagram to configure the IntelliSense for their string control
   1. One or many string control from a same VI can be configure with this
   2. Provide polymorphic VI if possible for the configuration
2. Developer should be able to specify the common and string control specific IntelliSense option
   1. Specify the specific suggestion text in description of string control
   2. Specify the suggestion text through a simple text file or a string input
   3. Specify the suggestion dynamically\programmatically
   4. Suggestion could be case sensitive\insensitive
   5. Suggestion will be provided for each String Control separately



1. Suggestion UI will be customizable (Suggestion IO library)
2. Suggestion business logic is not customizable but efficient to display quick results and configurable search intelligence

## Design Overview

### Different libraries and the dependency linkage

Quick Suggestions library

LabVIEW Code

Suggestion IO library

IntelliSense String Control

String IntelliSense API exposed

String IntelliSense library – Exposed API layer responsible for interacting with the User Code, Gets the string reference to monitor, Gets the Suggestions texts and Gets the Suggestion IO type through exposed API

* This is like a String Control (XControl \ QControl)
* Contains an engine to control the properties of it

Quick Suggestions library – Business logic to get and suggest the list of possible text based on what is types in the string control. Strings monitor library will provide the information on what is being typed on the string control with context to the suggestions library.

Suggestion IO library – this is class based structure. By default parent class provides the suggestion as an array of strings and have an API through which user can trigger back the selected text.

* Based on how the suggestion has to be provided to the user, appropriate child object of the Suggestion IO library will be chosen

### String IntelliSense Library API’s

#### Instantiate (Suggestion IO + String ref + String Suggestion List)

* Add string control & suggestion text [single]
* Add string control & suggestion text [many] – appends
* Start Monitoring – when the first IntelliSense String Control is created

#### Add common suggestion text

#### Destroy / Destroy all

* Stop Monitoring – when no IntelliSense String control available

### Quick Suggestion library

### Suggestion IO library

#### Default \ Parent implementation

String Monitor library will have one Suggestion IO library created with it using which the interaction with the example VI or user happens.

Default or Parent Suggestion IO library has two methods exposed (public), 1. Get Suggestions and 2. Set Selected Suggestion text

User can query the Suggestion IO object from the ‘string monitor library’ and call these methods to perform actions.

Additionally, Suggestion IO provides an Event refnum, that developer can register in their VI and it will generate event whenever a suggestion is available for user or a suggestion is selected.