Businessobjekt

- Definition
- Konfiguration
  - 1.Businessobjekt erstellen
  - 2. Allgemeine Eigenschaften
  - 3. Definition von Attributen
  - 4. Attribute bearbeiten
    - 4.1 Eigenschaften
    - 4.3 Verknüpfung zu anderen Businessobjekten /Referenzfeld
    - 4.6 Valuelist Provider
    - 4.7 Client Regeln
    - 4.8 Übersetzungen
    - 4.9 Allgemeine Eigenschaften
  - 5. Kontext und Aktionen definieren
  - 6. Menü konfigurieren
  - 7. Beschriftung und Baumdarstellung definieren
    - 7.1 Baumdarstellung
    - 7.2 Unterknoten bearbeiten
  - 8. Rechte und Sperrfunktion verwalten
  - 9. Übersetzen
  - 10. Darstellung konfigurieren

Definition

Menüaufruf: (Konfiguration) - (Businessobjekt)

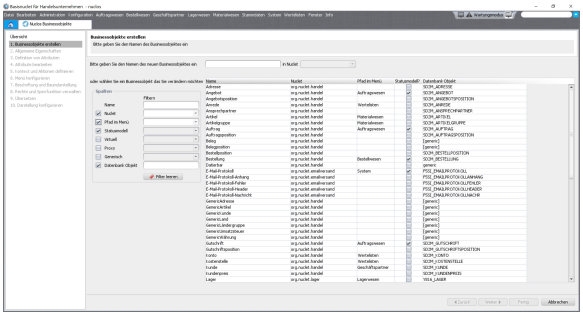
Alle in Nuclos erfassten Daten des Unternehmens werden in Businessobjekte geordnet. Ein Businessobjekt entspricht in der Sprache der Datenmodelle einer Entität. Anders ausgedrückt ist es eine Abstraktion von Dingen, die man im Unternehmen gern verwalten möchte. Der Name des Businessobjektes kennzeichnet diese Abstraktion sprachlich idealerweise so, dass jede/r im Unternehmen weiß, was damit gemeint ist. Typische Beispiele sind z.B. "Kunde", "Artikel" oder "Auftrag".

Businessobjekte können, müssen aber nicht, ein Statusmodell haben. Im Statusmodell werden meist betriebliche Prozesse dieses Businessobjektes abgebbildet, z. B. bestellt geliefert bezahlt. Es ist aber auch möglich, im Statusmodell bestimmte dauerhafte Verwaltungszustände darzustellen wie z. B. aktiv - inaktiv.

Im Folgenden wird ausführlich und Schritt für Schritt erklärt wie sich ein Businessobjekt anlegen lässt und Erklärungen zu verschiedenen Einstellungen.

Konfiguration

1.Businessobjekt erstellen



Die erste Seite des BO-Wizard gibt einen Überblick über die existierenden Businessobjekte.

Im Bereich **Spalten** kann die Ansicht über das Feld "Name" **gefiltert** werden. Desweiten können **Spalten ein- oder ausgeblendet** werden, indem die Checkboxen aktiviert oder deaktiviert werden. Weitere **Filterm öglichkeiten** über die Werte der einzelnen Spalten ergeben sich über die Auswahl eines Wertes aus der Combobox.

Die eingeblendeten **Spalten** können außerdem **alphabetisch sortiert** werden, indem auf den jeweiligen Spaltenheader geklickt wird.

- **Erstellen**

um ein neues BO anzulegen, wird im oberen Textfeld der Name eingegeben und ein ensprechendes Nuclet gewählt werden. Ist der Name in diesem Nuclet bereits vergeben, wird eine Meldung ausgegeben. Somit ist es möglich, gleichnamige Businessobekte anzulegen sofern diese nicht dem gleichen Nuclet zugeordnet sind.

Anhand dieses Namens wird später auch der Tabellenname der Datenbank erzeugt.

Der Interne Name darf aus max. 25 Zeichen bestehen.

- **Ändern**

  um ein BO zu ändern, wird dieses in der Liste markiert und auf **weiter** geklickt. Die weitere Bearbeitung findet in den unten folgenden Schritten statt.

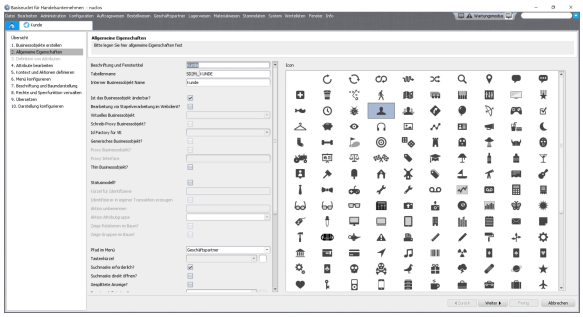
- **Löschen**

  markieren Sie das zu löschende BO in der Liste und klicken Sie auf Businessobjekt entfernen. Wird ein BO gelöscht, werden sowohl alle zugehörigen Metadaten aus dem Nuclet entfernt als auch die zugehörige Tabelle in der Datenbank.

Ein BO kann erst gelöscht werden, wenn kein anderes BO mehr darauf referenziert. Darüber wird der Anwender in einem Dialog informiert.

