

# Value input

This category includes operators responsible for inputting a value:

- [Value input \(INPUT\)](#)
- [Value request \(REQUEST\)](#)

In addition to the above operators, data input is also performed via the *dialog forms* of message display ([ASK](#)) and [form opening \(DIALOG\)](#) operators in an interactive view.

## Cancellation and input result

In value input operators, an input operation can be *canceled* (for each operator, this situation is determined in its own way). If it happens, a **TRUE** value is written to the **System.requestCanceled** property (otherwise it will be **NULL**).

For all value input operators, the platform allows to specify an action (let's call it *main action*) that will be executed only if the input operation is successful (i.e. not canceled). The input result (if any) is passed to this action as a parameter. You can specify an *alternative* action for all value input operators that will be executed if an operation is canceled.

## Initial values and automatic change

Value input often assumes the presence of a certain *initial* (previous) value that the input starts from (that is, the initial value is assigned to the current one) and that the user can subsequently change. Like most other values, the initial value is defined as a property.

Also, in many cases (especially in property [change event](#) handlers), it is sometimes necessary to not just input value, but also to [write](#) this value to a certain property (as a rule, the one for which the change event handler is invoked. This is necessary to ensure WYSIWIG). In most cases, the changed property equals the initial value property. To make the implementation of such a scenario more convenient, you can specify a special option (**CHANGE**) in value input operators. This option will automatically add property change to the end of the main action. The changed property, in this case, will be the initial value property, and changed value — the input result (however, if necessary, the developer can specify changed property explicitly). It is worth noting that the described feature is nothing more than syntactic sugar, however, it allows to make the code a lot more concise and readable.

In the current platform implementation, initial values and automatic changes are supported only for primitive input operators (**INPUT**) and dialog form opening (**DIALOG**).