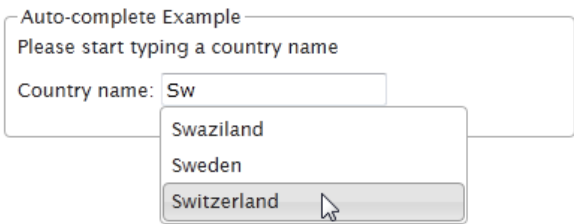
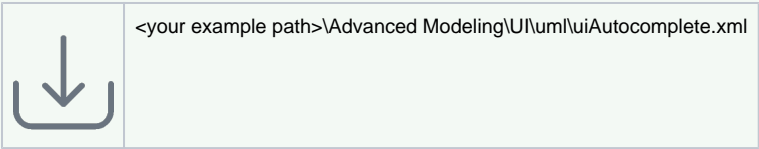


Auto complete

The auto complete widget is a type of extension for a text field. While the end user is typing into the text field, he gets a suggestion list displayed and can choose from this list. In the corresponding example project a suggestion list pops up right below the text field giving the user a list of possible countries starting with the letters he typed into the field. The autocomplete functionality does not restrict the user to only use the values listed. The list is just a suggestion to the user.

Example File (Builder project Advanced Modeling/UI):



On this Page:

- [Autocomplete Setup](#)
- [Autocomplete Service Call](#)

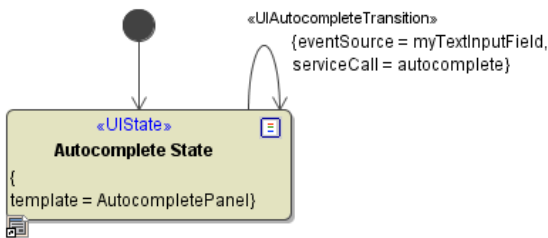
Related Pages:

Usage of the UI Widgets:

- [Autocomplete](#)
- [Checkbox](#)
- [Combo Box](#)
- [Date Chooser](#)
- [Multi-select Lists](#)
- [Radio Button Group](#)
- [Tables](#)
- [Tabs](#)

Autocomplete Setup

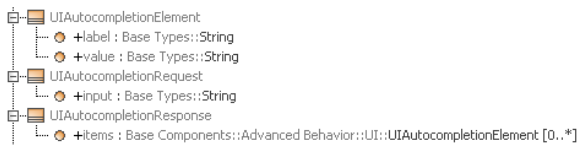
The autocomplete mechanism is implemented by using a `<<UIAutocompleteTransition>>` within the web pages state machine. The stereotype is set on a **Transition to Self** as shown in the figure below.



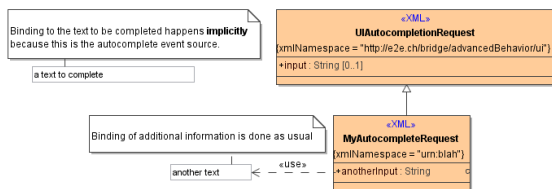
There are two tagged values which need to be set on this `<<UIAutocompleteTransition>>` being the **Service Call** to the service which delivers the suggestion list data and the actual **Event Source**, the text field, which should offer the auto complete feature.

Autocomplete Service Call

The service call for this feature needs to implement the built in input and output types **UIAutocompletionRequest** and **UIAutocompletionResponse**. They are part of the **Base Components** and can be found in the corresponding repository package. The input data of the request object is a string value (input) taken **automatically** from the eventSource **myTestInputField**. The response object consists of an multiplicity of **UIAutocompetionElement** object. Last is a simple key value pair construct (label, value). The request and response objects do not need to be bound to the text element. This is all done automatically.



However, sometimes it is necessary to add additional information to the autocomplete request to do the appropriate lookups. In this case just take as input type a class derived from **UIAutocompletionRequest** as shown in the following figure. There the input type for the autocomplete request is the class **MyAutocompleteRequest**:



In the autocomplete example a database is used to retrieve the suggestion list according to the end user's input. Here a possible implementation using database requests:

