

Managing the Beas Script Assistant

How to use the Beas Script Assistant with the Beas Usability Extension

TABLE OF CONTENTS

1. Introduction	3
2. Opening the Script Assistant with the Beas Usability Extension.....	3
3. Script Assistant Field guide.....	5
3.1. Commands of the Script Assistant.....	5
3.2. The fields of each command.....	5
3.3. Saving the created script.....	16

1. INTRODUCTION

This document describes the procedure of using the *Beas Script Assistant* in the Universal Function with Beas Usability Extension.

The following prerequisites must be fulfilled to use the Universal Function:

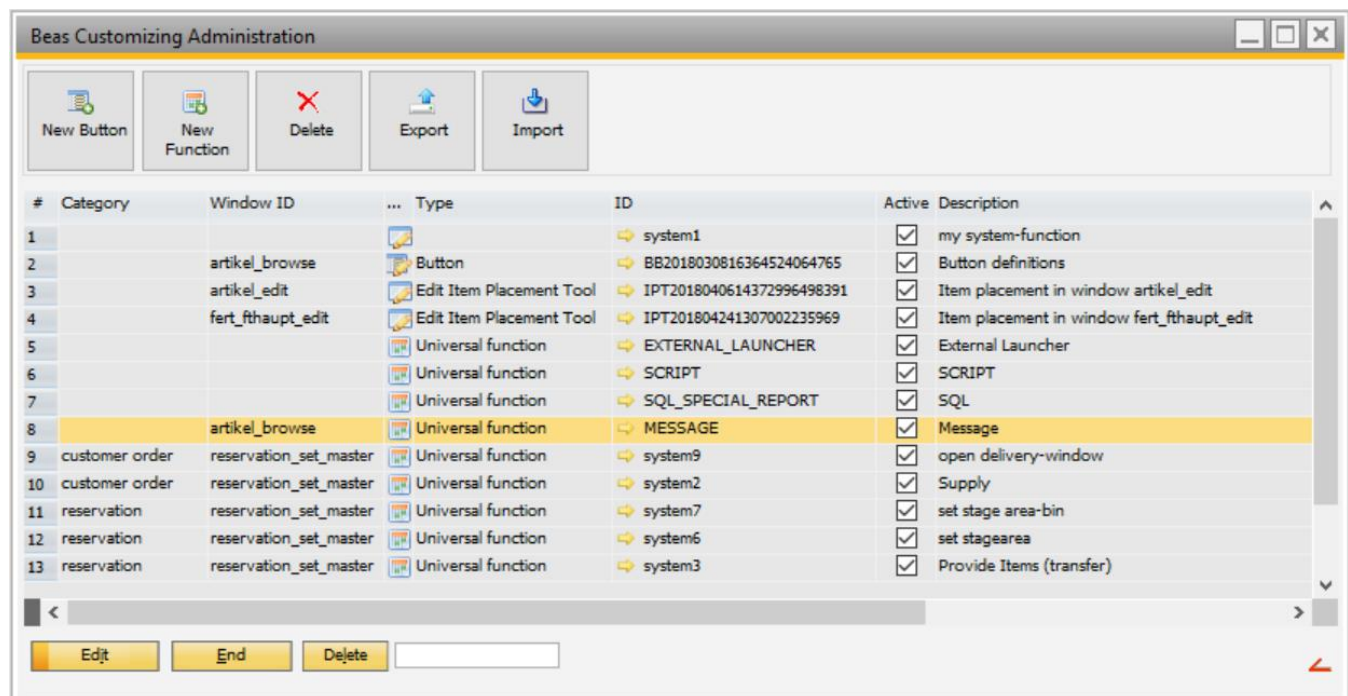
- B1UP version 2018.05 or later **must** be installed
- Beas version 9.3 PL00 or later **must** be installed

2. OPENING THE SCRIPT ASSISTANT WITH THE BEAS USABILITY EXTENSION

1. Go to the Beas Usability Extension add-on:

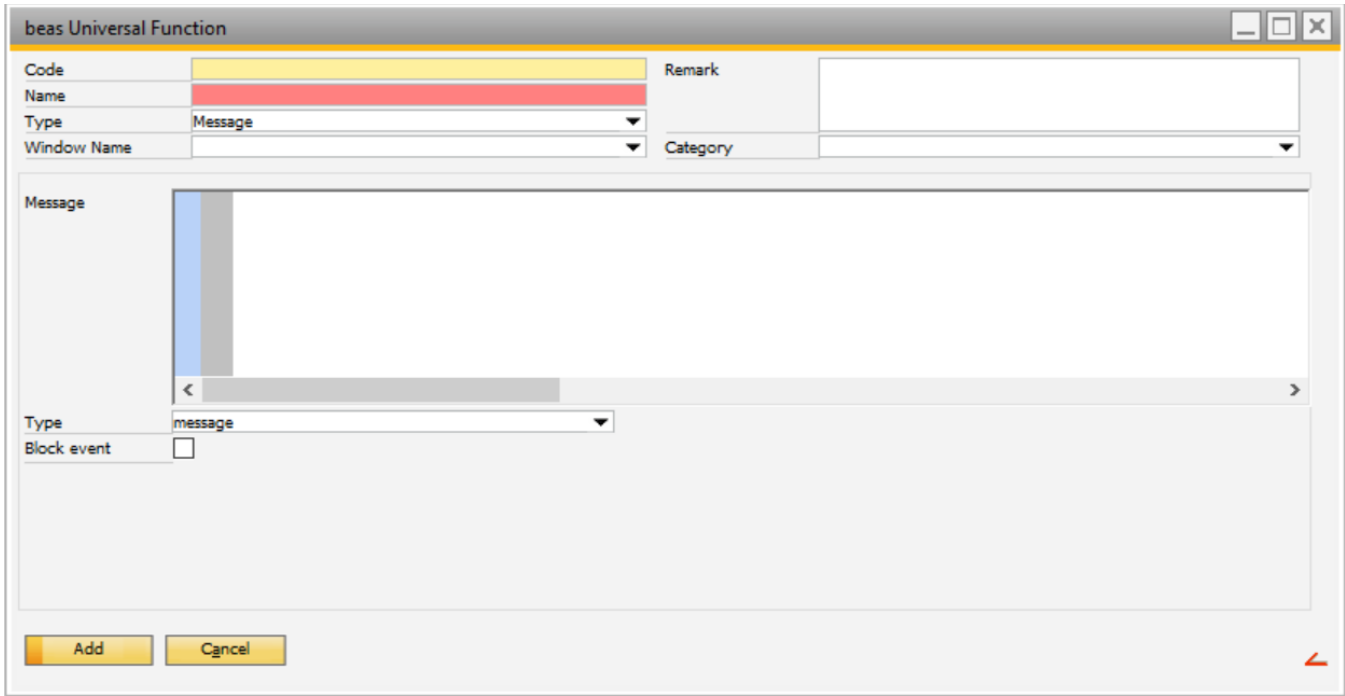
PATH to Universal Functions that apply to any window: Administration > Add-on > B1 Usability Package > Module Configuration > Beas Usability Extension

The Script Assistant is available wherever Beas script can be used.



