

Event - Update (final)

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

Event - Update (final)

Event rule „Update (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 record has been updated successfully. It can be assigned to a **business object**.

Structure

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

```
package org.nuclet.lager;

import org.nuclos.api.rule.UpdateFinalRule;
import org.nuclos.api.context.UpdateContext;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.exception.BusinessException;

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

@Rule(name="Aktualisieren Lagerposition im Anschluss", description="Aktualisieren Lagerposition im Anschluss")
public class AktualisierenLagerPositionImAnschluss implements UpdateFinalRule {

    public void updateFinal(UpdateContext context) throws BusinessException {
    }
}
```



Context: UpdateContext

The context UpdateContext 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.

Regel	Status	Aktion
AktualisierenLagerPositionImAnschl...		

Assignment of „Update (final)“.

Example

```
package org.nuclet.company;

import java.util.List;
import org.nuclos.api.rule.UpdateRule;
import org.nuclos.api.context.UpdateContext;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.exception.BusinessException;

/** @name
 * @description
 * @usage
 * @change
 */
@Rule(name="UpdateFinalBestellung", description="UpdateFinalBestellung")
public class UpdateFinalBestellung implements UpdateFinalRule {
    public void updateFinal(UpdateContext context) throws BusinessException {

        // this is the businessObject after storing the updated data in the database
        Bestellung curBestellung = context.getBusinessObject(Bestellung.class);

        // changes are not recognized, because data has already been saved
        // the following description will not be stored in the database
        curBestellung.setKurzbeschreibung("Bestellung wurde im Schnellverfahren abgeschlossen.");

        // we want to re-update all positions after the order has been updated properly
        // so we use the QueryProvider
        for (Bestellposition pos : curBestellung.getPosition()) {
            pos.setBemerkung("Position im Gesamtkontext von " + curBestellung.getNr());
            pos.save();
        }
    }
}
```

Related pages

Server-side rules

Basics.

[Open](#)

Server rule manager

Assign.

[Open](#)

Rule execution order

Order.

[Open](#)

