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: 1.1.0

Status: Release

Stand: 15.02.2018

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 |
| 0.7 | In Bearbeitung | 15.10.2014 | Clemens Portele | Ergänzung der Modellartsetzung beim Profiltool |
| 1.0 | In Bearbeitung | 18.12.2014 | Clemens Portele | Dokumentation zu oclConstraintTypeRegex ergänzt; Version für den Release der GeoInfoDok 7.0.1 aktualisiert |
| 1.1 | In Bearbeitung | 08.05.2015 | Clemens Portele | Aktualisierung NAS-Tool für 7.0  Release der AAA-Tools 1.0.0 |
| 1.2 | In Bearbeitung | 19.06.2015 | Clemens Portele | Korrekturen nach Review durch Herrn Böhmer  Ergänzung der CSV-Ausgabe für den Nutzungsartenkatalog  Release der AAA-Tools 1.0.2 |
| 1.3 | In Bearbeitung | 05.04.2016 | Ramona Kurstedt | Korrekturen nach Praxistest durch Frau Kurstedt |
| 1.4 | In Bearbeitung | 12.08.2016 | Clemens Portele | Ausgabe von Tagged Values in Objektartenkatalogen |
| 1.1.0 |  | 15.02.2018 | Clemens Portele | Aktualisierung für GeoInfoDok 7.0.3  Benennung der Version des Dokuments an die Version der Tools |
| 1.1.1 |  | 31.01.2019 | Clemens Portele | Aktualisierung für die GeoInfoDok 7.1.0 |

Inhaltsverzeichnis

1 Komponenten und Artefakte 4

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 16

2 Installation 16

3 Ausführung 18

3.1 AAA-Katalogtool 19

3.2 AAA-Profiltool 21

3.3 NAS-Tool 24

3.4 Störmerkmale 25

# Komponenten und Artefakte

## Übersicht

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

## AAA-Modell

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

## Enterprise Architect (EA)

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

**Aufgrund einiger Änderungen in den letzten Programmversionen beim Umgang mit dem UML-Metamodell in EA, wird die Verwendung der Version 12.0 oder später empfohlen.**

## ShapeChange

### 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-basierten 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".

### 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|7.1.0|7.1.0-Aenderungen).xml** (AAA-Katalogtool für den ALKIS-Objektartenkatalog)

**AAAKatalog-AFIS-(6.0.1|7.1.0|7.1.0-Aenderungen).xml** (AAA-Katalogtool für den AFIS-Objektartenkatalog)

**AAAKatalog-BasisDLM-(6.0.1|7.1.0|7.1.0-Aenderungen).xml** (AAA-Katalogtool für den ATKIS-Basis-DLM-Objektartenkatalog)

**AAAKatalog-DLM50-(6.0.1|7.1.0|7.1.0-Aenderungen).xml** (AAA-Katalogtool für den ATKIS-DLM50-Objektartenkatalog)

**AAAKatalog-AS-7.1.0.xml** (AAA-Katalogtool für alle Katalogausgaben der GeoInfoDok 7.1.0)

**AAAProfil-(6.0.1|7.1.0).xml** (AAA-Profiltool)

**NAS-(6.0.1|7.1.0).xml** (NAS-Tool für das AAA-Anwendungsschema)

**NAS-(AK|GV|LB|LN)-7.1.0.xml** (NAS-Tool für die weiteren Schemata)

Diese beziehen sich teilweise auf weitere in XML-Dokumenten gekapselte Teile, die per Xinclude referiert werden. Seit der Version 1.0.1 der Tools wird nicht mehr auf Dokumente über das Web zugegriffen, sondern ausschliesslich lokal im Verzeichnis „resources/config“:

**resources/config/StandardMapEntries\_<Quelle>.xml**

wobei <Quelle> durch den zugrundeliegenden Standard zu ersetzen ist.

Zur Konfigurationsdatei gehört ein XML-Schema-Dokument namens

**resources/schema/ShapeChangeConfiguration.xsd**.

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

Zusätzlich sind auch einige Batchdateien zur Ausführung in der "DOS-Box" enthalten. Diese verwenden das Modell für die GeoInfoDok 7.1.0, können aber leicht für andere Konfigurationen selbst angepasst werden.

