



Alfabet-Daten- integrationsframework

Alfabet-Referenzhandbuch

Dokumentationsversion Alfabet 10.11.0

Urheberrechtlich geschützt © 2013 - 22 Software AG, Darmstadt, Deutschland und/oder Software AG USA Inc., Reston VA, USA und/oder ihre Tochtergesellschaften und/oder ihre Lizenzgeber.

Der Name Software AG und die Namen der Software AG Produkte sind Marken der Software AG und/oder Software AG USA Inc., einer ihrer Tochtergesellschaften oder ihrer Lizenzgeber. Namen anderer Gesellschaften oder Produkte können Marken ihrer jeweiligen Schutzrechtsinhaber sein. Genaue Informationen über die geschützten Marken und Patente der Software AG und ihrer Tochtergesellschaften sind veröffentlicht unter <http://softwareag.com/licenses>.

Die Nutzung dieser Software unterliegt den Lizenzbedingungen der Software AG. Diese Bedingungen sind Bestandteil der Produktdokumentation und befinden sich unter <http://softwareag.com/licenses> und/oder im Wurzelverzeichnis des lizenzierten Produkts.

Diese Software kann Teile von Software-Produkten Dritter enthalten. Urheberrechtshinweise, Lizenzbestimmungen sowie zusätzliche Rechte und Einschränkungen dieser Drittprodukte können dem Abschnitt "License Texts, Copyright Notices and Disclaimers of Third Party Products" entnommen werden. Diese Dokumente enthalten den von den betreffenden Lizenzgebern oder den Lizenzen wörtlich vorgegebenen Wortlaut und werden daher in der jeweiligen Ursprungssprache wiedergegeben. Für einzelne, spezifische Lizenzbeschränkungen von Drittprodukten siehe PART E der Legal Notices abrufbar unter dem Abschnitt „License Terms and Conditions for Use of Software AG Products / Copyrights and Trademark Notices of Software AG Products“. Diese Dokumente sind Teil der Produktdokumentation, die unter <http://softwareag.com/licenses> oder im Verzeichnis der lizenzierten Produkte zu finden ist.

Die Produkte der Software AG stellen Funktionalität zur Verfügung, die für die Verarbeitung persönlicher Daten entsprechend der EU-Datenschutz-Grundverordnung (DSGVO) genutzt werden kann. Die Beschreibungen zur Nutzung dieser Funktionalität finden Sie in der Administrationsdokumentation des jeweiligen Produkts.

Konventionen für die Dokumentation

Konvention	Bedeutung
Fett	<p>Wird für alle Elemente verwendet, die auf der Benutzeroberfläche dargestellt werden, wie zum Beispiel Menüelemente, Schaltflächen, Registerkarten, Dialogfelder, Titel von Ansichtsseiten und Kommandos.</p> <p>Beispiel: Klicken Sie nach Beenden des Setups auf Fertigstellen.</p>
<i>Kursiv</i>	<p>Wird für Hervorhebungen und Verweise auf Dokumententitel und Kapitelüberschriften verwendet. Wird im Code für Variablen verwendet</p> <p>Beispiel: Informationen hierzu finden Sie im Referenzhandbuch <i>Administration</i>.</p> <p>Beispiel: <code><XmlElement XmlAttribute="Anwendername"/></code></p>
Anführungszeichen oben	<p>Kennzeichnet einzugebende Werte und feststehende Namen im Text.</p> <p>Beispiel: Wenn der Objektstatus "Aktiv" ist, dann...</p>
Begriffe komplett in Großbuchstaben	<p>Tastaturtasten</p> <p>Beispiel: STRG+UMSCHALT</p>
Datei > Öffnen	<p>Wird für Menüaktionen verwendet, die vom Anwender durchzuführen sind.</p> <p>Beispiel: Um die Applikation zu schließen, wählen Sie Datei > Beenden</p>
< >	<p>Steht für Variablen, die vom Anwender eingegeben werden.</p> <p>Beispiel: Erzeugen Sie einen neuen Anwender und geben Sie <Anwendername> ein. (Ersetzen Sie den Begriff inklusive Klammern mit dem jeweiligen aktuellen Wert.)</p>
	<p>Dies ist ein Hinweis, der Zusatzinformationen gibt.</p>
	<p>Dies ist ein Hinweis, der Prozessinformationen gibt.</p>
	<p>Dies ist ein Beispiel.</p>
	<p>Dies ist eine Warnung.</p>

Inhaltsverzeichnis

Kapitel 1: Alfabet Datenintegrationsframework	7
Vorteile des Alfabet Datenintegrationsframeworks	8
Voraussetzungen	8
Erforderlicher Personalbedarf	8
Lizenzen	9
Verwandte Dokumente	9
Support	9
Kapitel 2: Das Alfabet -Metamodell in der Alfabet -Datenbank	11
Beschreibung des Standardmetamodells	12
Objektklassen	12
Objektklasseneigenschaften	15
Speicherung von Objektdaten in der Alfabet -Datenbank	17
Struktur der Datenbanktabelle und eindeutige Bezeichner	17
Speicherung von übersetzbaren Objektklasseneigenschaften	18
Datentypen und Formate	18
Speicherung von Beziehungen zwischen Objektklassen	21
Speicherung der Überwachungshistorie	22
Vorkonfigurierte und benutzerdefinierte Einschränkungen für die Dateneingabe	23
Objektklassenkonfiguration	24
Konfiguration von Objektklasseneigenschaften	27
XML-basierte Definitionen	29
Referenz- und Bewertungsdaten sowie Referenzdatenzuordnung	30
Kapitel 3: Konfigurieren von ADIF-Schemata	31
Konfigurieren von ADIF über den ADIF-Explorer	31
Der untergeordnete Baum "ADIF-Schemata" des ADIF-Explorers	31
Der untergeordnete Baum "Metamodell" des ADIF-Explorers	38
Konfigurieren von ADIF über XML-Bearbeitung	42
Interner XML-Editor	42
Externe Editoren	43
Kapitel 4: Konfigurieren von Datenimporten mit ADIF	45
Datenverarbeitung während des Imports	45
Ausführungsschritte des Importvorgangs	46
Vorbereiten von Daten für den Import	47
Microsoft® Excel®-Dateien	48
XML-Dateien	48
JSON-Dateien	49
Kommagetrennte Formate (*.csv)	49
Externe Datenbanken	49
Konzipieren des Datenimports	50
Konfigurieren des ADIF-Importschemas	50
Erstellen eines ADIF-Importschema	52
Konfigurieren des Imports aus verschiedenen externen Quellformaten	57
Definieren des Hochladens von Daten aus einer externen Datenbank in eine Datenbank-Importgruppe	59

Definieren des Hochladens von Daten aus einer LDAP-Tabelle in eine LDAP-Importgruppe	64
Definieren des Datenimports aus XML-Dateien in einer XML-Importgruppe	69
Definieren des Datenimports aus JSON-Dateien	76
Definieren des Datenimports aus Microsoft® Excel-Dateien und kommagetrennten Dateiformaten	86
Definieren des Imports externer Daten in einen Importeintrag	88
Zuordnen von externen Daten zu temporären Tabellen und Alfabet -Datenbanktabellen	90
Definieren von Beziehungen	100
Sammeln von Daten aus mehreren externen Dateien oder Datenbanktabellen in einer temporären Tabelle	111
Halbautomatische Erzeugung von Importeintragsdefinitionen für Importdateien	113
Konfigurieren von SQL-Befehlen für optionale erweiterte Importfunktionen	115
Erzeugen eines SQL-basierten Berichts	118
Anwendungsfall: Ausfüllen leerer Eigenschaftswerte mit Standardwerten beim Importieren in Pflicht-Objektklasseneigenschaften	120
Anwendungsfall: Definieren importbezogener benutzerdefinierter Eigenschaften	121
Konfigurieren der Ausführung des Importschemas in Abhängigkeit von aktuellen Parametern	122
Konfigurieren der bedingten Ausführung von Teilen des Importschemas	122
Konfigurieren des Imports in Abhängigkeit von Parametern	123
Konfigurieren von Protokollierungsparametern	130
Konfigurieren des Inhalts von Protokollmeldungen	134
Konfigurieren der Importhistorie	136
Bereinigen der Überwachungstabellen	136
Informationen über den Importanwender	137
Konfigurieren des automatischen Starts von Workflows während des Imports	137
Kapitel 5: Konfigurieren von Datenexporten mit ADIF	140
Der Exportprozess	140
Standardkonfiguration des Datenexports	141
Erzeugen eines ADIF-Exportschemas	142
Konfigurieren des Exports in externe Datenbanktabellen	144
Konfigurieren des Exports in XML-Dateien	150
Konfigurieren des Exports in kommagetrennte Datendateien	162
Konfigurieren des Exports in Microsoft® Excel®-Dateien	166
Konfigurieren von SQL-Befehlen für optionale Exporterweiterungen	172
Erzeugen eines SQL-basierten Berichts	174
Use-Case: Aktualisieren von bestehenden Datensätzen in einer Zieldatenbank	175
Anwendungsfall: Definieren exportbezogener benutzerdefinierter Eigenschaften	177
Anwendungsfall: Datenrestrukturierung vor dem Export mit temporären Tabellen	178
Konfigurieren der Ausführung in Abhängigkeit von aktuellen Parametern	179
Konfigurieren der bedingten Ausführung von Teilen des Exportschemas	179
Konfigurieren des Exports in Abhängigkeit von Parametern	181
Konfigurieren von Protokollierungsparametern	188
Konfigurieren der Speicherung und Handhabung von Protokolldateien	189
Konfigurieren des Inhalts von Protokollmeldungen	191
Kapitel 6: Debugging einer ADIF-Konfiguration	195
Aufbau der Debugger-Benutzeroberfläche	195
Ablauf des Debugging-Prozesses	196
Erforderliche Konfiguration des ADIF-Schemas vor dem Debugging	197
Durchführen eines Debug-Durchlaufs	199

Erforderliche Konfiguration des ADIF-Schemas nach dem Debuggen	203
Kapitel 7: Starten von Datenimport, Datenexport oder Datenänderung mit ADIF	204
Ausführen von ADIF über ein Befehlszeilen-Tool	205
Bereitstellen relevanter Daten und Konfiguration	206
Starten der ADIF-Konsolenanwendung	208
Ausführen von ADIF über RESTful-Serviceaufrufe	214
Ausführen von ADIF über ein Ereignis	214
Ausführen von ADIF über eine Schaltfläche in der Alfabet-Benutzeroberfläche	215
Konfigurieren des ADIF-Schemas zur Ausführung über die Benutzeroberfläche	215
Hinzufügen einer Schaltfläche zur ADIF-Ausführung zu einem konfigurierten Bericht	215
Hinzufügen einer Schaltfläche zur ADIF-Ausführung zu einer Objektansicht	216
Ausführen und Überprüfen von ADIF über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" in der Alfabet -Benutzeroberfläche	217
Konfigurieren der Verfügbarkeit von ADIF-Ausführung und -Überprüfung in der Benutzeroberfläche	218
Konfigurieren der asynchronen Ausführung	219
Konfigurieren von Zugriffsberechtigungen für Ordner in der internen Dokumentenauswahl für den asynchronen Export in eine Datei	219
Konfigurieren eines Anwenders zur Ausführung selbstreflektierender Ereignisse	220
Überprüfen des Erfolgs von ausgeführten ADIF-Prozessen	222
Ausführen von ADIF	225
Testen der ADIF-Schema-Ausführung	226
Planen von ADIF-Prozessen für die Ausführung zu einem definierten Zeitpunkt mit der Funktionalität "Prozesszeitplan"	226
Voraussetzungen zum Verwenden des Job-Planers	227
Erzeugen eines Prozesszeitplans für das Löschen alter ADIF-Sitzungsinformationen per Batch-Prozess	231
Vordefinierte ADIF-Schemata	235
Importschema für die automatische Neuordnung von Kostenstellenkosten	241

Kapitel 1: Alfabet Datenintegrationsframework

Die erste Herausforderung, der ein Unternehmen bei der Implementierung einer zentralen IT-Management-Software wie Alfabet gegenübersteht, ist die Integration vorhandener Daten zur IT-Architektur des Unternehmens und der Verwaltung in der Datenbank der IT-Management-Software. Im Normalfall wurden Daten mit unterschiedlichen Methoden in einer Vielzahl von Formaten gepflegt, von einfachen Textdateien oder Microsoft® Excel®-Arbeitsblättern bis hin zu Datenbanken, die Daten in unterschiedlichen Formaten speichern. Es kann ein mühsamer und zeitaufwendiger Prozess sein, diese Daten aus den unterschiedlichen Quellen zusammenzutragen und sie in ein Format zu konvertieren, das der Inventory-Struktur für das zentrale IT-Management entspricht.

Die zweite Herausforderung, mit der ein Unternehmen konfrontiert werden kann, ist die Anbindung an andere Tools über Schnittstellen während der Laufzeit des Systems. Daten, die im zentralen Inventory gespeichert sind, können beispielsweise buchhaltungsrelevante Daten enthalten, die gleichzeitig in der von der Buchhaltungssoftware verwendeten Datenbank gespeichert werden müssen. Es muss also entschieden werden, wo die Daten gespeichert werden sollen und wie Änderungen in die andere Datenbank übernommen werden.

Normalerweise erfordern Lösungen für diese Probleme die Implementierung einer Vielzahl von Technologien, die individuell für die Schnittstellenanbindung an jede externe Quelle angepasst wurden. Derartige Lösungen erfordern in der Regeln einen erheblichen Entwicklungsaufwand, der zusätzlich zur Standard-Software und einem zeitaufwendigen und fehleranfälligen Konvertierungsprozess in Auftrag gegeben werden muss, um die Datenformate aus verschiedenen Quellen für das entsprechende Importschema zu standardisieren.

Software AG bietet jetzt eine Lösung, die den Bedarf nach hoher Flexibilität erfüllt und die Kunden in die Lage versetzt, ihre eigenen Lösungen für die Datenintegration zu entwickeln, ohne Daten in ein festes Importformat konvertieren zu müssen.

Das Alfabet-Datenintegrationsframework (ADIF genannt) ist eine Technologie für den Batchimport und -export sowie die Batchmanipulation sehr großer Datenmengen in der Alfabet-Datenbank. Die ADIF-Schnittstelle ist in hohem Maße anpassbar, und zwar über native SQL-Befehle in Kombination mit Konvertierungsverfahren, die von Software AG bereitgestellt werden. Mit ADIF können Daten aus den folgenden Quellen importiert bzw. in die folgenden Quellen exportiert werden:

- externe Datenbanktabellen
- CSV-Dateien
- XLS-Dateien
- XLSX-Dateien
- XML-Dateien

Über die ADIF-Schnittstelle können Daten aus einem beliebigen kundendefinierten Format innerhalb der Datenquelle importiert werden. Die Daten lassen sich in nur einem Schritt aus mehreren Quellen integrieren, so dass Sie die Schnittstelle beispielsweise so konfigurieren können, dass die Daten teils aus einer Datenbankabfrage und teils aus XML-Dateien importiert werden.

Im Referenzhandbuch wird der Prozess der Datenintegration in die Alfabet-Datenbank beschrieben und das notwendige Wissen vermittelt, um die Schnittstelle erfolgreich so zu konfigurieren, dass die im kundenspezifischen Format bereitgestellten Daten in das Format konvertiert werden, das für die Speicherung der Daten in den Alfabet-Datenbank -Tabellen erforderlich ist. Außerdem wird erläutert, wie man Alfabet -

Daten für den Export in ein Format zur weiteren manuellen oder durch Drittanbieter-Tools ausgelösten Verarbeitung außerhalb der Alfabet-Komponenten konvertiert.



Bitte beachten Sie, dass ADIF nur zur Änderung von Daten verwendet werden darf, die in bestehenden Datenbanktabellenspalten der Alfabet-Datenbank gespeichert sind. Obwohl SQL-Befehle zum Anlegen oder Löschen von Datenbanktabellen und Datenbanktabellenspalten in ADIF erlaubt sind, um temporäre Datenbanktabellen zu erzeugen, die nur während des Importvorgangs bestehen bleiben, dürfen Sie die bestehende Datenbankstruktur NICHT ändern. Software AG kann nicht für Schäden an der Alfabet-Datenbank haftbar gemacht werden, die durch vom Kunden definierte ADIF-Import- oder -Exportaktionen, die dieser Regel nicht entsprechen, verursacht wurden.

Vorteile des Alfabet Datenintegrationsframeworks

Die ADIF-Fähigkeiten bieten die folgenden Vorteile:

- Unabhängigkeit vom Datenformat der externen Daten
- Verarbeitung von unterschiedlichen Datenformaten in einem Schritt
- Grafische Benutzeroberfläche, die die Konfiguration der Business-Logik für Importe, Exporte oder Datenmanipulation unterstützt:
 - automatische Strukturerkennung der zu importierenden Daten
 - vordefinierte Elemente, die den Entwickler durch die Konfiguration der erforderlichen Import-, Export- oder Datenmanipulationsaktionen führen
- Funktionen zur Vereinfachung der Entwicklungsaufgaben:
 - Editoren mit Unterstützung für die korrekte Konfiguration
 - visuelle Debugger, die zulassen, dass das Ergebnis jedes Prozessschrittes während der Schnittstellenkonfiguration kontrolliert wird, einschließlich Prüfung der Zwischenergebnisse
 - anpassbare Protokollierung der Import-, Export- und Datenmanipulationsschritte
 - Möglichkeit zur Durchführung von Testdurchläufen, sodass die Ergebnisse nur temporär auf die Datenbank angewendet werden.

Voraussetzungen

In diesen Abschnitten werden die Voraussetzungen beschrieben, die für die Verwendung von ADIF erfüllt sein müssen, und es werden weitere verfügbare Informationsquellen aufgeführt.

Erforderlicher Personalbedarf

Die ADIF-Schnittstelle ist in hohem Maß anpassbar und kann vom Kunden ohne Vorkonfiguration oder zusätzliche Entwicklung durch die Software AG konfiguriert werden. Folgende Kenntnisse sind für die Konfiguration der Import-Schnittstelle erforderlich.

- Sie müssen wissen, wie man Befehle in native-SQL schreibt, um das Alfabet Data Integration Framework zu konfigurieren. Befehle müssen in der SQL-Version geschrieben werden, die vom Datenbankserver, der die Alfabet-Datenbank hostet, interpretiert werden können.
- Kenntnisse des Alfabet -Metamodells sind erforderlich, um das Mapping der Daten in die Datenbanktabellen in Alfabet zu implementieren. Weitere Informationen finden Sie unter [Das Alfabet -Metamodell in der Alfabet -Datenbank](#).

Lizenzen

Das Alfabet Data Integration Framework erfordert eine spezielle Lizenzvereinbarung mit Software AG. ADIF ist nicht in den Standard-Grundfunktionen von Alfabet enthalten.

Verwandte Dokumente

Es wird empfohlen, bei der Konfiguration der ADIF-Schnittstelle die folgenden Dokumente heranzuziehen

- *Dokumentation des Alfabet -Metamodells (nur in englischer Sprache verfügbar)*

Eine detaillierte Beschreibung aller relevanten Objektklassen des Alfabet -Metamodells und ihrer Eigenschaften, ihrer Bedeutung innerhalb des Klassenmodells, ihrer Beziehung zu anderen Objektklassen und Informationen dazu, wie Eigenschaften in der Datenbanktabelle der Objektklasse dargestellt werden. Dieses Dokument trägt zum Verständnis der Zielstruktur für den Datenimport und der Quellstruktur für den Datenexport oder die Datenmanipulation bei.
- *Referenzhandbuch Konfigurieren von Alfabet mit Alfabet Expand*

Eine Anpassung der Alfabet -Standardlösung kann beim Import von Daten in die Alfabet-Datenbank erforderlich sein, um die für den Datenimport erforderliche Grundstruktur bereitzustellen. So können beispielsweise benutzerdefinierte Eigenschaften den Objektklassen hinzugefügt werden, um kundenspezifische Daten zu speichern, die nicht Teil des Standard-Metamodells sind, oder es können Zugriffsrechte für Objektklassen definiert werden.
- *Referenzhandbuch Konfigurieren von Bewertungen und Referenzdaten in Alfabet*

Dieses Dokument bietet Informationen zu den Konfigurationsschritten, die erforderlich sind, bevor Daten im Zusammenhang mit der Kostenplanung importiert, Rollen implementiert und Objekte bewertet werden.

Support

Sollten Sie bei der Nutzung von ADIF auf Probleme stoßen, wenden Sie sich bitten an Software AG Support:

Bitte öffnen Sie ein Ticket im Empower eService, wenn Sie eine Serviceanfrage haben, aber auch für Anfragen nach über den Standard hinausgehende Unterstützung, wie Trainingsanfragen, Skriptentwicklungen und Datenintegration.

<https://empower.softwareag.com>

Wenn Sie ein Ticket für eine Service-Anforderung übermitteln, sollten Sie die Release-Nummer und die Patch-Version Ihres Alfabet -Produkts angeben. Sie können auf diese Informationen zugreifen, indem Sie auf **Hilfe > Über Alfabet** klicken. Die Tickets werden aufgenommen und an das jeweils zuständige Team weitergeleitet.

Empower eService bietet außerdem:

- Ticket-Status nachverfolgen
- Lokale Telefonnummern des Supports.

Zusätzlich zu den lokalen Telefonnummern für den Support können Sie die folgende gebührenfreie Telefonnummer verwenden:

+800 2747 4357

Kapitel 2: Das Alfabet -Metamodell in der Alfabet -Datenbank

Eine der Hauptaufgaben beim Konfigurieren eines ADIF-Schemas ist das Zuordnen externer Daten zu den Daten in der Alfabet-Datenbank. Die Datenspeicherung in der Alfabet-Datenbank wird durch das Alfabet -Metamodell bestimmt, welches die Grundlage der Alfabet -Lösung bildet. Im Metamodell sind die in der Alfabet -Lösung verwendeten Objektklassen vorgegeben, die Eigenschaften der Objektklassen sowie die Beziehungen zwischen Objektklassen.

Dieser Abschnitt enthält grundlegende Informationen zur Struktur des Metamodells und dessen Speicherung in Datenbanktabellen. Es wird dringend empfohlen, diesen Abschnitt zu lesen, damit Sie die detaillierte Beschreibung verstehen, die in der technischen Dokumentation des Metamodells für die einzelnen Metamodellklassen bereitgestellt wird.

Folgende Informationen sind verfügbar:

- [Beschreibung des Standardmetamodells](#)
 - [Objektklassen](#)
 - [Integritätsreferenzen](#)
 - [Objektklasseneigenschaften](#)
- [Speicherung von Objektdaten in der Alfabet -Datenbank](#)
 - [Struktur der Datenbanktabelle und eindeutige Bezeichner](#)
 - [Speicherung von übersetzbaren Objektklasseneigenschaften](#)
 - [Datentypen und Formate](#)
 - [Speicherung von Beziehungen zwischen Objektklassen](#)
 - [Speicherung der Überwachungshistorie](#)
- [Vorkonfigurierte und benutzerdefinierte Einschränkungen für die Dateneingabe](#)
 - [Objektklassenkonfiguration](#)
 - [Objektklassenstereotypen](#)
 - [Klassenschlüssel](#)
 - [Mandantendatenverarbeitung](#)
 - [Konfiguration von Objektklasseneigenschaften](#)
 - [Standardwerte für Objektklasseneigenschaften](#)
 - [Größenbeschränkungen für Objektklasseneigenschaften](#)
 - [Validierer für Objektklasseneigenschaften](#)
 - [Objektklasseneigenschaften zugeordnete geschützte und benutzerdefinierte Aufzählungen](#)
- [XML-basierte Definitionen](#)
- [Referenz- und Bewertungsdaten sowie Referenzdatenzuordnung](#)

Beschreibung des Standardmetamodells

Das von Software AG bereitgestellte Standardmetamodell enthält ein Klassenmodell, das aus Objektklassen und deren Objektklasseneigenschaften besteht. Die folgenden Informationen sind relevant für Standardobjektklassen und Objektklasseneigenschaften. Im weiteren Verlauf dieses Kapitels im Abschnitt [Vorkonfigurierte und benutzerdefinierte Einschränkungen für die Dateneingabe](#) werden Informationen beschrieben, die für den ADIF-Prozess für konfigurierte Objektklassen und Objektklasseneigenschaften wichtig sind.

Folgende Informationen sind über das Klassenmodell verfügbar:

- [Objektklassen](#)
- [Integritätsreferenzen](#)
- [Objektklasseneigenschaften](#)

Objektklassen

Das Alfabet -Klassenmodell basiert auf Objektklassen, von denen einige im Konfigurationswerkzeug Alfabet Expand sichtbar sind. Jede Objektklasse verfügt über einen Satz vorkonfigurierter Objektklasseneigenschaften und -attribute.

Das Klassenmodell kann hinsichtlich der Relevanz für den Datenimport in drei verschiedene Kategorien unterteilt werden:

- Das erweiterbare Metamodell, mit dem Instanzobjektklassen repräsentiert werden.
Diese Klassen werden zum persistenten Speichern von Daten über die IT-Landschaft des Kunden in der Alfabet-Datenbank verwendet. Sie werden im Explorer des Konfigurationswerkzeugs Alfabet Expand angezeigt. Die meisten dieser Klassen sind anpassbar, und Kunden können den Klassen benutzerdefinierte Eigenschaften hinzufügen, um unternehmensspezifische Daten zu speichern, die nicht in den Standardeigenschaften der Objektklasse wiedergespiegelt werden. Das erweiterbare Metamodell ist das Hauptziel für Datenimporte über das ADIF-Schema.
- Objektklassen, mit denen Objekte zum Unterstützen der Funktionalität repräsentiert werden.
Diese Klassen werden zum persistenten Speichern von Daten über Objektklassen verwendet, die die Funktionalität unterstützen. Dazu gehört zum Beispiel die Klasse "TimeStatus", in der der Lebenszyklus von Objekten gespeichert wird, oder die Klasse "RoleTypeConfig", in der Informationen darüber gespeichert werden, welche Rollen ein Anwender in Bezug auf ein Objekt einnehmen kann. Diese Objektklassen sind ebenfalls für den Datenimport mithilfe des ADIF-Schemas relevant. Im Explorer in Alfabet Expand sind nur die wichtigsten Objektklassen zum Speichern funktionalitätsbezogener Daten sichtbar, doch die Dokumentation des Metamodells enthält detaillierte Informationen zu all diesen Klassen.
- Hilfskonstrukte, die für die Funktionalität von Alfabet erforderlich sind.
Diese Objektklassen sind nicht relevant für den Datenimport und werden daher nicht in der Dokumentation des Metamodells beschrieben. Einige der unterstützenden Objektklassen werden persistent in Datenbanktabellen gespeichert, andere nicht.



Hinweis: Objektklassen, für die das Attribut **Automatisch verwaltet** auf `true` gesetzt ist, sind Objektklassen, die nur durch Mechanismen geändert werden dürfen, die von den

Komponenten der Alfabet -Software angestoßen werden. Sie dürfen nicht durch einen ADIF-Import oder Komponenten von Drittanbietern geändert werden.

Im Explorer **Klassenmodell** im Konfigurationswerkzeug Alfabet Expand wird auf der Registerkarte **Meta-modell** ein Satz von Objektklassen angezeigt, die für die Konfiguration der Alfabet -Lösung relevant sind. Die Objektklassen und deren Objektklasseneigenschaften werden in der Explorer-Struktur als Knoten dargestellt.

Unterhalb des Knotens **Klassen** können Sie unter Umständen Folgendes sehen:

- Eine private Klasse ist durch ein rotes Schlosssymbol gekennzeichnet und kann nicht geändert werden. Für private Klassen können keine benutzerdefinierten Eigenschaften konfiguriert werden.
- Eine geschützte Klasse weist ein orangefarbenes Schlosssymbol auf und kann eingeschränkt bearbeitet werden. Eine geschützte Klasse verfügt in der Regel über private Eigenschaften, die nicht bearbeitet werden können; doch können für eine geschützte Klasse benutzerdefinierte Eigenschaften erzeugt und konfiguriert werden.
- Bei öffentlichen Klassen wird kein Schlosssymbol angezeigt. Dies ist eine benutzerdefinierte Klasse, die in der Regel mithilfe des Konfigurationswerkzeugs Alfabet Expand von Ihrem Unternehmen erzeugt wurde. Alle relevanten Attribute können bearbeitet und die benutzerdefinierte Klasse kann gelöscht werden. Für benutzerdefinierte Klassen, die von Ihrem Unternehmen erzeugt und definiert wurden, muss ausdrücklich das Attribut **Technischer Name** definiert werden.

Für jede Objektklasse, die auf der Benutzeroberfläche von Alfabet Expand angezeigt wird (und auch für diejenigen, die nicht sichtbar sind), gibt es eine Datenbanktabelle. Jede Klasseneigenschaft, die unterhalb ihres Objektklassenknotens angezeigt wird, entspricht einer Spalte in der Datenbanktabelle.

Wenn in der Explorer-Struktur ein Objektklassenknoten oder ein Objektklasseneigenschaftsknoten ausgewählt ist, werden die Attribute der Klasse/Eigenschaft in der Benutzeroberfläche von Alfabet Expand im rechten Bereich angezeigt. Mit diesen Attributen werden die technischen Daten der Objektklasse/Objektklasseneigenschaft festgelegt, mit denen die Datenbanktabelle aufgebaut wird und Objektdaten innerhalb der Alfabet -Lösung verarbeitet werden.

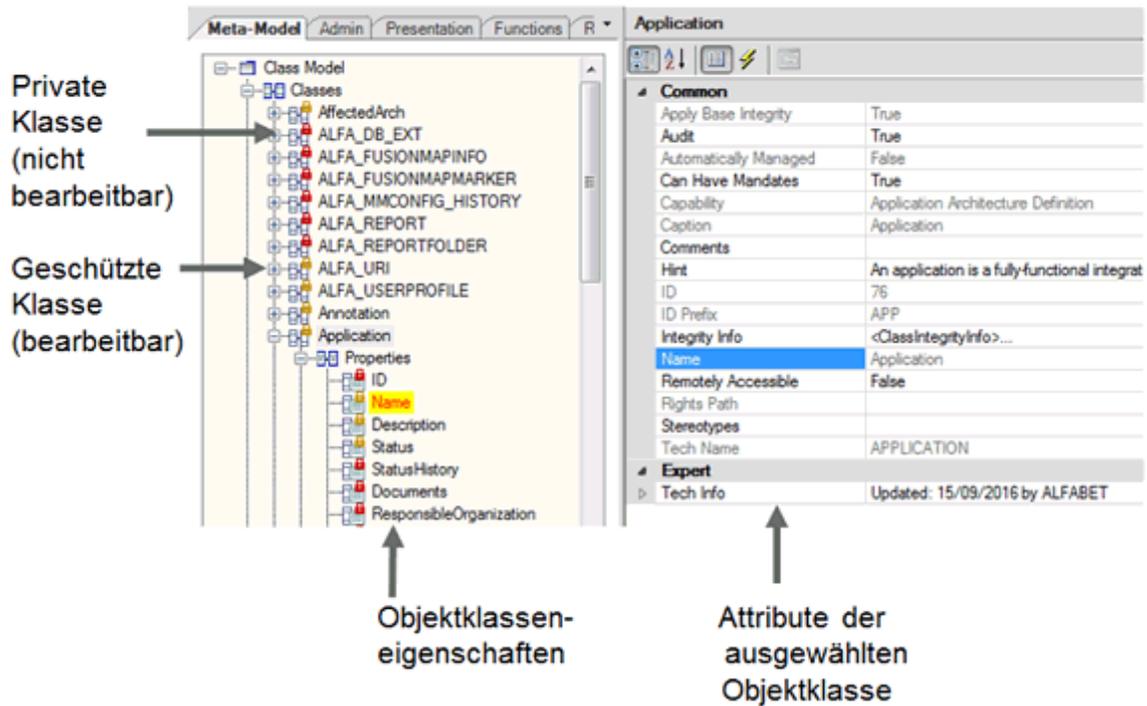


Abbildung: Attribute und Eigenschaften der Objektklasse "Application", angezeigt im Explorer des Werkzeugs Alfabet Expand

Die Objektklassen innerhalb der Alfabet -Lösung können anhand der folgenden drei Attribute erkannt werden:

- **Name:** Das Attribut **Name** ist für jede Objektklasse im Metamodell sowie für jede Eigenschaft in der Objektklasse eindeutig. Mit dem Attribut **Name** werden die Objektklassen und Objektklasseneigenschaften in der Benutzeroberfläche von Alfabet Expand angezeigt sowie Objektklassen und Objektklasseneigenschaften in der Alfabet-Abfragesprache festgelegt. (Die Alfabet-Abfragesprache wird zum Suchen nach und Abrufen von Daten aus der Datenbank verwendet.)
- **Technischer Name:** Das Attribut **Technischer Name** ist für jede Objektklasse eindeutig. Damit wird der Name der Datenbanktabelle für die Objektklasse angegeben. Mit dem Attribut **Technischer Name** einer Objektklasseneigenschaft wird der Name der Datenbanktabellenspalte für die Eigenschaft angegeben. Der **Technische Name** ist daher relevant für das Identifizieren von Objektklassen und deren Eigenschaften in Native-SQL-Abfragen, wenn mithilfe des ADIF-Schemas Daten importiert werden.
- **Titel:** Mit dem Attribut **Titel** wird der Name der Objektklasse bzw. der Objektklasseneigenschaft angegeben, der in der Alfabet -Benutzeroberfläche angezeigt wird.

Die Dokumentation des Standardmetamodells enthält die für jede Objektklasse relevanten Informationen. Bei einigen Objektklassen können die Daten, die in die Datenbanktabelle eingegeben werden dürfen, aufgrund von Anpassungen an der Objektklasse eingeschränkt sein. Solche Anpassungen umfassen beispielsweise die Konfiguration von Objektklassenstereotypen; Klassenschlüssel, die eindeutige Dateneingaben erfordern sowie die Implementierung des Mandantenkonzepts.

Beim Importieren von Daten mithilfe des ADIF-Schemas müssen solche Einschränkungen berücksichtigt werden, wenn Daten zugeordnet und in die Datenbank importiert werden. Ansonsten ist der Datenimport möglicherweise ungültig. Eine Übersicht potenzieller Konfigurationen, die sich auf die Zuordnung von

Objektklassen im ADIF-Schema auswirken können, finden Sie im Abschnitt [Vorkonfigurierte und benutzerdefinierte Einschränkungen für die Dateneingabe](#).

Integritätsreferenzen

Im Alfabet -Metamodell können Objekte in einer Objektklasse eine Abhängigkeitsbeziehung zu anderen Objekten in anderen Klassen haben. Dies wird im Attribut **Integritäts-Info** beschrieben.

Wenn ein Objekt gelöscht wird, werden alle abhängigen Objekte von abhängigen Objektklassen ebenfalls gelöscht. Wird z. B. eine Anwendung gelöscht, werden auch alle Informationsflüsse, für die diese Anwendung in den Eigenschaften "Eigentümer", "An" oder "Von" eines Objekts angegeben ist, ebenfalls gelöscht.

Die Dokumentation des Standardmetamodells enthält die für jede Objektklasse relevanten Abhängigkeitsinformationen.



Beachten Sie, dass für das Attribut **Basisintegrität anwenden** für private und geschützte Objektklassen ein Standardwert festgelegt ist. Der Wert `True` ist festgelegt, wenn für die private oder geschützte Objektklasse ein Basisintegritätsverwaltung durchgeführt wird. In diesem Fall werden auch alle Referenzen auf das gelöschte Objekt gelöscht, wenn ein Objekt der Objektklasse gelöscht wird. Wenn der Wert `False` festgelegt ist, wird für die Objektklasse keine Integritätsverwaltung durchgeführt.

Zum Verbessern der Leistung kann der Wert `False` für einige benutzerdefinierte Objektklassen festgelegt werden, wenn keine Objektklassen auf die benutzerdefinierte Objektklasse verweisen. Es wird jedoch dringend empfohlen, dass Sie sich an den Software AG Support wenden, wenn Sie den Wert für eine benutzerdefinierte Objektklasse auf `False` setzen möchten. Das Einstellen der Eigenschaft **Basisintegrität anwenden** auf `False` kann zu schwerwiegenden Inkonsistenzproblemen in der Datenbank führen.

Objektklasseneigenschaften

Die meisten Objektklassen im Metamodell verfügen über Objektklasseneigenschaften. Unterhalb des Knotens **Eigenschaften** kann Folgendes angezeigt werden:

- Eine private Eigenschaft ist durch ein rotes Schlosssymbol gekennzeichnet und kann nicht geändert werden.
- Für eine geschützte Eigenschaft wird ein orangefarbenes Schlosssymbol angezeigt. Dies ist eine Standardeigenschaft, bei der einige Eigenschaftswerte geändert werden können, zum Beispiel das Attribut **Titel** und das Attribut **Hinweis**. Eine Standardeigenschaft kann nicht gelöscht werden.
- Bei einer öffentlichen Eigenschaft wird kein Schlosssymbol angezeigt. Dies ist eine benutzerdefinierte Eigenschaft, die in der Regel mithilfe des Konfigurationswerkzeugs Alfabet Expand von Ihrem Unternehmen erzeugt wurde. Alle relevanten Attribute können bearbeitet und die benutzerdefinierte Eigenschaft kann gelöscht werden. Für benutzerdefinierte Eigenschaften, die von Ihrem Unternehmen erzeugt und definiert wurden, muss ausdrücklich das Attribut **Technischer Name** definiert werden.

Jede Objektklasseneigenschaft verfügt über ein Attribut **Eigenschaftstyp**, mit dem die Datentypen für die Eigenschaft angegeben werden (z. B. Zeichenfolge, Datum, Ganzzahl, Referenz usw.). Anhand der Definition des Eigenschaftstyps wird bestimmt, welche weiteren Attribute für die Objektklasseneigenschaft definiert werden können. Für eine Eigenschaft des Typs "Zeichenfolge" zum Beispiel können Sie optional die

Attribute **Standard**, **Enum**, **Größe** oder **Validator** definieren. Eine Übersicht über die Datentypen, die für das Attribut **Eigenschaftstyp** möglich sind, finden Sie im Abschnitt [Datentypen und Formate](#).

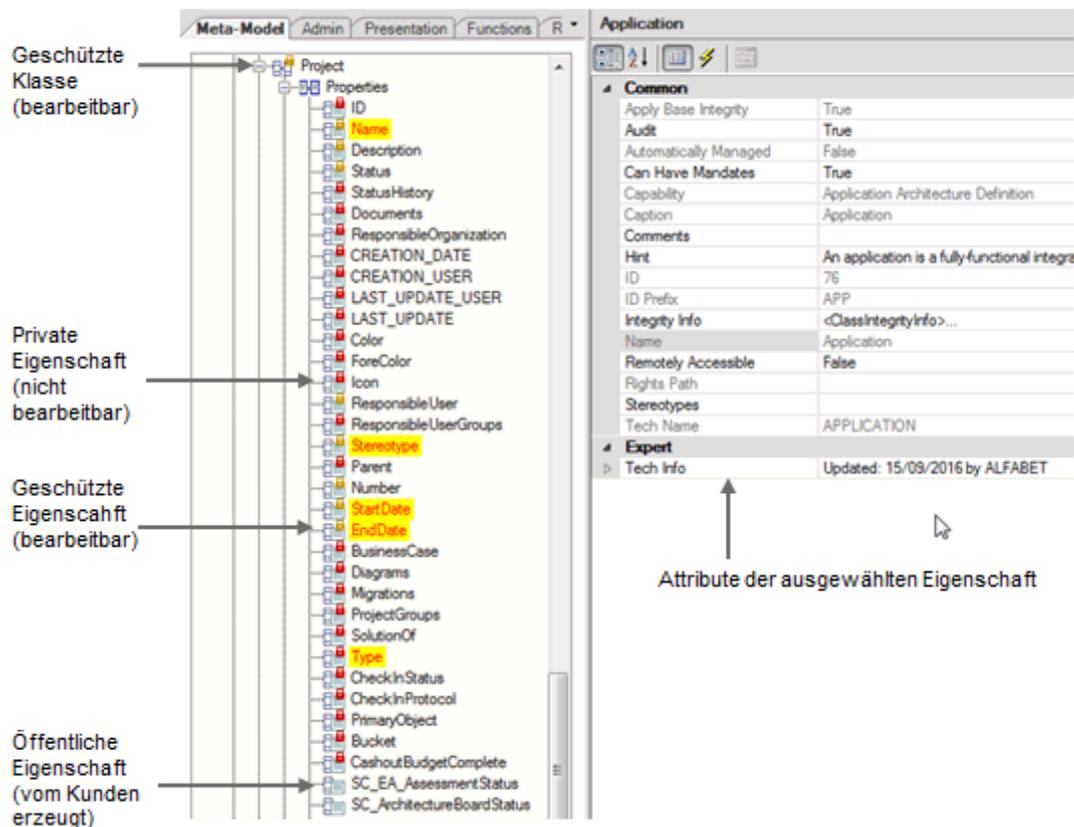


Abbildung: Attribute der Objektklasseneigenschaft "StartDate" der Objektklasse "Project", angezeigt im Explorer des Werkzeugs Alfabet Expand

Die Dokumentation des Standardmetamodells enthält die für jede Objektklasseneigenschaft relevanten Informationen. Bei einigen Eigenschaftstypen können die Daten, die in die Datenbanktabelle geschrieben werden dürfen, aufgrund zusätzlicher Konfigurationen des Metamodells oder der Lösung eingeschränkt sein. Solche Anpassungen umfassen beispielsweise die Konfiguration von Dateneingabevalidierern, Aufzählungen, Referenzdaten oder in XML-Objekten vorgenommene Angaben.

Beim Importieren von Daten mithilfe des ADIF-Schemas müssen solche Einschränkungen berücksichtigt werden, wenn Daten zugeordnet und in die Datenbank importiert werden. Ansonsten ist der Datenimport möglicherweise ungültig. Eine Übersicht potenzieller Konfigurationen, die sich auf die Zuordnung von Objektklassen im ADIF-Schema auswirken können, finden Sie im Abschnitt [Vorkonfigurierte und benutzerdefinierte Einschränkungen für die Dateneingabe](#).



Weitere Informationen über das Definieren benutzerdefinierter Eigenschaften finden Sie im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* im Abschnitt *Konfigurieren benutzerdefinierter Eigenschaften für geschützte oder öffentliche Objektklassen*.

Speicherung von Objektdaten in der Alfabet -Datenbank

In Alfabet-Datenbank ist für jede persistente Objektklasse eine Datenbanktabelle vorhanden. In zusätzlichen Datenbanktabellen werden Beziehungen zwischen Objekten sowie die Überwachungshistorie von Objekten gespeichert.

Folgende Informationen sind verfügbar:

- [Struktur der Datenbanktabelle und eindeutige Bezeichner](#)
- [Speicherung von übersetzbaren Objektklasseneigenschaften](#)
- [Datentypen und Formate](#)
- [Speicherung von Beziehungen zwischen Objektklassen](#)
- [Speicherung der Überwachungshistorie](#)

Struktur der Datenbanktabelle und eindeutige Bezeichner

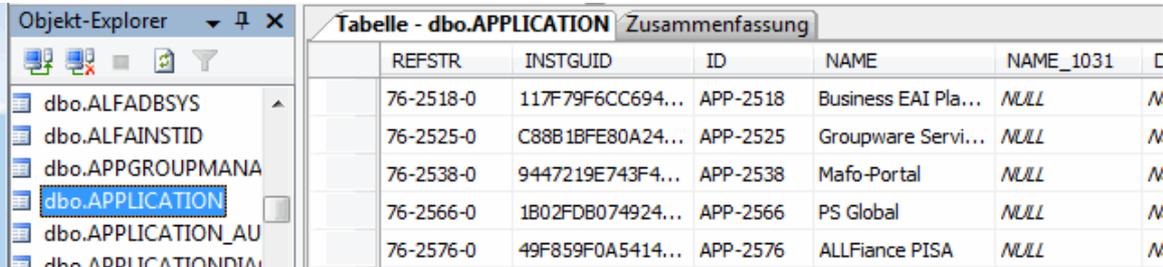
Der Name der Datenbanktabelle wird über das Attribut **Technischer Name** der Objektklasse spezifiziert. Die Namen der Spalten in der Tabelle werden über die Attribute **Technischer Name** der Objektklasseneigenschaften der jeweiligen Objektklasse spezifiziert. Bei den meisten Standardobjektklassen und -eigenschaften in Alfabet ist der **Technische Name** identisch mit dem Attribut **Name** der Objektklasse oder -eigenschaft. Alle Attribute **Technischer Name** werden in Alfabet in Großbuchstaben gespeichert (<TECHNAME>).

Wenn der technische Name, der aus dem Namen der Objektklasse abgeleitet wird, mit Schlüsselwörtern identisch ist, die für das Managementsystem für relationale Datenbanken (RDBMS) von Oracle oder Microsoft SQL Server reserviert sind, wird der **Technische Name** von Objektklassen mit dem Präfix "T_" und von Spalten, die für Attribute stehen, mit dem Präfix "A_" versehen.

Zusätzlich zu den Spalten für Objektklasseneigenschaften enthält eine Datenbanktabelle mehrere Spalten, die automatisch von der Alfabet -Software hinzugefügt werden, wenn der Datenbank ein Objekt hinzugefügt wird. Werte für diese Spalten dürfen NICHT in den Datenimport für neue Objekte einbezogen werden. Solche Spalten können jedoch zum Identifizieren vorhandener Objekte verwendet werden:

- **REFSTR:** REFSTR ist eine eindeutige Referenzzeichenfolge, mit der das Objekt in der Datenbank identifiziert wird. Mit REFSTR werden Beziehungen zwischen Objekten gespeichert. Während des Datenimports mithilfe des ADIF-Schemas müssen für neue Objekte REFSTRs festgelegt werden, bevor eine Beziehung zwischen einem neuen Objekt und einem beliebigen anderen Objekt erzeugt wird. Der Mechanismus zum Festlegen von REFSTRs ist Bestandteil der Funktionsweise des ADIF-Importschemas.
- **INSTGUID:** INSTGUID ist ein eindeutiges Attribut, mit dem das Objekt datenbankübergreifend identifiziert werden kann. INSTGUID ist sogar in Bezug auf verschiedene Datenbanken eindeutig. Wenn Objekte zwei unterschiedlichen Datenbanken hinzugefügt werden, können sie in beiden Datenbanken denselben REFSTR und dieselbe ID haben, aber das INSTGUID ist unterschiedlich. Während des Datenimports mithilfe des ADIF-Schemas werden INSTGUIDs für neue Objekte automatisch festgelegt und müssen nicht in der Importspezifikation enthalten sein.
- **ALFA_INSTID:** ALFA_INSTID ist eine Teilmenge der Eigenschaft REFSTR. Während des Datenimports mithilfe des ADIF-Schemas werden ALFA_INSTIDs für neue Objekte automatisch festgelegt. ALFA_INSTID darf nicht in die Importspezifikation einbezogen werden.

- **ID:** Für einige Objektklassen, die in der Alfabet -Benutzeroberfläche sichtbar sind, wird eine ID festgelegt. Dadurch können Anwender, die in der Alfabet -Benutzeroberfläche arbeiten, Objekte identifizieren oder danach suchen. Bei sichtbaren Objektklassen, für die die Eigenschaft ID vorhanden ist, wird ID automatisch ausgefüllt. Das Attribut ID besteht aus einem Präfix, das auf dem Wert des Attributs **ID-Präfix** der Objektklasse und einer Zahl basiert. Während des Datenimports mithilfe des ADIF-Schemas werden IDs für neue Objekte automatisch festgelegt und sollten NICHT in der Importspezifikation enthalten sein.



	REFSTR	INSTGUID	ID	NAME	NAME_1031	C
	76-2518-0	117F79F6CC694...	APP-2518	Business EAI Pla...	NULL	N
	76-2525-0	C88B1BFE80A24...	APP-2525	Groupware Servi...	NULL	N
	76-2538-0	9447219E743F4...	APP-2538	Mafo-Portal	NULL	N
	76-2566-0	1B02FDB074924...	APP-2566	PS Global	NULL	N
	76-2576-0	49F859F0A5414...	APP-2576	ALLFiance PISA	NULL	N

Abbildung: Datenbanktabelle für die Objektklasse "Anwendung" mit Attributen, die der Tabelle automatisch hinzugefügt werden

Speicherung von übersetzbaren Objektklasseneigenschaften

In Alfabet ist immer eine Sprache als Standardsprache konfiguriert. Falls Ihr Unternehmen jedoch zusätzliche Sprachoptionen für die Alfabet -Benutzeroberfläche benötigt, konfiguriert Ihr Lösungsentwickler im Konfigurationswerkzeug Alfabet Expand relevante Landeseinstellungen für das Unternehmen.

Ist das Attribut **Datenübersetzung unterstützen** für eine Landeseinstellung auf `True` gesetzt, enthalten die Datenbanktabellen für einige Objektklassen Spalten zum Übersetzen. Bei Objektklassen, die in der Alfabet -Benutzeroberfläche sichtbar sind, können einige Objektklassen übersetzt werden, und Anwender können im relevanten Objektklasseneditor Übersetzungen für Eigenschaften angeben (hauptsächlich die Eigenschaften `Name` und `Description`).

Die Übersetzungen werden in der Datenbanktabelle in separaten Spalten mit dem Spaltennamen `<Tech Name Eigenschaft>_<Sprachcode>` gespeichert. Beispielsweise kann die Eigenschaft `Name` der Klasse "Anwendung" ins Deutsche übersetzt werden. Die Zahl 1031 ist der Sprachcode für Deutsch. Daher wird die Spalte **NAME_1031** automatisch der Datenbanktabelle hinzugefügt. Beachten Sie, dass Sprachspalten nur dann in Datenbanktabellen verfügbar sind, wenn das Attribut **Datenübersetzung unterstützen** für eine Landeseinstellung auf `True` gesetzt ist.

Datentypen und Formate

Jede Objektklasseneigenschaft verfügt über ein Attribut **Eigenschaftstyp**, mit dem bestimmt wird, wie Daten definiert werden müssen (z. B. als Zeichenfolge, Datum, Ganzzahl, Referenz usw.). Anhand des **Eigenschaftstyps** wird bestimmt, wie die Daten in den Datenbanktabellen gespeichert werden. Der Datenspeicherungsmodus kann bei Oracle®-Datenbankservern und Microsoft® SQL Servern® unterschiedlich sein.

Einige Eigenschaftstypen können mithilfe weiterer Attribute der Objektklasseneigenschaft weiter konfigurierbar sein. Für eine Eigenschaft vom **Eigenschaftstyp** `String` zum Beispiel können Sie optional eine Aufzählung definieren, mit der Werte angegeben werden, die für die Eigenschaft Zeichenfolge verwendet werden dürfen. Entsprechend lassen viele Eigenschaftstypen einen Standardwert zu. Der für den

Standardwert festgelegte Wert wird automatisch in die Datenbank geschrieben, wenn der Anwender in der Alfabet -Benutzeroberfläche nicht ausdrücklich einen Wert für die Eigenschaft definiert. Weitere Informationen über die möglichen Abhängigkeiten von Objektklasseneigenschaften finden Sie im Abschnitt [Vorkonfigurierte und benutzerdefinierte Einschränkungen für die Dateneingabe](#).

Die folgende Tabelle bietet eine Übersicht über die zulässigen Eigenschaftstypen für Objektklasseneigenschaften, den Datenspeichermodus sowohl für Microsoft® SQL Server® als auch für Oracle®-Datenbankserver sowie zusätzliche Informationen, die für den Eigenschaftstyp relevant sind.

Eigen-schaftstyp	Gespeichert in MS SQL Server®-Datenbanken als:	Gespeichert in Oracle-Datenbanken als:	Zusätzliche Informationen
Boolean	bit	NUMBER	<p>Mithilfe des Attributs Standard kann für Eigenschaften vom Eigenschaftstyp <code>Boolean</code> ein Standardwert definiert werden.</p> <p>Zulässige Werte: 1 0 (1 = true, 0 = false)</p>
Date	datetime	TIMESTAMP	
DateTime	datetime	TIMESTAMP	
Integer	int	NUMBER	<p>Mithilfe des Attributs Enum kann für Eigenschaften vom Eigenschaftstyp <code>Integer</code> eine Aufzählung definiert werden.</p> <p>Mithilfe des Attributs Standard kann für Eigenschaften vom Eigenschaftstyp <code>Integer</code> ein Standardwert definiert werden.</p> <p>Weitere Informationen finden Sie unter Vorkonfigurierte und benutzerdefinierte Einschränkungen für die Dateneingabe.</p>
Real	float	FLOAT	<p>Mithilfe des Attributs Enum kann für Eigenschaften vom Eigenschaftstyp <code>Real</code> eine Aufzählung definiert werden.</p> <p>Mithilfe des Attributs Standard kann für Eigenschaften vom Eigenschaftstyp <code>Real</code> ein Standardwert definiert werden.</p> <p>Für Eigenschaften vom Eigenschaftstyp <code>Real</code> kann mit dem Attribut Genauigkeit die Anzahl der zulässigen Dezimalstellen definiert werden.</p>

Eigenschaftstyp	Gespeichert in MS SQL Server®-Datenbanken als:	Gespeichert in Oracle-Datenbanken als:	Zusätzliche Informationen
			Weitere Informationen finden Sie unter Vor-konfigurierte und benutzerdefinierte Einschränkungen für die Dateneingabe .
RealArray	nvarchar (max)	NCLOB	Mithilfe des Attributs Enum kann für Eigenschaften vom Eigenschaftstyp RealArray eine Aufzählung definiert werden.
Reference	varchar (20)	VAR-CHAR2 (20)	Das Attribut Typ-Info muss für eine Eigenschaft vom Eigenschaftstyp Reference definiert werden.
ReferenceArray	nvarchar (max)	NCLOB	<p>Die Daten werden nur dann als NCLOB in der Datenbanktabelle des Objekts gespeichert, wenn das Attribut Referenzunterstützung = "false" ist. Wenn das Attribut Referenzunterstützung = "true" ist, werden die Daten in der Tabelle RELATIONS gespeichert.</p> <p>Das Attribut Typ-Info muss für eine Eigenschaft des Typs ReferenceArray definiert werden.</p> <p>Weitere Informationen über die Auswirkungen des Attributs Referenzunterstützung auf die Datenspeicherung sowie zusätzliche Details über die Speicherung von Beziehungen finden Sie unter Speicherung von Beziehungen zwischen Objektklassen</p>
String	varchar (\$Size\$)	NVAR-CHAR (\$Size\$)	<p>Für Eigenschaften des Typs String kann mithilfe des Attributs Enum eine Aufzählung definiert werden.</p> <p>Für Eigenschaften des Typs String kann ein erforderliches Format für Eingabewerte definiert werden.</p> <p>Für Eigenschaften des Typs String kann mithilfe des Attributs Standard ein Standardwert definiert werden.</p> <p>Für Eigenschaften des Typs String kann mithilfe des Attributs Größe die Höchstzahl zulässiger Zeichen definiert werden.</p>

Eigen- schaftstyp	Gespeichert in MS SQL Server®-Daten- banken als:	Gespeichert in Oracle-Da- tenbanken als:	Zusätzliche Informationen
			Weitere Informationen finden Sie unter Vor- konfigurierte und benutzerdefinierte Ein- schränkungen für die Dateneingabe .
StringArray	nvarchar (max)	NCLOB	Für Eigenschaften des Typs <code>StringArrays</code> kann mithilfe des Attributs Enum eine Aufzählung definiert werden. Weitere Informationen finden Sie unter Vor- konfigurierte und benutzerdefinierte Ein- schränkungen für die Dateneingabe .
Text	nvarchar (max)	NCLOB	
Time	datetime	TIMESTAMP	
Url	nvarchar (max)	NCLOB	Die Daten werden als Titelttext gespeichert und die URL wird durch einen Zeilenumbruch getrennt (geschrieben als Zeichencode 13 gefolgt von Zeichencode 10). Wenn kein Titel angegeben wird, muss die Definition mit dem Zeilenumbruch beginnen. Beispiel: <pre>Title http://target.com</pre> was identisch ist mit <pre>TitleCHAR(13)CHAR(10)http:/ /target.com</pre>

Speicherung von Beziehungen zwischen Objektklassen

Das Attribut `REFSTR` ist eine eindeutige interne Objekt-ID, mit der jedes Objekt in der Datenbank ausdrücklich identifiziert wird, und die für Referenzen zwischen Objekten in der Datenbank verwendet wird.

In den Datenbankspalten von Objektklasseneigenschaften des Datentyps `Reference` ist das Attribut `REFSTR` des referenzierten Objekts in einer SQL-Datenbank als Datentyp `varchar(20)` und in einer Oracle-Datenbank als Datentyp `VARCHAR2 (20)` gespeichert.

Je nach Einstellung des Attributs **Referenzunterstützung** der Objektklasseneigenschaft gibt es für Objektklasseneigenschaften des Typs `ReferenceArray` zwei unterschiedliche Verarbeitungsmöglichkeiten:

- Ist **Referenzunterstützung** auf `true` gesetzt, werden die von der Eigenschaft angegebenen Beziehungen in der Tabelle `RELATIONS` gespeichert. In der Tabelle der Objektklasse selbst ist keine Datenbankspalte für die Eigenschaft verfügbar. In der Tabelle `RELATIONS` wird die Beziehung in den folgenden Datenbankspalten angegeben:
 - **FROMREF**: Der Wert `REFSTR` des Objekts, das ein anderes Objekt referenziert. Diese wird in einer SQL-Datenbank als Datentyp `varchar(20)` und in einer Oracle-Datenbank als Datentyp `NVARCHAR2(20)` gespeichert.
 - **PROPERTY**: Die Eigenschaft des Objekts, die mit **FROMREF** angegeben wird, die die Beziehung zwischen den Objekten definiert. Diese wird in einer SQL-Datenbank als Datentyp `nvarchar(64)` und in einer Oracle-Datenbank als Datentyp `VARCHAR(64)` gespeichert.
 - **TOREF**: Der Wert `REFSTR` des referenzierten Objekts wird in einer SQL-Datenbank als Datentyp `varchar(20)` und in einer Oracle-Datenbank als Datentyp `NVARCHAR2(20)` gespeichert.

Wenn in der Tabelle `RELATIONS` gespeicherte Referenzen über ein ADIF-Schema importiert werden sollen, wird die Tabelle `RELATIONS` automatisch vom ADIF-Mechanismus ausgefüllt, wenn ein angegebenes Datenformat erfüllt ist.

- Ist **Referenzunterstützung** auf `false` gesetzt, werden die Beziehungen direkt in der Datenbanktabelle des Objekts gespeichert; in einer SQL-Datenbank als Zeichenfolge des Datentyps `nvarchar(max)` und in einer Oracle-Datenbank als Datentyp `NCLOB`. Die `REFSTR` sämtlicher Objekte, auf die die Eigenschaft verweist, werden durch Leerzeichen voneinander getrennt aufgelistet.

Objektklasseneigenschaften der Datentypen `Reference` oder `ReferenceArray` sind nur für Objekte in einer definierten Teilmenge von Objektklassen im Alfabet -Metamodell gedacht. Diese Teilmenge wird durch das Attribut **Typ-Info** der Objektklasseneigenschaft definiert, welches die Referenz oder das Referenz-Array definiert. Mit dem Attribut **Typ-Info** der Objektklasseneigenschaft werden die Namen der zulässigen Zielobjektklassen als kommagetrennte Liste angegeben.

Speicherung der Überwachungshistorie

Standardmäßig wird die Überwachungshistorie pro Objektklasse in der Alfabet -Benutzeroberfläche angezeigt. Der Anwender kann auf die Änderungshistorie eines ausgewählten Objekts zugreifen, indem er auf die Schaltfläche **Historie**  klickt. Sie können native, auf SQL basierende, konfigurierte Berichte erzeugen, um weitere Informationen aus der Überwachungshistorie zu extrahieren, etwa auf Anwender konzentrierte Änderungsverfolgung, Berichterstattung bezüglich Historientrends, Berichterstattung zu neuen und geänderten Objekten (z. B. Anwendungen) sowie Informationsqualitätsmanagement.

Historieninformationen für die einzelnen Objektklassen werden in einer separaten Datenbanktabelle gespeichert. Für jede Objektklasse, bei der das Attribut **Überwachung** auf `"true"` gesetzt ist, wird eine Tabelle namens `<ObjectClassTechName>_AU` erzeugt. Immer wenn ein Objekt geändert wird, wird eine neue Zeile in die Überwachungstabelle geschrieben, in der Informationen über die vorgenommene Änderung gespeichert werden; über den Anwender, der die Änderung vorgenommen hat und die Uhrzeit, zu der die Änderung vorgenommen wurde.



Informationen über die Struktur der Datenbanktabelle für die Historie finden Sie im Abschnitt *Definieren von konfigurierten Berichten bezüglich Historienmanagement* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Während des Imports über ADIF werden die Überwachungshistorientabellen der Objektklassen für Objekte, die mithilfe der ADIF-Prozedur geändert, erzeugt oder gelöscht wurden, automatisch mit den entsprechenden Informationen ausgefüllt. Der Anwender, der den ADIF-Import auslöst, wird als jener Anwender definiert, der die Änderung durchführt.

Einige der für den ADIF-Import erforderlichen Mechanismen führen zu Änderungen in Datenbankspalten, die nicht über eine entsprechende Spalte in der Überwachungstabelle verfügen (z. B. für **INSTGUID**). Wird eine Änderung an dem Objekt vorgenommen, wird der Überwachungstabelle eine neue Zeile hinzugefügt. Mit dieser Zeile wird angezeigt, dass eine Änderung vorgenommen wurde. Die Art der Änderung ist jedoch nicht in der Überwachungstabelle sichtbar, da alle Objektklasseneigenschaften unverändert bleiben. Die ADIF-Benutzeroberfläche lässt zu, dass das Importschema darauf konfiguriert wird, Zeilen ohne sichtbare Änderung automatisch aus der Überwachungstabelle zu löschen.



Die ADIF-Importdefinitionen sollten niemals die Überwachungstabellen beinhalten. Das Ändern der Überwachungstabelle für ein Objekt kann zu schwerwiegenden Datenbankinkonsistenzen führen und möglicherweise einen Fehler verursachen.

Vorkonfigurierte und benutzerdefinierte Einschränkungen für die Dateneingabe

Alfabet ist sehr anpassungsfähig, damit die Software auf die Bedürfnisse Ihres Unternehmens konfiguriert werden kann. Zusätzlich zu den Anpassungen des Klassenmodells, einschließlich beispielsweise dem Konfigurieren von Objektklassenstereotypen oder dem Erzeugen und Definieren benutzerdefinierter Eigenschaften, sind unter Umständen eine Reihe weiterer Konfigurationen erforderlich, um verschiedene Funktionalitäten in Alfabet zu unterstützen.

- Die Konfiguration von Referenz- und Bewertungsdaten bestimmt die zulässigen Werte für einige Objektklasseneigenschaften. Bewertungen und deren Kennzahlen, Kosten- und Nutzenarten, Rollentypen und Diagrammansichten sind Beispiele für einige der Objekte, die auf Konfigurationen basieren, die einem Satz von Objektklassen zugewiesen werden können, der in Ihrem Unternehmen verwendet wird. Wenn Sie einen anderen Wert verwenden als diejenigen, die im Modul **Konfiguration** für die Objektklasse angegeben sind, kann ein Fehler auftreten.
- Die Konfiguration von XML-basierten Objekten in Alfabet Expand bestimmt die zulässige Dateneingabe für einige Objektklassen bzw. Objektklasseneigenschaften. Zum Beispiel verfügen mehrere Objektklassen über die Eigenschaft `Status`, mit der der Release-Status des Objekts im Genehmigungsprozess definiert werden kann. Die möglichen Werte für Release-Statusdefinitionen werden im XML-Objekt **ReleaseStatusDefs** konfiguriert. Wenn Sie einen anderen Wert verwenden als diejenigen, die in solch einer XML-Definition für die Objektklasse angegeben sind, kann ein Fehler auftreten.



Das Anpassen des Metamodells muss vor dem Importieren der Daten mithilfe des ADIF-Schemas stattfinden.

Für den Erfolg des Integrationsprozesses ist es unerlässlich, dass alle angepassten Konfigurationen den Parteien mitgeteilt werden, die am Datenzuordnungsprozess beteiligt sind. Für jede am Metamodell vorgenommene Anpassung wird dringend empfohlen, dass Ihr Unternehmen die Richtigkeit der Zuordnung der betroffenen Objekte im ADIF-Schema überprüft. Ansonsten kann es zu Fehlfunktionen kommen.

Weitere Informationen über potenzielle Konfigurationen, die an der Alfabet -Lösung vorgenommen werden können, finden Sie in den Referenzhandbüchern *Konfigurieren von Alfabet mit Alfabet Expand* und *Konfigurieren von Bewertungen und Referenzdaten in Alfabet*.

In den folgenden Abschnitten werden potenzielle Anpassungsbereiche von Alfabet erörtert. Falls Ihre Lösungskonfiguration solche Anpassungen enthält, unterscheidet sich das Metamodell Ihres Unternehmens von dem hier dokumentierten Standardmetamodell.

Zu den potenziellen Anpassungsbereichen gehören:

- [Objektklassenkonfiguration](#)
 - [Objektklassenstereotypen](#)
 - [Klassenschlüssel](#)
 - [Mandantendatenverarbeitung](#)
- [Konfiguration von Objektklasseneigenschaften](#)
 - [Standardwerte für Objektklasseneigenschaften](#)
 - [Größenbeschränkungen für Objektklasseneigenschaften](#)
 - [Validierer für Objektklasseneigenschaften](#)
 - [Objektklasseneigenschaften zugeordnete geschützte und benutzerdefinierte Aufzählungen](#)
- [XML-basierte Definitionen](#)
- [Referenz- und Bewertungsdaten sowie Referenzdatenzuordnung](#)

Objektklassenkonfiguration

Bei einigen Objektklassen können die Daten, die in die Datenbanktabelle geschrieben werden dürfen, aufgrund von Anpassungen an der Objektklasse eingeschränkt sein. Die Dokumentation des Metamodells enthält Informationen über die Einschränkungen, die für die einzelnen Objektklassen im Standardmetamodell gelten.

Solche Einschränkungen müssen berücksichtigt werden, wenn Daten zugeordnet und in die Datenbank importiert werden. Ansonsten ist der Datenimport möglicherweise ungültig. Es liegt in Ihrer Verantwortung, über die Anpassungen in der Lösungskonfiguration Ihres Unternehmens informiert zu sein, die für den Integrationsprozess relevant sind.

