

# AFIS-ALKIS-ATKIS-Tools

auf der Basis von ShapeChange

## Dokumentation

Clemens Portele (interactive instruments GmbH)  
Reinhard Erstling (interactive instruments GmbH)  
Stefan Olk (interactive instruments GmbH)

Version: 0.6

Status: In Bearbeitung

Stand: 22.08.2013

## Zusammenfassung

Dieses Dokument gibt alle erforderlichen Hinweise für den Betrieb der AFIS-ALKIS-ATKIS-Tools

- zur Erstellung von Objektartenkatalogen (Katalogtool)
- zur Erstellung und zum Laden von Profilen (Profiltool)
- zur Ableitung der NAS (NAS-Tool)

Erläutert werden der Aufbau des Systems und seine Konfigurationsmöglichkeiten, die Regeln für die weitere Benutzung, Beschränkungen und möglichen Fehlersituationen.

## Historie

Version	Status	Datum	Autor(en)	Erläuterung
0.1	In Bearbeitung	18.03.2010	Clemens Portele, Stefan Olk, Reinhard Erstling	Erste Version
0.2	In Bearbeitung	31.03.2010	Clemens Portele	Anpassungen an die Beta2-Version der Tools
0.3	In Bearbeitung	02.05.2010	Clemens Portele	Anpassungen an die Beta3-Version der Tools
0.4	In Bearbeitung	06.11.2011	Clemens Portele	Anpassungen an die Beta4-Version der Tools
0.5	In Bearbeitung	19.08.2012	Clemens Portele	Anpassung an aktuelle ShapeChange-Version
0.6	In Bearbeitung	22.08.2013	K.-H. Böhmer, Clements Portele	Aktualisierung der Dokumentation

# Inhaltsverzeichnis

<b>1</b>	<b>Komponenten und Artefakte</b>	<b>4</b>
1.1	Übersicht	4
1.2	AAA-Modell	4
1.3	Enterprise Architect (EA)	4
1.4	ShapeChange	4
1.4.1	Übersicht	4
1.4.2	ShapeChange Konfiguration	5
1.4.2.1	Überblick	5
1.4.2.2	Konfigurationselement: input	6
1.4.2.3	Konfigurationselement: dialog	7
1.4.2.4	Konfigurationselement: log	8
1.4.2.5	Konfigurationselement: targets	8
1.4.2.5.1	Target-Konfiguration: XmlSchema	8
1.4.2.5.2	Target-Konfiguration: AAA-Katalogtool	10
1.4.2.5.3	Target-Konfiguration: AAA-Profiltool	12
1.4.3	Schema-Fehlermeldungen aus ShapeChange	15
<b>2</b>	<b>Installation</b>	<b>16</b>
<b>3</b>	<b>Ausführung</b>	<b>18</b>
3.1	AAA-Katalogtool	18
3.2	AAA-Profiltool	20
3.3	NAS-Tool	23
3.4	Störmerkmale	23
3.4.1	Meldungen bei der Katalogerzeugung	23
3.4.2	Meldungen bei der Profilverarbeitung	24
3.4.3	Meldungen bei der Modelltransformation bei der NAS-Erzeugung	24

# 1 Komponenten und Artefakte

## 1.1 Übersicht

Dieser Abschnitt erläutert die wesentlichen Komponenten und Artefakte, die für die AAA-Tools benötigt werden.

## 1.2 AAA-Modell

Das AAA-Modell ist ein Enterprise-Architect-Repository (Dateiendung „eap“). Die jeweils gültige Version des AAA-Modells wird von der [AdV](#) veröffentlicht. Das Modell ist in der GeoInfoDok beschrieben.

## 1.3 Enterprise Architect (EA)

Zentraler Software-Baustein für das AAA-Modell ist UML-Modellierungswerkzeug *Enterprise Architect (EA)*, ein Produkt der Firma SparxSystems Ltd.

Verwendet wird die Version 7.5 (Build 848) oder eine spätere Version. Getestet wurden die Tools zuletzt mit Version 10.0 (Build 1008).

## 1.4 ShapeChange

### 1.4.1 Übersicht

ShapeChange ist eine in Java geschriebene Software von interactive instruments. Die in den AAA-Tools verwendete Version ist unter der Gnu Public License (GPL) verfügbar.

Die Software liest das Modell aus dem EA-Repository, verarbeitet dieses und erzeugt daraus verschiedene Zielschemata, unter anderem GML-Applikationsschemata und Codelisten-Dictionaries. Für die Verarbeitung des AAA-Modells wurden die Erzeugung von Objektartenkatalogen (AAA-Katalogtool) und die Verarbeitung von Profilen (AAA-Profiltool) ergänzt. Außerdem wurde ShapeChange um die Möglichkeit Java-basierter einer Modelltransformation vor Erzeugung des GML-Applikationsschemas ergänzt (NAS-Tool).

Weitere Informationen zu ShapeChange sind unter <http://shapechange.net/> verfügbar.

Die AAA-Tools erweitern ShapeChange um weitere "Targets".

## 1.4.2 ShapeChange Konfiguration

Dieser Abschnitt beschreibt die Steuerung von ShapeChange über Konfigurationsdateien. Alle Verzeichnisse und wesentlichen Einstellungen von ShapeChange sind konfigurierbar gestaltet, so dass eine Anpassung an eine konkrete Verteilung auf Rechner jederzeit erfolgen kann.

Im regulären Umgang sollte eine Änderung an den Konfigurationsdateien nur selten erforderlich sein. Sie kann jedoch durchgeführt werden, wenn die eingestellten Verzeichnisse geändert werden sollen oder wenn von den diversen Optionen beim Generierungsprozess Gebrauch gemacht werden soll.

Die vordefinierten ShapeChange-Konfigurationen befindet sich in den Dateien (alle im Verzeichnis „Konfigurationen“):

**AAAKatalog-ALKIS-(6.0.1|7.0).xml** (AAA-Katalogtool für den ALKIS-Objektartenkatalog)

**AAAKatalog-AFIS-(6.0.1|7.0).xml** (AAA-Katalogtool für den AFIS-Objektartenkatalog)

**AAAKatalog-BasisDLM-(6.0.1|7.0).xml** (AAA-Katalogtool für den ATKIS-Basis-DLM-Objektartenkatalog)

**AAAProfil-(6.0.1|7.0).xml** (AAA-Profiltool)

**NAS-(6.0.1|7.0).xml** (NAS-Tool)

Diese bezieht sich auf weitere in XML-Dokumenten gekapselte Teile, die per Xinclude referiert werden, nämlich:

- **<http://shapechange.net/resources/config/StandardAliases.xml>**
- **<http://shapechange.net/resources/config/StandardNamespaces.xml>**
- **<http://shapechange.net/resources/config/StandardMapEntries.xml>**
- **<http://shapechange.net/resources/config/StandardRules.xml>**

Zur Konfigurationsdatei gehört ein XML-Schema-Dokument namens

**<http://shapechange.net/resources/schema/ShapeChangeConfiguration.xsd>.**

Die Konfigurationsdatei wird grundsätzlich gegen dieses vorgegebene Schema validiert, die Verletzung des Schemas führt zu einem „FatalError“.

### 1.4.2.1 Überblick

Im Folgenden ist die Struktur der Konfigurationsdatei dargestellt.

Innerhalb des <ShapeChangeConfiguration>-Elements gliedert sich die Datei in verschiedene „funktionale Elemente“, welche in den folgenden Abschnitten einzeln besprochen werden.

Die Namen der Elemente lassen auf ihre Funktion schließen, z.B. bestimmt <input> aus welcher Quelle das Modell bezogen wird.

