

# Tastaturbedienung des Clients

AID 048 DE



© 2017 ADITO Software GmbH

Diese Unterlagen wurden mit größtmöglicher Sorgfalt hergestellt. Dennoch kann für Fehler in den Beschreibungen und Erklärungen keine Haftung übernommen werden. Wir sind für Feedback zu den Themen, Inhalten, aber auch noch vorhandenen Fehlern dankbar und würden uns freuen, Ihre Meinung zu hören. Die in diesen Unterlagen enthaltenen Daten und Angaben, einschließlich URLs und anderer Verweise können ohne vorherige Ankündigung geändert werden. Alle in diesen Unterlagen aufgeführten Produkt- und Firmennamen sind unter Umständen Marken oder geschützte Zeichen der einzelnen Firmen. Ohne ausdrückliche schriftliche Einverständniserklärung der ADITO Software GmbH darf kein Teil dieses Dokumentes vervielfältigt oder in einer Datenverarbeitungsanlage gespeichert oder in diese eingelesen werden. Diese Einschränkung gilt unabhängig von Art und Weise der Datenerfassung.

Autor: FA, MW, KN. Version 10.1. Zuletzt geändert 27.09.2017

Version	Änderungen
<b>10.1</b>	Anpassung der Formatierungen
<b>10.0</b>	Anpassung an ADITO4.6
<b>3.5</b>	Letzter Stand vor Übernahme in Versionierung

# Inhaltsverzeichnis

<b>1.</b>	<b>Zweck des Dokuments</b> .....	<b>4</b>
<b>2.</b>	<b>Mnemonics in Frames</b> .....	<b>5</b>
2.1.	Mnemonics in Frames anlegen .....	5
<b>3.</b>	<b>Verwenden eigener Tastatureingabe (Tastendefinition und Tastenaktion)</b> .....	<b>6</b>
3.1.	Tastendefinition / userKeystrokeAssignment .....	6
3.1.1.	Vorgenommene Aktion .....	6
3.1.2.	Ausführungstyp .....	6
3.2.	Tastenaktion / onUserKeystroke .....	7
3.3.	Beispiele .....	7
3.3.1.	Edit-Komponente mit mehreren Reaktionsmöglichkeiten .....	7
3.3.2.	Weiterspringen mit Pfeiltasten .....	8
3.3.3.	Weiterspringen in Tabellenkomponenten .....	9
3.4.	Liste der Tasten .....	10
<b>4.</b>	<b>Tastaturkürzel</b> .....	<b>13</b>

## 1. Zweck des Dokuments

Dieses Dokument beschreibt

- die Bedienung des Clients mit Hilfe der Tastatur,
- erklärt die Tastenkombinationen und die Bedienung der Menüs mit Mnemonics,
- beschreibt, wie Sie selbst auf Tastenaktionen reagieren können.

Die nachfolgenden Kapitel beschreiben die Tastaturbedienung des ADITO online Clients in der Standardversion. Abhängig von der kundenspezifischen Anpassung der Clients kann die Tastaturbedienung abweichen. Fest vorgegeben sind auf jeden Fall die Tastenkürzel des letzten Kapitels.

## 2. Mnemonics in Frames

Innerhalb von Bildschirmmasken (Frames) stehen Mnemonics zur Verfügung, sofern sie angelegt werden. So kann wie auch in den Menüs mit Hilfe der **Alt**-Taste auf das entsprechende Element zugegriffen werden, sofern ein Mnemonic dafür definiert wurde.

