



ADOIT
Enterprise Architecture Suite

ADOIT[®]

Version 18.1

Hilfe zur Excel-Schnittstelle



Adressen der BOC Gruppe:

Hotline: <https://support.boc-group.com>

Homepage: <https://www.boc-group.com>

© Benutzerdokumentation für ADOIT® 2025

Weitergehende Veröffentlichung oder Berichterstattung sowie Nachdruck und Vervielfältigungen - gleich in welcher Form, ganz oder teilweise - bedarf der schriftlichen Genehmigung der BOC Products & Services AG.

Die ADOIT® 18.1 Hilfe zur Excel-Schnittstelle

© 2025, BOC Gruppe

Inhaltsverzeichnis

I Willkommen in der Hilfe zur Excel-Schnittstelle	1
1 Aufbau der Excel-Datei	2
2 Konfiguration erstellen	3
2.1 <config>	3
2.2 <sheet>	3
2.2.1 Eindeutige Objektidentifizierung mittels ID	4
2.2.2 Beispiel	4
2.3 <attribute>	5
2.3.1 <attribute type="simple">	6
2.3.1.1 Beispiel	6
2.3.2 <attribute type="date">	7
2.3.2.1 Beispiel	8
2.3.3 <attribute type="enum">	9
2.3.3.1 domain_mapping	9
2.3.3.2 Beispiel	10
2.3.4 <attribute type="enum_list">	11
2.3.5 <attribute type="treeenumlist">	12
2.3.5.1 Beispiel	12
2.3.6 <attribute type="bool">	13
2.3.6.1 Beispiel	14
2.3.7 <attribute type="relation">	15
2.3.7.1 Eingehende Beziehungen vs. ausgehende Beziehungen	16
2.3.7.2 Beispiel 1	16
2.3.7.3 Beispiel 2	17
2.3.8 <attribute type="file_pointer">	18
2.3.8.1 Erreichbarkeit von Links zu Dateien	18
2.3.8.2 Beispiel	18
3 Excel-Import durchführen	20
3.1 Konfiguration importieren	20
3.2 Excel-Datei als Vorlage importieren	20
3.3 Objekte aus Excel in ADOIT importieren	20
3.4 Nach dem Import	21
II Appendix	22
4 Konfliktmanagement beim Import	23
5 Sprachunabhängige Namen von Metamodellelementen anzeigen	24
Index	25

I Willkommen in der Hilfe zur Excel-Schnittstelle

In ADOIT ist eine konfigurierbare Excel-Schnittstelle zur schnellen Datenerfassung integriert. Über die Excel-Schnittstelle können Sie Repository-Objekte mit ihren Attributen und Relationen aus einer Excel-Datei importieren und aktualisieren. Für diesen Prozess wird die Struktur der Excel-Datei in einer XML-Konfigurationsdatei beschrieben.

In diesem Teil des Handbuchs machen wir Sie mit den wichtigsten Grundlagen zur Verwendung der Excel-Schnittstelle vertraut. Folgende Themenkreise werden behandelt:

- **[Aufbau der Excel-Datei](#)**
Eine Übersicht über die Struktur einer Excel-Datei zum Import von Objekten
- **[Konfiguration erstellen](#)**
Eine Anleitung zur Erstellung einer XML-Konfigurationsdatei, die die Zuordnung von Objekten aus der Excel-Datei zum ADOIT Metamodell enthält
- **[Excel-Import durchführen](#)**
Eine Anleitung zum Importieren von Objekten aus Excel



Um erfolgreich eine Konfiguration zu erstellen, müssen Sie Zugriff auf die Metamodellverwaltung haben. Die [sprachunabhängigen Namen von Objekttypen, Attributen usw. sind dort zu finden](#) und werden für die Konfiguration benötigt.

1 Aufbau der Excel-Datei

Eine Excel-Datei zum Import von Repository-Objekten ist folgendermaßen aufgebaut:

	A	B	C	D	E	F	G
1	Anwendung 1	29.03.2016	Kein Eintrag	AS-IS	ja	Schnittstelle 1;Schnittstelle 2	
2	Anwendung 2	30.03.2016	Kein Eintrag	AS-IS;TO-BE	nein	Schnittstelle 3;Schnittstelle 4	
3	Anwendung 3	31.03.2016	In Entwicklung	PLAN;TO-BE	ja	Schnittstelle 6	
4	Anwendung 4	01.04.2016	Stillgelegt	DECOMMISSIONED	nein		
5	Anwendung 5	01.05.2016	Stillgelegt	DECOMMISSIONED	ja	Schnittstelle 7	
6	Anwendung 6	01.06.2016	In Produktion	AS-IS	nein		
7	Anwendung 7	01.07.2016	Entwurf	TO-BE	nein		
8							
9							

Abb. 1: Beispiel einer Excel-Datei mit Objekten und ihren Attributen

- Die Excel-Datei kann aus beliebig vielen Tabellenblättern bestehen
- Jedes Tabellenblatt enthält nur Objekte eines Typs (1)
- Jede Zeile des Tabellenblatts enthält ein Objekt (2)
- Jede Spalte enthält ein Objektattribut oder eine Relation zu einem anderen Objekt (3)
- Eine eindeutige Kennung (Name, ID etc.) wird für jedes Objekt benötigt
- Folgende Attributtypen können importiert werden: `simple`, `date`, `enum`, `treenumlist`, `enum_list`, `bool`, `relation` und `file_pointer`

Wenn Sie die ArchiMate-Anwendungsbibliothek verwenden, sind bereits eine oder mehrere Beispielkonfigurationen in der ADOIT Administration verfügbar. Zu jeder Beispielkonfiguration ist eine passende Excel-Datei als Vorlage hinterlegt. Die Vorlage steht im Excel Import-Dialog in ADOIT zum Download bereit.

2 Konfiguration erstellen

Eine XML-Konfigurationsdatei enthält die Zuordnung von Objekten aus der Excel-Datei zum ADOIT Metamodell.