Das Element <target> enthält die Konfigurationen der einzelnen Zielkodierungen.

Sie sind unter den Elementen <Target> oder <TargetXmlSchema> (letzteres im Spezialfall GML-Applikationsschema-Erzeugung) unter Angabe des Namens der Klasse, die das „Target“ implementiert, und dem Optionsattribut mode=“...“ spezifiziert.

```
<?xml version="1.0" encoding="UTF-8"?>
<ShapeChangeConfiguration
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns="http://www.interactive-instruments.de/ShapeChange/Configuration/1.1"
  xmlns:sc="http://www.interactive-instruments.de/ShapeChange/Configuration/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.interactive-instruments.de/ShapeChange/Configuration/1.1
http://shapechange.net/resources/schema/ShapeChangeConfiguration.xsd">
  <input>
    ...
  </input>
  <dialog>
    ...
  </dialog>
  <log>
    ...
  </log>
  <targets>
    <TargetXmlSchema class="de.interactive_instruments.ShapeChange.Target.XmlSchema.XmlSchema" mode="enabled">
      ...
    </TargetXmlSchema>
    <Target class="de.adv_online.aaa.katalogtool.Katalog" mode="enabled">
      ...
    </Target>
  </targets>
</ShapeChangeConfiguration>
```

#### 1.4.2.2 Konfigurationselement: input

Das <input>-Element legt fest, aus welcher Quelle das Modell bezogen wird und definiert einige Parameter, welche die Interpretation des Modells steuern.

Die Angabe der Parameter erfolgt strikt nach dem auch sonst überall verwendeten Schema von *name* und *value*.

```
<input>
  <parameter name="inputModelType" value="EA7"/>
  <parameter name="inputFile" value="./AAA.eap"/>
  <parameter name="appSchemaName" value="AFIS-ALKIS-ATKIS Anwendungsschema"/>
  <parameter name="publicOnly" value="true"/>
  <parameter name="checkingConstraints" value="disabled"/>
  <parameter name="representTaggedValues" value
="AAA:Kennung,AAA:Datum,AAA:Organisation,AAA:Modellart,AAA:Profile,AAA:Grunddatenbestand,
AAA:Nutzungsart,AAA:Nutzungsartkennung,AAA:objektbildend,AAA:Themen "/>
  <xi:include href="http://shapechange.net/resources/config/StandardAliases.xml"/>
</input>
```

Eine sinnvolle Möglichkeit zum Eingriff ergibt sich bei folgenden Namen:

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
inputModelType	Pflichtfeld	Der Wert ist für das AAA-Modell stets "EA7", d.h. es erfolgt ein Direktzugriff auf

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
		das Modell über die Programmierschnittstelle von Enterprise Architect.
inputFile	Pflichtfeld	Der Parameter nennt das EA-Repository, welches Input in den Prozess ist.
appSchemaName	alle Anwendungsschemata im Modell	Name des zu verarbeitenden Anwendungsschemas. Für das AAA-Modell ist hier stets „AFIS-ALKIS-ATKIS Anwendungsschema“ anzugeben.
publicOnly	true	Nur Attribute, welche als „public“ ausgezeichnet sind, werden berücksichtigt.
checkingConstraints	enabled	Sofern der Parameter nicht "disabled“ ist, werden die OCL-Constraints im Modell analysiert und geprüft, unabhängig davon, ob diese in der weiteren Ausführung verwendet werden.
representTaggedValues	keiner	Die im UML-Profil des AAA-Modells definierten Tagged Values, die für die AAA-Tools relevant sind.
transformer	keine Modelltransformation	Optionale Modelltransformation vor der Verarbeitung des Modells. Nur für das NASTool relevant. Hier sollte der Wert "de.adv_online.aaa.nastool.NasTransformer" sein.
transformerTargetPath	leerer String	Nur im NASTool relevant. Die URLs der Schemadokumente aus den Tagged Values 'xsdDocument' werden mit diesem Wert präfixt.

### 1.4.2.3 Konfigurationselement: dialog

Mit dem <dialog>-Element wird – sofern ein Dialog im Tool verwendet wird – der Dialog festgelegt.

Die Angabe der Parameter erfolgt strikt nach dem auch sonst verwendeten Schema von *name* und *value*.

```
<dialog>
  <parameter name="dialogClass" value="de.adv_online.aaa.katalogtool.KatalogDialog"/>
</dialog>
```

Die Parameter sind wie folgt:

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
-------------	----------------------	------------------

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
dialogClass	leer	Angabe der verwendeten Dialogklasse.  Für das AAA-Katalogtool ist dies de.adv_online.aaa.katalogtool.KatalogDialog.  Für das AAA-Profiltool ist dies de.adv_online.aaa.profiltool.ProfilDialog.  Das NAS-Tool verwendet keinen Dialog.

#### 1.4.2.4 Konfigurationselement: log

Dieses Element kontrolliert das Logging von Meldungen durch ShapeChange.

```
<log>
  <parameter name="reportLevel" value="WARNING"/>
  <parameter name="logFile" value="AAA-Katalogtool-Log.xml"/>
</log>
```

Die Parameter sind wie folgt:

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
reportLevel	INFO	Möglich sind INFO, WARNING, ERROR.
logFile	leer	Der Dateiname der Datei für Meldungen.

#### 1.4.2.5 Konfigurationselement: targets

##### 1.4.2.5.1 Target-Konfiguration: XmlSchema

Dieser Abschnitt steuert die Erzeugung des GML-Applikationsschemas aus dem Modell. Er kooperiert dabei mit einer Reihe von *tagged values* im Modell an Paketen mit dem Stereotype <<applicationSchema>>.

```
<TargetXmlSchema class="de.interactive_instruments.ShapeChange.Target.XmlSchema.XmlSchema" mode="enabled">
  <targetParameter name="outputDirectory" value="Ausgaben/NAS"/>
  <targetParameter name="defaultEncodingRule" value="NAS"/>
  <xi:include href="http://shapechange.net/resources/config/StandardRules.xml"/>
  <rules>
    <EncodingRule name="NAS" extends="iso19136_2007">
      <rule name="rule-xsd-all-no-documentation"/>
    </EncodingRule>
  </rules>
  <xi:include href="http://shapechange.net/resources/config/StandardNamespaces.xml"/>
  <xi:include href="http://shapechange.net/resources/config/StandardMapEntries.xml"/>
  <xmlNamespaces>
    <XmlNamespace nsabr="ows" ns="http://www.opengis.net/ows" location="http://schemas.opengis.net/ows/1.0.0/owsAll.xsd"/>
    <XmlNamespace nsabr="wfs" ns="http://www.adv-online.de/namespaces/adv/gid/wfs" location="http://repository.gdi-de.org/schemas/adv/wfs/1.0/WFS-transaction.xsd"/>
    <XmlNamespace nsabr="feCap" ns="http://www.opengis.net/ogc/filterCapabilities" location="http://repository.gdi-de.org/schemas/adv/filter/1.0/filterCapabilities.xsd"/>
  </xmlNamespaces>
</TargetXmlSchema>
```

```

    </xmlNamespaces>
    <xsdMapEntries>
      <XsdMapEntry type="URI" xsdEncodingRules="iso19136_2007" xmlPropertyType="anyURI" xmlType="anyURI"
xmlTypeType="simple" xmlTypeContent="simple"/>
      <XsdMapEntry type="URL" xsdEncodingRules="iso19136_2007" xmlPropertyType="anyURI" xmlType="anyURI"
xmlTypeType="simple" xmlTypeContent="simple"/>
      <XsdMapEntry type="doubleList" xsdEncodingRules="iso19136_2007" xmlPropertyType="gml:doubleList"
xmlType="gml:doubleList" xmlTypeType="simple" xmlTypeContent="simple"/>
      <XsdMapEntry type="AC_FeatureCatalogue" xsdEncodingRules="iso19136_2007" xmlProperty-
Type="gml:ReferenceType" xmlType="gml:ReferenceType"/>
      <XsdMapEntry type="CapabilitiesBase" xsdEncodingRules="iso19136_2007"
xmlType="ows:CapabilitiesBaseType"/>
      <XsdMapEntry type="Identification" xsdEncodingRules="iso19136_2007" xmlType="ows:IdentificationType"/>
      <XsdMapEntry type="Query" xsdEncodingRules="iso19136_2007" xmlType="wfs:QueryType" xmlEle-
ment="wfs:Query" xmlPropertyType="_P_"/>
      <XsdMapEntry type="FeatureCollection" xsdEncodingRules="iso19136_2007"
xmlType="wfs:FeatureCollectionType" xmlElement="wfs:FeatureCollection" xmlPropertyType="_P_"/>
      <XsdMapEntry type="Transaction" xsdEncodingRules="iso19136_2007" xmlType="wfs:TransactionType" xmlEl-
ement="wfs:Transaction" xmlPropertyType="_P_"/>
      <XsdMapEntry type="Filter_Capabilities" xsdEncodingRules="iso19136_2007"
xmlType="feCap:Filter_CapabilitiesType" xmlElement="feCap:Filter_Capabilities" xmlPropertyType="_P_"/>
    </xsdMapEntries>
  </TargetXmlSchema>

