

# UICONSTANTS



Apr 05, 2024 01:42

Supported Clients

SmartClient WebClient NGClient

## Constants Summary

|        |                                          |                                                               |
|--------|------------------------------------------|---------------------------------------------------------------|
| String | ALLOW_JAVASCRIPT_LINK_INPUT              | Property that can be set using element.                       |
| String | CALENDAR_NG_SHOW_ISO_WEEK_NUMBER         | Property that can be set using application.                   |
| String | COMBOBOX_ENABLE_FILTER                   | Property that can be set using application.                   |
| 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 | HTML_EDITOR_CONFIGURATION                | Property that can be set on editable html area using element. |
| String | LEAVE_FIELDS_READONLY_IN_FIND_MODE       | Property that can be set using application.                   |
| String | NG_BLOCK_DUPLICATE_EVENTS                | Property that can be set using application.                   |
| String | RELATED_NULL_SEARCH_ADD_PK_CONDITION     | Property that can be set using element.                       |
| String | TABLEVIEW_NG_OPTIMIZED_READONLY_MODE     | Property that can be set using application.                   |
| String | TABLEVIEW_NG_PAGE_SIZE_FACTOR            | Property that can be set using application.                   |
| String | TABLEVIEW_WC_DEFAULT_SCROLLABLE          | Property that can be set using application.                   |
| String | TABLEVIEW_WC_SCROLLABLE_KEEP_LOADED_ROWS | Property that can be set using application.                   |
| String | TABLEVIEW_WC_USE_KEY_NAVIGATION          | Property that can be set using application.                   |
| String | TOOLTIP_DISMISS_DELAY                    | Property that can be set using application.                   |
| String | TOOLTIP_INITIAL_DELAY                    | Property that can be set using application.                   |
| String | TRUST_DATA_AS_HTML                       | Property that can be set using element.                       |
| 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 that can be set using application.                   |
| String | VALUELIST_MAX_ROWS                       | Property that can be set using application.                   |

## Constants Details

### ALLOW\_JAVASCRIPT\_LINK\_INPUT

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

If set to true, the element will accept javascript links in the input  
If set to false, all 'javascript:' texts will be removed from the input

The value can be true/false  
DEFAULT: false

#### Returns

String

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
element.putClientProperty(APP_UI_PROPERTY.ALLOW_JAVASCRIPT_LINK_INPUT, true)
```

### CALENDAR\_NG\_SHOW\_ISO\_WEEK\_NUMBER

Property that can be set using `application.putClientProperty()`, preferably in solution onOpen handler (or anyway before forms containing calendars are shown).

If set to true, the default calendar, bootstrap calendar, bootstrap inline calendar and nggrid calendar will show week number according to ISO 8601. Other 3rd party (calendar) components are free to take this value into consideration as they please.  
By default (false) those will show week number according to locale.

The value can be true/false  
DEFAULT: false

#### Returns

[String](#)

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
application.putClientProperty(APP_UI_PROPERTY.CALENDAR_NG_SHOW_ISO_WEEK_NUMBER, true)
```

### COMBOBOX\_ENABLE\_FILTER

Property that can be set using `application.putClientProperty()`. It is used only in NGClient.

This is a global setting, it will affect all COMBOBOX fields. It must be set as soon as possible, ie. on solution open.  
Value can be true/false/null.

If set to false, ALL COMBOBOXes will hide the search box when gaining focus.  
DEFAULT: null.

#### Returns

[String](#)

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
// make all COMBOBOX fields hide the search box when gaining focus
application.putClientProperty(APP_UI_PROPERTY.COMBOBOX_ENABLE_FILTER, false);
```

### COMBOBOX\_SHOW\_POPUP\_ON\_FOCUS\_GAIN

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

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](#)

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
// make all COMBOBOX fields not show the pop-up when gaining focus
application.putClientProperty(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.putClientProperty()` 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](#)

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
// make all date formatted fields (created after this line is executed) not use lenient mode
application.putClientProperty(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.putClientProperty()` 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](#)

#### Supported Clients

SmartClient,WebClient,NGClient

#### Sample

```
// make all date formatted fields use roll instead of add
application.putClientProperty(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);
```

### HTML\_EDITOR\_CONFIGURATION

Property that can be set on editable html area using `element.putClientProperty()`

The value must be a valid json string according to TinyMCE editor configuration (<http://www.tinymce.com/wiki.php/configuration>).  
 It will be used to override/set configuration properties in order to customize the editor.

#### Returns

[String](#)

#### Supported Clients

SmartClient,WebClient,NGClient

**Sample**

```
// adding a new TinyMCE plugin (same for skin or theme)
// this code assumes plugin.min.js was copied in web server at specified path
// NOTE: we use external plugin, not plugin, in order for file to be accessible from web server; for this
// example, plugin.min.js file must be copied under ServoyInstall/application_server/server/webapps/ROOT/tinyMCE
//plugins/link
element.putClientProperty(APP_UI_PROPERTY.HTML_EDITOR_CONFIGURATION, '{"external_plugins": { "link": "../..
/tinyMCE/plugins/link/plugin.min.js"}}')

// change the editor configuration (add menubar, status bar and change toolbar)
element.putClientProperty(APP_UI_PROPERTY.HTML_EDITOR_CONFIGURATION, '{"menubar": "tools table format view
insert edit", "statusbar" : true, "toolbar": "undo redo | styleselect | bold italic"}')
```

**LEAVE\_FIELDS\_READONLY\_IN\_FIND\_MODE**

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

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](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

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

**NG\_BLOCK\_DUPLICATE\_EVENTS**

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

If set to true, any events of same type and on same component will be blocked (cancelled) until first event is finished.

The value can be true/false  
 DEFAULT: false

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
application.putClientProperty(APP_UI_PROPERTY.NG_BLOCK_DUPLICATE_EVENTS, true)
```

**RELATED\_NULL\_SEARCH\_ADD\_PK\_CONDITION**

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

If set to true, related find/search will only return records that have a related match, also in case of left outer joins.  
 Otherwise a related search on a field may return records where the related search does not match.