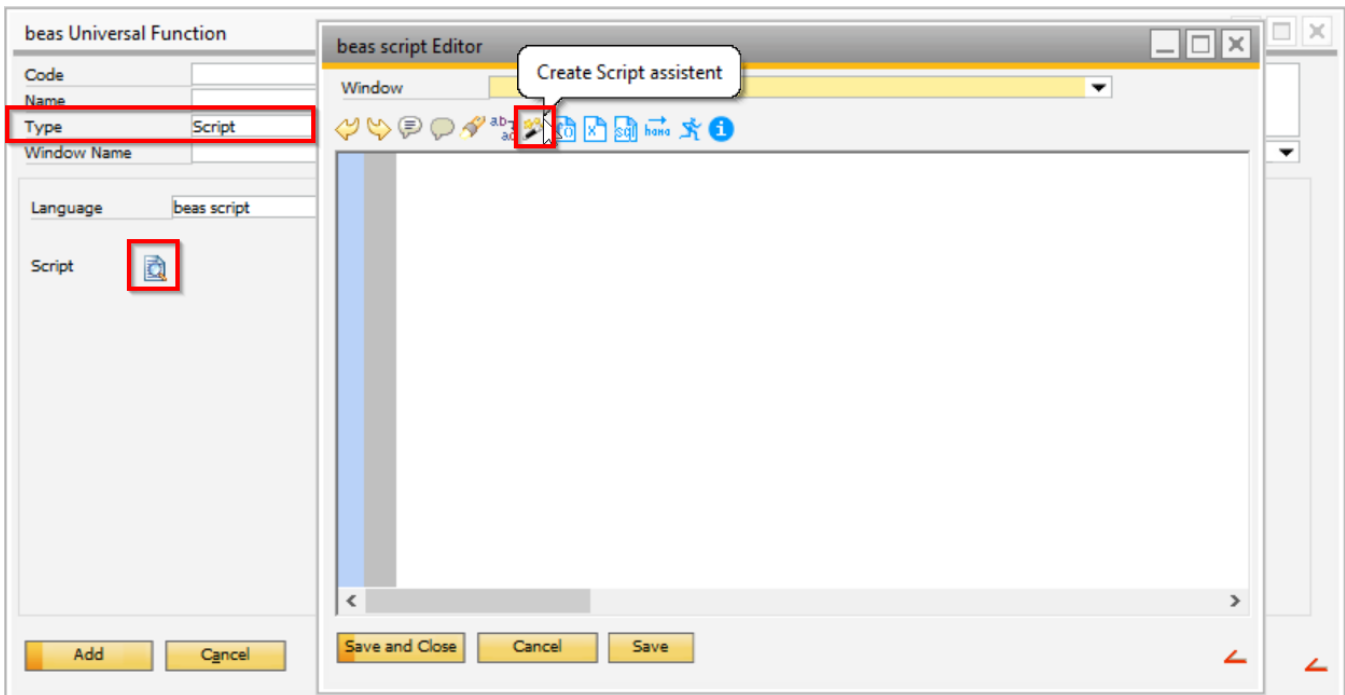
2. To create a new function, click the New Function button. A new screen is displayed



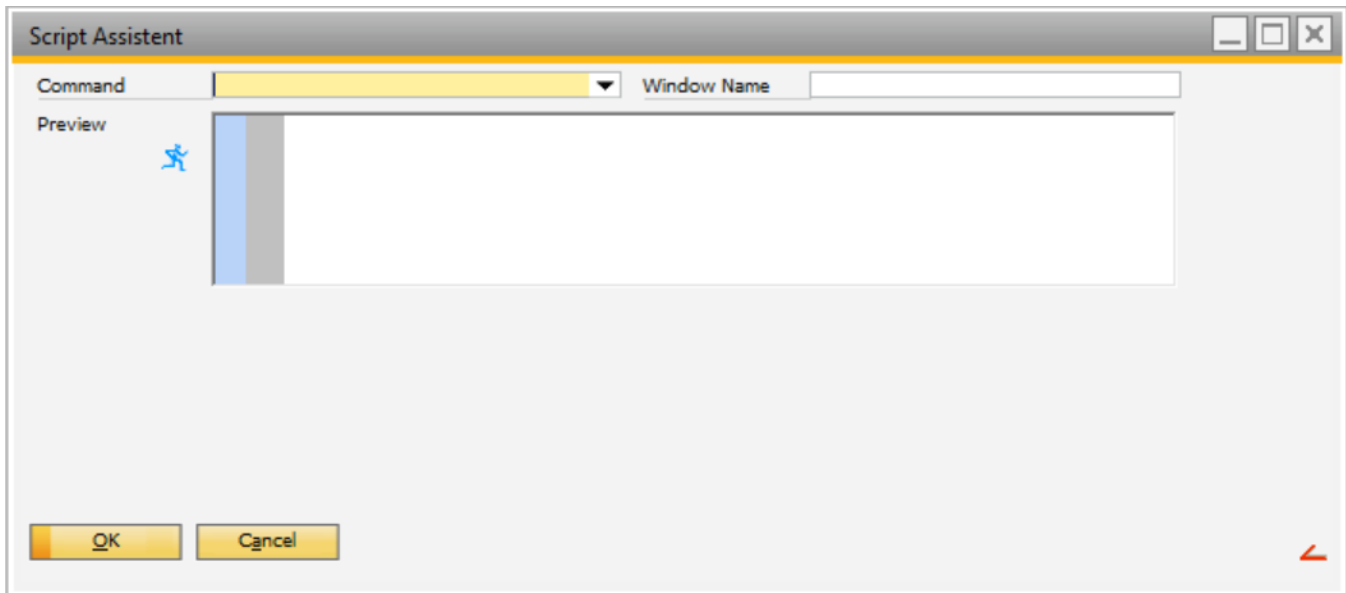


IMPORTANT: To access the Beas Script Assistant, select Script as the the *Type* of function and click the magnifying glass icon. The universal functions can also be accessed directly from the *Function Button* screen.

The *Beas Script Editor* is displayed.