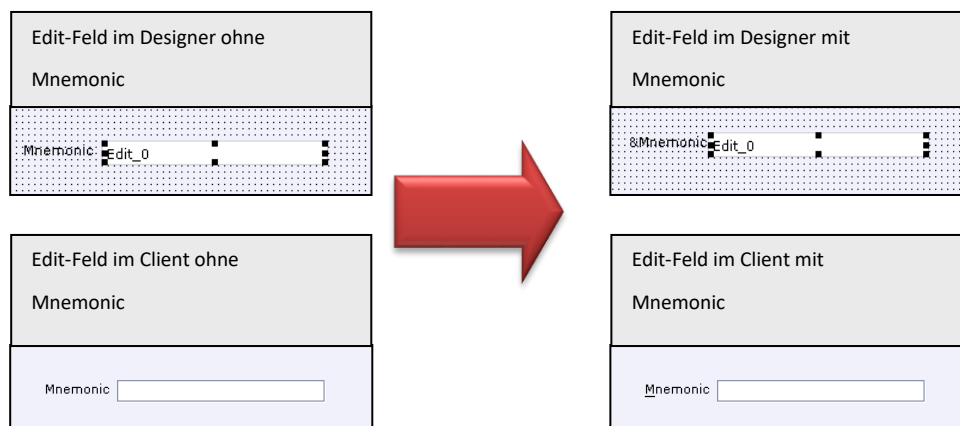
### 2.1. Mnemonics in Frames anlegen

Mithilfe des ADITO online Designers können für viele Elemente Mnemonics vergeben werden. Diese Elemente werden im Detail im ADITO online Designer-Handbuch beschrieben. Für folgende Elemente können Mnemonics angelegt werden:

- Combobox
- Edit-Feld
- Memo-Feld
- Register

Um Mnemonics anzulegen, muss vor den gewünschten Buchstaben ein „&“-Zeichen gesetzt werden. Abhängig von der Komponente geschieht dies an verschiedenen Stellen.

Bei **Comboboxen**, **Edit-** und **Memo-Feldern** erfolgt die Definition im Feld `Label` im Reiter `Eigenschaften` des Komponenten-Inspektors.



Das in dieser Abbildung gezeigte Edit-Feld kann im entsprechenden Frame dann mit der Tastenkombination **Alt** + **M** aufgerufen werden.

ADITO online 3.1: Bei **Registern** werden die Mnemonics im Feld `Tab` des Reiters `Register` des Komponenten-Inspektors gesetzt.

ADITO4: Bei **Registern** werden die Mnemonics in der Eigenschaft `Caption` des RegisterTabs gesetzt.

Ab ADITO4 kann auch für **Schaltflächen im Ribbon** ein Mnemonic gesetzt werden. Dieses wird analog dem Register gesetzt.

Sollten mehrere Mnemonics die gleichen Tastaturkürzel verwenden, so werden diese Mnemonics nicht ausgewertet.

## 3. Verwenden eigener Tastatureingabe (Tastendefinition und Tastenaktion)

Sie können Tastenaktionen in Komponenten definieren, um auf bestimmte Tastatureingaben des Benutzers zu reagieren. So können Sie beispielsweise in einer editierbaren Tabelle auf Tastendruck reagieren, um die Zeile oder die Spalte zu wechseln.

Tastendefinition und Tastenaktion können Sie in folgenden Komponenten verwenden:

- **Edit-Komponente**
- **Combobox**
- **Checkbox**
- **Listenkomponente**

### 3.1. Tastendefinition / userKeystrokeAssignment

Um überhaupt mit eigenen Tastatureingaben arbeiten zu können, müssen zuerst die möglichen Tasten festgelegt werden, auf die reagiert werden kann. Diese Tasten legen Sie im Prozess `Tastendefinition / userKeystrokeAssignment` fest.

In diesem Prozess geben Sie ein Array zurück, das nach folgendem Muster aufgebaut ist:

```
result.object ([  
  ["typed q", "EXECUTE_PROCESS_SHOW"],  
]);
```

#### 3.1.1. Vorgenommene Aktion

Im ersten Element des Sub-Arrays geben Sie die Taste und den Ausführungstypen an. Es stehen Ihnen folgende Aktionen zur Verfügung:

- **typed:** Für alle Buchstaben- und Zahlentasten (a-z; A-Z; 0-9). Diese Aktion wird immer aufgerufen, wenn eine Taste „getippt“ wurde. Die typed-Aktion kann sich auch wiederholen, wenn z.B. länger eine Buchstaben-Taste gedrückt wird, abhängig von den Tastatureinstellungen.
- **pressed:** Für alle Steuerungstasten (Shift, Alt, etc.): Taste gedrückt.
- **released:** Für alle Steuerungstasten (Shift, Alt, etc.): Taste gelöst. „Released“ funktioniert nur bei Tasten, die auch „pressed“ unterstützen.

Diese Tasten lassen sich auch kombinieren, z.B. „Alt Shift Released“.

