

# UICONSTANTS

## Constants Summery

- String** [#DATE\\_FORMATTERS\\_LENIENT](#)  
Property that can be set using application.
- String** [#DATE\\_FORMATTERS\\_ROLL\\_INSTEAD\\_OF\\_ADD](#)  
Property that can be set using application.
- Number** [#FULL\\_SCREEN](#)  
Value that can be used to specify that a dialog/window should completely fill the screen.
- String** [#TYPE\\_AHEAD\\_SHOW\\_POPUP\\_ON\\_FOCUS\\_GAIN](#)  
Property that can be set using application.
- String** [#TYPE\\_AHEAD\\_SHOW\\_POPUP\\_WHEN\\_EMPTY](#)  
Property that can be set using application.

## Constants Details

### [DATE\\_FORMATTERS\\_LENIENT](#)

Property that can be set using `application.setUIProperty()` or `element.putClientProperty()`.

If set on application it will affect all date formatted fields. CAUTION: this property must be set on application before the fields are created (for example in solution onOpen handler). Changing it after fields were created will not affect these existing fields.

If set on an element it will affect only that date formatted element/field (with priority over the application property).

Value can be true/false/null.

If set to false, date formatted fields will not allow input of out-of-bounds values (like 62 minutes means 2 minutes and +1 hour).

DEFAULT: true.

#### Returns

[String](#)

#### Sample

```
// make all date formatted fields (created after this line is executed) not use lenient mode
application.setUIProperty(APP_UI_PROPERTY.DATE_FORMATTERS_LENIENT, false);
// make one date formatted field use lenient mode - overrides the application property set
forms.someForm.elements.typeAheadElement.putClientProperty(APP_UI_PROPERTY.DATE_FORMATTERS_LENIENT, true);
```

### [DATE\\_FORMATTERS\\_ROLL\\_INSTEAD\\_OF\\_ADD](#)

Property that can be set using `application.setUIProperty()` or `element.putClientProperty()`. It is used only in Smart Client.

If set on application it will affect all date formatted fields. If set on an element it will affect only that date formatted element/field (with priority over the application property).

Value can be true/false/null.

If set to true, only selected part of the date will be affected when using up/down keys to cycle through values. (for example, pressing up when cursor is on minutes and minutes shows 59 will not result in hour change)

DEFAULT: false.

#### Returns

[String](#)

#### Sample

```
// make all date formatted fields use roll instead of add
application.setUIProperty(APP_UI_PROPERTY.DATE_FORMATTERS_ROLL_INSTEAD_OF_ADD, true);
// make one date formatted field use add instead of roll - overrides the application property set
forms.someForm.elements.typeAheadElement.putClientProperty(APP_UI_PROPERTY.DATE_FORMATTERS_ROLL_INSTEAD_OF_ADD, false);
```

### [FULL\\_SCREEN](#)

Value that can be used to specify that a dialog/window should completely fill the screen.

#### Returns

[Number](#)

### Sample

```
application.showFormInDialog('customers',APP_UI_PROPERTY.FULL_SCREEN,APP_UI_PROPERTY.FULL_SCREEN,  
    APP_UI_PROPERTY.FULL_SCREEN,APP_UI_PROPERTY.FULL_SCREEN,'My Title',true,true,'customers_dialog',true)
```

### TYPE\_AHEAD\_SHOW\_POPUP\_ON\_FOCUS\_GAIN

Property that can be set using `application.setUIProperty()` or `element.putClientProperty()`.

If set on application it will affect all TYPE-AHEAD fields. If set on an element it will affect only that TYPE-AHEAD element/field (with priority over the application property).

Value can be true/false/null.

If set to true, the affected TYPE\_AHEAD(s) will not show the pop-up when gaining focus.

DEFAULT: true.

### Returns

[String](#)

### Sample

```
// make all TYPE-AHEAD fields not show the pop-up when gaining focus  
application.setUIProperty(APP_UI_PROPERTY.TYPE_AHEAD_SHOW_POPUP_ON_FOCUS_GAIN, false);  
// make one TYPE-AHEAD field show the pop-up when gaining focus - overrides the application property set  
forms.someForm.elements.typeAheadElement.putClientProperty(APP_UI_PROPERTY.TYPE_AHEAD_SHOW_POPUP_ON_FOCUS_GAIN,  
true);
```

### TYPE\_AHEAD\_SHOW\_POPUP\_WHEN\_EMPTY

Property that can be set using `application.setUIProperty()` or `element.putClientProperty()`.

If set on application it will affect all TYPE-AHEAD fields. If set on an element it will affect only that TYPE-AHEAD element/field (with priority over the application property).

Value can be true/false/null.

If set to true, the affected TYPE\_AHEAD(s) will not show the pop-up when the field content is empty.

DEFAULT: true.

### Returns

[String](#)

### Sample

```
// make all TYPE-AHEAD fields not show the pop-up when there is empty content in the field  
application.setUIProperty(APP_UI_PROPERTY.TYPE_AHEAD_SHOW_POPUP_WHEN_EMPTY, false);  
// make one TYPE-AHEAD field show the pop-up when there is empty content in the field - overrides the  
application property set  
forms.someForm.elements.typeAheadElement.putClientProperty(APP_UI_PROPERTY.TYPE_AHEAD_SHOW_POPUP_WHEN_EMPTY,  
true);
```