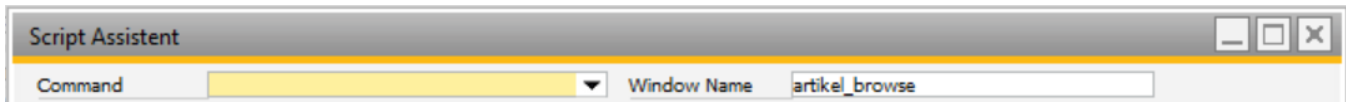


3. Click the Script Assistant icon and a new screen is displayed.



3. SCRIPT ASSISTANT FIELD GUIDE

3.1. Commands of the Script Assistant



Command: Choose the command to start using Beas script. Depending on the command selected, more options appear on the screen. Eight options are available:

- *Open Window* – to open another Beas screen
- *Open own and special windows* – to open own screens
- *Message* – to display a message
- *Dropdown* – to add a dropdown lists to an existing field in a Beas screen
- *If...then...end if* – to add an if statement command
- *Transfer Value* – to transfer a value from one field / screen to another
- *For .. next* – to loop through data lines
- *Work Order Function* – to automate work order operations

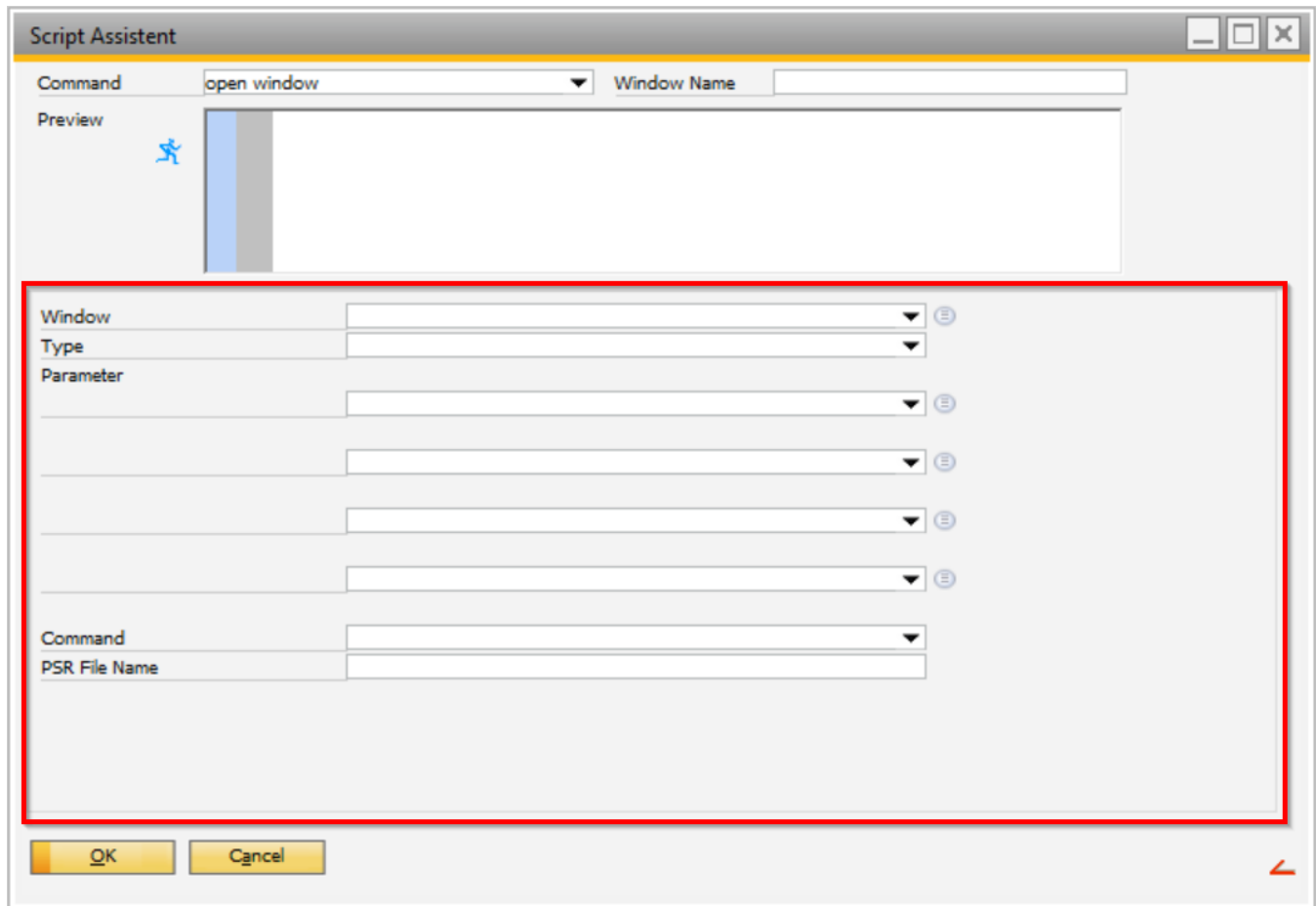
Window Name: Specify the window where the script is created. This field is filled automatically when creating the Universal Function with a right click in a Beas screen.

3.2. The fields of each command

1.1.1. The Open Window command fields

The bottom part of the screen changes depending on the selected *Command*.

For more information about open window command, please refer to the [online help](#) documentation.



Window: Opens another Beas screen.

Type: Sets the type of action by Beas. The options available are:

- *Open child without update* – Opens a new screen without saving changes in the parent screen. This is used when the user changes information between the child and parent screen but does not want to use the *Response* function. For more information about the *Response* function, please refer to the [online help](#) documentation.
- *Open child as modal form* – Opens a new screen and freezes the parent screen. The user can only continue using the parent screen if the new window is closed.
- *Make update and then open child window* – Opens a new screen in relation to the parent screen. Before opening the screen, changes in the parent window are saved.
- *Open new window* – Opens a new screen with no relation to the parent screen, therefore work can be done on both screens in parallel.

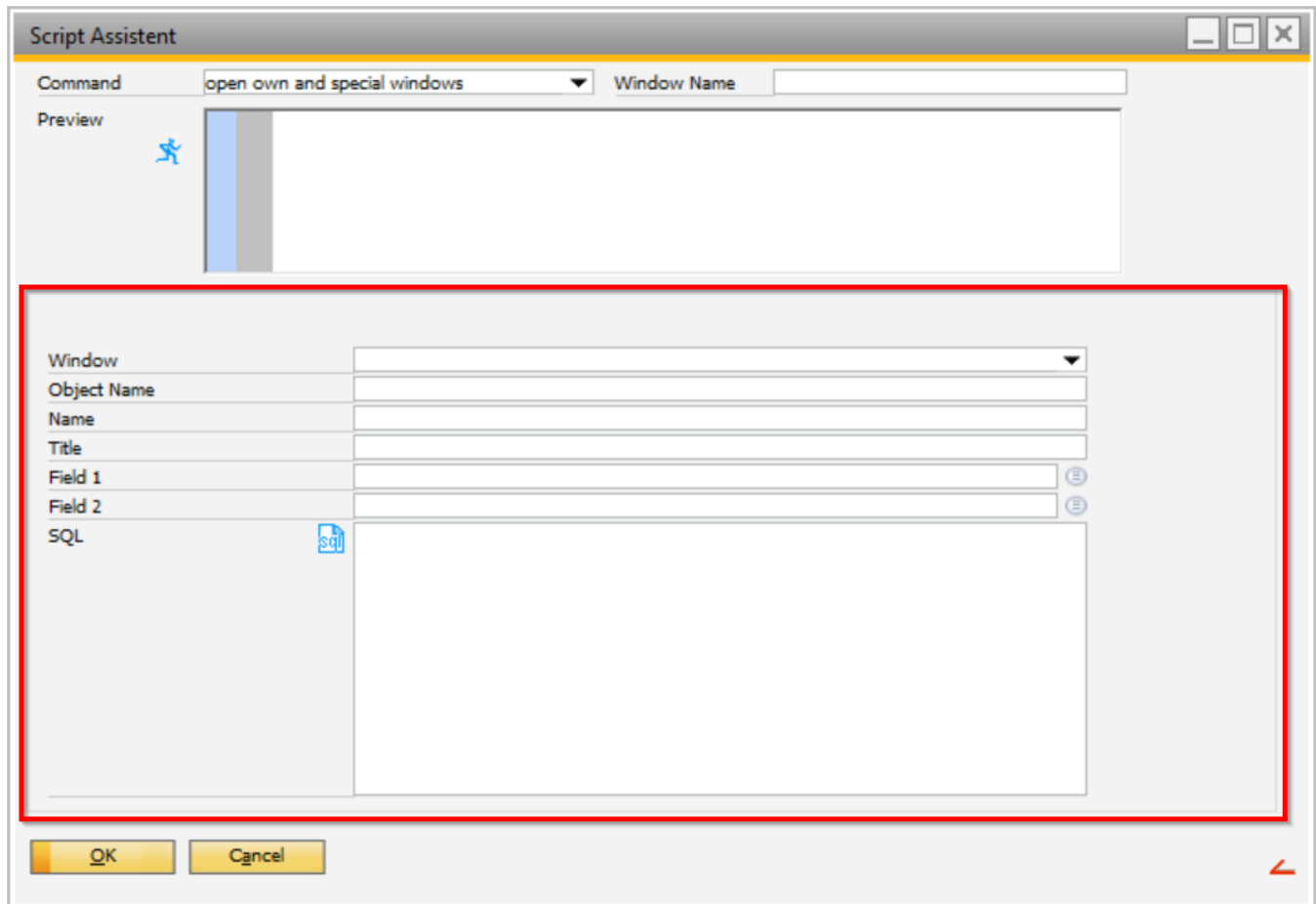
Parameter: Field used to set additional parameters. This field is only available on certain screens.