```

Dieser Abschnitt beschreibt die für die Erzeugung der NAS relevanten Parameter.

Die Parameter sind wie folgt:

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
outputDirectory	.	Verzeichnis für die Ausgabe der XSD-Dokumente des GML-Applikationsschemas.
defaultEncodingRule	iso19136_2007	Die i.d.R. anzuwendende Encoding Rule. Bei der NAS ist dies "NAS", was gleichbedeutend mit der Standard Encoding Rule iso19136_2007 mit der Erweiterung, dass die Dokumentation nicht in das XML Schema abgebildet wird.  Die anzuwendende Encoding Rule wird im Modell für jedes Element gesetzt (einschließlich der Ausgabe von ISO/TS 19139 konformen Metadatenelementen).
rules/EncodingRule	keine zusätzliche Encoding Rule	Definition von Encoding Rules, die nicht bereits in StandardRules.xml definiert sind.

Die drei per XInclude referierten Dokumente stehen für weitere Definitionen.

- **StandardRules.xml** spezifiziert die typischen, von ShapeChange unterstützten Encoding Rules.
- **StandardNamespaces.xml** liefert die typischen XML-Namespace-Definitionen für die Erzeugung von GML 3.2 Anwendungsschemata.

- **StandardMapEntries.xml** definiert die komplette Abbildung großer Teile des ISO 19100 Modells in entsprechende GML-Elemente.

Die Elemente `<xmlNamespaces>` und `<xsdMapEntries>` enthalten zusätzliche AAA-Modell-spezifische XML-Namespace-Definitionen und vordefinierte XML-Schema-Abbildungen.

#### 1.4.2.5.2 Target-Konfiguration: AAA-Katalogtool

In diesem Abschnitt wird die Konfiguration der Erzeugung der Objektartenkataloge beschrieben. Die Konfiguration erfolgt über einen entsprechenden Target-Abschnitt in der Konfigurationsdatei.

Ein konkretes Beispiel für den Abschnitt in der Konfigurationsdatei sieht wie folgt aus:

```
<Target class="de.adv_online.aaa.katalogtool.Katalog" mode="enabled">
  <targetParameter name="Verzeichnis" value="Ausgaben/Kataloge"/>
  <targetParameter name="paket" value="AFIS-ALKIS-ATKIS Fachschema"/>
  <targetParameter name="modellarten" value="DLKM,DKKM500,DKKM1000,DKKM2000,DKKM5000"/>
  <targetParameter name="profile" value=""/>
  <targetParameter name="profilquelle" value="Modell"/>
  <targetParameter name="nurGrunddatenbestand" value="false"/>
  <targetParameter name="letzteAenderung" value="true"/>
  <targetParameter name="geerbteEigenschaften" value="false"/>
  <targetParameter name="schemakennungen" value="*/>
  <targetParameter name="ausgabeformat" value="HTML,RTF,XML"/>
  <targetParameter name="notesRuleMarker" value="---"/>
</Target>
```

Die Parameter sind wie folgt:

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
Verzeichnis	.	Basisverzeichnis für die Ausgabe der Katalogdateien. Der Pfad darf keine Leerzeichen beinhalten.
paket	das gesamte Anwendungsschema	Die Ausgabe kann auf ein Paket im Anwendungsschema eingeschränkt werden. Im Rahmen des AAA-Modells ist dies stets das „AFIS-ALKIS-ATKIS Fachschema“.
modellarten	alle Modellarten	Liste der Modellarten, durch Kommata getrennt, für die der Katalog erzeugt werden soll. Mindestens eine Modellart muss bei der Katalogerstellung angegeben werden.
profile	kein Profil	Zusätzlich zur Beschränkung der Modellarten kann die Katalogerstellung auch auf ein oder mehrere Profile (erzeugt mit dem AAA-Profiltool) eingeschränkt werden. Im Parameter wird der Name der Profile angegeben.
profilquelle	Modell	<i>Datei</i> oder <i>Modell</i> . Bei <i>Modell</i> wird auf das entsprechende Tagged Value im Modell zugegriffen, bei <i>Datei</i> auf ein Profildatei; das Tool

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
		versucht hierzu im Verzeichnis gemäß Parameter „Verzeichnis“ auf die Datei mit dem Namen des Profils und der Dateierweiterung „.3ap“ zuzugreifen.
nurGrunddatenbestand	false	<i>true</i> oder <i>false</i> . Durch <i>true</i> wird die Katalogerstellung auf den Grunddatenbestand der jeweiligen Modellarten eingeschränkt werden.
letzteAenderung	false	<i>true</i> oder <i>false</i> . Schalter für die Ausgabe der Angabe zu letzten Änderung pro Modellelement (Version, Revisionsnummer).
geerbteEigenschaften	false	<i>true</i> oder <i>false</i> . Über den Schalter kann festgelegt werden, ob Eigenschaften aus Oberklassen auch in abgeleiteten Klassen explizit aufgeführt werden sollen. Dabei werden nur Oberklassen berücksichtigt, deren Präfix in der obigen Liste enthalten ist.
schemakennungen	*	In den ISO 19100-Normen werden verschiedene Namensräume durch zweibuchstabile Präfixe gekennzeichnet. Diese Vorgehensweise wurde im AAA-Modell übernommen. Es werden beim Export nur UML-Klassen berücksichtigt, deren Präfix im zweiten Eingabefeld enthalten ist (zwei Buchstaben plus Unterstrich). Ein "*" sorgt dafür, dass alle Klassen berücksichtigt werde
ausgabeformat	Pflichtfeld	Die auszugebenden Katalogformate.  Weitere Werte sind neben dem genannten Default u.a. die folgenden Werte:  HTML RTF XML Nart-PDF GFC CSV
notesRuleMarker	----	Die Zeichenkette mit der in den Dokumentationsfeldern die Definitionen von den Bildungsregeln, Konsistenzbedingungen und Erfassungskriterien getrennt werden.
xsltPfad	src/main/resources/xslt	URI oder lokaler Pfad zum Verzeichnis, in dem die XSLT-Stylesheets für die Katalogerstellung liegen