#### 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-Applikationschema-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>

#### 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-7.1.0.EAP"/> <parameter name="appSchemaName" value="AFIS-ALKIS-ATKIS Anwendungsschema"/> <parameter name="excludedPackages" value="ISO/TC 211,OGC,AAA\_Ausgabekatalog,AAA\_Objektartenkatalog,AAA Versionierungsschema,Web Feature Service Erweiterungen,AAA\_Signaturenkatalog"/> <parameter name="publicOnly" value="true"/> <parameter name="checkingConstraints" value="enabled"/> <parameter name="oclConstraintTypeRegex" value=""/> <parameter name="addTaggedValues" value="AAA:Kennung,AAA:Datum,AAA:Organisation,AAA:Modellart,AAA:Profile,AAA:Grunddatenbestand,AAA:Nutzungsart,AAA:Nutzungsartkennung,AAA:objektbildend,AAA:Themen,AAA:GueltigBis"/> <parameter name="representTaggedValues" value="Status"/> <stereotypeAliases> <StereotypeAlias wellknown="Application Schema" alias="applicationSchema" /> <StereotypeAlias wellknown="Application Schema" alias="schema" /> <StereotypeAlias wellknown="Application Schema" alias="requirementsClass" /> <StereotypeAlias wellknown="Application Schema" alias="ApplicationSchema" /> <StereotypeAlias wellknown="Application Schema" alias="Schema" /> <StereotypeAlias wellknown="Type" alias="interface" /> </stereotypeAliases> </input>

Eine sinnvolle Möglichkeit zum Eingriff ergibt sich bei folgenden Namen:

| ***name*** | **default-*value*** | **Bedeutung** |
| --- | --- | --- |
| inputModelType | Pflichtfeld | Der Wert ist für das AAA-Modell stets "EA7", d.h. es erfolgt ein Direktzugriff auf 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. |
| excludePackages | Kein Paket | Alle aufgeführten Pakete werden ignoriert und gar nicht gelesen. |
| 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. |
| oclConstraintTypeRegex | (OCL|Invariant) | Die Constraint-Typen von Enterprise Architect, bei denen der Constraint als OCL-Constraint behandelt werden soll. Da das AAA-Modell aktuell keine verarbeitbaren OCL-Constraints enthält, sollte der Wert auf "" gesetzt werden. |
| addTaggedValues | keiner | Die im UML-Profil des AAA-Modells definierten Tagged Values, die für die AAA-Tools relevant sind. |
| representTaggedValues | keiner | Tagged Values, die zusätzlich in den Objektartenkatalogen ausgegeben werden sollen. |
| 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 gepräfixt. |
| sortedOutput | false | "true" führt zur Verarbeitung der Pakete und Klassen in alphabetischer Reihenfolge. |

#### 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:

| ***name*** | **default-*value*** | **Bedeutung** |
| --- | --- | --- |
| 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. |

#### 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:

| ***name*** | **default-*value*** | **Bedeutung** |
| --- | --- | --- |
| reportLevel | INFO | Möglich sind INFO, WARNING, ERROR. |
| logFile | leer | Der Dateiname der Datei für Meldungen. |