Folgende Konfigurationsprobleme sind für Objektklassen relevant:

- [Objektklassenstereotypen](#)
- [Klassenschlüssel](#)
- [Mandantendatenverarbeitung](#)

Objektklassenstereotypen

Ein Objektklassenstereotyp ist eine Sub-Klassifikation innerhalb einer Objektklasse. Durch das Konfigurieren von Objektklassenstereotypen können Sie vorgeben, dass eine Objektklasse über mehrere Klassentypen verfügt, von denen jeder einen vorgegebenen Satz von Objektklasseneigenschaften, Referenzdaten und Referenzdatenzuordnungen erfasst.

Stereotypen können nur für die Objektklassen "Anwendung", "Komponente", "Gerät", "ICT-Objekt", "Domäne" und "Projekt" konfiguriert werden. Der Lösungsentwickler erstellt die Objektklassenstereotypen für diese Objektklassen mithilfe des Attributs **Stereotyp** der Objektklasse in einer XML-Definition.

Diese Objektklassen verfügen auch über eine Objektklasseneigenschaft **Stereotypen** des Datentyps `String`. Wenn der Datenbank mithilfe des Integrationsprozesses Objekte hinzugefügt werden, muss der Wert für die Objektklasseneigenschaft **Stereotyp** mit dem Namen eines der Stereotypen identisch sein, die im Attribut **Stereotyp** der Objektklasse definiert sind.

Es liegt in Ihrer Verantwortung, über benutzerdefinierte Objektklassenstereotypen in der Lösungskonfiguration Ihres Unternehmens informiert zu sein, die für den Integrationsprozess relevant sind.

Die folgenden Informationen sind relevant für die Implementierung von Objektklassenstereotypen in Ihrem Unternehmen:

- Für die Objektklassen `Project` und `Domain` müssen die konfigurierten Objektklassenstereotypen hierarchisch strukturiert und spezifiziert sein. Diese Stereotypen erfordern die zusätzliche Konfiguration von XML-Objekten. Weitere Informationen über die zusätzlichen Konfigurationsanforderungen für die Objektklassen **Projekt** und **Domäne** finden Sie im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* in den Abschnitten *Konfigurieren der Projektmanagement-Funktionalität* und *Konfigurieren eines Domänenmodells und der Domänenplanung*.
- Ein Stereotyp-Framework kann auch für Steuerungselemente konfiguriert werden, die in die Funktionalität **Strategieableitung** implementiert sind. In diesem Fall wird die Konfiguration der Stereotypen nicht in der Objektklasse vorgenommen, sondern im XML-Objekt **ValueManager**. Wenn der Datenbank mithilfe des Integrationsprozesses Objekte hinzugefügt werden, müssen Sie sichergehen, dass der Wert für die Objektklasseneigenschaft "Stereotyp" mit dem Namen eines der Steuerungselement-Stereotypen identisch ist, die im XML-Objekt **ValueManager** definiert sind.

Weitere Informationen über diese Konfiguration finden Sie im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* im Abschnitt *Konfigurieren der Strategieableitungsfunktionalität*.

Klassenschlüssel

Klassenschlüssel können für Artefakt-Objektklassen konfiguriert werden. Mit einem Klassenschlüssel, dessen Attribut **Eindeutig** auf `True` gesetzt ist, wird eine Objektklasseneigenschaft oder eine Kombination von Objektklasseneigenschaften angegeben, die für diese Klasse eindeutig sein muss. Wenn die Daten nicht die Anforderungen erfüllen, die in der Klassenschlüsseldefinition festgelegt sind, kann das Objekt nicht in der Datenbank erzeugt werden.



Beispielsweise wird mit einem Klassenschlüssel für die Objektklasse `Application` festgelegt, dass die Kombination der Eigenschaften "Name" und "Version" eindeutig sein muss. Wenn ein Anwender über die Alfabet -Benutzeroberfläche versucht, eine Applikation zu erstellen, die sowohl hinsichtlich des Namens als auch der Version mit einer vorhandenen Applikation in der

Abbildung: Beispiel für die Mandantenmaske in einer Konfiguration mit 5 Mandanten. Das Objekt ist den Mandanten 2 und 3 zugeordnet.

Um zu beurteilen, ob ein Objekt für einen Anwender sichtbar ist, wird die Mandantenmaske bitweise mit einer über die Mandanteneinstellungen des aktuellen Benutzers generierten Mandantenmaske verglichen.

Bei der Auswertung der Mandanten-bezogenen Zugriffsberechtigungen für ein Objekt wird die Mandantenmaske des aktuellen Objekts mit der des aktuellen Mandanten verglichen, mit dem der Anwender angemeldet ist. Diese Mandantenmaske besteht aus einem Bit pro Mandant. Das Bit an der Position des aktuellen Mandanten und das erste Bit im String werden auf "1" gesetzt.



Die Mandantenmaske, die zur Bewertung der aktuellen Zugriffsberechtigungen des Anwenders generiert wird, ist nicht identisch mit der Mandantenmaske des Anwenders, der in der Eigenschaft `MANDATEMASK` der Objektklasse `Person` gespeichert ist. Über die in der Eigenschaft `MANDATEMASK` gespeicherte Mandantenmaske wird festgelegt, auf welche vom Anwender in der Alfabet-Datenbank gespeicherten Daten andere Anwender zugreifen dürfen. Die Mandantenmaske des aktuellen Anwenders, die zur Bewertung der Zugriffsberechtigungen generiert wurde, ist nicht in der Datenbank gespeichert, sondern wird während des Bewertungsvorgangs als Parameter generiert, der die aktuelle Umgebung beschreibt.

Während der Bewertung der Zugriffsberechtigungen wird die Mandantenmaske des aktuellen Anwenders mit der des Objekts bitweise verglichen. Wenn das Bit an einer Position sowohl für das Objekt als auch für den Anwender auf "1" gesetzt ist, ist das resultierende Bit an dieser Position "1". Wenn entweder der Anwender oder das Objekt oder beide einen Bitwert von "0" haben, wird das Bit an dieser Position auf "0" gesetzt.

Als Ergebnis erhält man einen Bit-String, der eine Ganzzahl darstellt. Ist im Bit-String mindestens ein Bit auf "1" gesetzt, dann ist die resultierende Ganzzahl größer "0" und das Objekt ist für den Anwender sichtbar.



Abbildung: Berechnung der Zugriffsberechtigung auf ein Objekt auf Basis der Mandantenmasken des aktuellen Anwendermandanten und des Objekts

Konfiguration von Objektklasseneigenschaften

Bei einigen Eigenschaftstypen können die Daten, die in die Datenbanktabelle eingegeben werden dürfen, aufgrund zusätzlicher Konfigurationen des Metamodells oder der Lösung eingeschränkt sein. Beim Importieren von Daten mithilfe des ADIF-Schemas müssen solche Einschränkungen berücksichtigt werden, wenn Daten zugeordnet und in die Datenbank importiert werden. Ansonsten ist der Datenimport möglicherweise ungültig.

Die Dokumentation des Standardmetamodells enthält von Software AG vorkonfigurierte Einschränkungen, die für jede Objektklasseneigenschaft gelten. Des Weiteren werden relevante Informationen bereitgestellt, wenn eine benutzerdefinierte Konfiguration möglich ist, die sich auf die Dateneingabe auswirken kann. In diesem Fall liegt es in Ihrer Verantwortung, über Anpassungen an der Lösungskonfiguration Ihres Unternehmens informiert zu sein, die für den Integrationsprozess relevant sind.

Folgende Konfigurationsprobleme sind für Objektklasseneigenschaften relevant:

- [Standardwerte für Objektklasseneigenschaften](#)
- [Größenbeschränkungen für Objektklasseneigenschaften](#)
- [Validierer für Objektklasseneigenschaften](#)
- [Objektklasseneigenschaften zugeordnete geschützte und benutzerdefinierte Aufzählungen](#)

Standardwerte für Objektklasseneigenschaften

Standardwerte können für Objektklasseneigenschaften vom **Eigenschaftstyp** `String`, `Real`, `Integer` und `Boolean` angegeben werden.

Wird ein Standardwert für eine Objektklasseneigenschaft mithilfe des Attributs **Standardwert** definiert, wird der Standardwert für jedes neue Objekt in die Datenbanktabellenspalte für die Objektklasseneigenschaft eingegeben, falls von dem Anwender, der das Objekt erstellt, kein anderer Wert angegeben wurde.

Standardwerte für benutzerdefinierte Objektklasseneigenschaften werden nicht automatisch während des Imports festgelegt. Wenn Sie mithilfe des ADIF-Schemas Daten in die Alfabet-Datenbank importieren, muss mit der Importdefinition der Standardwert für benutzerdefinierte Objektklasseneigenschaften angegeben werden, für die kein relevanter Wert definiert ist.

Es liegt in Ihrer Verantwortung, über Standardwerte informiert zu sein, die in der Lösungskonfiguration Ihres Unternehmens für benutzerdefinierte Objektklasseneigenschaften konfiguriert sind.

Größenbeschränkungen für Objektklasseneigenschaften

Größenbeschränkungen können für den Wert angegeben werden, der für Objektklasseneigenschaften vom **Eigenschaftstyp** `String` definiert ist. Falls für das Attribut **Größe** einer Objektklasseneigenschaft ein Wert definiert ist, ist die Variable `$Size$` für `varchar` zum Speichern der Eigenschaft gleich **Größe** +1.

Validierer für Objektklasseneigenschaften

Validierer können für Objektklasseneigenschaften vom **Eigenschaftstyp** `String` angegeben werden. Das Attribut **Validierer** der Objektklasseneigenschaft lässt die Angabe eines regulären Ausdrucks zu, der das für diese Eigenschaft erforderliche Format durchsetzt.

Es liegt in Ihrer Verantwortung, über benutzerdefinierte Validierer informiert zu sein, die in der Lösungskonfiguration Ihres Unternehmens für Objektklasseneigenschaften konfiguriert sind.



Weitere Informationen über Syntax-Vorgaben für reguläre Ausdrücke finden Sie unter [http://msdn.microsoft.com/en-us/library/hs600312\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/hs600312(VS.71).aspx).

Objektklasseneigenschaften zugeordnete geschützte und benutzerdefinierte Aufzählungen

Durch Aufzählungen kann ein Bereich zulässiger Werte für eine Eigenschaft definiert werden. Aufzählungen können für Objektklasseneigenschaften der **Eigenschaftstypen** `String`, `StringArray` und `Integer` in Alfabet Expand definiert und angegeben werden. Eine Aufzählung kann für mehrere Objektklasseneigenschaften von mehreren Objektklassen wiederverwendet werden.

In Alfabet Expand sind unterhalb des Explorerknotens **Aufzählungen** auf der Registerkarte **Metamodell** vorhandene geschützte und benutzerdefinierte Aufzählungen sichtbar. Jede Aufzählung verfügt über ein Attribut **Name**, und die zulässigen Werte werden mithilfe des Attributs **Elemente** definiert. Sobald die Aufzählung mithilfe ihres Attributs **Aufzählung** der relevanten Objektklasseneigenschaft zugeordnet ist, sind die zulässigen Werte, die Anwender für eine Objektklasseneigenschaft auswählen können, auf die für die Aufzählung definierten Werte beschränkt.

Geschützte Aufzählungen sind Bestandteil des Standardmetamodells und wurden von Software AG vor-konfiguriert. Diese Aufzählungen können nicht gelöscht werden. Es können jedoch die zulässigen Werte bearbeitet werden, die im Attribut **Elemente** definiert sind. Wenn die Konfiguration einer geschützten Aufzählung von Ihrem Unternehmen angepasst wurde, unterscheiden sich die zulässigen Werte, die der Objektklasseneigenschaft zugeordnet werden können, von den standardmäßig zulässigen Werten, die in der Dokumentation des Metamodells beschrieben sind.

Es liegt in Ihrer Verantwortung, über Anpassungen informiert zu sein, die in der Lösungskonfiguration Ihres Unternehmens an geschützten Aufzählungen vorgenommen wurden.



Informationen über das Konfigurieren geschützter und benutzerdefinierter Aufzählungen finden Sie im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* im Abschnitt *Definieren von geschützten und benutzerdefinierten Aufzählungen*.

XML-basierte Definitionen

Bei einigen Objektklasseneigenschaften ist die Eingabe zulässiger Daten auf Werte beschränkt, die mit einem XML-Objekt verbunden sind, das in Alfabet Expand konfiguriert sein muss. Zum Beispiel verfügen mehrere Objektklassen über die Eigenschaft "Status", mit der der Release-Status des Objekts im Genehmigungsprozess definiert werden kann. Die möglichen Werte für Release-Statusdefinitionen werden im XML-Objekt **ReleaseStatusDefs** konfiguriert. Wenn Sie einen anderen Wert verwenden als diejenigen, die in solch einer XML-Definition für die Objektklasse angegeben sind, kann ein Fehler auftreten.

Es wird daher dringend empfohlen, die Dokumentation des Metamodells zu konsultieren, bevor Sie mithilfe des ADIF-Schemas Werte für eine Objektklasseneigenschaft zuordnen und importieren. Die Dokumentation des Metamodells enthält detaillierte Informationen über potenzielle Abhängigkeiten zwischen XML-Objekt-Konfigurationen und der Definition von Eingabewerten für Objektklasseneigenschaften.

Es liegt in Ihrer Verantwortung, über Anpassungen informiert zu sein, die in der Lösungskonfiguration Ihres Unternehmens an XML-Objektdefinitionen vorgenommen wurden.



Informationen über das Konfigurieren von XML-Objekten finden Sie im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* im Abschnitt *Arbeiten mit XML-Objekten*.

Referenz- und Bewertungsdaten sowie Referenzdatenzuordnung

Die zulässigen Werte für einige Objektklasseneigenschaften werden durch die Konfigurationen bestimmt, die an der zugehörigen Objektklasse durchgeführt wurden. Diese Konfigurationen werden im Modul **Konfiguration** von Alfabet durchgeführt. Bewertungen und deren Kennzahlen, Kosten- und Nutzenarten, Rollentypen und Diagrammansichten sind Beispiele für einige der Objekte, die auf Konfigurationen basieren, die einem Satz von Objektklassen zugewiesen werden können, der in Ihrem Unternehmen verwendet wird.

Solche Konfigurationsobjekte werden unabhängig von einer Objektklasse erzeugt. Die Konfigurationsobjekte müssen dann der/den Objektklasse/n zugeordnet werden, für die sie relevant sind. Dadurch können Sie Referenz- oder Bewertungsdaten erstellen, die für einen bestimmten Objektklassenkontext spezifisch sind. Zum Beispiel unterscheiden sich Kennzahlen, die zum Bewerten von Anwendungen erforderlich sind, in der Regel von den Kennzahlen, die zum Bewerten von Geräten erforderlich sind.

Es wird daher dringend empfohlen, die Dokumentation des Metamodells zu konsultieren, bevor Sie mithilfe des ADIF-Schemas Werte für eine Objektklasseneigenschaft zuordnen und importieren. Die Dokumentation des Metamodells enthält detaillierte Informationen über potenzielle Abhängigkeiten zwischen Konfigurationsobjekten und der Definition von Eingabewerten für Objektklasseneigenschaften.

Es liegt in der Verantwortung des Lösungsentwicklers Ihres Unternehmens, alle Informationen über relevante Konfigurationsobjekte an die Personen weiterzugeben, die für den Integrationsprozess verantwortlich sind.

Kapitel 3: Konfigurieren von ADIF-Schemata

Dieses Kapitel bietet allgemeine Informationen zur Arbeit mit den Benutzeroberflächen, mit denen Sie ADIF-Schemata konfigurieren können. Der Prozess für den Datenimport und -export und die erforderliche Struktur und Konfiguration von ADIF-Schemata für den Import und Export von Daten unterscheiden sich. Deshalb findet sich die ausführliche Beschreibung, wie die Datenmanipulation über ADIF definiert wird, in separaten Kapiteln, die Kenntnisse der in diesem Kapitel beschriebenen Konfigurationsgrundlagen voraussetzen:

- *Konfigurieren von Datenimporten mit ADIF*
- *Konfigurieren von Datenexporten mit ADIF*

Das ADIF-Schema ist XML-basiert. Die Konfiguration des ADIF-Schemas kann in einer von Software AG entwickelten Benutzeroberfläche oder direkt in einem Texteditor durchgeführt werden. Beide Optionen werden nachfolgend beschrieben.

Konfigurieren von ADIF über den ADIF-Explorer

ADIF-Import- und Exportschemata können im Tool Alfabet Expand oder in einer separaten ADIF-Benutzeroberfläche erstellt und verwaltet werden. Das Design ist in beiden Fällen identisch.

ADIF-Import- und Exportschemata werden über einen Explorer-Baum verwaltet. Der Explorer weist zwei Knoten auf Root-Ebene auf: **ADIF Schemes** und **Metamodell**. Mit dem untergeordneten Baum **ADIF Schemes** können Sie ADIF-Import- und Exportschemata erstellen und bearbeiten. Die Registerkarte **Metamodell** bietet Informationen zum Alfabet -Metamodell, die Sie beim Definieren Ihrer ADIF-Schemata unterstützen.

Der untergeordnete Baum "ADIF-Schemata" des ADIF-Explorers

Jedes von Ihnen erstellte ADIF-Import- oder Exportschema wird im Explorer unterhalb des Root-Knotens **ADIF-Schemata** aufgeführt. Das ADIF-Schema wird konfiguriert, indem dem Explorer Befehls-elemente als Unterelemente des ADIF-Schemas hinzugefügt und die relevanten Attribute dieser Unterelemente festgelegt werden.



Alle im ADIF-Explorer von Alfabet Expand vorgenommenen Änderungen werden in der Alfabet-Datenbank gespeichert. Die Schaltfläche **Speichern** ist nicht erforderlich, um die Änderungen am ADIF-Schema zu speichern.

Hinzufügen und Konfigurieren von ADIF-Schemaelementen

Neue ADIF-Schemaelemente können mit dem für die vorhandenen Explorer-Knoten verfügbaren Kontextmenü dem Explorer-Baum hinzugefügt werden. Die Methode zum Hinzufügen eines neuen ADIF-Schemaelements ist immer gleich, unabhängig vom Elementtyp (z. B. Attribute oder SQL-Befehle). So fügen Sie ein neues ADIF-Schemaelement zu einem vorhandenen Explorer-Knoten hinzu:

- 1) Klicken Sie im Explorer mit der rechten Maustaste auf das Explorer-Knotenelement, dem Sie ein neues ADIF-Schemaelement hinzufügen möchten, und wählen Sie **< Einfügeoption > erzeugen**.

Das Kontextmenü bietet nur die Optionen, die für das ausgewählte Explorer-Knotenelement zutreffen.

Im Explorer werden der Name des Explorer-Knotenelements und ein Symbol angezeigt, das den Elementtyp zeigt. Das Symbol hat eine grüne Markierung, um anzuzeigen, dass das neue Explorer-Knotenelement aktiv ist. Neue Explorer-Knotenelemente sind standardmäßig aktiv. Sie können alle Explorer-Knotenelemente deaktivieren, um die Ausführung zu verhindern, indem Sie das Knotenattribut "Ist aktiv" auf "False" setzen. Das Explorer-Knotenelement wird dann mit einem roten Sternchen markiert:



Abbildung: Aktive und inaktive ADIF-Schemaknoten

Wenn ein XML-Elementknoten voraussichtlich eine große Anzahl von Elementtypen als untergeordnete Knoten hat, werden dem XML-Element automatisch Ordnerknoten hinzugefügt, um den Inhalt zu strukturieren. Diese Ordner sind nur im Explorer verfügbar und wirken sich nicht auf die Speicherung der Daten im XML aus.

Wenn Sie im Explorer einen Knoten anklicken, werden die Attribute des XML-Elements, das durch den Knoten dargestellt wird, im Eigenschaftsfenster auf der rechten Seite des Explorer-Baums angezeigt. Sie können alle in Schwarz angezeigten Attribute bearbeiten. Die in Grau angezeigten Attribute dienen nur der Ansicht.

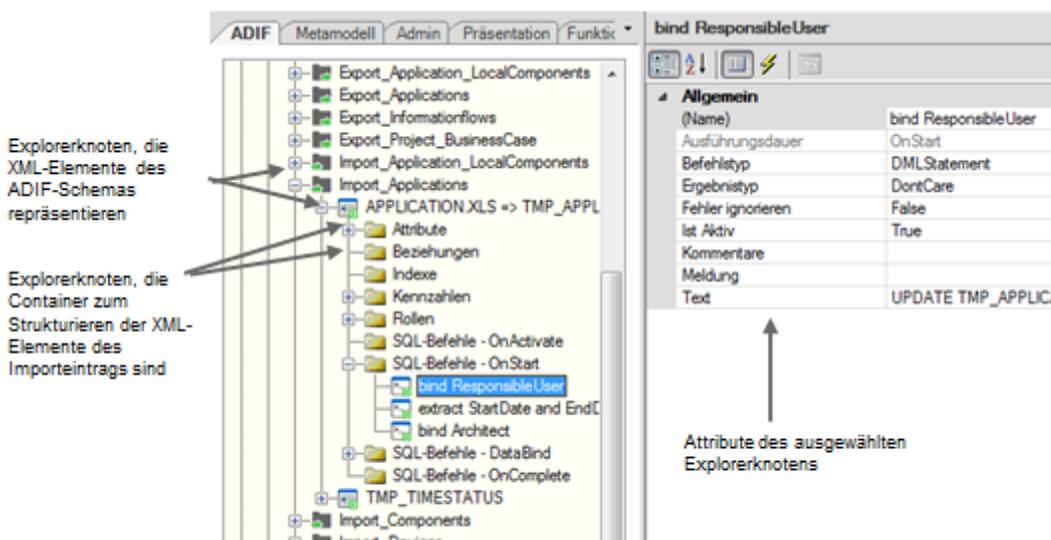


Abbildung: Der untergeordnete Baum "ADIF Schemes" des ADIF-Explorers

Wenn Sie ein neues ADIF-Schema erzeugen, ist das ADIF-Schemaelement öffentlich, das heißt, es ist komplett editierbar. Alfabet bietet vordefinierte ADIF-Standardschemata für spezielle Importaktionen, zum Beispiel im Kontext von Schnittstellen mit Applikationen von Drittanbietern. Diese vordefinierten ADIF-Schemata sind entweder privat, das heißt, sie können überhaupt nicht verändert werden, oder sie sind geschützt, was bedeutet, dass Sie nur eine Teilmenge der Attribute der vorkonfigurierten untergeordneten Elemente des ADIF-Schemata bearbeiten können. Es ist möglich, neue ADIF-Elemente zu geschützten ADIF-Schemata hinzuzufügen. Die vom Kunden hinzugefügten ADIF-Elemente sind öffentlich und vollständig editierbar. Private und geschützte ADIF-Schemata haben ein Schlosssymbol, das ihren Bearbeitbarkeitsstatus anzeigt. Das Schloss ist orangefarben bei geschützten ADIF-Schemata und rot bei privaten

ADIF-Schemata. Innerhalb eines geschützten ADIF-Schemas ist die Bearbeitbarkeit der Attribute nur in den Attributfenstern der untergeordneten Elemente sichtbar. Deaktivierte Attribute werden in Grau angezeigt.

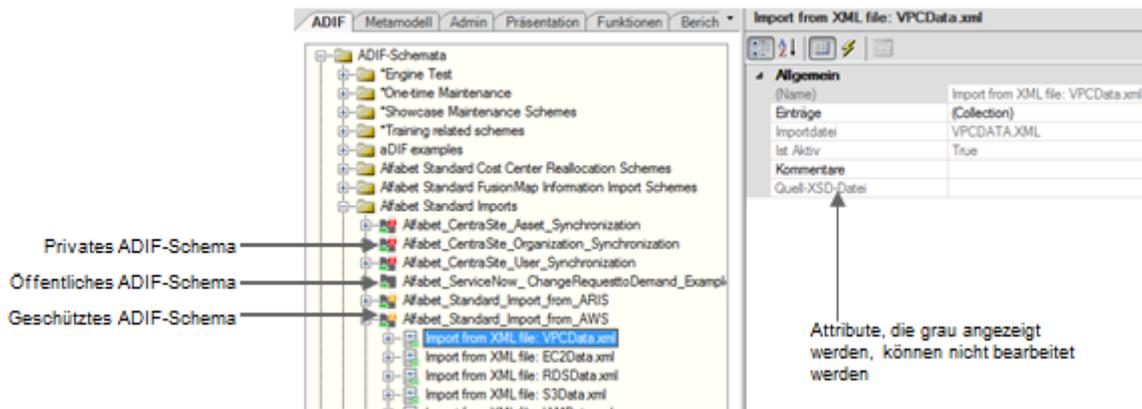


Abbildung: Private, geschützte und öffentliche ADIF-Schemata im ADIF-Explorer

Kopieren oder Verschieben von ADIF-Schemaelementen

Das Kontextmenü der Knoten im Explorer beinhaltet auch eine Funktion zum Kopieren und Einfügen, um bereits konfigurierte Elemente in einem anderen Kontext weiterzuverwenden.

Anstatt Elemente zu kopieren, können Sie die Elemente auch durch Ausschneiden und Einfügen an einen anderen Speicherort verschieben.

- 1) Klicken Sie im Explorer mit der rechten Maustaste auf das Element, das Sie weiterverwenden möchten, und wählen Sie **Kopieren** aus, um es in ein anderes Element zu kopieren, oder **Ausschneiden**, um es in ein anderes Element zu verschieben.
- 2) Klicken Sie im Explorer mit der rechten Maustaste auf das übergeordnete Element, dem Sie das kopierte Element als untergeordnetes Element hinzufügen möchten, und wählen Sie **Einfügen**. Das Element wird unter dem Namen <alter Name>_1 an den neuen Speicherort kopiert.
- 3) Klicken Sie optional im Explorer auf das hinzugefügte Element, und ändern Sie das Attribut **Name** im Eigenschaftsfenster.



Beachten Sie, dass dieses Verhalten nicht für das Kopieren von SQL-Befehlen in andere SQL-Befehle gilt. Wenn Sie versuchen, einen SQL-Befehl in einen anderen SQL-Befehl zu kopieren, wird der aktuelle SQL-Befehl überschrieben, anstatt den kopierten SQL-Befehl als untergeordnetes Element hinzuzufügen. Um einen SQL-Befehl in einen anderen SQL-Befehl zu kopieren, fügen Sie dem übergeordneten SQL-Befehl des Ziels einen neuen untergeordneten SQL-Befehl hinzu, und fügen Sie den kopierten SQL-Befehl in den neu erstellten untergeordneten SQL-Befehl ein.

Ändern der Reihenfolge von untergeordneten Elementen eines ADIF-Schemaelements

Die Elemente eines ADIF-Schemas werden in der konfigurierten Reihenfolge ausgeführt. Sie können die Reihenfolge der Ausführung von untergeordneten Elementen innerhalb eines Knotens im Explorer der ADIF-Konfigurationsschnittstelle ändern, indem Sie die Reihenfolge der Knoten ändern.



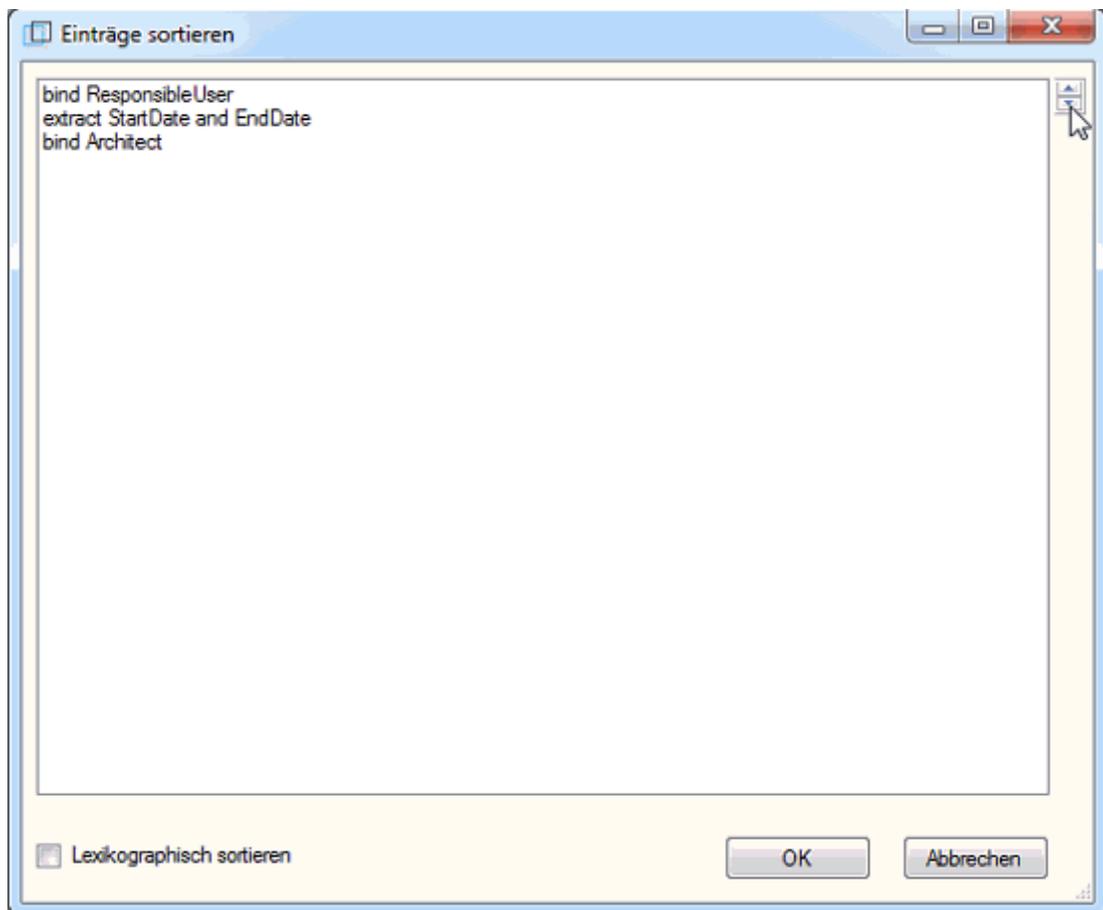
Für den allgemeinen Mechanismus gelten folgende Ausnahmen:

- Der Mechanismus zum Ändern der Sortierreihenfolge ist für alle Knoten gleich, mit Ausnahme der untergeordneten Knoten in den Unterordnern **Beziehungen** und **Indexe** der ADIF-Importeinträge
- Die Sortierreihenfolge der Unterordner für SQL-Befehle oder Attributdefinitionen in einem ADIF-Import- oder Exporteintrag kann nicht geändert werden.

So ändern Sie die Reihenfolge der untergeordneten Knoten in einem ADIF-Schema, einer Import- oder Exportgruppe oder eines der Ordner für SQL-Befehle:

- 1) Klicken Sie auf den Knoten, für den Sie die Reihenfolge der untergeordneten Elemente ändern möchten.
- 2) Klicken Sie im Attributbereich auf das Attribut, das Sie sortieren möchten, und klicken Sie auf die Schaltfläche **Durchsuchen** , um den Editor zu öffnen, der das Sortieren ermöglicht. Für folgende Attribute ist das Sortieren möglich:
 - Knoten ADIF-Importschema und ADIF-Exportschema Attribut **Elemente**
 - Knoten Importgruppe und Exportgruppe: Attribut **Einträge**
 - Attributordner der Importeinträge: Attribut **Attribute**.
 - Ordner **SQL-Befehle - < Aktionstyp >**: Attribut **Untergeordnete Befehle**.

Ein Editor wird geöffnet, der eine Liste der untergeordneten Knoten enthält.



- 3) Klicken Sie auf einen der untergeordneten Knoten in der Liste, und klicken Sie auf die Schaltflächen **Nach oben/Nach unten**  in der oberen rechten Ecke, um die ausgewählten Elemente in der Liste nach oben oder unten zu verschieben. Sortieren Sie alle untergeordneten Knoten nach Bedarf.



Alternativ können Sie das Kontrollkästchen **Sortieren** in der linken unteren Ecke aktivieren, um alle Elemente alphabetisch zu sortieren.

- 4) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern. Die Reihenfolge der untergeordneten Knoten im Explorer-Baum ändert sich gemäß Ihren Einstellungen.

So ändern Sie die Reihenfolge der untergeordneten Knoten im Ordner **Beziehungen** oder **Indexe** eines ADIF-Importeintrags:

- 1) Erweitern Sie im Explorer den Knoten, für den Sie die Sortierreihenfolge der untergeordneten Knoten ändern möchten.
- 2) Klicken Sie auf den untergeordneten Knoten, den Sie innerhalb des Ordners verschieben möchten, und ziehen Sie ihn an die neue Position.

Löschen von ADIF-Schemaelementen

Sie können entweder komplette ADIF-Schemata oder Elemente von ADIF-Schemata löschen. Wenn Sie einen Knoten aus dem Explorer löschen, werden auch alle untergeordneten Knoten gelöscht.

So löschen Sie ein ADIF-Schema:

- 1) Klicken Sie im Explorer mit der rechten Maustaste auf das ADIF-Schema, das Sie löschen möchten, und wählen Sie **Nutzung anzeigen** aus. Es wird ein neues Fenster geöffnet, in dem alle Ereignisvorlagen und Schaltflächen aufgeführt sind, die zum Auslösen der Ausführung des ADIF-Schemas konfiguriert sind.
- 2) Wenn in der Liste eine Nutzung des ADIF-Schemas angezeigt wird, entfernen Sie die Konfigurationen, die dieses ADIF-Schema verwenden, bevor Sie das ADIF-Schema löschen.
- 3) Klicken Sie im Explorer mit der rechten Maustaste auf das ADIF-Schema, und wählen Sie **Löschen** aus. Das Element und all seine Unterelemente werden gelöscht.

So löschen Sie ein ADIF-Schemaelement:

- 1) Klicken Sie im Explorer mit der rechten Maustaste auf das ADIF-Schemaelement, das Sie löschen möchten, und wählen Sie **Löschen** aus.

Strukturieren von ADIF-Schemata in Gruppen

Mithilfe der ADIF-Konfigurationsbenutzeroberfläche können Sie ADIF-Schemata in Gruppenknoten strukturieren. Diese Elemente sind nicht Teil der ADIF-XML-Spezifikation, sondern leere Explorer-Knoten, die zur visuellen Strukturierung der ADIF-Schemadefinitionen verwendet werden.

Eine Gruppe wird erstellt, indem mindestens ein ADIF-Schema zur Gruppe hinzugefügt wird.

- 1) Wählen Sie im ADIF-Explorer mindestens eines der ADIF-Schemata, die Sie der neuen Gruppe hinzufügen möchten.

- 2) Geben Sie im Attributfenster des ADIF-Schemas den Namen der Gruppe in das Attribut **Gruppe** ein. Es wird ein neuer Gruppenknoten mit dem angegebenen Namen als übergeordnetes Element des ausgewählten ADIF-Schemas im Explorer erstellt.
- 3) Sie können jetzt folgende Aktionen ausführen:
 - Um ein anderes bereits vorhandenes ADIF-Schema der Gruppe hinzuzufügen, setzen Sie das Attribut **Gruppe** des ADIF-Schemas auf den Namen der vorhandenen Gruppe.
 - Um der Gruppe ein neues ADIF-Schema hinzuzufügen, klicken Sie mit der rechten Maustaste auf den Gruppenknoten, und wählen Sie entweder **Importschema erstellen** oder **Exportschema erstellen**.
 - Löschen Sie den Gruppennamen aus dem Attribut **Gruppe**, um ein ADIF-Schema auf die Root-Ebene der Hierarchie zu verschieben.
 - Ändern Sie den Gruppennamen im Attribut **Gruppe** des Schemas, um ein ADIF-Schema in einer Gruppe zu verschieben. Die Zielgruppe kann entweder eine vorhandene Gruppe oder ein neuer Name sein, der zur Erstellung einer neuen Gruppe im Explorer führt.
 - Ändern Sie das Attribut **Gruppe** aller ADIF-Schemata in der Gruppe, um eine Gruppe zu löschen. Ein Gruppenknoten ist so lange vorhanden, wie er in einem der ADIF-Schemata definiert ist.

Definieren von SQL-Abfragen für SQL-Befehle

ADIF-Import- und Exportschemata werden hauptsächlich über SQL-Abfragen konfiguriert. Diese Abfragen werden im Attribut **Text** des SQL-Befehlselements innerhalb des ADIF-Schemas definiert. Wenn Sie im Feld für das Attribut **Text** auf die Schaltfläche **Durchsuchen**  klicken, wird ein Texteditor geöffnet, mit dem Sie die Abfrage definieren können.

Definieren von SQL-Abfragen für verschiedene Datenbankserver in einem SQL-Befehl

Die SQL-Syntax hängt von dem Datenbankserver ab, auf dem die Abfrage ausgeführt wird. Sie sollten ein ADIF-Schema auf Datenbanken ausführen, die sich entweder auf einem Microsoft SQL Server oder einem Oracle -Datenbankserver befinden. Für diese Situation können Sie die Abfrage für einen SQL-Befehl in verschiedenen Syntaxen innerhalb desselben ADIF-Schemas definieren. Jede Version der definierten Abfrage muss mit dem Namen des Servertypen in einem Kommentar beginnen (eine separate Zeile, die mit zwei Bindestrichen und einem Leerzeichen beginnt).

Folgende Definitionen von Servertypen sind zulässig:

- SQLSERVER
- ORACLE



Zum Beispiel eine Abfrage, die für die Ausführung entweder auf einem Microsoft SQL Server oder einem Oracle -Datenbankserver definiert ist:

```
-- SQLSERVER
UPDATE TMP_TABLE SET TMP_TABLE.ATTR1 = APPLICATION.ATTR2 FROM
APPLICATION WHERE APPLICATION.REFSTR = TMP_TABLE.REFSTR
```

```
-- ORACLE

UPDATE TMP_TABLE SET (TMP_TABLE.ATTR1, TMP_TABLE.ATTR2) = (SELECT
ATTR1, ATTR2 FROM APPLICATION WHERE TMP_TABLE.REFSTR =
APPLICATION.REFTR)
```

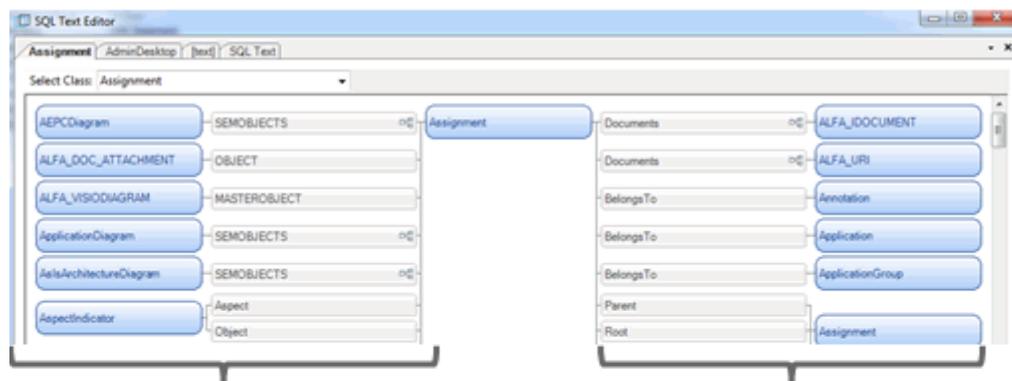
Hilfe beim Definieren von SQL-Abfragen

Der Texteditor für die Definition von SQL-Abfragen in der ADIF-Konfigurationsbenutzeroberfläche stellt in getrennten Registerkarten Hilfe für die Definition der SQL-Abfrage bereit.

- Referenzen auf andere Objektklassen

Wenn Sie den Texteditor zum ersten Mal öffnen, wird eine Registerkarte **AdminDesktop** angezeigt. Die Objektklasse AdminDesktop ist die erste Klasse in alphabetischer Reihenfolge im Alfabet -Metamodell. Wählen Sie im Feld **Klasse auswählen** die Objektklasse, deren Referenzen auf andere Objektklassen Sie sehen möchten. Es öffnet sich eine neue Registerkarte mit dem Namen der ausgewählten Objektklasse. Auf dieser Registerkarte wird eine Grafik angezeigt, die die Beziehungen zwischen der ausgewählten Objektklasse und anderen Objektklassen zeigt.

Das ausgewählte Objekt wird in der Mitte der Grafik angezeigt. Auf der linken Seite werden alle Objektklassen aufgeführt, die die ausgewählte Objektklasse über eine oder mehrere ihrer Eigenschaften referenzieren. Die Eigenschaft, die die Referenz herstellt, wird in einem weißen Kästchen zwischen den beiden Klassen angezeigt. Auf der rechten Seite werden alle Eigenschaften der ausgewählten Klasse in einem weißen Kästchen angezeigt, die Referenzen zu anderen Objektklassen speichern. Die Zielklassen sind im blauen Kästchen auf der rechten Seite aufgeführt.



Objektklassen mit einer Referenz auf „Assignment“ (blau) und die Eigenschaften, über die die Referenz definiert wird (weiß)

Eigenschaften der Objektklasse „Assignment“ (weiß), die eine Referenz auf eine andere Objektklasse definieren, und die Ziel-Objektklasse (blau)

Wenn Sie in der Grafik auf eine referenzierte Objektklasse doppelklicken, öffnet sich eine neue Registerkarte, die die Referenzen der ausgewählten Objektklasse zu anderen Objektklassen anzeigt.

- Objektklassendokumentation

Wenn Sie die Registerkarte **[text]** öffnen und im Feld **Klasse auswählen** eine Objektklasse auswählen, wird das Hilfe-Dokument für diese Objektklasse angezeigt. Die Dokumentation enthält eine Beschreibung des Zwecks der Objektklasse sowie alle Eigenschaften der Objektklasse und führt alle relevanten Attribute in einer Tabelle auf. Darüber hinaus werden Informationen zu

Klassenabhängigkeiten basierend auf Datenbankauslösern bereitgestellt. Die Dokumentation enthält außerdem die Regeln, die definieren, welche abhängigen Objekte gelöscht werden, wenn ein Objekt der ausgewählten Klasse gelöscht wird.

Der untergeordnete Baum "Metamodell" des ADIF-Explorers

Der untergeordnete Baum **Metamodell** des ADIF-Explorers bietet Informationen zum Alfabet –Metamodell, die bei der Konfiguration der ADIF-Benutzeroberfläche hilfreich sein können. Der Explorer-Knoten **Metamodell** hat zwei untergeordnete Knoten: **AUFZÄHLUNGEN** und **Klassenmodell**.

AUFZÄHLUNGEN

Der untergeordnete Baum **Aufzählungen** listet alle aktuell konfigurierten Aufzählungen auf. Wenn eine Eigenschaft auf einer Aufzählung basiert, können Sie überprüfen, welche Eingabewerte für die Eigenschaften zulässig sind, indem Sie im Explorer auf die Aufzählung klicken und dann auf die Schaltfläche **Durchsuchen**  für das Attribut **Elemente** klicken. Ein Texteditor wird geöffnet, der jeden zulässigen Wert in einer separaten Zeile aufführt.

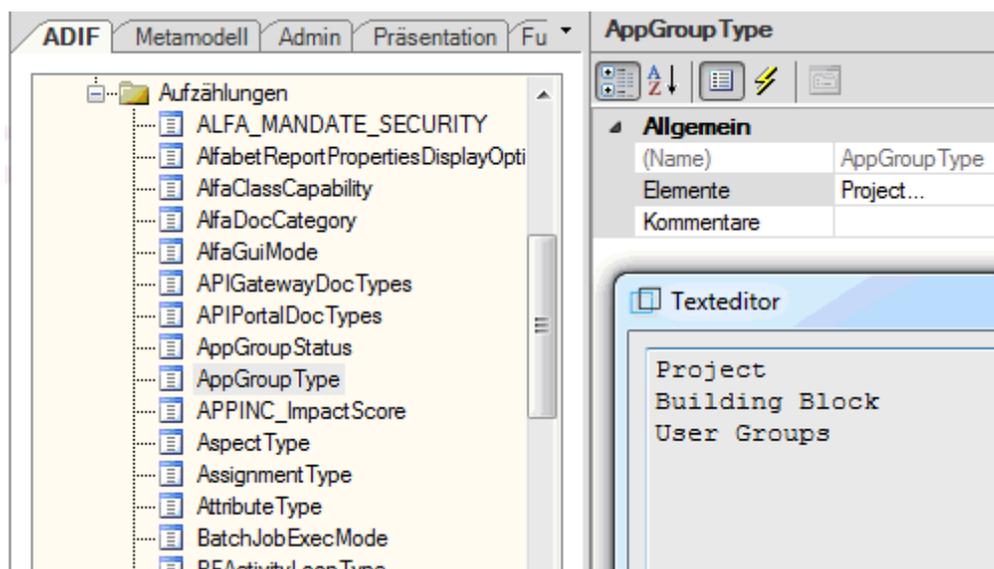


Abbildung: Zulässige Werte für Eigenschaften, die auf der Aufzählung *AppGroupType* basieren

Klassenmodell

Im untergeordneten Baum **Klassenmodell** werden alle Klassen des Alfabet –Metamodells aufgelistet. Die Klassen werden entsprechend ihrer Nutzung in der Anwendung strukturiert. Die im Ordner **SEMANTIC-CLASS** aufgeführten Klassen sind für ADIF relevant. Für den Ordner **SEMANTICCLASS** sind die folgenden Unterordner definiert:

- **ALFA_QUESTIONARY**: Kundendefinierte, temporäre Klassen, die für die Datensammlung über die Fragenkatalog-Funktionen relevant sind.

- **Artifakt:** Klassen, die zum Speichern von Daten über die IT-Infrastruktur des Kunden verwendet werden. Alle **Artifakt** -Klassen, für die ein verantwortlicher Anwender definiert werden kann, werden im Unterordner **ArtifactAuthorized** des Ordners **Artifakt** aufgelistet.
- **ITClass:** Objektklassen, die Funktionen unterstützen Dazu gehört zum Beispiel die Klasse "TimeStatus", in der der Lebenszyklus von Objekten gespeichert wird, oder die Klasse "RoleType", in der Informationen darüber gespeichert werden, welche Rollen ein Anwender in Bezug auf ein Objekt einnehmen kann.

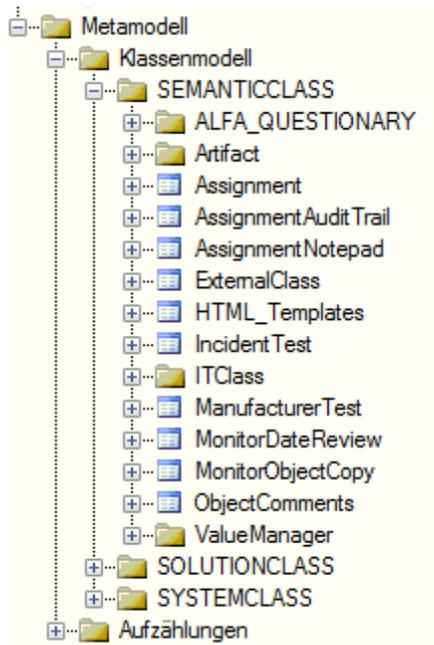


Abbildung: Klassenmodell-Explorer mit erweitertem Ordner SEMANTICCLASS

Die folgenden Informationen zu jeder Objektklasse sind über den **Klassenmodell** -Explorer verfügbar:

- Objektklassenattribute

Wenn Sie im Explorer eine Objektklasse anklicken, werden die Attribute der Objektklasse im Attributbereich auf der rechten Seite des Explorers angezeigt.

The screenshot shows the ADIF software interface. On the left is a tree view of the class hierarchy under 'ADIF-Schemata'. The 'Assignment' class is selected. On the right is the 'Assignment' properties window, showing the 'Allgemein' (General) tab with a table of attributes.

Assignment	
(Name)	Assignment
Automatisch verwaltet	False
Hat Tabelle	True
Hinweis	An assignment is a task that is
Historie	False
ID-Präfix	ASMT
Integritäts-Info	<ClassIntegrityInfo>...
Kann Mandanten haben	False
Kommentare	
Objektclassenstereotypen	
Technischer Name	ASSIGNMENT
Titel	Assignment



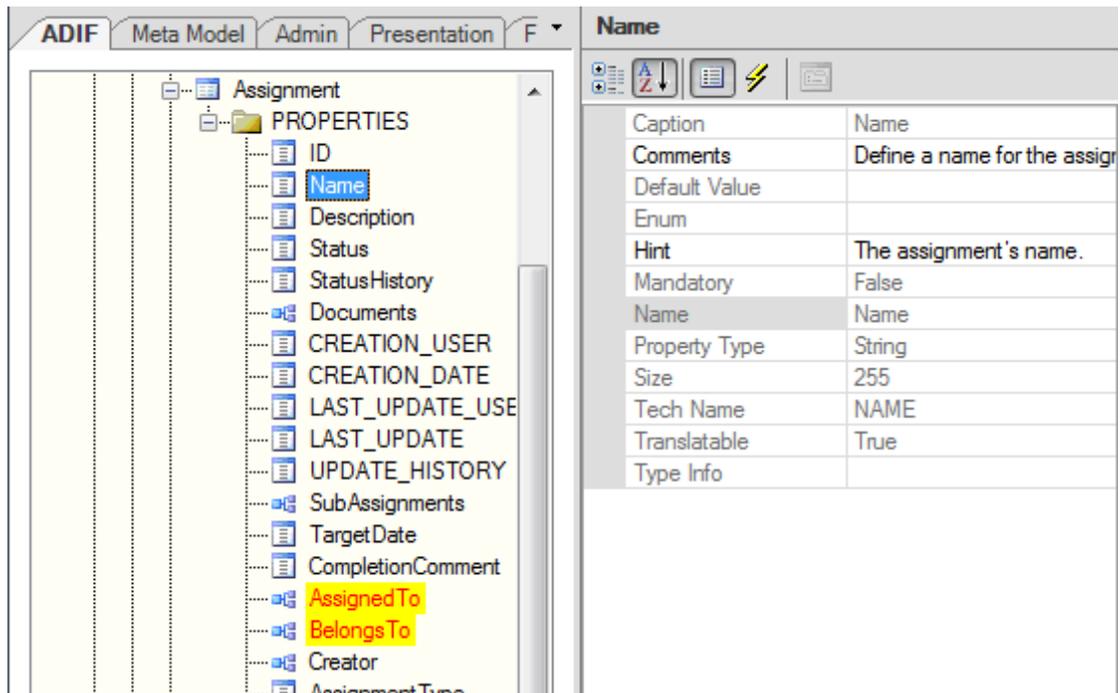
Hinweis: Objektklassen, für die das Attribut **Automatisch verwaltet** auf `true` gesetzt ist, sind Objektklassen, die nur durch Mechanismen geändert werden dürfen, die von den Komponenten der Alfabet-Software angestoßen werden. Sie dürfen nicht durch einen ADIF-Import oder Komponenten von Drittanbietern geändert werden.

- Objektklasseneigenschaften

Wenn Sie im Explorer einen Objektklassenknoten erweitern, wird ein Ordner **Eigenschaften** angezeigt. Wenn Sie den Knoten **Eigenschaften** erweitern, werden die im Explorer aufgeführten Eigenschaften der Objektklasse angezeigt. Pflichteigenschaften werden gelb hervorgehoben und in roten Buchstaben geschrieben. Wenn Sie einen Eigenschaftsknoten anklicken, werden die Attribute der Eigenschaft im Attributbereich auf der rechten Seite des Explorers angezeigt.



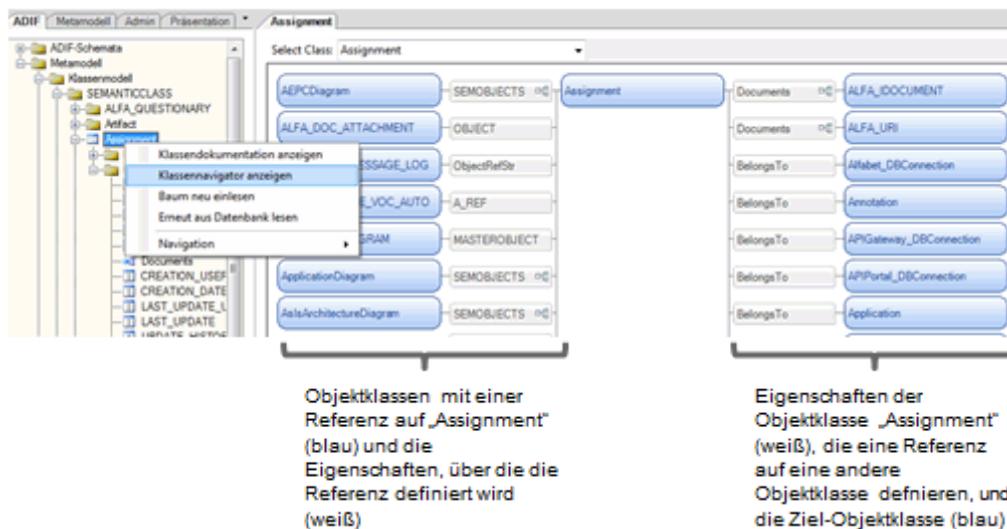
Die Eigenschaft "Name" der Objektklassen, die unter dem Knoten **Artifakt** aufgeführt sind, werden von der übergeordneten Artefaktklasse geerbt. Sie ist als obligatorisch gekennzeichnet, da sie für die meisten untergeordneten Objektklassen obligatorisch ist. Die untergeordneten Klassen erben alle Attribute der Eigenschaft der übergeordneten Klasse. Deshalb wird das Attribut "Name" auch für die wenigen untergeordneten Klassen als obligatorisch gekennzeichnet, die kein Attribut "Name" erfordern.



- Referenzen auf andere Objektklassen

Wenn Sie im Explorer mit der rechten Maustaste auf eine Objektklasse klicken und die Option **Klassennavigator anzeigen** auswählen, wird zwischen dem Explorer-Bereich und dem Attributfenster ein neues Fenster geöffnet. Die Beziehung der ausgewählten Objektklasse zu anderen Objektklassen über Referenzen wird in einer Grafik angezeigt.

Das ausgewählte Objekt wird in der Mitte der Grafik angezeigt. Auf der linken Seite werden alle Objektklassen aufgeführt, die die ausgewählte Objektklasse über eine oder mehrere ihrer Eigenschaften referenzieren. Die Eigenschaft, die die Referenz herstellt, wird in einem weißen Kästchen zwischen den beiden Klassen angezeigt. Auf der rechten Seite werden alle Eigenschaften der ausgewählten Klasse in einem weißen Kästchen angezeigt, die Referenzen zu anderen Objektklassen speichern. Die Zielklassen sind im blauen Kästchen auf der rechten Seite aufgeführt.



- Objektklassendokumentation

Wenn Sie im Explorer mit der rechten Maustaste auf eine Objektklasse klicken und die Option **Klassendokumentation anzeigen** auswählen, wird zwischen dem Explorer-Bereich und dem Attributfenster ein neues Fenster geöffnet. Das für die Klasse bereitgestellte Hilfedokument wird angezeigt. Die Dokumentation enthält eine Beschreibung des Zwecks der Objektklasse sowie alle Eigenschaften der Objektklasse und führt alle relevanten Attribute in einer Tabelle auf. Darüber hinaus werden Informationen zu Klassenabhängigkeiten basierend auf Datenbankauslösern bereitgestellt. Die Dokumentation enthält außerdem die Regeln, die definieren, welche abhängigen Objekte gelöscht werden, wenn ein Objekt der ausgewählten Klasse gelöscht wird.

Konfigurieren von ADIF über XML-Bearbeitung

Die Konfiguration des ADIF-Schemas ist XML-basiert. Anstatt ADIF-Schemata in der ADIF- oder Alfabet Expand -Benutzeroberfläche zu konfigurieren, können die Schemata in einem XML-Editor erstellt oder bearbeitet werden. Der ADIF-Explorer-Baum bietet eine Möglichkeit, das Schema in einem internen Editor zu öffnen, der Hilfe und Debuggingfunktionen bereitstellt. Alternativ können Sie jedes Schema exportieren und es in einem beliebigen Standard-XML-Editor bearbeiten, oder Sie können ein Schema in einem XML-Editor komplett neu erstellen und es in die ADIF-Benutzeroberfläche importieren.

Interner XML-Editor

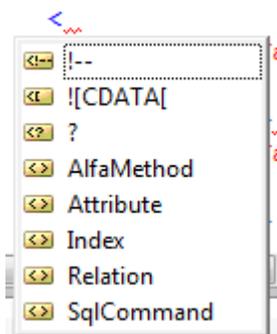
So öffnen Sie das XML-Objekt des ADIF Schemas im internen XML-Editor der ADIF-Benutzeroberfläche:

- 1) Klicken Sie im ADIF-Explorer mit der rechten Maustaste auf ein ADIF-Schema, und wählen Sie **XML-Editor öffnen**. Der XML-Editor wird in einem neuen Fenster geöffnet.

Der interne XML-Editor in ADIF bietet verschiedene Hilfsfunktionen für die Erstellung eines ADIF-Schemas:

- Hilfe für die Dateneingabe:

Wenn Sie ein "<" im Editor schreiben, wird eine Liste der Elemente gezeigt, die an dieser Position des XML-Codes zulässig sind. Sie können auf einen Elementnamen in der Liste doppelklicken, um ihn zum Code hinzuzufügen. Wenn Sie nach einem Elementnamen ein Leerzeichen hinzufügen, wird eine Liste der für das Element zulässigen Attribute angezeigt. Sie können ein Attribut hinzufügen, indem Sie es in der Liste doppelklicken.



- Hierarchie-Informationen für Elemente und Attribute:

Wenn Sie den Cursor über einen Elementnamen bewegen, wird die übergeordnete Hierarchie für das Element angezeigt. Wenn Sie den Cursor über einen Attributnamen bewegen, wird der Name des Elements angezeigt, für das das Attribut definiert wurde.

```

1  <ADIF_ImportScheme Name="Import
2  <AlfaTechInfo TechCreationDat
3  <ImportEntry Name="APPLICATIO
4  <Attribute ImportColumn="ID
5  <Attribute ImportColumn="Na
6  <Attribute ImportColumn="Ve
7  <Element Attribute
8  <
9  <
10 <
11 <Attribute ImportColumn="Li

```

Element Attribute

Parent element hierarchy:
ADIF_ImportScheme / ImportEntry

- Fehlermeldungen bei falschen oder unvollständigen Elementen:

Unterhalb des Bearbeitungsbereichs werden Fehlermeldungen angezeigt, wenn Sie ein Element hinzufügen, das im XML-Schema oder an der Position, an der Sie es hinzugefügt haben, nicht zulässig ist oder wenn Attribute oder ein Attributwert nicht der Schemaspezifikation entsprechen. Die Fehlermeldungen bieten Informationen zur Zeilen- und Spaltenposition des Fehlers und zur Art des aufgetretenen Fehlers. Die Fehlermeldungen sind nur in Englisch verfügbar.

```

18  <SqlText><![CDATA[UPDATE TMP_APPLICATION
19  </SqlCommand>
20  <SqlCommand test="1" Name="bind Responsi
21  <SqlText><![CDATA[UPDATE TMP_APPLICATION

```

Line	Col	Message
20	17	The 'test' attribute is not declared.

Externe Editoren

Sie können ADIF-Schemata in jedem Texteditor erstellen und bearbeiten.

Die bearbeiteten ADIF-Schemata können entweder über die ADIF-Konfigurationsbenutzeroberfläche in die Datenbank hochgeladen oder über das lokale Dateisystem bereitgestellt werden. Die ADIF-Konsolenanwendung, die zum Start der Import- oder Exportprozesse verwendet wird, arbeitet sowohl mit einem über XML-Datei bereitgestellten ADIF-Schema als auch mit einem ADIF-Schema, das Teil der Ziel-/Quelldatenbank ist.

Sie können ein einzelnes ADIF-Schema, das über die ADIF-Konfigurationsbenutzeroberfläche konfiguriert wurde, in eine externe XML-Datei exportieren, um es in einem externen XML-Editor zu bearbeiten.

So speichern Sie ADIF-Schemata in einer XML-Datei:

- 1) Klicken Sie im ADIF-Explorer auf der rechten Seite auf das ADIF-Schema, das Sie exportieren möchten.
- 2) Wählen Sie im Kontextmenü die Option **Speichern unter**.
- 3) Geben Sie im daraufhin angezeigten Explorer einen Namen für die XML-Datei ein, und speichern Sie sie am gewünschten Speicherort.

Vollständige ADIF-Schemata, die in einer XML-Datei konfiguriert wurden, können folgendermaßen in die Datenbank hochgeladen werden:

- 1) Klicken Sie im ADIF-Explorer mit der rechten Maustaste auf den Root-Knoten **ADIF_Schemes**.

2) Wählen Sie im Kontextmenü eine der folgenden Optionen aus:

- **Aus Datei zusammenführen:** Neue Schemata werden der Datenbank hinzugefügt, und Schemata, die sowohl in der XML-Datei als auch in der Datenbank verfügbar sind, werden in der Datenbank mit dem ADIF-Schema in der Datei überschreiben. Schemata, die nur in der Datenbank verfügbar sind, werden nicht geändert. Während der Zusammenführungsaktion werden ADIF-Schemata nach ihrem Attribut **Name** zugeordnet.
- **Aus Datei ersetzen:** Alle ADIF-Schemata in der Datenbank werden gelöscht, und die ADIF-Schemata in der XML-Datei werden dann in die Datenbank hochgeladen.

Kapitel 4: Konfigurieren von Datenimporten mit ADIF

Dieses Kapitel behandelt den Prozess für den Import von Daten in die Alfabet-Datenbank über die ADIF-Schnittstelle.

Datenverarbeitung während des Imports

Der Datenimport kann Daten aus Datenbanktabellen, Microsoft® Excel®-Dateien, kommagetrennten Datenformatdateien (.csv oder .txt), JSON- und XML-Dateien enthalten. Alle im Import enthaltenen Dateien müssen in einer ZIP-Datei bereitgestellt werden. Der Import kann als einzelner Importvorgang durchgeführt werden, der auf mehreren Importquellen basiert.



ZIP-Dateien werden beim Hochladen auf die Größe des Inhalts nach der Dekomprimierung überprüft. Die Datei wird nicht hochgeladen, wenn die dekomprimierte Größe mehr als 100 % der komprimierten Größe beträgt oder wenn das Speichern des Dateiinhalts auf dem lokalen Laufwerk zu weniger als 1 GB freiem Speicherplatz führen würde oder wenn Abweichungen von den normalen Komprimierungsmechanismen erkannt werden.

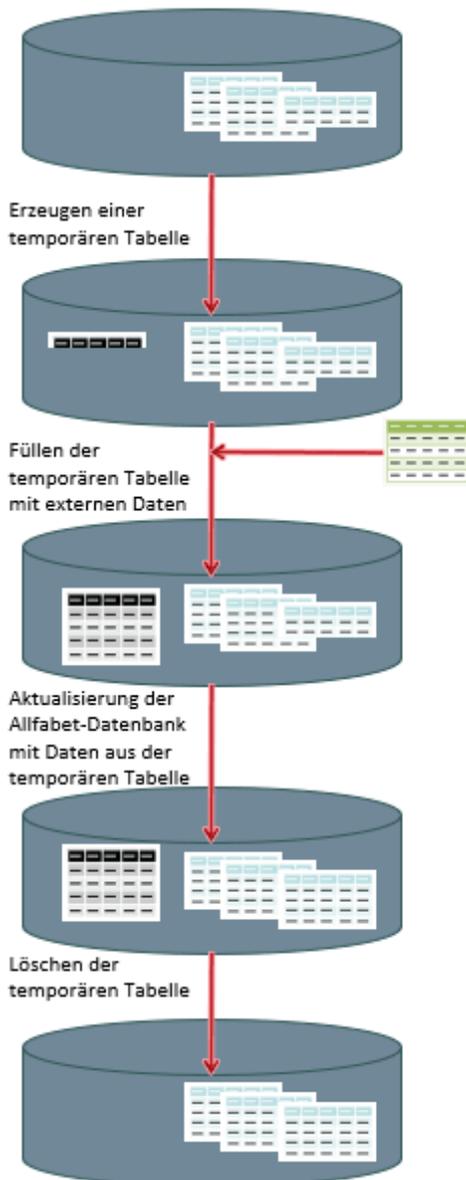
Während des Imports müssen die externen Daten der Alfabet-Datenbank zugeordnet werden:

- Die externe Datentabelle muss der korrekten Datenbanktabelle in der Alfabet-Datenbank zugeordnet werden.
- Die externen Datenwerte müssen den korrekten Spalten in der Alfabet-Datenbank -Tabelle zugeordnet werden.
- Die Datensätze in der externen Datentabelle müssen den korrekten Zeilen in der bestehenden Alfabet-Datenbank -Tabelle zugeordnet werden. In den meisten Fällen enthält die Alfabet-Datenbank bereits Daten. In diesem Fall sollten die externen Daten nicht einfach der Alfabet-Datenbank -Tabelle hinzugefügt werden. Objektdaten sollten aktualisiert werden, wenn das Objekt bereits existiert, und neue Einträge sollten nur für neue Objekte in die Datenbank geschrieben werden. Es kann sogar nötig sein, während des Imports existierende Daten zu löschen.

Der ADIF-Importmechanismus zielt auf die Struktur und Eindeutigkeitseinschränkungen der Alfabet-Datenbank ab. Die Informationen zu den externen Daten müssen durch den Kunden spezifiziert werden, damit der ADIF-Importmechanismus die Daten zuordnen kann. Kunden können den Import per SQL konfigurieren.

Die Struktur der externen Daten kann sich erheblich von der Struktur der Alfabet-Datenbank -Tabellen unterscheiden. Um den Importprozess so flexibel wie möglich zu halten, führt der ADIF-Importmechanismus den Import in zwei Schritten durch:

- Schritt 1: Aufgrund der Daten in der Eingabedatei/der externen Datenbanktabelle werden temporäre Tabellen erstellt. Die temporäre Datenbanktabelle kann mithilfe von SQL-Befehlen geändert und vervollständigt werden, bevor mit dem zweiten Importschritt begonnen wird.
- Schritt 2: Daten aus den temporären Tabellen werden in die Alfabet-Datenbank -Tabellen für Objektklassen im Alfabet -Metamodell geschrieben.



Dieser Mechanismus gewährt ein hohes Maß an Flexibilität. Sie können aus Ihren externen Daten mehrere temporäre Tabellen mit einer unterschiedlichen Struktur und unterschiedlichen Inhalten definieren (und sogar temporäre Tabellen aus dem Inhalt bereits generierter temporärer Tabellen definieren), um die Daten gemäß Ihrer Bedürfnisse umzustrukturieren. Die Alfabet-Datenbank -Tabellen werden dann auf der Basis des Inhalts der temporären Tabellen geändert und nicht auf der Basis des Inhalts der ursprünglichen externen Daten.

