

## 2.192 UIAlphaEntry - User Alpha Entry

## Usage

UIAlphaEntry (*User Interaction Alpha Entry*) is used to let the operator enter a string from the available user device, such as the FlexPendant. A message is written to the operator, who answers with a text string. The string is then transferred back to the program.

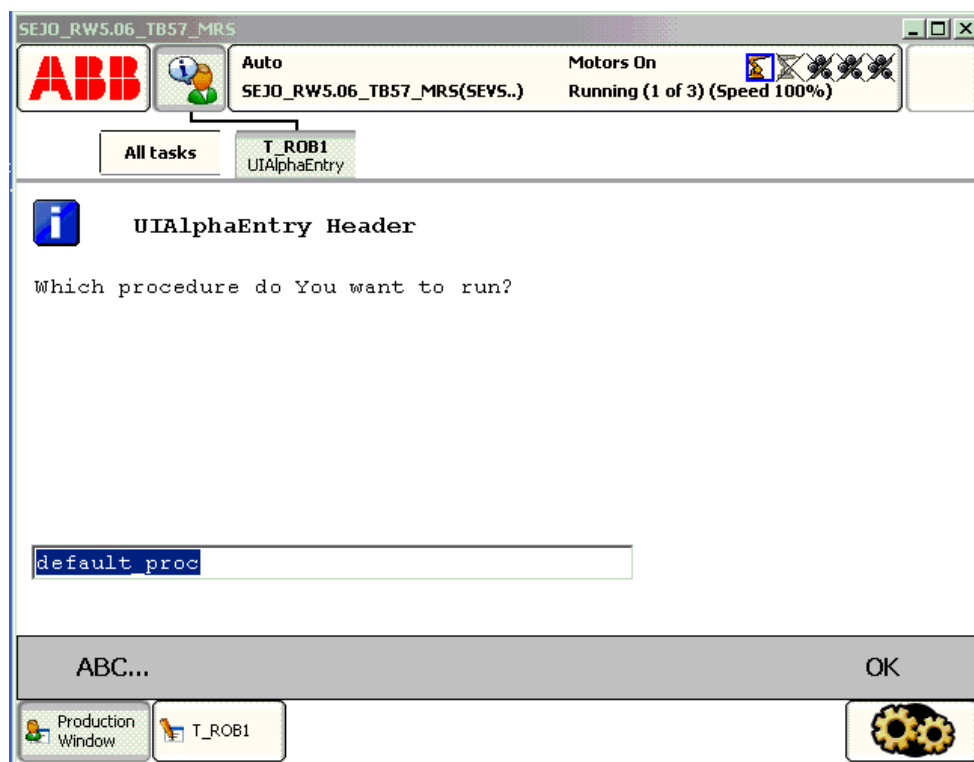
## Basic examples

The following example illustrates the function UIAlphaEntry.

See [More examples on page 1428](#).

## Example 1

```
VAR string answer;
...
answer := UIAlphaEntry(
  \Header:= "UIAlphaEntry Header",
  \Message:= "Which procedure do You want to run?"
  \Icon:=iconInfo
  \InitString:= "default_proc");
%answer%;
```



xx0500002437

Above alpha message box with icon, header, message, and init string are written on the FlexPendant display. The user edit init string or write a new string with the supported Alpha Pad. Program execution waits until OK is pressed and then the

*Continues on next page*

## 2 Functions

---

### 2.192 UIAlphaEntry - User Alpha Entry

RobotWare-OS

Continued

written string is returned in the variable `answer`. The program then calls the specified procedure with late binding.

---

#### Return value

**Data type:** `string`

This function returns the input string.

If function breaks via `\BreakFlag`:

- If parameter `\InitString` is specified, this string is returned
- If parameter `\InitString` is not specified, empty string "" is returned.

If function breaks via `ERROR` handler, no return value will be returned at all.

---

#### Arguments

```
UIAlphaEntry ([\Header] [\Message][\MsgArray]
              [\Wrap][\Icon][\InitString] [\MaxTime] [\DIBreak] [\DIPassive]
              [\DOBreak] [\DOPassive] [\BreakFlag] [\UIActiveSignal])
```

`[\Header]`

**Data type:** `string`

Header text to be written at the top of the message box. Max. 40 characters.

