

UTIL INTERFACE

View online: <https://www.construct.net/en/make-games/manuals/addon-sdk/reference/ui-interfaces/utilities>

The `SDK.UI.Util` interface provides access to various user interface utilities in the SDK.

Methods

AddDragDropFileHandler(callback, opts)

Register a callback for handling files drag-and-dropped in to the Construct 3 window. This is part of the Custom Importer API, allowing addons to handle importing files in a custom format. The given callback is invoked when a file is dropped in to the Construct 3 window, providing nothing else has handled it first. The callback must return a promise that resolves with `true` if the drop was recognised and imported, otherwise `false` if the drop was not recognised as a supported format (in which case Construct will continue running other handlers).

The `opts` parameter of `AddDragDropFileHandler()` is an options object, which may specify the following:

- `isZipFormat` : boolean indicating to handle dropped zip files only. If `true`, the callback will only be run if Construct recognises the dropped file as a zip file. Consequently the `file` parameter of the callback will be an `IZipFile`, from which the contents of the zip file can be read. If `false`, the callback will only be run if Construct does *not* recognise the dropped file as a zip file, and consequently the `file` parameter of the callback will be a `Blob`.
- `toLayoutView` : boolean indicating to handle files dropped to an open Layout View only. If `true`, the callback will only be run if a Layout View is open, and the `opts` parameter of the callback will contain information about the Layout View and the drop position. If `false`, the callback will be run regardless of whether a Layout View is open or not, and no further options will be provided to the callback.

The `callback` should have the signature `async function(filename, file, opts)`. The type of the `file` parameter is an `IZipFile` or `Blob` depending on the `isZipFormat` option. The `opts` parameter of the callback will provide the following additional details only when the `toLayoutView` option was specified:

- `layoutView` : an `ILayoutView` interface representing the Layout View that was open when the file was dropped. This also provides access to the associated project, layout, current active layer, and so on.

- `clientX` and `clientY` : the drop position within the window in client co-ordinates.
- `layoutX` and `layoutY` : the drop position within the Layout View in layout co-ordinates. This is the position to create any new instances relevant to.

static ShowLongTextPropertyDialog(text, caption)

Show the same dialog used to edit `longtext` properties. This is simply a large multi-line text field in a dialog, allowing for long text strings to be more conveniently edited, since the Properties Bar often can only show a small amount of text. Returns a promise that resolves with `null` if the dialog was cancelled, else a string of the text in the dialog.