2.1 <config>

Die XML-Datei startet und endet mit einem <config>-Tag:

XML-Beispiel

```
<config>
```

```
...
```

```
</config>
```

2.2 <sheet>

In der XML-Datei wird jedes aus der Excel-Datei zu importierende Tabellenblatt in einem eigenständigen Tag <sheet> beschrieben. Folgende Attribute müssen je <sheet> definiert werden:

Attributname	Beschreibung	Beispielwerte
name	Name des Tabellenblatts in Excel	Anwendungen
class_name	Sprachunabhängiger Name des Objekttyps	[C_APPLICATION]
id	Legt die Nummer der Spalte fest, die das Objekt eindeutig identifiziert. [Optional] Kommaseparierte Liste von Spaltennummern zur eindeutigen Identifizierung des Objekts.	1
data_row	Gibt die erste Zeile an, die ein Objekt enthält	1
unique_name [Optional]	Auf true setzen, damit es nicht möglich ist, zwei Objekte desselben Typs mit	[true false]

	demselben Namen im gleichen Sprachkontext zu erstellen	
--	--	--

2.2.1 Eindeutige Objektidentifizierung mittels ID

Das Feld `id` definiert, welches Objektattribut in der Excel-Datei ein Objekt eindeutig identifiziert.

Typische Attribute für die eindeutige Objektidentifikation sind das Attribut „Name“ oder dedizierte „ID“-Attribute. Die Verwendung eines dedizierten „ID“-Attributs bietet den Vorteil, dass Objekte im Lauf der Zeit in Drittsystemen umbenannt werden können, beim Import jedoch weiter als identische Objekte identifiziert werden.

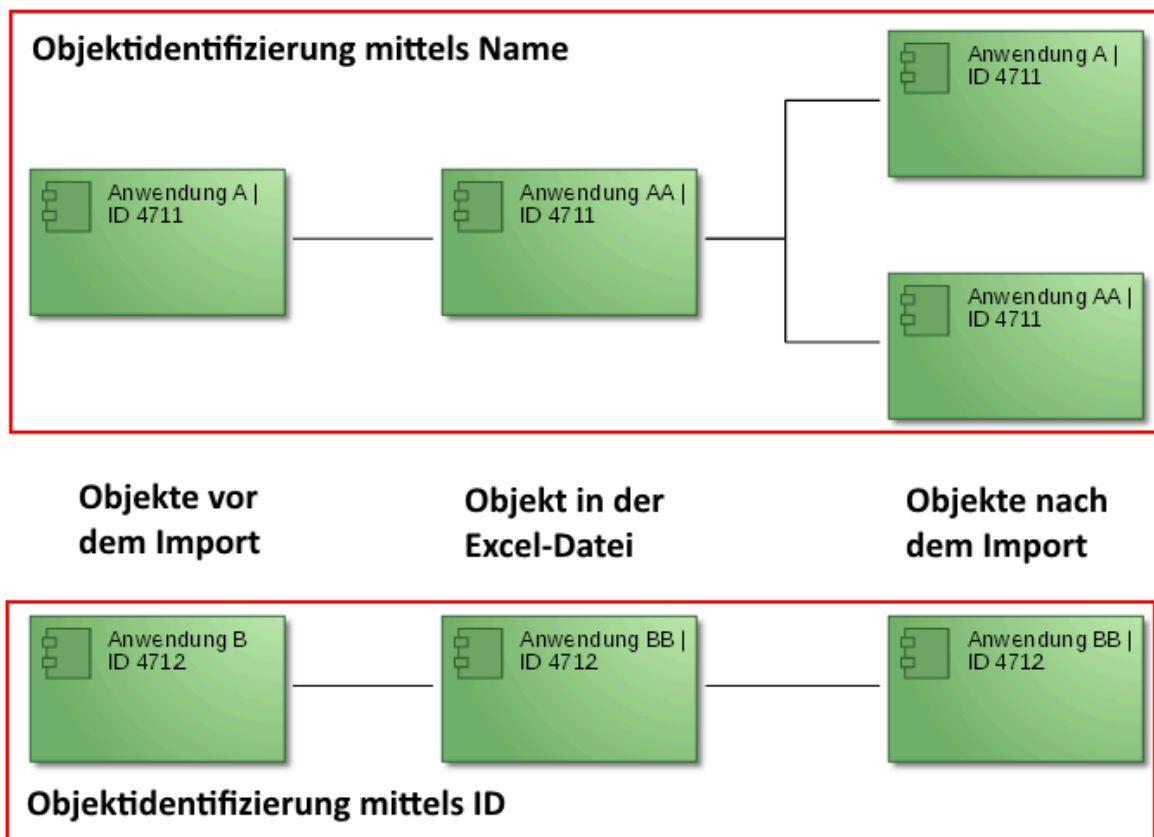


Abb. 2: Eindeutige Objektidentifizierung mittels „ID“

2.2.2 Beispiel

Sie wollen drei Tabellenblätter importieren, die Objekte vom Typ *Anwendung*, *Schnittstelle* und *Dokument* enthalten:

	A	B	C	D	E	F	G
1	Anwendung 1	29.03.2016	Kein Eintrag	AS-IS	ja	Schnittstelle 1;Schnittstelle 2	
2	Anwendung 2	30.03.2016	Kein Eintrag	AS-IS;TO-BE	nein	Schnittstelle 3;Schnittstelle 4	
3	Anwendung 3	31.03.2016	In Entwicklung	PLAN;TO-BE	ja	Schnittstelle 6	
4	Anwendung 4	01.04.2016	Stillgelegt	DECOMMISSIONED	nein		
5	Anwendung 5	01.05.2016	Stillgelegt	DECOMMISSIONED	ja	Schnittstelle 7	
6	Anwendung 6	01.06.2016	In Produktion	AS-IS	nein		
7	Anwendung 7	01.07.2016	Entwurf	TO-BE	nein		
8							
9							

Abb. 3: Import eines Tabellenblatts

Hier ist ein Auszug aus der XML-Datei mit den Abschnitten zum Import der Tabellenblätter:

XML- Beispiel

```

<config>
<sheet name="Anwendungen" class_name="C_APPLICATION" id="1" data_row="1">
...
</sheet>
<sheet name="Schnittstellen" class_name="C_INTERFACE" id="1" data_row="1">
...
</sheet>
<sheet name="Dokumente" class_name="C_DOCUMENT" id="1" data_row="1">
...
</sheet>
</config>
    
```

2.3 <attribute>

In der XML-Konfigurationsdatei wird jedes aus der Excel-Datei zu importierende Objektattribut in einem eigenständigen Tag <attribute> beschrieben.

