

Macro conditions greatly increase the flexibility of certain features. By preparing conditions, you allow the system to dynamically make decisions. For example, when to perform an action, under which circumstances to make an item visible, and so on.

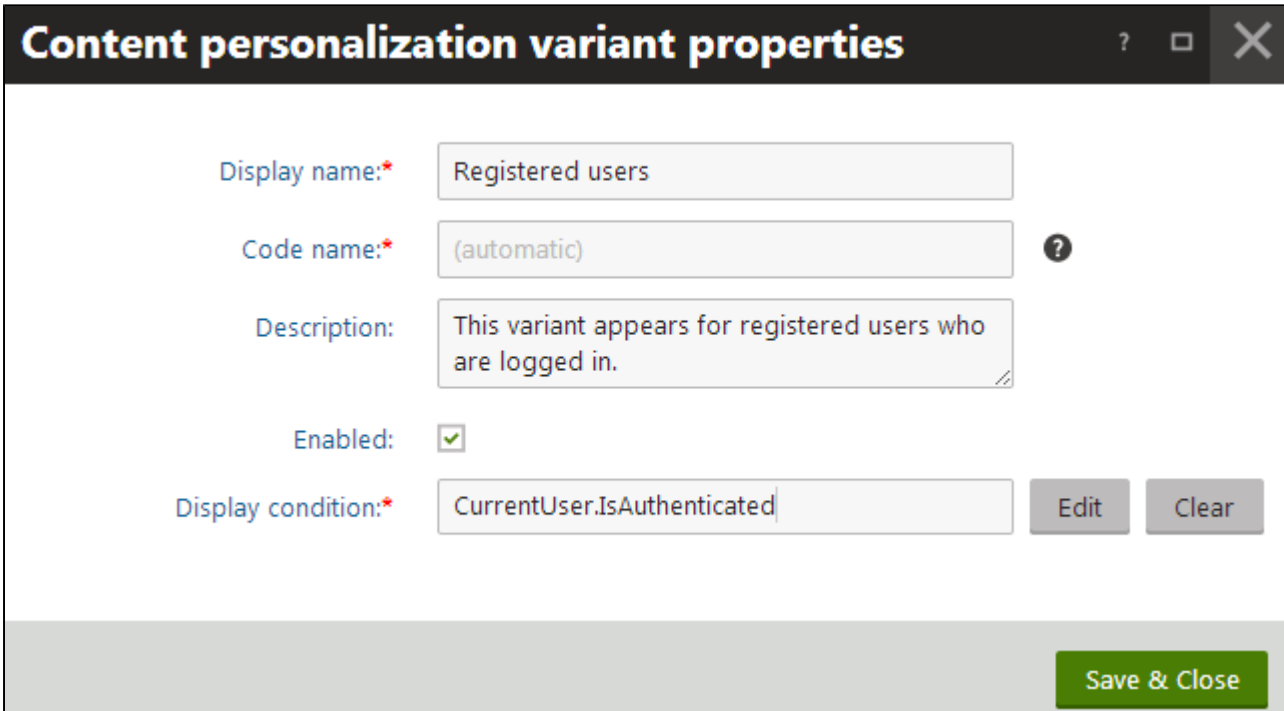
Each condition is defined by a macro expression that returns a true or false value. When the system evaluates the condition, the macro is resolved according to the currently available context data, and the result determines whether the condition is fulfilled or not.

For illustration, the following are examples of functionality that uses macro conditions:

- [Content personalization variants](#) (display conditions)
- [Form fields](#) (visibility and enabled conditions)
- [Marketing automation processes](#) (triggers and step transition conditions)
- [Advanced workflow steps](#) (step transition and branching conditions)

Entering macro conditions

The user interface provides dedicated fields for entering macro conditions. For example, [content personalization variants](#) have a **Display condition** field.



You can choose between two ways of creating macro conditions:

a) Build the condition out of macro rules

Macro rules are predefined, text-based clauses that define certain requirements. You form the overall condition by combining any number of macro rules. Using macro rules does not require knowledge of K# syntax.

To work with macro rules, click **Edit** next to the condition field. See [Building conditions using macro rules](#) for more information.

b) Write the code of the condition directly

Write a K# macro expression into the condition field. The result must be a boolean value (true/false). Use [macro methods](#) with a boolean return type, or comparison and equality operators.

Do not enclose the expression into the standard macro parentheses — the system automatically processes the content of the condition as a K# macro. You can leverage the [macro autocomplete help](#) when typing in condition fields.

✓ **Tip:** If you wish to add a macro condition field into a custom form, use the following [form control](#): **Condition builder**