For example,  

```
* <pre>
    if (foundset.find()) {
        foundset.myleftouterjoinrelation.myfield = 'someval';
        foundset.search();
    }
</pre>
```

With this setting to false records of the foundset table that have no related records via the relation will also be returned.

The value can be true/false  
 DEFAULT: servoy property servoy.client.relatedNullSearchAddPkCondition/true

**Returns**[String](#)**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
application.putClientProperty(APP_UI_PROPERTY.RELATED_NULL_SEARCH_ADD_PK_CONDITION, false)
```

**TABLEVIEW\_NG\_OPTIMIZED\_READONLY\_MODE**

Property that can be set using application.putClientProperty().

If set to true, the tableview will be seen as fully readonly and NGClient will generate an optimized version (textfields are replaced)

The value can be true/false  
DEFAULT: false

**Returns**[String](#)**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
application.putClientProperty(APP_UI_PROPERTY.TABLEVIEW_NG_OPTIMIZED_READONLY_MODE, true)
```

**TABLEVIEW\_NG\_PAGE\_SIZE\_FACTOR**

Property that can be set using application.putClientProperty().

A number that defines the factor of what the next page size should be is in the tableview/listview/portal, this value is used to get the initial size (numberOfVisibleRows \* thisPageSize). So a value of 2 (default) will load in 20 records if the number of visible rows is 10. Then if you scroll down the new set of records will be: (numberOfVisibleRows \* thisPageSize) - numberOfVisibleRows so that will load for the default value 2, 1 page which is the number of visible rows (10 in this example).

The value can be any number but it should be bigger than 1.

WARNING the bigger the number, the more data is pushed initially to the client (and more is pushed in every new page)

DEFAULT: 2

**Returns**[String](#)**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
application.putClientProperty(APP_UI_PROPERTY.TABLEVIEW_NG_PAGE_SIZE_FACTOR, 3)
```

**TABLEVIEW\_WC\_DEFAULT\_SCROLLABLE**

Property that can be set using application.putClientProperty().

If set to true, table views in web client are scrollable by default  
If set to false, table views in web client are not scrollable, but pageable by default

The value can be true/false  
DEFAULT: false

When this property is set to true, you can control the size of the page by setting the servoy property "servoy.webclient.scrolling.tableview.multiplier". The default value is 2. Setting the property to a higher value, will result in more data to be queried at once. You can also set it to a lower value, like 1 or 1.5 for example. We strongly recommend that the default or lower size be used in order to avoid blocking situations due to the big request being made to the server.

**Returns**[String](#)

---

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
application.putClientProperty(APP_UI_PROPERTY.TABLEVIEW_WC_DEFAULT_SCROLLABLE, true)
```

**TABLEVIEW\_WC\_SCROLLABLE\_KEEP\_LOADED\_ROWS**

Property that can be set using application.putClientProperty().

If set to true, scrollable table views in web client will keep the already loaded rows in the view  
If set to false, scrollable table views in web client will unload not visible rows in the view

The value can be true/false  
DEFAULT: false

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
application.putClientProperty(APP_UI_PROPERTY.TABLEVIEW_WC_SCROLLABLE_KEEP_LOADED_ROWS, true)
```

**TABLEVIEW\_WC\_USE\_KEY\_NAVIGATION**

Property that can be set using application.putClientProperty().

If set to true, you can change selection in webclient tableview using up/down keys  
If set to false, you cannot change selection via keyboard arrows

The value can be true/false  
DEFAULT: true

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
application.putClientProperty(APP_UI_PROPERTY.TABLEVIEW_WC_USE_KEY_NAVIGATION, false)
```

**TOOLTIP\_DISMISS\_DELAY**

Property that can be set using application.putClientProperty() and indicates the delay in milliseconds after the tooltip is dismissed.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

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

**TOOLTIP\_INITIAL\_DELAY**

Property that can be set using application.putClientProperty() and indicates the delay in milliseconds before the tooltip is shown.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

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

**TRUST\_DATA\_AS\_HTML**

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

If set to true, data showed on elements like buttons or labels will not be sanitized.

Showing unsanitized data can make the system vulnerable to XSS attacks, for example, an user registers with name 'John Doe<script>someEvilJavascript</script>', when this data is shown in a label (by another user) the javascript in the script tags will be executed.

Only enable this setting if the data shown can always be trusted and is never composed of data from an external system or user.

The value can be true/false  
DEFAULT: false

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
element.putClientProperty(APP_UI_PROPERTY.TRUST_DATA_AS_HTML, true)
```

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

Property that can be set using `application.putClientProperty()` 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 show the pop-up when gaining focus.  
DEFAULT: true.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
// make all TYPE-AHEAD fields not show the pop-up when gaining focus
application.putClientProperty(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.putClientProperty()` 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 show the pop-up when the field content is empty.  
DEFAULT: true.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
// make all TYPE-AHEAD fields not show the pop-up when there is empty content in the field
application.putClientProperty(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.putClientProperty()`.

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](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

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

**VALUELIST\_MAX\_ROWS**

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

The value can be a positive integer representing the maximum number of rows that will be retrieved by query in database or related valuelist.  
Can have a maximum value of 1000.

DEFAULT: 500

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient

**Sample**

```
application.putClientProperty(APP_UI_PROPERTY.VALUELIST_MAX_ROWS, 1000)
```