#### Konfigurationselement: targets

##### 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"/> <rules> <EncodingRule name="NAS" extends="gml33"> <rule name="rule-xsd-all-no-documentation"/> <rule name="req-all-prop-sequenceNumber"/> </EncodingRule> </rules> <xmlNamespaces> <XmlNamespace nsabr="sc" ns="http://www.interactive-instruments.de/ShapeChange/AppInfo" location="http://shapechange.net/resources/schema/ShapeChangeAppinfo.xsd" /> <XmlNamespace nsabr="xsi" ns="http://www.w3.org/2001/XMLSchema-instance"/> <XmlNamespace nsabr="xlink" ns="http://www.w3.org/1999/xlink" location="http://www.w3.org/1999/xlink.xsd" /> <XmlNamespace nsabr="gml" ns="http://www.opengis.net/gml/3.2" location="http://schemas.opengis.net/gml/3.2.1/gml.xsd" /> <XmlNamespace nsabr="gmlexr" ns="http://www.opengis.net/gml/3.3/exr" location="http://schemas.opengis.net/gml/3.3/extdEncRule.xsd" /> <XmlNamespace nsabr="gmd" ns="http://www.isotc211.org/2005/gmd" location="http://schemas.opengis.net/iso/19139/20070417/gmd/gmd.xsd" /> <XmlNamespace nsabr="gco" ns="http://www.isotc211.org/2005/gco" location="http://schemas.opengis.net/iso/19139/20070417/gco/gco.xsd" /> <XmlNamespace nsabr="wfsext" ns="http://www.adv-online.de/namespaces/adv/gid/wfsext/2.0" location="http://repository.gdi-de.org/schemas/adv/wfsext/2.0/WFS-Erweiterungen.xsd"/> <XmlNamespace nsabr="ows" ns="http://www.opengis.net/ows/1.1" location="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/> <XmlNamespace nsabr="wfs" ns="http://www.opengis.net/wfs/2.0" location="http://schemas.opengis.net/wfs/2.0/wfs.xsd"/> <XmlNamespace nsabr="fes" ns="http://www.opengis.net/fes/2.0" location="http://schemas.opengis.net/filter/2.0/filterAll.xsd"/> <XmlNamespace nsabr="gmlcov" ns="http://www.opengis.net/gmlcov/1.0" location="http://schemas.opengis.net/gmlcov/1.0/gmlcovAll.xsd"/> </xmlNamespaces> <xi:include href="resources/config/StandardMapEntries\_iso19136\_2007.xml"/> <xi:include href="resources/config/StandardMapEntries\_iso19139\_2007.xml"/> <xi:include href="resources/config/StandardMapEntries\_iso19107.xml"/> <xi:include href="resources/config/StandardMapEntries\_iso19108.xml"/> <xi:include href="resources/config/StandardMapEntries\_iso19111.xml"/> <xi:include href="resources/config/StandardMapEntries\_iso19115.xml"/> <xi:include href="resources/config/StandardMapEntries\_iso19123.xml"/> <xi:include href="resources/config/StandardMapEntries\_gmlcov.xml"/> <xsdMapEntries> <XsdMapEntry type="URI" xsdEncodingRules="NAS" xmlPropertyType="anyURI" xmlType="anyURI" xmlTypeType="simple" xmlTypeContent="simple"/> <XsdMapEntry type="URL" xsdEncodingRules="NAS" xmlPropertyType="anyURI" xmlType="anyURI" xmlTypeType="simple" xmlTypeContent="simple"/> <XsdMapEntry type="doubleList" xsdEncodingRules="NAS" xmlPropertyType="gml:doubleList" xmlType="gml:doubleList" xmlTypeType="simple" xmlTypeContent="simple"/> <XsdMapEntry type="doubleOrNilReasonList" xsdEncodingRules="NAS" xmlPropertyType="gml:doubleOrNilReasonList" xmlType="gml:doubleOrNilReasonList" xmlTypeType="simple" xmlTypeContent="simple"/> <XsdMapEntry type="AC\_FeatureCatalogue" xsdEncodingRules="NAS" xmlPropertyType="gml:ReferenceType" xmlType="gml:ReferenceType"/> <XsdMapEntry type="AA\_Antragsart" xsdEncodingRules="NAS" xmlPropertyType="gml:ReferenceType" xmlType="gml:ReferenceType"/> <XsdMapEntry type="AA\_Projektsteuerungsart" xsdEncodingRules="NAS" xmlPropertyType="gml:ReferenceType" xmlType="gml:ReferenceType"/> <XsdMapEntry type="AA\_Vorgangsart" xsdEncodingRules="NAS" xmlPropertyType="gml:ReferenceType" xmlType="gml:ReferenceType"/> <XsdMapEntry type="AA\_Aktivitaetsart" xsdEncodingRules="NAS" xmlPropertyType="gml:ReferenceType" xmlType="gml:ReferenceType"/> <XsdMapEntry type="CapabilitiesBase" xsdEncodingRules="NAS" xmlElement="" xmlType="ows:CapabilitiesBaseType"/> <XsdMapEntry type="Identification" xsdEncodingRules="NAS" xmlElement="" xmlType="ows:IdentificationType"/> <XsdMapEntry type="Query" xsdEncodingRules="NAS" xmlType="wfs:QueryType" xmlElement="wfs:Query" xmlPropertyType="\_P\_"/> <XsdMapEntry type="FeatureCollection" xsdEncodingRules="NAS" xmlType="wfs:FeatureCollectionType" xmlElement="wfs:FeatureCollection" xmlPropertyType="\_P\_"/> <XsdMapEntry type="Transaction" xsdEncodingRules="NAS" xmlType="wfs:TransactionType" xmlElement="wfs:Transaction" xmlPropertyType="\_P\_"/> <XsdMapEntry type="Filter\_Capabilities" xsdEncodingRules="NAS" xmlElement="fes:Filter\_Capabilities" xmlPropertyType="\_P\_"/> <XsdMapEntry type="TM\_Primitive" xsdEncodingRules="NAS" xmlType="gml:AbstractTimeGeometricPrimtiveType" xmlElement="gml:AbstractTimeGeometricPrimtive" xmlPropertyType="gml:TimeGeometricPrimtiveProertyType" /> <XsdMapEntry type="CI\_Responsibility" xsdEncodingRules="NAS" xmlType="gmd:CI\_ResponsibleParty\_Type" xmlElement="gmd:CI\_ResponsibleParty" xmlPropertyType="\_MP\_" /> <XsdMapEntry type="LI\_ProcessStep" xsdEncodingRules="NAS" xmlElement="gmd:LI\_ProcessStep" xmlPropertyType="\_MP\_" /> <XsdMapEntry type="AX\_Fortfuehrungsnachweis" xsdEncodingRules="NAS" xmlElement="gml:AbstractObject" xmlPropertyType="\_P\_" /> </xsdMapEntries> </TargetXmlSchema>