Command: Field used to set a command for the window. The options available are:

- Normal : Allows editing the current entry.
- Delete: After opening the window a prompt is displayed to confirm whether the current entry should be deleted.
- New Entry: Opens a screen without any data, so a new entry can be added. (use to enter new data in the screen)

PSR File Name: Set a PSR file to be used in the script. This option is filled automatically by Beas once the screen is selected.

1.1.2. The Open Own and Special Windows command



Window: The type of screen to be opened. The following options are available:

- *SQL Editor* – Opens a screen where you can test, create or analyze SQL statements
- *Text Editor* – Opens an editor screen which allows you to edit and save changes to a field
- *Script Editor* – Opens the Beas Script Editor
- *HTML Browser* – Opens a HTML window
- *HTML Editor* – Opens an editor screen, which allows editing HTML documents
- *Own Edit Window* – Opens the user’s own screen

Depending on the selected option, the screen displays the fields linked to the selected window type.

Object Name: The name of the object. This variable is filled in automatically. This field is available if the "SQL Editor", "Text Editor", "HTML Editor" or "Own Edit Window" option is selected in the *Window* field.

Name: Sets the name of the new screen. This field is available if the "Own Edit Window" option is selected in the *Window* field.

Title: Sets the title of the new screen. This field is displayed if the "Own Edit Window" option is selected in the *Window* field.

For field: Sets the parameters for the screen. This field is displayed if the option "SQL Editor", "Text Editor" or "Script Editor" in the *Window* field is selected.

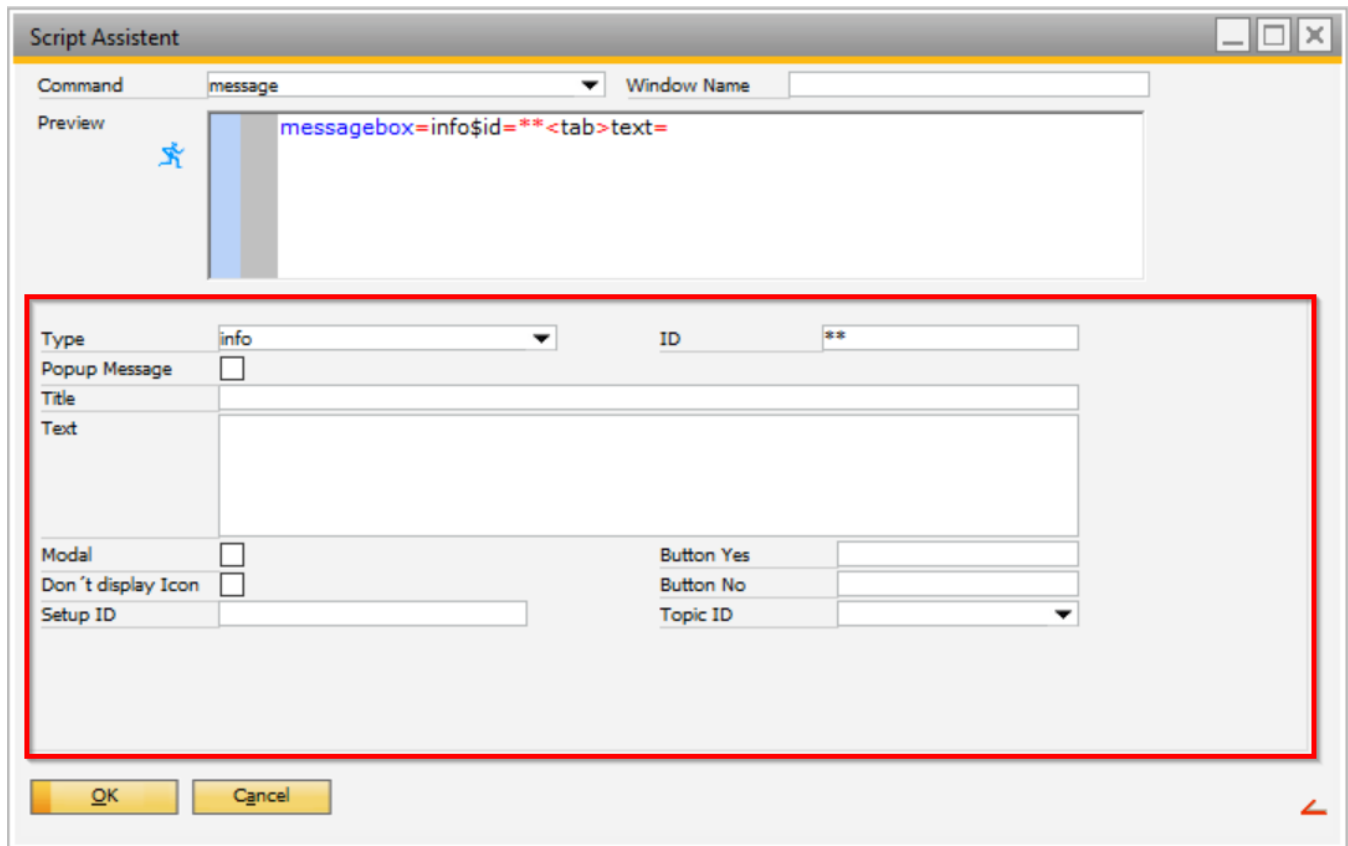
Fill Field 1: Sets additional parameters for the window. This field is displayed if the "Own Edit Window" option is selected in the *Window* field.

