

Server-side rules

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

Server-side rules

Java business rules on the server: bound to events, with access to business objects, providers and the full Nuclos API.

REFERENZ

DEVELOPER

UPDATED: JUL 2026

APPLIES TO NUCLOS 4.2026.X

On this page

- What are server-side rules?
- Events
- Rule editor
- Application-specific objects
- Providers (supporting classes)
- Rules, subforms and the rule context
- Related pages

What are server-side rules?

The Nuclos API enables Java-based business rules that read and write content on certain **events**. When the other low-code means are not enough, implement complex validations, calculations, web service calls or automatic data changes here.

 API

The full API is documented at api.nuclos.de.

Menu: [Configuration](#) [Rule sets](#) [Server-side rules](#)

Events

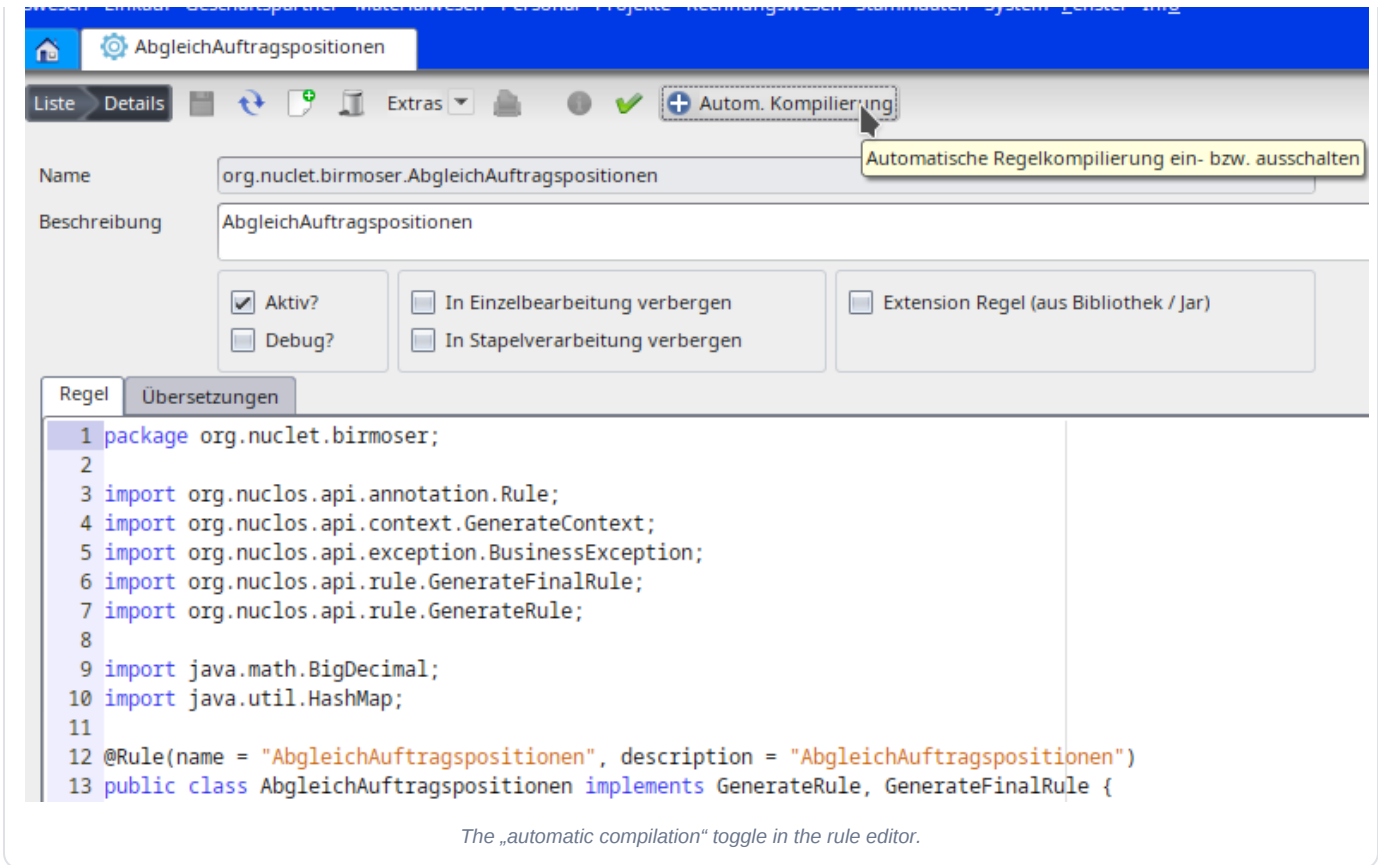
Event	Rule interface(s)	Example
Create	InsertRule, InsertFinalRule	new record (e.g. invoice)
Update	UpdateRule, UpdateFinalRule	existing record (e.g. order)
Delete	DeleteRule, DeleteFinalRule	existing record
Status change	StateChangeRule, StateChangeFinalRule	e.g. open reminder
Object generator	GenerateRule, GenerateFinalRule	e.g. delivery note from order
Button	CustomRule	user action in the layout
Job	JobRule	scheduled, recurring

Rule editor

The rule editor is the central tool for editing rules. For convenient development/debugging there is also the IntelliJ integration.

 Automatic compilation

The **automatic compilation** toggle temporarily disables compiling on save (useful for interdependent, not-yet-compilable rules). **Note:** uncompiled rules are not visible in the [rule manager](#).



Application-specific objects

Nuclos automatically generates representing objects for business objects, status models, data sources and reports (with getters/setters). If this metadata changes, dependent rules no longer compile – a direct hint at what to adjust.



BigDecimal

Decimal attributes are mapped to `java.math.BigDecimal`. Use `BigDecimal.valueOf(double)` instead of `new BigDecimal(double)` and always `.divide(BigDecimal, RoundingMode)` for division.

Providers (supporting classes)

Providers offer functions to change content from rules: **QueryProvider**, **StatemodelProvider**, **GenerationProvider**, **DatasourceProvider**, **BusinessObjectProvider**, **MailProvider**, **PrintoutProvider**, **ReportProvider**, **FileProvider**.

Rules, subforms and the rule context

- In rules **before** the event (insert/update/state change/generate), changes to the context object via attribute setters are applied automatically; `delete()` on dependent objects counts as a change.
- In **final** rules an explicit `save()` is needed to change other objects – this triggers their update/update-final rules.



Endless loops

A `save()` on the context object in an `UpdateFinalRule` can cause an **endless loop** – consider [disabling rule execution](#) or [SaveFlags](#).

Related pages



Rule sets

Overview.

[Open](#)



Server rule manager

Assign rules.

[Open](#)



Rule execution order

Order per action.

[Open](#)