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

Die Parameter sind wie folgt:

| ***name*** | **default-*value*** | **Bedeutung** |
| --- | --- | --- |
| 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 (bis GeoInfoDok 6) bzw. gml33 (ab GeoInfoDOk 7) mit der Erweiterung, dass die Dokumentation nicht in das XML Schema abgebildet wird. |
|  |  | Die anzuwendende Encoding Rule wird im Modell für jedes Element gesetzt (einschließelich der Ausgabe von ISO/TS 19139 konformen Metadatenelementen). |
| rules/EncodingRule | keine zusätzliche Encoding Rule | Definition von Encoding Rules, die nicht bereits in ShapeChange standardmäßig vordefiniert sind. |

Die Elemente <xmlNamespaces> und <xsdMapEntries> enthalten XML-Namespace-Definitionen und vordefinierte XML-Schema-Abbildungen. Die per XInclude referierten Dokumente stehen für weitere XML-Schema-Abbildungen von großen Teilen des ISO 19100 Modells in entsprechende GML-Elemente.

##### 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="stillgelegteElemente" value="false"/>

<targetParameter name="schemakennungen" value="\*"/>

<targetParameter name="ausgabeformat" value="HTML,RTF,XML"/>

<targetParameter name="notesRuleMarker" value="-==-"/>

</Target>

Die Parameter sind wie folgt:

| ***name*** | **default-*value*** | **Bedeutung** |
| --- | --- | --- |
| 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 Katalogerzeugung angegeben werden. |
| profile | kein Profil | Zusätzlich zur Beschränkung der Modellarten kann die Katalogerzeugung auch auf ein oder mehrere Profile (erzeugt mit dem AAA-Profiltool) eingeschränkt werden. Im Parameter wird der Name der Profile angegeben. |
| profilquelle | Modell | *Datei* oder *Modell*. Bei *Modell* wird auf das entsprechende Tagged Value im Modell zugegriffen, bei *Datei* auf ein Profildatei; das Tool versucht hierzu im Verzeichnis gemäß Parameter „Verzeichnis“ auf die Datei mit dem Namen des Profils und der Dateierweiterung „.3ap“ zuzugreifen. |
| nurGrunddatenbestand | false | *true* oder *false*. Durch *true* wird die Katalogerzeugung auf den Grunddatenbestand der jeweiligen Modellarten eingeschränkt werden. |
| letzteAenderung | false | *true* oder *false*. Schalter für die Ausgabe der Angabe zu letzten Änderung pro Modellelement (Version, Revisionsnummer). |
| geerbteEigenschaften | false | *true* oder *false*. 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. |
| stillgelegteElemente | false | *true* oder *false*. Über den Schalter kann festgelegt werden, ob auch als stillgelegt gekennzeichnete Modellelemente (Klassen, Attribute, Relationsrollen) aufgeführt werden sollen. |
| schemakennungen | \* | In den ISO 19100-Normen werden verschiedene Namensräume durch zweibuchstabige 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 |
| 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 |
| xslnartcsvFile | aaa-nart-csv.xsl | XSLT-Stylesheet für die Ausgabe des Nutzungsartenverzeichnisses als CSV-Datei |
| 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 Funktionaliät ist nur bei der HTML-Ausgabe möglich, ansonsten sollte kein Vorgängermodell angegeben werden.  Dieser Parameter darf nur in Verbindung mit HTML-Ausgaben und unter Verwendung der Defaultwerte der anderen Parameter (d.h. z.B. ohne Anwendung eines Profils) angewendet werden. Ausgenommen sind nur die Parameter "paket", "modellarten" und "xsltPfad". |
| 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.  Dieser Parameter darf nur in Verbindung mit HTML-Ausgaben und unter Verwendung der Defaultwerte der anderen Parameter (d.h. z.B. ohne Anwendung eines Profils) angewendet werden. Ausgenommen sind nur die Parameter "paket", "modellarten" und "xsltPfad". |

##### 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, Wertearten) 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:

| ***name*** | **default-*value*** | **Bedeutung** |
| --- | --- | --- |
| Verzeichnis | . | Verzeichnis für die Profildateien. Der Pfad darf keine Leerzeichen beinhalten. |
| Modellart | Pflichtfeld | Modellart für das Profil |
| Profil | kein Profil | Name des Profils innerhalb der Modellart.  Der vollständige Name des Profils ergibt sich aus der Kombination von Modellart, einem Unterstrich und diesem Parameter.  Sofern der Parameter leer ist, wird vom Profiltool nicht das Tagged Value "AAA:Profil" exportiert/gesetzt, sondern "AAA:Modellart". D.h. das Profiltool kann auch zur Pflege weiterer, länderspezifischer Modellarten genutzt werden. |
| Quelle | Pflichtfeld | Bezeichnet die Quelle der Profildefinition.  *Datei*: 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“. Ist der Parameter "Profil" nicht gesetzt, dann ist der Name der Wert des Parameters „Modellart“, gefolgt von „.3ap“.  *Modell*: Die Profildefinition wird aus dem AAA-Modell geladen.  *Neu\_Minimal*: Ein neues Profil wird erzeugt, auf das Minimalprofil eingeschränkt. Nur erlaubt, wenn der Parameter "Profil" gesetzt ist.  *Neu\_Maximal*: Ein neues Profil wird erzeugt – mit allen zu der Modellart gehörenden Modellelementen. Nur erlaubt, wenn der Parameter "Profil" gesetzt ist. |
| Ziel | Pflichtfeld | Bezeichnet das Ziel der Profildefinition.  *Datei*: Das Profil wird in eine Datei geschrieben. Der Dateiname ist eine Kombination des Parameters „Modellart“, einem Unterstrich, dem Parameter „Profil“ und „\_\_export.3ap“.  *Modell*: Die Profildefinition wird in das AAA-Modell geschrieben.  *DateiModell*: Die Profildefinition wird sowohl in die Datei als auch das Modell geschrieben.  *Ohne*: Es erfolgt keine Ausgabe des Profils. |

Das Profiltool ist so ausgelegt, dass bei sauberer Definition der Modellarten stets ein konsistenter Profilzustand 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 Umbennenungen 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 Wertearten 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 Werteart) 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 --- Werteart

# AA\_AdVStandardModell/DKKM500 --- Werteart

# AA\_AdVStandardModell/DKKM1000 --- Werteart

...

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 Werteart der Name der Codeliste und der Code. Bei einer Relationsart schließlich wird der Name der Objektart und der Name der Relationsart angegeben.

### 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.*

Die Meldungen werden in der Logdatei dokumentiert.