Nachdem der Import abgeschlossen ist, werden die temporären Tabellen aus der Alfabet-Datenbank gelöscht.

Ausführungsschritte des Importvorgangs

Der Import wird von den ADIF-Importschemata ausgelöst, die mithilfe von SQL-Befehlen vom Kunden konfiguriert wurden. Mit einem ADIF-Importschema kann der Import aus mehreren Datenbanktabellen oder -dateien ausgelöst werden, die in einem oder mehreren Dateiformaten vorliegen.

Der Importvorgang wird mit dem ADIF-Debugger oder der ADIF-Konsolenanwendung gestartet.

Folgender Prozess wird ausgeführt, wenn der ADIF-Import gestartet wird:

- 1) Wenn die Daten aus einer oder mehreren Dateien importiert werden, werden die Importdateien (die sich in einer ZIP-Datei befinden müssen) in die Alfabet-Datenbank hochgeladen.
- 2) Das ADIF-Importschema wird überprüft. Der Import wird beendet, wenn die Überprüfung fehlschlägt.
- 3) Auf der Serverseite wird ein temporäres Verzeichnis für Protokoll- und Importdateien erstellt.
- 4) Die ZIP-Datei mit Importdateien wird in das temporäre Verzeichnis hochgeladen und entpackt.
- 5) Befehlszeilenargumente werden in die Alfabet-Datenbank geschrieben. Wenn der Import durch ein ADIF-Importschema in einer XML-Datei ausgelöst wird, wird das ADIF-Importschema auch in die Datenbank geschrieben.
- 6) Temporäre Tabellen, die in allen Importspezifikationen des ADIF-Importschemas angegeben sind, werden in die Datenbank geschrieben. Dieses Kapitel enthält Einzelheiten dazu.
- 7) Die Alfabet-Datenbank -Tabellen werden mit Daten aus den temporären Datenbanktabellen aktualisiert, wie in den Importspezifikationen im ADIF-Importschema definiert. Details finden sich in diesem Kapitel.
- 8) Alle angegebenen Workflows werden gestartet.
- 9) Protokollinformationen zum Importprozess werden in den Ordner `ADIF_SYS` im **Dokument-Explorer** in der Alfabet-Datenbank hochgeladen.
- 10) Das temporäre Verzeichnis auf der Serverseite (das vor dem Ausführen des Imports erzeugt wurde) und sämtlicher Inhalt des temporären Verzeichnisses werden gelöscht.
- 11) Falls im ADIF-Importschema nicht anders festgelegt, werden alle während des Imports erzeugten temporären Tabellen entfernt.
- 12) Wenn der Import über die ADIF-Konsolenanwendung gestartet wird, wird der Inhalt der Protokolldatei in die Protokolldatei der Konsole heruntergeladen.

Im folgenden Abschnitt wird beschrieben, wie unterschiedliche Importfunktionen innerhalb des ADIF-Importschemas konfiguriert werden. Details zu den Prozessen, die zum Durchführen der oben beschriebenen Importschritte ausgeführt werden, sowie die Ausführungsreihenfolge unterschiedlicher Elemente des ADIF-Schemas, die während des Imports verarbeitet werden, werden zusammen mit einer Beschreibung der verschiedenen Importschritte bereitgestellt.

Vorbereiten von Daten für den Import

Daten können in den folgenden Datenformaten importiert werden:

- XML-Dateien
- JSON-Dateien
- Microsoft® Excel-Dateien (.xls und .xlsx)
- CSV-Dateien
- externe Datenbanktabellen

ADIF ist bezüglich der Übersetzung externer Datenstrukturen, damit diese in das Alfabet -Metamodell passen, sehr flexibel. Nichtsdestotrotz müssen die externen Daten, die für den Import verwendet werden, den folgenden Anforderungen entsprechen, damit die Daten während des Importvorgangs richtig gelesen werden können.

In den folgenden Abschnitten sind die spezifischen Anforderungen für die einzelnen Importdateiformate aufgelistet.

Microsoft® Excel®-Dateien

Sowohl XLS- als auch XLSX-Dateien können importiert werden. Die folgenden Regeln gelten für beide Dateiformate.

- Alle Daten müssen in dem ersten Arbeitsblatt der Microsoft® Excel®-Datei enthalten sein. Alle weiteren Arbeitsblätter werden während des Imports ignoriert.
- Wenn die Daten eine Kopfzeile enthalten, dürfen die Kopfzellen nur alphanumerische Daten ohne Sonderzeichen enthalten.
- Die Tabelle muss eine flache Struktur aufweisen.
- Die zulässigen Datenformate sind Zahlen, Zeichenfolgen und Daten. Alle Sonderzeichen, einschließlich NULL-Werte (Hexadezimalwert 0x00) können nicht importiert werden.
- Die Tabelle darf keine Formeln enthalten.

XML-Dateien

- Die Datei muss gültige XML-Definitionen enthalten.
- Folgende Codierungen sind zugelassen:
 - ASCII
 - UTF7
 - UTF8
 - UTF32
 - Unicode
- Textinhalte von XML-Elementen können nur vom Importmechanismus interpretiert werden, wenn das XML-Element, das den Text enthält, keine eigenen Attribute aufweist und ein untergeordnetes Element eines XML-Elements mit Attributen ist.
- Wenn die XML-Datei auf eine DTD-Datei verweist, muss die DTD-Datei in der zu importierenden ZIP-Datei enthalten sein, wenn die XML-Datei importiert wird.
- Zeichenfolgen, die Zeilenumbrüche enthalten, müssen in CDATA-Elemente geschrieben werden, oder die Zeilenumbrüche müssen gemäß XML-Regeln in Escape-Zeichen gesetzt werden.
- Die Daten müssen XML-konform sein, um importiert werden zu können. Alle Zeichen, die nicht XML-konform sind, werden beim Import automatisch durch Leerzeichen ersetzt.

JSON-Dateien

- Die Dateierweiterung muss ".json" lauten.

Kommagetrennte Formate (*.csv)

Sowohl CSV- als auch TXT-Dateien können importiert werden. Die folgenden Regeln gelten für beide Dateiformate.

- Die Daten müssen in einer Tabelle repräsentiert und jede Zeile muss in eine neue Zeile geschrieben sein. Jede Spalte in einer Zeile muss durch ein Spaltentrennzeichen separiert sein. Jeder Datensatz (jede Zeile) muss über dieselbe Anzahl von Spalten verfügen.
- Folgende Codierungen sind zugelassen:
 - ASCII
 - UTF7
 - UTF8
 - UTF32
 - Unicode
- Die Dateigröße darf die maximale Anzahl von Datensätzen, die mit Microsoft® Excel® in einem einzigen Datenblatt verarbeitet werden können, nicht überschreiten. Die gleiche Komponente wird für den Import von Daten aus Microsoft Excel- und CSV-Dateien verwendet, und diese Komponente hat eine integrierte Verarbeitungsgrenze, die den für Microsoft Excel definierten Grenzen entspricht.

Externe Datenbanken

Aus folgenden Arten von Datenbanken kann importiert werden:

- Microsoft® SQL
- Oracle®
- Microsoft® Access
- PostgreSQL 9.1
- LDAP-Tabellen

Sie müssen gewährleisten, dass die Zugriffsberechtigungen der externen Datenbank zulassen, dass die Daten vom ADIF-Prozess gelesen werden.

Der Konfiguration muss eine Verbindungszeichenfolge für die Verbindung mit der Datenbank hinzugefügt werden. Dies sollte vor der Konfiguration geschehen.

Konzipieren des Datenimports

Der erste und wichtigste Schritt für den Datenimport über ADIF ist das Konzipieren des Datenimports. Sie sollten sich Folgendes vor Augen führen, bevor Sie das ADIF-Importschema konfigurieren:

- Welche Daten sollen importiert werden?
- Wie können Daten dem Alfabet -Metamodell zugeordnet werden?
- Ist eine Konfiguration des Alfabet -Metamodells erforderlich? Werden beispielsweise benutzerdefinierte Eigenschaften für Objektklassen benötigt, da Daten, die importiert werden sollen, nicht durch die Standardeigenschaften des Alfabet -Metamodells repräsentiert werden?
- Welche Datenformate haben die Importdaten und stimmen sie mit den Datenformaten der Alfabet-Datenbank -Tabellen überein?
- Sind alle Pflichteigenschaften für die Alfabet -Objektklasse im Import neuer Datensätze enthalten?
- Enthalten meine Daten eindeutige Attribute, die es ermöglichen, dass die Daten existierenden Alfabet-Datenbank -Tabellen zugeordnet werden?
- Entsprechen meine Datenimportformate den im Abschnitt *Vorbereiten von Daten für den Import* beschriebenen Regeln?
- Falls die Importdaten Beziehungen zu anderen Objekten enthalten, müssen die Datenbanktabellen dann vor dem Import für die zugehörigen Objektklassen aktualisiert werden?
- Welche Zwischenschritte, die temporäre Tabellen nutzen, sind erforderlich, um die Daten erfolgreich in die Alfabet-Datenbank zu importieren?

Es wird empfohlen, jener Person, die das ADIF-Schema konfiguriert, ein wohldurchdachtes Konzept in schriftlicher Form bereitzustellen.

Konfigurieren des ADIF-Importschemas

Bei einem ADIF-Importschema handelt es sich um ein XML-Objekt, das entweder in einem XML-Editor oder in der ADIF-Konfigurationsbenutzeroberfläche, die als Teil des Konfigurationstools Alfabet Expand bereitgestellt wird, bearbeitet werden kann. In der Benutzeroberfläche werden die XML-Elemente als Knoten im Explorer angezeigt, während Attribute eines XML-Elements im Attributfenster auf der rechten Seite des Explorers angezeigt werden.

Alle Informationen und Befehle, die zum Importieren von Daten aus der Alfabet-Datenbank erforderlich sind, werden mithilfe der Elemente des ADIF-Importschemas definiert. Die Elemente müssen einer definierten Sequenz folgen, die von einem XSD-Schema vorgegeben wird. Im ADIF-Explorer geleiten Sie die Kontextmenüoptionen, die für die Elemente des Importschemas verfügbar sind, durch die Erstellung aller erforderlichen Elemente. Im Kontextmenü werden z. B. nur die Optionen zum Erzeugen zulässiger untergeordneter Elemente für ein Importschemaelement angezeigt. Wenn Sie das ADIF-Importschema in einem XML-Editor definieren, müssen Sie im XSD-Schema nach Informationen zur zulässigen Reihenfolge von Objekten suchen.

Die folgenden Elemente sind Bestandteil des ADIF-Importschemas:

Name des XML-Elements	Titel in der ADIF-Benutzeroberfläche	Zweck
ADIF_ImportScheme	Importschema	Element für die Konfiguration grundlegender und umfassender Importausführungsparameter.
Parameter	Parameter	<p>Eine Parameterdefinition, die in allen Abfragen innerhalb des ADIF-Importschemas als Variable verwendet werden kann, die mit einem Wert gefüllt ist, der beim Ausführen des ADIF-Imports definiert wurde.</p> <p>Dieses Element ist in ADIF-Importschemata, die vor Alfabet 10.4 definiert wurden, nicht sichtbar, da das Attribut Parameter - Abwärtskompatibilitätsmodus auf <code>True</code> gesetzt ist.</p>
XmlImportSet	XML-Importsatz/JSON-Importsatz	<p>Element, mit dem eine JSON- oder XML-Dateistruktur für den Datenimport aus XML-Dateien definiert wird.</p> <p> Für den Import aus JSON-Dateien ist die Option zum Erzeugen eines JSON-Importsatzes nur aus Gründen der Abwärtskompatibilität verfügbar. Für den Import aus JSON sollte stattdessen ein Eintrag für das hierarchische JSON erzeugt werden.</p>
FileImportSet	Datei-Importgruppe	Strukturierendes Element, das als Container für Elemente dient, mit denen der Import aus Microsoft® Excel®- oder CSV-Dateien definiert wird.
DBImportSet	DB-Importgruppe	Element, mit dem die Verbindung zu einer externen Datenbank definiert wird.
ADImportSet	LDAP-Importgruppe	Element, mit dem die Verbindung zu einer externen LDAP-Tabelle definiert wird.
ImportEntry	Eingabe	Element defining the import from one external database table or file to the Alfabet-Datenbank.
Attribute	Attribut	Unterelement von <code>ImportEntry</code> -Elementen für die Definition der Zuordnung von externen Daten zu einer Datenbankspalte in der Alfabet-Datenbank.
Relation	Beziehung	Den <code>ImportEntry</code> -Elementen untergeordnetes Element für die Definition einer Beziehung zwischen Objekten, die während des Imports erzeugt werden sollen.

Name des XML-Elements	Titel in der ADIF-Benutzeroberfläche	Zweck
Index	Index	Den <code>ImportEntry</code> -Elementen untergeordnetes Element für die Definition eines Index, der während des Imports erzeugt werden soll.
SQLCommand	SQL-Befehl	Den <code>ImportEntry</code> -Elementen untergeordnetes Element für die Definition von SQL-Befehlen, die während des Imports ausgeführt werden sollen.
AlfaMethod		Unterelement von <code>ImportEntry</code> -Elementen, um dem Eintrag benutzerdefinierten Code zuzuordnen. <code>AlfaMethod</code> -Elemente sind nur relevant, wenn benutzerdefinierter Code für spezielle Kundenanforderungen von Software AG entwickelt wurde. Informationen zur Konfiguration werden individuell mit dem benutzerdefinierten Code bereitgestellt.

Erstellen eines ADIF-Importschema

Im Folgenden wird das Erstellen und Bearbeiten eines Importschemas mithilfe der ADIF-Konfigurationsbenutzeroberfläche beschrieben. Beachten Sie, dass nur jener Satz von Attributen erläutert wird, der für die hier beschriebene Aufgabe erforderlich ist.

So erstellen Sie ein neues ADIF-Importschema:

- 1) Klicken Sie im Explorer mit der rechten Maustaste auf den Root-Knoten **ADIF Schemes**, und wählen Sie **Importschema erstellen** aus. Das neue Importschema wird dem Explorer hinzugefügt. Das Attributfenster des neuen Importschemas wird rechts angezeigt.
- 2) Legen Sie im Attributfenster die folgenden Attribute für das ADIF-Importschema fest:
 - **Name:** Geben Sie einen eindeutigen Namen ein. Der Name dient der Identifikation des ADIF-Importschemas in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.
 - **Titel:** Geben Sie einen aussagekräftigen und eindeutigen Titel ein. Der Titel wird verwendet, um das ADIF-Importschema in der Alfabet -Benutzeroberfläche der Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** zu identifizieren.
 - **Beschreibung:** Geben Sie eine aussagekräftige und kurze Beschreibung des Ergebnisses des ADIF-Imports ein. Die Beschreibung wird in der Alfabet Benutzeroberfläche der Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** im Vorschauenfenster eines ADIF-Schemas angezeigt.
 - **Änderungen beibehalten:** Ist diese Option auf `True` gesetzt, wird das Ergebnis des Datenimports dauerhaft in die Alfabet-Datenbank geschrieben. Ist diese Option auf `False` gesetzt, wird der Importvorgang nach der Ausführung zurückgenommen, und es werden keine Änderungen in die Datenbank geschrieben. Die Konfiguration des automatischen Starts von Workflows während des Imports wird ignoriert, wenn **Änderungen beibehalten** auf `False`

gesetzt ist. Es wird empfohlen, **Änderungen beibehalten** für ein neues Importschema auf `False` zu setzen, um das Debuggen ohne Risiko der Beschädigung der Datenbank zuzulassen. Nach der erfolgreichen Prüfung des Datenimports und der Überprüfung, dass die resultierenden Änderungen an der Alfabet-Datenbank wie erwartet ausgefallen sind, können Sie das Attribut **Änderungen beibehalten** auf `True` zurücksetzen, um reguläre Datenimporte durchzuführen.



Beachten Sie Folgendes:

- Wenn das Attribut **Änderungen beibehalten** festgelegt wird, werden alle Änderungen an Datensätzen in vorhandenen Tabellen, die durch DML-Anweisungen verursacht wurden, zurückgenommen. Das Erstellen oder Löschen von Tabellen ist nicht im Rollback enthalten. Wenn Sie zum Beispiel ein ADIF-Schema testen, das darauf konfiguriert ist, temporäre Tabellen dauerhaft in die Datenbank zu schreiben, werden diese temporären Tabellen dauerhaft erzeugt, selbst wenn **Änderungen beibehalten** auf `False` gesetzt ist. SQL-Befehle des Typs **OnActivate** sind ebenfalls vom Rollback ausgeschlossen.
- Wenn während eines Importauftrags neue Objekte erzeugt werden, weist der Datenbindungsmechanismus `REFSTR`-Werte für die neuen Objekte zu. Wenn **Änderungen beibehalten** auf `False` gesetzt ist, werden die Objekte nicht in der Datenbank erstellt, die `REFSTR`-Werte werden jedoch trotzdem als verwendet betrachtet und bei der nächsten ADIF-Ausführung nicht zur Datenbindung verwendet, wenn nicht der Alfabet-Server oder die Alfabet Expand -Applikation, die zur Verarbeitung des ADIF-Auftrags verwendet wird, neu gestartet wird.
- Von `OnActivate`-Befehlen ausgelöste Änderungen werden nicht zurückgenommen, wenn die Option **Änderungen beibehalten** im Importschema auf `False` gesetzt ist.
- **Temporäre Tabellen entfernen:** Ist dieses Attribut auf `True` gesetzt, werden alle temporären Tabellen nach dem Import entfernt. Nur die Änderungen an der Alfabet-Datenbank werden dauerhaft gespeichert. Ist dieses Attribut auf `False` gesetzt, bleiben die temporären Tabellen nach dem Import in der Datenbank erhalten. Die persistente Speicherung temporärer Tabellen ist nur für besondere Import-/Exportzyklen erforderlich, die für die Datenmanipulation entwickelt wurden und Eingaben von den temporären Tabellen eines zuvor festgelegten Imports benötigen. In den meisten Fällen wird empfohlen, dieses Attribut auf `True` zu setzen, um die Datenbank von Daten zu bereinigen, die nicht Bestandteil des Alfabet -Metamodells sind.
- **Nachträgliche Aktion in Datei:** Mit diesem Attribut wird die Verarbeitung von Importdateien nach dem Import definiert. Wählen Sie eine der folgenden Optionen aus:
 - `None`: Importdateien werden nicht geändert.
 - `Delete`: Importdateien werden nach dem Import gelöscht. Falls der Import auf einer ZIP-Datei basiert, wird die ZIP-Datei gelöscht. Werden Dateien aus einem Verzeichnis gelesen, werden alle Dateien aus dem angegebenen Importverzeichnis gelöscht.
 - `RenameUsingTimeStamp`: Importdateien werden nach dem Import umbenannt. Falls der Import auf einer ZIP-Datei basiert, wird die ZIP-Datei umbenannt. Werden Dateien beim Importieren aus einem Verzeichnis gelesen, werden alle Dateien aus dem angegebenen Importverzeichnis umbenannt. Umbenannten Dateien wird ein Zeitstempel zum

Dateinamen hinzugefügt, und alle Buchstaben des Dateinamens werden in Großbuchstaben geändert.



Der Parameter `RenameUsingTimeStamp` wird während des Debuggens ignoriert.

- **Import-Datei erforderlich:** Wählen Sie `True` aus, wenn Ihr Importschema Konfigurationen für den Datenimport aus Dateien enthält. Legen Sie diesen Parameter auf `False` fest, um nur Daten aus externen Datenbanken oder Active Directories zu importieren, oder wenn Sie einen Import basierend auf einem Assistenten definieren. Im Rahmen der Alfabet - Integrationslösungen stehen Assistenten zum Vordefinieren des Datenimports aus externen Services zur Verfügung. Informationen hierzu finden Sie unter [Vordefinierte ADIF-Schemata](#).
- **Blockverarbeitung:** Wenn ein neues Objekt über den ADIF-Import erzeugt wird und ein Standardwert für die Eigenschaften der Objektklasse definiert ist, prüft der Importmechanismus, ob der Import einen Wert für die Eigenschaft enthält. Falls kein Wert angegeben ist, wird der Standardwert festgelegt. Um die Leistung und die Verarbeitungszeit von ADIF-Importprozessen für sehr große Datensätze (über 1 Million Datensätze) im Batch zu verbessern, kann das Attribut **Blockverarbeitung** des ADIF-Schemas auf `True` eingestellt werden. Der Mechanismus der Blockverarbeitung scannt alle Objektklasseneigenschaften, für die in der temporären Tabelle des Imports keine Spalte definiert ist. Wenn für die Objektklasseneigenschaft ein Standardwert definiert ist, wird der temporären Tabelle für diese Objektklasseneigenschaft eine zusätzliche Spalte hinzugefügt, die den Standardwert für jede Zeile zurückgibt. Dadurch muss beim Importmechanismus der jeweilige Standardwert für jedes Objekt nicht separat festgelegt werden. Die **Blockverarbeitung** verbessert die Leistung für das erneute Abfragen von Rückbeziehungen für Importe, die den Import von Beziehungen mit entsprechenden Rückbeziehungen in den Zielobjekten umfassen. Die **Blockverarbeitung** sollte nur aktiviert werden, wenn der Importdatensatz sehr groß ist und für eine Reihe von Objektklasseneigenschaften, für die über die importierten Daten kein Wert angegeben ist, ein Standardwert definiert wurde.
- **Automatisch ausführen** Wenn dies auf `True` festgelegt ist, wird das ADIF-Schema automatisch nach jeder Aktualisierung des Alfabet -Metamodells über eine *.amm-Datei oder jeder Wiederherstellung der Datenbank mit einer *.adb-Datei ausgeführt.



Beachten Sie Folgendes:

- Wenn das Attribut **Auto Run** auf `True` gesetzt ist, muss das Attribut **Import-Datei erforderlich** auf `False` gesetzt sein.
- Wenn ADIF-Importschemata automatisch ausgeführt werden, werden keine Protokolldateien generiert. Das ADIF-Schema muss während der Konfiguration getestet werden, um zu gewährleisten, dass es zur Laufzeit ordnungsgemäß ausgeführt wird.
- Die Datenbankverbindung wird vor einer automatischen Ausführung geschlossen und wieder geöffnet. Dies kann einige Zeit in Anspruch nehmen.



Weitere Informationen über das Aktualisieren des Metamodells und das Wiederherstellen eines Datenbankarchivs finden Sie im Referenzhandbuch *Systemadministration*.

- **Verhalten der Alfabet-Benutzeroberfläche:** Wählen Sie eine der folgenden Optionen aus, um die Verfügbarkeit des ADIF-Schemas in der Alfabet -Benutzeroberfläche zu definieren:
 - `VisibleExecutable`: Der ADIF-Import kann in der **ADIF-Prozessverwaltung** ausgelöst werden, und der Erfolg der ADIF-Prozesse, die mit diesem ADIF-Schema ausgeführt wurden, kann über die Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** überprüft werden.
 - `VisibleNotExecutable`: The success of the ADIF jobs executed with this ADIF scheme can be controlled via the **ADIF Job Administration** and **My ADIF Jobs** functionalities, but the ADIF import cannot be triggered via the Alfabet user interface.
 - `NotVisible`: Das ADIF-Schema und die Informationen zu ADIF-Prozessen, die mit diesem ADIF-Schema ausgeführt wurden, sind in den Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** nicht sichtbar. Dies ist der Standardwert für neue ADIF-Importschemata.



Informationen zur Verwaltung und Ausführung des ADIF-Schemas über die Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** finden Sie unter [Ausführen und Überprüfen von ADIF über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" in der Alfabet -Benutzeroberfläche](#).

- **Für REST-API verwendbar:** Setzen Sie das Attribut auf **True**, wenn das ADIF-Schema über einen RESTful-Serviceaufruf an den Endpunkt `adifimport` der RESTful API von Alfabet ausgeführt werden soll, entweder über einen RESTful-Serviceaufruf von einem externen RESTful-Client oder über ein Alfabet -Ereignis, das die Ausführung des ADIF-Schemas auslöst, wenn ein Anwender einen Assistenten oder Workflow-Schritt aufruft oder verlässt oder wenn ein Ereignis zur Ausführung eines RESTful-Serviceaufrufs an eine Drittanbieterapplikation abgeschlossen wird.



Die Ausführung des ADIF-Schemas über die RESTful API von Alfabet erfordert die Einrichtung der RESTful-Services. Informationen über die Anforderungen und die Ausführung des Serviceaufrufs finden Sie im Referenzhandbuch *Alfabet RESTful API*.

Informationen zur zusätzlichen Konfiguration zur Implementierung der Ausführung eines Ereignisses finden Sie unter *Konfigurieren von Ereignissen*.

- **Gruppe:** Mithilfe der ADIF-Konfigurationsbenutzeroberfläche können Sie ADIF-Schemata in Gruppenordnerknoten strukturieren. Diese Elemente sind nicht Teil der ADIF-XML-Spezifikation, sondern leere Explorer-Knoten, die zur visuellen Strukturierung der ADIF-Schemadefinitionen im Explorer verwendet werden. Geben Sie einen Namen für einen vorhandenen Gruppenordner ein, um das ADIF-Schema zur Gruppe hinzuzufügen, oder geben Sie einen neuen Namen ein, um einen neuen Gruppenordnerknoten zu erstellen und das ADIF-Schema der neuen Gruppe hinzuzufügen.



Nach dem Erstellen des ADIF-Importschemas müssen Sie Importdefinitionen hinzufügen, um den Import aus einem oder mehreren Importdatenformaten in das Importschema auszulösen. Fahren Sie mit dem Abschnitt *Konfigurieren des Imports aus verschiedenen externen Quellformaten* fort, um das Importschema zu konfigurieren.

Optional können Sie Folgendes konfigurieren:

- *Konfigurieren der Ausführung des Importschemas in Abhängigkeit von aktuellen Parametern*

- *Konfigurieren von SQL-Befehlen für optionale erweiterte Importfunktionen*
- *Konfigurieren der Importhistorie*
- *Konfigurieren des automatischen Starts von Workflows während des Imports*
- *Konfigurieren von Protokollierungsparametern*

Konfigurieren des Imports aus verschiedenen externen Quellformaten

Der Import von Daten in eine temporäre Tabelle vor der Aktualisierung der Alfabet-Datenbank -Tabellen teilt die Konfigurationsprobleme in zwei Teile: die Konfiguration, um Daten in die temporären Tabellen zu importieren (was von den unterschiedlichen Importformaten abhängt), und die Definition der Zuordnungsbedingungen, um Daten in den Alfabet-Datenbank -Tabellen zu aktualisieren (was für alle Importformate gleich ist).

Für jeden Import müssen Sie Folgendes angeben:

ADIF-Schemaelement	Für den Import aus Datenbanktabellen	Für den Import aus Dateien (XLS, XLSX, CSV, TXT)	Für den Import aus XML	Für den Import aus JSON
Importgruppe  /  /  / 	Die Verbindung zur externen Datenbank muss in einer Datenbank-Importgruppe  oder einer LDAP-Importgruppe  definiert werden. Die Importgruppe ist allen anderen Konfigurationselementen für den Import aus dieser Datenbank übergeordnet.	Die Spezifikation einer Importgruppe ist nicht erforderlich. Dennoch können Sie Datei-Importgruppen  als Strukturierungselemente definieren, die beispielsweise alle wichtigen Definitionen für den Import aus demselben Dateiformat oder für den Import im Hinblick auf dieselbe Aufgabe enthalten.	Für jede externe XML-Datei muss dem Importschema eine XML-Importgruppe  hinzugefügt werden. Die Importgruppe ist allen Importdefinitionen für XML-Elemente in der Datei übergeordnet.	Die Angabe einer Importgruppe ist nicht erforderlich. Aus Gründen der Abwärtskompatibilität ist es weiterhin möglich, einen Importsatz für den JSON-Import zu erzeugen. Dieser Import hat nicht alle Anwendungsfälle abgedeckt und wurde daher durch einen direkten JSON-Import aus einem Importeintrag ersetzt.
Importeintrag 	Für jede externe Datenbanktabelle, aus der Daten importiert werden sollen, muss der Datenbank-Importgruppe	Für jede Datei, aus der Daten importiert werden sollen, muss einer Datei-Importgruppe oder dem ADIF-Importschema direkt wenigstens ein Importeintrag hinzugefügt werden.	Für jeden XML-Elementtyp in der XML-Datei, in die Daten importiert werden sollen, muss der XML-Importgruppe wenigstens ein Importeintrag hinzugefügt werden.	Mindestens ein Importeintrag muss dem ADIF-Importschema über die Kontextmenüoption Eintrag für hierarchische JSON erzeugen für jede Datei hinzugefügt werden, aus der Daten importiert werden sollen. Der daraus resultierende

ADIF-Schemaelement	Für den Import aus Datenbanktabellen	Für den Import aus Dateien (XLS, XLSX, CSV, TXT)	Für den Import aus XML	Für den Import aus JSON
	<p>wenigstens ein Importeintrag hinzugefügt werden.</p> <p>Mit einem Importeintrag wird der Import zwischen nur einer externen Datenbanktabelle und einer Alfabet-Datenbank -Tabelle festgelegt. Wenn Sie Daten aus einer externen Datenbanktabelle in mehrere Alfabet-Datenbank -Tabellen importieren möchten, müssen Sie mehrere Importeinträge angeben.</p>	<p>Mit einem Importeintrag wird der Import zwischen nur einer externen Datei und einer Alfabet-Datenbank -Tabelle festgelegt. Wenn Sie Daten aus einer externen Datei in mehrere Alfabet-Datenbank -Tabellen importieren möchten, müssen Sie mehrere Importeinträge angeben.</p>	<p>Mit einem Importeintrag wird der Import zwischen nur einem XML-Elementtyp und einer Alfabet-Datenbank -Tabelle festgelegt. Wenn Sie Daten aus einem XML-Elementtyp in mehrere Alfabet-Datenbank -Tabellen importieren möchten, müssen Sie mehrere Importeinträge angeben.</p>	<p>Importeintrag hat eine vordefinierte temporäre Tabellenstruktur.</p> <p>Mit einem Importeintrag wird der Import zwischen nur einer externen Datei und einer Alfabet-Datenbank -Tabelle festgelegt. Wenn Sie Daten aus einer externen Datei in mehrere Alfabet-Datenbank -Tabellen importieren möchten, müssen Sie mehrere Importeinträge angeben.</p>
<p>Attribut </p> <p>Beziehung </p> <p>Index </p> <p>Kennzahl </p> <p>Rolle </p> <p>SQL-Befehl </p>	<p>Die untergeordneten Elemente des Importeintrags ordnen den Inhalt der externen Datenbanktabelle, die Spalten der externen Microsoft® Excel®-Tabelle und die kommagetrennten Werte in der externen CSV-Datei zu. Es stehen verschiedene Typen untergeordneter Elemente zur Verfügung, um die Zuordnung der Dateien zu Alfabet-Datenbank -Spalten, die Erzeugung von Beziehungen zwischen Objekten und die Erzeugung eines Index zu erleichtern. Mechanismen, die über das ADIF-Schema oder die Spezifikation von SQL-Befehlen bereitgestellt werden, bieten Unterstützung bei der Zuordnung von Einträgen zu bereits existierenden Einträgen der Datenbanktabellen.</p>			



Um zu erfahren, wie externe Daten während des Imports interpretiert werden und wie die Importgruppe definiert werden muss, fahren Sie mit dem Abschnitt fort, der für Ihr Importformat relevant ist:

- *Definieren des Hochladens von Daten aus einer externen Datenbank in eine Datenbank-Importgruppe*
- *Definieren des Hochladens von Daten aus einer LDAP-Tabelle in eine LDAP-Importgruppe*
- *Definieren des Datenimports aus XML-Dateien in einer XML-Importgruppe*
- *Definieren des Datenimports aus Microsoft® Excel-Dateien und kommagetrennten Dateiformaten*

Definieren des Hochladens von Daten aus einer externen Datenbank in eine Datenbank-Importgruppe

Wenn Sie Daten aus einer externen Datenbank importieren möchten, muss eine Verbindung zu der externen Datenbank hergestellt werden, um die Daten abzurufen. Mit dem ADIF-Importvorgang werden die Daten nicht direkt aus den Datenbanktabellen der externen Datenbank gelesen. Die Daten müssen über eine benutzerdefinierte Abfrage hochgeladen werden, die in der externen Datenbank ausgeführt wird. Der aus der Abfrage resultierende Datensatz stellt die externen Daten dar, die mithilfe des ADIF-Importvorgangs importiert werden.

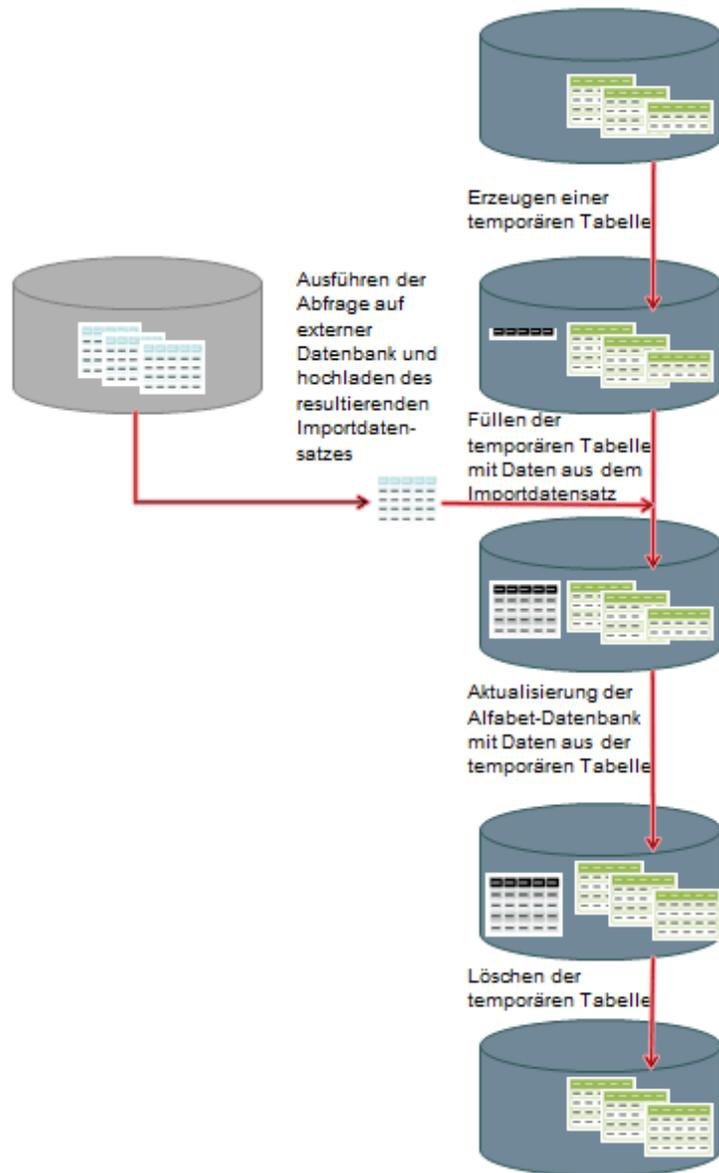
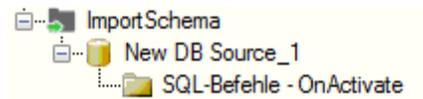


Abbildung: Importieren von Daten aus einer externen Datenbank

So konfigurieren Sie den Datenupload aus der externen Datenbank:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Importschema, und wählen Sie **DB-Importgruppe erzeugen** aus. Dem Explorer wird eine neue Datenbank-Importgruppe  mit einem Standardnamen für die Datenbank-Importgruppe hinzugefügt. Unter der Datenbank-Importgruppe  wird automatisch ein Ordner für die Erzeugung von SQL-Befehlen erzeugt, die bei Aktivierung der Datenbank-Importgruppe ausgeführt werden.



- 2) Geben Sie im Attributfenster der Datenbank-Importgruppe  die folgenden Attribute an, um die Verbindung zu der externen Datenbank zu konfigurieren:
 - **Name:** Geben Sie einen aussagekräftigen Namen für die Datenbank-Importgruppe ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit der Datenbank-Importgruppe implementiert wird.
 - **Treibertyp:** Der Import kann von Datenbanken auf Microsoft® SQL Servern®, Oracle®-Datenbankservern oder Microsoft® Access-Datenbanken durchgeführt werden. Wählen Sie die Art von Datenbankserver aus, auf dem sich die externe Datenbank befindet:
 - `SqlServer` für den Zugriff auf Microsoft SQL Server
 - `Oracle` für den Zugriff auf Oracle
 - `Access` für den Zugriff auf Microsoft Access über einen OLE-Treiber
 - `ODBC` für den Zugriff auf Microsoft Access, Oracle® MySQL oder PostgreSQL 9.1 über einen ODBC-Treiber
 - `Hadoop` für den Zugriff auf Hadoop-Systeme
 - **Verbindungsstring:** Geben Sie den Verbindungsstring ein, der für die Verbindung mit der externen Quelle erforderlich ist. Der Verbindungsstring hängt vom verwendeten Datenbankserver ab. Sie enthält in der Regel den Quellspeicherort, den Anwendernamen sowie weitere Parameter, die für die Anforderungen des Datenbankservers relevant sind.



Es wird empfohlen, das Datenbank-Login im Verbindungsstring über einen Benutzernamen und ein Kennwort zu definieren. Wenn die Windows-Authentifizierung zur Anmeldung verwendet wird und die ADIF-Konsolenapplikation mit einem Remote-Alias gestartet wird, wird nur dann erfolgreich eine Verbindung zur Datenbank hergestellt, wenn der Alfabet-Server mit demselben Domänen-Benutzernamen und denselben Zugriffsrechten gestartet wird wie im Verbindungsstring angegeben.



- Hinweis: In den Hilfe-Links können Server-Variablen genutzt werden. Mithilfe von Servervariablen können Sie die gesamte oder einen Teil der Definition der Server-Alias-Konfiguration des Alfabet-Servers definieren, anstatt diese direkt im ADIF-Schema zu definieren. Der Einsatz von Servervariablen ist vor allem dann sinnvoll, wenn Sie zum Beispiel die Konfiguration in einer Test- bzw. Produktionsumgebung mit unterschiedlichen externen Quellen verwenden. In beiden Umgebungen kann dieselbe Definition der externen Quelle verwendet werden. Mit der in der Test- und Produktionsumgebung verwendeten Definition des Serveralias werden die korrekten Verbindungsdaten für die externe Quelle definiert, die in der jeweiligen Umgebung verwendet wird. Die Nutzung von Servervariablen wird im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* im Kapitel *Berichte konfigurieren* im Abschnitt *Verwenden von Servervariablen in der Spezifikation von Weblinks und Datenbankservern* erläutert.
- Befehlszeilenparameter können als Variablen im Verbindungsstring verwendet werden.



ODBC can be used for connections to Microsoft Access or PostgreSQL 9.1 database servers. Please note, however, that performance will be lower compared to native drivers and stored procedures are not supported.



Im Folgenden finden Sie Verbindungsstrings für Verbindungen zu verschiedenen Datenbankservern.

Verbindungsstring für Verbindung mit einem Microsoft SQL Server® mit Standardanmeldung:

```
Server=myServerAddress;Database=myDataBase;User Id=myUsername;Password=myPassword;
```

Verbindungsstring für direkte Verbindung mit einem Oracle®-Datenbankserver:

```
User Id=APP_ID;Password=*****;Direct=true;Data Source=servername;Port=Port;SID=orcl
```

Verbindungsstring für direkte Verbindung mit einem Oracle®-Datenbankserver einschließlich Servicename:

```
User Id=*****;Password=*****;Server=ServerFQDN/IP;Unicode=True;Connection Timeout=60;Max Pool Size=250;Direct=True;Service Name=ServiceName;Port=1521
```

Verbindungsstring für indirekte Verbindung mit einem Oracle®-Datenbankserver:

```
User Id=***;Password=*****;Server=orcl;Unicode=True;Connection Timeout=60;Max Pool Size=250
```

- **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung aller Importeinträge innerhalb der Importgruppe auszuwählen. Wählen Sie `False` aus, um die Ausführung zu deaktivieren.
- 3) Klicken Sie mit der rechten Maustaste im Explorer auf die Datenbank-Importgruppe , und wählen Sie **Eintrag erzeugen** aus. Der Datenbank-Importgruppe  wird ein neuer Importeintrag hinzugefügt.
 - 4) Klicken Sie mit der rechten Maustaste auf den Ordner **SQL-Befehle - DataUpload** unter dem Importeintrag  und wählen Sie **SQL-Befehl erzeugen** aus. Dem Ordner wird ein neuer SQL-Befehl  hinzugefügt.
 - 5) Klicken Sie im Explorer auf den neuen SQL-Befehl, und legen Sie im Attributfenster folgende Attribute fest:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
 - **Befehlstyp:** Wählen Sie im Dropdown-Listefeld `DMLStatement` aus.
 - **Ergebnistyp:** Wählen Sie `Undefined` aus.
 - **Fehler ignorieren:** Wählen Sie `True` aus, wenn das Hochladen selbst dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn das Hochladen im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
 - **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.
 - **Text:** Definieren Sie eine SQL-Abfrage mit einer `SELECT`-Anweisung, die einen Datensatz zurückgibt, wenn sie auf der externen Datenbank ausgeführt wird. Der resultierende Datensatz wird durch ADIF als Importdaten hochgeladen. Schreiben Sie die SQL-Abfrage entweder direkt in das Attributfeld, oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.



Der Texteditor zur Definition von SQL-Abfragen in der ADIF-Konfigurationsbenutzeroberfläche bietet Hilfe zum Definieren von SQL-Abfragen in separaten Registerkarten. Informationen hierzu finden Sie unter *Definieren von SQL-Abfragen für SQL-Befehle* im Abschnitt *Konfigurieren von ADIF-Schemata*.



Wenn Sie komplexe Importszenarien konfigurieren möchten, können Sie dem Importeintrag zusätzliche SQL-Befehle hinzufügen, die vor oder nach dem Datenupload ausgeführt werden. Weitere Informationen über die Definition zusätzlicher SQL-Befehle innerhalb eines Importeintrags finden Sie unter *Konfigurieren von SQL-Befehlen für optionale erweiterte Importfunktionen*.

- 6) Definieren Sie den Import des Datensatzes, der mithilfe des SQL-Befehls für den Datenupload in den Importeintrag erstellt wurde. Weitere Informationen über die erforderliche Konfiguration finden Sie unter *Definieren des Imports externer Daten in einen Importeintrag*.
- 7) Soll mehr als ein Datensatz aus der externen Datenbank hochgeladen werden, wiederholen Sie die Schritte 3 bis 6.



Nach der Erzeugung der Datenbank-Importgruppe müssen Sie Importeinträge hinzufügen, die die Daten in der externen Datenbank den Daten in der Alfabet-Datenbank zuordnen. Informationen hierzu finden Sie unter *Definieren des Imports externer Daten in einen Importeintrag*.

Optional können Sie SQL-Befehle definieren, mit denen Sie diesen Teil des Importschemas deaktivieren können, wenn bestimmte Voraussetzungen erfüllt sind. Weitere Informationen hierzu finden Sie unter *Konfigurieren der Ausführung des Importschemas in Abhängigkeit von aktuellen Parametern*.

Definieren des Hochladens von Daten aus einer LDAP-Tabelle in eine LDAP-Importgruppe

Wenn Sie Daten aus einer externen LDAP-Tabelle importieren möchten (z. B. Microsoft® Active Directory®), muss eine Verbindung zu der LDAP-Tabelle hergestellt werden, um die Daten abzurufen. Mit dem ADIF-Importvorgang werden die Daten nicht direkt aus den LDAP-Tabellen gelesen. Stattdessen müssen die Daten über einen benutzerdefinierten LDAP-Suchfilter hochgeladen werden, der auf die LDAP-Tabelle angewandt wird. Der aus dem Suchfilter resultierende Datensatz stellt die externen Daten dar, die mithilfe des ADIF-Importvorgangs importiert werden.

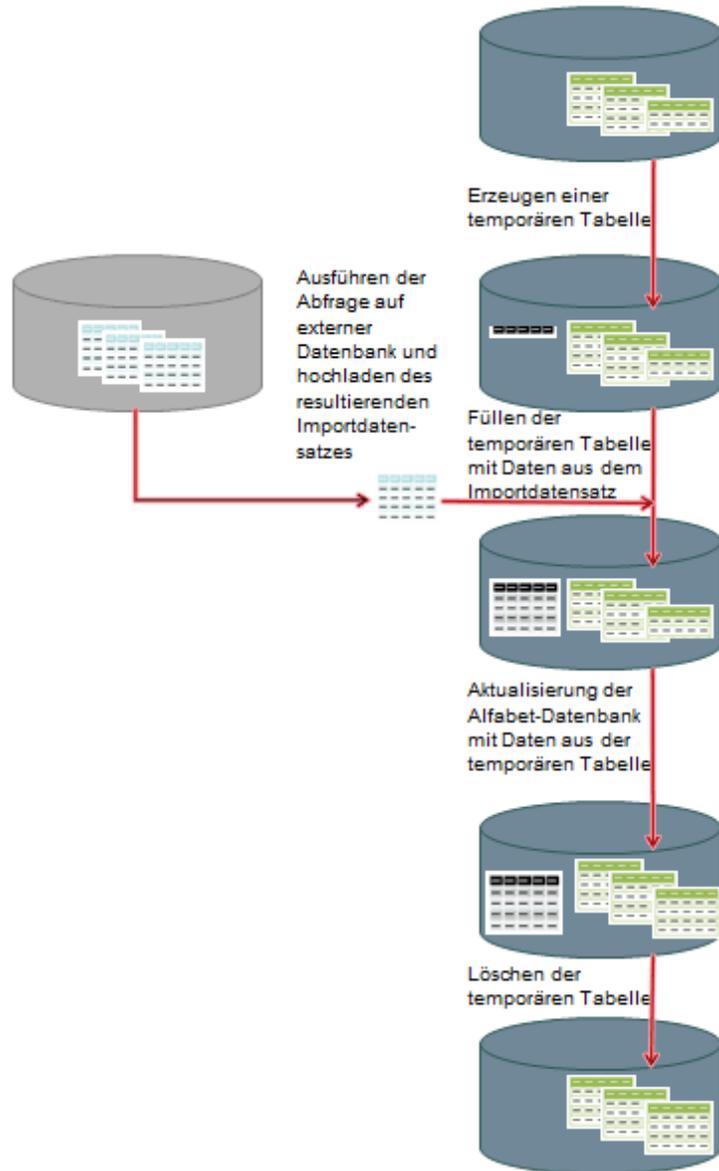


Abbildung: Importieren von Daten aus einer externen LDAP-Tabelle

So konfigurieren Sie das Hochladen von Daten aus der LDAP-Tabelle:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Importschema, und wählen Sie **LDAP-Importgruppe erzeugen** aus. Dem Explorer wird eine neue LDAP-Importgruppe  hinzugefügt. Für die LDAP-Importgruppe wird ein Standardname bereitgestellt. Unter der LDAP-Importgruppe  ist ein Ordner für die Erzeugung von SQL-Befehlen verfügbar, die bei Aktivierung der LDAP-Importgruppe ausgeführt werden.
- 2) Geben Sie im Attributfenster der LDAP-Importgruppe  die folgenden Attribute an, um die Verbindung zu der LDAP-Datenbank zu konfigurieren:
 - **Name:** Geben Sie einen aussagekräftigen Namen für die LDAP-Importgruppe ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit der LDAP-Importgruppe implementiert wird.
 - **Authentifizierungsmethode:** Wählen Sie im Dropdown-Listefeld die Authentifizierungsmethode aus, die für die Anmeldung bei der externen LDAP-Tabelle erforderlich ist:
 - **None:** Im LDAP-Anbieter wird die Standardauthentifizierung (einfache Verknüpfung) verwendet.
 - **Secure:** Wählen Sie aus, ob der WinNT-Anbieter den Client mit NTLM authentifiziert. In den Active Directory-Domänendiensten wird Kerberos und möglicherweise NTLM verwendet, um den Client zu authentifizieren. Wenn der Benutzername und das Kennwort einen Nullverweis darstellen (Nichts in Visual Basic), stellt ADSI mithilfe des Sicherheitskontextes des aufrufenden Threads eine Verknüpfung her. Dies ist entweder der Sicherheitskontext des Benutzerkontos, unter dem die Anwendung ausgeführt wird, oder der jenes Benutzerkontos, das von dem aufrufenden Thread verkörpert wird.
 - **Encryption:** Fügt eine kryptografische Signatur an die Meldung an, mit der sowohl der Absender identifiziert als auch gewährleistet wird, dass die Meldung während des Übergangs nicht verändert wurde.
 - **SecureSocketsLayer:** Fügt eine kryptografische Signatur an die Meldung an, mit der sowohl der Absender identifiziert als auch gewährleistet wird, dass die Meldung während des Übergangs nicht verändert wurde. Für die Active Directory-Domänendienste muss der Zertifikatsserver installiert sein, damit die Secure Sockets Layer-Verschlüsselung (SSL) unterstützt wird.
 - **Anonymous:** Es wird keine Authentifizierung durchgeführt.
 - **FastBind:** Gibt an, dass ADSI nicht versuchen wird, die Eigenschaft "objectClass" der Active Directory-Domänendienste abzufragen. Daher werden nur diejenigen Basisschnittstellen offengelegt, die von allen ADSI-Objekten unterstützt werden. Andere Schnittstellen, die von dem Objekt unterstützt werden, stehen nicht zur Verfügung. Ein Anwender kann diese Option verwenden, um die Leistung in einer Reihe von

Objektmanipulationen zu steigern, die nur Methoden der Basisschnittstellen beinhalten. Mit ADSI wird jedoch nicht überprüft, ob eines der Anfrageobjekte tatsächlich auf dem Server vorhanden ist.

- **Signing:** Überprüft die Datenintegrität, um zu gewährleisten, dass die empfangenen Daten mit den gesendeten identisch sind. Die Kennzeichnung "System.DirectoryServices.AuthenticationTypes.Secure" muss ebenfalls festgelegt sein, damit "Signing" verwendet werden kann.
- **Sealing:** Verschlüsselt Daten mithilfe von Kerberos. Die Kennzeichnung "System.DirectoryServices.AuthenticationTypes.Secure" muss ebenfalls festgelegt sein, damit "Sealing" verwendet werden kann.
- **Delegation:** Ermöglicht der Active Directory Service Interface (ADSI), den Sicherheitskontext des Benutzers zu delegieren. Das ist erforderlich, damit Objekte zwischen Domänen verschoben werden können.
- **ServerBind:** Wenn Ihr "ADsPath" einen Servernamen enthält, geben Sie diese Kennzeichnung an, wenn Sie den LDAP-Anbieter verwenden. Verwenden Sie diese Kennzeichnung nicht für Pfade, die einen Domänennamen enthalten, und auch nicht für serverlose Pfade. Die Angabe eines Servernamens, ohne dass auch diese Kennzeichnung angegeben wird, führt zu unnötigem Netzwerkdatenverkehr.
- **Login-Name:** Geben Sie den Login-Namen für die Anmeldung in der externen Quelle an. Eine Servervariable kann verwendet werden, um den Login-Namen als Verknüpfung zu den Informationen zu definieren, die in der Server-Alias-Konfiguration von Alfabet-Server verschlüsselt gespeichert sind. Informationen über die Definition und Spezifikation von Servervariablen finden Sie unter *Definieren von Verbindungen auf Basis von Servervariablen* im Referenzhandbuch *Systemadministration*.
- **Login-Passwort:** Geben Sie das Login-Passwort für das Einloggen in der externen Quelle an. Eine Servervariable kann verwendet werden, um das Login-Passwort als Verknüpfung zu den Informationen zu definieren, die in der Server-Alias-Konfiguration von Alfabet-Server verschlüsselt gespeichert sind. Informationen über die Definition und Spezifikation von Servervariablen finden Sie unter *Definieren von Verbindungen auf Basis von Servervariablen* im Referenzhandbuch *Systemadministration*.
- **Verbindungszeichenfolge:** Geben Sie die Verbindungszeichenfolge ein, die für die Verbindung mit der externen LDAP-Tabelle erforderlich ist. Die Verbindungszeichenfolge ist von der Konfiguration der externen LDAP-Tabelle abhängig.



Es wird empfohlen, das Datenbank-Login im Verbindungsstring über einen Benutzernamen und ein Kennwort zu definieren. Wenn die Windows-Authentifizierung zur Anmeldung verwendet wird und die ADIF-Konsolenapplikation mit einem Remote-Alias gestartet wird, wird nur dann erfolgreich eine Verbindung zur Datenbank hergestellt, wenn der Alfabet-Server mit demselben Domänen-Benutzernamen und denselben Zugriffsrechten gestartet wird wie im Verbindungsstring angegeben.



Hinweis: In den Hilfe-Links können Server-Variablen genutzt werden. Mithilfe von Servervariablen können Sie die gesamte oder einen Teil der Definition der Server-Alias-Konfiguration des Alfabet-Server s definieren, anstatt diese direkt im ADIF-Schema zu definieren. Der Einsatz von

Servervariablen ist vor allem dann sinnvoll, wenn Sie zum Beispiel die Konfiguration in einer Test- bzw. Produktionsumgebung mit unterschiedlichen externen Quellen verwenden. In beiden Umgebungen kann dieselbe Definition der externen Quelle verwendet werden. Mit der in der Test- und Produktionsumgebung verwendeten Definition des Serveralias werden die korrekten Verbindungsdaten für die externe Quelle definiert, die in der jeweiligen Umgebung verwendet wird. Siehe *Definieren von Verbindungen auf Basis von Servervariablen* im Referenzhandbuch *Systemadministration*.

- **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung aller Importeinträge innerhalb der Importgruppe auszuwählen. Wählen Sie `False` aus, um die Ausführung zu deaktivieren.
- 3) Klicken Sie mit der rechten Maustaste im Explorer auf die LDAP-Importgruppe , und wählen Sie **Eintrag erzeugen** aus. Der LDAP-Importgruppe wird ein neuer Importeintrag hinzugefügt.
 - 4) Klicken Sie mit der rechten Maustaste auf den Ordner **SQL-Befehle - DataUpload**, und wählen Sie **SQL-Befehl erzeugen** aus. Dem Ordner wird ein neuer SQL-Befehl  hinzugefügt.
 - 5) Klicken Sie auf den neuen SQL-Objektknoten  und bearbeiten Sie im Attributfenster optional folgende Attribute:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
 - **Befehlstyp:** Wählen Sie im Dropdown-Listefeld `DMLStatement` aus.
 - **Ergebnistyp:** Wählen Sie `Undefined` aus.
 - **Fehler ignorieren:** Wählen Sie `True` aus, wenn das Hochladen selbst dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn das Hochladen im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
 - **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.
 - **Text:** Definieren Sie einen LDAP-Suchfilter gemäß RFC 2254, der bei Ausführung in der LDAP-Tabelle einen Datensatz zurückgibt. Der resultierende Datensatz wird durch ADIF als Importdaten hochgeladen. Schreiben Sie die Abfrage entweder direkt in das Attributfeld, oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der Abfrage auf der Registerkarte **SQL-Text** des Editors zu öffnen.



Der Texteditor zur Definition von Abfragen in der ADIF-Konfigurationsbenutzeroberfläche bietet Hilfe zum Definieren der Abfrage in separaten Registerkarten. Informationen hierzu finden Sie unter *Definieren von SQL-Abfragen für SQL-Befehle* im Abschnitt *Konfigurieren von ADIF-Schemata*.



Wenn Sie komplexe Importszenarien konfigurieren möchten, können Sie dem Importeintrag zusätzliche SQL-Befehle hinzufügen, die vor oder nach dem Datenupload ausgeführt werden. Weitere Informationen über die Definition zusätzlicher SQL-Befehle innerhalb eines Importeintrags finden Sie unter *Konfigurieren von SQL-Befehlen für optionale erweiterte Importfunktionen*.

- 6) Definieren Sie den Import des Datensatzes, der über den SQL-Befehl für den Datenupload in die Importgruppe erstellt wurde. Weitere Informationen über die erforderliche Konfiguration finden Sie unter *Definieren des Imports externer Daten in einen Importeintrag*.
- 7) Soll mehr als ein Datensatz aus der externen Datenbank hochgeladen werden, wiederholen Sie die Schritte 3 bis 6.



Nach der Erzeugung der LDAP-Importgruppe müssen Sie nun Importeinträge hinzufügen, die die Daten in der LDAP-Tabelle den Daten in der Alfabet-Datenbank zuordnen. Informationen hierzu finden Sie unter *Definieren des Imports externer Daten in einen Importeintrag*.

Optional können Sie SQL-Befehle definieren, mit denen Sie diesen Teil des Importschemas deaktivieren können, wenn bestimmte Voraussetzungen erfüllt sind.

Weiterführende Informationen finden Sie unter *Konfigurieren der bedingten Ausführung von Teilen des Importschemas*.

Definieren des Datenimports aus XML-Dateien in einer XML-Importgruppe

Wenn Sie Daten aus einer XML-Datei importieren, muss die Struktur der XML-Datei in die Tabellenstruktur der Datenbank übersetzt werden. XML-Dateien sind hierarchisch strukturiert, und in den meisten Fällen muss mehr als eine Tabelle erstellt werden, um die in der XML-Datei bereitgestellten Daten zu speichern. Daher erfordert der Import aus einer einzelnen XML-Datei, dass dem ADIF-Importschema eine Importgruppe als Container für alle temporären Tabellen hinzugefügt wird, die erzeugt und mit den Daten aus der XML-Datei gefüllt werden müssen.

Grundsätzlich gilt, dass für jedes XML-Element wenigstens eine temporäre Tabelle erzeugt werden muss, die die folgenden Bedingungen erfüllt:

- Der XML-Elementtyp enthält XML-Attribute
- Der XML-Elementtyp enthält untergeordnete Elemente ohne XML-Attribute und/oder Textinhalte.

Für jede temporäre Tabelle, die definiert werden muss, muss der **XML-Importgruppe** ein **Importeintrag** hinzugefügt werden. Der Name der Importtabelle ist der Name des XML-Elements.

Die Spalten der temporären Tabelle müssen dem Inhalt des XML-Elements zugeordnet werden. Die temporäre Tabellendefinition muss ein untergeordnetes Element **Attribut** für die Konfiguration einer Datenbankspalte enthalten. Das Attribut **Importspalte** des Elements **Attribut** definiert, welche Daten aus dem XML-Element in die Spalte der temporären Tabelle geschrieben werden. Im Importeintrag müssen Sie Folgendes definieren:

- Ein Element **Attribut** für jedes XML-Attribut des Import-XML-Elements. Der Name des XML-Attributs muss in der **Importspalte** angegeben werden.
- Ein Element **Attribut** für den Textinhalt im Import-XML-Element. `<Name des XML-Elements>_text` muss in der **Importspalte** angegeben werden.
- Ein Element **Attribut** für jedes untergeordnete XML-Element ohne XML-Attribute (jedoch mit Textinhalt), das im Import-XML-Element enthalten ist. Der Name des untergeordneten XML-Elements muss in der **Importspalte** angegeben werden.



Wenn Text, der in ein XML-Element geschrieben wurde, Elemente enthält oder vollständig darin eingehüllt ist, z. B. für die Formatierung, kann der Text während des Imports nicht gelesen werden. Im folgenden Code z. B. kann der Status der Anwendung nicht aus dem Text importiert werden, da der Status `Retired` in ein untergeordnetes XML-Element mit eigenen XML-Attributen geschrieben ist:

```
<Application Name="App1">Active</Application>
<Application Name="App2">Planned</Application>
<Application Name="App3"><Style Format="Bold">Retired</Style></Application>
```

Außerdem können den temporären Tabellen Spalten hinzugefügt werden, mit denen die Hierarchie der XML-Elemente in der XML-Datei definiert wird. Vom Importmechanismus wird eine ID für alle XML-Elemente in der Hierarchie generiert. Folgende Konfiguration ist erforderlich, um untergeordnete XML-Elemente ihren jeweiligen übergeordneten Elementen zuzuordnen:

- In dem **Importeintrag**, der die temporäre Tabelle des übergeordneten XML-Elements definiert, muss ein Element **Attribut** hinzugefügt werden. Der Wert für **Importspalte** muss `<Elementname>_Id` sein.
- In dem **Importeintrag**, der die temporäre Tabelle des untergeordneten XML-Elements definiert, muss ein Element **Attribut** hinzugefügt werden. Der Wert für **Importspalte** muss als `<übergeordneter Elementname>_Id` definiert sein.

Die Hierarchieinformationen werden in der Regel nicht in Objektklasseneigenschaften im Ziel Alfabet-Datenbank importiert. Die Hierarchieinformationen müssen in den temporären Tabellen vorhanden sein, um das übergeordnete XML-Element in SQL-Befehlen zu identifizieren, die für den Import von Beziehungen zwischen Objekten definiert sind.

Weitere Informationen hierzu finden Sie unter im Abschnitt *Definieren von Beziehungen*.



Zum Beispiel sollen Daten über Geräte und deren Zuweisung zu Gerätegruppen aus der folgenden XML-Datei importiert werden:

```
<Groups>
  <DeviceGroup Name="ImportGroup1">
    <Description>Server hosts</Description>
    <Device Name="ImportDevice1" Version="1.0" StartDate="2011.01.01" EndDate="2015.01.05"/>
    <Device Name="ImportDevice2" Version="1.2" StartDate="2011.08.24" EndDate="2014.12.31"/>
  </DeviceGroup>
  <DeviceGroup Name="ImportGroup2">
    <Description>Web relevant devices</Description>
    <Device Name="Backup Server" Version="2.2.1" StartDate="2009.09.30" EndDate="2013.09.01"/>
    <Device Name="ImportDevice1" Version="1.0" StartDate="2011.01.01" EndDate="2015.01.05"/>
  </DeviceGroup>
</Groups>
```

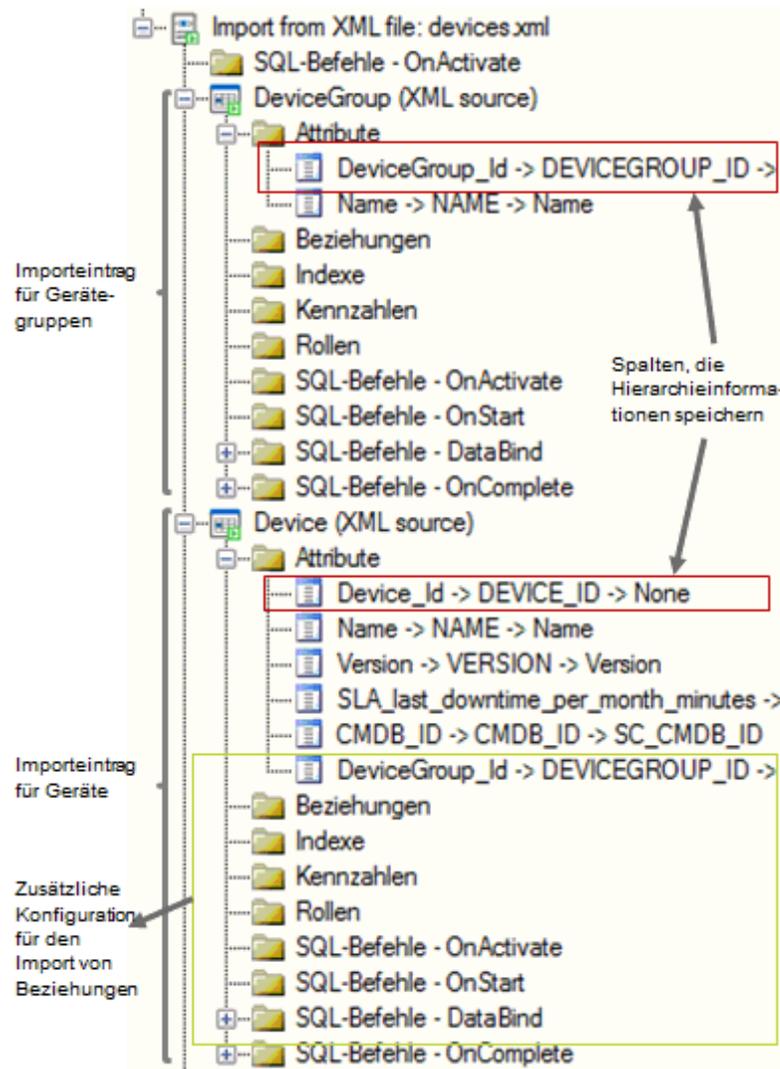
Die XML-Elemente in der Datei erfordern die folgende Konfiguration der XML-Importgruppe:

- **Gruppen:** Keine Konfiguration erforderlich. Das Root-XML-Element **Gruppen** enthält keine eigenen Informationen. Das bedeutet, dass keine XML-Attribute oder untergeordneten XML-Elemente ohne XML-Attribute einbezogen werden.
- **Gerätegruppe** Eine temporäre Tabelle muss mit einem **Import-Eintrag**  innerhalb der ADIF- **XML-Importgruppe**  definiert werden.
Dieses XML-Element speichert Informationen in XML-Attributen und verfügt über ein untergeordnetes XML-Element ohne XML-Attribute. Das XML-Element verfügt auch über untergeordnete XML-Elemente mit XML-Attributen. Damit alle Daten über Gerätegruppen aus der Datei importiert werden, muss die temporäre Tabelle über eine Spalte für das XML-Attribut **Name**, eine für das XML-Element **Beschreibung** und eine für die hierarchische ID-Definition verfügen.
- **Gerät:** Eine temporäre Tabelle muss mit einem Import-Eintrag  innerhalb der ADIF-XML-Importgruppe  definiert werden.

In diesem XML-Element werden Informationen in XML-Attributen gespeichert. Es ist einem übergeordneten XML-Element mit eigenen XML-Attributen zugeordnet. Damit alle Daten über Gerätegruppen aus der Datei importiert werden, muss die temporäre Tabelle über eine Spalte für jedes XML-Attribut und eine für die hierarchische ID-Definition des übergeordneten XML-Elements verfügen.

- **Beschreibung:** Das XML-Element verfügt nicht über eigene XML-Attribute, aber über Textinhalte und ein übergeordnetes XML-Element mit Attributen. Es ist als Spalte der temporären Tabelle seines übergeordneten XML-Elements definiert.

