

Event - Object generation (final)

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

📖 Event - Object generation (final)

Event rule „Object generation (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 object generation. It can be assigned to a **object generator**.

Structure

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

```
package org.nuclet.lager;

import org.nuclos.api.rule.GenerateFinalRule;
import org.nuclos.api.context.GenerateContext;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.exception.BusinessException;
/
** @name
 * @description
 * @usage
 * @change
 */

@Rule(name="Objektgenerierung im Anschluss", description="Objektgenerierung im Anschluss")
public class ObjektgenerierungImAnschluss implements GenerateFinalRule {

    public void generateFinal(GenerateContext context) throws BusinessException {
    }
}
```



Context: GenerateContext

The context `GenerateContext` 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.rule.GenerateRule;
import org.nuclos.api.context.GenerateContext;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.exception.BusinessException;
import java.util.Collection;

/** @name
 * @description
 * @usage
 * @change
 */
@Rule(name="BestellungErstellen", description="BestellungErstellen")

public class BestellungErstellen implements GenerateRule {

    public void generate(GenerateContext context) throws BusinessException {
        Collection<Anfrage> sourceObjects = context.getSourceObjects(Anfrage.class);

        int sumOfArticles = 0;

        for (Anfrage a: sourceObjects) {
            // How many articles are in this inquiry
            sumOfArticles += a.getAnfrageposition().size();
        }

        Bestellung targetObject = context.getTargetObject(Bestellung.class);

        // create new history entry
        BestellungHistory newHistory = new BestellungHistory();
        newHistory.setBestellungId(targetObject.getId());
        newHistory.setBemerkung(sumOfArticles + " Einträge");

        // save new entry
        BusinessObjectProvider.insert(newHistory);
    }
}

```

Related pages

Server-side rules

Basics.

[Open](#)

Server rule manager

Assign.

[Open](#)

Rule execution order

Order.

[Open](#)