# 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, 7 oder 10
* Enterprise Architect (EA) der Firma SparxSystems Ltd, Version 12.0 oder höher.
* Java Laufzeitsystem, Version 1.8 oder höher.

Weitere Hinweise:

* Die aktuellen Tools können von den Seiten der [AdV](http://www.adv-online.de/AAA-Modell/Dokumente-der-GeoInfoDok/) oder von [shapechange.net](http://shapechange.net/examples/#AFIS-ALKIS-ATKIS) heruntergeladen werden.
* Für die Ausführung der Tools ist eine Vollversion von EA erforderlich (weiters 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 die aktuelle AAA-Modelle der GeoInfoDok 6.0.1 und GeoInfoDok 7.1.0.

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-6.0.1.eap (für die GeoInfoDok 6.0.1) bzw. AAA-7.1.0.eap (für die GeoInfoDok 7.1.0) 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 (hier am Beispiel der Version 13):
  + Aufruf des Menüpunkt Start > Workspace > Preferences > Customize
  + Auswahl des Reiters ‚Tools’, dort Neu-Anlegen eines Tools
  + Einstellung des Namens, z.B.: ‚AAA-Katalogtool ALKIS’
  + Einstellungen:
    - ‚Command’: Pfad zum AAATools-Jar-File (z.B. "AAATools-1.0.2.jar"). Eine lokale Pfadangabe unter Berücksichtigung der Einstellung unter ‚Initial directory’ ist möglich.
    - ‚Arguments’: -d -c <config-file>  
        
      Bedeutung der Argumente:  
      -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.:

-d -c "Konfigurationen/AAAKatalog-ALKIS-7.1.0.xml"

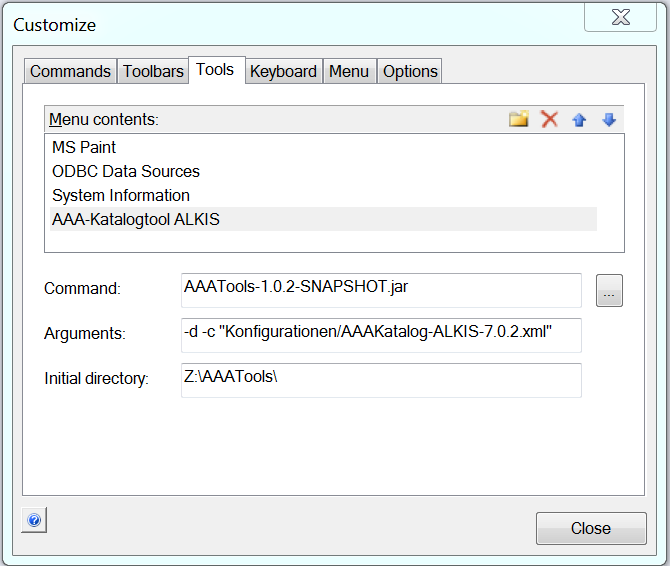
Für die Angabe der Konfigurationsdatei ist eine lokale Pfadangabe unter Berücksichtigung der Einstellung unter ‚Initial directory’ möglich.

* + - Sofern Probleme mit dem Speicher bei der Ausführung auftreten, ist der Speicher ggf. zu erhöhen, zB durch die Parameter „-Xms64m -Xmx512m“. In den meisten Umgebungen dürften diese nicht mehr erforderlich sein.
    - 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. Slashs ("/") zu verweden.
* Im Feld " Initial directory" sind immer Backslashs ("\") zu verwenden.
* Tipp: Für die Vewendung der AAA-Tools in EA sollte für jede Aufgabe ein eingenes EA-Tool eingerichtet werden.