Folgende Attributtypen können importiert werden:

Attributtyp	Beschreibung
simple	Erhebung von Text- und Zahlenwerten
date	Erhebung von Datumsangaben
enum	Übertragung eines Wertes auf ein Element einer Liste

<code>enum_list</code>	Übertragung von Werten auf die Elemente einer Liste
<code>treeenumlist</code>	Übertragung von Werten auf die Knoten eines Baums
<code>bool</code>	Erhebung von booleschen Werten
<code>relation</code>	Erhebung von Beziehungen zu anderen Objekten
<code>file_pointer</code>	Erhebung zu Links auf externe Dateien

2.3.1 <attribute type="simple">

Über den Attributtyp `simple` können Text- und Zahlenwerte importiert werden. Die importierten Werte können im ADOIT Metamodell folgenden Attributtypen zugeordnet werden:

- ADONIS Text (ADOSTRING)
- Ganzzahl (INTEGER)
- Gleitkommazahl (DOUBLE)
- Kurztext (SHORTSTRING)
- Langtext (LONGSTRING)
- Positive Ganzzahl (UNSIGNED INTEGER)
- Text (STRING)

Dazu müssen folgende Attribute für das Tag `<attribute>` definiert werden:

Attributtyp	Beschreibung
<code>name</code>	Sprachunabhängiger Name des Objektattributs, in das der importierte Wert geschrieben werden soll.
<code>type</code>	Angabe des Attributtyps – hier <code>simple</code>
<code>context</code>	Angabe der Sprache des Attributwerts. Mögliche Werte sind z.B. „de“ für Deutsch und „en“ für Englisch. Die verfügbaren Sprachen hängen von der Anwendungsbibliothek und der Lizenz ab.
<code>column</code>	Angabe der Spaltennummer des Objektattributs in der Excel-Datei

2.3.1.1 Beispiel

Sie wollen das Objektattribut *Name* von Objekten des Typs *Anwendung* importieren:

	A	B	C	D	E	F	G
1	Anwendung 1	29.03.2016	Kein Eintrag	AS-IS	ja	Schnittstelle 1;Schnittstelle 2	
2	Anwendung 2	30.03.2016	Kein Eintrag	AS-IS;TO-BE	nein	Schnittstelle 3;Schnittstelle 4	
3	Anwendung 3	31.03.2016	In Entwicklung	PLAN;TO-BE	ja	Schnittstelle 6	
4	Anwendung 4	01.04.2016	Stillgelegt	DECOMMISSIONED	nein		
5	Anwendung 5	01.05.2016	Stillgelegt	DECOMMISSIONED	ja	Schnittstelle 7	
6	Anwendung 6	01.06.2016	In Produktion	AS-IS	nein		
7	Anwendung 7	01.07.2016	Entwurf	TO-BE	nein		
8							
9							

Abb. 4: Import eines simple-Attributs

Hier ist ein Auszug aus der XML-Datei mit dem relevanten Abschnitt:

XML- Beispiel

```
<config>
<sheet name="Anwendungen" class_name="C_APPLICATION" id="1" data_row="1">
<attribute name="NAME" type="simple" context="de" column="1"/>
...
</sheet>
...
</config>
```

2.3.2 <attribute type="date">

Über den Attributtyp `date` können Datumswerte importiert werden. Die importierten Werte können im ADOIT Metamodell folgenden Attributtypen zugeordnet werden:

- Date (DATE)
- Coordinated Universal Time (UTC)

Datumswerte werden alle als 00:00 UTC importiert. Um sicherzustellen, dass alle kundenspezifischen geplanten Jobs auf Basis der Ortszeit ausgeführt werden, können Sie eine Differenz zur UTC konfigurieren.

Folgende Attribute müssen für das Tag `<attribute>` definiert werden:

Attributtyp	Beschreibung
<code>name</code>	Sprachunabhängiger Name des Objektattributs, in das der importierte Wert geschrieben werden soll.
<code>type</code>	Angabe des Attributtyps – hier <code>date</code>

context	Angabe der Sprache des Attributwerts. Mögliche Werte sind z.B. „de“ für Deutsch und „en“ für Englisch. Die verfügbaren Sprachen hängen von der Anwendungsbibliothek und der Lizenz ab.
column	Angabe der Spaltennummer des Objektattributs in der Excel-Datei. Diese Spalte sollte in der Excel-Datei als Datum formatiert sein.
utc_offset_hours [Optional]	Angabe der Differenz in Stunden der Ortszeit zur koordinierten Weltzeit (UTC), z.B. "1" für die mitteleuropäische Zeit (MEZ) oder "-5" für die Eastern Standard Time (EST).
utc_offset_minutes [Optional]	Angabe der Differenz in Minuten der Ortszeit zur koordinierten Weltzeit (UTC), z.B. "30" für die Indian Standard Time (IST) für eine Gesamtdifferenz von +05:30 (wenn utc_offset_hours auf "5" eingestellt ist).