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
xslhtmlFile	aaa-html.xsl	XSLT-Stylesheet für die Ausgabe des HTML-Objektartenkatalogs
xslrtfFile	aaa-rtf.xsl	XSLT-Stylesheet für die Ausgabe des RTF-Objektartenkatalogs (Microsoft Word)
xslxmlFile	aaa-xml.xsl	XSLT-Stylesheet für die Ausgabe des XML-Objektartenkatalogs gemäß XML Schema aus früheren Versionen des Katalogtools
xslgfcFile	aaa-gfc.xsl	XSLT-Stylesheet für die Ausgabe des XML-Objektartenkatalogs gemäß XML Schema in ISO 19110 (GFC)
xslcsvFile	aaa-csv.xsl	XSLT-Stylesheet für die Ausgabe des CSV-Objektartenkatalogs
xslfoFile	aaa-nart.xml	XSLT-Stylesheet für die Ausgabe des Nutzungsartenverzeichnisses in PDF
referenceModelFile	kein Vorgängermodell	Der Parameter nennt das EA-Repository der Vorgängerversion des AAA-Modells, gegen das die Änderungen bestimmt werden sollen. Diese werden als Änderungsmarkierungen ausgegeben. Diese Funktionalität ist nur bei der HTML-Ausgabe möglich, ansonsten sollte kein Vorgängermodell angegeben werden.
referenceModelType	kein Vorgängermodell	Der Wert ist für das AAA-Modell stets "EA7", d.h. es erfolgt ein Direktzugriff auf das Modell über die Programmierschnittstelle von Enterprise Architect.

#### 1.4.2.5.3 Target-Konfiguration: AAA-Profiltool

In diesem Abschnitt wird die Konfiguration der Pflege von Profilen beschrieben.

Unter einem Profil wird eine wohldefinierte und benannte Untermenge der zu führenden Informationen des AAA-Anwendungsschemas innerhalb einer Modellart verstanden. Die umfasst im Sinne des UML-Modells Klassen (Objektarten, Datentypen, Unions, Aufträge, Ergebnisse, Codelisten), Attribute (Attributarten, Wertarten) und Rollen (Relationsarten). Alle in dem Profil vorkommenden Elemente müssen auch der zugehörigen Modellart zugeordnet sein.

Für die Festlegung der Profile im AAA-Anwendungsschema wird das AAA-Profiltool verwendet, das die entsprechenden Profilangaben in einem speziellen Tagged Value in den UML-Elementen des Anwendungsschemas setzt bzw in Profildateien schreibt.

Ein Profil beschreibt typischerweise den von einer datenführenden Stelle gepflegte Datenumfang zu einer Modellart. Er umfasst mindestens den Grunddatenbestand.

Profile können bei der Katalogerzeugung berücksichtigt werden.

Die Konfiguration des AAA-Profiltools erfolgt über einen entsprechenden Target-Abschnitt in der Konfigurationsdatei.

Ein konkretes Beispiel für den Abschnitt in der Konfigurationsdatei sieht wie folgt aus:

```
<Target class="de.adv_online.aaa.profiltool.Profil"
mode="enabled">
  <targetParameter name="Verzeichnis" value="Ausgaben/Profile"/>
  <targetParameter name="Modellart" value="DLKM"/>
  <targetParameter name="Profil" value="TestMIN"/>
  <targetParameter name="Quelle" value="Neu_Minimal"/>
  <targetParameter name="Ziel" value="DateiModell"/>
</Target>
```

Die Parameter sind wie folgt:

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
Verzeichnis	.	Verzeichnis für die Profildateien. Der Pfad darf keine Leerzeichen beinhalten.
Modellart	Pflichtfeld	Modellart für das Profil
Profil	Pflichtfeld	Name des Profils innerhalb der Modellart.  Der vollständige Name des Profils ergibt sich aus der Kombination von Modellart, einem Unterstrich und diesem Parameter.
Quelle	Pflichtfeld	Bezeichnet die Quelle der Profildefinition.  <i>Datei:</i> Die Profildefinition wird aus einer Datei geladen. Das Tool versucht hierzu im Verzeichnis gemäß Parameter „Verzeichnis“ auf die Datei mit dem folgenden Namen zuzugreifen: Werte des Parameters „Modellart“, gefolgt von einem Unterstrich, dem Wert des Parameters „Profil“ sowie „3ap“.  <i>Modell:</i> Die Profildefinition wird aus dem AAA-Modell geladen.  <i>Neu_Minimal:</i> Ein neues Profil wird erzeugt, auf das Minimalprofil eingeschränkt.  <i>Neu_Maximal:</i> Ein neues Profil wird erzeugt – mit allen zu der Modellart gehörenden Modellelementen.
Ziel	Pflichtfeld	Bezeichnet das Ziel der Profildefinition.  <i>Datei:</i> Das Profil wird in eine Datei geschrieben. Der Dateiname ist eine Kombination des Parameters „Modellart“, einem Unterstrich,

<i>name</i>	<i>default-value</i>	<b>Bedeutung</b>
		dem Parameter „Profil“ und „__export.3ap“.
		<i>Modell</i> : Die Profildefinition wird in das AAA-Modell geschrieben.
		<i>DateiModell</i> : Die Profildefinition wird sowohl in die Datei als auch das Modell geschrieben.
		<i>Ohne</i> : Es erfolgt keine Ausgabe des Profils.

Das Profiltool ist so ausgelegt, dass bei sauberer Definition der Modellarten stets ein konsistenter Profilstand vorliegt. D.h., zu Beginn der Ausführung werden etwaige Angaben zu dem Profil aus allen Elementen des UML-Modells entfernt, durch die Ausführung erfolgt eine vollständige Neudefinition des Profils. Ermöglicht wird so die Führung von Profilen außerhalb des von der AdV in der GeoInfoDok veröffentlichten UML-Modells. In den lokalen Kopien des UML-Modells können mit dem Profiltool die Profildefinitionen in das Modell eingebracht werden. Bei ggf. auftretenden Fehlermeldungen (z.B. aufgrund von Änderungen des UML-Modells wie Umbenennungen oder fachliche Modelländerungen) muss die profildefinierende Stelle reagieren und Anpassungen an der Profildefinition vornehmen.

Das Profiltool berücksichtigt auch automatisch die folgenden Rahmenbedingungen:

- Pflichtattributarten und -relationsarten werden bei der Aufnahme einer Objektart in ein Profil automatisch dem Profil hinzugefügt.
- Dasselbe gilt für alle Oberklassen der Objektart.
- Alle Objektarten, Attributarten, Relationsarten und Wertarten des Grunddatenbestands einer Modellart werden ebenfalls automatisch dem Profil hinzugefügt.

Die externen Profildateien tragen die Dateikennung *3ap*. Der Aufbau der Dateien ist wie folgt:

Nach einem Headerblock wird jedes Element der Modellart (Objektart/Datentyp, Alltributart, Relationsart oder Wertart) durch eine Zeile repräsentiert. Ist das Element kein Bestandteil des Profils, dann wird es mit einem "#" als erstem Zeichen in der Zeile auskommentiert. Durch Entfernen des Kommentarzeichens oder Auskommentieren von Zeilen kann nun die Profildefinition geändert und wieder über das Profiltool in das UML-Modell oder bei der Katalogerzeugung übernommen werden.