Beispiel-Ansicht (die Versionsangabe im Namen des JAR-Files in „Command“ ist entsprechend der aktuellen Version anzupassen):



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

# Ausführung

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

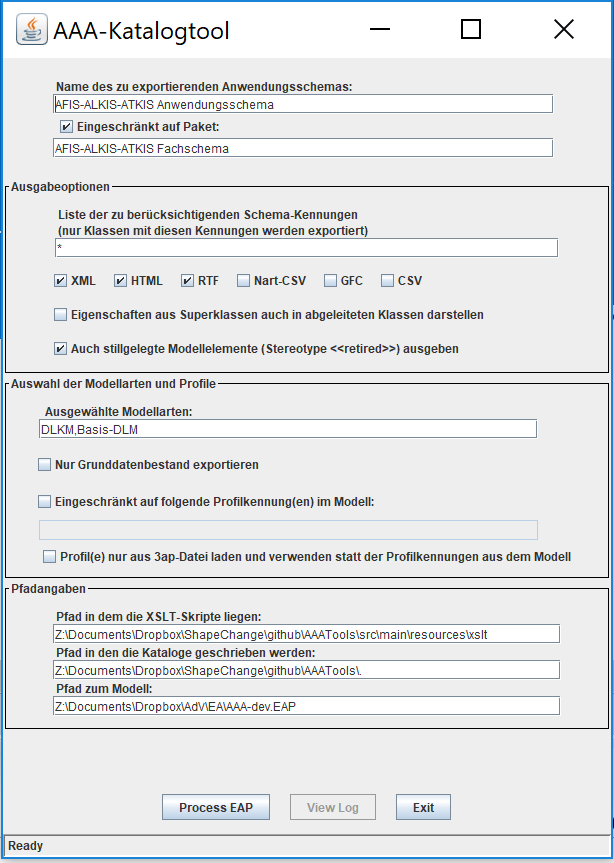
Grundsätzlich erzeugt das AAA-Katalogtool immer Kataloge, ggf. eingeschränkt auf den Inhalt eines Profils. Die Profile und Profilkennungen können mit dem AAA-Profiltool erzeugt und verwaltet werden.

## 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:



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

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.

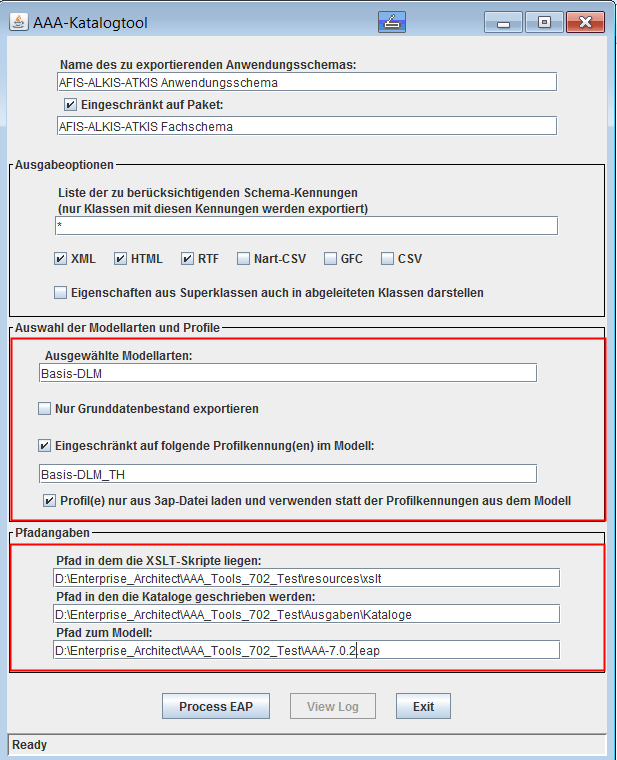
Es muss immer mindestens eine Modellart angegeben werden. Dabei ist auf die formal richtige Schreibweise zu achten. Werden mehrere Modellarten angegeben, so sind diese durch jeweils ein Komma (",") zu trennen. Dabei dürfen keine Leerzeichen eingefügt werden.

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 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 Objektartenkatatolgs 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 XSLT-Skripte werden zur Laufzeit des AAA-Tools eingebunden, deshalb ist bei der Ausführung des 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. Dabei ist darauf zu achten, dass das richtige Modell angegeben ist. Das AAA-Tool verarbeitet stets dieses Enterprise-Architect-Modell, unabhängig davon welches Modell ggf. gerade in der Enterprise-Architect-Anwendung geöffnet sein sollte.



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.

## 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 Elemente eines AAA-Modells, jedoch sind nur Elemente des Grunddatenbestandes nicht auskommentiert. Im Gegenzug beinhaltet ein Maximalprofil alle Elemente 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 Profilerzeugung gestartet, mit „Exit“ wird die Ausführung abgebrochen.