Fill Field 2: Sets additional parameters for the window. This field is displayed if the "Own Edit Window" option is selected in the *Window* field.

SQL: Defines a SQL statement to be executed in the new screen. This field is displayed if the "Own Edit Window" option is selected in the *Window* field. For more information, please refer to the *Universal Function* tutorial.

Complete URL Address or Placeholder: Specifies the URL address for Beas to connect to. This field is displayed if the "HTML Browser" or "HTML Editor" option is selected in the *Window* field.

1.1.3. The Message command



Type: Sets the type of message to be displayed. The options are:

- *Error* – Displays message as an error message
- *Info* – Displays message as an informational message
- *Ok* – Displays message as a confirmation message
- *Question* – Displays a question message with two buttons, one for “Yes” and one for “No”. The text of the “Yes” and “No” buttons can be changed using the fields *Button Yes* and *Button No*
- *Question-no* – Displays a question message with “No” as default value
- *SQL* – Displays the SQL code and SQL error message in the screen footer

IMPORTANT: Depending on the selected type only certain fields are available.

ID: Defines an ID for the message. This field is always displayed.

Popup Message: Sets whether the text is to be shown in a message pop-up screen. This field is always displayed.

Title: Sets a title for the new message. This field is displayed if the option “Question”, “Question-no” or “SQL” in the *type* field is selected.

Text: The text to be displayed in the message. This field is always displayed.

Modal: Stop the process that the running process. The user must close the message to continue. This field is displayed if the option "Question", "Question-no" or "SQL" in the *type* field is selected.

Don't Display Icon: Adds an option to display an icon on the message. This field is displayed if the option "Question", "Question-no" or "SQL" in the *type* field is selected.

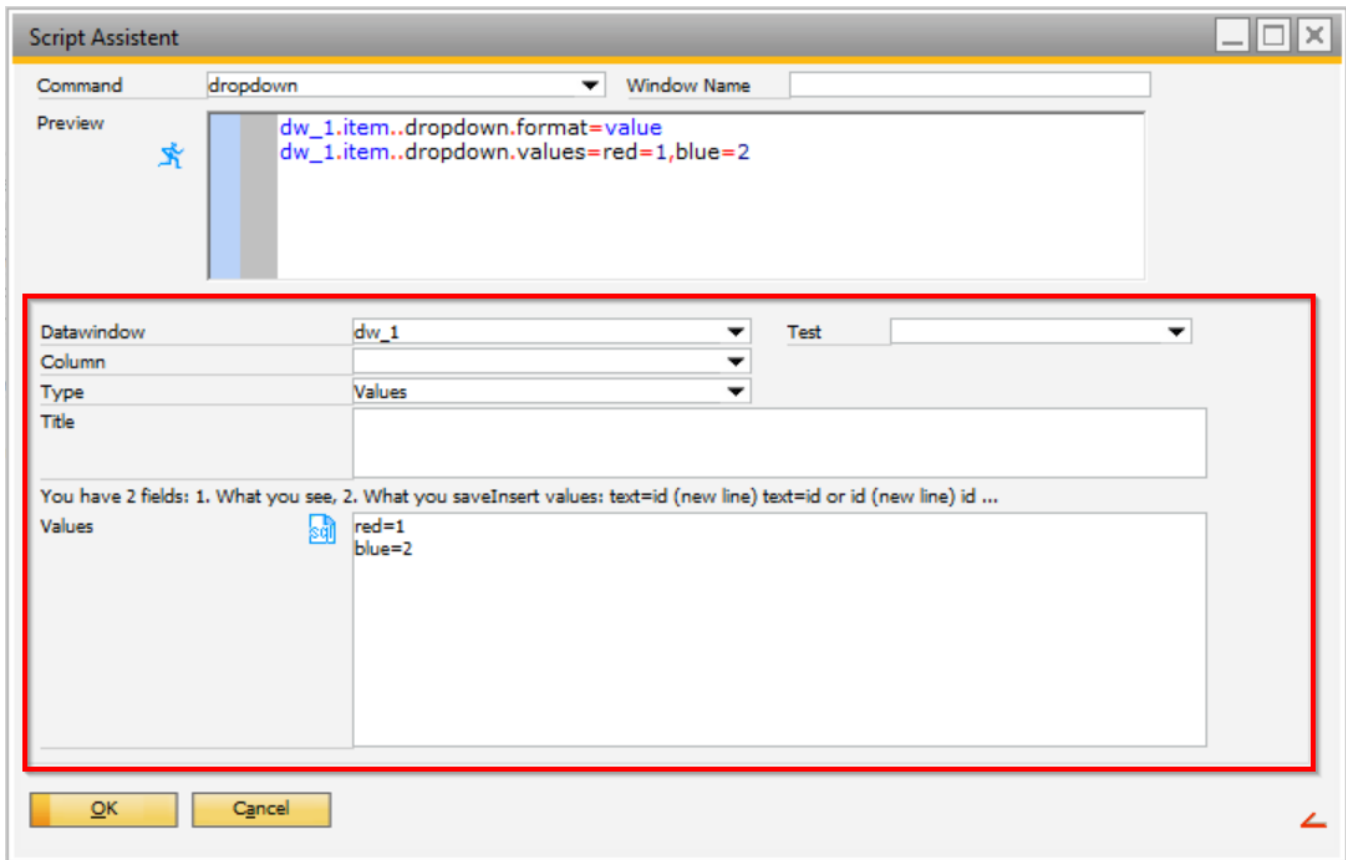
Setup ID: Defines a value that creates a *Setup* button in the message box. The function of this button is to connect to a specific section in the Configuration Wizard. This field is displayed if the option "Question", "Question-no" or "SQL" in the *type* field is selected.

Button Yes: Defines a value to replace the "Yes" text of the button. This is used for question messages. This field is displayed if the option "Question" or "Question-no" in the *type* field is selected.

Button No: Defines a value to replace the "No" text of the button. This is used for question messages. This field is displayed if the option "Question" or "Question-no" in the *type* field is selected.

Topic ID: Field used to link information from the online help documentation to the message. The dropdown arrow will open the help file and the user is able to select a specific topic. This creates a shortcut button on the popup message. This field is displayed if the option "Question", "Question-no" or "SQL" in the *type* field is selected.

1.1.4. The Dropdown command



Datawindow: Define the datawindow that contains the field for which the dropdown list is added.

Test: Field used to test your displayed values. The options available in this example are:

- Red
- Blue

Column: Set the field for which the dropdown list is added.

