

XML Strukturdefinition

- [Definition](#)
- [Import von Daten aus XML Dateien](#)
 - [XML Datei Import](#)
 - [Einschränkungen des XML Imports](#)

Definition

Menüaufruf: (Konfiguration) - (Import & Export) - (XML Strukturdefinition)

Der Import von Daten aus XML orientiert sich sehr stark an dem [Strukturdefinition](#). Daher wird hier nur auf die Unterschiede eingegangen. Insbesondere gibt es **XML Strukturdefinitionen** und den eigentlichen **XML Datei Import**.

Die XML Strukturdefinition beschreibt für **ein Business Object** (BO), wie die XML Inhalte aus der gegebenen XML Datei auf die Attribute (Felder) übertragen werden (*Mapping*). Die Strukturdefinition benötigt die folgenden Angaben:

Import von Daten aus XML Dateien

Match (Struktur)

Einfacher XPath Ausdruck, der die Attributverarbeitung (s.u.) der Strukturdefinition triggert.

Businessobjekt

Das BO, das durch diese Strukturdefinition erzeugt (aktualisiert/gelöscht) wird.

Attribute

Jede Zeile definiert, wie ein Attribut (Feld) des BOs aus dem XML extrahiert werden soll. Die einzelnen Spalten bedeuten:

Match (Attribut)

Einfacher XPath Ausdruck (**relativ zum Struktur Match**), der den XML Knoten (Element bzw. Attribute) auswählt, dessen Wert auf das BO Attribut übertragen wird. Details weiter unten.

Attribut (BO)

BO Attribut (Feld), in den der aus dem XML extrahierte Wert eingetragen wird.

Berechnungsausdruck

Über das Feld **Berechnungsausdruck** kann ein zu importierendes Attribut dynamisch berechnet werden. Dies ist z.B. notwendig, wenn der Wert *eines* BO Attributes aus *mehreren* XML Knoten-Werten zusammen gesetzt werden muss. Über ein Groovy-Script können Sie den zu importierenden Wert definieren. Dabei stehen (im Script) die Variablen `xpath2value` (Map<String, String>), `line` (java.lang.Integer, die aktuelle Zeilennummer in der XML Datei) und `log` (org.nuclos.server.fileimport.ImportLogger) zur Verfügung.

Die Variable `xpath2value` ist besonders interessant. Sie enthält zu den relativen XPath Ausdrücken (als Schlüssel) die Werte der dadurch ausgewählten XML Knoten (als Wert). (Beispiel weiter unten.)

Format (Parsestring)

Verhält sich genau wie beim CSV Import (weitere Informationen dort).



Format wird nur verwendet, wenn *kein* Berechnungsausdruck (Script) hinterlegt ist.

XML Datei Import

Der XML Datei Import definiert, in welcher Reihenfolge die Strukturdefinitionen abgearbeitet werden. Der eigentliche Import geht dann wie folgt vor sich:

1. Für jede Strukturdefinition wird die XML Datei einmal geöffnet (gelesen) und deren Inhalt von Anfang bis Ende verarbeitet.
2. Findet die Strukturdefinition den in ihr festgelegten *Struktur Match*, so wird der hier eingeschlossene XML (Teil-)Baum vollständig gelesen, um eine *Map* zu bilden, die den relativen XPath Ausdruck (bezüglich des Struktur Matches) dem Wert des Elements/Attributes zuordnet.
3. Die Attribute werden aus dieser *Map* gewonnen, indem der Attribut Match als Schlüssel zur *Map* verwendet wird. Der ausgelesene Wert wird als Wert für das angegebene Attribut (des BOs) verwendet.
4. (Optional.) Falls ein Berechnungsausdruck angegeben wird, steht in dem zugehörigen Script diese *Map* als Variable `xpath2value` zur Verfügung.
5. (Optional.) *Genau ein* Attribut Match darf auch 'nach außen greifen', d.h. dieses kann durch Werte vom XML Root Element bis zum Parent Element des *Struktur Matches* befüllt werden. Erlaubt sind in diesem Fall aber nur Attribute und direkte Kind Elemente der Eltern Elemente des *Struktur Matches*.

Einschränkungen des XML Imports

Aus dem gesagten ergeben sich folgende Einschränkungen des XML Imports:

- Die XML Datei muss *well-formed* sein. Ein XML Schema wird *nicht* benötigt (und kann auch nicht angegeben werden)
- Der XML Import von Dateien mit *mixed content* (d.h. in einem Tag können sich gleichzeitig Text und Unterelemente befinden) ist *nicht* vorgesehen. Anders gesagt: in jedem XML Element befindet sich *entweder* Text *oder* weitere XML Elemente (Tags).
- Die Inhalte für ein BO müssen sich im XML in einem zusammenhängenden Teilbaum befinden. Diese Teilbäume müssen für den Import komplett in den Speicher geladen werden; sie sollten also möglichst klein sein.

Der XML Import ist daher nicht für sehr tief verschachtelte XML Strukturen geeignet, speziell wenn die Attribute eines BOs sehr 'weit' voneinander entfernt sind.

Warum ist der XML Import *fern* von XML und so *nah* am CSV Import?

Das liegt an Einschränkungen im Nuclös Kern, insbesondere an:

- Kein Support für *rekursive* Datenstrukturen
- Ein BO kann nur *einmal* als Unterformular in einem anderen BO angezeigt werden. Es ist nicht möglich, dasselbe BO in mehreren (unterschiedlichen) Funktionen als Unterformular in ein anderes BO einzubetten.
- Nur eingeschränkter Support für Unterformulare von Unterformularen.

Diese Einschränkungen führen dazu, dass es nicht einfach möglich ist, eine (normale) XML Struktur mit Nuclös Bordmitteln auf der Metaebene (also auf der Ebene, wo sich z.B. XML Schema befindet) zu beschreiben.