Drawing Object Keyboard Shortcut Errors

This document is provided to give a working copy of a bug as reported in Python3docs Bugs #6. This bug is as follows:-

> The Jump-Boxes (which are LO TextBoxes, known as “Drawing Objects” in the Navigator) will not allow some keyboard shortcuts, although the menu WILL work. An example is that `menu:Edit | Paste Special | Paste Unformatted Text` DOES work, but `Shift+Ctrl+Alt+V` does NOT work.

To produce this document the following steps were taken:

(a) The styles from bug3.odt were loaded into this document under LO 24.8.3.2

(b) The table-in-frame *“Frame 3.3: Dictionary Methods Table”* was copied from chapter_03.odt into this document. You will notice that the Table has an attached *Jump-Box* near the top on the right-hand side of the margin.
Note: A Text-Box was used here because they can be given attractive rounded-corners, whereas Frames cannot be given such style.

(c) The Jump-Box was cleared of all text.

Purely for reference, the 1st 6 letters of the words in the top row of “Description” were given a Purple character border. Those 6 letters were then copied (`Ctrl-C`).

(d) Switching back to the TextBox, Those 6 letters were:

i. Pasted into the TextBox using the Menu (which is effective); then

ii. Pasted into the TextBox using the keyboard shortcut (ineffective).

Note: The purple text colours were added by myself afterwards, again purely for ease of reference.

Example (next page):

Table 6.1: Dictionary Methods

Syntax	*Description*
`d.clear()`	Remove s all items from `dict d`
`d.copy()`	Returns a shallow copy of `dict d`
`d.fromkeys(s, v)`	Returns a `dict` whose keys are the items in sequence `s` and whose values are `None` or `v` if `v` is given
`d.get(k)`	Returns `key k`'s associated value, or `None` if `k` isn't in `dict d`
`d.get(k, v)`	Returns `key k`'s associated value, or `v` if `k` isn't in `dict d`
`d.items()`	Returns a view[Error: Reference source not found] of all the `(key, value)` pairs in `dict d`
`d.keys()`	Returns a view[Error: Reference source not found] of all the keys in `dict d`
`d.pop(k)`	Returns `key k`'s associated value and removes the item whose `key` is `k`, or returns `v` if `k` isn't in `dict d`
`d.popitem()`	Returns and removes an arbitrary `(key, value)` pair from `dict d`, or raises a *KeyError* exception if `d` is `empty`
`d.setdefault(k, v)`	The same as the `dict.get()` method, except that if the `key` is not in `dict d`, a new item is inserted with the `key k`, and with a *value* of `None` or of `v` if `v` is given
`d.update(a)`	Adds every `(key, value)` pair from `a` that isn't in `dict d` to `d`, and for every *key* that is in both `d` and `a`, replaces the corresponding value in `d` with the one in `a`—`a` can be a *dictionary*, an *iterable* of `(key, value)` pairs, or *keyword* arguments
`d.values()`	Returns a view[Error: Reference source not found] of all the *values* in `dict d`

Menu:
Remove

Kbd:
V