

UICONSTANTS

Constants Summary

String	#COMBOBOX_SHOW_POPUP_ON_FOCUS_GAIN	Property that can be set using application.
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.
String	#LEAVE_FIELDS_READONLY_IN_FIND_MODE	Property than can be set using application.
String	#TOOLTIP_DISMISS_DELAY	Property than can be set using application.
String	#TOOLTIP_INITIAL_DELAY	Property than can be set using application.
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.
String	#USE_SYSTEM_PRINT_DIALOG	Property than can be set using application.

Constants Details

COMBOBOX_SHOW_POPUP_ON_FOCUS_GAIN

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

If set on application it will affect all COMBOBOX fields. If set on an element it will affect only that COMBOBOX element/field (with priority over the application property).
Value can be true/false/null.

If set to true, the affected COMBOBOX will show the pop-up when gaining focus.

DEFAULT: true.

Returns

String

Sample

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

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);
```

LEAVE_FIELDS_READONLY_IN_FIND_MODE

Property than can be set using `application.setUIProperty()`.

If set to true, fields that are read-only won't be editable in find mode

If set to false, fields that are read-only will be editable in find mode

The value can be true/false

DEFAULT: false

Returns

[String](#)

Sample

```
application.setUIProperty(APP_UI_PROPERTY.LEAVE_FIELDS_READONLY_IN_FIND_MODE, true)
```

TOOLTIP_DISMISS_DELAY

Property than can be set using `application.setUIProperty()` and indicates the delay in milliseconds after the tooltip is dismissed.

Returns

[String](#)

Sample

```
application.setUIProperty(APP_UI_PROPERTY.TOOLTIP_DISMISS_DELAY, 4000)
```

TOOLTIP_INITIAL_DELAY

Property than can be set using `application.setUIProperty()` and indicates the delay in milliseconds before the tooltip is shown.

Returns

[String](#)

Sample

```
application.setUIProperty(APP_UI_PROPERTY.TOOLTIP_INITIAL_DELAY, 2000)
```

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);
```

USE_SYSTEM_PRINT_DIALOG

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

If set to true, the system standard Printing dialog will be used when printing is needed.

If set to false, the Servoy Printing dialog will be used.

The value can be true/false/null.

DEFAULT: false

Returns

[String](#)

Sample

```
application.setUIProperty(APP_UI_PROPERTY.USE_SYSTEM_PRINT_DIALOG, true)
```