Die resultierende Importschemadefinition ist unten zu sehen. Das Beispiel ist eine vollständige Konfiguration eines funktionierenden Imports. Daher sind auch zusätzliche Konfigurationen für den Import der Beziehungen enthalten:





Es wird empfohlen, den halbautomatischen Interpretationsmechanismus zum Importieren von XML-Dateien zu verwenden, der in der ADIF-Konfigurationsbenutzeroberfläche bereitgestellt wird, anstatt die Konfigurationsschritte wie im Folgenden beschrieben nacheinander durchzuführen. Weitere Informationen hierzu finden Sie unter *Halbautomatische Erzeugung von Importeintragsdefinitionen für Importdateien*.

So konfigurieren Sie das Hochladen von Daten aus einer XML-Datei:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Importschema, und wählen Sie **XML-Importgruppe erzeugen** aus. Dem Explorer wird eine neue **XML-Importgruppe**  hinzugefügt. Für die XML-Importgruppe wird ein Standardname bereitgestellt. Unterhalb der **XML-Importgruppe**  wird automatisch ein Ordner für die Erzeugung von SQL-Befehlen hinzugefügt, die bei Aktivierung der XML-Importgruppe ausgeführt werden sollen.
- 2) Legen Sie im Attributfenster der **XML-Importgruppe**  die folgenden Attribute zur Definition der Importdatei fest:
 - **Name:** Geben Sie einen aussagekräftigen Namen für die XML-Importgruppe ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit der XML-Importgruppe implementiert wird.
 - **XML-Datei importieren:** Geben Sie den Namen der XML-Datei ein, aus der die Daten importiert werden sollen. Der Name muss die Dateierweiterung enthalten.
 - **Quell-XSD-Datei:** Geben Sie optional den Namen der XSD-Datei ein, mit der das Schema für die XML-Importdatei definiert wird. Dieses Attribut dient lediglich als Information für den Entwickler des ADIF-Schemas und hat keine technischen Auswirkungen.
 - **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung aller Importeinträge innerhalb der Importgruppe auszuwählen. Wählen Sie `False` aus, um die Ausführung zu deaktivieren.
- 3) Klicken Sie mit der rechten Maustaste im Explorer auf die **XML-Importgruppe** , und wählen Sie **Eintrag erzeugen** aus. Der XML-Importgruppe wird ein neuer **Importeintrag**  hinzugefügt.
- 4) Definieren Sie den Import eines XML-Elementtyps in Ihrer Import-XML-Datei in einer temporäre Tabelle und die Aktualisierung einer Alfabet-Datenbank - Zieltabelle mit den Informationen in der temporären Tabelle.

Weitere Informationen über die erforderliche Konfiguration finden Sie unter *Definieren des Imports externer Daten in einen Importeintrag*.

- 5) Ist in Ihrer Import-XML-Datei mehr als ein XML-Element enthalten, das den Kriterien für die Zuordnung zu einer temporären Tabelle entspricht, wiederholen Sie für jedes XML-Element die Schritte 3 und 4.



Optional können Sie SQL-Befehle definieren, mit denen Sie diesen Teil des Importschemas deaktivieren können, wenn bestimmte Voraussetzungen erfüllt sind.

Weiterführende Informationen finden Sie unter *Konfigurieren der bedingten Ausführung von Teilen des Importschemas*.

Vorverarbeiten von XML-Dateien als Teil des Importprozesses

Wenn die XML-Struktur, die Sie importieren möchten, nicht mit der gewünschten Struktur für den Import über ein ADIF-Importschema übereinstimmt, z. B. weil Sie Formatierungselemente im Text oder Elemente mit unterschiedlichen Informationen in verschiedenen Hierarchieebenen der XML-Datei den gleichen Namen haben, können Sie im ADIF-Importschema eine XSL-Transformationsdefinition definieren, die die XML-Struktur aller zu importierenden Dateien in ein für den ADIF-Import geeignetes Format konvertiert und die aus der XSL-Transformation resultierende XML-Struktur importiert.

Wenn Sie den XSL-Transformationscode im ADIF-Importschema vor der Definition der ADIF-Elemente des XML-Imports definieren, wird die XSL-Transformation auch auf das importierte XML angewendet, bevor die Funktion **XML-Gruppe aus Datei erzeugen** ausgeführt wird.

So wenden Sie die XSL-Vorverarbeitung auf importierte XML-Dateien an:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche auf das ADIF-Importschema, um das XML zu importieren. Das Attributfenster öffnet sich.
- 2) Klicken Sie im Feld des Attributs „XSL-Transformation“ auf die Schaltfläche **Durchsuchen** . Ein Texteditor wird angezeigt.
- 3) Definieren Sie den XSL-Code im Editor.



Bitte beachten Sie, dass der Begriff "Aktivieren" in der XSL-Datei aus Sicherheitsgründen nicht erlaubt ist. Zum Beispiel eine Anweisung wie:

```
<xsl:apply-templates select="//*[@*]"/>
```

ist in der XSL bei Zuordnung zu einem ADIF-Importschema nicht erlaubt.

- 4) Klicken Sie auf OK, um Ihre Änderungen zu speichern.

Definieren des Datenimports aus JSON-Dateien

JSON-Daten werden in eine vordefinierte temporäre Tabellenstruktur importiert. Für jedes Schlüssel-/Wertepaar in einem JSON-Objekt wird eine Zeile in die temporäre Tabelle geschrieben.

Die temporäre Tabelle verfügt über folgende Spalten:

- **OBJID:** Jedem JSON-Objekt in der Importdatei wird eine interne ID zugewiesen, um das JSON-Objekt eindeutig zu identifizieren und die hierarchische Struktur zu erzeugen.
- **PARENTOBJID:** Wenn ein JSON-Objekt in einem Eigenschaftsfeld eines übergeordneten JSON-Objekts definiert wird, wird die interne Objekt-ID in der Spalte **OBJID** des übergeordneten Objekts in die Spalte **PARENTOBJID** für das untergeordnete JSON-Objekt geschrieben.
- **PARENTPROPNAME:** Wenn ein JSON-Objekt in einem Schlüssel eines übergeordneten JSON-Objekts definiert ist, enthält diese Spalte den Namen des Schlüssels.
- **PROPNAME:** Der Name des JSON-Schlüssels.
- **PROPSLIST:** Ein boolescher Wert, der "wahr" lautet, wenn der Schlüssel eine Liste von JSON-Objekten enthält.
- **VALUEINTEGER:** Wenn der Wert für den Schlüssel eine Ganzzahl ist, wird er in diese Spalte geschrieben.
- **VALUEFLOAT:** Wenn der Wert für den Schlüssel ein Gleitkommawert ist, wird er in diese Spalte geschrieben.
- **VALUEDATE:** Wenn der Wert für den Schlüssel ein Datum ist, wird er in diese Spalte geschrieben.
- **VALUETEXT:** Wenn der Wert für den Schlüssel ein Text oder eine Zeichenfolge ist, wird er in diese Spalte geschrieben.
- **VALUEBOOLEAN:** Wenn der Wert für den Schlüssel ein boolescher Wert ist, wird er in diese Spalte geschrieben.

Die Tabelle wird für verschiedene JSON-Strukturen wie folgt gefüllt. Beispiel:

Import von Text-, Zeichenfolgen-, Ganzzahl-, Gleitkomma-, booleschen und Datums-Schlüssel-/Wertpaaren auf Root-Ebene

Für jedes Schlüssel-/Wertepaar ist in der Tabelle eine Zeile vorhanden.

Die **OBJID** ist für alle Schlüssel-/Wertpaare identisch, die zum selben Objekt gehören.

PARENTOBJID und **PARENTPROPNAME** sind auf der Root-Ebene leer.

Der Schlüssel wird in die Spalte `PROPNAME` geschrieben, während der Wert in die Spalte `VALUE<Datentyp>` geschrieben wird, die dem Datentyp des Schlüssels entspricht.

```
{
  "TextProperty": "This is a root property",
  "DateProperty": "2018-02-13T09:15:00",
  "IntegerProperty": 16
}
```

	OBJID	PARENTOBJID	PARENTPROPNAME	PROPNAME	PROPSLIST	VALUEINTEGER	VALUEFLOAT	VALUEDATE	VALUETEXT
1	1			TextProperty					This is a root property
2	1			DateProperty				13/02/2018 09:15:00	
3	1			IntegerProperty		16			

Importieren eines Array-Schlüssel-/Wertpaars auf Root-Ebene

Für jeden Wert im Array ist in der Tabelle eine Zeile vorhanden.

Alle Werte gehören zum gleichen Objekt und zum gleichen Schlüssel. Daher sind die Objektkennung, die in die Spalte `OBJID` geschrieben wird, und der in die Spalte `PROPNAME` geschriebene Schlüsselname für alle Zeilen identisch.

Die Spalte `PROPSLIST` ist auf "1" (= wahr) gesetzt, da der Schlüsselwert eine Liste von Werten ist. Im folgenden Beispiel des resultierenden Datensatzes wird dieser vom SQL-Parser in ein x konvertiert.

`PARENTOBJID` und `PARENTPROPNAME` sind auf der Root-Ebene leer.

Die Werte werden in die Spalte `VALUE<Datentyp>` geschrieben, die ihrem Datentyp entspricht.

```
{
  "ArrayProperty": ["Value1", "Value2"]
}
```

	OBJID	PARENTOBJID	PARENTPROPNAME	PROPNAME	PROPSLIST	VALUEINTEGER	VALUEFLOAT	VALUEDATE	VALUETEXT
1	1			ArrayProperty	x				Value1
2	1			ArrayProperty	x				Value 2

Importieren einer Objektliste auf Root-Ebene

Für jedes Schlüssel-/Wertpaar der Root-Objekte ist in der Tabelle eine Zeile vorhanden.

Jedes Objekt in der Liste verfügt über eine eindeutige OBJID. Die OBJID ist für alle Schlüssel-/Wertpaare identisch, die zum selben Objekt gehören.

PARENTOBJID und PARENTPROPNAME sind leer, da alle Objekte Root-Objekte sind.

Der Schlüssel wird in die Spalte PROPNAME geschrieben, während der Wert in die Spalte VALUE<Datentyp> geschrieben wird, die dem Datentyp des Schlüssels entspricht.

```
[
  {
    "RootTextProperty": "This is the first object in a list",
    "RootIntegerProperty": 1
  },
  {
    "RootTextProperty": "This is the second object in a list",
    "RootIntegerProperty": 2
  }
]
```

OBJID	PARENTOBJID	PARENTPROPNAME	PROPNAME	PROPSLIST	VALUEINTEGER	VALUEFLOAT	VALUEDATE	VALUETEXT
1			RootTextProperty					This is the first object in a list
2			RootIntegerProperty		1			
3			RootTextProperty					This is the second object in a list
4			RootIntegerProperty		2			

Importieren eines Schlüssel-/Wertpaars eines Root-Objekts, das ein untergeordnetes Objekt enthält

Es gibt eine Zeile für den Schlüssel, der das Objekt als Wert enthält, und eine Zeile für jedes Schlüssel-/Wertpaar des untergeordneten Objekts.

In der Zeile für den Schlüssel, der das Objekt enthält, wird die Objekt-ID des Root-Objekts in die Spalte OBJID geschrieben und der Schlüssel in die Spalte PROPNAME. Alle anderen Spalten sind leer.

In den Zeilen für die Schlüssel-/Wertpaare der untergeordneten Objekte sind `PARENTOBJID` und `PARENTPROPNAME` mit den Spaltenwerten `OBJID` und `PROPNAME` des Schlüssels identisch, der das Objekt enthält.

Die `OBJID` des untergeordneten Objekts ist eine eindeutige ID, die Informationen über die Hierarchie in der Datei bereitstellt. Der Objekt-ID-Wert beginnt mit der Objekt-ID des Root-Vorgängerobjekts und fügt die ID jedes Objekts in der Hierarchie hinzu, wobei die Ebenen durch einen Punkt getrennt sind. Ein Objekt auf der dritten Ebene der Hierarchie hätte z. B. die Objekt-ID 1.1.1.

In den Zeilen für die Schlüssel-/Wertpaare der untergeordneten Objekte wird der Schlüssel `PROPNAME` in die Spalte geschrieben, während der Wert in die Spalte `VALUE<Datentyp>` geschrieben wird, die dem Datentyp des Objekts entspricht.

```
{
  "ObjectProperty":
  {
    "SubTextProperty": "This is a subordinate object",
    "SubIntegerProperty": 1
  }
}
```

	OBJID	PARENTOBJID	PARENTPROPNAME	PROPNAME	PROPLIST	VALUEINTEGER	VALUEFLOAT	VALUEDATE	VALUETEXT
1	1			ObjectProperty					
2	1.1	1	ObjectProperty	SubTextProperty					This is a subordinate object
3	1.1	1	ObjectProperty	SubIntegerProperty		1			

Importieren des Schlüssel-/Wertpaars eines Root-Objekts, das eine Liste von Objekten enthält

Für jedes Schlüssel-/Wertpaar der untergeordneten Objekte gibt es eine Zeile.

In jeder Zeile wird die Objekt-ID des Objekts, das den Schlüssel enthält, der die Objektliste zurückgibt, in die Spalte `OBJID` geschrieben. Der Name dieses Schlüssels wird in die Spalte `PROPNAME` geschrieben.

Die `OBJID` der untergeordneten Objekte ist eine eindeutige ID, die Informationen über die Hierarchie in der Datei bereitstellt. Der Objekt-ID-Wert beginnt mit der Objekt-ID des Root-Vorgängerobjekts und fügt die ID jedes Objekts in der Hierarchie hinzu, wobei die Ebenen durch einen Punkt getrennt sind. Ein Objekt auf der dritten Ebene der Hierarchie hätte z. B. die Objekt-ID 1.1.1. Alle Zeilen, die zum gleichen Objekt gehören, haben dieselbe ID.

In den Zeilen für die Schlüssel-/Wertpaare der untergeordneten Objekte wird der Schlüssel `PROPNAME` in die Spalte geschrieben, während der Wert in die Spalte `VALUE<Datentyp>` geschrieben wird, die dem Datentyp des Objekts entspricht.

```
{
  "ObjectListProperty": [
    {
      "SubTextProperty": "This is the first object in a list",
      "SubIntegerProperty": 1
    },
    {
      "SubTextProperty": "This is the second object in a list",
      "SubIntegerProperty": 2
    }
  ]
}
```

OBJID	PARENTOBJID	PARENTPROPNAME	PROPNAME	PROPSLIST	VALUEINTEGER	VALUEFLOAT	VALUEDATE	VALUETEXT
1	1.1	ObjectListProperty	SubTextProperty					This is the first object in a list
2	1.1	ObjectListProperty	SubIntegerProperty		1			
3	1.2	ObjectListProperty	SubTextProperty					This is the second object in a list
4	1.2	ObjectListProperty	SubIntegerProperty		2			

So definieren Sie einen Importeintrag für den Import eines JSON-Objekts:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Importschema, und wählen Sie **Eintrag für hierarchische JSON erzeugen** aus. Dem Explorer wird ein neuer **Importeintrag**  hinzugefügt. Für den Importeintrag wird ein Standardname bereitgestellt.



Die Kontextmenüoptionen **JSON-Importgruppe erzeugen** und **JSON-Gruppe aus Datei erzeugen** sind aus Gründen der Abwärtskompatibilität verfügbar. Diese Optionen können nicht für alle JSON-Formate verwendet werden. So kann beispielsweise eine JSON-Objektliste auf der obersten Ebene der JSON-Datei nicht über diese Optionen importiert werden. Sie wurden daher durch die Methode **Eintrag für hierarchische JSON erzeugen** ersetzt.

- 2) Legen Sie im Attributfenster von **Importeintrag**  die folgenden Attribute fest:

- **Name:** Es wird empfohlen, den Standardnamen des Importeintrags in eine aussagekräftige Zeichenfolge zu ändern, anhand derer Sie den Zweck des Importeintrags erkennen können.
 - **Tabelle importieren:** Geben Sie den Namen einschließlich der Dateierweiterung der JSON-Importdatei ein, aus der die Daten importiert werden sollen. Die Dateierweiterung muss ".json" lauten.
 - **Temporäre Datenbanktabelle:** Geben Sie einen Namen für die temporäre Datenbanktabelle ein, in die die Daten importiert werden sollen. Die temporäre Datenbanktabelle wird während des Imports automatisch in der Alfabet-Datenbank -Tabelle erzeugt und mit den externen Daten gefüllt. Beim Erzeugen des Importeintrags wird automatisch ein Standardwert eingegeben. Der Name der temporären Tabelle muss innerhalb des ADIF-Importschemas eindeutig sein.
 - **Kommentare:** Geben Sie eine Beschreibung des Importvorgangs ein, der vom JSON-Importeintrag ausgelöst wird.
 - **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung des Importeintrags zu aktivieren. Wählen Sie `False` aus, um die Ausführung zu deaktivieren.
- 3) Der Ordner **Attribute** des JSON-Importeintrags enthält automatisch die Attributeinträge, die erforderlich sind, um die vordefinierte temporäre Tabellenstruktur für den JSON-Import zu erzeugen. Die Attribute dürfen nicht geändert werden.
- 4) Erzeugen Sie für jede Datenbanktabelle, die mit Daten im JSON-Import aktualisiert werden soll, einen **Importeintrag** , wie im Abschnitt *Definieren des Imports externer Daten in einen Importeintrag* definiert. Die in der temporären Tabelle des JSON-Importeintrags definierten Daten können wie folgt als Eingabe für die temporären Tabellen in anderen Importeinträgen verwendet werden:
- Lassen Sie die Importspalte jedes Attributs, das für den Importeintrag definiert ist, leer.
 - Erstellen Sie einen SQL-Befehl des Typs **OnStart**, der ein `UPDATE` - oder `INSERT INTO`-Statement definiert, um die temporäre Tabelle des aktuellen Eintrags mit den Daten aus der temporären Tabelle zu füllen, die im JSON-Importeintrag definiert ist.



Ausführliche Informationen zu der erforderlichen Konfiguration finden Sie unter *Definieren des Imports externer Daten in einen Importeintrag*. Schauen Sie sich dazu auch das unten stehende Beispiel an.



Das folgende Beispiel beschreibt einen Import von Anwendungsgruppen-Kurzbezeichnungen über eine JSON-Datei mit der folgenden Struktur:

```
[
  {
    "ClassName": "ApplicationGroup",
```

```

    "Values": {
      "name": "OptiRetail CRM",
      "shortname": "OR CRM"
    }
  },
  {
    "ClassName": "ApplicationGroup",
    "Values": {
      "name": "AI CRM AS-IS Situation",
      "shortname": "AI CRM IS"
    }
  }
]

```

Beide im JSON-Beispiel aufgeführten Applikationsgruppen verfügen über keine Kurzbezeichnung in der Alfabet -Datenbank:

	REFSTR	NAME	SHORTNAME
1	95-2-0	OptiRetail CRM	
2	95-21-0	AI CRM AS-IS Situation	

Das ADIF-Importschema enthält zwei Einträge:

- `JSONImport` für den Import der JSON-Datei in die vordefinierte temporäre Tabelle.
- `AppgImport` zur Aktualisierung der Tabelle `APPLICATIONGROUP` der Alfabet-Datenbank.

Der Importeintrag `AppgImport` definiert die temporäre Tabelle und die Zuordnung zu den Objektklasseneigenschaften der Klasse "Applikationsgruppe" über die Einstellungen des Importeintrags und die Definition von Attributen im Ordner **Attribute**. Die Attribute werden wie folgt angezeigt:

Spalte in Importdatei -> Spalte in temporärer Tabelle -> Spalte in der Alfabet -Objektklassentabelle

Die Importspalte in den Attributen ist nicht definiert, da der Import nicht über eine externe Datenquelle durchgeführt wird. Die temporäre Tabelle wird über SQL-Befehle des Typs `OnStart` gefüllt.

Die temporäre Tabelle verfügt über drei Spalten:

- `IDVAL`, um die Daten einem der JSON-Objekte in der JSON-Datei zuzuordnen. Diese Spalte wird nicht in die Alfabet -Objektklassentabelle importiert.
- `APPGNAME`, die als Klassenschlüssel definiert und der Objekteigenschaft `Name` der Klasse `ApplicationGroup` zugeordnet ist. Der ADIF-Importmechanismus verwendet die Daten in dieser Spalte, um die Daten in der temporären Tabelle der richtigen Applikationsgruppe zuzuordnen.
- `APPGSHORTNAME`, die der Objekteigenschaft `ShortName` der Klasse `ApplicationGroup` zugeordnet ist.

Im Ordner **SQL-Befehle - OnStart** sind zwei SQL-Befehle definiert, die nach dem Import der Daten aus der Importdatei und vor der Datenbindung ausgeführt werden, die die temporären Tabellen der Importeinträge Alfabet -Objekten zuordnet (sofern zutreffend).

Importeintrag für den Import von JSON in eine temporäre Tabelle

In den Attributen des Importeintrags wird die Zuordnung zu einer Objektklasse und der Name der temporären Datei festgelegt.

Die Attribut-Definition legt den Spaltennamen der temporären Tabelle sowie die Zuordnung zu einer Objektklasse fest. Eine Importspalte ist nicht definiert.

Die SQL-Befehle dienen dazu, die Daten aus der temporären Tabelle, die für den Importeintrag JSONImport erzeugt wurde, in die temporäre Tabelle des Importeintrags AppgImport zu schreiben

Allgemein	
(Name)	AppgImport
Anwenderdaten	
Assistent	
Ereignisse	(Collection)
Daten	
Historie-Tabelle komprimieren	False
Importspaltenzuordnung	ByName
Importstabelle	0
Parameter lesen	True
Tabelle importieren	
Tabellen-Eigentümer	True
Temporäre Datenbanktabelle	APPGROUPIMPORT_TMP
Eintrag	
Formatierung	
Klasse	
Aktualisierungseigenschaften verwenden	False
Instanzen aktualisieren	True
Instanzen erzeugen	True
Klassename	ApplicationGroup

Allgemein	
Datenbankspalte	APPNAME
Datentyp	String
Eigenschaftname	Name
Ende abschneiden	True
Größe	255
Größenanpassung durch Kürzung	False
Importspalte	
Ist Klassenschlüssel	True
Rückbeziehung	
Rück-Relationen erneut abfragen	True
Sprache	
Start abschneiden	False
Zu Klasse	

Während des Imports werden zunächst alle temporären Tabellen erzeugt und dann aus der JSON-Beispieldatei für den JSON-Importeintrag `JSONImport` importiert. Daraus ergibt sich der folgende Datensatz in der temporären Tabelle `TMP_JSON_IMPORT`:

	OBJID	PARENTOBJID	PARENTPROPNAME	PROPNAME	PROPL...	VA...	V...	VA...	VALUETEXT
1	1			ClassName					ApplicationGroup
2	1			Values					
3	1.1	1	Values	name					OptiRetail CRM
4	1.1	1	Values	shortname					OR CRM
5	2			ClassName					ApplicationGroup
6	2			Values					
7	2.2	2	Values	name					AI CRM AS-IS Situation
8	2.2	2	Values	shortname					AI CRM IS

Anschließend führt der Importmechanismus die SQL-Befehle im Importeintrag `AppgImport` in der angegebenen Reihenfolge aus. Mit dem ersten SQL-Befehl wird die Tabelle `TMP_JSON_IMPORT` nach Zeilen durchsucht, in denen `PROPNAME` dem Wert `name` entspricht, und es werden Werte für `VALUETEXT` und `OBJID` dieser Zeilen in die temporäre Tabelle `APPGROUPIMPORT_TMP` geschrieben:

```
INSERT INTO APPGROUPIMPORT_TMP (APPGNAME, APPGSHORTNAME, IDVAL)
SELECT VALUETEXT, NULL, OBJID
FROM TMP_JSON_SRC
WHERE TMP_JSON_SRC.PROPNAME = 'name'
```

	APPGNAME	APPGSHORTNAME	IDVAL	REFSTR
1	OptiRetail CRM		1.1	
2	AI CRM AS-IS Situation		2.2	

Die Spalte `REFSTR`, die in der temporären Tabelle angezeigt wird, wurde vom ADIF-Importmechanismus hinzugefügt, um die Zuordnung zu Applikationsgruppen in einem späteren Schritt durchzuführen.

Mit dem zweiten SQL-Befehl wird die Tabelle `TMP_JSON_IMPORT` nach Zeilen durchsucht, in denen `PROPNAME` dem Wert `shortname` entspricht, und die vorhandenen Zeilen in der Tabelle `APPGROUPIMPORT_TMP` werden aktualisiert. Dazu werden die Werte in der Spalte `VALUETEXT` in die Spalte `APPGSHORTNAME` geschrieben, und zwar unter Verwendung der Spalten `OBJID` und `IDVAL` in den beiden Tabellen, damit die Daten der richtigen Zeile zugeordnet werden.

```
UPDATE APPGROUPIMPORT_TMP SET APPGROUPIMPORT_TMP.APPGSHORTNAME = TMP_JSON_SRC.VALUETEXT FROM APPGROUPIMPORT_TMP,
TMP_JSON_SRC WHERE TMP_JSON_SRC.OBJID = APPGROUPIMPORT_TMP.IDVAL AND TMP_JSON_SRC.PROPNAME = 'shortname'
```

	APPGNAME	APPGSHORTNAME	IDVAL	REFSTR
1	OptiRetail CRM	OR CRM	1.1	
2	AI CRM AS-IS Situation	AI CRM IS	2.2	

Der ADIF-Importmechanismus ordnet nun die Einträge in der Tabelle den vorhandenen Applikationsgruppen zu und füllt den REFSTR der Applikationsgruppen in der Spalte REFSTR der Tabelle APPGROUPIMPORT_TMP:

	APPGNAME	APPGSHORTNAME	IDVAL	REFSTR
1	OptiRetail CRM	OR CRM	1.1	95-2-0
2	AI CRM AS-IS Situation	AI CRM IS	2.2	95-21-0

Im nächsten Schritt verwendet der ADIF-Importmechanismus die Informationen in den Spalten REFSTR und APPGSHORTNAME der Tabelle APPGROUPIMPORT_TMP, um die Alfabet-Standarddatenbanktabelle APPLICATIONGROUP mit der richtigen Kurzbezeichnung pro Objekt zu aktualisieren:

	REFSTR	NAME	SHORTNAME
1	95-2-0	OptiRetail CRM	OR CRM
2	95-21-0	AI CRM AS-IS Situation	AI CRM IS

Danach werden die temporären Tabellen aus der Datenbank entfernt.

Definieren des Datenimports aus Microsoft® Excel-Dateien und kommagetrennten Dateiformaten

Beim Import von Daten aus CSV-, TXT-, XLSX- oder XLS-Dateien werden die Daten aus der Importdatei direkt in die temporäre Tabelle in der Alfabet-Datenbank geschrieben. Die Daten in der Importdatei sollten einer Datenbanktabellenstruktur entsprechen:

- **Interpretation von kommagetrennten Dateiformaten**

Die CSV-Datei wird als Darstellung einer Datenbanktabelle angesehen, bei der die Spaltenüberschriften der Datenbank in der ersten Zeile der Datei angegeben sind. Die folgenden Zeilen in der Datei stellen die Datenzeilen in der Datenbanktabelle dar.

- **Interpretation von Excel®-Dateiformaten**

Daten werden nur aus dem ersten Arbeitsblatt der Datei gelesen. Die Daten in der Excel®-Datei müssen eine flache Tabellenstruktur aufweisen. Die Spaltenüberschriften müssen in der ersten Zeile angegeben sein. Die folgenden Zeilen in der Tabelle stellen die Datenzeilen in der Datenbanktabelle dar.

Beim Import aus Microsoft® Excel®-Dateien oder kommagetrennten Dateiformaten können die Importeinträge, die festlegen, wie die Daten beim Import verarbeitet werden, direkt im Importschema definiert werden. Dennoch ist es möglich, eine Gruppierung innerhalb von Datei-Importgruppen zu definieren. Die Datei-Importgruppe  ist wie ein Ordnerknoten im ADIF-Explorer. Die einzige Funktionalität, die der Ordner bietet, ist die Batch-Aktivierung bzw. -Deaktivierung aller Importeinträge innerhalb der Datei-Importgruppe.

So erzeugen Sie eine Datei-Importgruppe:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Importschema, und wählen Sie **Datei-Importgruppe erzeugen** aus. Dem Explorer wird eine neue **Datei-Importgruppe**  mit einem Standardnamen für die Datei-Importgruppe hinzugefügt. Unterhalb der **Datei-Importgruppe**  wird automatisch ein Ordner für die Erzeugung von SQL-Befehlen hinzugefügt, die bei Aktivierung der Datei-Importgruppe ausgeführt werden sollen.
- 2) Geben Sie im Attributfenster der Datei-Importgruppe  die folgenden Attribute an, um die Verbindung zu der externen Datenbank zu konfigurieren:
 - **Name:** Geben Sie einen aussagekräftigen Namen für die Datei-Importgruppe ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit der Datei-Importgruppe implementiert wird.
 - **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung aller Importeinträge innerhalb der Importgruppe auszuwählen. Wählen Sie `False` aus, um die Ausführung zu deaktivieren.



Nach der Erzeugung der Datei-Importgruppe müssen Sie nun Importeinträge hinzufügen, die die Daten in externen Dateien den Daten in der Alfabet-Datenbank zuordnen.

Weitere Informationen hierzu finden Sie unter *Definieren des Imports externer Daten in einen Importeintrag*.

Optional können Sie SQL-Befehle definieren, mit denen Sie diesen Teil des Importschemas deaktivieren können, wenn bestimmte Voraussetzungen erfüllt sind. Weiterführende Informationen finden Sie unter *Konfigurieren der bedingten Ausführung von Teilen des Importschemas*.

Definieren des Imports externer Daten in einen Importeintrag

Ein Importeintrag definiert den Import von Daten in die Alfabet-Datenbank für eine einzelne Objektklasse aus einer einzelnen Quelle. Sie können mehrere Importeinträge für eine einzelne Importdatei oder Datenbanktabelle definieren, um mehrere Klassen aus einer Quelle zu importieren. Auch wenn die Dateien alle den gleichen Dateityp haben, müssen Sie mehrere Importeinträge definieren, um Daten für eine Klasse aus mehreren Datenbanktabellen oder Dateien zu importieren.

Wenn Sie Daten aus Microsoft® Excel®-Dateien oder kommagetrennten Dateiformaten importieren, wird jede Importdatei als Datenbanktabelle betrachtet. Daher müssen die Daten in der Datei der Struktur der Datenbanktabelle entsprechen.

- **Interpretation von kommagetrennten Dateiformaten**

Die CSV-Datei wird als Darstellung einer Datenbanktabelle angesehen, bei der die Spaltenüberschriften der Datenbank in der ersten Zeile der Datei angegeben sind. Die folgenden Zeilen in der Datei stellen die Datenzeilen in der Datenbanktabelle dar.

- **Interpretation von Excel®-Dateiformaten**

Daten werden nur aus dem ersten Arbeitsblatt der Datei gelesen. Die Daten in der Excel®-Datei müssen eine flache Tabellenstruktur aufweisen. Die Spaltenüberschriften müssen in der ersten Zeile angegeben sein. Die folgenden Zeilen in der Tabelle stellen die Datenzeilen in der Datenbanktabelle dar.

- **Interpretation externer Datenbanktabellen**

Der als Importdaten verwendete Datensatz wird nicht direkt aus den Datenbanktabellen der externen Datenbank gelesen. In dem ADIF-Importeintrag, mit dem der Import definiert wird, muss eine SQL-Abfrage mit einer `SELECT`-Anweisung als SQL-Befehl für das Hochladen von Daten definiert sein. Der aus der Abfrage resultierende Datensatz wird von ADIF als Importdaten hochgeladen. Die Spaltenüberschriften der Importtabelle werden von den Alias-Spezifikationen in der `SELECT`-Anweisung definiert.

Weitere Informationen über das Hochladen von Daten aus einer externen Datenbank und die Konfiguration, die erforderlich ist, bevor die unten beschriebene Konfiguration ausgeführt werden kann, finden Sie unter *Definieren des Hochladens von Daten aus einer externen Datenbank in eine Datenbank-Importgruppe*.

- **Interpretation von LDAP-Tabellen**

Der als Importdaten verwendete Datensatz wird nicht direkt aus der Datenbank mit den LDAP-Tabellen gelesen. In dem ADIF-Importeintrag, mit dem der Import definiert wird, muss eine Abfrage in einer LDAP-spezifischen Abfragesyntax geschrieben und als SQL-Befehl für das Hochladen von Daten definiert sein. Der aus der Abfrage resultierende Datensatz wird von ADIF als Importdaten hochgeladen. Die Spaltenüberschriften der Importtabelle werden von den Kopfzeilendefinitionen des Datensatzes in der Abfrage definiert. Weitere Informationen über das Hochladen von Daten aus LDAP und die Konfiguration, die erforderlich ist, bevor die in diesem Abschnitt beschriebene Konfiguration ausgeführt werden kann, finden Sie unter *Definieren des Hochladens von Daten aus einer LDAP-Tabelle in eine LDAP-Importgruppe*.

- **Interpretation von XML-Dateien**

Jedes XML-Element, das XML-Attribute und/oder untergeordnete XML-Elemente mit Textinhalt aber ohne eigene XML-Attribute enthält, wird als Datenbanktabelle betrachtet. Der Name der Datenbanktabelle ist der Name des XML-Elements. Die Namen der Datenbankspalten sind die Namen der XML-Attribute des XML-Elements sowie die Namen der untergeordneten Elemente ohne eigene XML-Attribute, aber mit Textinhalt.

Für jedes XML-Element wird während des Imports automatisch eine ID erzeugt, um Informationen über die Hierarchie von XML-Elementen zu speichern, die in der Importdatei in den temporären Tabellen verfügbar sind, die während des ADIF-Imports generiert werden. Jeder temporären Tabelle muss eine Datenbankspalte hinzugefügt werden, in der die ID des XML-Elements gespeichert wird; außerdem muss der temporären Tabelle für jedes XML-Element, das über ein übergeordnetes XML-Element verfügt, eine Datenbankspalte hinzugefügt werden, in der die ID des übergeordneten XML-Elements gespeichert wird.

Ein Importeintrag löst den Import in zwei Schritten aus:

- 1) Externe Daten werden in eine temporäre Tabelle geschrieben
- 2) Daten aus der temporären Tabelle werden in die Alfabet-Datenbank -Tabelle geschrieben.

Die für den Aufbau der temporären Tabelle erforderlichen Informationen und die in der Alfabet-Datenbank -Tabelle vorzunehmenden Änderungen müssen in den untergeordneten Elementen des **Import-Eintrags**  angegeben werden. Die temporäre Tabelle muss nicht unbedingt eine exakte Kopie der Daten sein, wie sie in der Importdatenbank oder -datei bereitgestellt werden. Sie können der Tabelle zusätzliche Spalten hinzufügen, die entweder leer bleiben oder mit einem statischen Wert oder einem Wert, der mit Hilfe einer SQL-Abfrage angegeben wird, befüllt werden. Daten können aus den anfänglichen temporären Datenbanktabellen ausgewählt werden, um neue temporäre Tabellen zu erzeugen.

Als letzten Schritt konfigurieren Sie, welche temporären Tabellen verwendet werden, um die Daten in der Alfabet-Datenbank -Tabelle der Zielobjektklasse zu ändern.

Die Konfiguration des Imports in die Alfabet-Datenbank kann ein komplexer Prozess sein, der die Definition verschiedener untergeordneter Elemente des Importeintrags und der während des Prozesses auszuführenden SQL-Befehle umfasst. In den folgenden Abschnitten finden Sie grundlegende Informationen zur Konfiguration eines Importeintrags und seiner untergeordneten Elemente im Zusammenhang mit grundlegenden Aufgaben, die am häufigsten Teil eines Importvorgangs sind. Bei der Definition von Importeinträgen müssen Sie die verschiedenen möglichen Konfigurationsschritte kombinieren und an Ihre individuellen Anforderungen anpassen.

Beachten Sie unbedingt die Ausführungsreihenfolge der Importeinträge im ADIF-Importschema, wenn Sie den Import konfigurieren. Beim Importvorgang werden alle Importeinträge mehrmals gelesen, um die folgenden aufeinanderfolgenden Aktionen durchzuführen:

- 1) Erzeugen Sie alle temporären Tabellen, die in allen Importeinträgen definiert sind, in der Reihenfolge der Definitionen des **Importeintrags**  im ADIF-Importschema.
- 2) Schreiben von Datensätzen aus den externen Dateien/Datenbanktabellen in die temporären Tabellen in der Reihenfolge der Importeintragsdefinition im ADIF-Importschema.
- 3) Führen Sie alle in SQL-Befehlen  vom Typ `OnStart` definierten DML-Anweisungen aus.
- 4) Aktualisieren Sie die Alfabet -Objektklassen-Datenbanktabellen mit Daten aus den temporären Tabellen entsprechend der Angabe in den Importeinträgen in der Reihenfolge der Definition der Importeinträge im ADIF-Importschema.
- 5) Aktualisieren Sie die RELATIONS-Tabelle der Alfabet-Datenbank, wenn Beziehungsaktualisierungen in Importeinträgen in der Reihenfolge der Definition von Importeinträgen im ADIF-Importschema definiert sind.

- 6) Führen Sie alle in SQL-Befehlen  vom Typ `OnComplete` definierten DML-Anweisungen aus.
- 7) Entfernen aller temporären Tabellen, wenn das ADIF-Importschema zum Entfernen temporärer Tabellen konfiguriert ist.



Das ADIF-Importschema ermöglicht die Spezifikation zusätzlicher SQL-Befehle, die zu verschiedenen Zeitpunkten während des Importvorgangs ausgeführt werden sollen. Weitere Informationen über die Ausführungsreihenfolge dieser SQL-Befehle finden Sie im Abschnitt *Konfigurieren von SQL-Befehlen für optionale erweiterte Importfunktionen*.

Diese Ausführungsreihenfolge lässt z. B. zu, dass Daten in allen temporären Tabellen gesucht und verarbeitet werden, die in den SQL-Befehlen eines Importeintrags für andere Importeinträge erzeugt wurden.

Zuordnen von externen Daten zu temporären Tabellen und Alfabet -Datenbanktabellen

Bevor Sie Ihren Import konfigurieren, müssen Sie zunächst auswerten, wie die externen Daten dem Alfabet -Metamodell und den Alfabet-Datenbank -Tabellen zugeordnet werden.

- *In welche Zielklasse werden meine Daten importiert?*

Eine Importdatei kann Daten für den Import in mehrere Klassen enthalten. Sie müssen dann mehrere Importeinträge für den Import der gleichen Daten definieren. Jeder Importeintrag definiert eine temporäre Tabelle, die die für den Import relevante Teilmenge von Daten in eine Objektklasse des Alfabet -Metamodells importiert.

- *Verfügt die Zielklasse über die erforderlichen Eigenschaften, um die Daten zu speichern, die in die Klasse importiert werden sollen?*

Wenn die Importdaten eine Eigenschaft enthalten, die in den Standardobjektklasseneigenschaften der Zielobjektklasse im Alfabet -Metamodell nicht vorhanden ist, können im Alfabet -Metamodell neue Eigenschaften für die Zielobjektklasse erzeugt werden.



Benutzerdefinierte Eigenschaften müssen im Konfigurationswerkzeug Alfabet Expand konfiguriert werden. Informationen über das Erzeugen einer benutzerdefinierten Eigenschaft finden Sie im Abschnitt *Konfigurieren benutzerdefinierter Eigenschaften für geschützte oder öffentliche Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.



Sie müssen die Konfigurationsfunktionen in Alfabet Expand verwenden, um einer Datenbanktabelle einer Alfabet -Objektklasse neue Eigenschaftsspalten hinzuzufügen. Eigenschaftsspalten dürfen nicht mit einer DDL-Anweisung in den SQL-Befehlen des ADIF-Schemas hinzugefügt werden. **Bestehende Alfabet-Datenbank -Standardtabellen dürfen nicht durch SQL-Befehle über ADIF geändert werden.**

- *Welche Datenformate und -typen sind für die Importdaten erforderlich?*

Überprüfen Sie, ob die Importdaten mit dem Datenformat der Spalte der Zieldatenbanktabelle in der Alfabet-Datenbank übereinstimmen. Die Datentypen und -formate werden in der Tabelle [Datentypen und Formate](#) im Kapitel [Das Alfabet -Metamodell in der Alfabet -Datenbank](#) beschrieben.

Das ADIF-Importschema bietet Mechanismen zur automatischen Anpassung von Datums-, Zahlen- und booleschen Werten an die in der Alfabet-Datenbank erforderliche Formatierung. Sie können Informationen über das in Ihren externen Daten verwendete Datenformat für Datumsangaben, Zahlen und boolesche Ausdrücke angeben. Diese Werte werden dann beim Import automatisch in das richtige Format umgewandelt.

Weitere Informationen hierzu finden Sie unter *Definieren des Imports externer Daten in einen Importeintrag*.

Das ADIF-Importschema lässt auch zu, dass der temporären Tabelle leere Spalten hinzugefügt werden. Diese werden mithilfe von SQL-Befehlen mit Werten gefüllt, nachdem die externen Daten in die temporäre Tabelle geschrieben wurden. Sie können auch SQL-Befehle definieren, um Daten zu ändern, die in die temporäre Datenbanktabelle geschrieben wurden. Mithilfe dieses Mechanismus können Sie Importdaten in ein beliebiges Datenformat konvertieren oder Standardwerte festlegen, wenn die Importdaten keine konsistenten Informationen für eine Eigenschaft bereitstellen.

Weitere Informationen hierzu finden Sie unter *Anwendungsfall: Ausfüllen leerer Eigenschaftswerte mit Standardwerten beim Importieren in Pflicht-Objektklasseneigenschaften*.



Leere Zeichenfolgen können nicht in die Alfabet-Datenbank importiert werden. Wenn für einen Eigenschaftswert eine leere Zeichenfolge importiert wird, wird der Eigenschaftswert in der Alfabet-Datenbank auf NULL gesetzt. Wenn Sie nicht möchten, dass für die Eigenschaft NULL festgelegt wird, können Sie vor dem Import in Alfabet-Datenbank -Tabellen mit SQL-Befehlen anstelle von leeren Zeichenfolgen in den temporären Tabellen Standardwerte festlegen.

- *Stellen die Importdaten alle obligatorischen Informationen bereit, wenn während des Imports neue Objekte erzeugt werden?*

Einige Objektklasseneigenschaften im Alfabet -Metamodell sind obligatorisch. Wenn eine solche Eigenschaft beim Importieren fehlt, könnte Ihre Alfabet-Datenbank Schaden nehmen. Bitte beachten Sie, dass obligatorische Eigenschaften im Metamodell-Browser in der ADIF-Konfigurationsoberfläche gelb hervorgehoben sind. Sie müssen sicherstellen, dass die Daten für alle obligatorischen Eigenschaften bereitgestellt werden.

Abgesehen von den als obligatorisch markierten Objektklasseneigenschaften können aufgrund Ihrer spezifischen Anforderungen weitere Eigenschaften obligatorisch sein oder müssen festgelegt werden, damit die Anwender das importierte Objekt finden können. Sie können beispielsweise ein Objekt ohne Angabe eines autorisierten Anwenders, einer Zuweisung oder einer anderen Definition importieren, mit der Zugriffsberechtigungen für Anwender begründet werden. Dadurch verfügt keiner der Anwender über Bearbeitungsberechtigungen für das importierte Objekt.

Werden bei einigen Objekten keine Werte für Pflichteigenschaften bereitgestellt, können Sie den Importeintrag mithilfe von SQL-Befehlen konfigurieren, um vor dem Datenimport Eigenschaftswerte in der temporären Tabelle festzulegen. Weitere Informationen hierzu finden Sie unter *Anwendungsfall: Ausfüllen leerer Eigenschaftswerte mit Standardwerten beim Importieren in Pflicht-Objektklasseneigenschaften*.

- *Ist die Datenübersetzung für die Ziel- Alfabet-Datenbank aktiviert?*

Die Datenübersetzungsfunktionalität der Alfabet-Datenbank ermöglicht die Übersetzung von Objektnamen und -beschreibungen. Wenn die Funktionalität aktiviert ist, werden in den Datenbanktabellen für jede Sprache, für die die Datenübersetzung aktiviert ist, zwei neue Spalten mit den Namen `NAME_ <Sprachcode>` und `DESCRIPTION_ <Sprachcode>` hinzugefügt. Die übersetzten Werte müssen über ADIF zusammen mit den Originalwerten importiert werden. Im ADIF-Importschema

beinhaltet die Zuordnung der Daten zu den Eigenschaften der Zielobjektklasse eine Sprachspezifikation. Für jede importierte Sprache muss eine Datenzuordnung definiert werden.



Ist die Eigenschaft "Name" einer Zielklasse beispielsweise ins Deutsche übersetzt, müssen sowohl der ursprüngliche als auch der übersetzte Wert der Eigenschaft "Name" der Zielklasse zugeordnet werden. Die ursprünglichen Werte werden ohne Sprachspezifikation zugeordnet. Die deutschen Werte werden mit einer deutschen Sprachspezifikation zugeordnet.

Weitere Informationen hierzu finden Sie unter *Zuordnen von externen Daten zu temporären Tabellen und Alfabet -Datenbanktabellen*.



Weitere Informationen zum Konfigurieren der Objektdatenübersetzung finden Sie im Abschnitt *Konfigurieren der Übersetzung von Objektdaten* im Kapitel *Lokalisierung und Mehrsprachenunterstützung für die Alfabet -Benutzeroberfläche*.

- *Wie können die Datensätze vorhandenen Objekten in der Zieldatenbanktabelle zugeordnet werden?*

Das Alfabet -Metamodell speichert Daten über jedes Objekt in der Alfabet-Datenbank mit einer Reihe von eindeutigen Bezeichnern, die automatisch festgelegt werden, wenn ein neues Objekt erzeugt wird. Beim Datenimport werden diese eindeutigen Bezeichner automatisch zu neuen Datensätzen hinzugefügt, mit Ausnahme der eindeutigen Eigenschaft `REFSTR`. Diese Eigenschaft wird verwendet, um die importierten Datensätze bereits in der Alfabet-Datenbank vorhandenen Datensätzen zuzuordnen. Deshalb wird die Eigenschaft `REFSTR` der Objekte beim Import über einen speziellen Datenbindungsmechanismus festgelegt, der die Zuordnung zu bestehenden Daten berücksichtigt. Die Eigenschaft `REFSTR` von Objekten wird ausschließlich über diesen Mechanismus festgelegt. Daher muss auch beim erstmaligen Datenimport in leere Datenbanktabellen eine Datenbindung durchgeführt werden.

Der Datenbindungsmechanismus erfordert, dass jedes Objekt in den externen Daten über einen eindeutigen Bezeichner verfügt, der in den Import einbezogen wird. Werden Daten in Datenbanktabellen importiert, die bereits Objekte enthalten, muss dieser Bezeichner eine eindeutige Eigenschaft der Zielobjektklasse sein, anhand der erkannt werden kann, ob das importierte Objekt mit einem vorhandenen Objekt identisch ist.

Im folgenden Abschnitt werden die Mechanismen beschrieben, die von ADIF für den sicheren Import von Daten bereitgestellt werden. Außerdem können Sie Ihre eigenen SQL-Befehle verwenden, um dem Basisimport komplexe Funktionen hinzuzufügen.

Informationen über zusätzliche SQL-Befehle finden Sie unter *Konfigurieren von SQL-Befehlen für optionale erweiterte Importfunktionen*.



Die ADIF-Importmechanismen sind darauf ausgelegt, ein hohes Maß an Sicherheit zu gewährleisten. Überprüfen Sie immer, ob ein Import mithilfe eines von ADIF bereitgestellten Importmechanismus durchgeführt werden kann, bevor Sie Ihre eigenen SQL-Befehle zum Durchführen des Imports definieren.

Erzeugen eines Importeintrags zum Zuordnen der Datenbanktabellen

Sie müssen mindestens einen Importeintrag pro externe Datei oder externe Datenbanktabelle in das ADIF-Importschema aufnehmen. Ein Importeintrag definiert die Zuordnung zwischen einer externen Datei oder externen Datenbanktabelle und einer Zieldatenbanktabelle des Alfabet -Metamodells.

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Importschema oder die Importgruppe, für das/die Sie einen Importeintrag erstellen möchten, und wählen Sie **Eintrag erzeugen** aus.
- 2) Legen Sie im Attributfenster von **Importeintrag**  die folgenden Attribute fest:
 - **Name:** Es wird empfohlen, den Standardnamen des Importeintrags in eine aussagekräftige Zeichenfolge zu ändern, anhand derer Sie den Zweck des Importeintrags erkennen können.
 - **Kommentare:** Geben Sie eine Beschreibung des Importvorgangs ein, der von der Datenbank-Importgruppe konfiguriert wurde.
 - **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung des Importeintrags zu aktivieren. Wählen Sie `False` aus, um die Ausführung zu deaktivieren.
 - **Historie-Tabelle komprimieren:** Wählen Sie `True` aus, um bedeutungslose Einträge aus der Historie-Tabelle zu entfernen, die aus technischen Gründen während des Imports erzeugt wurden.

Weitere Informationen über das Komprimieren von Historie-Tabellen finden Sie unter *Bereinigen der Überwachungstabellen*.

- 3) Definieren Sie im Attributfenster den Import in die temporäre Datenbanktabelle:
 - **Tabelle importieren:** Legen Sie dieses Attribut gemäß der Anforderungen Ihres Importformats fest:
 - Für den Import aus einer XLSX-, XLS-, JSON-, CSV- oder TXT-Datei: Geben Sie den Namen einschließlich der Dateierweiterung jener Datei ein, aus der der Import durchgeführt werden soll.
 - Beim Importieren aus einer XML-Datei: Geben Sie den Namen des XML-Elements in die XML-Importdatei ein, aus der die Daten importiert werden sollen.
 - Beim Importieren aus einer externen Datenbank oder aus LDAP: Dieses Attribut wird ignoriert.
 - **Temporäre Datenbank-Tabelle:** Geben Sie einen Namen für die temporäre Datenbanktabelle ein, in die die Daten importiert werden sollen. Die temporäre Datenbanktabelle wird während des Imports automatisch in der Alfabet-Datenbank -Tabelle erzeugt und mit den externen Daten gefüllt.



Beachten Sie Folgendes:

- Die Kurzbezeichnung muss eindeutig sein. **Sie dürfen nicht den Namen einer vorhandenen Alfabet-Datenbank -Tabelle verwenden!** Es wird empfohlen, die Namen aller temporären Tabellen mit dem Präfix „TMP_“ zu beginnen, um zu gewährleisten, dass es nicht zu Konflikten mit der vorhandenen Datenbank-Konfiguration kommt.

- Für den Import aus JSON ist der Tabellename vorkonfiguriert und muss `TMP_JSON_SRC` lauten.
- **Tabellen-Eigentümer** Wählen Sie `True`, um die automatische Erzeugung der temporären Tabelle und die Ausführung der Datensatzzuordnung mit Daten in der Alfabet-Datenbank zu aktivieren.
-  Die Festlegung dieses Attributs auf `False`, um die Verarbeitung temporärer Tabellen zu deaktivieren, ist nur relevant, wenn der Datenimport aus mehreren externen Dateien/Datenbanktabellen in eine einzelne temporäre Tabelle konfiguriert wird.
- Weitere Informationen hierzu finden Sie unter *Sammeln von Daten aus mehreren externen Dateien oder Datenbanktabellen in einer temporären Tabelle*.
- 4) Definieren Sie die Datenformate in den Importdaten, um eine automatische Konvertierung in die in der Alfabet-Datenbank benötigten Formate zu ermöglichen:
- **Boolean 'False' (Falsch):** Geben Sie denjenigen Wert in Ihre externen Daten ein, der beim Import als boolescher Wert `False` interpretiert werden soll. Wenn Sie z. B. „nein“ definieren und das ADIF-Schema so konfiguriert ist, dass eine Eigenschaft als boolesch importiert wird, wird der Wert „nein“ beim Import automatisch in das boolesche Format `False` konvertiert, das im Alfabet -Metamodell verwendet wird.
 - **Boolean 'True' (Wahr):** Geben Sie denjenigen Wert in Ihre externen Daten ein, der beim Import als boolescher Wert `True` interpretiert werden soll. Wenn Sie z. B. „ja“ definieren und das ADIF-Schema so konfiguriert ist, dass eine Eigenschaft als boolesch importiert wird, wird der Wert „ja“ beim Import automatisch in das boolesche Format `True` konvertiert, das im Alfabet -Metamodell verwendet wird.
 - **Datumsformat:** Geben Sie das Datumsformat ein, das in Ihren externen Daten verwendet wird.
 - **Anzahl der Dezimalziffern:** Geben Sie die Anzahl der Dezimalziffern ein, die in Ihren Importdaten verwendet wird.
 - **Anzahl der Dezimaltrennzeichen:** Geben Sie das Dezimaltrennzeichen ein, das bei Zahlen in Ihren Importdaten verwendet wird.
 - **Anzahl der Gruppentrennzeichen:** Geben Sie das Gruppentrennzeichen ein, das bei Zahlen in Ihren Importdaten verwendet wird.
 - **Verschlüsselung:** Nur für den Import von kommasetrennten Dateien: Wählen Sie im Dropdown-Listefeld die Verschlüsselung der Importdatei aus.
 - **Trennzeichen:** Nur für den Import von kommasetrennten Dateien: Legen Sie das Trennzeichen fest, mit dem in der Datei Daten innerhalb eines Datensatzes voneinander getrennt werden.
- 5) Definieren Sie im Attributfenster die Zuordnung der Datensätze zur Alfabet-Datenbank -Zieltabelle:



Die folgenden Spezifikationen sind optional. Sie können einen Importeintrag nur für den Import in eine temporäre Tabelle definieren (z. B. wenn die Daten ausschließlich in SQL-Befehlen anderer Importeinträge verwendet werden sollen). Wenn Daten aus der temporären Datenbanktabelle in eine Alfabet-Datenbank -Zieltabelle importiert werden, sind diese Angaben erforderlich.

- **Klassenname:** Wählen Sie die Zielobjektklasse des Alfabet -Metamodells aus.
- **Instanzen erzeugen:** Wählen Sie `True`, um zu erlauben, dass während des durch diesen Importeintrag ausgelösten Imports den Alfabet-Datenbank -Tabellen neue Datensätze hinzugefügt werden. Wählen Sie `False`, um den Import auf die Manipulation vorhandener Datensätze zu beschränken.
- **Instanzen aktualisieren** Wählen Sie `True`, um zu erlauben, dass bestehende Daten in den Alfabet-Datenbank -Tabellen während des Imports durch diesen Importeintrag geändert werden können.

Erstellen von Attributeinträgen zum Zuordnen der Datenbanktabellenspalten

Die Spalten der temporären Tabelle und die Zuordnung der temporären Tabellenspalten sowohl zu den Importdaten als auch zu den Zielspalten in der Alfabet-Datenbank werden im Attribut „Untergeordnete Elemente“ des **Importeintrags**  definiert. Für jede Spalte der temporären Tabelle muss ein Element „Attribut“  definiert werden:



Bei der Angabe der Spaltennamen in der Datenbank wird die Groß-/Kleinschreibung beachtet.

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsoberfläche mit der rechten Maustaste auf den **Importeintrag** , für den Sie ein Element **Attribut** erzeugen möchten, und wählen Sie **Attribut erzeugen**.
- 2) Legen Sie im Attributfenster des Elements **Attribut**  Folgendes fest, um eine Spalte in der temporären Tabelle zu definieren:
 - **Datenbankspalte:** Geben Sie einen Namen für die Datenbankspalte der temporären Tabelle ein.
 - **Spalte importieren:** Geben Sie den Namen der Spalte der externen Datentabelle ein, aus der Daten geschrieben werden sollen. Die Definition dieses Attributs ist optional. Sie können Spalten in der temporären Tabelle definieren, die nicht mit externen Daten ausgefüllt werden.
 - **Eigenschaftsname:** Wählen Sie in der Dropdown-Liste die Eigenschaft der Zielobjektklasse aus, für die Sie Daten aktualisieren möchten. Das Attribut **Name** der Objektklasseneigenschaft aus dem Alfabet -Metamodell muss zur Festlegung der Objektklasse verwendet werden. Dieses Attribut ist nicht unbedingt identisch mit dem Namen der Datenbanktabellenspalte. Die Definition dieses Attributs ist optional. Sie können in der temporären Tabelle Spalten definieren, die keiner Alfabet-Datenbank -Tabellenspalte zugeordnet sind.



Die Spezifikation der Zielobjektklasseneigenschaft über das Attribut **Name** der Objektklasseneigenschaft beschränkt den Datenimport auf Spalten in der Alfabet-Datenbank -Zieltabelle, die Eigenschaftsdefinitionen enthalten. Wenn Sie Daten in Datenbanktabellenspalten aktualisieren möchten, die durch Datenbank-Auslöser in der Alfabet-Datenbank erzeugt werden (z. B. die Datenbankspalten für die Übersetzung von Objektnamen und -beschreibungen in definierte Sprachen), müssen Sie die Aktualisierung über einen SQL-Befehl im Ordner **SQL-Befehle - OnComplete** definieren.

Weitere Informationen hierzu finden Sie unter *Konfigurieren von SQL-Befehlen für optionale erweiterte Importfunktionen*.

- **Ist Schlüssel** Wählen Sie `True`, wenn diese Eigenschaft verwendet werden soll, um beim Import Datensätze vorhandenen Datensätzen in der Alfabet-Datenbank zuzuordnen. Die automatische Zuordnung der Daten wird nur durchgeführt, wenn für mindestens ein Attributelement des Importeintrags ein Attribut **Ist Schlüssel** auf `True` festgelegt ist.
 - Informationen über das Zuordnen von Objekten finden Sie unter *Definieren der Zuordnung von Datensätzen*.
- 3) Während des Imports müssen die Daten in einem Format in die temporäre Tabelle geschrieben werden, das den Anforderungen der Tabellenspalte in der Zieldatenbank der Alfabet-Datenbank entspricht. Die meisten der folgenden Attribute werden automatisch auf die erforderlichen Werte eingestellt, wenn Sie das Attribut **Eigenschaftsname** festlegen. Sie können sie an Ihre spezifischen Anforderungen anpassen:
- **Datentypen:** Wählen Sie in der Dropdown-Liste den Datentyp der Zielobjektklasseneigenschaft in der Alfabet-Datenbank aus.
 - **Größe:** Wenn `String` im Feld **Datentyp** ausgewählt ist, legen Sie die Größe der Zeichenfolge-Spalte in der temporären Datenbanktabelle fest. Die Größenspezifikation sollte mit der Größenspezifikation der Zielobjektklasseneigenschaft in der Alfabet-Datenbank übereinstimmen.
-  Das Attribut **Größe** kann nur dann manuell festgelegt werden, wenn die Spalte der temporären Tabelle keiner Alfabet-Datenbank -Spalte zugeordnet ist (durch Festlegen des Attributs **Eigenschaftsname**). Wenn **Eigenschaftsname** angegeben ist, wird der Wert **Größe** automatisch auf die gewünschte Größe gesetzt und bei manueller Änderung auf den definierten Wert zurückgesetzt.
- Aus technischen Gründen muss der Wert **Größe** auf `<Größe in der Zieldatenbankta-
belle>+1` gesetzt sein. Wenn Sie z. B. einen Wert mit der zulässigen Größe von 16 importieren möchten, müssen Sie das Attribut **Größe** mit 17 definieren.
- **Größenanpassung durch Kürzung:** Nur für Eigenschaften des **Datentyps** `String`. Wählen Sie `True` aus, um importierte Zeichenfolgen abzuschneiden, die größer sind als in der Spezifikation **Größe** angegeben. Wählen Sie `False` aus, um den Import mit einer Ausnahme abzubrechen, wenn die Importdaten nicht der angegebenen Größe entsprechen.
 - **Ende abschneiden:** Wenn dieses Attribut auf `True` gesetzt ist, werden Leerzeichen am Ende der importierten Zeichenfolgen entfernt, wenn Sie die Daten in die Temporärtabelle importieren. Bitte beachten Sie, dass geschützte Leerzeichen durch diesen Mechanismus nicht entfernt werden. Standardmäßig ist für das Attribut `False` ausgewählt.
 - **Start abschneiden:** Wenn dieses Attribut auf `True` gesetzt ist, werden Leerzeichen am Anfang der importierten Zeichenfolgen entfernt, wenn Sie die Daten in die Temporärtabelle importieren. Bitte beachten Sie, dass geschützte Leerzeichen durch diesen Mechanismus nicht entfernt werden. Standardmäßig ist für das Attribut `False` ausgewählt.
- 4) Wenn die Zieleigenschaft eine Referenz zu einer anderen Objektklasse herstellt und das Alfabet - Metamodell von der referenzierten Objektklasse auf die referenzierende Objektklasse zurückverweist, wird diese Rückreferenz beim Import automatisch aktualisiert, sofern die folgenden Attribute definiert sind:
- **Rück-Relationen erneut abfragen:** Wählen Sie `True` aus, um Rückbeziehungen automatisch festzulegen.



Setzen Sie **Rück-Relationen erneut abfragen** immer auf `True`, wenn Sie in eine Objektklasseneigenschaft des Typs `Reference` importieren, die für einen Attributenamen das Attribut **BackReference** definiert. Andernfalls führt Ihr Import zu Inkonsistenzen in der Datenbank.

- **Klasse:** Wählen Sie in der Dropdown-Liste die Zielobjektklasse in der Alfabet-Datenbank aus, für die die Rückreferenz festgelegt werden soll. Das Attribut **Name** der Objektklasse aus dem Alfabet -Metamodell muss zur Festlegung der Objektklasse verwendet werden. Dieses Attribut ist nicht unbedingt identisch mit dem Namen der Datenbanktabelle. Die Definition dieses Attributs ist optional. Wenn die Zielklasse der Referenz im Attribut **TypeInfo** der diesem Attribut-Element zugeordneten Eigenschaft definiert ist, ist es nicht erforderlich, dieses Attribut festzulegen.
 - **Rückbeziehung:** Geben Sie den Namen der Zielobjektklasseneigenschaft ein, die den Rückverweis herstellt. Die Definition dieses Attributs ist optional. Ist die Eigenschaft, welche den Rückverweis herstellt, bereits im Attribut **BackReference** der Eigenschaft angegeben, die diesem Attributelement zugeordnet ist, muss dieses Attribut nicht festgelegt werden.
- 5) Handelt es sich bei den importierten Daten um einen übersetzten Wert für eine Eigenschaft "Name" oder "Beschreibung" der Zielobjektklasse, muss die Zielsprache mit dem folgenden Attribut definiert werden:
- **Sprache:** Wählen Sie im Dropdown-Listenfeld die Landeseinstellung aus, für die die Werte importiert werden.



Das Attribut **Sprache** soll nicht für Werte eingestellt werden, die für die ursprünglichen, standardmäßigen Landeseinstellungen importiert werden.

Weitere Informationen über die Speicherung von Objekt-Eigenschaftsübersetzungen in der Alfabet-Datenbank finden Sie unter [Speicherung von übersetzbaren Objektklasseneigenschaften](#) im Kapitel [Das Alfabet -Metamodell in der Alfabet -Datenbank](#).

Definieren der Zuordnung von Datensätzen

Die eindeutige Eigenschaft `REFSTR` von Objekten in der Alfabet-Datenbank wird verwendet, um Objekte vorhandenen Objekten in der Alfabet-Datenbank zuzuordnen.

Das ADIF-Importschema bietet einen Datenbindungsmechanismus, der die Eigenschaft `REFSTR` zu den externen Daten in den temporären Datenbanktabellen hinzufügt, wobei die Zuordnung von Objekten zu bestehenden Datensätzen in der Alfabet-Datenbank berücksichtigt wird. Für bestehende Objekte wird die für das Objekt in der Tabelle Alfabet-Datenbank konfigurierte `REFSTR` in die temporäre Tabelle geschrieben, und für neue Objekte wird eine eindeutige `REFSTR` generiert und der temporären Tabelle hinzugefügt. Basierend auf der `REFSTR`, die der temporären Tabelle hinzugefügt wurde, können der Import in Alfabet-Datenbank -Tabellen und die Definition von Beziehungen zwischen Objekten durchgeführt werden.



Der Datenbindungsmechanismus muss während des Imports ausgeführt werden, selbst wenn die Datenbanktabelle, in die importiert wird, leer ist. Während alle anderen eindeutigen Eigenschaften, wie z. B. die `INSTGUID` von neuen Objekten, automatisch beim Schreiben des neuen Datensatzes in die Datenbanktabelle festgelegt werden, wird die `REFSTR`-Eigenschaft in den Alfabet-Datenbank -Tabellen ausschließlich durch den Datenbindungsmechanismus festgelegt.

Die korrekte Aktualisierung und Erzeugung von Objekten in den Objektklassen-Datenbanktabellen der Alfabet-Datenbank, einschließlich der Einstellung der erforderlichen GUIDs und der

Angabe der Uhrzeit der letzten Aktualisierung und Erzeugung sowie der Benutzerinformationen, hängen von der Datenbindung ab. Der ADIF-Import muss über die temporäre Tabelle erfolgen. Die Verwendung von ADIF für den direkten Import von Daten in Objektklassen-Datenbanktabellen der Alfabet-Datenbank über SQL-Befehle kann zu einem Verlust der Datenbankintegrität führen, deshalb sollte davon abgesehen werden.



Alle Eigenschaften der Gruppe `Artifact` verfügen über Eigenschaften, mit denen Informationen über Änderungen am Objekt gespeichert werden.

Die folgenden Eigenschaften werden festgelegt, wenn ein Objekt über den datenbindungs-basierenden ADIF-Import erzeugt oder aktualisiert wird:

- Die Eigenschaft `LAST_UPDATE` wird auf das aktuelle Datum gesetzt.
- Die Eigenschaft `LAST_UPDATE_USER` wird auf den Anwender festgelegt, der die Aktualisierung auslöst. Wenn ein Anwender nicht identifiziert werden kann, wird die Eigenschaft auf `ALFABET_INTERN` gesetzt.

Folgende Eigenschaften werden zusätzlich festgelegt, wenn ein Objekt erzeugt wird:

- Die Eigenschaft `CREATION_DATE` wird auf das aktuelle Datum gesetzt.
- Die Eigenschaft `CREATION_USER` wird auf den Anwender festgelegt, der die Aktualisierung auslöst. Wenn ein Anwender nicht identifiziert werden kann, wird die Eigenschaft auf `ALFABET_INTERN` gesetzt.

Wenn eine Historientabelle für eine Objektklasse verfügbar ist, wird auch die Historie aktualisiert. Informationen hierzu finden Sie unter [Speicherung der Überwachungshistorie](#).

