

ServiceNow Application Developer

Client-side Scripting > UI Policy Actions

UI Policy Actions are client-side logic in a UI Policy used to set three field attributes:

- *Mandatory*
- *Visible*
- *Read only*

Although you can use scripts to set these attributes using the *GlideForm* (*g_form*) API, UI Policy Actions do NOT require scripting to set the field attributes.

Creating UI Policy Actions

1. In Studio, create a UI Policy or open an existing UI Policy for editing.
2. Scroll to the *UI Policy Actions* related list. If creating a new UI Policy, the UI Policy must be saved before the *UI Policy Actions* related list is visible.
3. Click the **New** button.
4. Configure the UI Policy Action.
 - a. Select a *Field* name.
 - b. Set the *Mandatory*, *Visible*, or *Read-only* field values.

True: Apply the attribute to the field.

False: Do not apply the attribute to the field.

Leave alone: The attribute does not apply to the field.
5. To clear any existing value from the field, select the **Clear the field value** option.
6. Click the **Submit** button.

When the UI Policy condition tests *true*, the UI Policy Actions are applied. In the example, the *State* field is *Read only*. The *Mandatory* and *Visible* attribute values are not changed by the UI Policy Action.

UI Policy Action state [Advanced view]

UI policy: Make fields read-only on close

Application: Global

Table: Incident [incident]

* Field name: State

Mandatory: Leave alone

Visible: Leave alone

Read only: True

Clear the field value: ☐

Contact type: -- None --

State: Closed

Impact: 1 - High

What happens when the UI Policy condition tests *false*? There are two possible outcomes:

- No action is taken
- The opposite action is taken

How does ServiceNow know which to do? The decision is made by the *Reverse if false* option in the UI Policy trigger.

- If *Reverse if false* is selected (default), the opposite action is taken in the UI Policy Actions. If a field is mandatory (true), the field will no longer be mandatory (false). That is to say, attributes which were true become false, and false become true. There are no changes to attributes which are set to Leave alone.
- If *Reverse if false* is not selected, no UI Policy Action logic is applied.

Creating UI Policy Related List Actions

Use UI Policy Related List Actions to show or hide related lists. The *Problem* form has two related lists:

- *Incidents*
- *Problem Tasks*

Incidents (1) Problem Tasks

Incidents New Edit... Search Short description Search

Problem = PRB00000003

	Number	Opened	Short description	Caller	Priority	State	Category	Assignment group
<input type="checkbox"/>	INC00000020	2018-08-09 16:51:35	I need a replacement iPhone, please	Fred Luddy	5 - Planning	In Progress	Inquiry / Help	(empty)

Actions on selected rows...

1. In Studio, create a UI Policy or open an existing UI Policy for editing.
2. Scroll to the **UI Policy Related List Actions** related list. If creating a new UI Policy, the UI Policy must be saved before the *UI Policy Related List Actions* related list is visible.
3. Click the **New** button.
4. Configure the UI Policy Related List Action.
 - a. Select a *List* name.
 - b. Set the *Visible* field value.

True: Apply the attribute to the related list.

False: Do not apply the attribute to the related list.

Leave alone: The attribute does not apply to the related list.
5. Click the **Submit** button.

UI Policy Related List Action

New record [Advanced view]

UI policy

Hide incidents related list

-- None --

✓ Incident->Problem

Problem Task->Problem

Application

Global

List name

Visible

False

Submit

The *Incidents* related list is hidden:

Problem Tasks

Problem Tasks

New

Edit...

Search

Number

Search

Problem = PRB0000003

Number ▲

Configuration item

Priority

State

Short description

Assignment group

Assigned to

No records to display