2.3.2.1 Beispiel

Sie wollen das Objektattribut *Datum Produktivsetzung* von Objekten des Typs *Anwendung* importieren:

	A	B	C	D	E	F	G
1	Anwendung 1	29.03.2016	Kein Eintrag	AS-IS	ja	Schnittstelle 1;Schnittstelle 2	
2	Anwendung 2	30.03.2016	Kein Eintrag	AS-IS;TO-BE	nein	Schnittstelle 3;Schnittstelle 4	
3	Anwendung 3	31.03.2016	In Entwicklung	PLAN;TO-BE	ja	Schnittstelle 6	
4	Anwendung 4	01.04.2016	Stillgelegt	DECOMMISSIONED	nein		
5	Anwendung 5	01.05.2016	Stillgelegt	DECOMMISSIONED	ja	Schnittstelle 7	
6	Anwendung 6	01.06.2016	In Produktion	AS-IS	nein		
7	Anwendung 7	01.07.2016	Entwurf	TO-BE	nein		
8							
9							

Abb. 5: Import eines date-Attributs

Hier ist ein Auszug aus der XML-Datei mit dem relevanten Abschnitt:

```

XML-Beispiel
<config>
<sheet name="Anwendungen" class_name="C_APPLICATION" id="1" data_row="1">
...
<attribute name="A_VALID_FROM" type="date" context="de" column="2"/>
...
</sheet>
    
```

```
...
</config>
```

2.3.3 <attribute type="enum">

Über den Attributtyp `enum` können Attributwerte importiert und den Elementen einer Liste zugewiesen werden. Die importierten Werte können im ADOIT Metamodell folgendem Attributtyp zugeordnet werden:

- Enumeration (ENUM)

Ein ENUM-Attribut ist eine Liste mit vordefinierten Werten (z.B. "Niedrig", "Mittel" und "Hoch"). Nur ein Wert kann aus der Liste ausgewählt werden.

Sie können frei definieren, welcher Wert aus der Excel-Datei auf welchen Wert in der Liste übertragen wird. Dazu müssen folgende Attribute für das Tag `<attribute>` definiert werden:

Attributtyp	Beschreibung
<code>name</code>	Sprachunabhängiger Name des Objektattributs, in das der importierte Wert geschrieben werden soll.
<code>type</code>	Angabe des Attributtyps – hier <code>enum</code>
<code>context</code>	Angabe der Sprache des Attributwerts. Mögliche Werte sind z.B. „de“ für Deutsch und „en“ für Englisch. Die verfügbaren Sprachen hängen von der Anwendungsbibliothek und der Lizenz ab.
<code>column</code>	Angabe der Spaltennummer des Objektattributs in der Excel-Datei
<code>domain_mapping</code>	Definition des Mappings zwischen den Werten in der Excel-Datei und den Elementen der Liste (genauere Informationen finden Sie unterhalb dieser Tabelle).
<code>separator_domain_mapping</code> [Optional]	Definiert das Trennzeichen zwischen mehreren Werten im Attribut <code>domain_mapping</code> . Wenn die Werte in der Excel-Datei z.B. als "Kein Eintrag@v0,..." erfasst sind, ist der Wert des Trennzeichens "@".

2.3.3.1 domain_mapping

Jedes Element einer Liste in ADOIT hat einen sprachunabhängigen Namen – z.B. `v0`, `v1`, `v2` etc. Die Werte in der Excel-Datei müssen den sprachunabhängigen Namen der Elemente

der Liste zugeordnet werden. Dies erfolgt über das Attribut `domain_mapping`. Hierbei wird kommasepariert jedem Wert aus der Excel-Datei ein zugehöriger Wert der Liste zugeordnet:

```
Domain_mapping="<Excel-Wert 0>,v0,<Excel-
-Wert 1>,v1,<Excel-Wert 2>,v2,..."
```



Die Werte aus der Excel-Datei und die *sprachabhängigen* Werte des ENUM-Attributs müssen nicht gleich sein. Sie können z.B. den Excel-Wert "Rosarot" in der Liste dem vordefinierten Wert "Rot" zuordnen.

2.3.3.2 Beispiel

Sie wollen das Objektattribut *Lebenszyklusstatus* von Objekten des Typs *Anwendung* importieren:

	A	B	C	D	E	F	G
1	Anwendung 1	29.03.2016	Kein Eintrag	AS-IS	ja	Schnittstelle 1;Schnittstelle 2	
2	Anwendung 2	30.03.2016	Kein Eintrag	AS-IS;TO-BE	nein	Schnittstelle 3;Schnittstelle 4	
3	Anwendung 3	31.03.2016	In Entwicklung	PLAN;TO-BE	ja	Schnittstelle 6	
4	Anwendung 4	01.04.2016	Stillgelegt	DECOMMISSIONED	nein		
5	Anwendung 5	01.05.2016	Stillgelegt	DECOMMISSIONED	ja	Schnittstelle 7	
6	Anwendung 6	01.06.2016	In Produktion	AS-IS	nein		
7	Anwendung 7	01.07.2016	Entwurf	TO-BE	nein		
8							
9							

Abb. 6: Import eines enum-Attributs

Hier ist ein Auszug aus der XML-Datei mit dem relevanten Abschnitt:

XML-Beispiel

```
<config>
<sheet name="Anwendungen" class_name="C_APPLICATION" id="1" data_row="1">
...
<attribute name="A_LIFECYCLE_STATE" type="enum" context="de" column="3"
domain_mapping="Kein Eintrag,v0,Entwurf,v1,In Entwicklung,v2,In
Produktion,v3,Stillgelegt,v4"/>
...
</sheet>
...
</config>
```

2.3.4 <attribute type="enum_list">

Über den Attributtyp `enum_list` können Attributwerte importiert und den Elementen einer Liste zugewiesen werden. Die importierten Werte können im ADOIT Metamodell folgendem Attributtyp zugeordnet werden:

- Aufzählungsliste (ENUMLIST)

Ein ENUMLIST ist eine Liste mit vordefinierten Werten (z.B. "Niedrig", "Mittel" und "Hoch"). Im Gegensatz zu Attributen vom Typ ENUM können gleichzeitig mehrere Werte aus der Liste ausgewählt werden.