Der Header enthält in der ersten Zeile den festen Wert "AFIS-ALKIS-ATKIS-Profildefinition", in der zweiten Zeile die Angabe der Versionsnummer des AAA-Anwendungsschemas aus der die Profilbeschreibung erzeugt wurde und in der dritten Zeile den vollständigen Namen des Profils.

Ein Beispiel für eine Profilbeschreibung:

```
AFIS-ALKIS-ATKIS-Profildefinition
Version: 6.0.1
Name: DLKM_Min
#
AA_ZUSO --- Objektart/Datentyp
AA_Objekt --- Objektart/Datentyp
```

```

AA_Objekt/identifikator --- Attributart
AA_Objekt/lebenszeitintervall --- Attributart
AA_Objekt/modellart --- Attributart
# AA_Objekt/anlass --- Attributart
# AA_Objekt/zeigtAufExternes --- Attributart
# AA_Objekt/istTeilVon --- Relationsart
AA_ZUSO/bestehtAus --- Relationsart
AA_REO --- Objektart/Datentyp
# AA_REO/traegtBeiZu --- Relationsart
# AA_REO/istAbgeleitetAus --- Relationsart
# AA_REO/hatDirektUnten --- Relationsart
AA_NREO --- Objektart/Datentyp
# AA_Fachdatenverbindung --- Objektart/Datentyp
# AA_Fachdatenverbindung/art --- Attributart
# AA_Fachdatenverbindung/fachdatenobjekt --- Attributart
AA_Lebenszeitintervall --- Objektart/Datentyp
AA_Lebenszeitintervall/beginnt --- Attributart
# AA_Lebenszeitintervall/endet --- Attributart
AA_UUID --- Objektart/Datentyp
AA_UUID/UUID --- Attributart
AA_UUID/UUIDundZeit --- Attributart
# AA_Fachdatenobjekt --- Objektart/Datentyp
# AA_Fachdatenobjekt/name --- Attributart
# AA_Fachdatenobjekt/uri --- Attributart
AA_Modellart --- Objektart/Datentyp
AA_Modellart/advStandardModell --- Attributart
AA_Modellart/sonstigesModell --- Attributart
# AA_AdVStandardModell/DLKM --- Wertart
# AA_AdVStandardModell/DKKM500 --- Wertart
# AA_AdVStandardModell/DKKM1000 --- Wertart
...

```

Bei einer Objektart oder einem Datentyp ist in der Zeile der Name angegeben. Bei einer Attributart wird der Name der Objektart bzw. des Datentyps und der Name der Attributart angegeben. Bei einer Wertart der Name der Codeliste und der Code. Bei einer Relationsart schließlich wird der Name der Objektart und der Name der Relationsart angegeben.

### 1.4.3 Schema-Fehlermeldungen aus ShapeChange

ShapeChange erzeugt bei der Analyse des Modells und bei der Übersetzung in die Targets Fehlermeldungen, wenn es Modellelemente findet, die nicht den jeweiligen Abbildungsregeln entsprechen. Die Meldungen sind Langtexte in englischer oder deutscher Sprache, welche die jeweilige Fehlersituation hinreichend beschreiben.

Meldungen gibt es in verschiedener Schwere: *Info*, *Warning*, *Error*, *FatalError*.

Eine Liste der möglichen Fehlermeldungen in jeder Phase findet sich unter 3.4.

Die Meldungen werden in der Logdatei dokumentiert.

## 2 Installation

Die AAA-Tools setzen eine Reihe von vorinstallierten Komponenten voraus, die alle vorhanden sein müssen, um die Software geordnet installieren zu können:

- Windows Betriebssystem, XP, Vista, 7 oder 8
- Enterprise Architect (EA) der Firma SparxSystems Ltd, Version 7.5 oder höher.
- Java Laufzeitsystem, Version 1.6 oder höher.

Weitere Hinweise:

- Die aktuellen Tools tragen die Bezeichnung: "AAATools-1.0.0-20130618.160614-4.zip" und können von den Seiten der AdV heruntergeladen werden.
- Für die Ausführung der Tools ist ggf. eine Vollversion von EA erforderlich (weitere in Kapitel 3 "Ausführung")
- Java kann kostenlos im Internet heruntergeladen werden unter: <http://www.java.com/de/download/>.

Die AAA-Tools werden in einem ZIP-Archiv bereitgestellt. Es enthält ShapeChange und die AAA-Tools-Erweiterungen sowie das aktuelle AAA-Modell der GeoInfoDok 6.0.1.

Bei Auslieferung sind die Konfigurationsdateien so eingestellt, dass die Anwendung lauffähig ist und die lokalen Verzeichnisse verwendet werden. Dies kann durch Bearbeitung der Konfigurationsdateien individuell angepasst werden, bei Bedarf können auch andere Konfigurationsdateien erstellt und genutzt werden.

Die Konfigurationsdateien erwarten, dass das AAA-Modell im Wurzelverzeichnis des ausgepackten Archivs liegt und den Namen AAA.eap trägt.

Durchführung der Installation:

- Vorbereitung der Enterprise Architect Java-API: Die Datei SSJavaCom.dll muss im System-Pfad abgelegt werden (<Windows Verzeichnis>/System32 auf einem 32-Bit-System, <Windows Verzeichnis>/SysWOW64 auf einem 64-Bit-System) abgelegt werden. Quelle dieser Datei: <EA-Installationsverzeichnis>/Java API/

Hinweis: Auf einem 64-Bit-System ist java.exe aus dem <Windows Verzeichnis>/SysWOW64-Verzeichnis zu verwenden, da Enterprise Architect eine 32-Bit-Anwendung ist.

- Auspacken des Archivs in ein Installationsverzeichnis. Der Pfad des Installationsverzeichnisses darf keine Leer- oder Sonderzeichen beinhalten.
- Ggf. Anpassung der Konfigurationsdateien im Verzeichnis *Konfigurationen*: z.B. Angabe der Zielverzeichnisse für die Erzeugung der Targets
- Optional Einbau in Enterprise Architect:
  - Aufruf des Menüpunkt Tools / Customize
  - Auswahl des Reiters ‚Tools‘, dort Neu-Anlegen eines Tools

- Einstellung des Namens, z.B.: ‚AAA-Katalogtool Basis-DLM‘
- Einstellungen:
  - ‚Command‘: Pfad zum ShapeChangeAAATools-Jar-File ("AAATools-1.0.0-SNAPSHOT.jar"). Eine lokale Pfadangabe unter Berücksichtigung der Einstellung unter ‚Initial directory‘ ist möglich.
  - ‚Arguments‘: -Xms64m -Xmx512m -d -c <config-file>

Bedeutung der Argumente:

-Xms64m -Xmx512m – zusätzliche Speicherallokation

-d – Aufruf des Dialogs

-c <config-file> – Angabe der zu verwendenden Konfigurationsdatei

Da die Größe des Dialogfensters begrenzt ist, ist in der Abbildung unten nicht der gesamte Pfad der Konfigurationsdatei erkennbar. Der komplette Eintrag lautet z. B.:

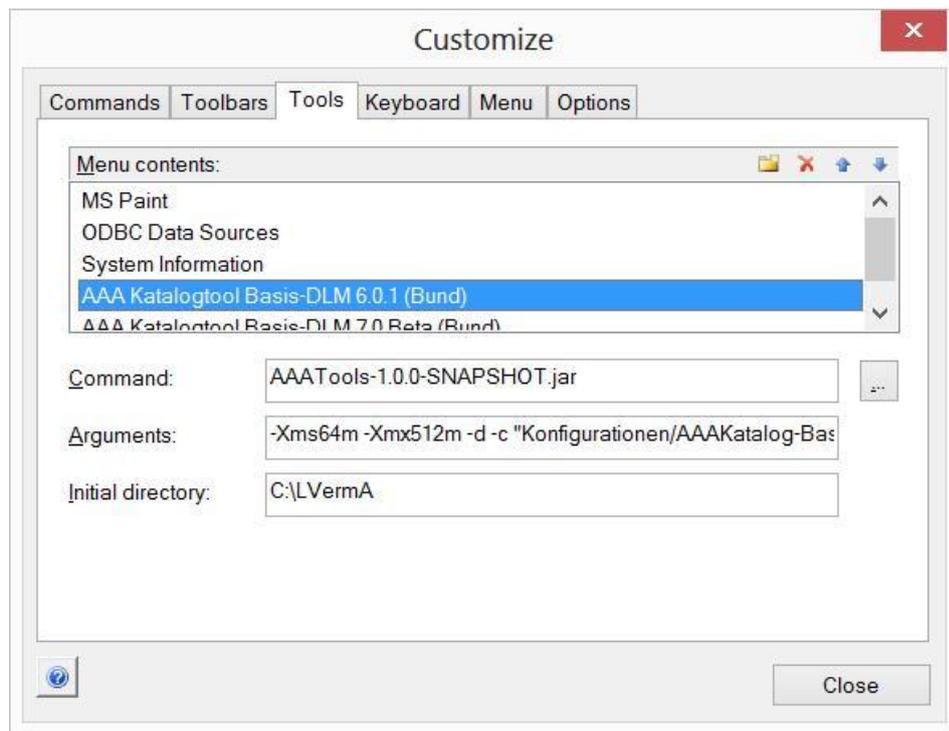
-Xms64m -Xmx512m -jar -d -c "Konfigurationen/AAAKatalog-ALKIS-6.0.1.xml"

Für die Angabe der Konfigurationsdatei ist eine lokale Pfadangabe unter Berücksichtigung der Einstellung unter ‚Initial directory‘ möglich.

- Einstellung ‚Initial directory‘: Das hier eingestellte Verzeichnis wird als initiales Arbeitsverzeichnis verwendet und muss das Installationsverzeichnis sein.

Weitere Hinweise:

- Bei individuell geänderten Pfaden zum ShapeChangeAAATools-Jar-File sind im Feld Command ggf. Slashes ("/") zu verwenden.
- Im Feld " Initial directory" sind immer Backslashes ("\") zu verwenden.
- Tipp: Für die Verwendung der AAA-Tools in EA sollte für jede Aufgabe ein eingenes EA-Tool eingerichtet werden.



- Tipp: Es ist sinnvoll, die AAA-Tools zuerst über die Batchdateien zu verwenden, die Konfigurationsdateien ggf. für eigene Belange zu modifizieren bzw. eigene Dateien anzulegen und dann erst die entsprechenden EA-Tools einzurichten.

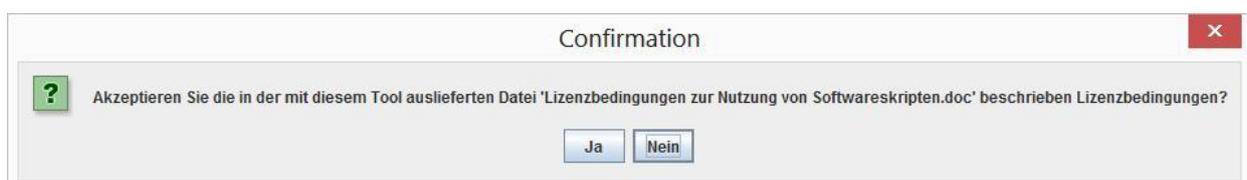
### 3 Ausführung

Die Ausführung der AAA-Tools erfolgt über Kommandozeileneingaben, Aufruf der Batchdateien aus der Kommandozeile, oder durch den Aufruf des entsprechenden Menüpunktes in EA im Menü "Tools". Von den in den Konfigurationsdateien eingetragenen Parametern können die meißten Parameter im Dialog zum Zeitpunkt der Ausführung verändert werden.

Grundsätzlich erzeugt das AAA-Katalogtool immer Kataloge, ggf. eingeschränkt auf den Inhalt eines Profils. Die Profile und Profilkennungen können mit dem AAA-Profiltool erzeugt und verwaltet werden.

#### 3.1 AAA-Katalogtool

Bei der Ausführung muss zuerst bestätigt werden, dass die Lizenzbedingungen (im Pfad "Dokumentation") akzeptiert werden. Ansonsten wird die Ausführung abgebrochen.



Anschließend wird der Dialog des AAA-Katalogtools geöffnet:

**Name des zu exportierenden Anwendungsschemas:**  
AFIS-ALKIS-ATKIS Anwendungsschema

**Eingeschränkt auf Paket:**  
AFIS-ALKIS-ATKIS Fachschema

**Ausgabeoptionen**

Liste der zu berücksichtigenden Schema-Kennungen  
(nur Klassen mit diesen Kennungen werden exportiert)  
\*

XML  HTML  RTF  Nart-PDF  GFC  CSV

Eigenschaften aus Superklassen auch in abgeleiteten Klassen darstellen

**Auswahl der Modellarten und Profile**

**Ausgewählte Modellarten:**  
Basis-DLM

Nur Grunddatenbestand exportieren

Eingeschränkt auf Profile:

Profile aus 3ap-Datei statt aus dem Modell laden

**Pfadangaben**

**Pfad in dem die XSLT-Skripte liegen:**  
http://shapechange.net/resources/xslt

**Pfad in den die Kataloge geschrieben werden:**  
C:\LVerMAAusgaben\Kataloge

**Pfad zum Modell:**  
C:\LVerMAAAA-6.0.1.eap

Process EAP View Log Exit

Ready

Das Beispiel zeigt die Einstellungen für die Erzeugung eines Objektartenkatalogs mit allen Elementen des AAA-Modells, die die Modellart "Basis-DLM" haben, ohne Ausgabe der Eigenschaften von Superklassen.

Die Steuerungsmöglichkeiten sind weitgehend selbsterklärend, sind aber im Kapitel 1.4.2 "ShapeChange Konfiguration" und folgende zusätzlich erläutert. Im weiteren wird hauptsächlich auf die Möglichkeiten eingegangen, die i. d. R. überwiegend verwendet werden.

Es muss immer mindestens eine Modellart angegeben werden. Dabei ist auf die formal richtige Schreibweise zu achten. Werden mehrere Modellarten angegeben, so sind diese durch jeweils ein Komma (",") zu trennen. Dabei dürfen keine Leerzeichen eingefügt werden.

Die Konfiguration "Nur Grunddatenbestand exportieren" beschränkt den Inhalt des Objektartenkatalogs auf die Elemente des AAA-Modells mit dieser Kennung. Diese Funktion entspricht der Verwendung eines Minimalprofils (siehe Kapitel 3.2. "AAA-Profiltool").

Die Konfiguration "Eingeschränkt auf folgende Profilkennungen im Modell:" beschränkt den Inhalt des Objektartenkatalogs auf die Elemente des AAA-Modells, die die anzugebene Profilkennung im AAA-Modell im Tagged Value: "AAA:Profile" enthalten. Eine Profilkennung kann direkt in EA, oder über das Profiltool (siehe Kapitel 3.2. "AAA-Profiltool") eingetragen worden sein. In dem Eingabefeld für die Profilkennung muss mindestens eine Kennung angegeben werden. Dabei ist auf die formal richtige Schreibweise zu achten. Werden mehrere Profilkennungen angegeben, so sind diese durch jeweils ein Komma (",") zu trennen. Dabei dürfen keine Leerzeichen eingefügt werden.