Alle vorhandenen Daten werden unwiederruflich gelöscht. Ein entfernes BO kann nicht wieder hergestellt werden.

2. Allgemeine Eigenschaften



Hier werden alle allgemeinen Eigenschaften für das Businessobjekt festgelegt.

- **Beschriftung und Fenstertitel**: Beschriftung des Fensters (für die BO-Maske) anpassen
- **Tabellenname**: Der Tabellenname eines Businessobjekts kann verändert/angepasst werden.
- **Interner BO Name**: Der interne Businessobjekt-Name kann ebenso angepasst werden. Er ist die Grundlage für die Methodenbenennungen in der BusinessRule API.
- **Ist das Businessobjekt änderbar?**: Wenn gesetzt, ist das BO für Änderungen durch den Anwender befähigt. Ist der Haken nicht gesetzt, so steht das BO nur lesend zur Verfügung, unabhängig von der Konfiguration in den Benutzergruppen. Sinnvoll für z.B. Schnittstellendaten, feste Wertelisten, dessen Daten aus anderen Systemen bezogen werden.
- **Bearbeitung via Stapelverarbeitung im Webclient?**:
- **Virtuelles Businessobjekt:** Die Daten eines virtuellen BOs basieren auf einer View
- **Pfad im Menü**: Unter welchem Menüpunkt soll das BO aufrufbar sein. Unterordner werden durch einen " \ " getrennt.
- **Benutzerdefiniertes Icon**: Anstelle der vorgegebenen Icons können auch eigene genutzt werden (müssen als Ressource angelegt werden)
- **Tastenkürzel**: Es können Tastenkombinationen zum Öffnen der BO-Maske angegeben werden
- **Suchmaske erforderlich?:** Falls für das BO eine Suchmaske (Suche nach Attributen) benötigt wird
- **Suchmaske direkt öffnen?**: Soll beim Öffnen des BO direkt eine Suchmaske geöffnet werden, ansonsten wird initial die Listenansicht geöffnet.
- **Ist das BO änderbar?**: Sollen Daten des BO änderbar sein oder schreibgeschützt. Im schreibgeschützten Modus sind alle Buttons deaktiviert.

- **geplittete Anzeige**: Es wird innerhalb eines Fensters die Suchmaske und Suchergebnisse angezeigt
- **Historie**: Daten eines BO können historisiert werden.
- **Statusmodell**: Das BO enthält ein Statusmodell. Die Definition eines Statusmodells ist dann Pflicht.
- **Kürzel für Identifizierer**: Kürzel für Datensatzidentifizierer. Dieser setzt sich zusammen aus

  ```
  <Kürzel><MM><YY>-<fortlaufende Nummer>.
  Beispiel: KU0716-00001.
  ```

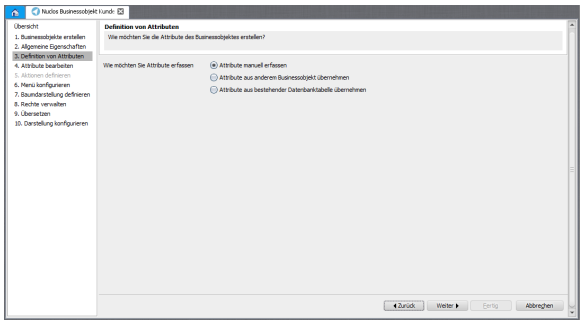
  Der Identifizierer kann nur im Zusammenhang mit einem Statusmodell aktiviert werden.
- **Identifizierer in eigener Transaktion erzeugen**: Normalerweise werden alle Datenbankoperationen beim Anlegen eines BO-Datensatzes in einer Transaktion ausgeführt. Dadurch sind die Systemidentifizierer für ein BO lückenlos fortlaufend, aber es kann nur ein Datensatz gleichzeitig angelegt werden. Ist diese Option aktiviert wird der Systemidentifizierer in einer eigenen Transaktion erzeugt. Dadurch wird das parallele Anlegen mehrerer Datensätze ermöglicht, aber einzelne Nummern für Systemidentifizierer werden u.U. übersprungen.
- **Aktion umbenennen**: Das Feld "Aktion" bekommt einen abweichenden Namen (wird für Label, Spaltenname usw. verwendet
- **Mandantenabhängigkeit**: Zuordnung einer Mandantenebene und somit die Aktivierung der Mandantenfähigkeit für dieses BO. Details siehe Mandant
- **Mandanten Erstbefüllung mit**: Mandantenzuordnung bei Neuanlage eines Datensatzes.
- **Eindeutigkeit pro Mandant?:** (ab Nuclos v4.25) Wenn eindeutige Attribute für dieses BO existieren, so wird der Mandant ebenfalls Teil des Schlüssels (Unique Keys). Damit kann z.B. ein Artikelname, obwohl als eindeutig definiert, pro Mandant einmalig verwendet werden. Gibt es keine eindeutigen Attribute wird diese Information ignoriert, bis das erste eindeutige Attribut angelegt wird.

  > Achtung: Die Geschäftslogik sollte auf Konflikte geprüft werden! Es könnte z.B. mandantenübergreifende Import Jobs geben die, wenn über einen Artikelnamen ein Artikel gesucht, erwarten, dass nur ein Datensatz gefunden wird. Auch Datenbank Prozeduren gehören zu den zu überprüfenden Komponenten.