Mit dem Datenbindungsmechanismus wird Folgendes durchgeführt:

- Der temporären Datenbanktabelle wird automatisch die Spalte `REFSTR` hinzugefügt.
- Nach dem Import der externen Daten in die temporäre Tabelle vergleicht der Datenbindungsmechanismus die Datensätze in der temporären Datenbanktabelle mit den Datensätzen in der Alfabet-Datenbank -Zieltabelle anhand von benutzerdefinierten Schlüsseleigenschaften. Wenn Datensätze in der temporären Tabelle mit einem Datensatz in der Alfabet-Datenbank -Tabelle übereinstimmen, wird die `REFSTR` aus dem Alfabet -Datensatz in die Spalte `REFSTR` der temporären Datenbanktabelle geschrieben.
- Für jede Zeile in der temporären Tabelle, für die eine `REFSTR` definiert ist, werden die Daten aus der temporären Tabelle verwendet, um die Daten mit einer entsprechenden `REFSTR` in der Alfabet-Datenbank -Tabelle zu aktualisieren.
- Für alle Zeilen in der temporären Datenbanktabelle, für die keine `REFSTR` definiert ist, werden neue Objekte zur Alfabet-Datenbank -Zieltabelle hinzugefügt. Beim Erzeugen der neuen Objekte wird die Eigenschaft `REFSTR` automatisch in den Alfabet-Datenbank -Zieltabellen festgelegt.
- Die `REFSTR` der neuen Objekte werden der temporären Datenbanktabelle hinzugefügt. Wenn die Verarbeitung des Bindungsbefehls des Importeintrags abgeschlossen ist, ist die Spalte `REFSTR` der temporären Tabelle vollständig mit den richtigen `REFSTR` aller Objekte eingestellt.

Die Datenbindung erfordert die folgende Definition:

- Der Name der Zielobjektklasse im Alfabet -Metamodell muss im Attribut **Klassenname** des Importeintrags definiert sein.

- Für den Importeintrag muss wenigstens ein Element **Attribut** mit dem Attribut **Eigenschaftsname** definiert werden, mit dem der **Name** einer Eigenschaft in der Zielobjektklasse definiert wird.
- Die Zuordnungsbedingungen müssen definiert sein. Es gibt zwei Möglichkeiten, Zuordnungsbedingungen zu definieren:
 - Das Attribut **Ist Schlüssel** ist bei mindestens einem Element **Attribut** mit einer Definition **Eigenschaftsname** auf `True` festgelegt. Der ADIF-Datenbindungsmechanismus führt dann eine automatisch erzeugte Bindungsabfrage aus, die den Inhalt der Schlüsselspalte in der temporären Datenbanktabelle dem Inhalt der Spalte in der Alfabet-Datenbank -Tabelle zuordnet, die mit dem Attribut **Eigenschaftsname** angegeben wurde. Sie können das Attribut **Ist Schlüssel** für mehrere Elemente vom Typ **Attribut** auf `True` festlegen. Der Bindungsmechanismus ordnet dann Datensätze zu, wenn alle definierten Bedingungen zutreffen.
 - Die Spezifikation der Datenzuordnung geschieht über eine benutzerdefinierte SQL-Abfrage, die im Ordner **SQL-Befehle - DataBind** definiert ist.

Beide Mechanismen können parallel zueinander verwendet werden.

Es wird empfohlen, sich wann immer möglich auf die Zuordnung mithilfe von Schlüsselattribute zu verlassen.

So definieren Sie einen eigenen SQL-Befehl für die Datenbindung:

- 1) Navigieren Sie im Explorer der ADIF-Konfigurationsoberfläche zu dem entsprechenden **Importeintrag** .
- 2) Klicken Sie mit der rechten Maustaste auf den Ordner **SQL-Befehle - DataBind**, und wählen Sie **SQL-Befehl erzeugen** aus. Dem Ordner wird ein neuer SQL-Befehl  als untergeordneter Knoten hinzugefügt.
- 3) Klicken Sie auf den neuen SQL-Befehl , bearbeiten Sie im Attributfenster optional folgende Attribute:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
 - **Befehlstyp:** Wählen Sie DMLStatement aus.
 - **Ergebnistyp:** Wählen Sie `Undefined` aus.
 - **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Import auch dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Import im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
 - **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.
- 4) Klicken Sie im Attributfenster im Attribut **Text** auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.



Der Texteditor zur Definition von SQL-Abfragen in der ADIF-Konfigurationsbenutzeroberfläche bietet Hilfe zum Definieren von SQL-Abfragen in separaten Registerkarten. Weitere Informationen finden Sie unter *Definieren von SQL-Abfragen für SQL-Befehle* im Kapitel *Konfigurieren von ADIF-Schemata*.

Die SQL-Abfrage muss eine `UPDATE`-Anweisung sein, die auf die temporäre Datenbanktabelle zielt. Die Aktualisierungsanweisung muss so definiert sein, dass sie die Spalte `REFSTR` der temporären Datenbanktabelle mit dem Wert `REFSTR` von übereinstimmenden Objekten in der Zieldatenbanktabelle der Alfabet-Datenbank befüllt. Der Import von neuen Objekten, für die kein passendes Objekt in der Alfabet-Datenbank -Zieltabelle gefunden wurde, und das Festlegen der `REFSTR` für neue Objekte erfolgt automatisch durch den Bindemechanismus nach Ausführung der Anbindungsabfrage.



Zum Beispiel eine typische SQL-Bindungsabfrage für den Import von Gerätegruppen, die vorhandenen Objekten mit demselben Namen zugeordnet sind:

```
UPDATE TMP_DEVICEGROUP
SET REFSTR = (SELECT dg.REFSTR FROM DEVICEGROUP dg WHERE
dg.NAME = TMP_DEVICEGROUP.NAME)
```

Definieren von Beziehungen

Beziehungen zwischen Objekten verschiedener Klassen werden in Objektklasseneigenschaften vom Datentyp `Reference` oder `ReferenceArray` in der Alfabet-Datenbank gespeichert. Das Festlegen von Referenzen zwischen Objekten in den Alfabet-Datenbank kann in den ADIF-Import einbezogen werden.



Beim Importieren von Referenzen zwischen Objekten müssen Sie sicherstellen, dass ein Objekt nicht auf sich selbst verweist. Schleifenreferenzen können in der Alfabet-Datenbank schwerwiegende Probleme verursachen.

Wenn Sie eine Referenz zwischen Objekten festlegen möchten, müssen Sie zunächst auswerten, wie die Referenz in der Alfabet-Datenbank hergestellt wird:

- *Ist die Beziehung Teil des Alfabet -Standardmetamodells?*

Wenn weder die referenzierende noch die referenzierte Objektklasse eine entsprechende Objektklasseneigenschaft vom Typ `Reference` oder `ReferenceArray` enthält, müssen Sie eine benutzerdefinierte Eigenschaft vom Typ `Reference` oder `ReferenceArray` definieren, um die Beziehung herzustellen.



Meilensteinvorlagen müssen im Konfigurationstool Alfabet Expand konfiguriert werden. Informationen über das Erzeugen einer benutzerdefinierten Eigenschaft finden Sie im Abschnitt *Konfigurieren benutzerdefinierter Eigenschaften für geschützte oder öffentliche Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.



Sie müssen die Konfigurationsfunktionen in Alfabet Expand verwenden, um einer Datenbanktabelle einer Alfabet -Objektklasse neue Eigenschaftsspalten hinzuzufügen. Eigenschaftsspalten dürfen nicht mit einer DDL-Anweisung in den SQL-Befehlen des ADIF-Schemas hinzugefügt werden. **Bestehende Alfabet-Datenbank -Standardtabellen dürfen nicht durch SQL-Befehle über ADIF geändert werden.**

- *Von welchem Datentyp ist die Eigenschaft, mit der die Beziehung hergestellt wird?*

Beziehungen vom Typ `Reference` stellen eine Beziehung zu nur einem Objekt her und werden in einem Element vom Typ **Attribut**  im Importeintrag für die referenzierende Klasse konfiguriert. Beziehungen vom Typ `ReferenceArray` stellen eine 1-zu-n-Beziehung zu mehreren Objekten her. Wenn das Attribut **Referenzunterstützung** auf `False` festgelegt ist, wird die Beziehung in einem Element vom Typ **Attribut** im Importeintrag für die referenzierende Klasse konfiguriert. Informationen hierzu finden Sie unter *Definieren von Beziehungen*.

Ist das Attribut **Referenzunterstützung** auf `True` gesetzt, wird die Beziehung in einem Importeintrag für das referenzierte oder referenzierende Objekt in einem Element **Beziehungen** konfiguriert. Weitere Informationen hierzu finden Sie unter.

- *Ist eine Rückbeziehung verfügbar?*

Für eine kleine Anzahl von Eigenschaften vom Typ `Reference` legt ein Datenbank-Auslöser in der Alfabet-Datenbank automatisch eine entsprechende Rückreferenz in einer Eigenschaft der referenzierten Objektklasse fest. Wenn eine Eigenschaft zur Erzeugung von Rückreferenzen führt, zeigt das Attribut **Rückreferenz** der Objektklasseneigenschaft den Namen der Eigenschaft der referenzierten Klasse an, die durch den Auslöser aktualisiert wird. Wenn eine Rückreferenz verfügbar ist, muss im ADIF-Element **Attribut**, das die Beziehung definiert, die Option **Rück-Relationen erneut abfragen** auf `True` gesetzt werden, damit die Rückreferenz automatisch aktualisiert wird.

- *Sind alle Objekte, die Ziel der Beziehung sind, in der Alfabet-Datenbank vorhanden?*

Beziehungen können nur zu bestehenden Objekten in der Alfabet-Datenbank hergestellt werden. Wenn Zielobjekte fehlen, müssen sie vor dem Import in der Alfabet-Datenbank erzeugt werden. Sie können z. B. das ADIF-Importschema so konfigurieren, dass die fehlenden Zielobjekte in der jeweiligen Alfabet-Datenbank -Tabelle erzeugt werden. Stellen Sie sicher, dass das Erzeugen von Objekten vor dem Festlegen von Beziehungen während des Importvorgangs erfolgt.

Das Herstellen von Beziehungen zwischen Objekten erfordert je nach Referenztyp eine andere Konfiguration in ADIF.

Speziellen Support erhalten Sie für die Referenzen, die importiert werden müssen, um einen Indikator oder eine Rolle für ein Objekt zu definieren. Die Definition einer Rolle oder einer Kennzahl für ein Objekt erfordert die Definition von Mehrfachreferenzen zwischen Objektklassen in der Alfabet-Datenbank. ADIF bietet einen Mechanismus, der den Rollen- und Kennzahlenimport für eine Objektklasse übernimmt, ohne dass alle erforderlichen Referenzen in der Konfiguration des ADIF-Schemas separat definiert werden müssen.

Folgende Informationen sind über den Referenzenimport verfügbar:

- [Import von Referenzen und ReferenceArrays, bei denen die Referenzunterstützung auf „False“ festgelegt ist](#)
- [Import von ReferenceArrays mit auf True gesetzter Referenzunterstützung](#)
- [Definieren einer Rolle für importierte Objekte](#)
- [Definieren einer Kennzahl für importierte Objekte](#)

Import von Referenzen und ReferenceArrays, bei denen die Referenzunterstützung auf „False“ festgelegt ist

Die Informationen über Referenzen werden in der Datenbanktabelle der referenzierenden Objektklasse gespeichert. Die Datenbankspalte der referenzierenden Objektklasseneigenschaft muss den Eigenschaftswert `REFSTR` der referenzierten Objektklasse enthalten.

Die `REFSTR` ist eine eindeutige Eigenschaft, die für das Alfabet -Metamodell spezifisch und normalerweise nicht Teil der externen Daten ist. Referenzierte Objekte werden in der Regel über Schlüsselattribute, wie den Namen des referenzierten Objekts, angegeben. Der Import von Werten für eine Eigenschaft vom Typ `Reference` erfordert daher normalerweise, dass die `REFSTR` des importierten Objekts vor dem Datenimport in die Alfabet-Datenbank über einen SQL-Befehl im Ordner **SQL-Befehle - OnStart** in die temporäre Tabelle geschrieben wird.



Der Import von Gerätedaten muss z. B. die Spezifikation eines autorisierten Anwenders enthalten. Die Objektklasse `Device` in der Alfabet-Datenbank hat eine Eigenschaft `ResponsibleUser`, die über eine Referenz auf die Objektklasse `Person` den verantwortlichen Anwender definiert. Der benötigte Wert zum Befüllen der Datenbankspalte der Eigenschaft `ResponsibleUser` ist die `REFSTR` der Objektklasse „Person“.

In der Importdatei wird der autorisierte Anwender durch den Anmeldenamen des Anwenders definiert, der der Eigenschaft `USER_NAME` der Objektklasse „Person“ entspricht. Der Spaltenname in der externen Datentabelle lautet `Owner`:

	A	B	C	D	E
1	Name	Version	StartDate	End Date	Owner
2	ImportDevice3	2.0	01.04.2011	18.05.2018	Ngombe
3	ImportDevice4	3.0	24.08.2011	31.12.2014	Picard

Führen Sie bei der Definition des den Import auslösenden Importeintrags die folgenden Konfigurationen im Importeintrag durch, um die Werte für die Eigenschaft `ResponsibleUser` der Objektklasse `Device` festzulegen:

- 1) Definieren Sie ein Element **Attribut** , das die Daten über den Anmeldenamen des Anwenders in die temporäre Tabelle schreibt. Ordnen Sie die Spalte nicht einer Eigenschaft der Zieldatenbanktabelle in der Alfabet-Datenbank zu.
- 2) Definieren Sie ein Element **Attribut** , mit dem eine temporäre Datenbankspalte erzeugt wird, die während des Imports nicht mit externen Daten ausgefüllt wird. Ordnen Sie die Spalte der Zielobjektklasseneigenschaft zu, mit der die Beziehung hergestellt wird. In dem Beispiel wird die leere Spalte in der Alfabet-Datenbank der Eigenschaft `ResponsibleUser` der Zielobjektklasse `Device` zugeordnet.
- 3) Klicken Sie mit der rechten Maustaste auf den Ordner **SQL-Befehle - OnStart**, und wählen Sie **SQL-Befehl erzeugen** aus.
- 4) Klicken Sie auf den neuen SQL-Befehl , und geben Sie im Attributfenster Folgendes an:
 - **Name:** Geben Sie einen Namen für die Servicelevel-Vereinbarung (SLA) ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
 - **Befehlstyp:** Wählen Sie `DMLStatement` aus.

- **Ergebnistyp:** Wählen Sie `Undefined` aus.
- **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Export auch dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Export im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
- **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.
- Klicken Sie im Attributfenster im Attribut **Text** auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.



Die SQL-Abfrage muss eine `UPDATE`-Anweisung sein, die auf die temporäre Datenbanktabelle zielt. Die Abfrage muss die Spalte für die Eigenschaft, die die Beziehung herstellt, mit den `REFSTR` der referenzierten Objekte befüllen:

```
UPDATE TMP_DEVICE
SET OWNER_REF = (SELECT p.REFSTR FROM PERSON p WHERE p.USER_NAME =
TMP_DEVICE.OWNER)
```

Import von ReferenceArrays mit auf `True` gesetzter Referenzunterstützung

Beziehungen vom Typ `ReferenceArray` werden in der Tabelle `RELATIONS` gespeichert. Die ADIF-Benutzeroberfläche stellt einen Mechanismus bereit, um die Tabelle `RELATIONS` während des Datenimports in eine Zielobjektklasse mit den erforderlichen Daten auszufüllen.



Es ist nicht zulässig, die Tabelle `RELATIONS` direkt über einen SQL-Befehl zu ändern, bei dem Daten aus einer temporären Tabelle in die Tabelle `RELATIONS` eingegeben werden. Zur Gewährleistung der Datenintegrität müssen `RELATIONS` des Typs `ReferenceArray`, bei denen das Attribut **Referenzunterstützung** auf `True` gesetzt ist, mit den unten beschriebenen Mechanismen importiert werden.

Die temporäre Tabelle muss die folgenden Kriterien erfüllen, um Referenz-Array-Einträge in die Tabelle `RELATIONS` zu importieren:

- Sowohl die `REFSTR` des referenzierenden Objekts als auch die `REFSTR` des referenzierten Objekts können aus der im Importeintrag definierten temporären Tabelle gelesen werden.
- Für jede Beziehung, die der Tabelle `RELATIONS` hinzugefügt werden soll (für jede Kombination zwischen referenzierendem Objekt und referenziertem Objekt), muss in der temporären Tabelle eine Zeile vorhanden sein.

Die `REFSTR` ist eine eindeutige Eigenschaft, die für das Alfabet -Metamodell spezifisch ist und normalerweise nicht Teil der externen Daten ist. Referenzierte Objekte werden in der Regel über Schlüsselattribute, wie den Namen des referenzierten Objekts, angegeben. Der Import von Werten für eine Eigenschaft vom Typ `Reference` erfordert daher in der Regel, dass die `REFSTR` des importierten Objekts vor dem Datenimport in die Alfabet-Datenbank über einen SQL-Befehl im Ordner **SQL-Befehle - OnStart** in die temporäre Tabelle geschrieben wird.

- Definieren Sie ein Element **Attribut** , das die Daten über den Anmeldenamen des Anwenders in die temporäre Tabelle schreibt. Ordnen Sie die Spalte nicht einer Eigenschaft der Zieldatenbanktabelle in der Alfabet-Datenbank zu.
- Definieren Sie ein Element **Attribut** , mit dem eine temporäre Datenbankspalte erzeugt wird, die während des Imports nicht mit externen Daten ausgefüllt wird. Ordnen Sie die Spalte der Zielobjektklasseneigenschaft zu, mit der die Beziehung hergestellt wird. In dem Beispiel wird die leere Spalte in der Alfabet-Datenbank der Eigenschaft `ResponsibleUser` der Zielobjektklasse `Device` zugeordnet.
- Definieren Sie einen SQL-Befehl  im Ordner **SQL-Befehle - OnStart**, und definieren Sie eine `UPDATE`-Anweisung, die für die temporäre Tabelle ausgeführt wird und die leere Spalte mit der `REFSTR` des referenzierten Objekts befüllt.

Um die Änderungen in der Tabelle `RELATIONS` auszulösen, müssen Sie dem Importeintrag ein Element „Beziehung“  hinzufügen:

- 1) Klicken Sie mit der rechten Maustaste auf den Ordner **Beziehungen** des Importeintrags , für den Sie die Reference-Array-Daten importieren möchten, und wählen Sie **Beziehung erzeugen**. Das neue Element „Beziehung“  wird als untergeordneter Knoten des Ordners **Beziehungen** hinzugefügt.
- 2) Klicken Sie auf das Element „Beziehung“ , und legen Sie im Attributfenster die folgenden Attribute fest:
 - **Von Klasse:** Wählen Sie in der Dropdown-Liste die Objektklasse der Alfabet-Datenbank aus, die die referenzierende Eigenschaft enthält.
 - **Beziehungsname** Geben Sie den Namen des LDAP Servers ein.
 - **Von Spalte:** Geben Sie den Namen der Spalte Ihrer temporären Tabelle an, in der die `REFSTR` des referenzierenden Objekts gespeichert ist.
 - **Klasse:** Wählen Sie in der Dropdown-Liste die referenzierte Objektklasse der Alfabet-Datenbank aus.
 - **Nach Spalte:** Geben Sie den Namen der Spalte Ihrer temporären Tabelle an, in der die `REFSTR` des referenzierten Objekts gespeichert ist.
 - **Vor dem Erzeugen löschen** Wählen Sie `True`, wenn alle Beziehungen, die durch die mit dem Attribut **Beziehungsname** definierte Eigenschaft hergestellt wurden, vor dem Import der neuen Beziehungen aus der Alfabet -Tabelle `RELATIONS` gelöscht werden sollen. Wählen Sie `False`, wenn Sie neue Beziehungen in die Tabelle `RELATIONS` schreiben möchten, ohne die Tabelle `RELATIONS` zu löschen.



Zum Beispiel sollen Daten zu Geräten und Informationen zur Zuordnung der Geräte zu Gerätegruppen über eine XML-Datei importiert werden. In der XML-Datei wird jede Gerätegruppe durch ein Element `DeviceGroups` repräsentiert. Die Geräte in der Gerätegruppe werden durch untergeordnete Elemente `Device` in den jeweiligen `DeviceGroup`-Elementen dargestellt:

```
<Groups>
  <DeviceGroup Name="ImportGroup1">
    <Description>Server hosts</Description>
```

```

<Device Name="ImportDevice1" Version="1.0"
  StartDate="2011.01.01" EndDate="2015.01.05"/>

<Device Name="ImportDevice2" Version="1.2"
  StartDate="2011.08.24" EndDate="2014.12.31"/>

</DeviceGroup>

<DeviceGroup Name="ImportGroup2">

  <Description>Web relevant devices</Description>

  <Device Name="Backup Server" Version="2.2.1"
    StartDate="2009.09.30" EndDate="2013.09.01"/>

  <Device Name="ImportDevice1" Version="1.0"
    StartDate="2011.01.01" EndDate="2015.01.05"/>

</DeviceGroup>

</Groups>

```

Im ADIF-Importschema für den Import der Daten wird ein Importeintrag für den Import von Gerätegruppen und einer für den Import von Geräten hinzugefügt. Die Beziehung zwischen den Gerätegruppen und Geräten wird über das automatisch generierte Attribut `DeviceGroup_Id` gespeichert.

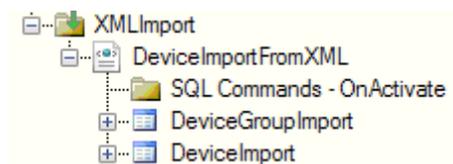
(Einzelheiten zur Bearbeitung der XML-Daten finden Sie unter *Definieren des Datenimports aus XML-Dateien in einer XML-Importgruppe*.)

Im Metamodell Alfabet wird die Beziehung zwischen Gerätegruppen und Geräten über die Objektklasse `Device` mit der Eigenschaft `DeviceGroups` und über die Objektklasse `DeviceGroup` mit der Eigenschaft `Devices` definiert. In diesem Beispiel wird die Eigenschaft `DeviceGroups` der Objektklasse `Device` verwendet, um die Beziehung zu definieren. Die Rückreferenz wird vom Importmechanismus beim Datenimport automatisch festgelegt.

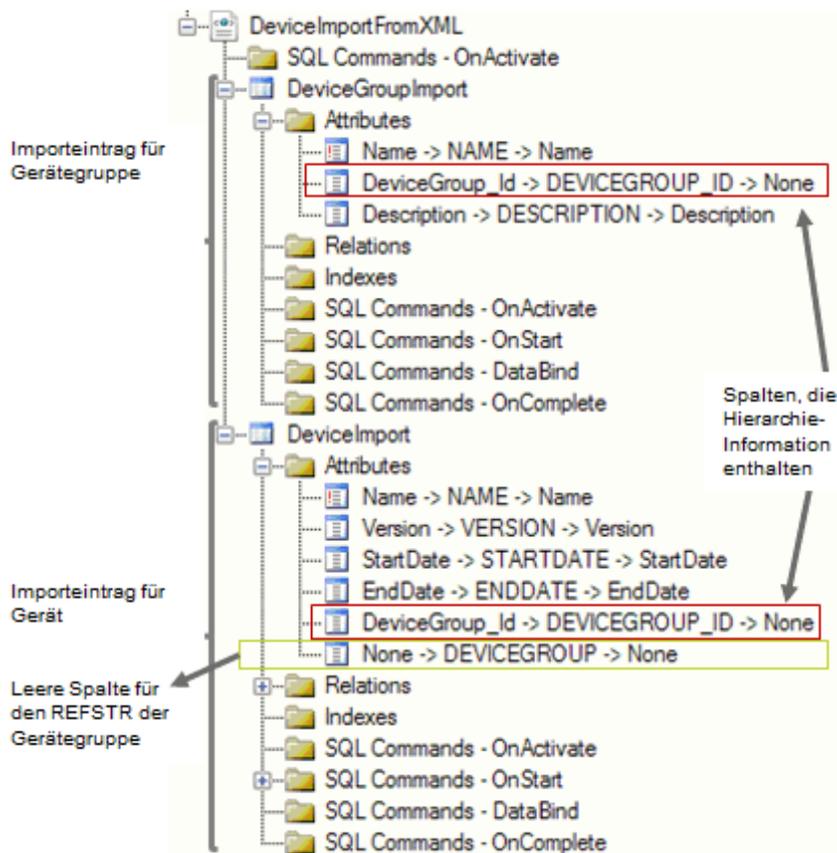
Um eine Beziehung herzustellen, muss der ADIF-Importmechanismus sowohl die `REFSTR` des referenzierenden Objekts als auch die `REFSTR` des referenzierten Objekts in der temporären Datenbanktabelle finden, für die das Element `Beziehung` angegeben ist. Das Beziehungselement wird nach dem Import von Daten in die Datenbanktabelle der Zielobjektklasse in der Alfabet-Datenbank verarbeitet, und somit kann die `REFSTR` des referenzierenden Geräts aus der automatisch erzeugten Spalte `REFSTR` der temporären Tabelle gelesen werden.



Die `REFSTR` der Gerätegruppe muss über einen SQL-Befehl vom Typ `OnStart` in die temporäre Tabelle aufgenommen werden. Daher ist die Reihenfolge der Importeinträge im ADIF-Schema so konfiguriert, dass der Import von Gerätegruppensdaten vor dem Import von Gerätedaten ausgelöst wird. Dies ist erforderlich, um sicherzustellen, dass Referenzen nur auf bereits in der Alfabet-Datenbank vorhandene Objekte festgelegt werden, und stellt parallel zu den Informationen über die `REFSTR` der Gerätegruppen auch Informationen über die `DeviceGroup_Id` in der temporären Tabelle für Gerätegruppen bereit.



Bei der Definition der temporären Tabelle für den Import von Gerätedaten wird der temporären Tabelle eine leere Datenbankspalte hinzugefügt, die weder mit externen Daten befüllt noch der Alfabet-Datenbank -Zieltabelle zugeordnet ist.



Im Ordner **SQL-Befehle - OnStart** wird ein SQL-Befehl hinzugefügt, der die leere Spalte DEVICEGROUP mit der REFSTR der übergeordneten Gerätegruppe unter Verwendung der DeviceGroup_Id als Zuordnungseigenschaft befüllt:

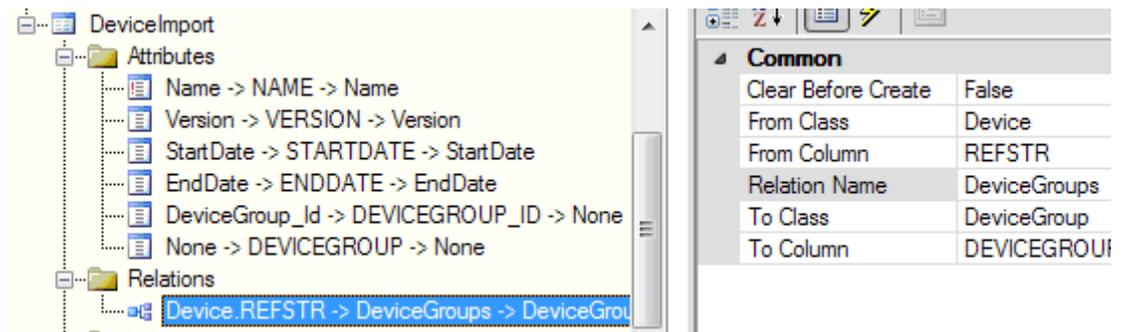
```
UPDATE TMP_DEVICE
SET DEVICEGROUP = ( SELECT REFSTR FROM TMP_DVG
WHERE TMP_DVG.DEVICEGROUP_ID = TMP_DEVICE.DEVICEGROUP_ID)
```

So sieht die daraus resultierende temporäre Tabelle aus:

	NAME	VERSION	STARTDATE	ENDDATE	DEVICEGROUP_ID	DEVICEGROUP	REFSTR
1	ImportDevice1	1.0	01/01/2011 00:00:00	05/01/2015 00:00:00	0	96-66-0	20-129-0
2	ImportDevice2	1.2	24/08/2011 00:00:00	31/12/2014 00:00:00	0	96-66-0	20-130-0
3	Backup Server	2.2.1	30/09/2009 00:00:00	01/09/2013 00:00:00	1	96-67-0	20-6-0
4	ImportDevice1	1.0	01/01/2011 00:00:00	05/01/2015 00:00:00	1	96-67-0	20-129-0

Die Tabelle enthält für jede Kombination aus Gerät und Gerätegruppe eine eigene Zeile. Das bedeutet, dass es für das Gerät ImportDevice1, das zwei Gerätegruppen zugeordnet ist, zwei Zeilen gibt. Die Datenbindung hat diese Informationen jedoch korrekt verarbeitet. Als Ergebnis zeigt die Spalte REFSTR in beiden Zeilen die gleiche REFSTR an.

Im Ordner **Beziehungen** des Importeintrags für den Geräteimport wurde ein Element „Beziehung“ hinzugefügt. Die Daten in den Spalten DEVICEGROUP und REFSTR der temporären Tabelle werden verwendet, um die Beziehung zu definieren:



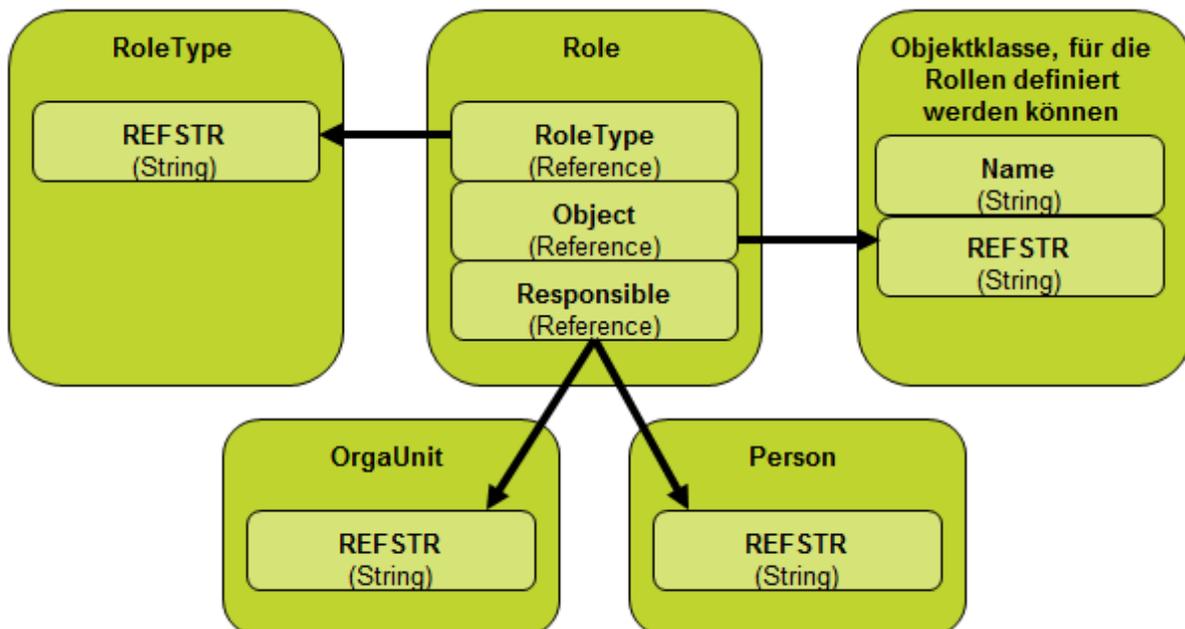
Definieren einer Rolle für importierte Objekte

Jedes Mal, wenn für ein auf einem vorkonfigurierten Rollentyp basierten Objekt eine Zuständigkeit definiert wird, wird ein Objekt der Objektklasse `Role` erzeugt.

Rollentypen definieren den Typ der Verantwortlichkeit, die ein Anwender oder eine Organisation für ein Objekt in der Alfabet-Datenbank hat. Rollentypen werden in der Funktionalität **Referenzdatendefinition** im Modul **Konfiguration** von Alfabet konfiguriert. Sie werden dann in der Funktionalität **Referenzdatenzuordnung** den Objektklassen zugeordnet.

Auf der Ansichtssseite **Verantwortlichkeiten** eines Objekts kann der Anwender jetzt eine Rolle erzeugen, indem er entweder für einen Anwender oder für eine Organisation eine Verantwortlichkeit definiert. Im Editor für die Rollendefinition werden der verantwortliche Anwender oder die verantwortliche Organisation und der Rollentyp definiert, der die Art von Verantwortlichkeit für das Objekt definiert.

Im Klassenmodell von Alfabet wird die Beziehung zwischen dem Rollentyp, dem Objekt und der verantwortlichen Person oder Organisation nur in der Datenbanktabelle für die Klasse `Role` angegeben. Die Informationen enthalten keine Objektklasseneigenschaften vom Typ „Referenz“ oder „ReferenceArray“ der Objektklasse „Person“ oder „OrgaUnit“, oder der Objektklasse, für die die Rolle definiert ist:



Um Informationen über die Rollen für eine importierte Objektklasse in den ADIF-Import einzuschließen, müssen Sie nicht zusätzlich den Import des Objekts `Role` in ihr ADIF-Schema über einen separaten ADIF-Importeintrag einbinden. Stattdessen kann die Erzeugung des Objekts `Role` automatisch beim Importieren von Objekten der Objektklasse durchgeführt werden, für die die Rolle mit folgender Konfiguration eingestellt ist:

- 1) Definieren Sie ein **Attribut-Element** , das eine eindeutige Eigenschaft des Benutzers (Objektklasse `Person`) oder der Organisation (Objektklasse `OrgaUnit`) schreibt, die über die Rolle die Verantwortung für das Objekt in die temporäre Tabelle übernehmen soll. In der Regel wird dies der Name des Benutzers oder der Organisation sein. Ordnen Sie die Spalte nicht einer Eigenschaft der Zieldatenbanktabelle in der Alfabet-Datenbank zu.



Informationen über das Definieren von **Attributen**  finden Sie unter [Erstellen von Attributeinträgen zum Zuordnen der Datenbanktabellenspalten](#).

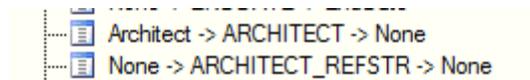
- 2) Definieren Sie ein Element **Attribut** , mit dem eine temporäre Datenbankspalte erzeugt wird, die während des Imports weder mit externen Daten ausgefüllt noch zu einer Zielobjekt-Klasseneigenschaft zugeordnet wird. In einem späteren Schritt wird die Spalte über eine SQL-Abfrage mit dem `REFSTR` des Benutzers oder der Organisation gefüllt.
- 3) Klicken Sie mit der rechten Maustaste auf den Ordner **SQL-Befehle – OnStart**, und wählen Sie **SQL-Befehl erzeugen** aus.
- 4) Klicken Sie auf den neuen SQL-Befehl , und geben Sie im Attributfenster Folgendes an:
 - **Name:** Geben Sie einen Namen für die Servicelevel-Vereinbarung (SLA) ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird. Der SQL-Befehl wird verwendet, um `REFSTR` des für das Objekt zuständigen Benutzers oder der Organisation in die temporäre Datenbanktabelle einzugeben.
 - **Befehlstyp:** Wählen Sie `DMLStatement` aus.
 - **Ergebnistyp:** Wählen Sie `Undefined` aus.
 - **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Export auch dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Export im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
 - **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren.
 - Klicken Sie im Attributfenster im Attribut **Text** auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.

Die SQL-Abfrage muss eine `UPDATE`-Anweisung sein, die auf die temporäre Datenbanktabelle zielt. Die Abfrage muss die Spalte für den `REFSTR` des für das Objekt verantwortlichen Benutzers oder der Organisation über die Rolle mit dem `REFSTR` des Benutzers oder der Organisation füllen.



Im folgenden Beispiel wird die Rolle „Architekt“ für importierte Applikationen definiert. Im Ordner „Attribut“ des Importeintrags für die Objektklasse `Application` wurden zwei **Attributelemente**  für den Import definiert. Die Eigenschaft `TECH_NAME` der Objektklasse `Person` wird während des Imports in die

Datenbankspalte `ARCHITECT` ohne weiteren Import in eine Standardobjektklassentabelle der Alfabet-Datenbank importiert. Die zusätzliche Spalte `ARCHITECT_REFSTR` wird für die temporäre Tabelle definiert, ohne Datenimport oder Datenintegration in eine Standardobjektklassentabelle der Alfabet-Datenbank.



Der folgende SQL-Befehl füllt diese Spalte mit der `REFSTR` des Benutzers, `TECH_NAME`, der durch den technischen Namen (Eigenschaft) in der Spalte `ARCHITECT` der Temporärtabelle bestimmt wird:

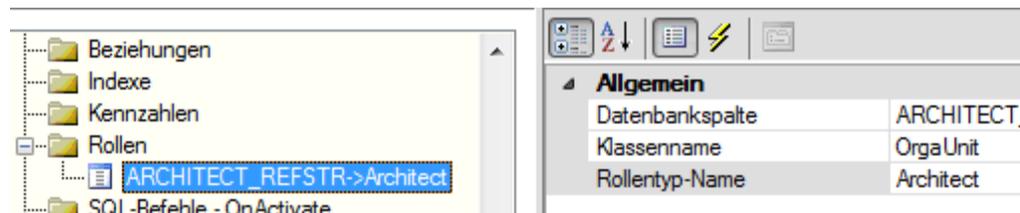
```
UPDATE TMP_APPLICATION
SET ARCHITECT_REFSTR = (SELECT REFSTR FROM PERSON WHERE
PERSON.TECH_NAME = TMP_APPLICATION.ARCHITECT)
```

5) Klicken Sie mit der rechten Maustaste auf den Ordner **Rollen**, und wählen Sie **Rolle erzeugen** aus.

6) Klicken Sie auf die neue Rolle  und geben Sie im Attributfenster Folgendes an:

- **Klassenname:** Geben Sie `Person` ein, wenn die Rolle einem Benutzer zugewiesen wird oder `OrgaUnit`, wenn die Rolle einer Organisation zugewiesen wird.
- **Datenbankspalte:** Geben Sie den Namen der Datenbankspalte der temporären Tabelle ein, in der der `REFSTR` des Benutzers oder der Organisation, dem oder der die Rolle zugewiesen wird, ausgefüllt wird.
- **Rollentyp-Name:** Geben Sie den Namen des Rollentyps ein, für den die Rolle definiert ist. Dies ist der Wert der Objektklasseneigenschaft `Name` des Objekts der Objektklasse `RoleType`, die der Rolle zugewiesen wird.

 Für das obige Beispiel lautet die erforderliche Definition des Rollenelements :



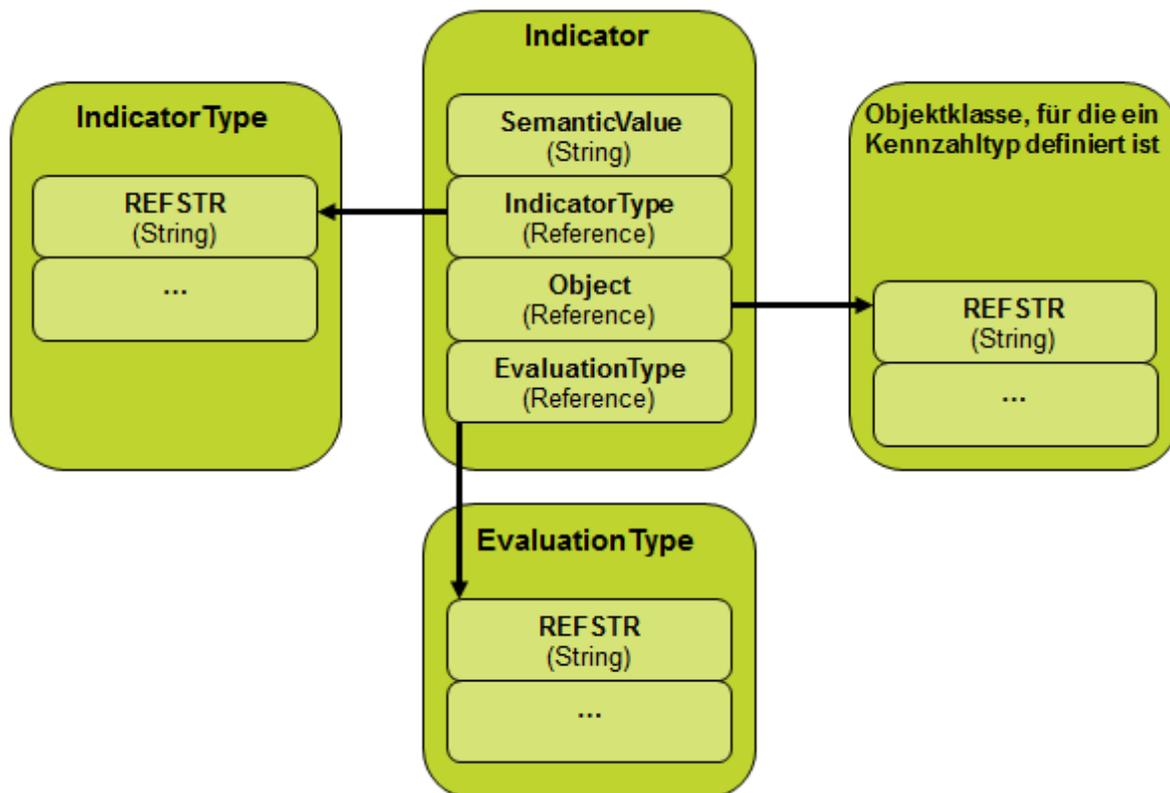
Definieren einer Kennzahl für importierte Objekte

Sie können in Alfabet im Modul **Konfiguration** mit der Funktion **Bewertungen und Portfolios** Kennzahlensysteme definieren, die einen Satz von Kennzahltypen bündeln, um Objekte in zu bewerten. Die Kennzahlensysteme werden in der Funktionalität **Referenzdatenzuordnung** den Objektklassen zugeordnet.

Die Kennzahltypen, die für die der Objektklasse zugeordneten Kennzahlensysteme definiert sind, werden auf der Ansichtseite *Bewertung* für jedes Objekt in der Objektklasse angezeigt. Der Anwender kann Kennzahlen für das Objekt definieren, indem er auf der Ansichtseite *Bewertung* den Wert des Kennzahltyps definiert.

Jedes Mal, wenn auf der Ansichtseite **Bewertung** für ein Objekt ein Wert für einen Kennzahltyp eingegeben wird, wird ein Objekt der Objektklasse "Kennzahl" erzeugt.

Im Klassenmodell von Alfabet wird der Wert, der für den Kennzahltyp eines Objekts definiert ist, nur in der Datenbanktabelle für die Klasse `Indicator` angegeben. Die Klasse `Indicator` gibt das Objekt und den Kennzahltyp an, für den jeder Wert definiert wurde, und das Kennzahlensystem, dem der Kennzahltyp zugeordnet ist:



Um Informationen über die Kennzahlen für eine importierte Objektklasse in den ADIF-Import einzuschließen, müssen Sie nicht zusätzlich den Import des Objekts `Indicator` in ihr ADIF-Schema über einen separaten ADIF-Importeintrag einbinden. Stattdessen kann die Erzeugung des Objekts `Indicator` automatisch beim Importieren von Objekten der Objektklasse durchgeführt werden, für die die Rolle mit folgender Konfiguration eingestellt ist:

- 1) Definieren Sie ein **Attributelement** , das die Werte für die Kennzahlen, die für die importierten Objekte gesetzt werden sollen, in eine Spalte der temporären Datenbanktabelle schreibt. Dieser Wert muss das für die Objektklasse `SemanticValue` der Objektklasse „Kennzahl“ erforderliche Format haben und der Konfiguration der zulässigen Kennzahlenwerte entsprechen, die sich aus den Definitionen von Kennzahltyp und Kennzahlensystem ergeben. Ordnen Sie die Spalte nicht einer Eigenschaft der Zieldatenbanktabelle in der Alfabet-Datenbank zu.

 Informationen über das Definieren von **Attributen**  finden Sie unter [Erstellen von Attributeinträgen zum Zuordnen der Datenbanktabellenspalten](#).

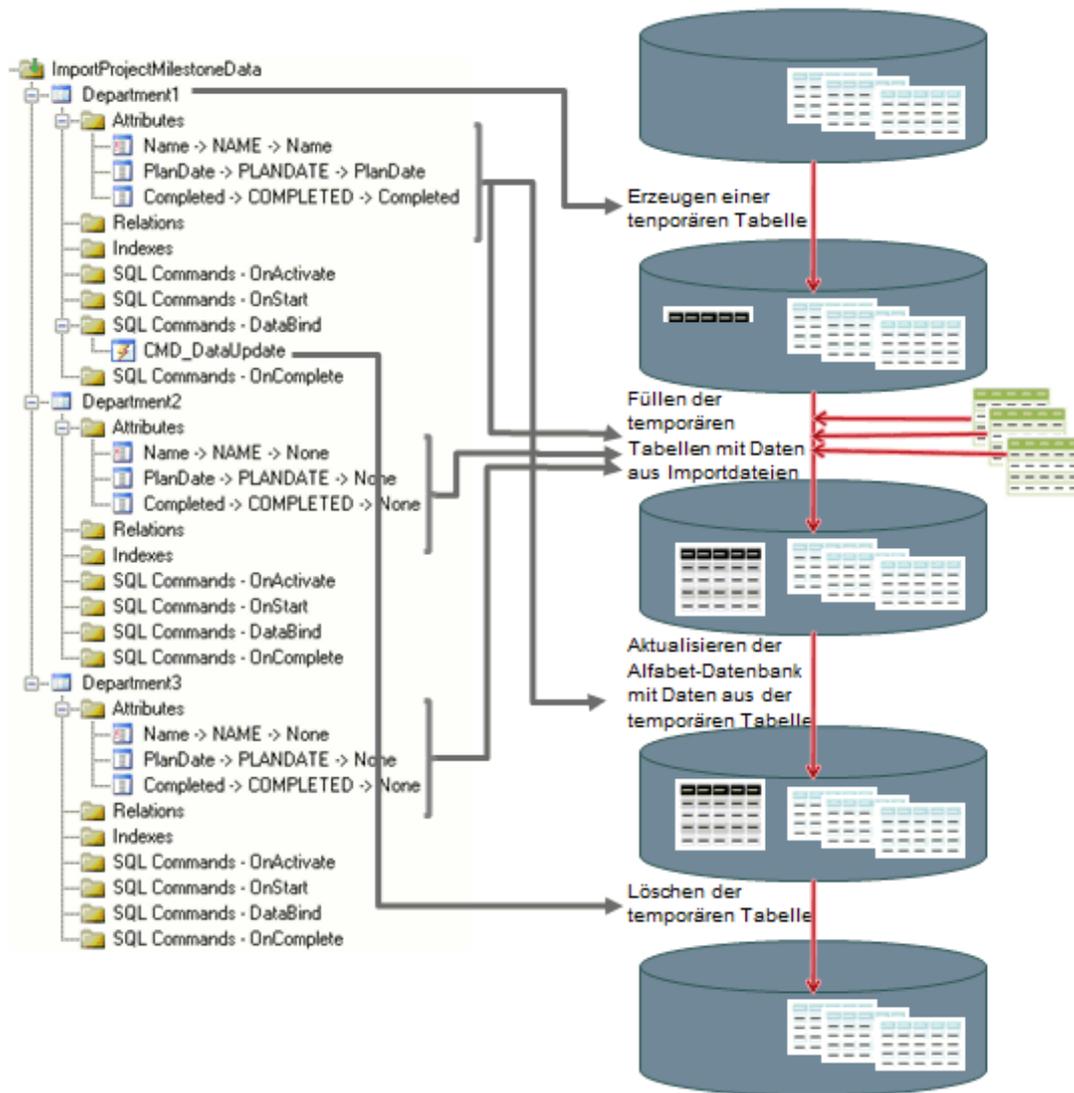
Informationen über die Konfiguration von Kennzahltypen und Kennzahlensystemen finden Sie im Kapitel *Konfigurieren von Bewertungen, Priorisierungsschemata und Portfolios* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

- 2) Klicken Sie mit der rechten Maustaste auf den Ordner **Kennzahlen**, und wählen Sie **Kennzahl erzeugen** aus.
- 3) Klicken Sie auf die neue Kennzahl  und geben Sie im Attributfenster Folgendes an:

- **Datenbankspalte:** Geben Sie den Namen der Datenbankspalte der temporären Tabelle ein, die beim Import mit dem Wert für die Kennzahl gefüllt wird.
- **Technischer Name des Kennzahltyps:** Geben Sie die Werte der Objekteigenschaft `TechnicalName` des `EvaluationType`, das der Kennzahl zugeordnet ist, gefolgt von `|` gefolgt von den Werten der Objekteigenschaft `TechnicalName` des `IndicatorType`, der der Kennzahl zugeordnet ist (`EvaluationTypeTechnicalName|IndicatorTypeTechnicalName`), ein. Wenn die Kennzahl keinem Kennzahlensystem zugeordnet werden soll, können Sie nur den technischen Namen des Kennzahltyps (`IndicatorType`) angeben (`IndicatorTypeTechnicalName`).

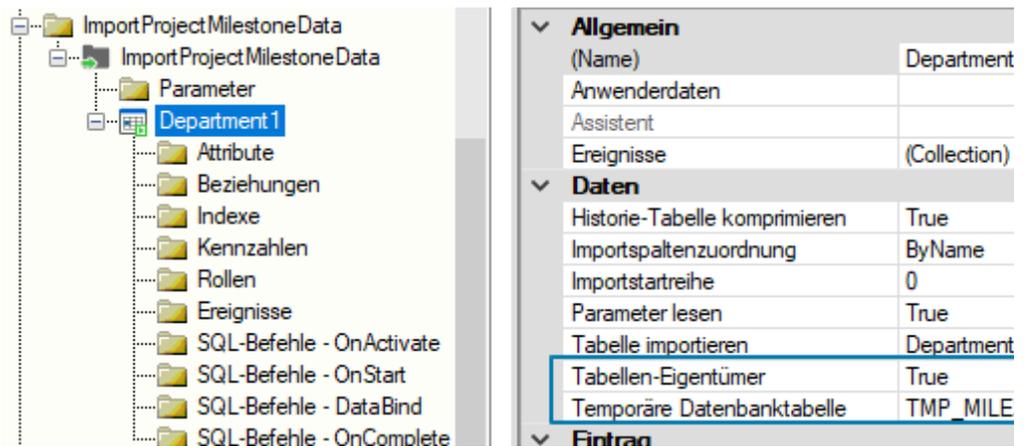
Sammeln von Daten aus mehreren externen Dateien oder Datenbanktabellen in einer temporären Tabelle

Der Standard-Datenimport führt zur Erzeugung einer temporären Tabelle für den Import von Daten aus jeder externen Datei oder Datenbanktabelle. Wenn Sie Daten aus mehreren Dateien mit der gleichen Datenstruktur, aber unterschiedlichem Inhalt importieren möchten, können Sie das ADIF-Importschema so konfigurieren, dass Daten aus mehreren Dateien / externen Datenbanktabellen in eine einzige temporäre Tabelle geschrieben werden. Diese Methode erhöht die Performance des Importvorgangs, da weniger Tabellen in der Datenbank angelegt und gelöscht werden müssen und die Zuordnung der importierten Datensätze zu den bereits in der Alfabet-Datenbank vorhandenen Datensätzen nur einmal durchgeführt werden muss.

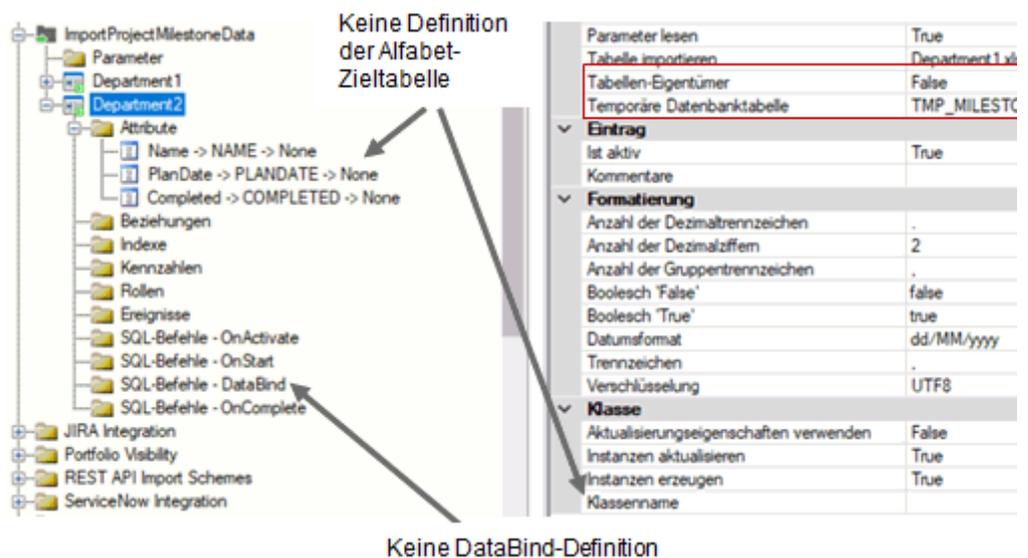


Konfigurieren Sie Folgendes, um während des Imports Daten aus mehreren Importquellen in einer temporären Tabelle zu verarbeiten:

- Erzeugen Sie einen **Importeintrag** , und konfigurieren Sie einen vollständigen Datenimport in die Alfabet-Datenbank –Zieltabelle über die temporäre Tabelle im Importeintrag. Dies erfordert, dass Sie mindestens die Elemente vom Typ „Attribut“  festlegen, die Daten zwischen den Importdaten, der temporären Tabelle und der Tabelle Alfabet-Datenbank zuordnen, und dass Sie das Attribut **Tabellen-Eigentümer** dieses Importeintrags auf `True` setzen. Dieser Importeintrag wird im Folgenden als Tabellen-Eigentümer bezeichnet.



- Erzeugen Sie für jede zusätzliche externe Importdatei/Datenbanktabelle einen **Importeintrag** , der den Import der Daten in die temporäre Tabelle definiert, die im Importeintrag des Tabellen-Eigentümers festgelegt wurde:
 - Das Attribut **Temporäre Datenbank-Tabelle** muss mit der **Temporären Datenbank-Tabelle** identisch sein, die für den Tabellen-Eigentümer angegeben ist.
 - Das Attribut **Tabellen-Eigentümer** muss auf `False` gesetzt sein.
 - Die Elemente **Attribut**  müssen nur den Import von Daten in die temporäre Tabelle angeben. Der Import in die Tabelle Alfabet-Datenbank muss undefiniert bleiben.
 - Es darf keine Datenbindung definiert sein.



Halbautomatische Erzeugung von Importeintragsdefinitionen für Importdateien

Die ADIF-Konfigurationsschnittstelle bietet einen Mechanismus, der eine Importdatei liest und automatisch Importeinträge und Elemente vom Typ **Attribut**  für den Importeintrag passend zu den Daten in einer

Importdatei erzeugt. Die automatisch erzeugten Einträge sind unvollständig, da sie die Definition des Imports aus der temporären Datenbanktabelle in die Zieltabelle Alfabet-Datenbank nicht abdecken können.



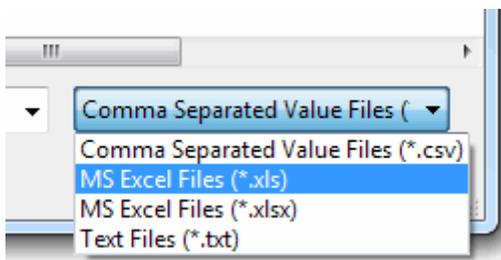
Während der automatischen Vorkonfiguration des Importeintrags wird der Name der temporären Tabelle als `TMP_<Name der Importdatei ohne Erweiterung>` definiert, und die Namen der Datenbankspalten für die temporäre Tabelle werden aus den Spaltenüberschriften der Importdatei abgeleitet. Der Mechanismus ersetzt Leerzeichen und Schrägstriche durch Unterstriche, aber andere Sonderzeichen im Dateinamen oder in den Spaltenüberschriften werden ignoriert oder führen gar zu ungültigen technischen Namen für Datenbanktabellen oder -spalten.

Sie können entweder einen einzelnen Importeintrag für einen Import aus einer einzelnen Microsoft® Excel®- oder kommagetrennten Datei definieren oder aber einen Dateisatz für den Import aus mehreren Importdateien. Für den Import von XML-Dateien kann mithilfe des Mechanismus auch ein Dateisatz erstellt werden, der den Import aller verfügbaren Elemente in der XML-Datei abdeckt.

Halbautomatisches Erstellen eines Importeintrags für eine einzelne Importdatei

Um diesen Mechanismus verwenden zu können, muss es sich bei der Importdatei um eine Microsoft® Excel®, CSV- oder TXT-Datei handeln, und sie muss im lokalen Dateisystem verfügbar sein.

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsoberfläche mit der rechten Maustaste auf das ADIF-Importschema oder die Datei-Importgruppe , für das bzw. die Sie einen Importeintrag erzeugen möchten, und wählen Sie **Eintrag aus Datei erzeugen**. Ein Explorer-Fenster wird geöffnet.
- 2) Wählen Sie im Explorer im Dropdown-Listenfeld neben dem Feld für den Dateinamen das Importdateiformat aus.



- 3) Navigieren Sie zu der Importdatei im lokalen Dateisystem, und klicken Sie auf **Öffnen**. Der neue Importeintrag  wird dem ADIF-Importschema hinzugefügt, wobei die Elemente **Attribut** , die den Importdaten entsprechen, automatisch dem Importeintrag hinzugefügt werden.
- 4) Untersuchen Sie die für die temporäre Tabelle und die Spalten der temporären Tabelle generierten Namen auf technische Korrektheit und Eindeutigkeit.
- 5) Vervollständigen und ändern Sie die automatisch generierte Definition. Details finden Sie unter: *Definieren des Imports externer Daten in einen Importeintrag*.

Halbautomatische Erzeugung von mehreren Importeinträgen in einer Datei-Importgruppe

Um diesen Mechanismus verwenden zu können, muss es sich bei der Importdatei um eine Microsoft® Excel®, CSV- oder TXT-Datei handeln. Alle Dateien, für die Importeinträge erzeugt werden sollen, müssen sich in demselben lokalen Verzeichnis befinden, und das Verzeichnis darf keine anderen Dateien enthalten.

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Importschema, für das Sie eine Importgruppe erstellen möchten, und wählen Sie **Dateisatz aus Verzeichnis erstellen** aus. Ein Selektorfenster wird angezeigt.
- 2) Wählen Sie in der Auswahl das Verzeichnis, das die Importdateien enthält, und klicken Sie auf **OK**. Die neue Importgruppe wird dem ADIF-Importschema hinzugefügt, wobei die Importeinträge  einschließlich der untergeordneten Elemente des Typs **Attribut**  den Importdaten entsprechen.
- 3) Untersuchen Sie die für die temporäre Tabelle und die Spalten der temporären Tabelle generierten Namen auf technische Korrektheit und Eindeutigkeit.
- 4) Vervollständigen und ändern Sie die automatisch generierte Definition.
- 5) Details finden Sie unter: *Definieren des Imports externer Daten in einen Importeintrag*.

Halbautomatisches Erstellen einer XML-Importgruppe aus einer XML-Datei

Um diesen Mechanismus verwenden zu können, muss es sich bei der Importdatei um eine gültige XML-Datei handeln, und sie muss im lokalen Dateisystem verfügbar sein.

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Importschema, für das Sie Importeinträge erstellen möchten, und wählen Sie **XML-Gruppe aus Datei erzeugen** aus. Ein Selektorfenster wird angezeigt.
- 2) Wählen Sie in der Auswahl das Verzeichnis, das die Importdateien enthält, und klicken Sie auf **OK**. Die neue Importgruppe wird dem ADIF-Importschema hinzugefügt, wobei die Importeinträge  einschließlich der untergeordneten Elemente des Typs **Attribut**  den Importdaten entsprechen.
- 3) Untersuchen Sie die für die temporäre Tabelle und die Spalten der temporären Tabelle generierten Namen auf technische Korrektheit und Eindeutigkeit.
- 4) Vervollständigen und ändern Sie die automatisch generierte Definition.
- 5) Details finden Sie unter: *Definieren des Imports externer Daten in einen Importeintrag*.

Konfigurieren von SQL-Befehlen für optionale erweiterte Importfunktionen

Die grundlegende Importkonfiguration kann durch die Definition zusätzlicher SQL-Befehle erweitert werden. Wenn Sie ADIF-Importschemata mithilfe des ADIF-Konfigurationsexplorers erstellen, werden dem Explorer automatisch Ordner für SQL-Befehle hinzugefügt, wenn für ein Element SQL-Befehle definiert werden können. In den Ordnern wird auch auf den Typ des SQL-Befehls hingewiesen.



In der XML-Datei des ADIF-Importschemas verfügt jeder SQL-Befehl über ein Attribut **Typ**. Für jeden zulässigen Befehlstyp wird dem Explorer ein separater Ordner namens **SQL-Befehle - <Typ>** hinzugefügt. Wenn SQL-Befehle in einem Ordner definiert werden, wird das Attribut **Typ** automatisch in der XML-Definition festgelegt. Wenn Sie SQL-Befehle in einem XML-Editor angeben, muss das Attribut **Typ** manuell definiert werden.

Folgende SQL-Befehlstypen sind verfügbar:

Typ	Position im ADIF-Schema	Beschreibung
OnActivate	Importgruppen und Importeinträge	<p>Mit SQL-Befehlen vom Typ <code>OnActivate</code> werden Bedingungen für die Ausführung desjenigen Elements festgelegt, für das sie definiert sind.</p> <p>Weitere Informationen hierzu finden Sie unter <i>Konfigurieren der bedingten Ausführung von Teilen des Importschemas</i>.</p> <p>Hinweis: Von <code>OnActivate</code>-Befehlen ausgelöste Änderungen werden nicht zurückgenommen, wenn die Option Änderungen beibehalten im Importschema auf <code>False</code> gesetzt ist.</p>
DataBind	Importeinträge	<p>SQL-Befehle vom Typ <code>DataBind</code> definieren, wie Daten in den temporären Tabellen den Daten in der Alfabet-Datenbank -Zieltabelle zugeordnet werden. Die Definition von Datenbindungsbefehlen ist optional. Alternativ können die Bedingungen für die Datenzuordnung in den Elementen Attribut  des Importeintrags  über das Attribut Ist Schlüssel definiert werden.</p> <p>Weitere Informationen hierzu finden Sie unter <i>Zuordnen von externen Daten zu temporären Tabellen und Alfabet -Datenbanktabellen</i>.</p>
DataUpload	Importeinträge	<p>Für den Datenimport aus externen Datenbanken und Active Directories sind SQL-Befehle vom Typ <code>DataUpload</code> erforderlich. Die Abfrage wird an der externen Quelle durchgeführt. Die Ausgabe der Abfrage muss in einem tabellarischen Datensatz erfolgen, der die externen Daten repräsentiert, die importiert werden.</p> <p>Weitere Informationen hierzu finden Sie unter <i>Definieren des Hochladens von Daten aus einer externen Datenbank in eine Datenbank-Importgruppe</i> oder <i>Definieren des Hochladens von Daten aus einer LDAP-Tabelle in eine LDAP-Importgruppe</i>.</p>
OnStart	Importeinträge	<p>SQL-Befehle vom Typ <code>OnStart</code> werden nach dem Schreiben von Daten in temporäre Datenbanktabellen und vor den SQL-Befehlen vom Typ <code>DataBind</code> ausgeführt. SQL-Befehle vom Typ <code>OnComplete</code> werden nach den SQL-Befehlen vom Typ <code>DataBind</code> ausgeführt. Die Befehle ermöglichen die Ausführung spezieller Operationen in der Alfabet-Datenbank vor oder nach dem Import von Daten in die Alfabet-Datenbank -Zieltabellen. Die SQL-Befehle können für folgende Zwecke definiert werden:</p> <ul style="list-style-type: none"> • Definition benutzerdefinierter Debuginformationen, die während der Ausführung des Imports in die Protokolldatei geschrieben werden. • Diese Konfiguration ist im Abschnitt <i>Konfigurieren von Protokollierungsparametern</i> beschrieben. • Manipulation der Alfabet-Datenbank oder der temporären Datenbanktabelle vor und/oder nach dem Datenimport. Die Definition von SQL-Befehlen für diesen Zweck wird in diesem Abschnitt beschrieben.

Typ	Position im ADIF-Schema	Beschreibung
OnComplete	Im- porteinträge	

Eines der wichtigsten Features der ADIF-Schnittstelle ist die flexible Konfiguration. Das ADIF-Importschema kann nicht nur dahingehend konfiguriert werden, dass es einfach Daten aus einer externen Datei liest und in die Alfabet-Datenbank einfügt, sondern Sie können ADIF auch so konfigurieren, dass es zusätzliche Operationen in der Alfabet-Datenbank, den temporären Datenbanktabellen, die während des Imports erzeugt werden, oder, im Falle des Imports aus einer externen Datenbank, in der externen Datenbanktabelle in verschiedenen Phasen der Verarbeitung des ADIF-Importschemas durchführt. SQL-Abfragen vom Typ DDL-Anweisung oder DML-Anweisung können in beliebiger Anzahl und Reihenfolge definiert werden.



Wenn SQL-Abfragen nicht ausreichen, um den gewünschten Import zu implementieren, können benutzerdefinierte DLLs bei der Software AG angefordert werden, um die Funktionalität bereitzustellen. Der benutzerdefinierte Code wird dem ADIF-Importschema über einen SQL-Befehl mit dem **Befehlstyp** `StoredProcedure` hinzugefügt. Die korrekte Einstellung der Konfigurationselemente und -attribute hängt vom benutzerdefinierten Code ab. Eine Beschreibung auf individueller Basis wird mit dem Code mitgeliefert.

`StoredProcedures` werden auf die gleiche Art und Weise ausgeführt wie DML-Anweisungen (in der Reihenfolge der Ausführung von SQL-Anweisungen, wie unten beschrieben).