Sie können frei definieren, welcher Wert aus der Excel-Datei auf welchen Wert in der Liste übertragen wird. Dazu müssen folgende Attribute für das Tag `<attribute>` definiert werden:

Attributtyp	Beschreibung
<code>name</code>	Sprachunabhängiger Name des Objektattributs, in das der importierte Wert geschrieben werden soll.
<code>type</code>	Angabe des Attributtyps – hier <code>enum_list</code>
<code>context</code>	Angabe der Sprache des Attributwerts. Mögliche Werte sind z.B. „de“ für Deutsch und „en“ für Englisch. Die verfügbaren Sprachen hängen von der Anwendungsbibliothek und der Lizenz ab.
<code>column</code>	Angabe der Spaltennummer des Objektattributs in der Excel-Datei
<code>separator</code>	Definiert das Trennzeichen zwischen mehreren Werten in der Excel-Datei, die der Liste zugeordnet werden sollen. So können Sie dem ENUMLIST-Attribut gleichzeitig mehrere Werte zuordnen. Wenn die Werte in der Excel-Datei z.B. als "Variante 1;Variante 2" erfasst sind, ist der Wert des Trennzeichens ";".
<code>domain_mapping</code>	Definition des Mappings zwischen den Werten in der Excel-Datei und den Elementen der Liste (siehe <attribute type='enum'> für genauere Informationen).
<code>separator_domain_mapping</code> [Optional]	Definiert das Trennzeichen zwischen mehreren Werten im Attribut <code>domain_mapping</code> . Wenn die Werte in der Excel-Datei z.B. als "Kein Eintrag@v0,..." erfasst sind, ist der Wert des Trennzeichens "@".

2.3.5 <attribute type="treeenumlist">

Über den Attributtyp `treeenumlist` können Attributwerte importiert und den Knoten eines Baumes zugewiesen werden. Die importierten Werte können im ADOIT Metamodell folgendem Attributtyp zugeordnet werden:

- `ENUMLIST_TREE`

Attribute vom Typ `ENUMLIST_TREE` sind Bäume mit vordefinierten Werten (z.B. "Variante 1", "Variante 2" und "Variante 3"). Es können gleichzeitig mehrere Werte aus dem Baum ausgewählt werden. Der Kategorienfilter in ADOIT ermöglicht das Filtern von Objekten nach diesen Werten (= Kategorien).

Sie können frei definieren, welcher Wert aus der Excel-Datei auf welchen Knoten übertragen wird. Dazu müssen folgende Attribute für das Tag `<attribute>` definiert werden:

Attributtyp	Beschreibung
<code>name</code>	Sprachunabhängiger Name des Objektattributs, in das der importierte Wert geschrieben werden soll.
<code>type</code>	Angabe des Attributtyps – hier <code>treeenumlist</code>
<code>context</code>	Angabe der Sprache des Attributwerts. Mögliche Werte sind z.B. „de“ für Deutsch und „en“ für Englisch. Die verfügbaren Sprachen hängen von der Anwendungsbibliothek und der Lizenz ab.
<code>column</code>	Angabe der Spaltennummer des Objektattributs in der Excel-Datei
<code>separator</code>	Definiert das Trennzeichen zwischen mehreren sprachunabhängigen Namen der Baumknoten in einer Zelle der Excel-Datei. So können Sie dem „ENUMLIST_TREE“-Attribut gleichzeitig mehrere Werte zuordnen. Wenn die Werte in der Excel-Datei z.B. als "PLAN;TO-BE" erfasst sind, ist der Wert des Trennzeichens ";".

2.3.5.1 Beispiel

Sie wollen das Objektattribut *Varianten* von Objekten des Typs *Anwendung* importieren:

	A	B	C	D	E	F	G
1	Anwendung 1	29.03.2016	Kein Eintrag	AS-IS	a	Schnittstelle 1;Schnittstelle 2	
2	Anwendung 2	30.03.2016	Kein Eintrag	AS-IS;TO-BE	nein	Schnittstelle 3;Schnittstelle 4	
3	Anwendung 3	31.03.2016	In Entwicklung	PLAN;TO-BE	a	Schnittstelle 6	
4	Anwendung 4	01.04.2016	Stillgelegt	DECOMMISSIONED	nein		
5	Anwendung 5	01.05.2016	Stillgelegt	DECOMMISSIONED	a	Schnittstelle 7	
6	Anwendung 6	01.06.2016	In Produktion	AS-IS	nein		
7	Anwendung 7	01.07.2016	Entwurf	TO-BE	nein		
8							
9							

Abb. 7: Import eines `treeenumlist`-Attributs

Hier ist ein Auszug aus der XML-Datei mit dem relevanten Abschnitt:

```

XML-Beispiel
<config>
<sheet name="Anwendungen" class_name="C_APPLICATION" id="1" data_row="1">
...
<attribute name="A_ARCHITECTURE_TREE" type="treeenumlist" context="de"
column="4" separator=";" />
...
</sheet>
...
</config>
    
```

2.3.6 <attribute type="bool">

Über den Attributtyp `bool` können boolesche Werte importiert werden ("TRUE" (wahr) und "FALSE" (falsch)). Die importierten Werte können im ADOIT Metamodell folgendem Attributtyp zugeordnet werden:

- Bool (BOOL)

Dazu müssen folgende Attribute für das Tag `<attribute>` definiert werden:

Attributtyp	Beschreibung
<code>name</code>	Sprachunabhängiger Name des Objektattributs, in das der importierte Wert geschrieben werden soll.
<code>type</code>	Angabe des Attributtyps – hier <code>bool</code>
<code>context</code>	Angabe der Sprache des Attributwerts. Mögliche Werte sind z.B. „de“ für Deutsch und „en“ für

	Englisch. Die verfügbaren Sprachen hängen von der Anwendungsbibliothek und der Lizenz ab.
column	Angabe der Spaltennummer des Objektattributs in der Excel-Datei
bool_mapping	Definition des Mappings zwischen den Werten der Excel-Datei und den Werten des Objektattributs "TRUE" und "FALSE". Das Attribut enthält zwei Werte, die durch ein Komma getrennt sind. Der erste Wert wird auf den booleschen Wert "TRUE", der zweite auf "FALSE" gemappt.