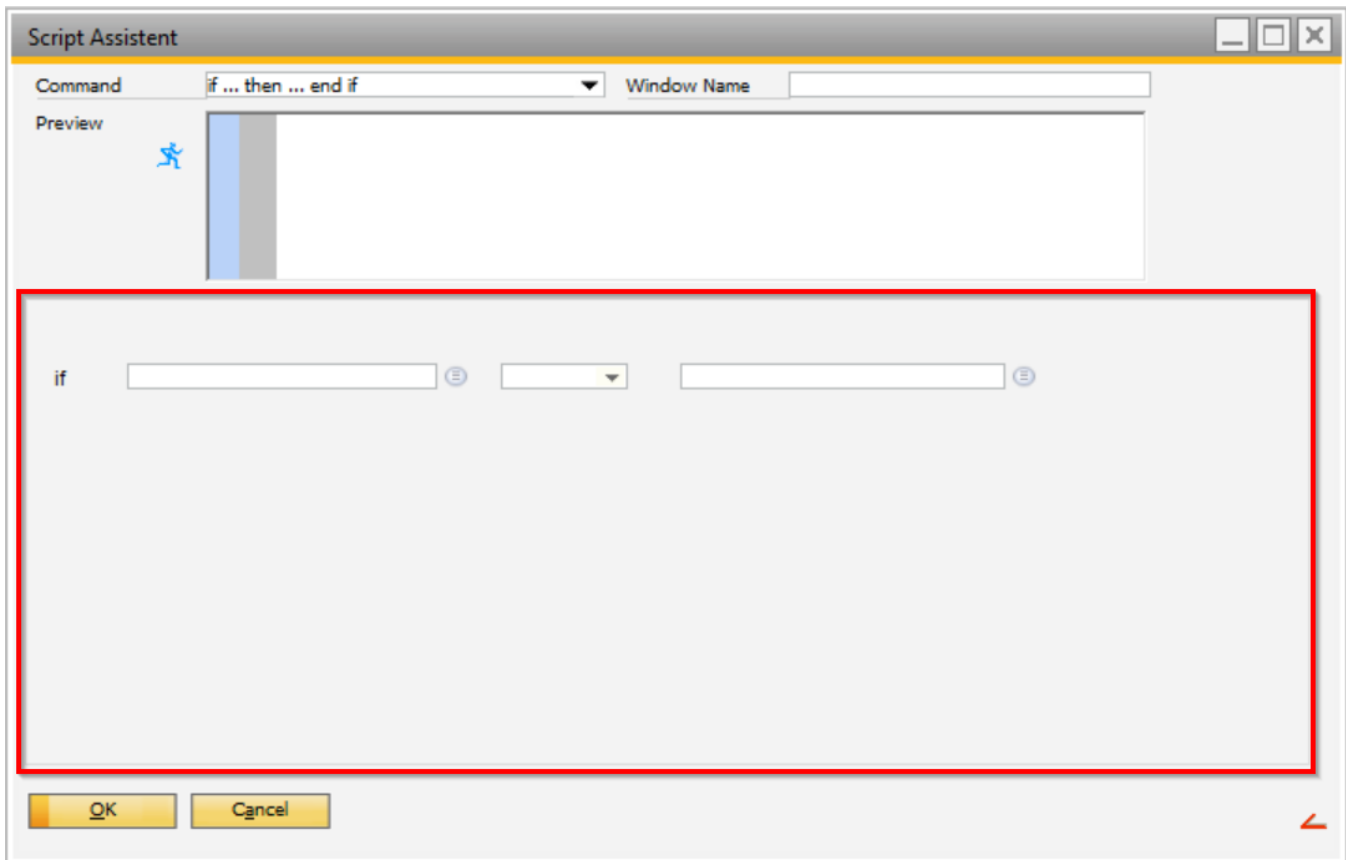
Type: Set the type of the dropdown. The following options are available:

- Picture
- Color
- Values
- Values with Title
- SQL Statement
- Calendar

Title: Defines a title for the dropdown list.

Values: Sets a list of values that are added to the dropdown list.

1.1.5. The conditional statement: *if...then...end if*



Variable1: Sets a variable for the *if* statement. This can be selected from the predefined list.

Operator: Sets an operator for the *if* statement. This can be selected from the predefined list.

Variable2: Sets a variable for the *if* statement. This can be selected from the predefined list.

The structure of an *if* statement is as follows:

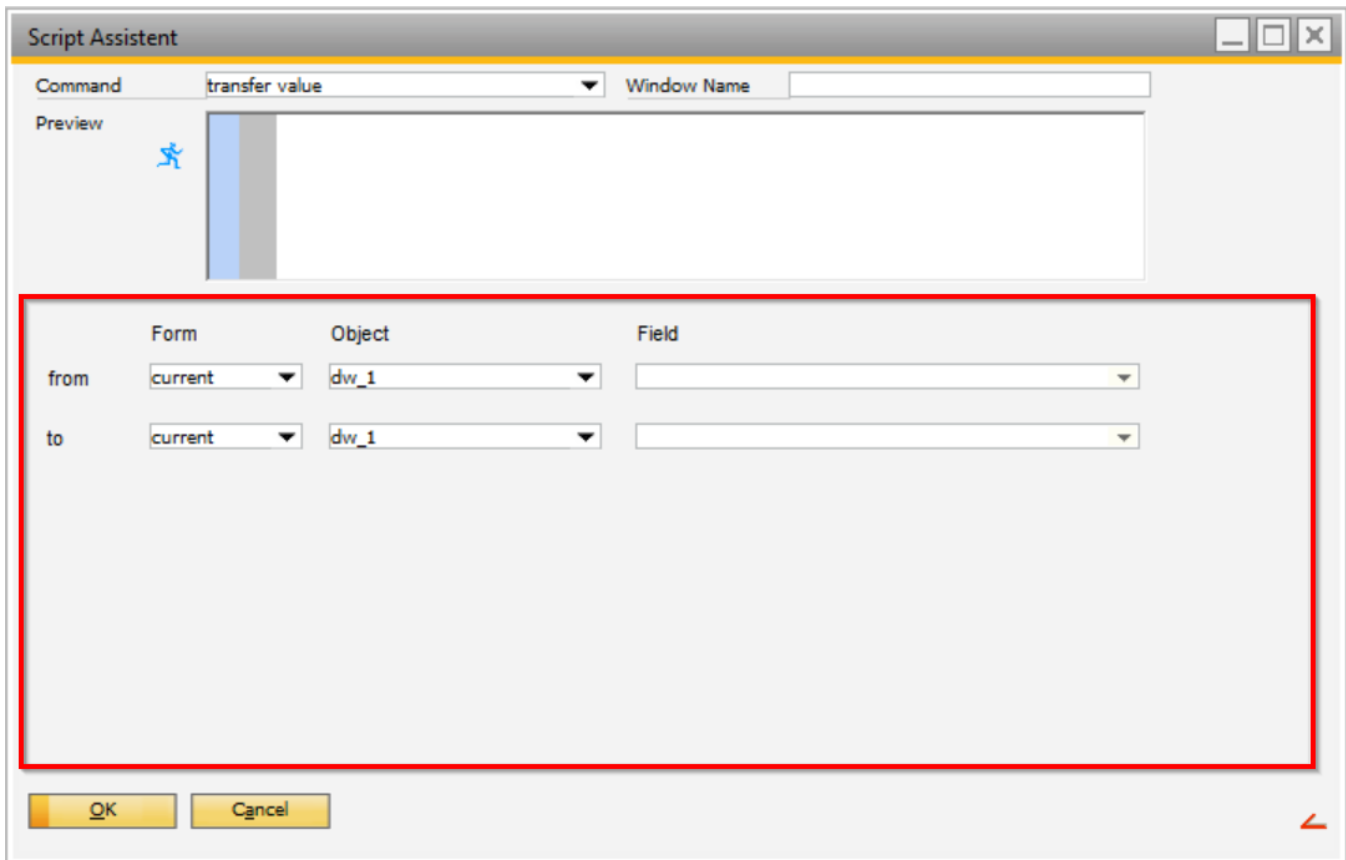
```
if [variable or field value] [comparison operator] [variable or field
value] then
...
elseif [variable or field value] [comparison operator] [variable or
field value] then
...
else
...
end if
```