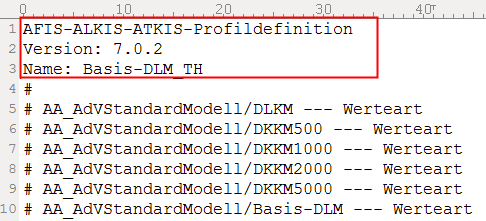
Sofern ein Profil aus einer Datei eingelesen wird, so muss diese den Namen

**<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 (Zeile 3) passen. Beispiele für Dateinamen:

**DLKM\_TestMin.3ap**

**Basis-DLM\_TH.3ap**



Die Datei muss außerdem im angebenen 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:

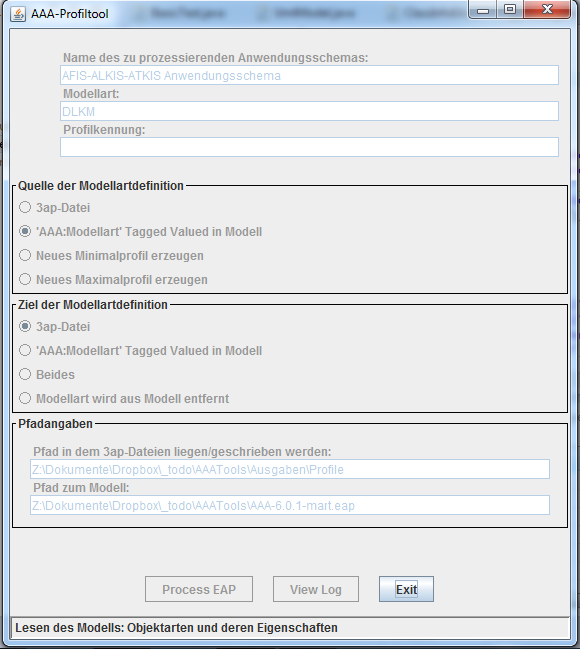
* Es werden typischerweise eine Reihe von Fehlern angezeigt, dass bestimmte Elemente nicht dem Profil zugeordnet werden können, das sie nicht Teil des Anwendungsschemas sind. Diese Meldungen sind korrekt, diese Aufnahme dieser Einträge in Profildateien im Rational-Rose-Tool war überflüssig. Es gibt zwei typische Situationen:
  + 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.“
  + 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. Beispiel:
    - „Relationsart bestehtAus zu Objektart/Datentyp AA\_ZUSO kann dem Profil nicht zugeordnet werden, die Eigenschaft wurde nicht im Anwendungsschema gefunden.“
* Es gibt einige Wertearten, die Umlaute enthalten. Dies kann – je nachdem mit welchen Editoren die Profile bearbeitet wurden – zu Fehlermeldungen führen.

Das Profiltool kann darüber hinaus auch zur Pflege weiterer, länderspezifischer Modellarten genutzt werden. Hierzu ist der Parameter "Profil" leer zu lassen und vom Profiltool wird nicht das Tagged Value "AAA:Profil" exportiert/gesetzt, sondern "AAA:Modellart". Entsprechend ändern sich dabei auch die Texte im Dialog.

Der Dateiname der 3ap-Datei folgt dabei dem Muster

**<Modellart>.3ap**

wobei <Modellart> der Kurzname der Modellart (Parameter „Modellart“ in der Konfigurationsdatei) ist.



## NAS-Tool

Für die Version 6.0.1 gilt: 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. Außerdem müssen alle Pakete von der Versionskontrolle abgekoppelt sein. Dies bedeutet, dass die Ausführung auf einer temporären Kopie des AAA-Modell durchgeführt werden sollte. Diese EAP-Datei sollte nach der Ausführung gelöscht werden.

Mit der Version 7.1.0 ist dies weniger strikt und lediglich das "AdV"-Paket muss von der Versionskontrolle entkoppelt sein. Das Modell wird jetzt nicht mehr irreparabel geändert, sondern es wird zuerst ein Klon des AAA-Anwendungsschemas mit dem Namen "NAS" angelegt und dann dieses in das Implementierungsschema transformiert.

Die Ausführung erfolgt ohne Dialog.

## 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.