

Event - Status change (final)

 Language: [Deutsch](#) · [English](#)

Event - Status change (final)

Event rule „Status change (final)“: interface, context and an example – how to react to this Nuclos event.

REFERENZ

DEVELOPER

UPDATED: JUL 2026

APPLIES TO NUCLOS 4.2026.X

On this page

- [When does the rule fire?](#)
- [Structure](#)
- [Assignment](#)
- [Example](#)
- [Related pages](#)

When does the rule fire?

This [server-side rule](#) runs **after** a successful status change. It can be assigned to a **status model**.

Structure

Skeleton of the rule class (the rule editor creates it automatically):

```
package org.nuclet.lager;

import org.nuclos.api.rule.StateChangeFinalRule;
import org.nuclos.api.context.StateChangeContext;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.exception.BusinessException;

/**
 * @name
 * @description
 * @usage
 * @change
 */

@Rule(name="Statuswechsel im Anschluss Lagerposition", description="Statuswechsel im Anschluss Lagerposition")
public class StatuswechselImAnschlussLagerposition implements StateChangeFinalRule {

    public void changeStateFinal(StateChangeContext context) throws BusinessException {
    }
}
```



Context: StateChangeContext

The context `StateChangeContext` provides the affected `BusinessObject` via `context.getBusinessObject(...)`; the `RuleContext` functions (providers) are also available. A `BusinessException` aborts the operation.

Assignment

Assign it to the target via drag-and-drop in the [server rule manager](#). Optionally restrict execution to a **status** or an **action**; control the order with the arrow buttons.

Example

```

package org.nuclet.company;

import org.nuclos.api.annotation.Rule;
import org.nuclos.api.context.StateChangeContext;
import org.nuclos.api.exception.BusinessException;
import org.nuclos.api.provider.StatemodelProvider;
import org.nuclos.api.rule.StateChangeFinalRule;

/** @name
 * @description
 * @usage
 * @change
 */

@Rule(name="Auftrag beantragen", description="Prüfung und Weiterführung eines neuen Antrags")
public class Auftragbeantragen implements StateChangeFinalRule {
    public void changeStateFinal(StateChangeContext context) throws BusinessException {

        Auftrag neuerAuftrag = context.getBusinessObject(Auftrag.class);

        // Wurde noch keine Auftragsnummer hinterlegt, kann der Auftrag nicht
        // ausgeführt werden. Wir brauchen eine weitere Prüfung
        if (neuerAuftrag.getAuftragsnr() == null) {
            // Neuer Status: Prüfen (Numeral 55)
            StatemodelProvider.changeState(neuerAuftrag, ProzessAuftragSM.State_55);
        }
        else {
            // Prüfung ok, neuer Status: beauftragt (Numeral 30)
            StatemodelProvider.changeState(neuerAuftrag, ProzessAuftragSM.State_30);
        }
    }
}

```

Related pages

Server-side rules

Basics.

[Open](#)

Server rule manager

Assign.

[Open](#)

Rule execution order

Order.

[Open](#)