Wenn Sie SQL-Befehle für ein ADIF-Importschema konfigurieren, müssen Sie die Ausführungsreihenfolge berücksichtigen. Die Elemente eines ADIF-Schemas werden in der folgenden Reihenfolge ausgeführt:

- Die SQL-Befehle vom Typ `OnActivate` werden zuerst ausgewertet, und die entsprechenden Elemente der ADIF-Schemata werden in den Ausführungsplan einbezogen oder davon ausgeschlossen.
- Alle SQL-Befehle vom Typ `OnStart` und dem **Befehlstyp** `DDLStatement`, deren Ziel die Alfabet-Datenbank ist, werden in der Reihenfolge ausgeführt, in der sie im ADIF-Importschema angegeben sind.
- Die Daten für alle Importeinträge werden in der Reihenfolge in die temporären Tabellen hochgeladen, die im ADIF-Importschema angegeben ist. Beim Import aus Datenbank-Importgruppen und LDAP-Importgruppen ist der Datenimport in die temporären Tabellen mit den folgenden Schritten verknüpft:
 - Die Verbindung mit der externen Datenbank wird hergestellt.
 - Alle DDL-Anweisungen von SQL-Befehlen des Typs `OnStart`, die die externe Datenbank zum Ziel haben, die in den Importeinträgen der Importgruppe konfiguriert ist, werden in der Reihenfolge ausgeführt, in der sie in der Importgruppe definiert sind.
 - Die Importeinträge werden wie folgt in der Reihenfolge verarbeitet, in der sie innerhalb der Datenbank-Importgruppe konfiguriert sind:
 - Alle DML-Anweisungen von SQL-Befehlen des Typs `OnStart`, die auf die externe Datenbank abzielen, werden in der Reihenfolge ausgeführt, in der sie im Importeintrag  angegeben sind.

- Der SQL-Befehl des Typs `DataUpload` wird ausgeführt, und Daten werden in die temporäre Tabelle hochgeladen.
- Alle DML-Anweisungen von SQL-Befehlen des Typs `OnComplete`, die auf die externe Datenbank abzielen, werden in der Reihenfolge ausgeführt, in der sie im Importeintrag  angegeben sind.
- Alle DDL-Anweisungen von SQL-Befehlen des Typs `OnComplete`, die die externe Datenbank zum Ziel haben, die in den Importeinträgen der Importgruppe konfiguriert ist, werden in der Reihenfolge ausgeführt, in der sie in der Importgruppe definiert sind.
- Alle Importeinträge werden wie folgt verarbeitet, unabhängig von ihrer Position inner- oder außerhalb einer Importgruppe:
 - Alle DML-Anweisungen von SQL-Befehlen vom Typ `OnStart` werden in der Reihenfolge ausgeführt, in der sie im Importeintrag  angegeben sind.
 - Der Datenimport in die Alfabet-Datenbank -Zieltabellen erfolgt über Elemente vom Typ **Attribut** , SQL-Befehle vom Typ `DataBind` und automatische Datenzuordnungsmechanismen des ADIF-Importprozesses.
 - Beziehungselemente werden verarbeitet.
 - Die DML-Anweisungen von SQL-Befehlen vom Typ `OnComplete` werden in der Reihenfolge ausgeführt, in der sie im Importeintrag  angegeben sind.
- Alle DDL-Anweisungen von SQL-Befehlen vom Typ `OnComplete`, die die für die Exporteinträge im ADIF-Exportschema konfigurierte Alfabet-Datenbank als Ziel haben, werden in der Reihenfolge ausgeführt, in der sie im ADIF-Exportschema definiert sind.

In diesem Abschnitt wird im Allgemeinen beschrieben, wie Sie im ADIF-Konfigurationsexplorer einen SQL-Befehl innerhalb eines Ordners **SQL-Befehle - OnStart** oder **SQL-Befehle - OnComplete** definieren. Des Weiteren werden zwei mögliche Anwendungsfälle für die Definition von SQL-Befehlen beschrieben.

- *Anwendungsfall: Ausfüllen leerer Eigenschaftswerte mit Standardwerten beim Importieren in Pflicht-Objektklasseneigenschaften*
- *Anwendungsfall: Definieren importbezogener benutzerdefinierter Eigenschaften*

Erzeugen eines SQL-basierten Berichts

So spezifizieren Sie einen SQL-Befehl, der in der Alfabet-Datenbank oder einer externen Zieldatenbank ausgeführt werden soll:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsoberfläche mit der rechten Maustaste auf den Ordner **SQL-Befehle - OnStart** oder **SQL-Befehle - OnComplete**, der beim Erzeugen eines Importeintrags  automatisch dem Explorer hinzugefügt wird.
- 2) Wählen Sie im Kontextmenü **SQL-Befehl erzeugen** aus. Dem Ordner wird ein neuer SQL-Befehl  als ein untergeordneter Knoten hinzugefügt.
- 3) Klicken Sie auf den neuen SQL-Befehl , und geben Sie im Attributfenster Folgendes an:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.

- **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
- **Befehlstyp:** Wählen Sie im Dropdown-Listefeld den Typ der SQL-Abfrage aus, die Sie definieren möchten. Es gibt folgende Auswahlmöglichkeiten:
 - `DMLStatement` für SQL-Datenmanipulationsanweisungen wie `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `MERGE`.
 - `DDLStatement` für SQL-Datendefinitionsanweisungen wie `CREATE`, `ALTER`, `DROP`, `TRUNCATE`, `RENAME`.



- DDL-Anweisungen sind nur zulässig, um neue Datenbanktabellen zu erzeugen, zu verändern oder zu verwerfen. Diese Befehle dürfen nicht auf Alfabet-Datenbank -Standardtabellen angewendet werden.
- Die ADIF-Importmechanismen sind darauf ausgelegt, ein hohes Maß an Sicherheit zu gewährleisten. Überprüfen Sie immer, ob ein Import mithilfe eines von ADIF bereitgestellten Importmechanismus durchgeführt werden kann, bevor Sie Ihre eigenen SQL-Befehle zum Durchführen des Imports definieren.
- **Ergebnistyp:** Wählen Sie `Undefined` aus.
- **Anwenden auf:** Wenn der SQL-Befehl für den Import aus einer externen Datenbanktabelle definiert ist, definieren Sie die Datenbank, an der der SQL-Befehl ausgeführt wird:
 - Lokal zur Ausführung in der Alfabet-Datenbank
 - extern zum Ausführen an der externen Datenbank



- Das Ausführen von SQL-Befehlen an externen Datenbanktabellen wird für den Datenimport nicht empfohlen. Externe Datenbanken sollten nur direkt in der Quelldatenbank manipuliert werden.
- **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Import auch dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Import im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
- **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.
- **Text:** Definieren Sie die Abfrage für den Bericht. Schreiben Sie die Abfrage entweder direkt in das Attributfeld, oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der Abfrage auf der Registerkarte **SQL-Text** des Editors zu öffnen.



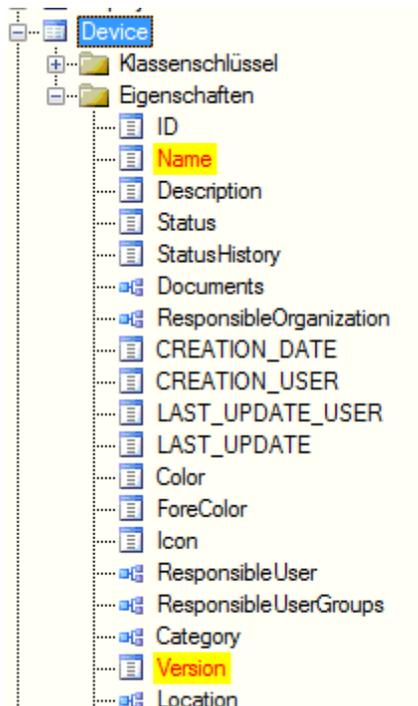
Die Abfragen, die an der externen Datenbank ausgeführt werden sollen, müssen mit der Syntax übereinstimmen, die für die externe Datenbank-/LDAP-Tabelle erforderlich ist.



Der Texteditor zur Definition von Abfragen in der ADIF-Konfigurationsbenutzeroberfläche bietet Hilfe zum Definieren von Abfragen in separaten Registerkarten. Weitere Informationen finden Sie unter *Definieren von SQL-Abfragen für SQL-Befehle* im Kapitel *Konfigurieren von ADIF-Schemata*.

Anwendungsfall: Ausfüllen leerer Eigenschaftswerte mit Standardwerten beim Importieren in Pflicht-Objektklasseneigenschaften

Einige Objektklasseneigenschaften von Objektklassen im Alfabet –Metamodell sind obligatorisch und müssen definiert werden, wenn neue Datensätze hinzugefügt werden. Obligatorische Eigenschaften erkennen Sie im untergeordneten Baum **Metamodell** des ADIF-Konfigurationsexplorers an ihrer gelb-roten Farbgebung:



Wenn Sie beim Import neue Einträge in der Alfabet-Datenbank erzeugen, müssen die obligatorischen Informationen für alle neuen Einträge importiert werden. Wenn der externe Datensatz, der importiert wird, die erforderlichen Informationen nicht bereitstellt, müssen die benötigten Werte über SQL-Befehle in die temporäre Tabelle geschrieben werden.

Während des Imports werden die Daten zuerst in die temporäre Tabelle geschrieben. Anschließend werden die SQL-Befehle vom Typ `OnStart` ausgeführt, bevor die Daten in der temporären Tabelle in die Zieldatenbanktable der Alfabet-Datenbank importiert werden.

Sie können z. B. einen SQL-Befehl `OnStart` definieren, mit dem die temporäre Tabelle aktualisiert wird, indem Sie einen Standardwert für alle leeren Felder in der Spalte für die Pflichteigenschaft festlegen.



Eine Importdatei mit Daten über Geräte stellt beispielsweise nicht für alle Geräte Versionsnummern bereit. Die Eigenschaft `Version` ist eine Pflichteigenschaft der Objektklasse `Device`.

Obligatorische
Versionsinformation wird
nicht überall bereitgestellt

	A	B	C	D
1	Name	Version	StartDate	End Date
2	ImportDevice1	4.5.9	02.01.2010	15.03.2015
3	ImportDevice2		15.06.2010	30.03.2012
4	ImportDevice3	2.0	01.04.2011	18.05.2018
5	ImportDevice4	3.1	24.08.2011	31.12.2014

Um die fehlenden Versionsinformationen zu der temporären Tabelle hinzuzufügen, die aus dem Import der oben angezeigten Daten entsteht, kann im Attribut **Text** eines SQL-Befehls des Typs **OnStart** folgende SQL-Abfrage definiert werden. Die Abfrage setzt die Version für alle Geräte, für die in den externen Daten keine Versionsinformationen zur Verfügung stehen, auf 1.0.

```
UPDATE TMP_DEVICE
SET Version = '1.0'
WHERE Version = NULL;
```

Anwendungsfall: Definieren importbezogener benutzerdefinierter Eigenschaften

Es kann hilfreich sein, die Informationen zum Import direkt in der Alfabet-Datenbank -Tabelle der Zielobjekte zu speichern. Diese Funktionalität kann beispielsweise in folgenden Situationen nützlich sein:

- Konfigurieren des Imports, sodass nur Objekte enthalten sind, die nicht als bereits in einem vorangegangenen Datenimport importiert markiert sind.
- Informieren des Anwenders über die Alfabet -Benutzeroberfläche darüber, ob und/oder wann zuletzt Daten importiert wurden.
- Verwenden des Attributs zum Definieren einer Workflow-Vorlage, mit dem nur dann ein Workflow gestartet wird, wenn das Attribut auf einen definierten Wert gesetzt ist. Das ADIF-Importschema kann außerdem dahingehend konfiguriert werden, dass die Workflows als Bestandteil des Importvorgangs automatisch gestartet werden.

Weiterführende Informationen finden Sie unter *Konfigurieren des automatischen Starts von Workflows während des Imports*.

- Verwendung des Attributs als Bedingung für die Ausführung von Aktionen über die Alfabet - Benutzeroberfläche (beispielsweise in Workflow-Vorbedingungen oder in Berichten, die nur Daten anzeigen, die über ADIF importiert wurden).

Die benutzerdefinierte Eigenschaft kann das Datum enthalten, an dem der Import durchgeführt wurde, oder einen booleschen Wert, der zwischen importierten und nicht importierten Daten unterscheidet.



Die folgenden Konfigurationsschritte sind erforderlich, um Informationen über den Import direkt in der Datenbanktabelle der importierten Objekte zu speichern:

- Der Objektklasse, für die Daten importiert werden, muss eine benutzerdefinierte Eigenschaft hinzugefügt werden. Meilensteinvorlagen müssen im Konfigurationstool Alfabet Expand konfiguriert werden. Informationen über das Erzeugen einer benutzerdefinierten Eigenschaft finden Sie im Abschnitt *Konfigurieren benutzerdefinierter Eigenschaften für geschützte oder öffentliche Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- Definieren Sie im Importeintrag , der den Import auslöst, eine leere Datenbankspalte in der temporären Datenbanktabelle und ordnen Sie diese der benutzerdefinierten Eigenschaft der Alfabet-Datenbank -Zieltabelle zu.
- Fügen Sie einen SQL-Befehl  mit einer UPDATE-Anweisung im Ordner **SQL-Befehle - OnStart** hinzu, die die Informationen über den Import in die Spalte für die benutzerdefinierten Eigenschaft der temporären Tabelle schreibt.



Sie müssen die Konfigurationsfunktionen in Alfabet Expand verwenden, um einer Datenbankta-
belle einer Alfabet -Objektklasse neue Eigenschaftsspalten hinzuzufügen. Eigenschaftsspalten
dürfen nicht mit einer DDL-Anweisung in den SQL-Befehlen des ADIF-Schemas hinzugefügt
werden. **Bestehende Alfabet-Datenbank -Standardtabellen dürfen nicht durch SQL-Be-
fehle über ADIF geändert werden.**

Konfigurieren der Ausführung des Importschemas in Abhängigkeit von aktuellen Parametern

ADIF-Importschemata können dafür konfiguriert werden, sich an die aktuellen Status von Objekten in der Alfabet-Datenbank anzupassen. Sie können Bedingungen für die Gesamtausführung von Teilen des ADIF-Importschemas definieren oder in der Befehlszeile der ADIF-Konsolenapplikation Bedingungen definieren, die die Auswahl des auszuführenden Datenobjekts festlegen; diese unterscheiden sich von Import zu Im-
port.

Konfigurieren der bedingten Ausführung von Teilen des Importschemas

Mit dem ADIF-Importschema können Sie die Ausführung von Teilen der Importdefinition an Bedingungen knüpfen. Sie können SQL-Befehle definieren, die die Alfabet-Datenbank daraufhin prüfen, ob Daten verfüg-
bar sind, und Importgruppen oder Importeinträge so konfigurieren, dass sie nur ausgeführt werden, wenn eine konfigurierte SQL-Abfrage, die in der Alfabet-Datenbank ausgeführt wird, entweder ein positives oder ein negatives Ergebnis zurückgibt.

Bedingungen können für die folgenden Elemente im ADIF-Importschema definiert werden:

- Datenbank-Importgruppe 
- Datei-Importgruppe 
- XML-Importgruppe 
- Importeintrag 

Wird eine Bedingung für ein Element angegeben, werden dieses Element und alle ihm untergeordneten Ele-
mente nur ausgeführt, wenn die Bedingung erfüllt ist.

So geben Sie eine Bedingung für die Ausführung eines Teils eines ADIF-Importschemas an:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf den Ordner **SQL-Befehle - OnActivate**, der dem Explorer automatisch hinzugefügt wird, wenn ein Element erzeugt wird.
- 2) Wählen Sie im Kontextmenü **SQL-Befehl erzeugen** aus. Dem Ordner **SQL-Befehle - OnActivate** wird ein neuer SQL-Befehl  als untergeordneter Knoten hinzugefügt.
- 3) Klicken Sie auf den neuen SQL-Befehl , bearbeiten Sie im Attributfenster optional folgende Attribute:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.

- **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
- **Anwenden auf** Wählen Sie `local` aus. SQL-Befehle vom Typ **OnActivate** können nur auf der Alfabet-Datenbank ausgeführt werden.
- **Ergebnistyp:** Wählen Sie im Dropdown-Listefeld eine der folgenden Optionen aus:
 - **PositiveCheck:** Wenn der Datensatz leer ist, der aus der Ausführung der Abfrage resultiert, die im Attribut **Text** definiert ist, wird das aktuelle Element des ADIF-Schemas nicht ausgeführt, und die im Attribut **Meldung** definierte Meldung wird in die Protokolldatei geschrieben.
 - **NegativeCheck:** Wenn der Datensatz nicht leer ist, der aus der Ausführung der Abfrage resultiert, die im Attribut **Text** definiert ist, wird das aktuelle Element des ADIF-Schemas nicht ausgeführt, und die im Attribut **Meldung** definierte Meldung wird in die Protokolldatei geschrieben.
- **Text:** Definieren Sie eine SQL-Abfrage mit einer `SELECT`-Anweisung, die einen Datensatz zurückgibt. Der resultierende Datensatz wird entsprechend den Einstellungen des Attributs **Ergebnistyp** geprüft, um festzulegen, ob das ADIF-Element ausgeführt wird. Schreiben Sie die SQL-Abfrage entweder direkt in das Attributfeld, oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.
- **Meldung:** Geben Sie einen Text ein, der in die Protokolldatei geschrieben wird, falls das ADIF-Element nicht ausgeführt wird.
- **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Import auch dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Import im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
- **Ist aktiv:** Wählen Sie `True` aus, um die Überprüfung zu aktivieren. Wählen Sie `False` aus, um die Überprüfung zu deaktivieren.

Die Bedingung für die Überprüfung kann zusammen mit Parametern definiert werden, die während der Laufzeit des ADIF-Imports festgelegt wurden.

Weitere Informationen hierzu finden Sie unter *Konfigurieren von Protokollierungsparametern*.

Konfigurieren des Imports in Abhängigkeit von Parametern

Der Import kann so konfiguriert werden, dass er auf Parametern basiert, die z. B. in der Befehlszeile der ADIF-Konsolenapplikation oder in den JSON-Nutzdaten des RESTful-Serviceaufrufs definiert wurden, wenn ein Import gestartet wird. So können Sie ein ADIF-Importschema für verschiedene ähnliche Importe verwenden, die sich nur in kleineren Aspekten voneinander unterscheiden.



Es ist nicht möglich, Parameter für ADIF-Importe zu definieren, die über die Benutzeroberfläche ausgeführt werden. Trotzdem können diese Importe mit den Standardwerten für die Parameter ausgeführt werden, wenn diese Standardwerte im ADIF-Importschema definiert sind. Wenn die Konfiguration eines ADIF-Schemas obligatorische Parameter enthält, wird die Ausführungsoption in der Funktionalität **ADIF-Prozessverwaltung** (`ADMIN_AdifJobs`) deaktiviert.

Folgende Informationen sind verfügbar:

- [Konfigurieren eines ADIF-Importschemas zur Verwendung von Parametern in Importdefinitionen](#)
- [Definieren von Parameterwerten bei der Ausführung von ADIF-Schemata](#)
- [Definieren von Parameterwerten während des Tests des ADIF-Schemas](#)

Konfigurieren eines ADIF-Importschemas zur Verwendung von Parametern in Importdefinitionen

Parameter müssen im ADIF-Schema definiert werden. Nur Parametern, die im ADIF-Schema definiert sind, können während der Ausführung des ADIF-Schemas Werte zugewiesen werden.

So definieren Sie die Parameter für ein ADIF-Schema:

- 1) Klicken Sie im Explorer auf das ADIF-Importschema, das Sie mit Parametern starten möchten.
- 2) Definieren Sie im Attributfenster die für Parameter relevanten Attribute:

- **Parameter – Abwärtskompatibilitätsmodus:** Stellen Sie sicher, dass der Parameter auf `False` eingestellt ist. Für neue ADIF-Schemata ist dies der Standardwert.



Für ADIF-Schemata, die vor Alfabet 10.4 definiert wurden, ist der Kompatibilitätsmodus auf `True` eingestellt. Wenn Sie den Kompatibilitätsmodus für diese Schemata zurücksetzen, müssen Sie Parameterdefinitionen zu den ADIF-Importeinträgen im Schema hinzufügen und die Art und Weise, wie Parameter in den SQL-Befehlen definiert werden, überprüfen.

Bitte beachten Sie: Für ADIF-Schemata, bei denen **Parameter – Abwärtskompatibilitätsmodus** auf `True` eingestellt ist und die Parameterdefinitionen enthalten, kann die Ausführung nicht über die Funktionalität **Prozesszeitplan** geplant werden.

- **Argument-Tabellenname:** Zu Beginn eines ADIF-Imports werden die Parameterwerte in eine temporäre Datenbanktabelle geschrieben. Der Name der temporären Datenbanktabelle kann im ADIF-Importschema konfiguriert werden. Standardmäßig wird sie `ADIF_ARGS` genannt.



Für jeden Parameter wird in der Datenbanktabelle eine Zeile mit zwei Spalten angezeigt:

- `ARG_NAME` zum Speichern des Parameternamens.
- `ARG_VALUE` zum Speichern des Parameterwerts.

- 3) Klicken Sie für jeden Parameter, den Sie in einem der Importeinträge im Schema verwenden möchten, mit der rechten Maustaste auf den untergeordneten Knoten **Parameter** des ADIF-Importschemas, und wählen Sie **Parameter erzeugen** aus der Dropdown-Liste aus.
- 4) Klicken Sie auf den neu hinzugefügten Knoten **@Parameter**, und definieren Sie im Attributfenster folgende Attribute:
 - **Name:** Definieren Sie einen eindeutigen Namen für den Parameter. Der Parametername muss mit "@" beginnen und darf keine anderen Sonderzeichen oder Leerzeichen enthalten. Er kann in den SQL-Befehlen und Attributen des ADIF-Importschemas verwendet werden und wird während der Ausführung des ADIF-Importschemas durch den aktuellen Wert des Parameters ersetzt.
 - **Parametertyp:** Wählen Sie den Datentyp des Parameterwerts aus der Dropdown-Liste aus.



Die Parametertypen "StringArray" und "ReferenceArray" können verwendet werden, wenn eine Abfrage definieren soll, dass eine Zeichenfolge oder der REFSTR eines referenzierten Objekts innerhalb eines zulässigen Bereichs von Zeichenfolgen- oder REFSTR-Werten liegen soll.

- Standardwert:** Definieren Sie optional einen Standardwert, der verwendet werden soll, wenn während der Ausführung des ADIF-Schemas kein Parameterwert angegeben wird. Klicken Sie auf die Schaltfläche **Durchsuchen**  rechts neben dem Attributfeld, um den Editor zum Definieren des Standardwerts zu öffnen. Beachten Sie Folgendes bezüglich der Spezifikation von Werten:

 - Geben Sie für die Definition von Datumsangaben das Datum in das Feld **Wert** in dem Format ein, das in der Landeseinstellung der Sprache definiert ist, die Sie derzeit zum Anzeigen der Alfabet Expand -Benutzeroberfläche verwenden, oder klicken Sie auf die Schaltfläche **Auswählen**, und wählen Sie ein Datum aus dem Kalender aus.
 - Wenn der Parameter ein Platzhalter für eine Zeichenfolge ist und in einer Abfrage in einer `WHERE`-Bedingung verwendet wird, die überprüft, ob ein Wert sich in einem Bereich befindet, muss der **Parametertyp** `StringArray` sein. Das Feld **Wert** im Editor des Attributs **Standardwert** zeigt für `StringArray`-Parameter ein Textfeld an. Definieren Sie den Bereich der zulässigen Werte im Textfeld, wobei jeder Wert in einer separaten Zeile stehen muss. In der Abfrage, die auf das Array verweist, muss der Parameter in Klammern eingefügt werden:


```
WHERE app.OBJECTSTATE IN (@appState)
```
 - Wenn ein Zeichenfolgenwert ein einzelnes Anführungszeichen enthält (z. B. in „O'Hara“), wird das einfache Anführungszeichen automatisch durch ein zweites einfaches Anführungszeichen geschützt, wenn der Parameterwert verarbeitet wird. Das Schützen eines einzelnen Anführungszeichens mit einem zweiten einfachen Anführungszeichen ist eine Voraussetzung für die Integration in SQL-Abfragen. Aus Sicherheitsgründen werden zweite einfache Anführungszeichen zusätzlich in Zeichenfolgen eingefügt, um SQL-Injection zu verhindern.
 - "%" kann als Platzhalter in Zeichenfolgen und Texten verwendet werden. Es ist nicht zulässig, einen Platzhalter in einem Wert eines Zeichenfolgenarrays zu definieren.
 - Boolesche Werte können aus einer Dropdown-Liste ausgewählt werden. Beachten Sie, dass Objekteigenschaften des Typs `Boolean`, die weder obligatorisch sind noch einen definierten Standardwert haben, entweder auf "True", "False" oder "NULL" gesetzt werden können. Wenn der Standardwert `False` ist, werden nur Eigenschaftswerte der Objektklasse zurückgegeben, die als "False" definiert sind, während NULL-Werte nicht zurückgegeben werden. Dieses Verhalten kann über die SQL-Befehle im ADIF-Schema gehandhabt werden.
 - Für `ReferenceArray`-Eigenschaften können keine Standardwerte definiert werden.
- Obligatorisch:** Wählen Sie `True` aus, wenn die ADIF-Ausführung mit einer Fehlermeldung fehlschlagen soll, wenn für die aktuelle Ausführung kein Wert definiert ist. Wählen Sie "False" aus, wenn die ADIF-Ausführung auch dann ausgeführt werden soll, wenn der Parameter für die aktuelle Ausführung nicht definiert ist. Der Parameter wird dann durch den im Attribut **Standardwert** definierten Wert ersetzt. Wenn kein Standardwert angegeben wird, wird der Parameter durch `NULL` ersetzt.



Bei der ADIF-Ausführung werden WHERE-Klauseln in SQL-Befehlen, die einen Parameter enthalten, nicht aus der Abfrage entfernt, wenn kein Wert angegeben wird. Um sicherzustellen, dass das Abfrageergebnis sinnvoll ist, wenn der Parameterwert während der ADIF-Ausführung nicht angegeben und der Parameter daher auf NULL gesetzt wird, sollten Sie die Abfrage so definieren, dass NULL-Werte akzeptiert werden. Um beispielsweise sicherzustellen, dass Ergebnisse zurückgegeben werden, wenn ein Vergleich eines Datums mit dem Startdatum einer Applikation ohne Ergebnis bleibt, sollte die WHERE-Klausel folgendermaßen verfasst werden:

```
WHERE APPLICATION.STARTDATE >= @StartDate OR @StartDate IS
NULL
```

5) Verwenden Sie je nach Bedarf die Parameter, die im Ordner **Parameter** in den ADIF-Importeinträgen des ADIF-Importschemas in den folgenden Einstellungen definiert sind:

- In den SQL-Abfragen, die für einen beliebigen SQL-Befehl definiert sind.
- Im Attribut **Meldung** von SQL-Befehlen vom **Ergebnistyp** `DebugMessage`.
- Im Attribut **Verbindungsstring** eines Datenbankimportsatzes.

Um einen Parameter einzuschließen, ersetzen Sie den tatsächlichen Wert in der Abfrage oder der Textzeichenfolge durch den Parameternamen. Beachten Sie, dass Parameter ohne einfache Anführungszeichen um den Parameternamen eingeschlossen werden sollen. Wenn der Parametertyp die Definition von einfachen Anführungszeichen um den Wert erfordert, wird dies von der Software während der Ersetzung des Parameternamens durch den Parameterwert gehandhabt.



Eine Abfrage, die beispielsweise den Zeichenfolgenparameter `@orga` verwendet, kann folgendermaßen geschrieben werden:

```
SELECT app.NAME, app.VERSION
FROM APPLICATION app, ORGAUNIT org
WHERE app.RESPONSIBLEORGANIZATION = org.REFSTR
AND org.NAME LIKE @orga;
```

Wenn er während der Laufzeit durch den Wert `Trade%` ersetzt wird, lautet die Abfrage:

```
SELECT app.NAME, app.VERSION
FROM APPLICATION app, ORGAUNIT org
WHERE app.RESPONSIBLEORGANIZATION = org.REFSTR
AND org.NAME LIKE 'Trade%';
```

Eine Debugmeldung, die im Attribut **Meldung** eines SQL-Befehls vom **Ergebnistyp** `DebugMessage` definiert ist und die während des Imports der Applikationsdaten in die Importprotokolldatei geschrieben werden soll, kann auch auf die aktuelle Organisation verweisen:

```
Import of applications with responsible organization @orga.
```

Während der Laufzeit wird sie konvertiert in:

```
Import of applications with responsible organization Trade%.
```

Die Parameter "StringArray" oder "ReferenceArray" müssen wie folgt in die Abfrage eingeschlossen werden:

```
WHERE APPLICATION.STATUS IN (@StringArray)
```

Definieren von Parameterwerten bei der Ausführung von ADIF-Schemata

Parameterwerte werden bei der Ausführung des ADIF-Imports oder -Exports während der Laufzeit definiert. Bei Parameternamen muss die Groß-/Kleinschreibung beachtet werden. Wenn Sie Parameter für die Ausführung des ADIF-Imports oder -Exports definieren, müssen Sie den Parameternamen genau so verwenden, wie er im ADIF-Schema angegeben ist, einschließlich des Zeichens "@" am Anfang. Beachten Sie Folgendes bezüglich der Definition von Werten für die Parameter:

- Wenn ein Zeichenfolgenwert ein einzelnes Anführungszeichen enthält (z. B. in „O'Hara“), wird das einfache Anführungszeichen automatisch durch ein zweites einfaches Anführungszeichen geschützt, wenn der Parameterwert verarbeitet wird. Das Schützen eines einzelnen Anführungszeichens mit einem zweiten einfachen Anführungszeichen ist eine Voraussetzung für die Integration in SQL-Abfragen. Aus Sicherheitsgründen werden zweite einfache Anführungszeichen zusätzlich in Zeichenfolgen eingefügt, um SQL-Injection zu verhindern.
- Parameterwerte für Datentypen wie Zeichenfolgen oder Datumswerte werden ohne einfache Anführungszeichen am Anfang und am Ende definiert. Wenn in der Abfrage für den Datentyp einfache Anführungszeichen erforderlich sind, werden diese automatisch von den ADIF-Mechanismen hinzugefügt.
- Boolesche Werte müssen definiert werden als 1 (True) oder 0 (False):

```
@MyBooleanValue 1
```

- Werte für "StringArray" und "ReferenceArray" müssen durch "\r\n" getrennt werden:

```
@MyArrayValue First\r\nSecond
```

Ob und wie Befehlszeilenparameter zur Laufzeit definiert werden können, hängt von der Art und Weise ab, wie das ADIF-Schema ausgeführt wird:

- **Starten von ADIF über die Konsolenapplikation**

Um Parameter in der Befehlszeile der ADIF-Konsolenapplikation anzugeben, starten Sie die Konsolenapplikation mit der Befehlszeilenoption `<-<Parametername> <Parameterwert>`. Für jeden Parameter muss eine Befehlszeilenoption angegeben werden.



Um die Parameter `@lastupdate` und `@orga` festzulegen, kann die Befehlszeile beispielsweise wie folgt definiert werden:

```
ADIF_Console.exe -import -msalias Alfabet -alfaLoginName
UserName -alfaLoginPassword pw1234 -scheme AppImport -
importfile C:\Import\Applications.zip -@lastupdate 09/06/2018
-@orga ITGroup
```



Allgemeine Informationen zum Starten von ADIF über die Konsolenapplikation finden Sie unter [Ausführen von ADIF über ein Befehlszeilen-Tool](#).

- **Starten von ADIF über einen RESTful-Serviceaufruf an die RESTful API der Alfabet-Web-Applikation**

Parameter werden in den JSON-Nutzdaten der Anfrage in dem Feld `UserArgs` definiert, das ein JSON-Objekt mit einem Feld für jeden Parameter in folgendem Format zurückgibt: `"UserArgs": {" arg1name ":" arg1value "," arg2name ":" arg2value "}`. Der Feldname muss mit dem Variablennamen übereinstimmen, und der Feldwert definiert den Variablenwert für die aktuelle Ausführung des ADIF-Importschemas.



Um die Parameter `@lastupdate` und `@orga` festzulegen, können die JSON-Nutzdaten der Anfrage beispielsweise wie folgt definiert werden:

```
{
  "Scheme": "AppImport",
  "UserArgs": {"@lastupdate": "09/06/2018", "@orga": "ITGroup"},
  "Verbose": false,
  "Synchron": true
}
```



Wenn die Spezifikation der Parameter im Aufruf nicht korrekt ist, wird der ADIF-Prozess gestartet, die Ausführung kann aber fehlschlagen. Es werden an den RESTful-Serviceaufruf keine Informationen über den Erfolg der ADIF-Prozessausführung zurückgegeben. Der Rückgabewert des RESTful-Serviceaufrufs informiert über den letzten erfolgreichen Ausführungsschritt, also dem Start des ADIF-Prozesses. Der Erfolg der ADIF-Prozessausführung kann dann z. B. über die Funktionalität **ADIF-Prozessverwaltung** überprüft werden.

Weitere Informationen über die Funktionalität "ADIF-Prozessverwaltung" finden Sie unter [Ausführen und Überprüfen von ADIF über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" in der Alfabet -Benutzeroberfläche.](#)

- **Starten von ADIF über die Funktionalität "ADIF-Prozessverwaltung" in der Alfabet-Benutzeroberfläche**

Die Funktionalität "ADIF-Prozessverwaltung" bietet keinen Mechanismus zum Festlegen von Parameterwerten. Wenn ein ADIF-Prozess über die **ADIF-Prozessverwaltung** gestartet werden soll, darf er keine Parameter enthalten. Wenn das ADIF-Schema Parameterdefinitionen enthält, kann es über die Funktionalität **ADIF-Prozessverwaltung** nur mit den im ADIF-Schema für die Parameter definierten Standardwerten gestartet werden. Das Schema darf keine obligatorischen Parameterdefinitionen enthalten.



Allgemeine Informationen zum Starten von ADIF über die Funktionalität **ADIF-Prozessverwaltung** finden Sie unter [Ausführen und Überprüfen von ADIF über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" in der Alfabet -Benutzeroberfläche.](#)

- **Starten von ADIF über ein Ereignis**

Wenn Sie den ADIF-Prozess über ein Ereignis starten, können Parameterwerte während der Ausführung des Ereignisses über eine in der Ereignisvorlage definierte Abfrage übergeben werden. In den Attributen der Ereignisvorlage müssen die folgenden Attribute festgelegt werden:

- **Name der Variablen:** Wenn das ADIF-Schema Parameterdefinitionen enthält, müssen die Namen der Parameter als durch Kommas getrennte Liste in dieses Attribut geschrieben werden. Bei der Spezifikation muss zwischen Groß- und Kleinschreibung unterschieden werden. Die Werte für die Parameter müssen dann mit den Attributen **Werte für Variablen über Abfrage / Werte für Variablen über Abfrage als Text** angegeben werden.
- **Werte für Variablen über Abfrage / Werte für Variablen über Abfrage als Text:** Die Abfrage muss die Werte, die im Attribut **Namen der Variablen** für die Parameter definiert sind, in der gleichen Reihenfolge zurückgeben. Die Abfragen können entweder über Native-SQL-Abfragen oder über Alfabet -Abfragen definiert werden. Verwenden Sie das Attribut **Werte für**

Variablen über Abfrage, um eine Alfabet -Abfrage zu definieren. Verwenden Sie das Attribut **Werte für Variablen über Abfrage als Text**, um eine Native-SQL-Abfrage zu definieren. In der Abfrage können Parameter der Alfabet -Abfragesprache verwendet werden. Das über den Parameter `BASE` der Alfabet -Abfragesprache zurückgegebene Basisobjekt hängt davon ab, wie ein Ereignis ausgelöst wurde. Für Ereignisse, die über einen Workflow oder Assistenten ausgelöst wurden, gibt `BASE` den `REFSTR` des Objekts zurück, an dem der Anwender im Schritt des Assistenten oder Workflows gearbeitet hat, der das Ereignis auslöst. Bei Ereignissen, die über ein REST API-Aufrufereignis ausgelöst werden, entspricht das `BASE`-Objekt dem Objekt für das Ereignis, welches das Ereignis auslöst.

Die Abfrage muss einen Datensatz zurückgeben, dessen Spaltennamen mit den Parameternamen, die mit dem Attribut **Name der Variablen** definiert wurden, identisch sind.



Wenn **Name der Variablen** beispielsweise als `@AppName`, `@AppVersion` definiert ist, kann die Abfrage folgendermaßen definiert werden:

```
SELECT REFSTR, NAME AS '@AppName', VERSION AS
'@AppVersion'
FROM APPLICATION
WHERE REFSTR = @BASE
```



Allgemeine Informationen zum Starten von ADIF über die RESTful-Services von Alfabet finden Sie im Referenzhandbuch *Alfabet RESTful API*.

- **Starten von ADIF über eine Schaltfläche im Filterbereich eines konfigurierten Berichts oder in der Symbolleiste einer Alfabet-Ansicht**

Wenn ein ADIF-Prozess über eine Schaltfläche im Filterbereich eines konfigurierten Berichts oder in der Symbolleiste einer Objektansicht gestartet wird, können die Werte für die Parameterdefinition im konfigurierten Bericht wie folgt festgelegt werden:

- Wenn die Schaltfläche für eine Objektansicht oder einen konfigurierten Bericht definiert ist, der über das Attribut **Auf Klasse anwenden** einer Basisobjektklasse zugeordnet ist, werden die Informationen über das Basisobjekt, für das die Objektansicht oder der konfigurierte Bericht geöffnet wird, als `@BASE` zurückgegeben. Wenn das ADIF-Schema über eine Parameterdefinition verfügt, deren **Name** auf `@BASE` und deren **Parametertyp** auf `Reference` eingestellt ist, wird dieser Parameter bei der Ausführung des ADIF-Schemas durch den `REFSTR` des aktuellen Objekts ersetzt.
- Wenn die Schaltfläche für einen konfigurierten Bericht mit Filterfeldern definiert ist, werden die in den Filterfeldern festgelegten Werte als Parameterwerte an das ADIF-Schema übergeben, wenn der ADIF-Prozess über die Schaltfläche gestartet wird. Im ADIF-Schema muss der **Name** des Parameters mit dem Attribut **Name** des Filterfelds identisch sein, und der **Parametertyp** muss mit dem vom Filterfeld zurückgegebenen Datentyp identisch sein.

Beachten Sie, dass der Name des Filterfelds mit einem „@“ beginnen muss. Standardfilterfelder, die standardmäßig in Berichten des Typs `Query`, die auf einer Alfabet -Abfrage basieren, automatisch generiert werden, beginnen mit einem Doppelpunkt anstelle von „@“ und müssen manuell angepasst werden.

Bei Vergleichen zwischen Filterfeldern und Parameternamen wird zwischen Groß- und Kleinschreibung unterschieden.

Wenn die Filter eines konfigurierten Berichts bei der Ausführung des ADIF-Prozesses nicht festgelegt sind, wird kein Wert für den ADIF-Schemaparameter bereitgestellt. Daher sollten ADIF-Schemaparameter, die mit Werten aus Filterfeldern gefüllt werden, die im konfigurierten Bericht nicht obligatorisch sind, im ADIF-Schema nicht als obligatorisch definiert werden. Entweder sollte ein Standardwert für die Parameter definiert werden, oder in den SQL-Befehlen des ADIF-Schemas sollte die Verarbeitung von `NULL`-Werten berücksichtigt werden.



Allgemeine Informationen zum Starten von ADIF über eine Schaltfläche finden Sie unter [Ausführen von ADIF über eine Schaltfläche in der Alfabet-Benutzeroberfläche](#).

Definieren von Parameterwerten während des Tests des ADIF-Schemas

Wenn Sie ein ADIF-Schema mit dem ADIF-Debugger debuggen, kann keine Befehlszeile definiert werden. Zum Festlegen von Parametern für Tests können Parameterwerte für einen Test in das ADIF-Schema geschrieben werden.

So legen Sie Parameterwerte für Tests im ADIF-Schema fest:

- 1) Klicken Sie im ADIF-Explorer auf das ADIF-Schema, für das Sie Parameterwerte angeben möchten.
- 2) Legen Sie im Attributfenster des ADIF-Schemas die folgenden Attribute fest:

- **Debug-Argumente:** Definieren Sie die Werte für die verwendeten Parameter in einer kommagetrennten Zeichenfolge. Das Format der Zeichenfolge muss so aussehen: `<Parametername>=<Wert>`. Bitte beachten Sie, dass die Werte nicht in einfache Anführungszeichen geschrieben werden dürfen. Wenn in der Abfrage für den Parameterdatentyp, z. B. für Zeichenfolgen, einfache Anführungszeichen erforderlich sind, werden diese automatisch gesetzt, wenn der Wert in der Abfrage ersetzt wird.

Obligatorische Parameter müssen in den **Debug-Argumenten** definiert werden, um das ADIF-Schema über den ADIF-Debugger zu testen. Wenn der Parameter nicht obligatorisch ist und kein Wert für den Parameter in den **Debug-Argumenten** definiert ist, verwendet der ADIF-Debugger den für den Parameter definierten Standardwert. Wenn kein Standardwert definiert ist, wird der Parameter durch `NULL` ersetzt.



Um zum Beispiel den Import von Applikationen anzugeben, bei dem die für die Applikation verantwortliche Organisation und das letzte Aktualisierungsdatum für die Applikationsdaten Variablen sind, kann das Attribut **Debug-Argumente** wie folgt angegeben werden:

```
@lastupdate=09/10/2012,@orga=ITGroup
```

Konfigurieren von Protokollierungsparametern

Während des Imports über die ADIF-Konsolenanwendung werden Protokollnachrichten in folgende Speicherorte geschrieben:

- Eine temporäre Datenbanktabelle in der Alfabet-Datenbank.
- Die Protokolldatei der ADIF-Konsolenanwendung. Standardmäßig ist dies die Protokolldatei `ADIF_Console.log` im Arbeitsverzeichnis der ADIF-Konsolenanwendung.



Wenn die ADIF-Konsolenapplikation asynchron im Remote-Modus ausgeführt wird, werden keine sich auf den Import beziehenden Informationen in die Protokolldatei der ADIF-Konsolenapplikation geschrieben. Die Konsolenapplikation überträgt die Daten an die Alfabet-Server und beendet den Prozess nach erfolgreicher Übertragung der Daten.

Protokollierungsinformationen werden im folgenden Format in eine Protokolldatei geschrieben:

```
<date and time> <message type> <message text>
```



Beispiel:

```
2010-11-30T10:43:24.31Z DEBUG_INFO Query returns 314 records
```

Protokollmeldungen werden in der Sprache in die Protokolldatei geschrieben, die in den Landeseinstellungen der Alfabet-Datenbank definiert ist. Wenn die ADIF-Konsolenanwendung mit einem Serveralias ausgeführt wird, wird die Landeseinstellung verwendet, die in der Konfiguration von Alfabet als Standard definiert ist. Wird die ADIF-Konsolenanwendung mit einem Remotealias ausgeführt, wird die Landeseinstellung der Remotealias-Konfiguration verwendet.



Mehr Informationen über das Konfigurieren von Landeseinstellungen für die Alfabet-Datenbank finden Sie unter *Geben Sie die für Ihr Unternehmen relevanten Landeseinstellungen an.* im Kapitel *Lokalisierung und Mehrsprachenunterstützung für die Alfabet -Benutzeroberfläche* des Referenzhandbuchs *Konfigurieren von Alfabet mit Alfabet Expand*.

Informationen über die Definition der Landeseinstellungen des Remote-Alias finden Sie in Abschnitt *Konfigurationsattribute für die Alfabet -Komponenten* im Referenzhandbuch *Systemadministration*.

Der Zeitstempel richtet sich nach UTC-Zeit (Coordinated Universal Time) und kann sich daher von der Zeit in Ihrer Zeitzone unterscheiden. Der Zeitstempel wird nach ISO 8601 im kombinierten Format aus Datum und Uhrzeit wie folgt geschrieben: "Jahr-Monat-TagTStunden:Minuten:SekundenZ" geschrieben.

Der Nachrichtentyp kann wie folgt lauten:

- **FEHLER:** Es ist ein Fehler aufgetreten. Die Nachricht beschreibt die Art des Fehlers.
- **WARNUNG:** Es sind Probleme bei der Ausführung des Hilfsprogramms aufgetreten, die nicht so schwerwiegend sind wie ein Fehler. Der Prozess wurde ausgeführt, das Ergebnis sollte aber überprüft werden. Die Nachricht beschreibt das Problem.
- **INFO:** Enthält Informationen zur normalen Ausführung des Hilfsprogramms.
- **DEBUG_INFO:** Enthält Informationen über den Importvorgang. Diese Informationen werden auch in der Alfabet-Datenbank in einer temporären Datenbanktabelle gespeichert.



In der Debug-Information wird die Anzahl an Datensätzen zurückgegeben, die während des Importprozesses geändert wurden. Diese Zahl kann höher ausfallen als die Anzahl der direkt vom Importprozess betroffenen Objekte. Wenn zum Beispiel das ADIF-Schema die Löschung von Objekten aus der Alfabet-Datenbank auslöst, werden auch als abhängige Objekte definierte Objekte gelöscht oder geändert. Beispielsweise müssen Referenzen auf die gelöschten Objekte aus den Eigenschaften anderer Objekte oder aus der RELATIONS-Tabelle gelöscht werden. Die **DEBUG_INFO** gibt die Gesamtanzahl an Objekten zurück, die in der Datenbank geändert, erzeugt oder gelöscht wurden, wobei es sich für jedes gelöschte Objekt um mehrere Objekte handeln kann.

Während des Importvorgangs können E-Mails mit Protokollinformationen in einem definierten Zeitintervall an eine konfigurierbare E-Mail-Adresse gesendet werden.



Die E-Mail ist als Alive-Meldung des Systems gedacht, mit der der Empfänger über die Verfügbarkeit eines aktiven ADIF-Prozesses informiert wird. Der Inhalt der E-Mail ist auf die eine Zeile beschränkt, die in dem Moment in die Protokolldatei geschrieben wird, in dem die E-Mail generiert wird, und ist daher nicht für das Debuggen geeignet.

Das Speichern von Protokollinformationen und das Senden von E-Mails mit prozessbezogenen Informationen kann entweder im ADIF-Importschema oder in der Befehlszeile konfiguriert werden, wenn die ADIF-Konsolenanwendung gestartet wird. Wenn eine Konfiguration sowohl im ADIF-Importschema als auch in der Befehlszeile vorgenommen werden kann, überschreibt die Spezifikation über die Befehlszeilenoptionen die Konfiguration im ADIF-Importschema.

Die folgende Tabelle bietet eine Übersicht über die Konfigurationsoptionen:

Konfigurierbares Protokollierungsverhalten	Attribut des ADIF-Schemas	Befehlszeilenoption	Beschreibung
Zeitintervall für das Senden von E-Mails mit aktuellen Protokollinformationen	Debug-Heart-Beat	<code>-heartbeat</code> <Zeit in Minuten>	Geben Sie das Zeitintervall beim Senden von E-Mails mit der aktuellen Protokollmeldung der ADIF-Konsolenanwendung während des Imports in Minuten an. Der Debug-Heart-Beat ist standardmäßig auf -1 gesetzt; das bedeutet, dass keine E-Mails gesendet werden. HINWEIS: Wenn der Importvorgang abgeschlossen ist, bevor das Zeitintervall zum ersten Mal verstrichen ist, wird keine E-Mail gesendet.
E-Mail-Adresse des Empfängers der Protokolldaten	Empfänger-E-Mail	<code>-recipient-mail</code> <E-Mail-Adresse>	Geben Sie die E-Mail-Adresse der Person an, die die aktuelle Protokollmeldung der ADIF-Konsolenanwendung per E-Mail im konfigurierten Zeitintervall erhalten soll.
Als Absenderadresse in den E-Mails mit Protokolldaten verwendete E-Mail-Adresse	Absender-E-Mail	<code>-sendermail</code> <E-Mail-Adresse>	Geben Sie die E-Mail-Adresse an, die als Absenderadresse für E-Mails dienen soll, die mithilfe der ADIF-Konsolenanwendung gesendet werden.
Name der Protokolldatei der ADIF-Konsolenanwendung	-	<code>-logfile</code> <Dateiname>	Geben Sie den Namen der Protokolldatei der ADIF-Konsolenanwendung an. Zulässige Dateierweiterungen sind LOG und TXT. Der Standardname lautet <code>ADIF_Console.log</code> .

Konfigurierbares Protokollierungsverhalten	Attribut des ADIF-Schemas	Befehlszeilenoption	Beschreibung
Speicherort der Protokolldatei in der ADIF-Konsolenanwendung	-	<code>-logpath <Pfad></code>	Geben Sie den Pfad zur Protokolldatei in der ADIF-Konsolenanwendung an. Standardmäßig wird die Protokolldatei im Arbeitsverzeichnis der ADIF-Konsolenanwendung gespeichert. Der ADIF-Konsolenanwendungsprozess muss über Schreibberechtigungen für das angegebene Verzeichnis verfügen.
Menge der in die Protokolldatei geschriebenen Daten	-	<code>-logverbose</code>	Ist diese Option eingestellt, werden alle Informationen über den Prozess in die Protokolldatei geschrieben. Ist diese Option nicht eingestellt, werden nur Fehlermeldungen und Informationen über den Prozessstart und das Prozessende in die Protokolldatei geschrieben.
Entfernen alter Protokollmeldungen aus der Protokolldatei	-	<code>-logclear <Anzahl Tage></code>	<p>Wenn <code>-nologappend</code> festgelegt ist, wird jedes Mal eine neue Protokolldatei erzeugt, wenn das Hilfsprogramm mit derselben Spezifikation von <code>-logfile</code> und <code>-logpath</code> verwendet wird. Der Name der Protokolldatei erhält eine Erweiterung, die aus einem Zeitstempel besteht, der die aktuelle UTC-Zeit angibt.</p> <p>Wenn der Parameter <code>-nologappend</code> nicht festgelegt ist, werden die Protokollierungsinformationen jedes Mal an die vorhandene Protokolldatei angefügt, wenn das Hilfsprogramm verwendet wird.</p> <p>HINWEIS: Zur Beschränkung der Dateigröße können Sie die Option <code>-logclear</code> definieren, um alte Protokollmeldungen zu löschen.</p>
Erstellen einer neuen Protokolldatei für jeden Prozess gestartet	-	<code>-nologappend</code>	<p>Diese Option kann nur verwendet werden, wenn <code>-nologappend</code> nicht festgelegt ist. Wenn <code>-nologappend</code> festgelegt ist, wird die Einstellung <code>-logclear</code> ignoriert.</p> <p>Während der Protokollierung wird die Protokolldatei auf Protokollmeldungen überprüft, die einen Zeitstempel haben, der älter ist als die mit <code>-logclear</code> angegebene Anzahl an Tagen. Diese Meldungen werden dann gelöscht.</p> <p>HINWEIS: Der Prüfvorgang kann zu Leistungseinbußen führen.</p>

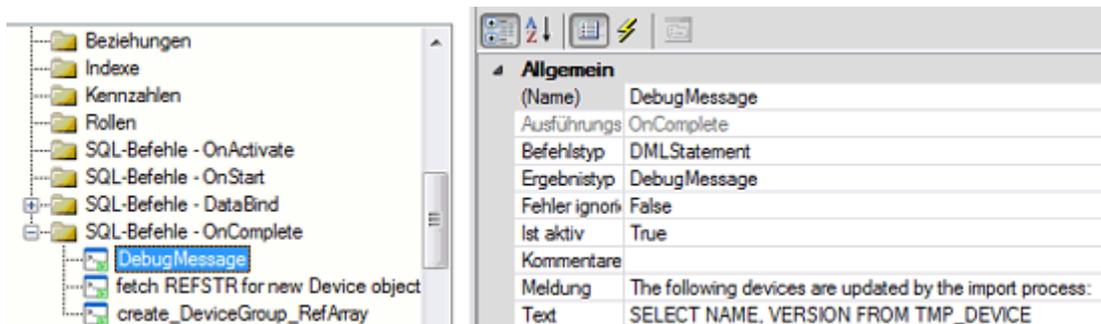
Konfigurieren des Inhalts von Protokollmeldungen

ADIF-Importschemata können so konfiguriert werden, dass sie die benutzerdefinierten Protokollmeldungen und Datensätze enthalten, die während der Ausführung des ADIF-Importschemas mithilfe von SQL-Befehlen in den Protokolldateien gefunden wurden. Inhalte von Protokolldateien können dem ADIF-Importschema mithilfe von SQL-Befehlselementen hinzugefügt werden. SQL-Befehlselemente für Debugmeldungen sind in jedem Ordner **SQL-Befehle - <Aktionstyp>** zu finden, mit Ausnahme des Ordners **SQL-Befehle - DataBind**.

Während der Ausführung des Imports werden die SQL-Befehle im ADIF-Importschema in der Reihenfolge ausgeführt, in der sie durch die Position des Befehls in der XML-Definition angegeben sind. Wenn der SQL-Befehl an der Reihe ist, wird in der Protokollmeldung die Meldung angezeigt, die im SQL-Befehlselement definiert ist und, falls angegeben, das Ergebnis der SQL-Abfrage als Datensatz.



In einem ADIF-Importschema, das Daten über Geräte aus einer XML-Datei importiert, werden Informationen über die Anzahl an Geräten, die während des Imports in der Alfabet-Datenbank aktualisiert werden, während des Importprozesses in die Protokolldatei geschrieben. Ein SQL-Befehl vom **Ergebnistyp DebugDataSet** wird dem Ordner **SQL-Befehle - OnComplete** des Importeintrags  hinzugefügt, der den Import von Gerätedaten aus der XML-Datei auslöst:



Allgemein	
(Name)	DebugMessage
Ausführungs	OnComplete
Befehlstyp	DMLStatement
Ergebnistyp	DebugMessage
Fehler ignoriert	False
Ist aktiv	True
Kommentare	
Meldung	The following devices are updated by the import process:
Text	SELECT NAME, VERSION FROM TMP_DEVICE

Im Eigenschaftsfenster wird das Attribut **Ergebnistyp** auf `DebugDataSet` festgelegt. Der Datensatz, der in der Protokolldatei angezeigt werden soll, wird per SQL-Abfrage festgelegt, die im **Text** definiert ist. Die im Beispiel definierte SQL-Abfrage findet Name und Versionsnummer der Geräte in der temporären Tabelle, die an ein vorhandenes Objekt in der Alfabet-Datenbank gebunden sind.

Vor dem Datensatz soll in der Protokolldatei ein Text angezeigt werden, der den Datensatz erläutert. Daher wird dem Attribut **Meldung** ein erklärender Text hinzugefügt.

Wenn der Import ausgeführt wird, werden der Text **Meldung** und der aus der SQL-Abfrage resultierende Datensatz der Protokolldatei hinzugefügt, wenn der SQL-Befehl in der Verarbeitungsreihenfolge des ADIF-Importschemas an der Reihe ist.

```
2011-04-26T15:41:08.333Z DEBUG_INFO The following devices are updated by the import process:
NAME|VERSION
Linux eBank Server1
Windows eBank App-Server A1
Windows eBank App-Server B1
```



Debugmeldungen und Datensätze können zu Beginn des Imports mithilfe von Variablen auf die aktuellen Importbedingungen verweisen, die von Befehlszeilenargumenten der `ADIF_Console.exe` abhängig sind.

Weitere Informationen über das Verwenden von Variablen, um auf Bedingungen zu verweisen, die in der Befehlszeile definiert sind, finden Sie unter *Konfigurieren der Ausführung des Importschemas in Abhängigkeit von aktuellen Parametern*.

Einfügen einer statischen Textmeldung

So fügen Sie der Protokolldatei eine statische Textmeldung hinzu:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf den Ordner **SQL-Befehle - <Aktionstyp>**, der zu jenem Zeitpunkt in der Reihenfolge der Befehle ausgeführt wird, zu dem die Textmeldung angezeigt werden soll.
- 2) Wählen Sie im Kontextmenü **SQL-Befehl erzeugen** aus. Dem Ordner **SQL-Befehle - <Aktionstyp>** wird ein neuer SQL-Befehl  als untergeordneter Knoten hinzugefügt.
- 3) Klicken Sie auf den neuen SQL-Befehl , bearbeiten Sie im Attributfenster optional folgende Attribute:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
 - **Ergebnistyp:** Wählen Sie `DebugMessage` aus.
 - **Meldung:** Geben Sie einen Text ein, der an jener Position in die Protokolldatei geschrieben wird, an der dieser SQL-Befehl ausgeführt wird.
 - **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.

Einfügen eines Datensatzes

So beziehen Sie Daten aus der Datenbank in die Protokolldatei ein:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf den Ordner **SQL-Befehle - <Aktionstyp>**, der zu jenem Zeitpunkt in der Reihenfolge der Befehle ausgeführt wird, zu dem die Textmeldung angezeigt werden soll.
- 2) Wählen Sie im Kontextmenü **SQL-Befehl erzeugen** aus. Dem Ordner **SQL-Befehle - <Aktionstyp>** wird ein neuer SQL-Befehl  als untergeordneter Knoten hinzugefügt.
- 3) Klicken Sie auf den neuen SQL-Befehl , und geben Sie im Attributfenster Folgendes an:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
 - **Ergebnistyp:** Wählen Sie `DebugDataSet` aus.
 - **Meldung:** Geben Sie einen Text ein, der an jener Position in die Protokolldatei geschrieben wird, an der dieser SQL-Befehl ausgeführt wird, und vor dem Datensatz, der aus der definierten SQL-Abfrage resultiert.

- **Text:** Definieren Sie eine SQL-Abfrage mit einer `SELECT`-Anweisung, die einen Datensatz zurückgibt. Der resultierende Datensatz wird an der Position in die Protokolldatei geschrieben, die die Ausführung des SQL-Befehls angibt. Wenn ein Text auch in dem Attribut **Meldung** definiert ist, wird dieser Text vor dem Datensatz in die Protokolldatei geschrieben. Schreiben Sie die SQL-Abfrage entweder direkt in das Attributfeld, oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.
- **Meldung:** Geben Sie einen Text ein, der als Einführung in den Datensatz in die Protokolldatei geschrieben werden soll.
- **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Import auch dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Import im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
- **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.

So legen Sie die Parameter im ADIF-Importschema fest:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche auf das Importschema, das Sie konfigurieren möchten. Die Attribute des ADIF-Importschemas werden im Attributfenster auf der rechten Seite angezeigt.
- 2) Legen Sie im Attributfenster die Parameter, die in der Tabelle darüber aufgelistet sind, als anwendbar fest.

Konfigurieren der Importhistorie

In Alfabet können Objektklassen so konfiguriert werden, dass sie eine Historie haben.

Informationen darüber, wie Historieninformationen in der Datenbank gespeichert werden, finden Sie unter [Speicherung der Überwachungshistorie](#) im Kapitel [Das Alfabet -Metamodell in der Alfabet -Datenbank](#).

Mithilfe des ADIF-Importschemas können Sie die Speicherung der Überwachungshistorie während des ADIF-Imports konfigurieren.



Die ADIF-Importdefinitionen sollten niemals die Überwachungstabellen beinhalten. Das Ändern der Überwachungstabelle für ein Objekt kann zu schwerwiegenden Datenbankinkonsistenzen führen und möglicherweise einen Fehler verursachen.

Bereinigen der Überwachungstabellen

Manchmal wird eine Änderung an einem Objekt vorgenommen, die den Wert einer Eigenschaft mit dem bereits in der Datenbanktabelle vorhandenen Wert überschreibt. Wird beispielsweise die Eigenschaft „Status“ eines Objekts auf den Wert „Plan“ gesetzt, obwohl der Wert bereits „Plan“ lautete, so handelt es sich dabei aus logischer Perspektive nicht um eine Änderung, technisch betrachtet wird aber eine redundante `UPDATE`-Anweisung ausgeführt. Wird das Objekt auf diese Weise geändert, wird der Historientabelle eine neue Zeile hinzugefügt. Diese Zeile gibt an, dass eine Änderung vorgenommen wurde, der Eintrag in der Historientabelle ist jedoch obsolet, da alle Objektklasseneigenschaften gleich bleiben.

Mithilfe der ADIF-Benutzeroberfläche kann das Importschema so konfiguriert werden, dass überflüssige Zeilen automatisch aus der Überwachungstabelle gelöscht werden. Die Konfiguration geschieht per **Importeintrag**, da die Einstellung nur für Klassen erforderlich ist, für die eine Überwachungstabelle vorhanden ist.



Es wird empfohlen, eine Bereinigung der Historientabelle für alle Objektklassen zu konfigurieren, die so konfiguriert sind, dass sie eine Historie haben. Andernfalls werden Anwendern, die die Historie über die Alfabet-Benutzeroberfläche betrachten, nutzlose Informationen angezeigt und die Größe der Alfabet-Datenbank erhöht sich beträchtlich.

- 1) Klicken Sie im Explorer auf den **Importeintrag**, für den Sie Überwachung konfigurieren möchten.
- 2) Setzen Sie im Attributfenster das Attribut **Historie-Tabelle komprimieren** auf `True`.

Informationen über den Importanwender

Der Anwender, der den ADIF-Import auslöst, wird als jener Anwender definiert, der die Änderung durchführt. Wenn die ADIF-Konsolenanwendung gestartet wird, wird der mithilfe der Befehlszeilenoption `-alfaLoginName` angegebene Anwender als derjenige Anwender hinzugefügt, der die Änderungen durchführt, die durch den Importvorgang an den Überwachungstabellen vorgenommen werden. Wird die Befehlszeilenoption `-alfaLoginName` nicht verwendet, wird `ALFABET_INTERN` in die Überwachungstabellen geschrieben, um den Anwender anzugeben, der für die Änderungen verantwortlich ist.



Der aktuelle Anwendername wird außerdem verwendet, um Eigenschaften von Objekten festzulegen, die während des Imports erzeugt oder geändert werden. Alle Eigenschaften der Gruppe `Artifact` verfügen über Eigenschaften, mit denen Informationen über Änderungen am Objekt gespeichert werden.

Folgende Eigenschaften werden festgelegt, wenn ein Objekt erzeugt oder aktualisiert wird:

- Die Eigenschaft `LAST_UPDATE` wird auf das aktuelle Datum eingestellt.
- Die Eigenschaft `LAST_UPDATE_USER` wird auf den aktuellen Anwender eingestellt.

Folgende Eigenschaften werden zusätzlich festgelegt, wenn ein Objekt erzeugt oder aktualisiert wird:

- Die Eigenschaft `CREATION_DATE` wird auf das aktuelle Datum eingestellt.
- Die Eigenschaft `CREATION_USER` wird auf den aktuellen Anwender eingestellt.

Während des Imports werden `LAST_UPDATE_USER` und `CREATION_USER` als `ALFABET_INTERN` angegeben, wenn die Befehlszeilenoption `-alfaLoginName` nicht festgelegt ist.

Konfigurieren des automatischen Starts von Workflows während des Imports

Alfabet unterstützt Ihr Unternehmen bei der Definition und Pflege von Workflows, in denen Sie die Aktivitäten, die von verschiedenen Anwendern in einer bestimmten Reihenfolge ausgeführt werden sollen, verfolgen können.



Weitere Informationen zu dieser Funktionalität finden Sie unter *Ausführen von Workflows und Teilnehmen an Workflowschritten* im Referenzhandbuch *Erste Schritte mit Alfabet*.

Workflows können automatisch als Teil des durch ADIF ausgelösten Imports gestartet werden. Dies kann hilfreich sein, um sicherzustellen, dass neue Daten in der Alfabet-Datenbank von für die Datenpflege verantwortlichen Anwendern nachbearbeitet werden.



Der automatische Start von Workflows erfordert folgende Konfiguration:

- Es muss eine Workflow-Vorlage in Alfabet Expand definiert werden.
- Dem ADIF-Importschema muss ein Element **Workflow-Eintrag** hinzugefügt werden, in dem der Name der Workflow-Vorlage angegeben wird.

Informationen über das Konfigurieren von Projektvorlagen finden Sie unter *Konfigurieren von Workflows* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Berücksichtigen Sie Folgendes, wenn Sie die Workflow-Vorlage und das ADIF-Importschema konfigurieren:

- Der Workflow muss so konfiguriert sein, dass er automatisch startet.
- Stellen Sie sicher, dass die importierten Objekte über eine Abfrage in der Workflow-Vorlage identifiziert werden können, um zu ermöglichen, Workflows nur für importierte Objekte zu starten. Eine Methode besteht darin, eine benutzerdefinierte Eigenschaft für die Zielklasse für den Import im Alfabet -Metamodell mit Alfabet Expand zu definieren. Die benutzerdefinierte Eigenschaft wird dann während des Imports befüllt, beispielsweise mit dem Importdatum.

Details finden Sie unter *Anwendungsfall: Definieren importbezogener benutzerdefinierter Eigenschaften* im Abschnitt *Konfigurieren von SQL-Befehlen für optionale erweiterte Importfunktionen*.



Zum Beispiel wird eine neue benutzerdefinierte Eigenschaft `ImportDate` (TechName: `IMPORTDATE`) für eine Zielobjektklasse definiert, etwa die Objektklasse `Device`. Die Eigenschaft `ImportDate` verfügt über den Datentyp `Date` und ist mit dem Datum ausgefüllt, das während der Aktualisierung aktuell war. Die Workflow-Vorlage kann jetzt mit der folgenden Abfrage konfiguriert werden, um nach jenen Objekten zu suchen, für die Workflows gestartet werden:

```
SELECT REFSTR
FROM DEVICE dev
WHERE dev.IMPORTDATE = @TODAY
```

Mit der Abfrage wird nach allen Objekten der Objektklasse `Device` gesucht, für die am entsprechenden Tag Daten aktualisiert oder erstellt wurden. Da die Abfrage während des Imports ausgeführt wird, stimmt das Importdatum mit dem Startdatum des Workflows überein.

- Workflows werden beim Import gestartet, nachdem alle Importeinträge ausgeführt wurden, aber vor dem Verwerfen der temporären Tabellen. Allerdings werden temporäre Tabellen von Alfabet -Funktionalitäten ignoriert, und die für Workflows definierten Abfragen interpretieren keine Daten aus temporären Tabellen.
- Workflow-Einträge werden ignoriert, wenn das Attribut **Änderungen beibehalten** des Importschemas auf `False` gesetzt ist.

So fügen Sie dem ADIF-Importschema einen Workflow hinzu:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Importschema, und wählen Sie **Workflow-Eintrag erzeugen** aus. Dem Explorer wird ein neuer Knoten mit einem Standardnamen für den Workflow-Eintrag hinzugefügt. Unterhalb des Knotens für den Workflow-Eintrag wird automatisch ein Ordner für die Erstellung von SQL-Befehlen hinzugefügt, die bei Aktivierung des Workflow-Eintrags ausgeführt werden sollen.
- 2) Legen Sie im Attributfenster des Workflow-Eintrags die folgenden Attribute fest:
 - **Name:** Geben Sie einen eindeutigen Namen für den Workflow-Eintrag ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem Workflow-Eintrag implementiert wird.
 - **Workflow-Vorlage:** Wählen Sie im Dropdown-Listefeld den Namen des Workflow-Eintrags aus.
 - **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung aller Importeinträge innerhalb der Importgruppe auszuwählen. Wählen Sie `False` aus, um die Ausführung zu deaktivieren.