- **Kommentar**: Kommentar für das BO
- **Virtuelles BO**: Auswahl eines vorher definierten virtuellen Businessobjektes. Die entsprechenden Spalten werden im nächsten Schritt als Attribute aufgelistet.
- **Schreib-Proxy BO**: Dadurch wird das Businessobjekt als Interface angelegt, z.B. um in einer Regel verwendet zu werden mit den Methoden zur Modifikation und Erstellung von Datensätzen. Gelesen werden die Daten aus der View die im virtuellem BO hinterlegt sind. Im Ergebnis ist das virtuelle BO änderbar. Wichtig, die Implements-Anweisung muss voll qualifiziert mit Package angegeben werden, damit Nuclos die Implementierung erkennt.
- **Id Factory für VE**: Auswahl einer vorher definierten Datenbankfunktion für die erzeugung einer ID, falls neue Daten eines virtuellen BOs erzeugt werden sollen.
- **Thin Businessobjekt**: Von der API werden keine Referenzattribute dieses BOs mehr aufgelöst, es gibt also nur noch die reinen ...Id(...) Methoden, wie z.B. die Methode "getKundeId()". "getKunde()" wird eingespart, womit die Performance gesteigert werden kann.
  Bei Thin-Businessobjekten können viele JOINS/SUBSELECTS eingespart werden, um den Anzeigeausdruck für Referenzen zu laden. Es werden nur die IDs der Referenzen geladen.
- **Proxy BO**: Dadurch wird das Businessobjekt als Interface angelegt, z.B. um in einer Regel verwendet zu werden
- **Proxy Interface**: Muss von einer Regelklasse implementiert werden. Wichtig, die Implements-Anweisung muss voll qualifiziert mit Package angegeben werden, damit Nuclos die Implementierung erkennt. Also z.B. "class MyProxyImpl implements de.mein.nuclet.MyProxyBO ..."

- **Generisches Businessobjekt:** dieses BO ist mit dem Setzen dieses Hakens als generisches BO für die Verwendung einer Generische Implementierung definiert.
- **Objektgenerator für Klonen**: Objektgenerator der ausgeführt wird, wenn der Klonen-Button betätigt wird. Das kann z.B. benötigt werden, wenn nicht alle Felder geklont werden sollen. In diesem Fall wird dann die Definition im Objektgenerator herangezogen.

3. Definition von Attributen

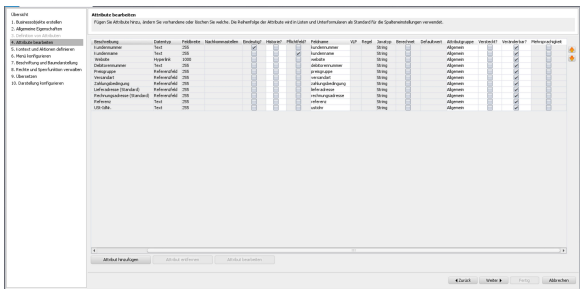


Es gibt 3 Arten um Attribute für ein BO zu erfassen: manuell, aus einem anderen BO oder einer Datenbanktabelle übernehmen.

Vorgehensweise wenn man Attribute aus anderer Quelle übernehmen will:

- Auswahl eines BO/Tabelle aus einer Dropdownliste
- Liste aller darin enthaltenen Attribute
- Auswahl welche dieser Attribute man übernehmen will

4. Attribute bearbeiten



Hier werden die Attribute des BOs gepflegt.

Die weiß hinterlegten Spalten, können direkt in dieser Liste verändert werden, ansonsten über "Attribut bearbeiten".

Mit den Pfeilbuttons wird die Reihenfolge der Attribute festgelegt. Diese wird in der Listenansicht als Default herangezogen. Hinweis: die Spalteneinstellungen für bestimmte Benutzergruppen überschrieben die o.g. Default-Reihenfolge.

Attribute können ebenfalls via Drag&Drop an die gewünscht Stelle gezogen werden.

Um Attribute leichter zu finden, ist auch eine Sortierung der Spalten möglich. Klicken sie dazu auf den jeweiligen Header der zu sortierenden Spalte. Bei aktivierter Sortierung, ist die Einstellung der Reihenfolge nicht möglich. Um die Sortierung wieder zu deaktivieren, klicken sie erneut auf den Header bis kein Sortierpfeil mehr angezeigt wird.

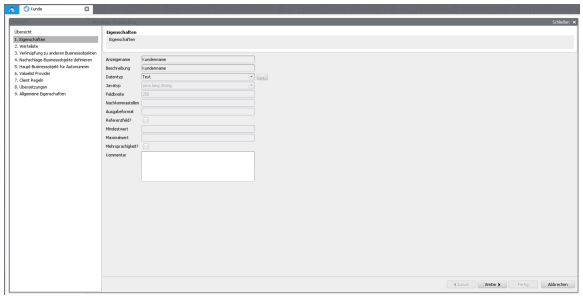
Neues Attribut hinzufügen, markiertes Attribut löschen oder markiertes Attribut bearbeiten (auch Doppelklick möglich).