#### 3.1.2. Ausführungstyp

Wie auf den Prozess reagiert wird, legen Sie im zweiten Element fest. Hier stehen Ihnen die folgenden Ausführungstypen zur Verfügung:

- **EXECUTE\_PROCESS\_SHOW:** führt den Prozess aus und zeigt die gedrückte Taste weiterhin in der entsprechenden Komponente an.

- **EXECUTE\_PROCESS**: führt den Prozess aus, zeigt die gedrückte Taste aber nicht mehr in der entsprechenden Komponente an (der Tastendruck wird also nicht mehr geschrieben).
- **IGNORE**: Drücken dieser Taste wird vollkommen ignoriert.

## 3.2. Tastenaktion / onUserKeystroke

In der Tastenaktion kann nun auf die definierten Tasten reagiert werden. Alle Tastenanschläge, die in Tastendefinition festgelegt wurden, werden in diesem Prozess aufgefangen und können weiterverarbeitet werden. Die Tastenaktionen werden in lokalen Variablen vorgegeben:

- **\$local.keycode** (Tastendruck: Tastencode)
- **\$local.keychar** (Tastendruck: Charakterdarstellung der Taste)
- **\$local.keymodifiers** (Tastendruck: Modifizierer (Strg, Alt, ...))
- **\$local.keystroke** (Tastendruck: Repräsentation als String)

## 3.3. Beispiele

### 3.3.1. Edit-Komponente mit mehreren Reaktionsmöglichkeiten

Prozess für Tastendefinition:

```
result.object([
["alt pressed M", "EXECUTE_PROCESS"], // ALT + M
["typed q", "EXECUTE_PROCESS_SHOW"], // q
["typed w", "EXECUTE_PROCESS"], // w
["pressed LEFT", "EXECUTE_PROCESS"], // <- gedrückt
["released LEFT", "EXECUTE_PROCESS"], // -> losgelassen
["typed a", "IGNORE"], // a
["alt shift released RIGHT", "EXECUTE_PROCESS"], // ALT + SHIFT + -> losgelassen
["pressed HOME", "EXECUTE_PROCESS"] // Pos1
]);
```

Prozess für Tastenaktion:

```
var keycode = vars.getString("$local.keycode");
var keychar = vars.getString("$local.keychar");
var keymodifiers = vars.getString("$local.keymodifiers");
var keystroke = vars.getString("$local.keystroke");

logging.log("keycode: " + keycode + " - keychar: " + keychar + " -
keymodifiers: " + keymodifiers + " - keystroke: " + keystroke);
```

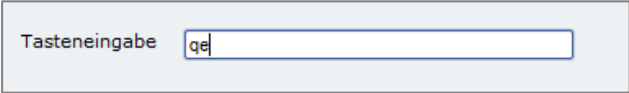
Nun werden folgende Tasten in dieser Reihenfolge gedrückt:

q, w, e, ←, a, Alt Shift →, Home

Und Folgendes wird geloggt:

```
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] keycode: 0 - keychar: q
- keymodifiers: 0 - keystroke: typed q
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] keycode: 0 - keychar: w
- keymodifiers: 0 - keystroke: typed w
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] keycode: 37 - keychar: ?
- keymodifiers: 0 - keystroke: pressed LEFT
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] keycode: 37 - keychar: ?
- keymodifiers: 0 - keystroke: released LEFT
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] keycode: 0 - keychar: a
- keymodifiers: 0 - keystroke: typed a
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] keycode: 39 - keychar: ?
- keymodifiers: 585 - keystroke: shift alt released
RIGHT
R-37-R-0000-S Projekt-Ausgabe. [ID 1] [->] keycode: 36 - keychar: ?
- keymodifiers: 0 - keystroke: pressed HOME
```

Die Anzeige hingegen zeigt nur die Buchstaben q und e, da nur bei q ein Prozess mit SHOW angegeben wurde, e hingegen überhaupt nicht als Tastendefinition bekannt ist:



### 3.3.2. Weiterspringen mit Pfeiltasten

Möchten Sie mit Hilfe einer Pfeiltaste von einer Komponente in die nächste springen, so können Sie das mit Hilfe dieses Beispiels tun. In diesem Beispiel befinden wir uns in der Komponente `edt_1` und möchten durch Drücken der Pfeiltaste rechts auf `edt_2` springen.