Die Konfiguration "Profile nur aus 3ap-Datei laden und verwenden statt aus dem Modell:" beschränkt den Inhalt des Objektartenkatalogs auf die Elemente des AAA-Modells, die in der entsprechenden Profildatei enthalten und nicht auskommentiert sind. Die Profilkennung(en) im AAA-Modell finden dann keine Beachtung. Die Profildatei muss im Ausgabeverzeichnis des zu erzeugenden Objektartenkatalogs stehen (bei Verwendung der Standardkonfiguration: "\Ausgaben\Kataloge"). Der Name der Profildatei wird automatisch aus der angegebenen Modellart und Profilkennung gebildet (z. B. "DLKM\_TestMIN.3ap").

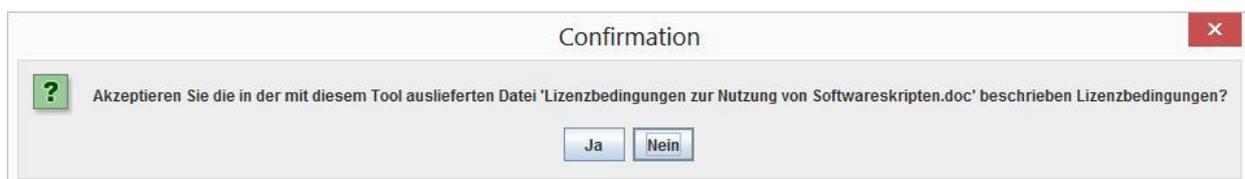
Die Pfadangaben dienen in erster Linie der Information. Die XLST-Skripte werden zur Laufzeit des AAA-Tools eingebunden, deshalb ist bei der Ausführung der Tools eine Internetverbindung erforderlich. Sollten sehr restriktive Netzeinstellungen das nicht erlauben, können die Skripte auch lokal abgelegt werden. Der Pfad zum Modell weist das verwendete AAA-Modell aus. Das ist besonders daher wichtig, da bei einer Einbindung der AAA-Tools in die Tools von EA das evtl. in EA geladene Modell nicht dem von dem AAA-Tool verwendeten Modell entsprechen muss.

Durch „Process EAP“ wird die Katalogerzeugung gestartet, mit „Exit“ wird die Ausführung abgebrochen. In der Statuszeile werden die jeweiligen Arbeiten des Tools angezeigt. Nach der erfolgreichen Erzeugung stehen im Ausgabepfad eine oder mehrere Dateien mit unterschiedlichen Suffix, aber dem immer gleichen Namen "aaa", z. B. "aaa.rtf" oder "aaa.xml". Die Ergebnisse sollten daher nach der Erzeugung umbenannt werden, um ein unbeabsichtigtes Überschreiben durch eine weitere Verwendung der Tools zu vermeiden. Nach der Erzeugung stehen im Arbeits-/Installationspfad Protokolldateien. Über die Schaltfläche "View Log" kann die HTML-Version sofort in den Standard-Browser geladen werden.

Alle Pfadangaben dürfen keine Leerzeichen beinhalten.

## 3.2 AAA-Profiltool

Bei der Ausführung muss zuerst bestätigt werden, dass die Lizenzbedingungen (im Pfad "Dokumentation") akzeptiert werden. Ansonsten wird die Ausführung abgebrochen.



Anschließend wird ein Dialog geöffnet, in dem die meisten der Konfigurationsparameter angepasst werden können:

Es gibt 2 Möglichkeiten, wie der Umfang der Kataloge eingeschränkt werden kann: über Profile oder über Profilkennungen.

Profile sind ASCII-Textdateien mit allen Elementen eines AAA-Modells. In diesen Dateien werden die Elemente, die nicht ausgegeben werden sollen auskommentiert. Der Name der Dateien setzt sich aus der Modellart und der Profilkennung zusammen, im Beispiel: "DLKM\_TestMIN.3ap". Die Profilkennung darf keine Leerzeichen enthalten. Weiter Informationen zum Aufbau sind im Kapitel: 1.4.2.5.3 "Target-Konfiguration: AAA-Profiltool" beschrieben. Ein Minimalprofil beinhaltet alle Elementen eines AAA-Modells, jedoch sind nur Elemente des Grunddatenbestandes nicht auskommentiert. Im Gegenzug beinhaltet ein Maximalprofil alle Elementen eines AAA-Modells ohne jede Auskommentierung. Die Min/Max-Profile dienen als Grundlage für eigene Anpassungen.

Profilkennungen können mit dem Profil-Tool über eine Profildatei in das AAA-Modell im Tagged Value: "AAA:Profile" eingetragen/gelöscht werden. Die Eintragungen/Löschungen können aber auch direkt in EA vorgenommen werden. Bei der Verwendung von Profilkennungen muss EA in einer Vollversion vorliegen ("EA-Lite" reicht nicht!), da im Modell geschrieben werden muss."

Das Profil-Tool lässt grundsätzlich alle denkbaren Kombinationsmöglichkeiten zu, die sich aus den Wahlmöglichkeiten der Dialogmaske ergeben, ggf. auch unsinnige Kombinationen.

Durch „Process EAP“ wird die Katalogerzeugung gestartet, mit „Exit“ wird die Ausführung abgebrochen.

Sofern ein Profil aus einer Datei eingelesen wird, so muss diese den Namen

**<Modellart>\_<Profilkennung>.3ap**

besitzen, wobei <Modellart> und <Profilkennung> der Kurzname der Modellart (Parameter „Modellart“ in der Konfigurationsdatei) bzw. die Kennung des Profils in dieser Modellart (Parameter „Profil“ in der Konfigurationsdatei) sein. Diese Angaben müssen zu den Angaben in der Datei passen. Beispiel für einen Dateinamen: **DLKM\_TestMin.3ap**

Die Datei muss außerdem im angegebenen Pfad (Parameter „Verzeichnis“ in der Konfigurationsdatei) liegen, im obigen Beispiel **C:\AAA\Tools\Ausgaben\Profile**.

Alle Pfadangaben dürfen keine Leerzeichen beinhalten.

Bei der Übernahme von Profilen, die mit dem Rational-Rose-AAA-Profiltool erzeugt wurden sind die folgenden Punkte zu beachten:

- Es werden typischerweise eine Reihe von Fehlern angezeigt, dass bestimmte Elemente nicht dem Profil zugeordnet werden können, das sie nicht Teil des Anwendungsschemas sind. Diese Meldungen sind korrekt, diese Aufnahme dieser Einträge in Profildateien im Rational-Rose-Tool war überflüssig. Es gibt zwei typische Situationen:
  - o Die Profildatei beinhaltet Einträge zu Klassen aus anderen Schemata, insbesondere den Schemata aus der ISO 19100-Normenserie. Beispiele:
    - „Objektart/Datentyp GM\_MultiSurface kann dem Profil nicht zugeordnet werden, die Klasse wurde nicht im Anwendungsschema gefunden.“
    - „Objektart/Datentyp Measure kann dem Profil nicht zugeordnet werden, die Klasse wurde nicht im Anwendungsschema gefunden.“
  - o Die Profildatei beinhaltet Einträge zu Relationsarten, die nicht navigierbar sind. Ob eine Relation Teil eines Profils ist oder nicht wird stets durch die navigierbare Richtung festgelegt. Eine unterschiedliche Festlegung pro Rolle ist nicht möglich. Beispiele:
    - „Relationsart bestehtAus zu Objektart/Datentyp AA\_ZUSO kann dem Profil nicht zugeordnet werden, die Eigenschaft wurde nicht im Anwendungsschema gefunden.“
    - „Relationsart traegtBeiZu zu Objektart/Datentyp AA\_REO kann dem Profil nicht zugeordnet werden, die Eigenschaft wurde nicht im Anwendungsschema gefunden.“
- Es gibt einige Wertarten, die Umlaute enthalten. Dies kann – je nachdem mit welchen Editoren die Profile bearbeitet wurden – zu Fehlermeldungen führen.