2.3.6.1 Beispiel

Sie wollen das Objektattribut *Handlungsbedarf* von Objekten des Typs *Anwendung* importieren:

	A	B	C	D	E	F	G
1	Anwendung 1	29.03.2016	Kein Eintrag	AS-IS	ja	Schnittstelle 1;Schnittstelle 2	
2	Anwendung 2	30.03.2016	Kein Eintrag	AS-IS;TO-BE	nein	Schnittstelle 3;Schnittstelle 4	
3	Anwendung 3	31.03.2016	In Entwicklung	PLAN;TO-BE	ja	Schnittstelle 6	
4	Anwendung 4	01.04.2016	Stillgelegt	DECOMMISSIONED	nein		
5	Anwendung 5	01.05.2016	Stillgelegt	DECOMMISSIONED	ja	Schnittstelle 7	
6	Anwendung 6	01.06.2016	In Produktion	AS-IS	nein		
7	Anwendung 7	01.07.2016	Entwurf	TO-BE	nein		
8							
9							

Abb. 8: Import eines bool-Attributs

Hier ist ein Auszug aus der XML-Datei mit dem relevanten Abschnitt:

XML-Beispiel

```
<config>
<sheet name="Anwendungen" class_name="C_APPLICATION" id="1" data_row="1">
...
<attribute name="A_NEED_FOR_ACTION" type="bool" context="de" column="5"
bool_mapping="ja,nein"/>
...
</sheet>
...
</config>
```

2.3.7 <attribute type="relation">

Über den Attributtyp `relation` können Beziehungen zwischen Objekten hergestellt werden. Im ADOIT Metamodell werden diese Beziehungen durch Relationsklassen repräsentiert.

Folgende Attribute müssen für das Tag `<attribute>` definiert werden:

Attributtyp	Beschreibung
<code>name</code>	Sprachunabhängiger Name der Relationsklasse, denen die referenzierten Objekte zugeordnet werden sollen.
<code>type</code>	Angabe des Attributtyps – hier <code>relation</code>
<code>context</code>	Angabe der Sprache des Attributs, das das referenzierte Objekt eindeutig identifiziert. Mögliche Werte sind z.B. „de“ für Deutsch und „en“ für Englisch. Die verfügbaren Sprachen hängen von der Anwendungsbibliothek und der Lizenz ab.
<code>column</code>	Angabe der Spaltennummer der Relation in der Excel-Datei
<code>separator</code>	Definieren Sie das Trennzeichen zwischen mehreren referenzierten Objekten in der Excel-Datei. Wenn die referenzierten Objekte z.B. als "interface1;interface2;interface3" erfasst sind, ist der Wert des Attributs ";".
<code>attr_separator</code> [Optional]	Definieren Sie das Trennzeichen zwischen mehreren Attributwerten, die zur Identifizierung der referenzierten Objekte verwendet werden. Wenn die referenzierten Objekte z.B. als "interface1@description1;interface2@description2" erfasst sind, ist der Wert des Attributs "@".
<code>lookup_attr_name</code>	Sprachunabhängiger Name des Attributs, das das referenzierte Objekt eindeutig identifiziert. Dies muss nicht zwingend das Attribut „Name“ sein. [Optional] Kommaseparierte Liste von sprachunabhängigen Attributnamen zur eindeutigen Identifizierung des referenzierten Objekts.
<code>direction</code>	Festlegung ob die Beziehung als eingehende oder ausgehende Beziehung interpretiert werden soll. Mögliche Werte sind „from“ (eingehende Beziehung) und „to“ (ausgehende Beziehung).

target_class	Sprachunabhängiger Name des referenzierten Objekttyps
strategy	Der Attributwert legt fest, wie die Konflikte bezüglich der Beziehungen während des Excelimports behandelt werden. Siehe Abschnitt Konfliktmanagement beim Import für mehr Informationen. Die Voreinstellung ist <code>overwrite</code> .

2.3.7.1 Eingehende Beziehungen vs. ausgehende Beziehungen

Für jede Beziehung muss festgelegt werden, welches das Quellobjekt und welches das Zielobjekt ist. Aus der Perspektive des Quellobjekts handelt es sich dann um eine ausgehende Beziehung (Symbol →), während es sich aus der Perspektive des Zielobjekts um eine eingehende Beziehung (Symbol ←) handelt.

2.3.7.2 Beispiel 1

Sie wollen Beziehungen anlegen im Attribut *Bereitgestellte Schnittstellen* von Objekten des Typs *Anwendung*:

	A	B	C	D	E	F	G
1	Anwendung 1	29.03.2016	Kein Eintrag	AS-IS	ja	Schnittstelle 1;Schnittstelle 2	
2	Anwendung 2	30.03.2016	Kein Eintrag	AS-IS;TO-BE	nein	Schnittstelle 3;Schnittstelle 4	
3	Anwendung 3	31.03.2016	In Entwicklung	PLAN;TO-BE	ja	Schnittstelle 6	
4	Anwendung 4	01.04.2016	Stillgelegt	DECOMMISSIONED	nein		
5	Anwendung 5	01.05.2016	Stillgelegt	DECOMMISSIONED	ja	Schnittstelle 7	
6	Anwendung 6	01.06.2016	In Produktion	AS-IS	nein		
7	Anwendung 7	01.07.2016	Entwurf	TO-BE	nein		
8							
9							

Abb. 9: Import eines relation-Attributs

Hier ist ein Auszug aus der XML-Datei mit dem relevanten Abschnitt:

```

XML Example
<config>
<sheet name="Anwendungen" class_name="C_APPLICATION" id="1" data_row="1">
...
<attribute name="RC_PROVIDED_INTERFACES" type="relation" context="de"
column="6" separator=";" lookup_attr_name="NAME" direction="TO"
target_class="C_INTERFACE"/>
...
    
```

```

</sheet>
...
</config>

```

2.3.7.3 Beispiel 2

Wie [Beispiel 1](#) oben, allerdings sieht der Inhalt der Zelle in der Excel-Datei in der Spalte F folgendermaßen aus:

F
Schnittstelle 1@Beschreibung 1;Schnittstelle 2@Beschreibung 2
Schnittstelle 3@Beschreibung 3;Schnittstelle 4@Beschreibung 4
Schnittstelle 6@Beschreibung 6
Schnittstelle 7@Beschreibung 7

Abb. 10: Import eines relation-Attributs

D.h. die referenzierten Objekte werden anhand von zwei Attributen eindeutig identifiziert.