Tastendefinition:

```
result.object([
  ["pressed RIGHT", "EXECUTE_PROCESS"],
]);
```

Tastenaktion:

```
var keystroke = vars.getString("$local.keystroke");
// nach rechts
if(keystroke == "pressed RIGHT")
{
```



```
swing.setFocus("$comp.edt_2");
swing.refresh("$comp.edt_2");
}
```



Werten Sie in der übergebenen Komponente (hier `edt_2`) verschiedene Prozesse aus, z.B. eine Hintergrundfarbe auf Basis einer Image-Variablen, so müssen Sie unbedingt den `swing.refresh`-Befehl setzen. Ein `swing.setFocus` aus der Tastenaktion heraus führt nicht zu einer Neuberechnung der angesprochenen Komponente!

### 3.3.3. Weiterspringen in Tabellenkomponenten

Auch in einer editierbaren Tabelle können Sie per Tastenaktion von Komponente zu Komponente springen. Sie greifen für eine editierbare Tabelle ohnehin auf Komponenten zurück, welche unsichtbar auf dem Frame liegen. Für diese Komponenten legen Sie die Tastenaktionen an.

Kontonummer	IBAN	Firma
?	?	?
?	?	?
?	?	?
?	?	?

edt\_ktonr    edt\_iban    lup\_org

In diesem Fall können Sie für die Komponenten `edt_ktonr` und `edt_iban` entsprechend die Tastendefinition und Tastenaktion setzen. Das folgende Beispiel zeigt die Einstellung von `edt_ktonr`.



Beachten Sie bei der Definition von Tastenaktionen in Tabellen, dass Sie nur in denjenigen Komponenten Tastenaktionen definieren können, in denen diese zur Verfügung stehen. In diesem Beispiel kann z.B. bei der Lookup-Komponente keine Tastenaktion angegeben werden.

Tastendefinition:

```
result.string([
["pressed ENTER", "EXECUTE_PROCESS_SHOW"],
]);
```

Tastenaktion:

```
var keystroke = vars.getString("$local.keystroke");
```

```
// nach rechts
if(keystroke == "pressed ENTER")
{
  logging.log("ENTER gedrückt");
  swing.setTableSelectionRelative("$comp.tbl_0", 0, 1, 1);
}
```

Zum Weiterspringen in eine weitere Komponente wird in diesem Fall kein `swing.setFocus`-Befehl ausgeführt, sondern der JDito-Befehl `swing.setTableSelectionRelative`. Dieser hat folgenden Aufbau:

1. Parameter: Angabe der Tabellenkomponente, in der die Tasteneingabe gesetzt werden soll.
2. Parameter: Angabe der Zeile, relativ von der aktuell fokussierten Zeile aus gesehen, in welcher der Tastenfokus gesetzt werden soll. Möchten Sie in der aktuellen Zeile bleiben, so belassen Sie den Wert 0.
3. Parameter: Angabe der Spalte, relativ von der aktuell fokussierten Zeile aus gesehen. Die Logik ist identisch mit der unter 2 Beschriebenen.
4. Parameter: Vertikale Anpassung. 0 bedeutet, dass die Anzeige in der obersten Zeile erfolgen soll (wenn es eine sehr große Tabelle ist); bei 1 erfolgt Sie in der untersten Zeile. Alle anderen Werte dazwischen. Ungültige Werte werden ignoriert.



Der Prozess zum Setzen der Tabellenselektion wird nur ausgeführt, wenn `EXECUTE_PROCESS_SHOW` angegeben wurde!

### 3.4. Liste der Tasten

ENTER	BACK_SLASH	DEAD_DIAERESIS
BACK_SPACE	CLOSE_BRACKET	DEAD_ABOVEING
TAB	NUMPADO	DEAD_DOUBLEACUTE
CANCEL	NUMPAD1	DEAD_CARON
CLEAR	NUMPAD2	DEAD_CEDILLA
SHIFT	NUMPAD3	DEAD_OGONEK
CONTROL	NUMPAD4	DEAD_IOTA
ALT	NUMPAD5	DEAD_VOICED_SOUND
PAUSE	NUMPAD6	DEAD_SEMIVOICED_SOUND
CAPS_LOCK	NUMPAD7	AMPERSAND
ESCAPE	NUMPAD8	ASTERISK