The following comparison operators are available:

- < lower than
- > greater than
- >= greater or equal than
- <= lower or equal than
- = equal
- <> not equal
- n< lower than (for numerical comparisons)
- n> greater than (for numerical comparisons)
- n<> not equal (for numerical comparisons)
- n= equal (for numerical comparisons)
- n>= greater or equal than (for numerical comparisons)
- n<= lower or equal than (for numerical comparisons)
- in check if operator 1 is in operator 2. Operator 2 must be in {...}. Example: if a in {1,5,20,10} then ..

IMPORTANT: There must be a space before and after the Operator. The Script Assistant adds this automatically but be aware if a change is made manually.

1.1.6. The Transfer Value command



From

Form: Define the form where the value is transferred from. The available options are:

- *Current*
- *Parent*
- *Child*

Object: Select the datawindow to use.

Field: Specify the field to transfer data from.

To

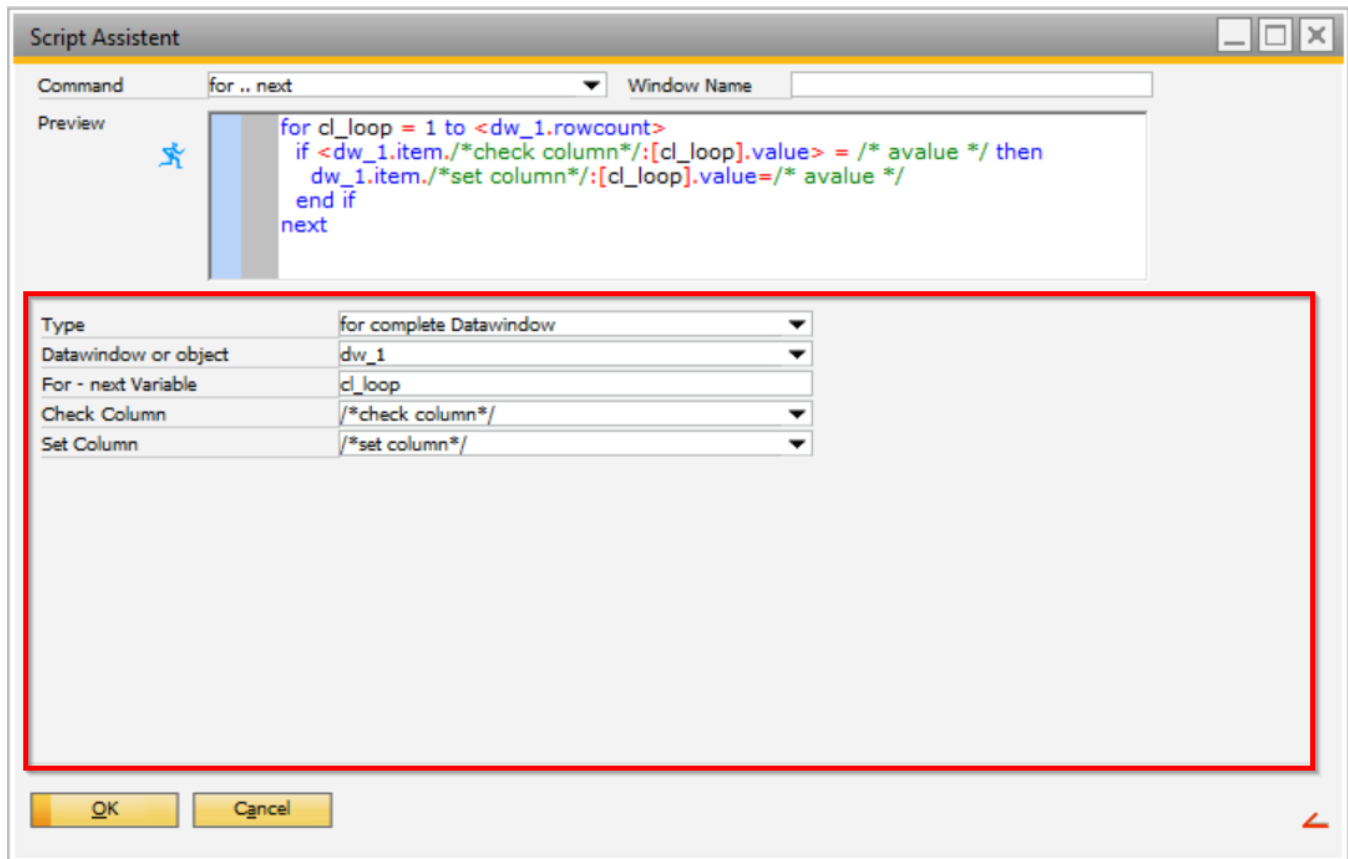
Form: Define the form where the value is transferred to. The available options are:

- *Current*
- *Parent*
- *Child*

Object: Select the datawindow to use.

Field: Specify the field to transfer data to.

1.1.7. The for...next loop



Type: Define the data container to be looped. The available options are:

- For Complete Datawindow
- For Selected Row in Datawindow
- Own SQL Datastore

IMPORTANT: Depending on the selected type, the screen displays only certain fields.

Datawindow or object: Set the part of the screen or object that is to be looped.

For – Next Variable: Define a numeric variable that is increased in each loop, until the loop has reached its final value.