Anzeigename	Beschreibung	Datentyp
Lagerumbuchung	Lagerumbuchung	Referenzfeld
Artikel	Artikel	Referenzfeld
Lager (von)	Lager (von)	Referenzfeld
Lagerort (von)	Lagerort (von)	Referenzfeld
Lager (nach)	Lager (nach)	Referenzfeld
Lagerort (nach)	Lagerort (nach)	Referenzfeld
Menge	Menge	Kommazahl (9,2)
Pos	Pos	Autonummer
Seriennr	Seriennr	Referenzfeld

Dunkelblaue Attribute repräsentieren Referenzen auf Integrationspunkte, angelehnt an den Regelmanager, welcher ebenfalls Integrationspunkte dunkelblau darstellt.

4.1 Eigenschaften

Folgende Felder können ausgefüllt werden:



- **Anzeigename:** wird als Label im Layout und bei sonstigen Beschriftungen angezeigt
- **Beschreibung**: wird als Tooltip auf dem Feld angezeigt
- **Datentyp:** Datentyp für das Feld. Erläuterungen s.u.). Die restlichen (ausgegrauten) Felder werden durch Auswahl des Datentyps ausgefüllt:
    - Javatyp
    - Feldbreite
    - Nachkommastellen
    - Ausgabeformat
    - Referenzfeld
- **Mindestwert:** nur bei Zahlen- und Datumstypen. Validierung bei der Dateneingabe
- **Maximalwert**: nur bei Zahlen- und Datumstypen. Validierung bei der Dateneingabe
- **Mehrsprachigkeit?**: Aktivierung der Mehrsprachigkeitsfunktion für dieses Attribut. Im Datensatz erscheint dann ein Icon für die Eingabe der Übersetzungen.
- **Kommentar:** interner Kommentar

Die folgenden Eingabefelder variieren je nach der Auswahl des Datentyps.

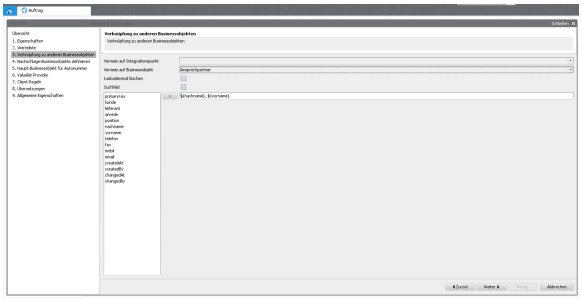
Datentypen

Es gibt verschiedene vordefinierte Datentypen (Tabelle) die standardmäßig zur Verfügung stehen, es können allerdings auch neue definiert werden.

Datentyp	Javatyp	Datenbanktyp	Länge	Bemerkung
Text	java.lang.String	varchar	255	Textfeld mit 255 Zeichen
Datum	java.util.Date	date		Datumsfeld (ohne Uhrzeit). Definition eines Ausgabeformates über Pattern möglich. `dd.MM.yyyy` erzeugt z.B. 15.07.2015
Dokumentenanhang		blob	255	Näheres unter Spezielle Datentypen
Ganzzahl	java.lang.Integer	number	9	Ganzahl mit 9 Stellen
Kommazahl (9,4)	java.lang.Double	number	9	 Kommazahl mit 9 Stellen insgesamt, davon 2 Nachkommastellen.

				Definition eines Ausgabeformates über Pattern möglich. `#,##0.00` erzeugt z.B. 1.248,40
Ja/Nein	java.langBoolean	number	1	Null-Werte nicht möglich. Attribut muss ein Pflichtfeld sein.
Verschlüsselter Text		varchar	255	Näheres unter Spezielle Datentypen
Bild		blob	255	Näheres unter Spezielle Datentypen
Autonummer	java.lang.Integer	number	9	Näheres unter Spezielle Datentypen
Referenzfeld	java.lang.String	varchar	255	Referenz zu anderen Businessobjekten (s.u. 4.3 Verknüpfung zu anderen Businessobjekten)
Nachschlagefeld	java.lang.String	varchar	255	Nachschlagefeld zu anderen Businessobjekten. Unterschied zu Referenzfeld: ein Nachschlagefeld bietet eine Auswahlkomponente auf ein Referenz-BO, gespeichert wird aber nur der Text. Verwendungszweck: erleichterte Eingabe für den Anwender bei z.B. Titel, Anrede usw. Anwender soll Auswahlmöglichkeiten erhalten statt eigenen Text einzugeben.
Email	java.lang.String	varchar	255	Näheres unter Datenkomponenten
PhoneNumber	java.lang.String	varchar	255	Näheres unter Datenkomponenten
Hyperlink	java.lang.String	varchar	1000	Näheres unter Datenkomponenten
Memo	java.lang.String	varchar	4000	
Text (Large Object)	java.lang.String	clob	2147483647	
Binär (Large Object)		blob		

4.3 Verknüpfung zu anderen Businessobjekten /Referenzfeld



Der Verweis kann entweder auf ein anderes Businessobjekt oder auf einen Integrationspunkt erfolgen.

Verweis auf Integrationspunkt: Angabe des Integrationspunktes, der referenziert werden soll.