Kapitel 5: Konfigurieren von Datenexporten mit ADIF

Dieses Kapitel behandelt den Prozess für den Datenexport aus der Alfabet-Datenbank mit dem ADIF-Exportschema.

Der Exportprozess

Daten können aus Alfabet-Datenbank -Tabellen in Datenbanktabellen externer Datenbanken, Microsoft® Excel®-Dateien, kommagetrennte Datenformatdateien (.csv oder.txt) und XML-Dateien exportiert werden.

Der Export wird von ADIF-Exportschemata ausgelöst, die vom Kunden mithilfe von SQL-Befehlen konfiguriert werden. Ein Exportschema kann den Export in mehrere Datenbanktabellen oder -dateien auslösen, die in einem oder unterschiedlichen Dateiformaten vorliegen.



Wenn Sie eine große Datenmenge in eine einzelne Datei exportieren, kann der Zeitaufwand für die Verarbeitung der Daten und die Speicherauslastung während ihrer Verarbeitung sehr hoch sein. Um Leistungsprobleme zu vermeiden, empfiehlt es sich, das ADIF-Exportschema so zu konfigurieren, dass die Daten in mehrere Dateien exportiert werden. Dies kann z. B. durchgeführt werden, um den Objektexport nach einem Wert einer der Objektklasseneigenschaften zu filtern. Zum Beispiel, indem alle Applikationen mit einem Namen, der mit "A" beginnt, in eine Datei exportiert werden, alle, die mit "B" beginnen, in eine andere Datei und so weiter.

Der Exportprozess wird über den ADIF-Debugger oder die ADIF-Konsolenanwendung gestartet.

Beim Starten des ADIF-Exports wird der folgende Prozess ausgeführt:

- 1) Exportschema wird überprüft. Der Export wird angehalten, wenn die Überprüfung fehlschlägt.
- 2) Ein temporäres Verzeichnis wird auf Serverseite erstellt, um es für Protokoll- und Exportdateien zu verwenden.
- 3) Schreiben von Befehlszeilenargumenten in die Alfabet-Datenbank. Wenn der Export durch ein Exportschema in einer XML-Datei ausgelöst wird, wird das Exportschema auch in die Datenbank geschrieben.
- 4) Die im ADIF-Exportschema konfigurierten Exportschritte werden ausgeführt. Die folgenden Beschreibungen der Exportkonfiguration liefern Informationen dazu, welche konkrete Aktion jeweils durch die einzelnen Elemente des Schemas ausgelöst wird. Während der Durchführung des Exports werden Exportdateien für den Export in XML-, CSV- oder Microsoft® Excel®-Formate im temporären Verzeichnis erstellt. Für den Export in externe Datenbanktabellen werden Daten in die externe Datenbank geschrieben.
- 5) ZIP-Datei aus allen im temporären Verzeichnis erzeugten Dateien wird erstellt.
- 6) Laden Sie die ZIP-Datei in den Ordner **ADIF_SYS** des **Dokument-Explorers** der Alfabet-Datenbank hoch.
- 7) Laden Sie die Protokollinformationen zum Exportprozess in den Ordner **ADIF_SYS** des **Dokument-Explorers** der Alfabet-Datenbank hoch.
- 8) Das temporäre Verzeichnis auf der Serverseite, das vor Ausführung des Exports erzeugt wurde, und alle Inhalte des Verzeichnisses werden gelöscht.
- 9) Zip-Datei wird an den Speicherort heruntergeladen, der in den Befehlszeilen-Argumenten angegeben ist.
- 10) Inhalt der Protokolldatei wird in die Konsolen-Protokolldatei heruntergeladen.

11) Zip-Datei wird entpackt, falls das Entpacken bei den Befehlszeilen-Parametern erforderlich ist.

In diesem Kapitel wird beschrieben, wie unterschiedliche Exportfunktionen innerhalb des ADIF-Exportschemas konfiguriert werden. Der Exportprozess ist in hohem Maß konfigurierbar, um die individuellen Kundenanforderungen an die daraus resultierende Datenausgabe zu erfüllen.

Standardkonfiguration des Datenexports

Bei einem ADIF-Exportschema handelt es sich um ein XML-Objekt, das entweder in einem XML-Editor oder in der ADIF-Konfigurationsbenutzeroberfläche, die als Teil des Konfigurationstools Alfabet Expand bereitgestellt wird, bearbeitet werden kann. In der ADIF-Konfigurationsbenutzeroberfläche werden die XML-Elemente als Knoten im Explorer angezeigt, während Attribute eines XML-Elements im Attributfenster auf der rechten Seite angezeigt werden.

Alle Informationen und Befehle, die zum Exportieren von Daten aus der Alfabet-Datenbank erforderlich sind, werden mithilfe der XML-Elemente des ADIF-Importschemas definiert. Die Elemente müssen einer definierten Sequenz folgen, die von einem XSD-Schema bestimmt wird. Im ADIF-Explorer werden Sie bei der Erzeugung von Elementen über das Kontextmenü der Explorer-Knoten durch die Erzeugung aller erforderlichen Elemente geführt. Es können nur Elemente, die als Unterelement zulässig sind, als untergeordnete Knoten eines Explorer-Knotens erzeugt werden. Wenn Sie das ADIF-Exportschema in einem XML-Editor definieren, müssen Sie im XSD-Schema nach Informationen zur zulässigen Reihenfolge von Objekten suchen.

Die folgenden Elemente sind Teil des ADIF-Exportschemas:

Name des XML-Elements	Titel in der ADIF-Benutzeroberfläche	Zweck
ADIF_ExportScheme	Exportschema	Element für die Konfiguration grundlegender und umfassender Exportausführungsparameter.
XmlExportSet	Eintrag für Klassen (XML)	Strukturierendes Element, das als Container für Elemente dient, mit denen der Export in XML-Dateien definiert wird.
FileExportSet	Eintrag für Klassen (Excel)	Strukturierendes Element, das als Container für Elemente dient, mit denen der Export in Microsoft® Excel®- oder CSV-Dateien definiert wird. HINWEIS: Über die ADIF-Benutzeroberfläche ist dieses Element nur für die Strukturierung von Excel-Expoteinträgen verfügbar.
DBExportSet	DB-Exportgruppe	Element, mit dem die Verbindung zu einer externen Datenbank definiert wird.
ExportEntry	Eingabe	Element, das den Inhalt des Exports in eine externe Datenbanktabelle oder eine externe Datei definiert.

Name des XML-Elements	Titel in der ADIF-Benutzeroberfläche	Zweck
SQLCommand	SQL-Befehl	Den <code>ExportEntry</code> -Elementen untergeordnetes Element für die Definition von SQL-Befehlen, die während des Exports ausgeführt werden sollen.
AlfaMethod		Sub-element of <code>ExportEntry</code> elements to assign custom code to the entry. AlfaMethod elements are only relevant if custom code has been developed by Software AG for special customer requirements. Information about the configuration is provided individually with the custom code.

In diesem Abschnitt wird die Basiskonfiguration beschrieben, die für das Auslösen eines einfachen Datenexports in unterschiedliche Dateiformate oder externe Datenbanktabellen erforderlich ist. Die Basiskonfiguration kann erweitert werden, um komplexe Exportszenarien zu ermöglichen, indem die in den folgenden Abschnitten beschriebenen Sonderfunktionen hinzugefügt werden:

- [Konfigurieren von Protokollierungsparametern](#)
- [Konfigurieren der Ausführung des Importschemas in Abhängigkeit von aktuellen Parametern](#)
 - [Konfigurieren der bedingten Ausführung von Teilen des Importschemas](#)
 - [Konfigurieren des Imports in Abhängigkeit von Parametern](#)
- [Konfigurieren von Protokollierungsparametern](#)
 - [Konfigurieren der Speicherung und Handhabung von Protokolldateien](#)
 - [Konfigurieren des Inhalts von Protokollmeldungen](#)

Erzeugen eines ADIF-Exportschemas

Führen Sie folgende Schritte aus, um ein neues ADIF-Exportschema zu erstellen:

- 1) Klicken Sie im Explorer mit der rechten Maustaste auf den Root-Knoten **ADIF Schemes**, und wählen Sie **Exportschema erstellen** aus. Das neue Exportschema wird dem Explorer hinzugefügt, und das Attributfenster für das neue Exportschema wird auf der rechten Seite angezeigt.
- 2) Legen Sie im Attributfenster die folgenden Attribute für das ADIF-Exportschema fest:
 - **Name:** Geben Sie einen eindeutigen Namen ein. Der **Name** dient der Identifikation des ADIF-Schemas in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.
 - **Titel:** Geben Sie einen aussagekräftigen und eindeutigen Titel ein. Der Titel wird verwendet, um das ADIF-Importschema in der Alfabet -Benutzeroberfläche der Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** zu identifizieren.

- **Beschreibung:** Geben Sie eine aussagekräftige und kurze Beschreibung des Ergebnisses des ADIF-Imports ein. Die Beschreibung wird in der Alfabet Benutzeroberfläche der Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** angezeigt.
- **Änderungen beibehalten:** Ist diese Option auf `True` gesetzt, wird das Ergebnis des Datenexports dauerhaft in die externen Dateien oder in die externe Zieldatenbank geschrieben. Ist diese Option auf `False` gesetzt, wird der Exportprozess nach der Ausführung zurückgenommen, und es werden keine Änderungen in die externe Datenbank oder Dateien geschrieben. Es wird empfohlen, **Änderungen beibehalten** für ein neues Exportschema für den Export in eine externe Datenbank auf `False` zu setzen, um das Debuggen zuzulassen, ohne eine Beschädigung der externen Datenbank zu riskieren. Nach der erfolgreichen Prüfung des Datenexports und der Überprüfung, dass die resultierenden Änderungen an der externen Datenbank wie erwartet ausgefallen sind, können Sie das Attribut **Änderungen beibehalten** auf `True` setzen, um reguläre Datenexporte durchzuführen.



Die folgenden Einschränkungen gelten beim Setzen des Attributs **Änderungen beibehalten** auf `False`:

- **Änderungen beibehalten** betrifft nur Datenbanktransaktionen. Wenn Sie Daten in eine Datei exportieren, wird die Exportdatei erstellt und die Daten werden auch dann der Datei hinzugefügt, wenn **Änderungen beibehalten** auf `False` gesetzt ist.
- Wenn das Attribut **Änderungen beibehalten** auf `False` gesetzt wird, werden dadurch alle Änderungen, die durch DML-Anweisungen verursacht wurden (Änderungen an Datensätzen in vorhandenen Tabellen), zurückgenommen. Das Erstellen oder Löschen von Tabellen ist nicht im Rollback enthalten. Das heißt, wenn Sie zum Beispiel ein ADIF-Schema testen, das dafür konfiguriert ist, temporäre Tabellen dauerhaft in die Datenbank zu schreiben, werden diese temporären Tabellen dauerhaft erzeugt, selbst wenn **Änderungen beibehalten** auf `False` gesetzt ist. SQL-Befehle vom Typ `OnActivate` sind ebenfalls vom Rollback ausgeschlossen.
- **Verhalten der Alfabet-Benutzeroberfläche:** Wählen Sie eine der folgenden Optionen aus, um die Verfügbarkeit des ADIF-Schemas in der Alfabet -Benutzeroberfläche zu definieren:
 - `VisibleExecutable`: Der ADIF-Export kann in der **ADIF-Prozessverwaltung** ausgelöst werden, und der Erfolg der ADIF-Prozesse, die mit diesem ADIF-Schema ausgeführt wurden, kann über die Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** überprüft werden.
 - `VisibleNotExecutable`: Der Erfolg der ADIF-Prozesse, die mit diesem ADIF-Schema ausgeführt wurden, kann über die Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** überprüft werden. Der ADIF-Export kann jedoch nicht über die Alfabet -Benutzeroberfläche ausgelöst werden.
 - `NotVisible`: Das ADIF-Schema und die Informationen zu ADIF-Prozessen, die mit diesem ADIF-Schema ausgeführt wurden, sind in den Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** nicht sichtbar. Für neue ADIF-Exportschemata ist dies der Standardwert.



Informationen zur Verwaltung und Ausführung des ADIF-Schemas über die Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** finden Sie unter

[Ausführen und Überprüfen von ADIF über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" in der Alfabet -Benutzeroberfläche.](#)

- **Für REST-API verwendbar:** Setzen Sie das Attribut auf **True**, wenn das ADIF-Schema über einen RESTful-Serviceaufruf an den Endpunkt `adifexport` der RESTful API von Alfabet ausgeführt werden soll, entweder über einen RESTful-Serviceaufruf von einem externen RESTful-Client oder über ein Alfabet -Ereignis, das die Ausführung des ADIF-Schemas auslöst, wenn ein Anwender einen Assistenten oder Workflow-Schritt aufruft oder verlässt oder wenn ein Ereignis zur Ausführung eines RESTful-Serviceaufrufs an eine Drittanbieterapplikation abgeschlossen wird.



Die Ausführung des ADIF-Schemas über die RESTful API von Alfabet erfordert die Einrichtung der RESTful-Services. Informationen über die Anforderungen und die Ausführung des Serviceaufrufs finden Sie im Referenzhandbuch *Alfabet RESTful API*.

Informationen zur zusätzlichen Konfiguration zur Implementierung der Ausführung eines Ereignisses finden Sie unter *Konfigurieren von Ereignissen*.



Nach dem Erstellen des Exportschemas können Sie nun Exportdefinitionen hinzufügen, um den Export in eines oder mehrere Datenformate auszulösen. Fahren Sie mit folgendem Schritt fort:

- [Konfigurieren des Exports in externe Datenbanktabellen](#)
- [Konfigurieren des Exports in XML-Dateien](#)
- [Konfigurieren des Exports in kommagetrennte Datendateien](#)
- [Konfigurieren des Exports in Microsoft® Excel®-Dateien](#)

Optional können Sie folgende Optionen konfigurieren:

- [Konfigurieren der Ausführung des Importschemas in Abhängigkeit von aktuellen Parametern](#)
- [Konfigurieren von Protokollierungsparametern](#)

Konfigurieren des Exports in externe Datenbanktabellen

Daten können in Datenbanktabellen von externen Datenbanken exportiert werden.

Der Datenexport wird durch eine SQL-Anweisung `SELECT` ausgelöst. Das Ergebnis der SQL-Abfrage ist eine Datentabelle, die folgendermaßen in der externen Datenbank enthalten ist:

- Daten werden in die Datenbanktabelle geschrieben, die mit dem Attribut **Exporttabelle** des Exporteintrags angegeben wurde, für das die SQL-Abfrage definiert ist.
- Die Daten werden den Spalten in der angegebenen Tabelle zugeordnet, indem die Alias-Namen der Anweisung `SELECT` den Zeilen der externen Datenbanktabelle zugeordnet werden.
- Alle Datensätze aus dem Dataset werden den bereits vorhandenen Datensätzen der externen Datenbanktabelle hinzugefügt.



Wenn Daten einer Datenbanktabelle hinzugefügt werden, die bereits Daten enthält, kann ein einfacher Datenexport diese vorhandenen Daten nicht aktualisieren, sondern fügt der Datenbanktabelle nur neue Zeilen hinzu. Wenn dies zu Verstößen gegen eindeutige Schlüssel führt (weil in der Datenbanktabelle bereits ein anderes Objekt mit demselben Schlüssel vorhanden ist), wird der Datenexport abgebrochen.

Das ADIF-Exportschema bietet verschiedene Mechanismen zur Behebung dieses Problems:

- Es können zusätzliche SQL-Befehle konfiguriert werden, um die Zieldatenbank vor oder nach dem Datenexport zu manipulieren. Sie können den Exportprozess so konfigurieren, dass die exportierten Daten in eine temporäre Tabelle geschrieben werden, die tatsächliche Zieldatenbanktabelle über entsprechende SQL-Befehle aktualisiert und dann die temporäre Tabelle entfernt wird.
- Es kann ein Mechanismus aktiviert werden, der vor der Durchführung des Datenexports alle Datensätze aus der Zieldatenbank entfernt.

Informationen über die Aktualisierung vorhandener Daten in einer Zieldatenbank finden Sie unter [Anwendungsfall: Ausfüllen leerer Eigenschaftswerte mit Standardwerten beim Importieren in Pflicht-Objektklasseneigenschaften](#) im Abschnitt [Konfigurieren von Protokollierungsparametern](#).



Wenn eine Zeichenfolge, die in eine Zieldatenbank importiert wird, die maximal zulässige Größe für Zeichenfolgen in der Zieldatenbank überschreitet, wird der Datenexport abgebrochen. Wenn die Zeichenfolgenreife für die Werte in der Zieldatenbank beschränkt ist, muss deshalb der für den Datenexport definierte SQL-Befehl so definiert sein, dass Daten auf die zulässige Dateigröße abgeschnitten werden. Zum Abschneiden der Daten-Zeichenfolgen können die Native-SQL-Funktionen `LEFT (Column, MaximumLength)` oder `SUBSTRING (Column, 1, MaximumLength)` verwendet werden.

Konfigurieren des Exports in eine Zieldatenbank



Die folgende Konfiguration innerhalb eines ADIF-Exportschemas ist für den Datenexport von der Alfabet-Datenbank in eine externe Datenbank erforderlich:

- Erstellen Sie eine Exportgruppe für die Spezifikation der Verbindung der externen Datenbank.
- Erstellen Sie einen Eintrag für die Spezifikation der Zieldatenbanktabelle.
- Definieren Sie die SQL-Abfrage für die Definition des Inhalts, der der externen Datenbanktabelle im Unterordner **SQL-Befehle - DataExport** des Exporteintrags hinzugefügt werden soll.



Sie dürfen nicht mehr als einen SQL-Befehl vom Typ `DataExport` für den Export in eine Zieldatenbank definieren.

Der Workflow für die Definition des Exporteintrags wird nachfolgend beschrieben.

Sie können optional folgende Aktionen ausführen:

- Definieren Sie zusätzliche SQL-Befehle, um komplexe Exportszenarien zu definieren, wie beispielsweise die Aktualisierung von Daten in einer Datenbanktabelle, anstatt der

bestehenden Datenbanktabelle einfach neue Zeilen hinzuzufügen. Informationen hierzu finden Sie unter [Konfigurieren von Protokollierungsparametern](#).

So erstellen Sie einen Exporteintrag für den Export in eine externe Datenbank:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Exportschema, dem Sie eine Exportgruppe hinzufügen möchten, und wählen Sie **DB-Exportgruppe erzeugen** aus. Dem Explorer wird eine neue Datenbankexportgruppe hinzugefügt.
- 2) Wählen Sie im Explorer die neue Datenbankexportgruppe aus, und bearbeiten Sie im Attributfenster folgende Attribute:
 - **Name:** Geben Sie einen aussagekräftigen Namen für die Datenbankexportgruppe ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit der Datenbankexportgruppe implementiert wird.
 - **Treibertyp:** Wählen Sie die Art von Datenbankserver aus, auf dem sich die externe Datenbank befindet. Der Import kann von Datenbanken auf Microsoft® SQL Server®, Oracle®-Datenbankservern oder Microsoft® Access-Datenbanken durchgeführt werden. Wählen Sie eine der folgenden Optionen aus:
 - `SqlServer` für den Zugriff auf Microsoft SQL Server
 - `Oracle` für den Zugriff auf Oracle
 - `Access` für den Zugriff auf Microsoft Access über einen OLE-Treiber
 - `ODBC` für den Zugriff auf Microsoft Access, Oracle® MySQL oder PostgreSQL 9.1 über einen ODBC-Treiber
 - `Hadoop` für den Zugriff auf Hadoop-Systeme
 - **Untergeordneter Treibertyp:** Wählen Sie für Verbindungen zu Oracle®-Datenbankservern `AlfaOraDevArt` aus.
 - **Verbindungsstring:** Geben Sie den Verbindungsstring ein, der für die Verbindung mit der externen Quelle erforderlich ist. Der Verbindungsstring hängt vom verwendeten Datenbankserver ab. Sie enthält in der Regel den Quellspeicherort, den Anwendernamen sowie weitere Parameter, die den Anforderungen des Datenbankservers entsprechen.



Es wird empfohlen, das Datenbank-Login im Verbindungsstring über einen Benutzernamen und ein Kennwort zu definieren. Wenn die Windows-Authentifizierung zur Anmeldung verwendet wird und die ADIF-Konsolenapplikation mit einem Remote-Alias gestartet wird, wird nur dann erfolgreich eine Verbindung zur Datenbank hergestellt, wenn der Alfabet-Server mit demselben Domänen-Benutzernamen und denselben Zugriffsrechten gestartet wird wie im Verbindungsstring angegeben.



Hinweis: In den Hilfe-Links können Server-Variablen genutzt werden. Mithilfe von Servervariablen können Sie die gesamte oder einen Teil der Definition der Server-Alias-Konfiguration des Alfabet-Servers definieren, anstatt diese direkt im ADIF-Schema zu definieren. Der Einsatz von Servervariablen ist vor allem dann sinnvoll, wenn Sie zum Beispiel die Konfiguration in einer Test- bzw. Produktionsumgebung mit unterschiedlichen externen Quellen verwenden. In beiden Umgebungen kann dieselbe Definition der externen Quelle verwendet werden. Mit der in der Test- und Produktionsumgebung verwendeten Definition des Serveralias werden die korrekten Verbindungsdaten für die externe Quelle definiert, die in der jeweiligen Umgebung verwendet wird. Die Verwendung von Servervariablen wird im Abschnitt *Verwenden*

von Servervariablen in der Spezifikation von Weblinks und Datenbankservern im Kapitel *Berichte konfigurieren* erläutert.



ODBC kann für Verbindungen zu Microsoft Access - oder PostgreSQL 9.1-Datenbankservern genutzt werden, allerdings wird die Performance im Vergleich zu nativen Treibern schlechter sein, und gespeicherte Prozeduren werden nicht unterstützt.



Im Folgenden finden Sie Verbindungsstrings für Verbindungen zu verschiedenen Datenbankservern.

Verbindungsstring für Verbindung mit einem Microsoft SQL Server® mit Standardanmeldung:

```
Server=myServerAddress;Database=myDataBase;User
Id=myUsername;Password=myPassword;
```

Verbindungsstring für direkte Verbindung mit einem Oracle®-Datenbankserver:

```
User Id=APP_ID;Password=*****;Direct=true;Data
Source=servername;Port=Port;SID=orcl
```

Verbindungsstring für direkte Verbindung mit einem Oracle®-Datenbankserver einschließlich Servicename:

```
User
Id=*****;Password=*****;Server=ServerFQDN/IP;Unicode
=True;Connection Timeout=60;Max Pool
Size=250;Direct=True;Service Name=ServiceName;Port=1521
```

Verbindungsstring für indirekte Verbindung mit einem Oracle®-Datenbankserver:

```
User
Id=***;Password=*****;Server=orcl;Unicode=True;Connection
Timeout=60;Max Pool Size=250
```

- **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung aller Exporteinträge innerhalb der Exportgruppe auszuwählen. Wählen Sie `False` aus, um die Ausführung zu deaktivieren.
- 3) Klicken Sie mit der rechten Maustaste auf die Datenbankexportgruppe, und wählen Sie **Eintrag erzeugen** aus. Dem Explorer wird ein neuer Exporteintragsknoten als untergeordneter Knoten der Datenbankexportgruppe hinzugefügt.
 - 4) Klicken Sie im Explorer auf den neuen Exporteintrag-Knoten.
 - 5) Legen Sie im Attributfenster rechts die folgenden Attribute für den Exporteintrag fest:
 - **Name:** Ändern Sie den Standardnamen des Eintrags in einen aussagekräftigen Namen. Der **Name** dient der Identifikation des Exporteintrags in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.
 - **Kommentare:** Optional können Sie einen Kommentar mit Informationen zum Export zur internen Verwendung eingeben.
 - **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung des Eintrags während des Exports zu aktivieren.
 - **Exporttabelle:** Geben Sie einen Namen für die Zieldatenbanktabelle in der externen Datenbank ein. Die Daten werden in die angegebene Tabelle geschrieben.

- **Tabelle nach Export löschen:** Wählen Sie `True` aus, um die Datenbanktabelle, die mit dem Attribut **Export-Tabelle** als Ziel für den Export angegeben wurde, nach dem Ausführen des Exports zu löschen. Die Tabelle wird gelöscht, nachdem alle SQL-Befehle ausgeführt wurden, die in der Datenbankexportgruppe für die Exporteinträge angegeben waren. Wählen Sie `False`, um Daten dauerhaft in die Zieldatenbank zu schreiben.



Wenn Sie vorhandene Daten in einer Zieldatenbank aktualisieren möchten, können Sie sie in eine temporäre Tabelle exportieren, zusätzliche SQL-Befehle angeben, die für die Zieldatenbank ausgeführt werden sollen, um die Daten in der tatsächlichen Zieldatenbanktabelle zusammenzuführen, und die temporäre Tabelle nach dem Export löschen. Die Erstellung der temporären Tabelle muss mit einem SQL-Befehl ausgelöst werden, das Löschen der Tabelle erfolgt jedoch automatisch, wenn das Attribut **Tabelle nach Export löschen** auf `True` gesetzt ist.

- **Tabelle vor dem Exportieren leeren:** Wählen Sie `True` aus, um alle Datensätze aus der Datenbanktabelle, die mit dem Attribut **Export-Tabelle** als Ziel für den Export angegeben wurde, vor dem Ausführen des Exports zu löschen. Die Tabelle wird gelöscht, nachdem alle SQL-Befehle ausgeführt wurden, die in der Datenbankexportgruppe für die Exporteinträge angegeben waren. Wählen Sie `False` aus, um Daten den vorhandenen Datensätzen hinzuzufügen.



Die Einstellungen für den eindeutigen Schlüssel und den Primärschlüssel hängen von der Konfiguration der Zieldatenbank ab und können daher nicht vom Standardexportmechanismus berücksichtigt werden, der von ADIF bereitgestellt wird. Aus diesem Grund kann das Hinzufügen von Daten in bestehende Datensätze zu Problemen beim Datenimport führen. Das Löschen von Datensätzen aus einer Tabelle vor der Ausführung des Datenexports ist Bestandteil der Mechanismen, die die Implementierung einer Exportkonfiguration ermöglichen, die das Aktualisieren bestehender Daten in einer Zieldatenbank erlaubt. Weitere Informationen zu einer derartigen Konfiguration finden Sie unter [Anwendungsfall: Ausfüllen leerer Eigenschaftswerte mit Standardwerten beim Importieren in Pflicht-Objektklasseneigenschaften](#) im Abschnitt [Konfigurieren von Protokollierungsparametern](#).

- **Boolean 'False':** Wenn eine boolesche Eigenschaft, die in eine externe Datenbank exportiert wird, in der Zieldatenbank durch eine Zeichenfolge dargestellt werden soll, wählen Sie den Zeichenfolgenwert, der verwendet werden soll, wenn ein Attribut des Datentyps "Boolean" auf `False` gesetzt ist. Die Standardeinstellung ist `True`.



Wenn eine boolesche Eigenschaft im Alfabet -Metamodell nicht obligatorisch ist und daher für einige Objekte auf `NULL` gesetzt ist, wird `NULL` nicht als `False` interpretiert, sondern als leeres Datenfeld. Wenn `NULL` als `False` behandelt werden soll, müssen Sie Ihre SQL-Abfrage entsprechend konfigurieren.

- **Boolean 'True':** Wenn eine boolesche Eigenschaft, die in eine externe Datenbank exportiert wird, in der Zieldatenbank durch eine Zeichenfolge dargestellt werden soll, wählen Sie den Zeichenfolgenwert, der verwendet werden soll, wenn ein Attribut des Datentyps "Boolean" auf `True` gesetzt ist. `True` ist Standard.
- **Anzahl der Dezimalziffern:** Wenn Zahlen beim Export in Zeichenfolgen konvertiert werden sollen, definieren Sie die Anzahl der Stellen, die in der Ausgabe für den Dezimalteil der Zahlen definiert werden sollen. Die Standardeinstellung lautet "2". Zweieinhalb wird beispielsweise als "2.50" geschrieben.

- **Anzahl der Dezimaltrennzeichen:** Wenn Zahlen beim Export in Zeichenfolgen konvertiert werden sollen, definieren Sie das Dezimaltrennzeichen für Zahlen. Die Standardeinstellung ist ein Punkt.
 - **Anzahl der Gruppentrennzeichen:** Wenn Zahlen beim Export in Zeichenfolgen konvertiert werden sollen, definieren Sie das Trennzeichen für Zahlengruppen. Die Standardeinstellung ist ein Komma. Dreitausend wird beispielsweise als "3,000" geschrieben.
 - **Datumsformat:** Wenn Daten beim Export in Zeichenfolgen konvertiert werden sollen, definieren Sie das Format für Daten, die in die Zieldatenbanktabelle geschrieben werden. Der Standard ist TT/MM/JJJJ.
- 6) Erweitern Sie im Explorer den Exporteintrag-Knoten des neuen Datenbankexporteintrags.
 - 7) Klicken Sie mit der rechten Maustaste auf den Ordner **SQL-Befehle – DataExport**, und wählen Sie **SQL-Befehl erzeugen** aus. Dem Explorer wird ein neuer SQL-Befehlsknoten hinzugefügt.
 - 8) Klicken Sie auf den neuen SQL-Befehlsknoten, und geben Sie im Attributfenster Folgendes an:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
 - **Befehlstyp:** Wählen Sie `DMLstatement` im Dropdown-Listefeld aus.
 - **Anwenden auf:** Wählen Sie `local` aus.
 - **Ergebnistyp:** Wählen Sie `Undefined` aus.
 - **Text:** Definieren Sie eine SQL-Abfrage mit einer `SELECT`-Anweisung, die einen Datensatz zurückgibt. Der resultierende Datensatz wird in die im Exporteintrag angegebene Zieldatenbanktabelle exportiert. Die aus der `SELECT`-Anweisung resultierenden Spaltentitel müssen den Spaltentiteln in der Zieldatenbank entsprechen. Schreiben Sie die SQL-Abfrage entweder direkt in das Feld oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.

 Wenn Sie komplexe Exportszenarien konfigurieren möchten, wie beispielsweise die Aktualisierung bestehender Daten in der Zieldatenbank, können Sie dem Exporteintrag zusätzliche SQL-Befehle hinzufügen. Weitere Informationen zur Definition von SQL-Befehlen in einem Exporteintrag finden Sie unter [Konfigurieren von Protokollierungsparametern](#).

 - **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Export auch dann ausgeführt werden soll, wenn die im Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Export im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
 - **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.
 - 9) Wiederholen Sie Schritt 3–8 für jede Datenbanktabelle der Zieldatenbank, in die Sie Daten exportieren möchten.

Konfigurieren des Exports in XML-Dateien

Daten können in XML-Dateiformate mit der Erweiterung.xml exportiert werden.

Der Datenexport wird durch eine mehrfache SQL-Anweisung `SELECT` ausgelöst. Die SQL-Anweisungen können hierarchisch strukturiert werden, um die Hierarchie der XML-Elemente innerhalb der Zielformat zu bilden. In diesem Abschnitt wird zunächst die Struktur beschrieben, die als Rahmen für die Definition einer XML-Exportdatei im ADIF-Exportschema im Abschnitt bereitgestellt werden muss. Dann wird anhand eines einfachen Beispiels im Abschnitt die Definition von XML in der Zielformat über SQL-Abfragen und Attribute des ADIF-Exportschemas erklärt.

Konfigurieren des Exports in eine XML-Zielformat



Die folgende Konfiguration innerhalb eines ADIF-Exportschemas ist für den Datenexport aus der Alfabet-Datenbank in eine einzelne XML-Datei erforderlich:

- Erstellen Sie einen Exporteintrag für die Spezifikation der Zielexportdatei.
- Definieren Sie die SQL-Abfrage für die Definition des Inhalts der Datei im Unterordner **SQL-Befehle - DataExport** des Exporteintrags.

Der Workflow für die Definition des Exporteintrags wird nachfolgend beschrieben.

Sie können optional folgende Aktionen ausführen:

- Definieren Sie zusätzliche SQL-Befehle, um komplexe Exportszenarien zu definieren. Informationen hierzu finden Sie unter [Konfigurieren von Protokollierungsparametern](#).
- Strukturieren Sie die Exporteinträge in einer Exportgruppe. Exportgruppen haben keine technisch relevanten Attribute und werden nur genutzt, um die Lesbarkeit der ADIF-Exportgruppe zu verbessern. Die Definition von XML-Exportgruppen ist derzeit nur mittels direkter Definition im Texteditor des `XMLExportSet`-Elements in der XML des ADIF-Exportschemas möglich. Informationen zur Definition von ADIF-Exportschemata in Texteditoren finden Sie unter *Konfigurieren von ADIF über XML-Bearbeitung*.

So erstellen Sie einen Exporteintrag für den Export in XML-Dateien:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das Exportschema, und wählen Sie **Eintrag erzeugen** aus. Dem Explorer wird ein neuer Exporteintrag-Knoten als Unterknoten des ADIF-Exportschemas hinzugefügt.



Alternativ können Sie die Option **XML-Eintrag für Klassen erzeugen** wählen, um einen XML-Exporteintrag zu erstellen, der bereits eine Basiskonfiguration einer XML-Struktur enthält, mit einem XML-Element pro Klasse parallel unterhalb des Root-Knotens.

Wenn Sie die Option **XML-Eintrag für Klassen erzeugen** auswählen, wird ein Klasselektor geöffnet, mit dem Sie die Objektklassen auswählen können, für die der ersten Ebene von XML-Elementen in der XML-Ausgabedaten hinzugefügt werden sollen. Für jede Klasse, die Sie im Klasselektor auswählen, wird dem XML-Eintrag automatisch ein SQL-Befehl hinzugefügt, der XML-Elemente für die Datenausgabe erzeugt, und es werden Standardwerte für alle Attribute gesetzt. Trotzdem sollten Sie nach den nächsten Schritten in diesem Workflow die Ausgabe kontrollieren.

- 2) Klicken Sie im Explorer auf den neuen Exporteintrag-Knoten.

- 3) Legen Sie im Attributfenster rechts die folgenden Attribute für den Exporteintrag fest:
- **Name:** Ändern Sie den Standardnamen des Eintrags in einen aussagekräftigen Namen. Der **Name** dient der Identifikation des Exporteintrags in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.
 - **Kommentare:** Optional können Sie einen Kommentar mit Informationen zum Export zur internen Verwendung eingeben.
 - **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung des Eintrags während des Exports zu aktivieren.
 - **Exporttabelle:** Geben Sie einen Namen für die Zieldatei für den Export ein. Die Exportdatei wird während des Exports erzeugt. Sie muss die Erweiterung.xml haben.
 - **XML-Root-Tag:** Geben Sie den Namen des Root-Elements der XML-Ausgabe des Exports ein.
 - **Verschlüsselung:** Wählen Sie die Verschlüsselung für die Daten in der Exportdatei aus dem Dropdown-Listefeld aus.
 - **Boolean 'False':** Wählen Sie den Zeichenfolgenwert aus, der in die Exportdatei geschrieben werden soll, wenn ein Attribut des Datentyps "Boolean" auf `False` gesetzt ist. Die Standardeinstellung ist `False`.
-  Wenn eine boolesche Eigenschaft im Alfabet -Metamodell nicht obligatorisch ist und daher für einige Objekte auf NULL gesetzt ist, wird NULL nicht als `False` interpretiert, sondern als leeres Datenfeld. Wenn NULL als `False` behandelt werden soll, müssen Sie Ihre SQL-Abfrage entsprechend konfigurieren.
- **Boolean 'True':** Wählen Sie den Zeichenfolgenwert aus, der in die Exportdatei geschrieben werden soll, wenn ein Attribut des Datentyps "Boolean" auf `True` gesetzt ist. `True` ist Standard.
 - **Anzahl der Dezimalziffern:** Definieren Sie die Anzahl der Stellen, die in der Ausgabe für den Dezimalteil der Zahlen definiert werden sollen. Die Standardeinstellung lautet "2". Zweieinhalb wird beispielsweise als "2,5" geschrieben.
 - **Anzahl der Dezimaltrennzeichen:** Definieren Sie das Dezimaltrennzeichen für Zahlen. Die Standardeinstellung ist ein Punkt.
 - **Anzahl der Gruppentrennzeichen:** Definieren Sie das Trennzeichen für Zahlengruppen. Die Standardeinstellung ist ein Komma. Dreitausend wird beispielsweise als "3,000" geschrieben.
 - **Datumsformat:** Definieren Sie das Format für Daten, die in die Exportdatei geschrieben werden. Der Standard ist TT/MM/JJJJ.
- 4) Erweitern Sie im Explorer den Exporteintrag-Knoten des neuen Exporteintrags.
- 5) Klicken Sie mit der rechten Maustaste auf den Ordner **SQL-Befehle - DataExport**, und wählen Sie **SQL-Befehl erzeugen** aus. Dem Explorer wird ein neuer SQL-Befehlsknoten hinzugefügt.
- 6) Klicken Sie auf den neuen SQL-Befehlsknoten, und geben Sie im Attributfenster Folgendes an:
- **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität angibt, die mit dem SQL-Befehl implementiert wird.
 - **Befehlstyp:** Wählen Sie `DMLStatement` im Dropdown-Listefeld aus.

- **Ergebnistyp:** Wählen Sie `Undefined` aus.
- **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Export auch dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Export im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
- **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.
- **Text:** Definieren Sie eine SQL-Abfrage mit einer `SELECT`-Anweisung, die einen Datensatz zurückgibt. Der resultierende Datensatz wird in die im Exporteintrag angegebene Zielfeld exportiert. Schreiben Sie die SQL-Abfrage entweder direkt in das Attributfeld, oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.

Definition des XML-Inhalts der Exportdatei

Der Inhalt der XML-Zielfeld wird über die Attribute des Exporteintrags und SQL-Befehle im Unterordner **SQL-Befehle - DataExport** des Exporteintrags definiert, sowie über die `SELECT`-Anweisungen der in den SQL-Befehlen definierten SQL-Abfragen.

Für die Definition von SQL-Abfragen gelten die folgenden allgemeinen Regeln:

- Die erste Eigenschaft, die in der `SELECT`-Anweisung definiert wird, muss die `REFSTR`-Eigenschaft der ausgewählten Objektklasse sein. Diese erste Eigenschaft wird nicht in die Ausgabedatei geschrieben. Sie wird nur für technische Prozesse verwendet. Wenn die Eigenschaft `REFSTR` des Objekts Teil der XML-Ausgabe sein soll, müssen Sie den `REFSTR` der ausgewählten Anweisung zweimal hinzufügen.
- Wenn ein aus der Spezifikation resultierender Spaltenname in der `SELECT`-Anweisung mit `A_` oder `E_` anfängt, wird das `A_` oder `E_` nicht in die Ausgabedatei geschrieben. Die Spezifikation `A_PROP` wird also nur als `PROP` ausgegeben. Das Präfix wird verwendet, um zu definieren, wie die Eigenschaften in XML dargestellt werden.

In diesem Abschnitt werden alle XML-Elementtypen aufgeführt, die in der Ausgabedatei enthalten sein können, und es wird beschrieben, wie Objektklassen und Eigenschaften in der Ausgabedatei dargestellt werden. Zum besseren Verständnis wurde ein Beispiel hinzugefügt.



Eine XML-Datei soll erstellt werden, die Informationen zur Ist-Architektur des Unternehmens enthält, das von einem definierten Projekt betroffen ist. Sowohl die Applikationen als auch die Business-Prozesse, die in der Ist-Architektur des Projekts definiert wurden, sollen aufgeführt werden.

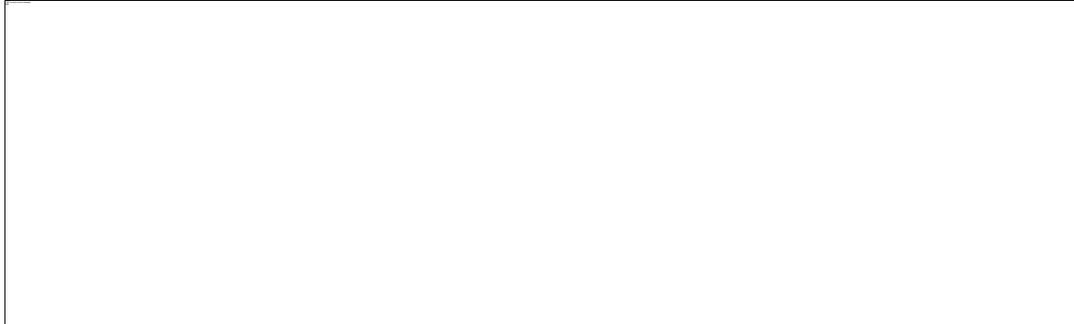
Der ADIF-Export bietet einen Mechanismus, um die Definition des übergeordneten Elements, für das die Information exportiert wird, während der Laufzeit der Exportausführung zu ändern. Informationen hierzu finden Sie unter [Konfigurieren des Imports in Abhängigkeit von Parametern](#). Um das folgende Beispiel einfach zu halten, wird dieser Mechanismus nicht eingesetzt, sondern die Daten werden für ein Beispielprojekt mit Namen „Consolidate Trading Applications“ exportiert.

Stammelement

Das XML-Stammelement von XML wird im Attributabschnitt des Exporteintrags im Attribut **XML-Root-Tag** definiert. Das XML-Stammelement kann keine Attribute haben.



Das Stammelement für die beispielhafte XML-Ausgabe ist als „ProjectArchitecture“ spezifiziert:



Die daraus resultierende XML-Ausgabe ist:

```
<?xml version="1.0" encoding="utf-8"?>
<ProjectArchitecture>
</ProjectArchitecture>
```

Klassenelemente

Klassen-XML-Elemente werden bei Ausführung der SQL-Befehle der XML-Ausgabe im Ordner **SQL-Befehle - DataExport** des XML-Exporteintrags hinzugefügt.

Für jeden Datensatz, der von der für den SQL-Befehl definierten `SELECT`-Anweisung gefunden wird, wird der XML-Ausgabe ein XML-Element hinzugefügt. Der XML-Elementname wird mit dem Attribut **XML-TagName** des SQL-Befehls angegeben. Die in der `SELECT`-Anweisung des SQL-Befehls definierten Eigenschaften werden als XML-Attribute in die XML-Elemente geschrieben. Für jede Eigenschaft wird dem XML-Element ein XML-Attribut mit dem Namen des Alias hinzugefügt, der in der `SELECT`-Anweisung für die Eigenschaft angegeben ist.

Bei der Angabe der Alias-Namen für die `SELECT`-Anweisung müssen Sie die folgenden Regeln berücksichtigen:

- Der Alias-Name darf keine Leerzeichen und keine Sonderzeichen außer "-" oder "_" enthalten.
- Wenn der Alias-Name mit A_ beginnt, ist der daraus resultierende Attributname die Alias-Zeichenfolge ohne das A_.
- Wenn der Alias-Name mit E_ beginnt, wird die Eigenschaft nicht als XML-Attribut in das Klassen-XML-Element aufgenommen, sondern als dem Klassen-XML-Element untergeordnetes Eigenschaft-XML-Element.
- Die erste in der `SELECT`-Anweisung definierte Eigenschaft wird nicht in die Ausgabedatei geschrieben. Aus technischen Gründen muss die erste in der `SELECT`-Anweisung definierte Eigenschaft den `REFSTR` eines Objekts angeben.



Um der XML-Ausgabe Informationen zu den in der Ist-Architektur des Projekts definierten Applikationen hinzuzufügen, wird dem Ordner **SQL-Befehle - DataExport** des XML-Exporteintrags ein SQL-Befehl hinzugefügt. Der Name des XML-Elements wird mit dem Attribut **XML-Element-Tag** des SQL-Befehls auf `Application` gesetzt.



Die Bedingungen `JOIN` und `WHERE` der SQL-Abfrage führen zur Auswahl von Applikationen, die nur dem Projekt "Consolidate Trading Applications" zugeordnet sind. Die `SELECT`-Anweisung der SQL-Abfrage für den SQL-Befehl definiert die XML-Attribute, die dem XML-Element `Application` hinzugefügt werden sollen. Dabei wird die Eigenschaft `REFSTR` der Objektklasse `Application` als erste Eigenschaft hinzugefügt. Die SQL-Abfrage:

```
SELECT app.REFSTR, app.NAME "Name", app.VERSION "Version",
app.STARTDATE "StartDate", app.ENDDATE "EndDate",
(SELECT COUNT(*) FROM RELATIONS rel WHERE rel.FROMREF = app.REFSTR
AND rel.PROPERTY = 'ApplicationGroups') AS "GroupAssignment"
FROM APPLICATION app
INNER JOIN PROJECT_ARCH pa ON pa.OBJECT = app.REFSTR
INNER JOIN PROJECT pro ON pro.REFSTR = pa.PROJECT
WHERE pro.NAME = 'Consolidate Trading Applications'
```

Generiert die folgende XML-Ausgabe:

```
<?xml version="1.0" encoding="utf-8"?>
<ProjectArchitecture>
  <Application Name="Summit" Version="3.1" StartDate="2003.03.28"
  EndDate="2016.01.16" GroupAssignment="2"/>
  <Application Name="GenLManager" Version="1.5"
  StartDate="2010.05.29" EndDate="2016.05.03"
  GroupAssignment="0"/>
</ProjectArchitecture>
```

Die Werte der XML-Attribute sind die Zeichenfolgenwerte der Eigenschaften für den aktuellen Datensatz. Die Konvertierung folgt den definierten Regeln:

- Beim ADIF-Exportschema können Regeln für die Konvertierung von Eigenschaftswerten in Zeichenfolgen konfiguriert werden. Diese Regeln sind in den Attributen des XML-Exporteintrags konfigurierbar (das bedeutet, dass für alle XML-Elemente in der Export-Datei dieselben Regeln gelten).

Sie können Datums- und Zahlenformate definieren sowie eine Zeichenfolge, die für die booleschen Werte `False` und `True` in die Ausgabedatei geschrieben werden soll. Eine Übersicht über die Attributeinstellungen im XML-Exporteintrag finden Sie unter [Konfigurieren des Exports in eine XML-Zieldatei](#).

- Datentypen der Eigenschaften von Alfabet -Objektklassen werden automatisch dem korrekten Formattyp zugeordnet und entsprechend den Einstellungen in den Exporteintragsattributen verarbeitet.

- Sie können definieren, dass numerische oder Zeichenfolgenwerte in Boolean konvertiert werden, indem Sie den Datentyp in der Alias-Spezifikation für die Eigenschaft als Suffix in Klammern angeben. Um als Boolean verarbeitet zu werden, muss der Eigenschaft-Alias angegeben werden als

```
<alias>[boolean]
```

und die Eigenschaft muss einen der folgenden Werte zurückgeben

- Den String `True` oder `False`.
- Eine Ganzzahl. Wenn die Ganzzahl 0 ist, wird sie als Boolean `False` betrachtet, alle anderen Ganzzahlen werden als Boolean `True` betrachtet.
- Das Attribut **Leere Attributwerte schreiben** des SQL-Befehls definiert, ob XML-Attribute ohne Werte in die XML-Ausgabe geschrieben werden, wenn eine Eigenschaft nicht ausgefüllt ist. Ein Festlegen des Attributs **Leere Attributwerte schreiben** auf `True` resultiert in folgender Ausgabe:

```
<Application Name="Summit" Version="3.1"IsVariant="" />
```

Wenn das Attribut auf `False` gesetzt wird, wird dasselbe XML-Element folgendermaßen geschrieben:

```
<Application Name="Summit" Version="3.1"/>
```



Im Beispiel wird die Anzahl an Applikationsgruppen, denen die Applikation zugeordnet ist, in das XML-Attribut `GroupAssignment` geschrieben. Wenn die Daten nur genutzt werden sollen, um herauszufinden, ob eine Applikation mindestens einer Applikationsgruppe zugeordnet ist, kann die `SELECT`-Anweisung der Alfabet -Abfrage so geändert werden, dass die von der untergeordneten `SELECT`-Anweisung zurückgegebene Ganzzahl in einen booleschen Wert umgewandelt wird:

```
SELECT app.REFSTR, app.NAME "Name", app.VERSION "Version",
app.STARTDATE "StartDate", app.ENDDATE "EndDate",
(SELECT COUNT(*) FROM RELATIONS rel WHERE rel.FROMREF = app.REFSTR
AND rel.PROPERTY = 'ApplicationGroups') AS
"GroupAssignment [boolean] "
FROM APPLICATION app
INNER JOIN PROJECT_ARCH pa ON pa.OBJECT = app.REFSTR
INNER JOIN PROJECT pro ON pro.REFSTR = pa.PROJECT
WHERE pro.NAME = 'Consolidate Trading Applications'
```

Generiert die folgende XML-Ausgabe, wobei `GroupAssignment` mit den Zeichenfolgenwerten für Boolean `True` und `False` ausgefüllt ist, die in den Attributen des Exporteintrags definiert sind:

```
<?xml version="1.0" encoding="utf-8"?>
<ProjectArchitecture>
  <Application Name="Summit" Version="3.1" StartDate="2003.03.28"
  EndDate="2016.01.16"GroupAssignment="yes" />
```

```

        <Application Name="GenLManager" Version="1.5"
        StartDate="2010.05.29"
        EndDate="2016.05.03"GroupAssignment="no" />
    </ProjectArchitecture>
    
```

Sie können dem Ordner **SQL-Befehle - DataExport** eines XML-Exporteintrags mehrere SQL-Befehle hinzufügen. Die Klassen-XML-Elemente werden der XML-Ausgabe in der Reihenfolge der Ausführung der SQL-Befehle hinzugefügt.

SQL-Befehle können hierarchisch strukturiert werden. Sie können den Befehlen der ersten Ebene untergeordnete Befehle hinzufügen. In der XML-Ausgabe führen untergeordnete Befehle dazu, dass untergeordnete XML-Elemente für ein XML-Element hinzugefügt werden. Wenn ein SQL-Befehl ausgeführt wird, wird der untergeordnete SQL-Befehl für jeden Datensatz des übergeordneten SQL-Befehls ausgeführt, um die untergeordneten XML-Elemente zu beurteilen, die den aktuellen Klassen-XML-Elementen hinzuzufügen sind.

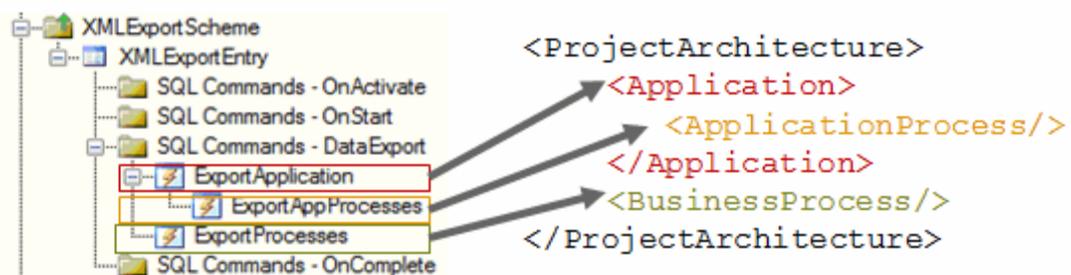
Die für untergeordnete Befehle definierten SQL-Abfragen müssen in einer Beziehung zum übergeordneten XML-Element stehen. Diese Beziehung wird durch Parameter hergestellt, die das übergeordnete XML-Element referenzieren. Ein übergeordnetes XML-Element kann mit dem Parameter @<element name> referenziert werden. Der Parameter wird durch den REFSTR ersetzt, der mit der ersten Eigenschaft der SELECT-Anweisung des übergeordneten SQL-Befehls angegeben wurde.



Die Business-Prozesse, die den Applikationen zugeordnet sind, sollen beispielsweise als Unterelemente <ApplicationProcess/> dem XML-Element <Application/> hinzugefügt werden, und die Business-Prozesse, die dem Projekt zugeordnet sind, sollen als XML-Elemente <BusinessProcess/> auf der obersten Hierarchieebene hinzugefügt werden.

Für die XML-Elemente <BusinessProcess/> auf der obersten Hierarchieebene wird dem Ordner **SQL-Befehle - Data Export** ein neuer SQL-Befehl hinzugefügt.

Für die XML-Elemente <ApplicationProcess/> wird dem vorhandenen SQL-Befehl für die Erstellung der <Application/>-Elemente ein untergeordneter Befehl hinzugefügt.



In der SQL-Abfrage für die XML-Elemente <ApplicationProcess/> werden die Applikationen auf der oberen Ebene mit dem Parameter @Application referenziert. Der Parameter gibt den REFSTR der Applikation zurück, auf die sich das übergeordnete XML-Element bezieht:

```

SELECT bp.REFSTR, bp.NAME "Name"
FROM BUSINESSPROCESS bp
INNER JOIN RELATIONS rel ON rel.FROMREF = bp.REFSTR
INNER JOIN APPLICATION app ON app.REFSTR = rel.TOREF
WHERE app.REFSTR = @Application AND rel.PROPERTY = 'Applications'
    
```

Die daraus resultierende Ausgabe zeigt die Daten zu den Business-Prozessen, die von den SQL-Abfragen in den zusätzlichen SQL-Befehlen definiert wurden:

```

<?xml version="1.0" encoding="utf-8"?>
<ProjectArchitecture>
  <Application Name="Summit" Version="3.1" StartDate="2003.03.28"
  EndDate="2016.01.16" GroupAssignment="yes">
    <ApplicationProcess Name="Equity Trading Trading" />
    <ApplicationProcess Name="Foreign Exchange Dealings" />
  </Application>
  <Application Name="GenLManager" Version="1.5"
  StartDate="2010.05.29" EndDate="2016.05.03" GroupAssignment="no">
    <ApplicationProcess Name="Custody Services" />
  </Application>
  <BusinessProcess Name="Equity Trading Trading" />
  <BusinessProcess Name="Foreign Exchange Dealings" />
  <BusinessProcess Name="Custody Services" />
</ProjectArchitecture>

```

Eigenschaftselement

Die Eigenschaften, die in der SQL-Anweisung eines SQL-Befehls für den Datenexport für eine Objektklasse definiert sind, können dem XML-Inhalt auch als XML-Elemente hinzugefügt werden, anstatt in XML-Attribute konvertiert zu werden. Dies kann auf zwei unterschiedliche Arten erfolgen:

- Definieren der Erzeugung von XML-Elementen über ein Präfix in der Alias-Spezifikation in der `SELECT`-Anweisung. Diese Methode erzeugt Eigenschafts-XML-Elemente ohne XML-Attribute. Der XML-Elementname entspricht der Alias-Spezifikation der `SELECT`-Anweisung des SQL-Befehls.



```

<Application Name="Summit">
  <Version>3.1</Version>
  <ObjectState>Active</ObjectState>
  <IsVariant>>false</IsVariant>
</Application>

```

- Definieren der Erzeugung von XML-Elementen über **Attribut**-Definitionen im **ADIF-Exporteintrag**. Diese Methode ist zu verwenden, um Eigenschafts-XML-Elemente mit XML-Attributen oder Eigenschafts-XML-Elemente mit gleichem Namen für mehrere Objektklasseneigenschaften zu erzeugen. Der Name des über eine **Attribut**-Definition erzeugten Eigenschafts-XML-Elements wird in der **Attribut**-Definition festgelegt. Das Eigenschafts-XML-Element hat stets ein XML-Attribut `Name`, dessen Wert mit einer Alias-Definition in der SQL-Anweisung übereinstimmt, die das Klassen-XML-Element festlegt, zu dem die Eigenschaft gehört.



Die **Attribut**-Definition kann beispielsweise dazu verwendet werden, Informationen zu Objektklasseneigenschaften in XML-Elementen `<Property />` zu speichern, bei denen ein XML-Attribut **Name** den Namen der Objektklasseneigenschaft definiert, für die dieses XML-Element erstellt wird, und ein XML-Attribut **Type** den Datentyp der Objektklasseneigenschaft definiert:

```

<Application Name="Summit">
  <Property Name="Version" Type="String">3.1</Property>

```

```

<Property Name="ObjectState"
Type="String">Active</Property>

<Property Name="IsVariant" Type="String">>false</Property>
</Application>

```

Folgendes gilt für beide Typen von Eigenschafts-XML-Elementen:

- Wenn der Wert der Objektklasseneigenschaft als CDATA in die Ausgabe geschrieben werden soll, müssen Sie der Alias-Spezifikation für die Objektklasseneigenschaft, die für das Schreiben in ein XML-Element definiert ist, ein Suffix [CDATA] hinzufügen.
- Das Attribut **Leere Elementwerte schreiben** des SQL-Befehls definiert, ob Eigenschafts-XML-Elemente ohne Werte in die XML-Ausgabe geschrieben werden, wenn eine Objektklasseneigenschaft nicht ausgefüllt ist. Ein Festlegen des Attributs **Leere Elementwerte schreiben** auf `True` resultiert in folgender Ausgabe:

```

<Application Name="Summit">
  <Version>3.1</Version>
  <IsVariant />
</Application>

```

Wenn das Attribut auf `False` gesetzt wird, wird dasselbe XML-Element folgendermaßen geschrieben:

```

<Application Name="Summit">
  <Version>3.1</Version>
</Application>

```

Definieren von Eigenschafts-XML-Elementen über die Alias-Spezifikation

Wenn Eigenschaften einer Objektklasse, die in der `SELECT`-Anweisung eines SQL-Befehls angegeben sind, als XML-Unterelemente und nicht als XML-Attribute des Klassen-XML-Elements in die Datei geschrieben werden sollen, müssen Sie für die Eigenschaft einen mit `E_` beginnenden Alias-Namen in der `SELECT`-Anweisung definieren. Der Wert der Eigenschaft wird dann als Dateninhalt in ein untergeordnetes XML-Element der Klasse geschrieben. Der Name des untergeordneten XML-Elements entspricht dem Alias-Namen für die Eigenschaft in der Abfrage, ohne das Präfix `E_`.



In diesem Beispiel soll die Objektklasseneigenschaft `Description` der Objektklasse `Application` in der Ausgabe enthalten sein. Da die Beschreibung sehr lang sein kann und möglicherweise Sonderzeichen und Zeilenumbrüche enthält, sollen die Daten als Eigenschafts-XML-Element und nicht als XML-Attribut des XML-Elements `<Application/>` hinzugefügt werden, und es soll ein `CDATA`-Element zum Umschließen des Beschreibungsinhalts verwendet werden. Die Eigenschaft "Beschreibung" wird folgendermaßen zur `SELECT`-Anweisung der SQL-Abfrage hinzugefügt:

```

SELECT app.REFSTR, app.NAME "Name", app.VERSION
"Version", app.STARTDATE "StartDate[date]", app.ENDDATE
"EndDate", app.VARIANT "IsVariant", (SELECT COUNT(*) FROM RELATIONS
rel WHERE rel.FROMREF = app.REFSTR AND rel.PROPERTY =
'ApplicationGroups') AS "GroupAssignment[boolean]", app.DESCRPTION
"E_Description[CDATA]"

```

```
FROM APPLICATION app
```

Daraus resultiert die folgende Ausgabe für eine der Applikationen in der Ausgabedatei:

```
<Application Name="Summit" Version="3.1" StartDate="2003.03.28"
EndDate="2016.01.16" GroupAssignment="yes">
  <Description><![CDATA [Unparalleled Document Processing and Check
Imaging Solution] ]></Description>
  <ApplicationProcess Name="Equity Trading Trading" />
  <ApplicationProcess Name="Foreign Exchange Dealings" />
</Application>
```

Definieren von Eigenschaftselementen über Attributdefinitionen

Wenn Eigenschaften einer Objektklasse, die in der `SELECT`-Anweisung eines **SQL-Befehls** angegeben sind, als XML-Unterelemente des Klassen-XML-Elements ohne eigene XML-Attribute in die Datei geschrieben werden sollen, müssen Sie Folgendes tun:

- 1) Fügen Sie in den **SQL-Befehl** -Exportdaten für die Objektklasse alle Eigenschaften hinzu, die Sie der `SELECT`-Anweisung als XML-Elemente für die Ausgabe hinzufügen möchten.
- 2) Klicken Sie für jede Eigenschaft, die als XML-Element exportiert werden soll, mit der rechten Maustaste auf den Ordner **Attribute** des Exporteintrags, und wählen Sie **Attribut erzeugen** aus. Definieren Sie im Attributfenster des neuen **Attribut** -Eintrags Folgendes:
 - **XML-Tag:** Geben Sie den Namen des XML-Elements ein.
 - **XML-Typ:** Wählen Sie ein `Element` aus dem Dropdown-Listefeld aus.
 - **Exportspalte:** Geben Sie den Alias-Namen in der `SELECT`-Anweisung des **SQL-Befehls** für die Objektklasseneigenschaft ein, die dieses XML-Element speichern soll.
 - **Titel:** Geben Sie den Wert für das XML-Attribut **Name** des XML-Elements ein.
 - **Datentyp:** Geben Sie den Typ der Daten ein, die im XML-Element als Wert gespeichert werden.
- 3) Um einem neuen Eigenschafts-XML-Element ein XML-Attribut hinzuzufügen, klicken Sie mit der rechten Maustaste im Explorer auf das **Attribut**, und wählen Sie **XML-Attribut erzeugen** aus. Definieren Sie im Attributfenster des neuen **XML-Attribut** -Eintrags Folgendes:
 - **Name:** Geben Sie den Namen des XML-Attributs ein.
 - **Wert:** Geben Sie den Wert für das XML-Attribut ein.

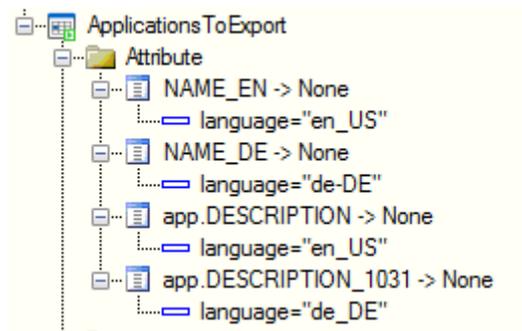


Im Beispiel sollen Informationen zu Applikationen exportiert werden, einschließlich der deutschen und englischen Version der Objektklasseneigenschaften `Name` und `Description`. Alle Objektklasseneigenschaften sollen als untergeordnete XML-Elemente `<Property />` des Elements `<Application />` gespeichert werden. Zusätzlich sollen dem Export Informationen zur Sprachkultur der exportierten Eigenschaft hinzugefügt werden.

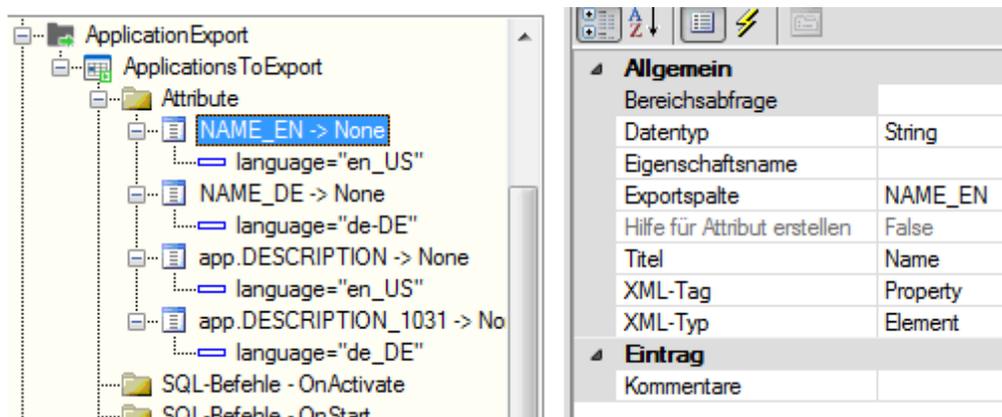
Der SQL-Befehl für den Export der Applikationsdaten enthält beide Sprachversionen der Eigenschaften `Name` und `Description` der Applikationen:

```
SELECT app.REFSTR,app.NAME + ' ' + app.VERSION As
"NAME_EN",app.NAME_1031 + ' ' + app.VERSION As "NAME_DE",
app.DESCRPTION, app.DESCRPTION_1031
FROM APPLICATION app
```

Für jede Objektklasseigenschaft wird ein **Attribut** -Eintrag erzeugt:



Der **Attribut** -Eintrag für die englische Version der Eigenschaft `Name` ist beispielsweise definiert als:



Für den Eintrag **Attribut** für die deutsche Version der Eigenschaft `Name` ist derselbe **Titel** definiert wie für die englische Version. Der Unterschied zwischen den beiden **Attribut** -Einträgen beschränkt sich auf die Zuordnung zu einem unterschiedlichen Teil der `SELECT`-Anweisung bei der Definition der **Exportspalte** und einer unterschiedlichen Wert-Definition für das XML-Attribut **language**.

Entsprechend beinhaltet die XML-Ausgabe des Exports zwei `<Property/>`-XML-Elemente, bei denen das XML-Attribut **Name** auf den Wert `Name` festgelegt ist, und zwei `<Property/>`-XML-Elemente, bei denen das XML-Attribut **Name** auf den Wert `Description` festgelegt ist. Die beiden `<Property/>`-XML-Elemente, bei denen der **Name** gleich ist, unterscheiden sich im Wert des XML-Attributs **language**. Der Inhalt des `<Property/>`-XML-Elements umfasst die jeweilige Übersetzung von Name und Beschreibung der Applikation:

```
<Application>
  <Property Name="Name" language="en-US">Customer Opinion DB
  6.1.4</Property>
  <Property Name="Name" language="de-DE">Kundenmeinungs-DB
  6.1.4</Property>
  <Property Name="Description" language="en-US">Database for
  storing results of customer surveys</Property>
  <Property Name="Description" language="de-DE">Datenbank zum
  Speichern der Ergebnisse von Kundenabfragen</Property>
</Application>
```

Sequenzelement

Sequenzelemente sind einfache XML-Elemente ohne XML-Attribute, die als Container für alle XML-Elemente dienen, die bei Ausführung eines SQL-Befehls erzeugt wurden. Ein Sequenz-XML-Element wird im Attributabschnitt des SQL-Befehls mit dem Attribut **Sequenz-XML-Tagname** definiert.