Hier ist ein Auszug aus der XML-Datei mit dem relevanten Abschnitt:

XML Example

```

<config>
<sheet name="Anwendungen" class_name="C_APPLICATION" id="1" data_row="1">
...
<attribute name="RC_PROVIDED_INTERFACES" type="relation" context="de"
column="6" separator=";" attr_separator="@";
lookup_attr_name="NAME,DESCRIPTION" direction="TO"
target_class="C_INTERFACE"/>
...
</sheet>
...
</config>

```

2.3.8 <attribute type="file_pointer">

Über den Attributtyp `file_pointer` können Beziehungen zu externen Dokumenten oder Programmen hergestellt werden. Die importierten Werte können im ADOIT Metamodell folgendem Attributtyp zugeordnet werden:

- Dateiverweis (FILE_POINTER)

Dazu müssen folgende Attribute für das Tag `<attribute>` definiert werden:

Attributtyp	Beschreibung
<code>name</code>	Sprachunabhängiger Name des Objektattributs, in das der importierte Wert geschrieben werden soll.
<code>type</code>	Angabe des Attributtyps – hier <code>file_pointer</code>
<code>context</code>	Angabe der Sprache des Attributwerts. Mögliche Werte sind z.B. „de“ für Deutsch und „en“ für Englisch. Die verfügbaren Sprachen hängen von der Anwendungsbibliothek und der Lizenz ab.
<code>column</code>	Angabe der Spaltennummer des Objektattributs in der Excel-Datei
<code>separator</code>	Definiert das Trennzeichen zwischen Programmname und Dateireferenz in der Excel-Datei. Wenn das Trennzeichen leer ist, wird der Zelleninhalt nur als Dateireferenz importiert und der Programmname bleibt leer.

2.3.8.1 Erreichbarkeit von Links zu Dateien

In modernen Browsern sind Links zu Dateien auf einer lokalen Festplatte, UNC-Pfade zu Dateien auf einem Server und Links zu Netzwerklaufwerken aus Sicherheitsgründen nicht erreichbar. Links zu Dateien auf anderen Websites sind normalerweise erreichbar.

2.3.8.2 Beispiel

Sie wollen Beziehungen zu externen Dokumenten anlegen im Attribut *Referenziertes Dokument* von Objekten des Typs *Dokument*:

	A	B	C
1	Dokument 1	file://O:\Temp\Dokument 1.pdf	
2	Dokument 7	file://O:\Temp\Dokument 7.pdf	
3			
4			
5			
6			
7			
8			
9			

Abb. 11: Import eines file_pointer-Attributs

Hier ist ein Auszug aus der XML-Datei mit dem relevanten Abschnitt:

XML Example

```
<config>
<sheet name="Dokumente" class_name="C_DOCUMENT" id="1" data_row="1">
...
<attribute name="A_REFERENCED_DOCUMENT" type="file_pointer" context="de"
column="2"/>
...
</sheet>
...
</config>
```

3 Excel-Import durchführen

Die Durchführung des Excel-Imports besteht aus folgenden Schritten:

3.1 Konfiguration importieren

Damit Sie eine bestimmte Konfiguration benutzen können, müssen Sie zuerst die XML-Konfigurationsdatei in der ADOIT Administration importieren:

1. Wechseln Sie zur Seite *Einstellungen*.
2. Klicken Sie mit der rechten Maustaste auf *Excel-Import*, und klicken Sie dann auf *Erstellen*.
3. Geben Sie im Feld *Name der Konfiguration* einen Namen für Ihre Konfiguration ein und klicken Sie dann auf *OK*.
4. Klicken Sie auf *Importieren* und laden Sie die XML-Konfigurationsdatei hoch. Die Konfiguration wird im Feld *Setup* angezeigt.
5. Klicken Sie auf *Speichern*.

Die Konfiguration wird gespeichert. Sie können jetzt Objekte aus jeder Excel-Datei importieren, die der Konfiguration entspricht.

3.2 Excel-Datei als Vorlage importieren

Zu jeder Konfiguration können Sie in der ADOIT Administration eine passende Excel-Datei als Vorlage importieren.


1. Wechseln Sie zu *Einstellungen > Excel-Import*.
2. Wählen Sie die Konfiguration aus, für die Sie eine Vorlage importieren möchten.
3. Klicken Sie auf *Excel-Vorlage importieren* und laden Sie die Excel-Datei hoch. Der Name der Vorlage wird im Feld *Vorlage* angezeigt.
4. Klicken Sie auf *Speichern*.

Die Konfiguration wird gespeichert. Die Vorlage steht jetzt im Excel Import-Dialog in ADOIT zum Download bereit. Anwender können die Vorlage herunterladen, darin Objekte erfassen und dann die Objekte importieren.