Verweis auf Businessobjekt: Angabe des BOs, das referenziert werden soll

Kaskadiert löschen: wird der Datensatz gelöscht, so wird auch der Datensatz der Referenz gelöscht

Suchfeld: im Layout wird per Default eine LOV Komponente eingebunden. Kann aber nachträglich geändert werden. Für den Webclient ist diese Einstellung nicht relevant

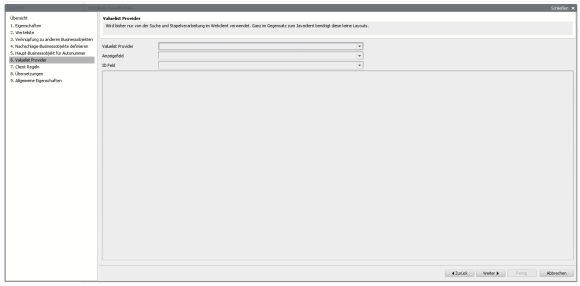
Für die Anzeige der Daten im Referenzfeld können Attribute des referenzierten BOs ausgewählt und mittels statischen Zeichen verkettet werden.

Beispiel: ${kundennummer} - ${name}

4.6 Valuelist Provider

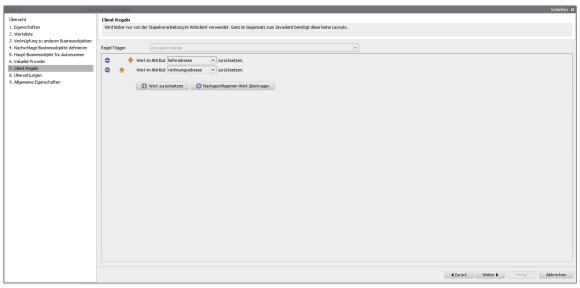
Ein Valuelist Provider kann sowohl für ein Feld innerhalb eines Layouts definiert werden, als auch attributübergreifend. In dieser Maske können sie einen Valuelist Provider auswählen, der dann für alle Auswahllisten dieses Attributes herangezogen wird.

Beispiel: ein Anwender in einer Zweigfiliale "Hamburg" darf generell nur Warengruppen auswählen, die diesen Standort betreffen.



4.7 Client Regeln

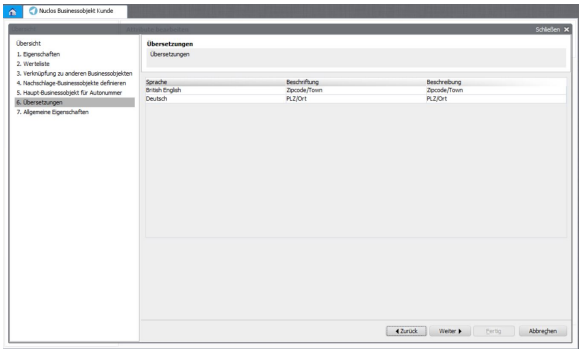
Wird bisher nur von der Stapelverarbeitung im Webclient verwendet. Ganz im Gegensatz zum Javaclient benötigt diese keine Layouts.



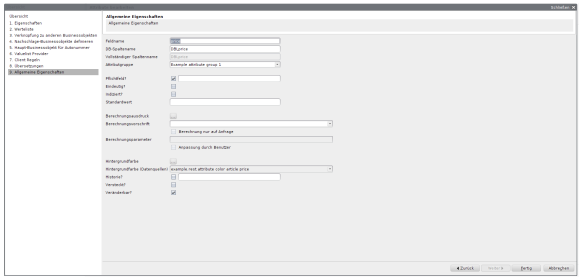
4.8 Übersetzungen

Je nachdem mit welcher Sprache sich ein Anwender im Logindialog anmeldet, wird die entsprechende Übersetzung in den Labels angezeigt. Die Beschreibung stellt im Richclient das Tooltip dar, wenn man mit der Maus über das Feld fährt.

Aktuell steht Deutsch und Englisch zur Verfügung.



4.9 Allgemeine Eigenschaften



- **Feldname:** Identifizierung des Attributs innerhalb von Regeln und im Layouteditor. Der Feldname kann nachträglich gändert werden.
- **DB-Spaltenname**: kann nachträglich geändert werden
- **Vollständiger Spaltenname:** wird vom System vergeben
- **Attributgruppe**: Zuweisung einer Attributgruppe. Wird für statusabhängige Berechtigungen verwendet
- **Pflichtfeld:** Validierung bei der Dateneingabe. Soll das Pflichtfeld nachträglich hinzugefügt werden, so muss ein Wert angegeben werden. Dieser wird bei allen bestehenden Daten eingetragen.
- **Eindeutig?:** Wert ist über alle Datensätze eindeutig. Sind mehrere Attribute eines BOs als Eindeutig markiert sind diese Felder immer in Kombination eindeutig. Soll die Eindeutigkeit nachträglich hinzugefügt werden, ist das aktuell nur möglich solange keine Daten existieren, ansonsten ist der Haken ausgegraut

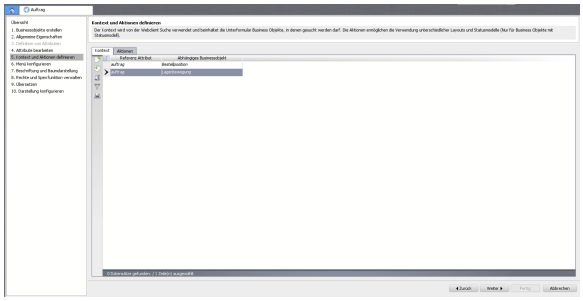
> Wenn es nicht möglich ist die Daten zu löschen, kann auch folgendermaßen vorgegangen werden. Voraussetzung: Die Eindeutigkeit der vorhandenen Daten ist sichergestellt !
>
> Nuclet in Verzeichnisstruktur exportieren
> Entsprechende .eoml Datei im Tag <unique> verändern
> Nuclet importieren.