`[\Message]`

**Data type:** `string`

One text line to be written on the display. Max 55 characters.

`[\MsgArray]`

**Message Array**

**Data type:** `string`

Several text lines from an array to be written on the display.

Only one of parameter `\Message` or `\MsgArray` can be used at the same time.

Max. layout space is 9 lines with 55 characters.

`[\Wrap]`

**Data type:** `switch`

If selected, all the specified strings in the argument `\MsgArray` will be concatenated to one string with single space between each individual strings and spread out on as few lines as possible.

Default, each string in the argument `\MsgArray` will be on separate line on the display.

`[\Icon]`

**Data type:** `icondata`

Defines the icon to be displayed. Only one of the predefined icons of type `icondata` can be used. See [Predefined data on page 1428](#).

Default no icon.

`[\InitString]`

**Data type:** `string`

*Continues on next page*

---

An initial string to be display in the text entry box as default.

[ \MaxTime ]

**Data type:** num

The maximum amount of time in seconds that program execution waits. If the OK button is not pressed within this time, the program continues to execute in the error handler unless the `BreakFlag` is used (see below). The constant `ERR_TP_MAXTIME` can be used to test whether or not the maximum time has elapsed.

[ \DIBreak ]

**Digital Input Break**

**Data type:** signaldi

The digital input signal that may interrupt the operator dialog. If the OK button is not pressed before the signal is set to 1 (or is already 1), the program continues to execute in the error handler, unless the `BreakFlag` is used (see below). The constant `ERR_TP_DIBREAK` can be used to test whether or not this has occurred.

[ \DIPassive ]

**Digital Input Passive**

**Data type:** switch

This switch overrides the default behavior when using `DIBreak` optional argument. Instead of reacting when signal is set to 1 (or already 1), the instruction should continue in the error handler (if no `BreakFlag` is used) when the signal `DIBreak` is set to 0 (or already is 0). The constant `ERR_TP_DIBREAK` can be used to test whether or not this has occurred.

[ \DOBreak ]

**Digital Output Break**

**Data type:** signaldo

The digital output signal that may interrupt the operator dialog. If the OK button is not pressed before the signal is set to 1 (or is already 1), the program continues to execute in the error handler, unless the `BreakFlag` is used (see below). The constant `ERR_TP_DOBREAK` can be used to test whether or not this has occurred.

[ \DOPassive ]

**Digital Output Passive**

**Data type:** switch

This switch overrides the default behavior when using `DOBreak` optional argument. Instead of reacting when signal is set to 1 (or already 1), the instruction should continue in the error handler (if no `BreakFlag` is used) when the signal `DOBreak` is set to 0 (or already is 0). The constant `ERR_TP_DOBREAK` can be used to test whether or not this has occurred.

[ \BreakFlag ]

**Data type:** errnum

*Continues on next page*

## 2 Functions

---

### 2.192 UIAlphaEntry - User Alpha Entry

RobotWare-OS

Continued

A variable (before used set to 0 by the system) that will hold the error code if `\MaxTime`, `\DIBreak` or `\DOBreak` is used. The constants `ERR_TP_MAXTIME`, `ERR_TP_DIBREAK` and `ERR_TP DOBREAK` can be used to select the reason. If this optional variable is omitted, the error handler will be executed.

`[\UIActiveSignal]`

**Data type:** `signaldo`

The digital output signal used in optional argument `UIActiveSignal` is set to 1 when the message box is activated on the FlexPendant. When the user selection has been done and the execution continue, the signal is set to 0 again.

No supervision of stop or restart exist. The signal is set to 0 when the function is ready, or when PP is moved.

---

### Program execution

The alpha message box with alpha pad, icon, header, message lines, and init string are displayed according to the programmed arguments. Program execution waits until the user edits or creates a new string and presses OK, or the message box is interrupted by time-out or signal action. The input string and interrupt reason are transferred back to the program.

New message box on TRAP level takes focus from message box on basic level.

---

### Predefined data

```
!Icons:
  CONST icondata iconNone := 0;
  CONST icondata iconInfo := 1;
  CONST icondata iconWarning := 2;
  CONST icondata iconError := 3;
```

---

### More examples

The following example illustrates the function `UIAlphaEntry`.