3.3 Objekte aus Excel in ADOIT importieren

Sobald Sie eine XML-Konfigurationsdatei importiert haben, können Sie in ADOIT Objekte aus jeder Excel-Datei importieren, die der Konfiguration entspricht:

1. Wählen Sie im Objektkatalog eine Objektgruppe als Ablageort für Ihre Objekte aus.

2. Klicken Sie mit der rechten Maustaste auf die Objektgruppe, zeigen Sie auf *Import/Export*, und klicken Sie dann auf *Objekte aus Excel importieren*. Ein Dialogfenster öffnet sich.
3. Wählen Sie die gewünschte Konfiguration aus der Dropdown-Liste *Konfiguration auswählen* aus.
4. Geben Sie den Pfad und den Namen der Importdatei im Feld *Excel-Datei* ein (entweder manuell oder über den *Datei auswählen...* Unterstützungsdialo ).
5. Klicken Sie auf *Weiter*. Ein weiteres Dialogfenster mit einer *Import-Vorschau* öffnet sich. Überprüfen Sie die Informationen, um sicherzustellen, dass die Daten korrekt erfasst sind.
6. Bestätigen Sie mit *Importieren*. Die Daten werden importiert und eine Bestätigung erscheint. Klicken Sie auf *Details*, um sicherzustellen, dass der Import vollständig war. Schließen Sie dann den Dialog.

Optional können Sie auch:

- Eine passende Excel-Datei als Vorlage herunterladen. Klicken Sie dazu auf die *Vorlage herunterladen* Schaltfläche . Sie können mit dieser Vorlage Objekte erfassen und dann die Objekte importieren.

3.4 Nach dem Import

Nach einem erfolgreichem Import finden Sie die importierten Objekte im Objektkatalog.

- Objekte, die bereits im Repository vorhanden waren und lediglich aktualisiert wurden, befindet sich an ihrer gewohnten Stelle im Objektkatalog.
- Haben Sie Objekte das erste Mal importiert, finden sich die importierten Objekte in der Objektgruppe "*_Nicht zugeordnet\<Benutzername>\Excel-Import <Datum>\<Objektyp>*".

<Benutzername> steht hier für den Benutzer, der für den Import verwendet wurde, z.B. "Admin".

II Appendix

Im Anhang werden die folgenden Themen beschrieben:

- [**Konfliktmanagement beim Import**](#)
Überblick über die Konfliktmanagement-Strategien beim Import von Objekten
- [**Sprachunabhängige Namen von Metamodellelementen anzeigen**](#)
Sprachunabhängige Namen von Objekttypen, Relationsklassen und Attributen anzeigen

4 Konfliktmanagement beim Import

Die folgenden Konfliktmanagement-Strategien werden beim Import von Objekten angewandt:

- Attribute, die nicht in der Excel-Datei bzw. nicht in der XML-Datei definiert sind, verändern sich durch einen Excel-Import nicht.
- Einfache Attributwerte vom Typ `simple` werden mit dem Wert aus der Excel-Datei überschrieben (auch wenn die Excel-Datei leere Werte enthält).
- Attributwerte vom Typ `date` werden überschrieben, wenn die Excel-Datei gültige Daten enthält. Die Werte werden gelöscht, wenn die Datumzelle in der Excel-Datei leer ist.
- Wenn der Parameter `strategy` auf `overwrite` gesetzt ist, werden die Relationen mit den Werten aus der Excel-Datei überschrieben:
 - Neue Relationen werden angelegt.
 - Relationen, die in der Excel-Datei nicht vorhanden sind, werden entfernt.
 - Wenn die Zelle mit der Relation leer ist, werden alle Beziehungen des konfigurierten Typs aus dem Objekt entfernt.

Wenn der Parameter `strategy` auf `add` gesetzt ist, werden vorhandene Relationen nicht entfernt. Neue Relationen werden angelegt.

- Handhabung von referenzierten Objekten beim Erstellen von Relationen:
 - Wenn das referenzierte Objekt im Repository vorhanden ist, wird die Relation erstellt.
 - Wenn das referenzierte Objekt in der Excel-Datei vorhanden ist, werden das referenzierte Objekt und die Relation erstellt.
 - Wenn das referenzierte Objekt weder in der Excel-Datei noch im Repository vorhanden ist, wird die Relation nicht erstellt.
 - Wenn das referenzierte Objekt in der Excel-Datei vorhanden ist, aber nicht erstellt werden konnte, wird die Relation nicht erstellt.
 - Wenn unklar ist welches Objekt referenziert werden soll weil es mehrere potenzielle Ziele gibt, wird die Relation nicht erzeugt.

5 Sprachunabhängige Namen von Metamodellelementen anzeigen

Um erfolgreich eine Konfiguration zu erstellen, müssen Sie Zugriff auf die Seite *Eigenschaften* in der ADOIT Administration haben. Die sprachunabhängigen Namen von Objekttypen, Attributen usw. sind dort zu finden und werden für die Konfiguration benötigt.

So zeigen Sie die sprachunabhängigen Namen von Metamodellelementen an:

1. Öffnen Sie die Seite *Eigenschaften*.
2. Wählen Sie oben auf der Seite in der Liste *Klasse oder Modelltyp auswählen* den gewünschten Objekttyp aus.
 - Rechts neben der Liste wird der sprachunabhängige Name des Objekttyps angezeigt.
 - Im Arbeitsbereich werden die Eigenschaften des Objekttyps angezeigt (organisiert in Kapitel). Der sprachunabhängige Name von Attributen und Relationen wird in der Spalte *Sprachunabhängig* angezeigt.

Index

A

attribute 5

B

bool 13

C

config 3

D

date 7

E

enum 9

enum_list 11

Excel-Datei (Aufbau) 2

Excel-Datei als Vorlage importieren 20

Excel-Import durchführen 20

F

file_pointer 18

K

Konfiguration erstellen 3

Konfiguration importieren 20

Konfliktmanagement beim Import 23

M

Metamodellverwaltung 24

N

Nach dem Import 21

O

Objekte aus Excel in ADOIT importieren 20

R

relation 15

S

sheet 3

simple 6

Sprachunabhängige Namen anzeigen 24

T

treenumlist 12



www.boc-group.com

BOC Group

Athen • Berlin • Dublin • Madrid • Paris • Warschau • Wien • Winterthur