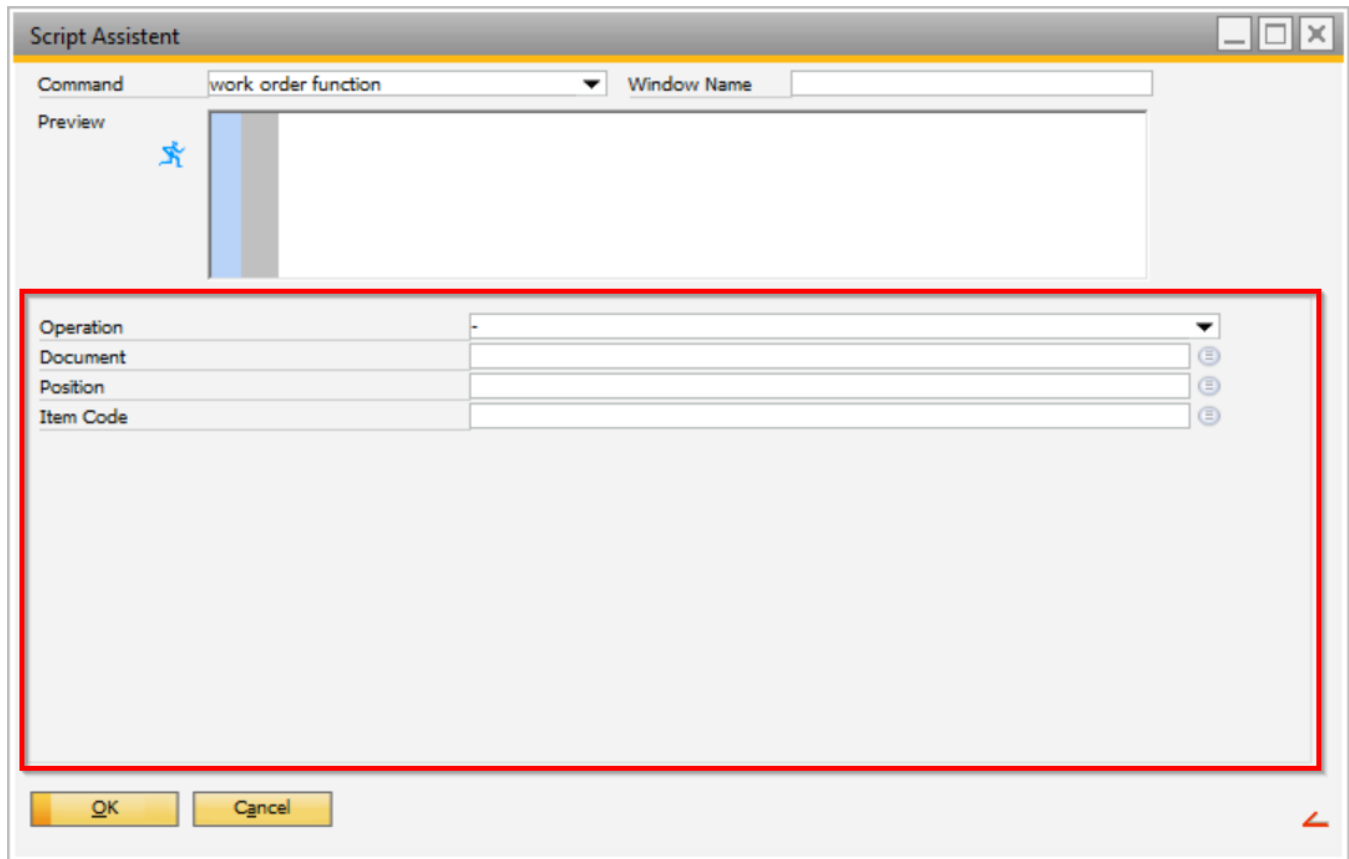
Example: For [for – next variable] = [initial value] to [final value]

Check Column: Set the field to be checked This field is only visible if the type is "For Complete Datawindow".

Set Column: Set the field to be set be set if the *Check Column* condition is met. This field is only visible if the type is "For Complete Datawindow" or "For Selected Row in Datawindow".

SQL Statement: Add a SQL statement if required. This field is only visible if the type is "Own SQL Datastore".

1.1.8. The Work Order Function command



With this feature, the user can set operations automatically depending on the parameter settings.

Example: On the Item list screen, the a button can be created to create a work order for the highlighted Item.

Operation: Set the operation to execute.

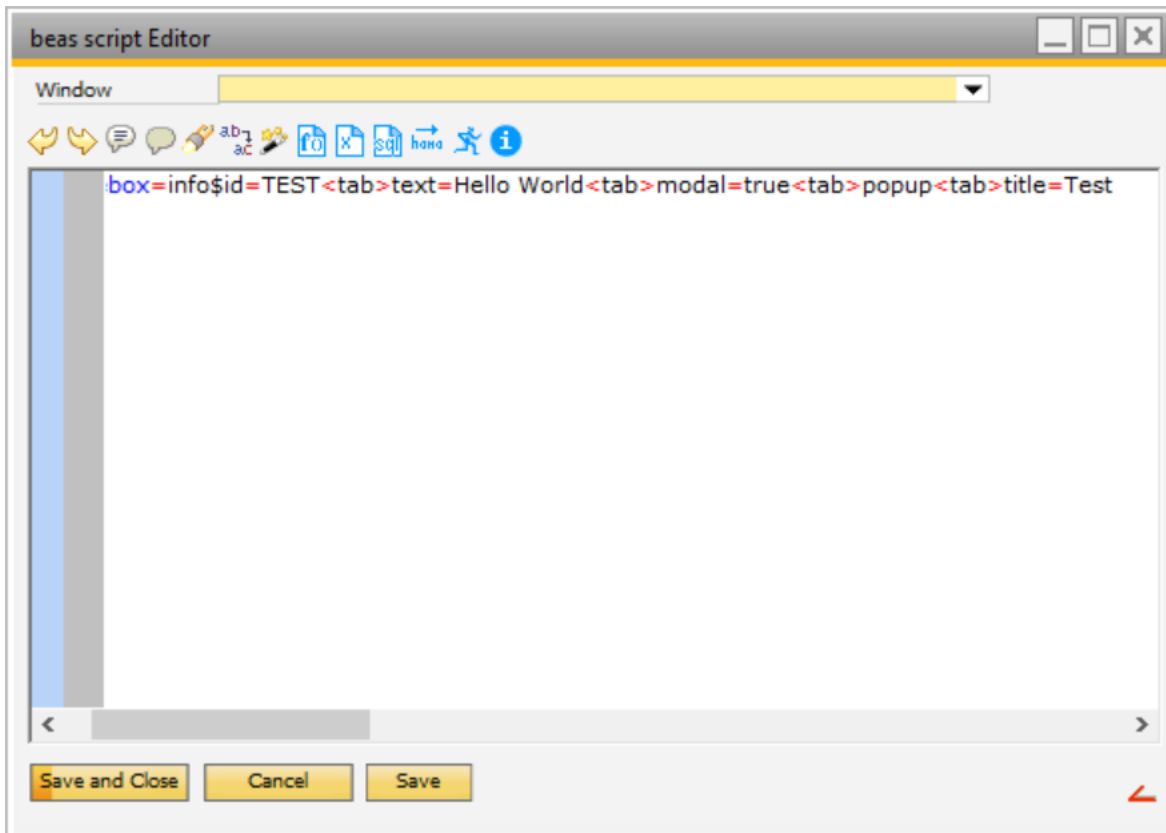
Document: Select the documents to update.

Position: Select the position to be updated.

There might be additional fields displayed that are related to the selected operation.

3.3. Saving the created script

After the Beas script is created with the *Beas Script Assistant*, it appears in the Beas script editor screen of the universal function.



The following options are available:

Save and Close: Button saves the script and close the screen.

Cancel: Button closes the screen without saving the script.

Save: Button saves the progress of the screen without closing the screen.

Refer to the [Universal Function documentation](#), which details how to connect a script to a button created by the Beas Usability extension.