- **Indiziert**: falls erwünscht, kann ein datenbankseitiger Index generiert werden
- **Standardwert**: wird in einem neuen Datensatz per Default eingetragen
- **Bererechnungsausdruck:** Groovy-Skriptcode für Dateninhalt. Dieser Ausdruck wird unmittelbar bei Änderungen in der Maske

ausgeführt. So erhalten die Anwender das Berechnungsergebnis sofort nach Eingabe eines Werts. Siehe dazu Regeln (clientseitig)

- **Berechnungsvorschrift:** Datenquelle oder Datenbankobjekt die immer beim Lesen des Datensatzes aus der Datenbank einen berechneten Wert liefern. Keine Aktualisierung während der Erfassung. Siehe auch Berechnete Attribute.
- **Berechnung nur auf Anfrage:** Ein so markiertes Attribut wird nicht mehr automatisch vom Layout oder von der Regel-API berechnet, das heißt, es ist auch im Layout oder in der API nicht verwendbar. Es wird nur dann berechnet und angezeigt, wenn man es in der Listenansicht einblendet. Diese Option ist für Attribute sinnvoll, deren Berechnung sehr lange dauert.
- **Berechnungsparameter:** ein Übergabeparamter für die Berechnungsfunktion (optional)
- **Anpassung durch Benutzer:** Der Benutzer hat die Möglichkeit ein berechnetes Attribut mehrfach mit verschiedenen Parametern einzublenden.
- **Hintergrundfarbe**: Groovy-Skriptcode für Hintergrundfarbe des Feldes
- **Hintergrundfarbe (Datenquelle)**: Datenquelle (Berechnetes Attribut), die die Hintergrundfarbe des Feldes als hex-kodierten String ausgibt, s.a. Berechnetes Attribut
- **Historie?**: wenn aktiviert, werden die Daten dieses Attribut historisiert
- **Versteckt?:** wenn aktiv, ist dieses Attribut für den Anwender nicht in der Spaltenauswahl sichtbar. Wird verwendet, wenn es sich um interne Attribute handelt, die für den Anwender nicht relevant sind
- **Veränderbar?:** wenn aktiviert, können Daten dieses Attributs nicht verändert werden. Diese Einstellung überregelt die Berechtigung in der Benutzergruppe bzw. Statusmodell

5. Kontext und Aktionen definieren



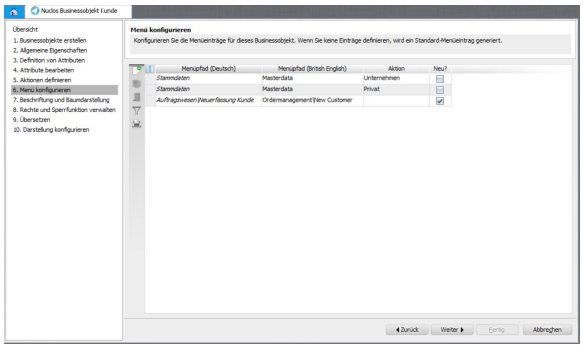
In diesem Bereich können Aktionen für das BO definiert werden. Mit dem Aktionsmechanismus hat man die Möglichkeit, für ein BO verschiedene Layouts zu gestalten oder verschiedene Prozesse (Statusmodelle) zu durchlaufen.

Anwendungsfall: die Eingabemaske für Kunde der Aktion "Unternehmen" hat Eingabefelder wie Standort, Abteilung oder USt-Identifikationsnummer, die ein Privatkunde nicht hat.

Für weitere Informationen siehe Aktionen

In diesem Bereich werden auch Kontexte für das BO definiert. Der Kontext wird von der Webclient Suche verwendet und beinhaltet die Unterformular Business Objekte, in denen gesucht werden darf.

6. Menü konfigurieren



Dieser Bereich der Menükonfiguration stellt eine Verfeinerung des Menüpfades unter Schritt 2 dar und ist optional.

Man kann Menüpfade aktionsabhängig konfigurieren oder auch die Neuanlage eines Datensatzes in einem anderen Menübereich platzieren.

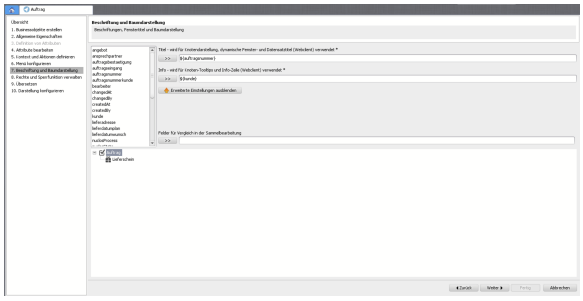
Ist der Haken "Neu?" aktiviert, so wird sofort die Eingabemaske für einen neuen Datensatz geöffnet, ohne in der Suchliste bzw. Suchmaske zu starten.