#### Example 1

```
VAR errnum err_var;
VAR string answer;
VAR string logfile;
...
answer := UIAlphaEntry (\Header:= "Log file name:"
  \Message:= "Enter the name of the log file to create?"
  \Icon:=iconInfo
  \InitString:= "signal.log"
  \MaxTime:=60
  \DIBreak:=di5\BreakFlag:=err_var);
TEST err_var
CASE ERR_TP_MAXTIME:
CASE ERR_TP_DIBREAK:
  ! No operator answer
  logfile:="signal.log";
CASE 0:
  ! Operator answer
```

*Continues on next page*

```

        logfile := answer;
    DEFAULT:
        ! No such case defined
    ENDTEST

```

The message box is displayed and the operator can enter a string and press OK. The message box can also be interrupted with time out or break by digital input signal. In the program it is possible to find out the reason and take the appropriate action.

## Error handling

The following recoverable errors are generated and can be handled in an error handler. The system variable `ERRNO` will be set to:

Name	Cause of error
<code>ERR_NO_ALIASIO_DEF</code>	The signal variable is a variable declared in RAPID and it has not been connected to an I/O signal defined in the I/O configuration with instruction <code>AliasIO</code> .
<code>ERR_TP_NO_CLIENT</code>	There is no client, for example, a FlexPendant, to take care of the instruction.

If parameter `\BreakFlag` is not used, these situations can then be dealt with by the error handler:

If there is a time-out (parameter `\MaxTime`) before an input from the operator, the system variable `ERRNO` is set to `ERR_TP_MAXTIME` and the execution continues in the error handler.

If digital input is set (parameter `\DIBreak`) before an input from the operator, the system variable `ERRNO` is set to `ERR_TP_DIBREAK` and the execution continues in the error handler.

If a digital output is set (parameter `\DOBBreak`) before an input from the operator, the system variable `ERRNO` is set to `ERR_TP_DOBREAK` and the execution continues in the error handler.

## Limitations

Avoid using too small a value for the time-out parameter `\MaxTime` when `UIAlphaEntry` is frequently executed, for example in a loop. It can result in an unpredictable behavior of the system performance, like slow response of the FlexPendant.

## Syntax

```

UIAlphaEntry '('
    ['\ Header :=' <expression (IN) of string>]
    ['\ Message :=' <expression (IN) of string>]
    | ['\ MsgArray :=' <array {*} (IN) of string>]
    ['\ Wrap]
    ['\ Icon :=' <expression (IN) of icondata>]
    ['\ InitString :=' <expression (IN) of string>]
    ['\ MaxTime :=' <expression (IN) of num>]
    ['\ DIBreak :=' <variable (VAR) of signaldi>]
    ['\ DIPassive]

```

*Continues on next page*

## 2 Functions

---

### 2.192 UIAlphaEntry - User Alpha Entry

RobotWare-OS

Continued

```
['\' DOBreak ':'<variable (VAR) of signaldo>]  
['\' DOPassive]  
['\' BreakFlag ':'<var or pers (INOUT) of errnum>]  
['\' UIActiveSignal ':'<variable (VAR) of signaldo>'] ''
```

A function with return value of the data type string.

---

#### Related information

For information about	See
Icon display data	<a href="#">icondata - Icon display data on page 1554</a>
User interaction message box type basic	<a href="#">UIMsgBox - User Message Dialog Box type basic on page 922</a>
User interaction message box type advanced	<a href="#">UIMessageBox - User Message Box type advanced on page 1453</a>
User interaction number entry	<a href="#">UINumEntry - User Number Entry on page 1460</a>
User interaction number tune	<a href="#">UINumTune - User Number Tune on page 1466</a>
User interaction list view	<a href="#">UIListView - User List View on page 1445</a>
System connected to FlexPendant etc.	<a href="#">UIClientExist - Exist User Client on page 1431</a>
Procedure call with late binding	<a href="#">Technical reference manual - RAPID overview</a>
Clean up the operator window	<a href="#">TPErase - Erases text printed on the FlexPendant on page 808</a>