### 3.3 NAS-Tool

Die Besonderheit bei der Ausführung des NAS-Tools ist, dass das AAA-Modell durch die Ausführung irreparabel geändert wird. Dies erfolgt im Rahmen der Modelltransformation vom konzeptuellen Modell in das NAS-spezifische Modell. Siehe GeoInfoDok für Details.

Dies bedeutet, dass die Ausführung auf einer temporären Kopie des AAA-Modell durchgeführt werden sollte. Diese EAP-Datei sollte nach der Ausführung gelöscht werden.

Die Ausführung erfolgt ohne Dialog.

### 3.4 Störmerkmale

Meldungen aus den Tools entstehen bei Verarbeitung des im Enterprise Architect (EA) Repository gespeicherten Modells durch ShapeChange.

Sie entstehen in verschiedenen Phasen der Verarbeitung, nämlich

1. bei der Übernahme des Modells aus der EA Automation-Schnittstelle,
2. bei der Erzeugung der diversen Zielkodierungen.

Bei der Übernahme des Modells von EA wird ein Großteil der Meldungen erkannt, aber nicht alle. Diese treten erst bei der Erzeugung der einzelnen Targets auf.

Das liegt daran, dass das Modell Zusatzinformationen (meist als *tagged values*) enthält, die im allgemeinen Modell irrelevant sind und auch z.B. bei der Ableitung von GML nicht gebraucht werden.

Die möglichen Meldungen von ShapeChange sind unter <http://shapechange.net/get-started/messages/> dokumentiert. Zusätzlich können aus den AAA-Tools-Erweiterungen die folgenden Meldungen erzeugt werden.

#### 3.4.1 Meldungen bei der Katalogerzeugung

Schwere	Meldungstext	Hinweise
Fatal	<a href="#">XML Parser was unable to be configured.</a>	
Error	<a href="#">Die Profilquelle '\$1\$' ist unbekannt, es wird 'Modell' verwendet.</a>	
Error	<a href="#">File \$1\$ is not readable, processing of \$2\$ is skipped.</a>	
Warning	<a href="#">Warning while transforming \$1\$. Message: \$2\$"</a>	Weiterleitung von Meldungen des Apache FOP
Error	<a href="#">Error while transforming \$1\$. Message: \$2\$"</a>	Weiterleitung von Meldungen des Apache FOP
Fatal	<a href="#">Fatal error while transforming \$1\$. Message: \$2\$"</a>	Weiterleitung von Meldungen des Apache FOP
Warning	<a href="#">Klasse mit unbekannter Art wurde nicht im Profil verarbeitet: \$1\$</a>	
Error	<a href="#">Export der Profildatei nicht möglich, da das Profil noch nicht definiert ist.</a>	
Error	<a href="#">Problem beim Schreiben von Profildatei: \$1\$</a>	

Error	Problem beim Lesen von Profildatei: '\$1\$'	
Error	Die Profildatei beginnt nicht mit dem erwarteten Wert 'AFIS-ALKIS-ATKIS-Profildefinition'. Der Ladevorgang wurde abgebrochen.	
Error	Es wurde keine Version angegeben. Der Ladevorgang wurde abgebrochen.	
Error	Die Version '\$1\$' des Profils passt nicht zur erwarteten Version '\$2\$'. Der Ladevorgang wurde abgebrochen.	
Error	Es wurde kein Profilname angegeben. Der Ladevorgang wurde abgebrochen.	
Error	Zeile mit unerlaubter Syntax: '\$1\$'	
Error	Objektart/Datentyp '\$1\$' kann dem Profil nicht zugeordnet werden, da es kein Bestandteil der betreffenden Modellart ist.	
Error	Objektart/Datentyp '\$1\$' kann dem Profil nicht zugeordnet werden, die Klasse wurde nicht im Anwendungsschema gefunden.	
Error	Attributart/Relationsart '\$1\$' zu Objektart/Datentyp '\$2\$' kann dem Profil nicht zugeordnet werden, die Eigenschaft wurde nicht im Anwendungsschema gefunden.	
Error	Attributart/Relationsart '\$1\$' zu Objektart/Datentyp '\$2\$' kann dem Profil nicht zugeordnet werden, da sie kein Bestandteil der betreffenden Modellart ist.	

### 3.4.2 Meldungen bei der Profilverarbeitung

Schwere	Meldungstext	Hinweise
Fatal	Der Parameter 'Ziel' fehlt. Die Ausführung des Profiltools wird abgebrochen	
Fatal	Der Parameter 'Quelle' fehlt. Die Ausführung des Profiltools wird abgebrochen	
Fatal	Der Parameter 'Modellart' oder 'Profil' fehlt, das Erzeugen eines Profils ist nicht möglich. Die Ausführung des Profiltools wird abgebrochen	
Fatal	Der Parameter 'Quelle' hat den unbekanntes Wert '\$1\$'. Die Ausführung des Profiltools wird abgebrochen	
Fatal	Der Parameter 'Ziel' hat den unbekanntes Wert '\$1\$'. Die Ausführung des Profiltools wird abgebrochen	

### 3.4.3 Meldungen bei der Modelltransformation bei der NAS-Erzeugung

Schwere	Meldungstext	Hinweise
Fatal	Das AAA-Anwendungsschema ist von der falschen Version. Erwartet wurde die Version 6.0, gefunden wurde die Version '\$1\$'	
Error	Fehler beim Löschen von hängender Relation in '\$1\$': '\$2\$'	
Error	Fehler beim Aktualisieren von inverser Rolle in '\$1\$' zu '\$2\$': '\$3\$'	
Error	Package '\$1\$' nicht gefunden.	
Error	Klasse '\$1\$' nicht gefunden	
Error	Fehler beim Clonen von Attribut '\$1\$': '\$2\$'	
Error	Fehler beim Clonen von Relation '\$1\$'/'\$2\$' - '\$3\$': '\$4\$'	
Error	Fehler beim Löschen von Package '\$1\$': '\$2\$'	
Error	Fehler beim Verschieben von Klasse '\$1\$': '\$2\$'	
Error	Fehler beim Löschen von Klasse '\$1\$': '\$2\$'	
Error	Fehler beim Löschen der Methoden von Klasse '\$1\$': '\$2\$'	
Error	Fehler beim Setzen des Stereotyps von Klasse '\$1\$': '\$2\$'	
Error	Fehler beim Setzen von Tagged Value '\$1\$'-'\$2\$': '\$3\$'	
Error	Fehler beim Setzen von Attribut '\$1\$'.'\$2\$': '\$3\$'	
Error	Fehler beim Setzen des Typs von Attribut '\$1\$'.'\$2\$': '\$3\$'	
Error	Fehler beim Setzen der Multiplizität von Attribut '\$1\$'.'\$2\$': '\$3\$'	
Error	Fehler beim Löschen von Attribut '\$1\$'.'\$2\$': '\$3\$'	

Error	Fehler beim Löschen von Rolle '\$1\$'. '\$2\$': '\$3\$'	
Fatal	EA-Fehler: '\$1\$'	