7. Beschriftung und Baumdarstellung definieren

- **Titel**: der Inhalt der definierten Attribute inkl. statischem Text werden im Fenstertitel, Historie, Baumdarstellung etc. angezeigt. Beispiel:

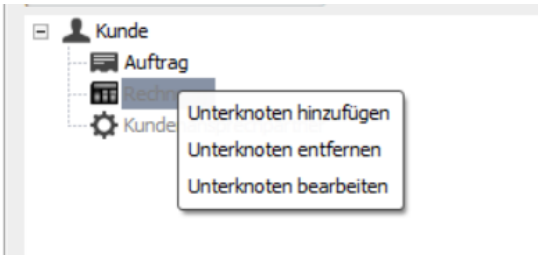
  ```
  ${kundenname}(${kundennummer}) -
  ${nuclosState}  -> Kunde AG(1001) - Aktiv
  ```

- **Info**: der Inhalt der definierten Attribute inkl. statischem Text wird im Tooltip angzeigt
- **Felder für Verlgeich in der Sammelbearbeitung**: nur relevant, wenn das BO als Unterformular eingebunden ist. Die hier angegebenen Attribute, werden in ihrer Kombination als Eindeutigkeit für die Sammelbearbeitung des Hauptdatensatzes herangezogen. Sind die Daten der hier angegebenen Felder gleich, so wird die Zeile in der Sammelbearbeitung entsprechend markiert bzw. zur Bearbeitung angeboten.
  Wenn hier nichts angegeben ist, wird die Eindeutigkeit über alle Felder bestimmt, was häufig nicht sinnvoll ist.

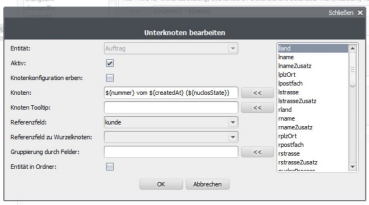


7.1 Baumdarstellung

In der Baumdarstellung können alle BOs eingebunden werden, die einen Fremdschlüssel auf das Haupt-Businessobjekt haben. Knoten können über das Kontextmenü hinzugefügt, gelöscht oder bearbeitet werden. Per Drag&Drop kann die Position der Unterknoten verändert werden. Die Unterelemente werden dann in dieser Reihenfolge angezeigt.