SPACE	NUMPAD9	QUOTEDBL
PAGE_UP	MULTIPLY	LESS
PAGE_DOWN	ADD	GREATER
END	SEPARATER	BRACELEFT
HOME	SEPARATOR	BRACERIGHT
LEFT	SUBTRACT	AT
UP	DDECIMAL	COLON
RIGHT	DIVIDE	CIRCUMFLEX
DOWN	DELETE	DOLLAR
COMMA	NUM_LOCK	EURO_SIGN
MINUS	SCROLL_LOCK	EXCLAMATION_MARK
PERIOD	F1	INVERTED_EXCLAMATION_MARK
SLASH	F2	LEFT_PARENTHESIS
0	F3	NUMBER_SIGH
1	F4	PLUS
2	F5	RIGHT_PARENTHESIS
3	F6	UNDERSCORE
4	F7	WINDOWS
5	F8	CONTEXT_MENU
6	F9	FINAL
7	F10	CONVERT
8	F11	NONCONVERT
9	F12	ACCEPT
SEMICOLON	F13	MODECHANGE
EQUALS	F14	KANA
A	F15	KANJI
B	F16	ALPHANUMERIC
C	F17	KATAKANA
D	F18	HIRAGANA
E	F19	FULL_WIDTH
F	F20	HALF_WIDTH
G	F21	ROMAN_CHARACTERS

H	F22	ALL_CADIDATES
I	F23	PREVIOUS_CANDIDATE
J	F24	CODE_INPUT
K	PRINTSCREEN	JAPANESE_KATAKANA
L	INSERT	JAPANESE_HIRAGANA
M	HELP	JAPANESE_ROMAN
N	META	KANA_LOCK
O	BACK_QUOTE	INPUT_METHOD_ON_OFF
P	QUOTE	CUT
Q	KP_UP	COPY
R	KP_DOWN	PASTE
S	KP_LEFT	UNDO
T	KP_RIGHT	AGAIN
U	DEAD_GRAVE	FIND
V	DEAD_ACUTE	PROPS
W	DEAD_CIRCUMFLEX	STOP
X	DEAD_TILDE	COMPOSE
Y	DEAD_MACRON	ALT_GRAPH
Z	DEAD_BREVE	BEGIN
OPEN_BRACKET	DEAD_ABOVEDOT	UNDEFINED

## 4. Tastaturkürzel

Folgende Tastenkürzel sind in ADITO online fest vorgegeben:

<b>F1</b>	Hilfe (allgemein)	<b>Shift + F5</b>	Suche verfeinern (Maskensuche)
<b>F2</b>	Neuanlage	<b>STRG + F5</b>	Suche erweitern (Maskensuche)
<b>F3</b>	Editieren	<b>STRG + F4</b>	Aktiven Frame oder Report schließen
<b>F4</b>	Löschen	<b>STRG + Enter</b>	Sidebar aktualisieren
<b>F5</b>	Suchmodus: Wechsel in Anzeigemodus Anzeigemodus: Wechsel in Suchmodus	<b>STRG + Tab</b>	Zwischen geöffneten Frames / Reports wechseln
<b>F6</b>	Maskenansicht: Wechsel in Tabellenansicht Tabellenansicht: Wechsel in Maskenansicht	<b>STRG + S</b>	Layout speichern
<b>F7</b>	Abbrechen	<b>STRG + P</b>	Bildschirmfoto drucken (Frames) / Bericht drucken (Reports)
<b>F8</b>	Speichern	<b>STRG + T</b>	Client in Taskleiste ablegen
<b>F9</b>	Aktualisieren	<b>Alt + oben</b>	Erster Datensatz
<b>F10</b>	Zurücksetzen der Maskensuche	<b>Alt + unten</b>	Letzter Datensatz
<b>STRG + +</b>	Weiteres Suchkriterium hinzufügen (Tabellensuche)	<b>Alt + links</b>	Vorheriger Datensatz
<b>STRG + -</b>	Suchkriterium entfernen (Tabellensuche)	<b>Alt + rechts</b>	Nächster Datensatz
<b>STRG + Q</b>	Indexsuche aufrufen	<b>Alt + F4</b>	Client beenden
<b>STRG + A</b>	Markieren aller Daten. Vorsicht, ist die Tabelle nicht vollständig geladen,	<b>Alt + F5</b>	Letzte Suchaktion

werden die nicht  
geladenen Zeilen als „?“  
übernommen.