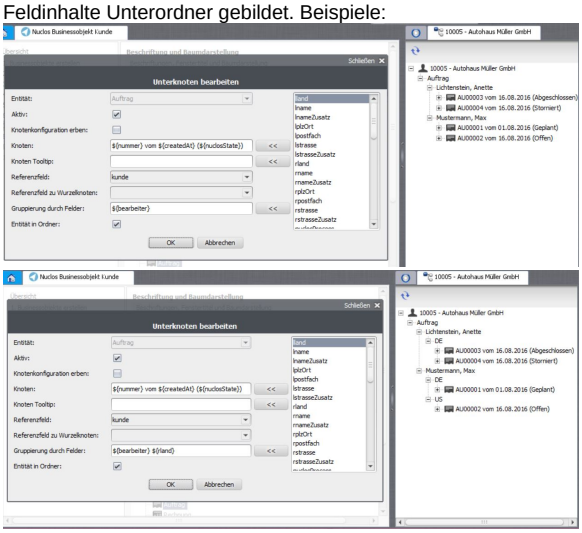


7.2 Unterknoten bearbeiten

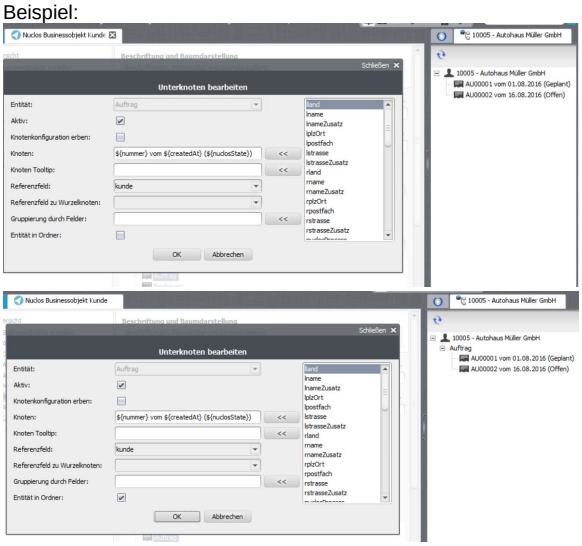


- **Entität**: Auswahl des BOs, dessen Datensätze als Unterknoten angezeigt werden sollen. Ausgewählt werden können Businessobjekt welche auf das gerade bearbeitete Businessobjekt referenzieren sowie Dynamische Businessobjekte und die generischen Dokumentenanhangsobjekte von Nuclos.
- **Aktiv**: Unterknoten aktivieren / deaktivieren. Deaktivierte Knoten werden hier grau dargestellt und sind für den Anwender nicht sichtbar.
- **Knotenkonfiguration erben**: (optional) Für die Knotenkonfiguration der Unterknoten wird die Knotenkonfiguration der in "Businessobjekt" ausgewählten Businessobjekt verwendet. Dies funktioniert nicht mit Dynamischen Businessobjekten.
- **Knoten**: (optional) Attribute und statischer Text, der im Unterknoten angezeigt wird. Wird hier nichts eingetragen wird der Wert aus dem in "Businessobjekt" ausgewählten Businessobjekt verwendet. Ist auch hier nichts eingetragen wird der lokalisierte Businessobjektname verwendet.
- **Knoten Tooltip**: (optional) Attribute und statischer Text, der als Tooltip auf dem Unterknoten angezeigt wird. Wird hier nichts eingetragen wird der Wert aus dem in "Businessobjekt" ausgewählten Businessobjekt verwendet. Ist auch hier nichts eingetragen wird der lokalisierte Businessobjektname verwendet.
- **Referenzfeld**: (optional) Attribut mit der Referenz zum Hauptknoten. Wird dieses Feld leer gelassen, werden alle Businessobjekte angezeigt.
- **Referenz zu Wurzelknoten**:(optional) Referenz aus dem in "Businessobjekt" ausgewählten Businessobjekt auf das aktuell bearbeitete Businessobjekt. Wird dieses Feld leer gelassen, wird keine Einschränkung vorgenommen.
- **Gruppierung durch Felder**: (optional) Angabe von einem oder mehreren Attributen mit Leerzeichen getrennt. In der angegebenen Reihenfolge werden je nach Gleichheit der

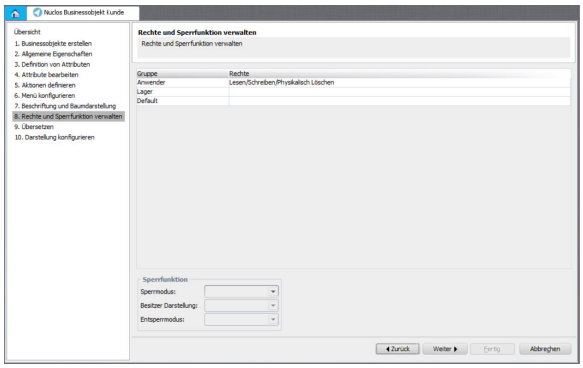
Feldinhalte Unterordner gebildet. Beispiele:



- **Entität in Ordner**: ist dieser Haken aktiviert, so wird das BO als übergeorneter Ordner angezeigt.
Beispiel:



8. Rechte und Sperrfunktion verwalten

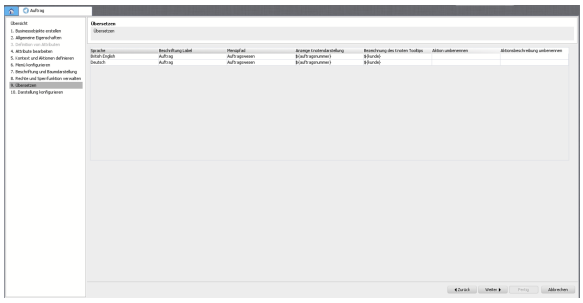


In diesem Bereich werden alle existierenden Benutzergruppen aufgelistet. Die entsprechenden Berechtigungen für dieses BO können an dieser Stelle gepflegt werden.

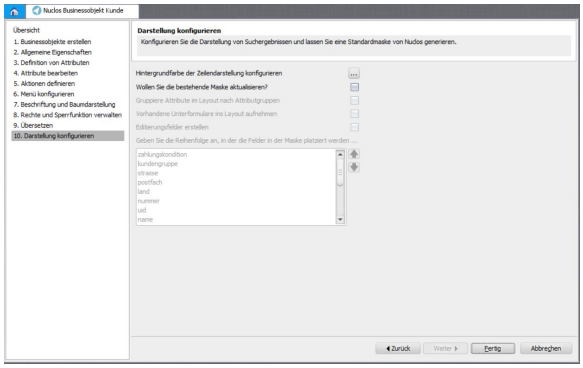
Desweiteren kann dieses BO für die Sperrfunktion aktiviert und entsprechend konfiguriert werden. Details siehe Sperrlogik.

9. Übersetzen

Für das Businessobjekt und den Menüpfad können Übersetzungen gepflegt werden,
ebenso die Knotendarstellung und das Tooltip sowie Aktionen können angepasst werden.



10. Darstellung konfigurieren



Für jedes Businessobjekt kann man automatisch ein Layout generieren lassen bzw. auch ein vorhandenes Layout aktualisieren.

> Das bereits bestehende Layout wird dabei überschrieben !

Hintergrundfarbe: Groovy-Skript für die Farbe der Zeilendarstellung. Gilt für Suchergebnislisten und für Unterformulare. Siehe dazu Regeln (clientseitig)

Maske aktualisieren ?: ist dieser Haken aktiviert, so wird das Layout neu erzeugt bzw. das Vorhandene überschrieben

Gruppierung: die einzelnen Attribute jeder Attributgruppe werden in einem Panel zusammengefasst

Unterformulare: jedes BO mit einem Fremdschlüssel auf dieses BO wird im Layout als Unterformular eingefügt.

Editierungsfelder: CreatedAt, CreatedBy, ChangedAt und ChangedBy werden im Layout eingefügt

Reihenfolge: hier wird die Reihenfolge der Attribute im Layout definiert. Die Attribute werden alle untereinander in einem Panel angeordnet.