Auf der obersten Ebene des Beispiels werden sowohl `<Application/>`- als auch `<BusinessProcess/>`-XML-Elemente hinzugefügt. Um den Inhalt weiter zu strukturieren werden die XML-Elemente `<Application/>` in ein Sequenz-XML-Element `<ProjectApplications/>` geschrieben:

Kommentare	
Meldung	
Text	SELECT app.REFST
Untergeordnete Befehle	(Collection)
XML	
Befehlsparametername	
Leere Attributwerte schreiben	False
Leere Elementwerte schreiben	False
Sequenz-XML-Tag-Name	ProjectApplications
XML-Tag-Name	

Die XML-Elemente `<BusinessProcess/>` werden in ein Sequenz-XML-Element `<ProjectProcesses/>` geschrieben: Das daraus resultierende XML sieht jetzt folgendermaßen aus:

```
<?xml version="1.0" encoding="utf-8"?>
<ProjectArchitecture>
  <ProjectApplications>
    <Application Name="Summit" Version="3.1"
      StartDate="2003.03.28" EndDate="2016.01.16"
      GroupAssignment="yes">
      <Description>Unparalleled Document Processing and Check
        Imaging Solution</Description>
      <ApplicationProcess Name="Equity Trading Trading" />
      <ApplicationProcess Name="Foreign Exchange Dealings" />
    </Application>
    <Application Name="GenLManager" Version="1.5"
      StartDate="2010.05.29" EndDate="2016.05.03"
      GroupAssignment="no">
      <ApplicationProcess Name="Custody Services" />
    </Application>
  </ProjectApplications>
  <ProjectProcesses>
    <BusinessProcess Name="Equity Trading Trading" />
    <BusinessProcess Name="Foreign Exchange Dealings" />
    <BusinessProcess Name="Custody Services" />
  </ProjectProcesses>
</ProjectArchitecture>
```

Konfigurieren des Exports in kommagetrennte Datendateien

Daten können in durch Trennzeichen getrennte Formate in Dateien mit der Erweiterung .txt oder .csv exportiert werden.

Der Datenexport wird durch eine SQL-Anweisung `SELECT` ausgelöst. Das Ergebnis der SQL-Abfrage ist eine Datentabelle, die folgendermaßen in die Datei geschrieben wird:

- Für jede Zeile in der Tabelle gibt es eine Zeile in der Datei. Die erste Zeile enthält die Kopfzeile, die nachfolgenden Zeilen enthalten die Datensätze, die über die SQL-Abfrage gefunden wurden.
- Das Trennzeichen für Datenspalten innerhalb eines Datensatzes ist im ADIF-Exportschema konfigurierbar.
- Die Datenspaltenüberschriften in der ersten Zeile sind mit den Spaltennamen identisch, die aus der `SELECT`-Anweisung der Abfrage resultieren.



Daten, die eingebettete Zeilenumbrüche enthalten, können nicht korrekt in CSV-Dateien exportiert werden. Ein Zeilenumbruch wird als Anfang eines neuen Datensatzes interpretiert. Wenn Sie Daten exportieren möchten, die Zeilenumbrüche enthalten, wählen Sie ein anderes Ausgabeformat (beispielsweise den Export in eine XML-Datei). Zeilenumbrüche können in Eigenschaften vom Typ `Text` eingebettet werden. Im Normalfall haben für Objekte bereitgestellte Beschreibungen den Datentyp `Text`.



Die folgende SQL-Abfrage beispielsweise:

```
SELECT app.REFSTR, app.NAME, app.VERSION, app.VARIANT
FROM APPLICATION app
WHERE app.NAME LIKE 'A%';
```

Ergibt den folgenden Datensatz in einem tabellarischen Format (links) und im CSV-Format (rechts):

REFSTR	NAME	VERSION	VARIANT	REFSTR,NAME,VERSION,VARIANT
76-2576-0	ALLFiance PISA	2.9		76-2576-0,ALLFiance PISA,2.9,
76-2704-0	ARS BV	1.0		76-2704-0,ARS BV,1.0,
76-2708-0	ARS AM	1.0.1		76-2708-0,ARS AM,1.0.1,
76-2787-0	Administrative General Ledger	1.0		76-2787-0,Administrative General Ledger,1.0,
76-2790-0	alfabet SITM	2.0		76-2790-0,alfabet SITM,2.0,
76-2796-0	ARBI	1.2.1		76-2796-0,ARBI,1.2.1,
76-2797-0	ARIS	6.2.1		76-2797-0,ARIS,6.2.1,
76-2801-0	Aviation DB	1.0.2		76-2801-0,Aviation DB,1.0.2,
76-3217-0	ACCOUNT	1		76-3217-0,ACCOUNT,1,
76-3219-0	Announcements for Federal Reserve	1.0		76-3219-0,Announcements for Federal Reserve,1.0,
76-3268-0	AF Good Buy	3.0		76-3268-0,AF Good Buy,3.0,
76-3269-0	AF Enterprise Control	4.0		76-3269-0,AF Enterprise Control,4.0,
76-3275-0	AF Enterprise Control	3.1		76-3275-0,AF Enterprise Control,3.1,
76-3276-0	AF Good Buy	2.0		76-3276-0,AF Good Buy,2.0,
76-3277-0	AF HR Online	2.2		76-3277-0,AF HR Online,2.2,
76-3278-0	AF WorkPortal	1.0		76-3278-0,AF WorkPortal,1.0,
76-3285-0	AF HR Online EU	2.2 Var.	x	76-3285-0,AF HR Online EU,2.2 Var.,yes
76-3286-0	AF HR Online US	2.2 Var.	x	76-3286-0,AF HR Online US,2.2 Var.,yes
76-3289-0	AVS	4.2		
76-3305-0	AF HR Online	3.0		
76-3329-0	ACCOUNT	1.2		
76-3339-0	Asset Management	6.0		

Konfigurieren des Exports in eine Ziel-CSV- oder -TXT-Datei



Die folgende Konfiguration innerhalb eines ADIF-Exportschemas ist für den Datenexport aus der Alfabet-Datenbank in eine Datei im CSV-Format erforderlich:

- Erstellen Sie einen Exporteintrag für die Spezifikation der Zielexportdatei.
- Definieren Sie die SQL-Abfrage für die Definition des Inhalts der Datei im Unterordner **SQL-Befehle – DataExport** des Exporteintrags.



Sie dürfen nicht mehr als einen SQL-Befehl vom Typ `DataExport` für den Export in eine CSV- oder TXT-Datei definieren.

Der Workflow für die Definition des Exporteintrags wird nachfolgend beschrieben.

Sie können optional folgende Aktionen ausführen:

- Definieren Sie zusätzliche SQL-Befehle, um komplexe Exportszenarien zu definieren. Informationen hierzu finden Sie unter [Konfigurieren von Protokollierungsparametern](#).
- Strukturieren Sie die Exporteinträge in einer Exportgruppe. Informationen hierzu finden Sie unter [Automatische Erstellung mehrerer CSV-Exporteinträge in einer Exportgruppe](#).

So erstellen Sie einen Exporteintrag für den Export in CSV- oder TXT-Dateien:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das Exportschema, und wählen Sie **Eintrag erzeugen** aus. Dem Explorer wird ein neuer Exporteintrag-Knoten als Unterknoten des ADIF-Exportschemas hinzugefügt.
- 2) Klicken Sie im Explorer auf den neuen Exporteintrag-Knoten.
- 3) Legen Sie im Attributfenster rechts die folgenden Attribute für den Exporteintrag fest:
 - **Name:** Ändern Sie den Standardnamen des Eintrags in einen aussagekräftigen Namen. Der **Name** dient der Identifikation des Exporteintrags in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.
 - **Kommentare:** Optional können Sie einen Kommentar mit Informationen zum Export zur internen Verwendung eingeben.
 - **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung des Eintrags während des Exports zu aktivieren.
 - **Exporttabelle:** Geben Sie einen Namen für die Zieldatei für den Export ein. Die Exportdatei wird während des Exports erzeugt. Sie muss die Erweiterung `.txt` oder `.csv` haben.
 - **Exporttyp:** Wählen Sie `CSV` aus.
 - **Trennzeichen:** Wählen Sie das Trennzeichen aus, das in der Exportdatei zwischen den Dateneinträgen innerhalb eines Datensatzes verwendet werden soll. Jeder Datensatz wird in eine separate Zeile geschrieben.



Wenn der Wert für ein Datenfeld das Trennzeichen enthält, wird der Wert in Anführungszeichen geschrieben.

- **Verschlüsselung:** Wählen Sie die Verschlüsselung für die Daten in der Exportdatei aus dem Dropdown-Listenfeld aus.

- **Boolean 'False':** Wählen Sie den Zeichenfolgenwert aus, der in die Exportdatei geschrieben werden soll, wenn ein Attribut des Datentyps "Boolean" auf `False` gesetzt ist. Die Standardeinstellung ist `False`.
 -  Wenn eine boolesche Eigenschaft im Alfabet -Metamodell nicht obligatorisch ist und daher für einige Objekte auf NULL gesetzt ist, wird NULL nicht als `False` interpretiert, sondern als leeres Datenfeld. Wenn NULL als `False` behandelt werden soll, müssen Sie Ihre SQL-Abfrage entsprechend konfigurieren.
 - **Boolean 'True':** Wählen Sie den Zeichenfolgenwert aus, der in die Exportdatei geschrieben werden soll, wenn ein Attribut des Datentyps "Boolean" auf `True` gesetzt ist. `True` ist Standard.
 - **Anzahl der Dezimalziffern:** Definieren Sie die Anzahl der Stellen, die in der Ausgabe für den Dezimalteil der Zahlen definiert werden sollen. Die Standardeinstellung lautet "2". Zweieinhalb wird beispielsweise als "2,5" geschrieben.
 - **Anzahl der Dezimaltrennzeichen:** Definieren Sie das Dezimaltrennzeichen für Zahlen. Die Standardeinstellung ist ein Punkt.
 - **Anzahl der Gruppentrennzeichen:** Definieren Sie das Trennzeichen für Zahlengruppen. Die Standardeinstellung ist ein Komma. Dreitausend wird beispielsweise als "3,000" geschrieben.
 - **Datumsformat:** Definieren Sie das Format für Daten, die in die Exportdatei geschrieben werden. Der Standard ist `TT/MM/JJJJ`.
- 4) Erweitern Sie im Explorer den Exporteintrag-Knoten des neuen Exporteintrags.
 - 5) Klicken Sie mit der rechten Maustaste auf den Ordner **SQL-Befehle - DataExport**, und wählen Sie **SQL-Befehl erzeugen** aus. Dem Explorer wird ein neuer SQL-Befehlsknoten hinzugefügt.
 - 6) Klicken Sie auf den neuen SQL-Befehlsknoten, und geben Sie im Attributfenster Folgendes an:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität angibt, die mit dem SQL-Befehl implementiert wird.
 - **Befehlstyp:** Wählen Sie `DMLStatement` aus dem Dropdown-Listefeld aus.
 - **Ergebnistyp:** Wählen Sie `Undefined` aus.
 - **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Export auch dann ausgeführt werden soll, wenn die im Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Export im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
 - **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.
 - **Text:** Definieren Sie eine SQL-Abfrage mit einer `SELECT`-Anweisung, die einen Datensatz zurückgibt. Der resultierende Datensatz wird in die im Exporteintrag angegebene Zielfeld exportiert. Schreiben Sie die SQL-Abfrage entweder direkt in das Attributfeld, oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.

Automatische Erstellung mehrerer CSV-Exporteinträge in einer Exportgruppe

Exporteinträge können sich direkt unter dem ADIF-Exportschemaelement befinden oder Unterelemente einer Exportgruppe sein. Es ist hilfreich, Exporteinträge innerhalb einer Exportgruppe zu definieren, um das ADIF-Exportschema zu strukturieren. Darüber hinaus kann die Definition einer Exportgruppe dazu verwendet werden, beim Definieren des ADIF-Schemas in der ADIF-Konfigurationsbenutzeroberfläche mehrfache Exporteinträge in einem Batch-Prozess zu erzeugen.

Wenn Sie Daten zu mehreren Alfabet -Objektklassen in ein CSV-Format exportieren möchten, können Sie die Objektklassen während der Erzeugung einer Exportgruppe über die ADIF-Konfigurationsbenutzeroberfläche auswählen. Mit einer vorkonfigurierten SQL-Abfrage für den Datenexport, die alle Eigenschaften aller Objekte der ausgewählten Objektklasse für den Export auswählt, wird der Exportgruppe automatisch ein Exporteintrag pro ausgewählte Klasse hinzugefügt.

So erstellen Sie mehrere Exporteinträge innerhalb einer Exportgruppe:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Exportschema, dem Sie eine Exportgruppe hinzufügen möchten, und wählen Sie **Exportgruppe für Klassen erzeugen (CSV, Excel)** aus. Eine Klassenauswahl wird geöffnet.
- 2) Aktivieren Sie in der Klassenauswahl das Kontrollkästchen der Klassen, die Sie in den Export einbeziehen möchten.
- 3) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern. Eine Exportgruppe mit einem Exporteintrag für jede ausgewählte Klasse wird dem Explorer hinzugefügt. Für jeden Exporteintrag wird ein SQL-Befehl im Ordner **SQL-Befehle - DataExport** mit einer SQL-Abfrage vordefiniert, die alle Objekte der ausgewählten Klasse sucht und alle Eigenschaften der ausgewählten Klasse exportiert.
- 4) Wählen Sie im Explorer die neue Exportgruppe aus, und bearbeiten Sie im Attributfenster folgende Attribute:
 - **Name:** Geben Sie einen aussagekräftigen Namen für die Exportgruppe ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit der Exportgruppe implementiert wird.
 - **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung aller Exporteinträge innerhalb der Exportgruppe auszuwählen. Wählen Sie `False` aus, um die Ausführung zu deaktivieren.
- 5) Wählen Sie für jeden Exporteintrag den Exporteintrag im Explorer aus, und ändern Sie mindestens die folgenden Attribute:
 - **Exporttyp:** Wählen Sie `CSV` im Dropdown-Listefeld aus.
 - **Exporttabelle:** Ändern Sie die Dateierweiterung entweder auf `CSV` oder auf `txt`. Optional können Sie den Dateinamen ändern.

Optional können Sie alle anderen Attribute festlegen. Eine Beschreibung der Attribute eines Exporteintrags finden Sie im Abschnitt [Konfigurieren des Exports in eine Ziel-CSV- oder -TXT-Datei](#).

- 6) Navigieren Sie für jeden Exporteintrag zum SQL-Befehl im Ordner **SQL-Befehle - DataExport**, und ändern Sie den vorkonfigurierten Befehl entsprechend ihren Anforderungen. Informationen über die Konfiguration von Projektkennzahlensystemen finden Sie unter [Konfigurieren des Exports in eine Ziel-CSV- oder -TXT-Datei](#).



Wenn Sie komplexe Exportszenarien konfigurieren möchten, können Sie dem Exporteintrag zusätzliche SQL-Befehle hinzufügen. Weitere Informationen zur Definition von SQL-

Befehlen in einem Exporteintrag finden Sie unter [Konfigurieren von Protokollierungsparametern](#).

- 7) Optional können Sie die Reihenfolge der Exporteinträge in der Exportgruppe ändern. Klicken Sie auf die Schaltfläche **Durchsuchen**  im Attribut **Einträge**, um eine Liste aller Exporteinträge in der Exportgruppe zu öffnen. Klicken Sie auf einen der untergeordneten Knoten in der Liste und klicken Sie auf die Schaltflächen **Nach oben/Nach unten**  oben rechts, um das ausgewählte Element in der Liste zu verschieben. Wenn die Exporteinträge korrekt sortiert sind, klicken Sie auf **OK**, um ihre Änderungen auf das ADIF-Exportschema anzuwenden.

Konfigurieren des Exports in Microsoft® Excel®-Dateien

Daten können als Microsoft® Excel® XLS- und XLSX-Dateiformate exportiert werden.

Der Datenexport wird durch eine SQL-Anweisung `SELECT` ausgelöst. Das Ergebnis der SQL-Abfrage ist eine Datentabelle, die folgendermaßen in die Datei geschrieben wird:

- Es kann nur ein Datensatz per Microsoft® Excel®-Datei exportiert werden. Die Daten werden in das erste Tabellenblatt der Excel-Datei geschrieben; dieses wird mit „Export“ benannt.
- Die mit der `SELECT`-SQL-Anweisung definierten Datenspalten definieren die Anzahl und Inhalte der Spalten in der Microsoft® Excel-Tabelle. Optional kann ADIF so konfiguriert werden, dass ab einer definierten Position in der Anzahl der Spalten Daten in die Datei geschrieben werden.
- Jede Zeile der Tabelle entspricht einer Zeile in der Datei. Die erste Zeile enthält die Tabellentitel, die nachfolgenden Zeilen enthalten die per SQL-Abfrage gefundenen Datensätze. Optional kann ADIF so konfiguriert werden, dass ein festgelegter Text als Titel in die erste Zeile des Microsoft Excel - Tabellenblatts geschrieben wird und die Ergebnistabelle in der zweiten Zeile beginnt. Der Titel wird mit rotem Text auf grüner Hintergrundfarbe dargestellt:

	A	B	C	D
1	Application Data			
2	REFSTR	NAME	VERSION	OBJEC
3	76-2518-0	Business EA	2.2	Active
4	76-2525-0	Groupware S	2.2	Active

- Die Datenspaltenüberschriften in der ersten Zeile sind mit den Spaltennamen identisch, die aus der `SELECT`-Anweisung der Abfrage resultieren.
- Excel interpretiert Zahlenformate, Datumsformate und boolesche Werte nach Ihren Standardeinstellungen für Ihre Microsoft® Excel®-Anwendung und zeigt sie entsprechend an.
- Die maximale Zeilenanzahl in einer Excel-Tabelle ist in einigen Microsoft® Excel®-Versionen auf 64.000 beschränkt. Darüber hinaus können Sie die Zeilenanzahl im ADIF-Exportschema weiter einschränken. Wenn die Anzahl der Zeilen im exportierten Datensatz die in einer Tabelle zulässige Anzahl der Zeilen überschreitet, wird eine neue Datei mit den verbleibenden Daten erstellt. Dateien werden fortlaufend als <Dateiname>.xls, <Dateiname>1.xls usw. nummeriert.
- Während des Exports können Dropdown-Listen mit vordefinierten Werten für Zellen in der Excel-Tabelle erzeugt werden. Dieser Mechanismus ist von Bedeutung, wenn Daten von einem Anwender in der Excel-Tabelle bearbeitet und anschließend wieder in die Alfabet-Datenbank importiert werden sollen.



Die folgende SQL-Abfrage beispielsweise:

```
SELECT app.REFSTR, app.NAME, app.VERSION, app.VARIANT
FROM APPLICATION app
WHERE app.NAME LIKE 'A%';
```

Ergibt den folgenden Datensatz in einem tabellarischen Format (links) und im Microsoft® Excel®-Format (rechts):

REFSTR	NAME	VERSION	VARIANT		A	B	C	D
76-2576-0	ALLFiance PISA	2.9		1	REFSTR	NAME	VERSION	VARIANT
76-2704-0	ARS BV	1.0		2	76-2576-0	ALLFiance PI	2.9	
76-2708-0	ARS AM	1.0.1		3	76-2704-0	ARS BV	1.0	
76-2787-0	Administrative General Ledger	1.0		4	76-2708-0	ARS AM	1.0.1	
76-2790-0	alfabet SITM	2.0		5	76-2787-0	Administrativ	1.0	
76-2796-0	ARBI	1.2.1		6	76-2790-0	alfabet SITM	2.0	
76-2797-0	ARIS	6.2.1		7	76-2796-0	ARBI	1.2.1	
76-2801-0	Aviation DB	1.0.2		8	76-2797-0	ARIS	6.2.1	
76-3217-0	ACCOUNT	1		9	76-2801-0	Aviation DB	1.0.2	
76-3219-0	Announcements for Federal Reserve	1.0		10	76-3217-0	ACCOUNT	1	
76-3268-0	AF Good Buy	3.0		11	76-3219-0	Announceme	1.0	
76-3269-0	AF Enterprise ConTrol	4.0		12	76-3268-0	AF Good Buy	3.0	
76-3275-0	AF Enterprise ConTrol	3.1		13	76-3269-0	AF Enterprise	4.0	
76-3276-0	AF Good Buy	2.0		14	76-3275-0	AF Enterprise	3.1	
76-3277-0	AF HR Online	2.2		15	76-3276-0	AF Good Buy	2.0	
76-3278-0	AF WorkPortal	1.0		16	76-3277-0	AF HR Online	2.2	
76-3285-0	AF HR Online EU	2.2 Var.	x	17	76-3278-0	AF WorkPort	1.0	
76-3286-0	AF HR Online US	2.2 Var.	x	18	76-3285-0	AF HR Online	2.2 Var.	WAHR
76-3289-0	AVS	4.2		19	76-3286-0	AF HR Online	2.2 Var.	WAHR
76-3306-0	AF HR Online	3.0		20	76-3289-0	AVS	4.2	
76-3329-0	ACCOUNT	1.2		21	76-3306-0	AF HR Online	3.0	
76-3339-0	Asset Management	6.0		22	76-3329-0	ACCOUNT	1.2	
				23	76-3339-0	Asset Manag	6.0	

Konfigurieren des Exports in eine Microsoft® Excel®-Zieldatei



Die folgende Konfiguration innerhalb eines ADIF-Exportschemas ist für den Datenexport aus der Alfabet-Datenbank in eine Microsoft® Excel®-Datei erforderlich:

- Erstellen Sie einen Exporteintrag für die Spezifikation der Zielexportdatei.
- Definieren Sie die SQL-Abfrage für die Definition des Inhalts der Datei im Unterordner **SQL-Befehle - DataExport** des Exporteintrags.



Sie dürfen nicht mehr als einen SQL-Befehl vom Typ `DataExport` für den Export in eine Microsoft® Excel®-Zieldatei definieren.

Der Workflow für die Definition des Exporteintrags wird nachfolgend beschrieben.

Sie können optional folgende Aktionen ausführen:

- Definieren Sie zusätzliche SQL-Befehle, um komplexe Exportszenarien zu definieren. Informationen hierzu finden Sie unter [Konfigurieren von Protokollierungsparametern](#).
- Strukturieren Sie die Exporteinträge in einer Exportgruppe. Informationen hierzu finden Sie unter [Automatische Erstellung mehrerer CSV-Exporteinträge in einer Exportgruppe](#).

So erzeugen Sie einen Exporteintrag für den Export in Microsoft Excel -Dateien:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das Exportschema, und wählen Sie **Eintrag erzeugen** aus. Dem Explorer wird ein neuer Exporteintrag-Knoten als Unterknoten des ADIF-Exportschemas hinzugefügt.
- 2) Klicken Sie im Explorer auf den neuen Exporteintrag-Knoten.
- 3) Legen Sie im Attributfenster rechts Folgendes für den Exporteintrag fest:

- **Name:** Ändern Sie den Standardnamen des Eintrags in einen aussagekräftigen Namen. Der **Name** dient der Identifikation des Exporteintrags in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.
- **Kommentare:** Optional können Sie einen Kommentar mit Informationen zum Export zur internen Verwendung eingeben.
- **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung des Eintrags während des Exports zu aktivieren.
- **Exporttabelle:** Geben Sie einen Namen für die Zielfeld für den Export ein. Die Exportdatei wird während des Exports erzeugt. Sie muss die Erweiterung `.xls` oder `.xlsx` haben. Die Dateierweiterung muss mit der im Attribut **Exporttyp** ausgewählten Erweiterung identisch sein.
- **Exporttyp:** Wählen Sie `XLS` aus, um in ein Microsoft® Excel® 2003-Format zu exportieren, oder `XLSX`, um in ein Microsoft® Excel® 2010-Format zu exportieren.
- **Excel-Höchstanzahl für Einträge:** Definieren Sie die maximale Anzahl von Datensätzen, die der Zielfeld hinzugefügt werden können. Wenn die Anzahl der Datensätze im Datensatz die Höchstzahl der Datensätze überschreitet, werden die verbleibenden Daten in eine neue Datei mit dem Namen `<Dateiname><laufende Nummer><Erweiterung>` geschrieben. Sie können eine beliebige Anzahl von Datensätzen, die höher ist als 100 oder `-1`, definieren, um Daten ohne Einschränkung bei der Anzahl der Datensätze in die Zielfeld zu schreiben.



Stellen Sie sicher, dass die konfigurierte Anzahl den technischen Einschränkungen der verwendeten Microsoft® Excel®-Version entspricht. Einige Excel®-Anwendungen können keine Tabellen mit mehr als 64.000 Datensätzen lesen.

- **Überschriftentext:** Wenn Sie möchten, dass in dem Microsoft Excel -Tabellenblatt ein allgemeiner Titel über dem Datensatz angezeigt wird, geben Sie den Text des Titels ein. Sie können im Attributfeld auf die Schaltfläche  klicken, um einen Texteditor zu öffnen. Wenn Sie einen Zeilenumbruch einfügen, ist dieser Zeilenumbruch auch im Titeltext in der Microsoft Excel -Datei enthalten.
- **Exportstartspalte:** Geben Sie die Nummer der ersten Spalte des exportierten Datensatzes in das Microsoft Excel -Tabellenblatt ein. Die erste Spalte des Tabellenblatts hat die Nummer 0.



Um beispielsweise ein Microsoft Excel -Tabellenblatt mit drei leeren Spalten vor den exportierten Daten zu erzeugen, setzen Sie das Attribut Exportstartspalte auf 3:

	A	B	C	D	E	F
1				REFSTR	NAME	VERSION
2				76-2518-0	Business EA	2.2
3				76-2525-0	Groupware S	2.2
4				76-2538-0	Mafo-Portal	2.6

- 4) Erweitern Sie im Explorer den Exporteintrag-Knoten des neuen Exporteintrags.

- 5) Klicken Sie mit der rechten Maustaste auf den Ordner **SQL-Befehle – DataExport**, und wählen Sie **SQL-Befehl erzeugen** aus. Dem Explorer wird ein neuer SQL-Befehlsknoten hinzugefügt.
- 6) Klicken Sie auf den neuen SQL-Befehlsknoten, und geben Sie die folgenden Attribute im Attributfenster an:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
 - **Befehlstyp:** Wählen Sie `DMLStatement` aus dem Dropdown-Listenfeld aus.
 - **Ergebnistyp:** Wählen Sie `Undefined` aus.
 - **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Export auch dann ausgeführt werden soll, wenn die im Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Export im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
 - **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.
 - **Text:** Definieren Sie eine SQL-Abfrage mit einer `SELECT`-Anweisung, die einen Datensatz zurückgibt. Der resultierende Datensatz wird in die im Exporteintrag angegebene Zielfeld exportiert. Schreiben Sie die SQL-Abfrage entweder direkt in das Attributfeld, oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.

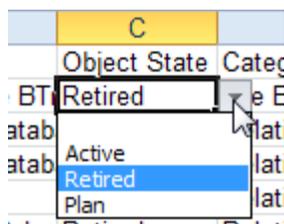
Erstellen von Dropdown-Listenfeldern in der exportierten Excel-Datei

Felder in der resultierenden Exportdatei können während des Exports als Dropdown-Listenfeld mit einem Satz definierter Werte geschrieben werden. Wenn die Daten exportiert werden, um in der Datei aktualisiert und anschließend per ADIF-Importschema wieder in die Alfabet-Datenbank importiert zu werden, stellt dieser Mechanismus sicher, dass die Person, die die Daten bearbeitet, keine ungültigen Werte eingibt.



Beispiel:

- Eine Eigenschaft basiert auf einer Aufzählung. Die für die Aufzählung definierten Werte können aus einer Dropdownliste ausgewählt werden. Es können keine ungültigen Werte produziert werden, in dem Tippfehler gemacht werden oder ein Benutzer Werte hinzufügt, die nicht in der Aufzählung enthalten sind.



- Eine Eigenschaft enthält eine Verknüpfung zu einem anderen Objekt in der Alfabet-Datenbank. Die Namen der Objekte, die Ziel der Verknüpfung sein können, werden in einer Dropdown-Liste definiert. Um beispielsweise Komponenten einer Komponentenkategorie zuzuordnen, werden die in der Alfabet-Datenbank definierten

Komponentenkategorien in der Liste angezeigt. Die Person, die die Daten bearbeitet, kann Beziehungen angeben, ohne die Alfabet -Benutzeroberfläche zu öffnen, um herauszufinden, welche Komponentenkategorien zur Verfügung stehen.

Name	Object State	Category	STA
Pervasive BT	Retired	File Based	30
Oracle Datab	Retired	File Based	30
Oracle Datab	Active	Hardware	04
IBM DB2	Retired	Hierarchical Data	24
Sybase Adva	Retired	Messaging System	01
IBM DB2	Retired	MS Office Comp	26
Microsoft Acc	Retired	Multidimensional	25
Microsoft SQL	Active	mySAP AddOns	12

So definieren Sie eine Dropdown-Liste in der Microsoft Excel -Ausgabedatei des Exports:

- 1) Klicken Sie mit der rechten Maustaste auf den Ordner **Attribute** des Exporteintrags, und wählen Sie **Attribut erzeugen** aus. Definieren Sie im Attributfenster des neuen **Attribut** -Eintrags Folgendes:
 - **Exportspalte:** Geben Sie für die Spalte im exportierten Datensatz, auf die das Dropdown-Listefeld angewendet werden soll, den Alias ein, der in der **SELECT**-Anweisung des **SQL-Befehls** für den Datenexport definiert ist.
 - **Bereichsabfrage:** Geben Sie eine native SQL-Abfrage ein, die die Daten zurückgibt, die in der Dropdown-Liste angezeigt werden sollen. Für jeden durch die Abfrage gefundenen Datensatz wird der Dropdown-Liste eine Zeile hinzugefügt. Alle in der **SELECT**-Anweisung der **Bereichsabfrage** definierten Daten werden in der Zeile der Dropdown-Liste angezeigt. Wenn die **SELECT**-Anweisung mehrere Argumente enthält, werden die Werte mit einem Leerzeichen als Trennzeichen in die Zeile geschrieben.



Wenn das Dropdown-Listefeld Werte anzeigen soll, die beispielsweise aus einer Aufzählung abgeleitet sind, kann die **Bereichsabfrage** so definiert werden, dass sie Textzeichenfolgen zurückgibt:

```
SELECT 'Active' UNION ALL SELECT 'Retired' UNION ALL SELECT
'Plan'
```

If the drop-down list shall return a list of available objects in the Alfabet-Datenbank, the **Range Query** can read the data directly from the database. Please note that the drop-down list must enable the person processing the exported data to identify the object unambiguously. For example, a drop-down list that allows to select applications must include application name and version, because uniqueness is only implemented for the combination of both:

```
SELECT NAME, VERSION
FROM APPLICATION
ORDER BY NAME
```

- **Kommentare:** Optional können Sie einen Kommentar mit Informationen zur Definition zur internen Verwendung eingeben.

Automatische Erstellung mehrerer Excel-Exporteinträge in einer Exportgruppe

Exporteinträge können sich direkt unter dem ADIF-Exportschemaelement befinden oder Unterelemente einer Exportgruppe sein. Es ist hilfreich, Exporteinträge innerhalb einer Exportgruppe zu definieren, um das ADIF-Exportschema zu strukturieren. Darüber hinaus kann die Definition einer Exportgruppe dazu verwendet werden, beim Definieren des ADIF-Schemas in der ADIF-Konfigurationsbenutzeroberfläche mehrfache Exporteinträge in einem Batch-Prozess zu erzeugen.

Wenn Sie Daten zu mehreren Alfabet -Objektklassen in eine Microsoft® Excel®-Datei exportieren möchten, können Sie diese Objektklassen während der Erzeugung einer Exportgruppe über die ADIF-Konfigurationsbenutzeroberfläche auswählen. Mit einer vorkonfigurierten SQL-Abfrage für den Datenexport, die alle Eigenschaften aller Objekte der ausgewählten Objektklasse für den Export auswählt, wird der Exportgruppe automatisch ein Exporteintrag pro ausgewählte Klasse hinzugefügt.

So erstellen Sie mehrere Exporteinträge innerhalb einer Exportgruppe:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf das ADIF-Exportschema, für das Sie eine Exportgruppe hinzufügen möchten, und wählen Sie **Exportgruppe für Klassen erzeugen (CSV, Excel)** aus. Eine Klassenauswahl wird geöffnet.
- 2) Aktivieren Sie in der Klassenauswahl das Kontrollkästchen der Objektklassen, die Sie in den Export einbeziehen möchten.
- 3) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern. Eine Exportgruppe mit einem Exporteintrag für jede ausgewählte Objektklasse wird dem Explorer hinzugefügt. Für jeden Exporteintrag wird ein SQL-Befehl im Ordner **SQL-Befehle - DataExport** mit einer SQL-Abfrage vordefiniert, die alle Objekte der ausgewählten Objektklasse findet und alle Objektklasseneigenschaften der ausgewählten Objektklasse exportiert.
- 4) Wählen Sie im Explorer die neue Exportgruppe aus, und bearbeiten Sie im Attributfenster folgende Attribute:
 - **Name:** Geben Sie einen aussagekräftigen Namen für die Exportgruppe ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit der Exportgruppe implementiert wird.
 - **Ist aktiv:** Wählen Sie `True` aus, um die Ausführung aller Exporteinträge innerhalb der Exportgruppe auszuwählen. Wählen Sie `False` aus, um die Ausführung zu deaktivieren.
- 5) Optional können Sie die Attribute der Exporteinträge entsprechend Ihren Bedürfnissen ändern. Eine Beschreibung der Attribute eines Exporteintrags finden Sie im Abschnitt [Konfigurieren des Exports in eine Ziel-CSV- oder -TXT-Datei](#).
- 6) Navigieren Sie für jeden Exporteintrag zum SQL-Befehl im Ordner **SQL SQL-Befehle - DataExport** und ändern Sie den vorkonfigurierten Befehl entsprechend Ihren Anforderungen. Informationen über die Konfiguration von Projektkennzahlensystemen finden Sie unter [Konfigurieren des Exports in eine Ziel-CSV- oder -TXT-Datei](#).



Wenn Sie komplexe Exportszenarien konfigurieren möchten, können Sie dem Exporteintrag zusätzliche SQL-Befehle hinzufügen. Weitere Informationen zur Definition von SQL-Befehlen in einem Exporteintrag finden Sie unter [Konfigurieren von Protokollierungsparametern](#).

- 7) Optional können Sie die Reihenfolge der Exporteinträge in der Exportgruppe ändern. Klicken Sie auf die Schaltfläche **Durchsuchen**  im Attribut **Einträge**, um eine Liste aller Exporteinträge in der Exportgruppe zu öffnen. Klicken Sie auf einen der untergeordneten Knoten in der Liste und klicken Sie auf die Schaltflächen **Nach oben/Nach unten**  oben rechts, um das ausgewählte Element

in der Liste zu verschieben. Wenn die Exporteinträge korrekt sortiert sind, klicken Sie auf **OK**, um ihre Änderungen auf das ADIF-Exportschema anzuwenden.

Konfigurieren von SQL-Befehlen für optionale Exporterweiterungen

Die Standardexportkonfiguration, die im Abschnitt [Standardkonfiguration des Datenexports](#) beschrieben wird, kann durch Definition zusätzlicher SQL-Befehle erweitert werden. Beim Erzeugen von ADIF-Exportschemata über den ADIF-Konfigurationsexplorer, werden dem Explorer automatisch Ordner für SQL-Befehle hinzugefügt, wenn für das Element SQL-Befehle definiert werden können. Die Ordner legen auch den Typ des SQL-Befehls fest.



In der XML-Datei des ADIF-Exportschemas verfügt jeder SQL-Befehl über ein Attribut **Typ**. Für jeden zulässigen Typ wird dem Explorer ein separater Ordner namens **SQL-Befehle - <Typ>** hinzugefügt. Wenn SQL-Befehle in einem Ordner definiert werden, wird das Attribut **Typ** automatisch in der XML-Definition festgelegt. Wenn Sie SQL-Befehle in einem XML-Editor angeben, muss das Attribut **Typ** manuell definiert werden.

Folgende SQL-Befehlstypen sind verfügbar:

Typ	Position im ADIF-Schema	Beschreibung
OnActivate	Exportgruppen und Exporteinträge	<p>SQL-Befehle vom Typ <code>OnActivate</code> definieren Bedingungen für die Ausführung des Elements, für das sie definiert sind. Informationen hierzu finden Sie unter Konfigurieren der bedingten Ausführung von Teilen des Importschemas.</p> <p>Hinweis: Von <code>OnActivate</code>-Befehlen ausgelöste Änderungen werden nicht zurückgenommen, wenn die Option Änderungen beibehalten im Importschema auf <code>False</code> gesetzt ist.</p>
DataExport	Exporteinträge	<p>SQL-Befehle vom Typ <code>DataExport</code> definieren die Daten, die in die Zielexportdatei oder die Exportdatenbanktabelle geschrieben werden sollen. Jeder Exporteintrag muss einen SQL-Befehl vom Typ <code>DataExport</code> enthalten. Die zulässige Anzahl an SQL-Befehlen für den Datenexport hängt vom Zielformat ab. Informationen hierzu finden Sie unter Standardkonfiguration des Datenexports.</p>
OnStart	Exporteinträge	<p>SQL-Befehle vom Typ <code>OnStart</code> werden vor dem SQL-Befehl vom Typ <code>DataExport</code> ausgeführt und SQL-Befehle vom Typ <code>OnComplete</code> werden nach dem SQL-Befehl vom Typ <code>DataExport</code> ausgeführt. Die Befehle ermöglichen es, vor oder nach dem Export spezielle Operationen für die Alfabet-Datenbank auszuführen. Die SQL-Befehle können für folgende Zwecke definiert werden:</p> <ul style="list-style-type: none"> • Definition benutzerdefinierter Debug-Informationen, die während der Ausführung von Exporten in die Protokolldatei geschrieben

Typ	Position im ADIF-Schema	Beschreibung
		<p>werden. Dies ist im Abschnitt Konfigurieren des Inhalts von Protokollmeldungen beschrieben.</p> <ul style="list-style-type: none"> Manipulation der Alfabet-Datenbank vor und/oder nach dem Datenexport. Die Definition von SQL-Befehlen für diesen Zweck wird in diesem Abschnitt beschrieben.
On-Complete	Exporteinträge	

Eines der wichtigsten Features der ADIF-Schnittstelle ist die flexible Konfiguration. Das ADIF-Exportschema kann nicht nur so konfiguriert werden, dass einfach bestehende Daten aus einer Alfabet-Datenbank -Tabelle in eine Exportdatei oder eine externe Datenbanktabelle ausgelesen werden. Sie können ADIF auch dazu konfigurieren, vor oder nach dem tatsächlichen Datenexport zusätzliche Operationen für die Alfabet-Datenbank oder, beim Export in eine externe Datenbank, für die externen Datenbanktabelle auszuführen. SQL-Abfragen vom **Befehlstyp** `DDLStatement` oder `DMLStatement` können in beliebiger Anzahl und Reihenfolge definiert werden.



Wenn SQL-Abfragen nicht ausreichen, um den gewünschten Export zu implementieren, können benutzerdefinierte DLLs bei der Software AG angefordert werden, um die Funktionalität bereitzustellen. Der benutzerdefinierte Code wird dem ADIF-Exportschema über einen SQL-Befehl mit dem **Befehlstyp** `StoredProcedure` hinzugefügt. Die korrekte Spezifikation von Konfigurationselementen und Attributen hängt vom benutzerdefinierten Code ab. Eine Beschreibung auf individueller Basis wird mit dem Code mitgeliefert.

Basierend auf der Reihenfolge der Ausführung der nachfolgend beschriebenen SQL-Anweisungen werden gespeicherte Prozeduren auf die gleiche Art und Weise ausgeführt wie DML-Anweisungen.

Wenn Sie SQL-Befehle für ein ADIF-Exportschema konfigurieren, müssen Sie die Ausführungsreihenfolge berücksichtigen. Die Elemente eines ADIF-Schemas werden in der folgenden Reihenfolge ausgeführt:

- Die SQL-Befehle vom Typ `OnActivate` werden zuerst ausgewertet, und die entsprechenden Elemente der ADIF-Schemata werden in den Ausführungsplan einbezogen oder davon ausgeschlossen.
- Alle SQL-Befehle vom Typ `OnStart` und dem **Befehlstyp** `DDLStatement`, deren Ziel die Alfabet-Datenbank ist, werden in der Reihenfolge ausgeführt, in der sie im ADIF-Exportschema angegeben sind.
- Alle Exporteinträge und Exportdatengruppen, die im ADIF-Schema angegeben sind, werden dann in der Reihenfolge verarbeitet, in der sie im ADIF-Exportschema angegeben sind
- Exportdatengruppen für den Export in externe Datenbanken werden folgendermaßen verarbeitet:
 - Die Verbindung mit der externen Datenbank wird hergestellt.

- Alle DDL-Anweisungen von SQL-Befehlen des Typs `OnStart`, die die externe Datenbank zum Ziel haben, die in den Exporteinträgen der Datenbank-Exportgruppe konfiguriert ist, werden in der Reihenfolge ausgeführt, in der sie in der Exportgruppe definiert sind.
- Die Exporteinträge werden wie folgt in der Reihenfolge verarbeitet, in der sie innerhalb der Datenbank-Exportgruppe konfiguriert sind:
 - Alle DML-Anweisungen von SQL-Befehlen des Typs `OnStart` werden in der Reihenfolge ausgeführt, in der sie im Exporteintrag angegeben sind.
 - Der SQL-Befehl des Typs `DataExport` wird ausgeführt.
 - Alle DML-Anweisungen von SQL-Befehlen vom Typ `OnComplete` werden in der Reihenfolge ausgeführt, in der sie im Exporteintrag angegeben sind. Diese SQL-Befehle können entweder die Alfabet oder die Zieldatenbank als Ziel haben.
- Alle DDL-Anweisungen von SQL-Befehlen des Typs `OnComplete`, die die externe Datenbank zum Ziel haben, die in den Exporteinträgen der Datenbank-Exportgruppe konfiguriert ist, werden in der Reihenfolge ausgeführt, in der sie in der Exportgruppe definiert sind.
- Alle anderen Exporteinträge werden wie folgt verarbeitet, unabhängig von ihrer Position inner- oder außerhalb einer Exportgruppe:
 - Alle DML-Anweisungen von SQL-Befehlen des Typs `OnStart` werden in der Reihenfolge ausgeführt, in der sie im Exporteintrag angegeben sind.
 - Die SQL-Befehle des Typs `DataExport` werden ausgeführt.
 - Die SQL-Befehle des Typs `OnComplete` werden in der Reihenfolge ausgeführt, in der sie im Exporteintrag angegeben sind.
- Alle DDL-Anweisungen von SQL-Befehlen vom Typ `OnComplete`, die die für die Exporteinträge im ADIF-Exportschema konfigurierte Alfabet-Datenbank als Ziel haben, werden in der Reihenfolge ausgeführt, in der sie im ADIF-Exportschema definiert sind.

In diesem Abschnitt wird im Allgemeinen beschrieben, wie Sie im ADIF-Konfigurationsexplorer einen SQL-Befehl innerhalb eines Ordners **SQL-Befehle - OnStart** oder **SQL-Befehle - OnComplete** definieren. Des Weiteren werden zwei Anwendungsfälle beschrieben, die durch die Definition von SQL-Befehlen abgedeckt werden können:

- *Use-Case: Aktualisieren von bestehenden Datensätzen in einer Zieldatenbank*
- *Anwendungsfall: Definieren exportbezogener benutzerdefinierter Eigenschaften*
- *Anwendungsfall: Datenrestrukturierung vor dem Export mit temporären Tabellen*

Erzeugen eines SQL-basierten Berichts

So spezifizieren Sie einen SQL-Befehl, der vor oder nach dem Datenexport für die Alfabet-Datenbank oder eine externe Zieldatenbank ausgeführt werden soll:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf den **SQL-Befehle - OnStart** oder **SQL-Befehle - OnComplete**, der dem Explorer automatisch hinzugefügt wird, wenn ein **Exporteintrag** -Knoten erzeugt wird.

- 2) Wählen Sie im Kontextmenü **SQL-Befehl erzeugen** aus. Dem Ordner wird ein neues Element **SQL-Befehl** als untergeordneter Knoten des Ordners hinzugefügt.
- 3) Klicken Sie auf den neuen SQL-Befehlsknoten, und geben Sie im Attributfenster Folgendes an:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität angibt, die mit dem SQL-Befehl implementiert wird.
 - **Befehlstyp:** Wählen Sie in der Dropdownliste den Typ der SQL-Abfrage aus, den Sie definieren möchten. Wählen Sie:
 - `DMLStatement` für SQL-Datenmanipulationsanweisungen wie beispielsweise `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `MERGE`.
 - `DDLStatement` für SQL-Datendefinitionsanweisungen wie beispielsweise `CREATE`, `ALTER`, `DROP`, `TRUNCATE`, `RENAME`.



DDL-Anweisungen sind nur zulässig, um neue Datenbanktabellen zu erzeugen, zu verändern oder zu verwerfen. Diese Befehle dürfen nicht auf Alfabet-Datenbank -Standardtabellen angewendet werden.

- **Ergebnistyp:** Wählen Sie `Undefined` aus.
- **Anwenden auf:** Wenn der SQL-Befehl für den Export in eine externe Datenbanktabelle definiert ist, definieren Sie die Datenbank, an der der SQL-Befehl ausgeführt wird:
 - `local` Zur Ausführung auf der Alfabet-Datenbank
 - `external` zum Ausführen an der externen Datenbank
- **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Export auch dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Export im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
- **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.
- **Text:** Definieren Sie die SQL-Abfrage für den Befehl. Schreiben Sie die SQL-Abfrage entweder direkt in das Attributfeld, oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.

Use-Case: Aktualisieren von bestehenden Datensätzen in einer Zieldatenbank

Wenn Sie Daten in eine externe Datenbanktabelle exportieren möchten, die bereits Daten enthält, müssen Sie im ADIF-Exportschema Mechanismen implementieren, die die Datenintegrität in der Zieldatenbanktabelle sicherstellen. Einschränkungen für Primärschlüssel und eindeutige Konfigurationen der Zieldatenbank hängen von der individuellen Konfiguration der Zieldatenbank ab und können deshalb vom ADIF-Datenexportmechanismus nicht berücksichtigt werden.

Während der Ausführung des SQL-Befehls für den Datenexport werden Daten einfach in neuen Datensätzen in die externe Datenbanktabelle geschrieben. Wenn dies zu Verstößen gegen Schlüssel führt, wird der Export angehalten, und eine Fehlermeldung wird in die Protokolldatei der ADIF-Konsolenanwendung

geschrieben. Selbst wenn die Daten mit neuen Schlüsseln für jedes Objekt zur Tabelle hinzugefügt werden können, führen Daten, die vorhandene Objektdaten aktualisieren sollen, zur Erzeugung eines neuen Datensatzes, anstatt den vorhandenen zu aktualisieren.

Der ADIF- Exporteintrag für den Export in externe Datenbanken ermöglicht es Ihnen, eine beliebige Anzahl von DDL- und DML-Anweisungen zu definieren, die entweder vor oder nach der Ausführung des Datenexports mit der Zieldatenbank durchgeführt werden sollen.

Um die Datenaktualisierung in vorhandenen Datenbanktabellen zu implementieren, können Sie beispielsweise vor dem Datenexport eine neue Datenbanktabelle in der Zieldatenbank erzeugen und die Daten in die neue Tabelle exportieren. Nach dem Datenexport können SQL-Befehle ausgeführt werden, die die tatsächlichen Zieltabellen in der externen Datenbank aktualisieren und dabei die erforderlichen Schlüsseleinschränkungen und anderer Aktualisierungsanforderungen berücksichtigen.

Die neue Datenbanktabelle kann dann gelöscht werden, nachdem die Zieltabellen geändert wurden. Zum Löschen der temporären Tabelle sind zwei Mechanismen verfügbar:

- Um die Tabelle als letzten Schritt der Ausführung des Exporteintrags und vor der Ausführung des nächsten Exporteintrags zu löschen, definieren Sie eine DDL-Anweisung, die die Tabelle im Ordner **SQL-Befehle – OnComplete** des Exporteintrags entfernt.
- Um die Tabelle zu löschen, nachdem alle SQL-Befehle in allen Exporteinträgen des Datensatzes ausgeführt wurden, setzen Sie das Attribut **Tabelle nach Export löschen** für den Exporteintrag auf `True`. Mit dieser Konfiguration können Sie Daten aus der Tabelle in SQL-Befehlen verwenden, die in anderen Exporteinträgen konfiguriert sind, die während des Datenexports in die externe Datenbank nach dem aktuellen Exporteintrag ausgeführt werden.



Bei regelmäßigen Exporten ist es möglicherweise einfacher, entweder während des ersten Imports oder durch Mechanismen außerhalb von ADIF in der Zieldatenbank eine Tabelle für den Datenexport zu erstellen und die Tabelle dauerhaft zu behalten. Sie können dann das Attribut **Tabelle vor dem Exportieren leeren** im Exporteintrag so festsetzen, dass vor jedem neuen Import von Daten in die externe Datenbank die Daten aus der Datenbanktabelle gelöscht werden.

Wenn Sie Daten primär in eine temporäre Tabelle exportieren, können Sie nur einen Exporteintrag verwenden, um mehrere Datenbanktabellen der externen Datenbank zu aktualisieren. Die Anzahl der Tabellen, die durch SQL-Befehle im Ordner **SQL-Befehle – OnComplete** geändert werden, ist unbegrenzt.

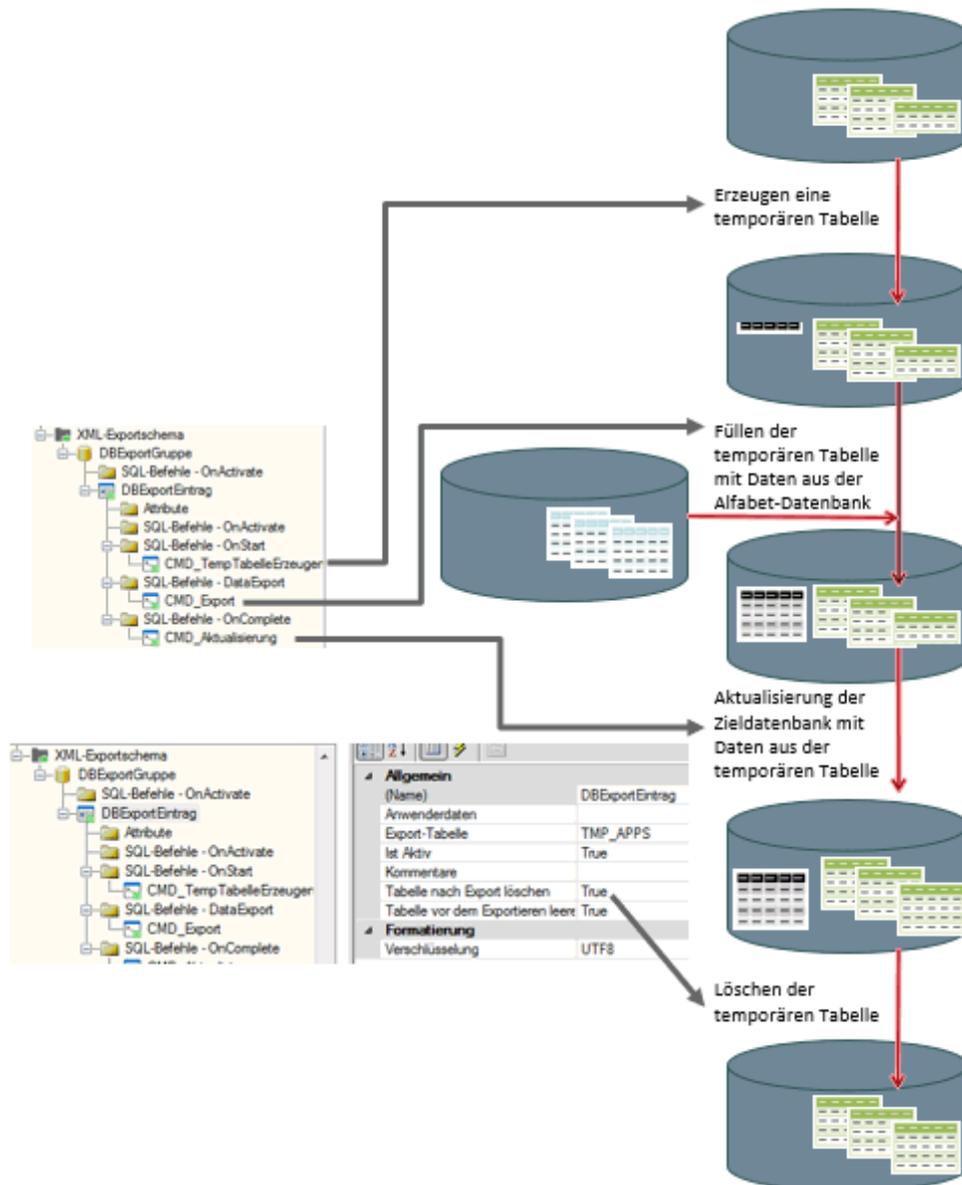


Abbildung: Übersicht über die Konfigurationsschritte, die zum Aktualisieren von Daten in einer externen Datenbank durchgeführt werden

Anwendungsfall: Definieren exportbezogener benutzerdefinierter Eigenschaften

Es kann aus folgenden Gründen nützlich sein, die Informationen zur Leistung des Exports direkt in der Datenbanktabelle der exportierten Objekte zu speichern, beispielsweise:

- Konfigurieren des Export, sodass nur Objekte enthalten sind, die nicht als bereits in einem vorangegangenen Datenexport exportiert markiert sind.
- Informieren der Anwender über die Alfabeta -Benutzeroberfläche, ob und/oder wann zuletzt Daten exportiert wurden.
- Nutzung des Attributs als Bedingung für die Ausführung von Aktionen über die Alfabeta - Benutzeroberfläche (beispielsweise in Vorbedingungen für Workflows oder in Berichten, um einen Bericht zu erzeugen, der nur diejenigen Daten zeigt, die bereits exportiert wurden).



Die folgenden Konfigurationsschritte sind erforderlich, um Informationen über die Exportleistung direkt in der Datenbanktabelle der exportierten Objekte zu speichern:

- Der Objektklasse, für die Daten exportiert werden, muss eine benutzerdefinierte Eigenschaft hinzugefügt werden. Die Anpassung von Objektklassen über benutzerdefinierte Eigenschaften ist Teil der Konfigurationsfähigkeiten, die Alfabet Expand unabhängig von der ADIF-Benutzeroberfläche bietet. Informationen über das Erzeugen einer benutzerdefinierten Eigenschaft finden Sie unter *Konfigurieren benutzerdefinierter Eigenschaften für geschützte oder öffentliche Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- Im Exporteintrag, der den Export auslöst, können Sie dann dem Ordner **SQL-Befehle - OnStart** oder **SQL-Befehle - OnComplete** einen SQL-Befehl mit einer `UPDATE`-Anweisung hinzufügen, der die exportierten Objekte findet und Informationen zum Export in das benutzerdefinierte Attribut schreibt. Das benutzerdefinierte Attribut kann das Datum enthalten, an dem der Export durchgeführt wurde, oder einen booleschen Wert, der zwischen exportierten und nicht exportierten Daten unterscheidet.



Nutzen Sie stets die Konfigurationsfunktionen von Alfabet Expand, um einer Datenbanktabelle einer Alfabet -Objektklasse neue Eigenschaftsspalten hinzuzufügen. Eigenschaftsspalten dürfen nicht mit einer DDL-Anweisung in den SQL-Befehlen des ADIF-Schemas hinzugefügt werden. Bestehende Alfabet-Datenbank -Standardtabellen dürfen nicht durch SQL-Befehle über ADIF geändert werden.

Anwendungsfall: Datenrestrukturierung vor dem Export mit temporären Tabellen

Bei den meisten Exporten (außer dem Export in XML) muss der Inhalt des Exports mit einer einzelnen SQL-Abfrage mit einer `SELECT`-Anweisung definiert werden. SQL bietet bei der Datensammlung in einer einzelnen `SELECT`-Anweisung ein hohes Maß an Flexibilität. Darüber hinaus bietet ADIF zusätzliche Funktionen für Anwendungsfälle, in denen Daten vor dem Export auf eine Art manipuliert oder strukturiert werden müssen, die durch das Definieren von SQL-Abfragen nicht abgedeckt werden kann.

Neue Datenbanktabellen können der Alfabet-Datenbank vor dem Export hinzugefügt werden, um die bestehenden Daten zu restrukturieren oder um neue Daten hinzuzufügen, die aus den Abfrageergebnissen abgeleitet wurden. Die SQL-Abfragen im Ordner **SQL-Befehle - DataExport** können dann definiert werden, um Daten ausschließlich aus den neuen Datenbanktabellen oder parallel zu den Daten aus den üblichen Alfabet-Datenbank -Tabellen zu exportieren.



Neue Datenbanktabellen sollten nur temporär zur Alfabet-Datenbank hinzugefügt werden. Wenn Sie eine Tabelle über einen SQL-Befehl hinzufügen, der beim Start des Exporteintrags ausgeführt werden soll, stellen Sie sicher, dass Sie einen SQL-Befehl hinzufügen, der beim Abschluss des Exports ausgeführt wird und die neue Datenbanktabelle verwirft.



Wenn die Ausgabe des Exports eine XML-Datei ist, können die Daten aus der temporären Tabelle nur dann in die Ausgabe aufgenommen werden, wenn die erste Spalte der `SELECT`-Anweisung der Exportdefinition den Alias-Namen „REFSTR“ hat. Diese erste Eigenschaft in der `SELECT`-Anweisung wird ignoriert.

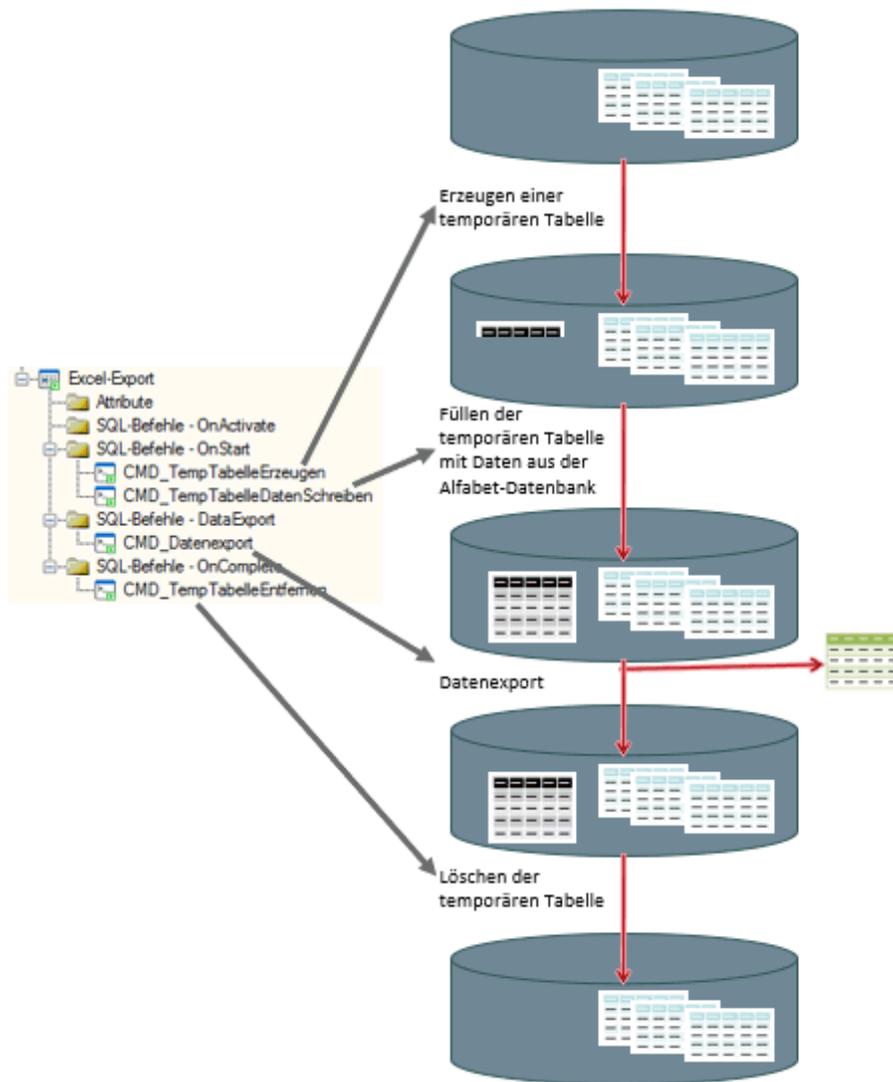


Abbildung: Erstellung von temporären Datenbanktabellen während des Exports über SQL-Befehle für einen Exporteintrag

Konfigurieren der Ausführung in Abhängigkeit von aktuellen Parametern

ADIF-Exportschemata können dafür konfiguriert werden, sich an die aktuellen Status von Objekten in der Alfabet-Datenbank anzupassen. Sie können Bedingungen für die Gesamtausführung von Teilen des Exportschemas definieren oder Bedingungen für die Auswahl von Objektdaten für die Ausführung, die während des Exports in der Befehlszeile der ADIF-Konsolenapplikation festgelegt werden und sich daher von Export zu Export unterscheiden können.

Konfigurieren der bedingten Ausführung von Teilen des Exportschemas

Mit dem ADIF-Exportschema können Sie die Ausführung von Teilen der Exportdefinition an Bedingungen knüpfen. Sie können SQL-Befehle definieren, die die Alfabet-Datenbank daraufhin prüfen, ob Daten

verfügbar sind, und Exportgruppen oder Exporteinträge so konfigurieren, dass sie nur ausgeführt werden, wenn eine konfigurierte SQL-Abfrage, die auf der Alfabet-Datenbank ausgeführt wird, entweder ein positives oder ein negatives Ergebnis zurückgibt.



Beispiel: Es sollen Daten über alle Anwendungen exportiert werden, für die ein bestimmter Indikator kritische Werte anzeigt. Der Export kann so konfiguriert werden, dass er nur ausgeführt wird, wenn für mindestens eine Anwendung ein kritischer Wert des Indikators erreicht wird.

Bedingungen können für die folgenden Elemente im ADIF-Exportschema definiert werden:

- DB-Exportgruppe
- Excel-Exportgruppe
- XML-Exportgruppe
- Exporteintrag

Wird eine Bedingung für ein Element angegeben, werden dieses Element und alle untergeordneten Elemente des aktuellen Elements nur ausgeführt, wenn die Bedingung erfüllt ist.

So geben Sie eine Bedingung für die Ausführung eines Teils eines ADIF-Exportschemas an:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf den Ordner **SQL-Befehle - OnActivate**, der dem Explorer automatisch hinzugefügt wird, wenn ein Element erzeugt wird.
- 2) Wählen Sie im Kontextmenü **SQL-Befehl erzeugen** aus. Dem Ordner **SQL-Befehle - OnActivate** wird ein neues SQL-Befehlselement als untergeordneter Knoten hinzugefügt.
- 3) Klicken Sie auf den neuen SQL-Befehlsknoten, und geben Sie im Attributfenster Folgendes an:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität bietet, die mit dem SQL-Befehl implementiert wird.
 - **Anwenden auf:** Wählen Sie `local` aus. SQL-Befehle vom Typ **OnActivate** können nur auf der Alfabet-Datenbank ausgeführt werden.
 - **Ergebnistyp:** Wählen Sie eine der folgenden Optionen in der Dropdownliste aus.
 - `PositiveCheck`: Wenn der Datensatz leer ist, der aus der Ausführung der mit dem Attribut **Text** definierten Abfrage resultiert, wird das aktuelle Element des ADIF-Schemas nicht ausgeführt, und die im Attribut **Meldung** definierte Meldung wird in die Protokolldatei geschrieben.
 - `NegativeCheck`: Wenn der Datensatz nicht leer ist, der aus der Ausführung der mit dem Attribut **Text** definierten Abfrage resultiert, wird das aktuelle Element des ADIF-Schemas nicht ausgeführt, und die im Attribut **Meldung** definierte Meldung wird in die Protokolldatei geschrieben.
 - **Text:** Definieren Sie eine SQL-Abfrage mit einer `SELECT`-Anweisung, die einen Datensatz zurückgibt. Der resultierende Datensatz wird entsprechend den Einstellungen des Attributs **Ergebnistyp** geprüft, um festzulegen, ob das ADIF-Element ausgeführt wird. Schreiben Sie die SQL-Abfrage entweder direkt in das Attributfeld, oder klicken Sie auf die Schaltfläche **Durchsuchen**  um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.

- **Meldung:** Geben Sie einen Text ein, der in die Protokolldatei geschrieben wird, falls das ADIF-Element nicht ausgeführt wird.
- **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Export auch dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Import im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
- **Ist aktiv:** Wählen Sie `True` aus, um die Überprüfung zu aktivieren. Wählen Sie `False` aus, um die Überprüfung zu deaktivieren.

Die Bedingung für die Prüfung kann abhängig von Parametern, die während der Laufzeit des ADIF-Imports festgelegt wurden, definiert werden. Informationen hierzu finden Sie unter [Konfigurieren des Imports in Abhängigkeit von Parametern](#).

Konfigurieren des Exports in Abhängigkeit von Parametern

Der Export kann so konfiguriert werden, dass er auf Parametern basiert, die in der Befehlszeile der ADIF-Konsolenapplikation oder in den JSON-Nutzdaten des RESTful-Serviceaufrufs definiert wurden, wenn ein Export gestartet wird. Dadurch können Sie ein ADIF-Exportschema für verschiedene ähnliche Exporte verwenden, die sich nur in kleineren Aspekten voneinander unterscheiden.



Es ist nicht möglich, Parameter für ADIF-Exporte zu definieren, die über die Benutzeroberfläche ausgeführt werden. Trotzdem können diese Exporte mit den Standardwerten für die Parameter ausgeführt werden, wenn diese Standardwerte im ADIF-Exportschema definiert sind. Wenn die Konfiguration eines ADIF-Schemas obligatorische Parameter enthält, wird die Ausführungsoption in der Funktionalität **ADIF-Prozessverwaltung** (`ADMIN_AdifJobs`) deaktiviert.



Ein Export beispielsweise, der darauf konfiguriert ist, eine externe Datenbank mit Informationen zu aktualisieren, die in der Alfabet-Datenbank geändert wurden, kann mit den neuesten Änderungsdaten von Objekten als Parameter durchgeführt werden. Sie können die ADIF-Benutzeroberfläche dahingehend konfigurieren, dass Daten nur dann exportiert werden, wenn das Objekt nach einem bestimmten Datum geändert wurde, das durch einen Parameter definiert ist. Bei jedem Export wird der Parameter in der Befehlszeile auf das Datum festgelegt, an dem der letzte Export durchgeführt wurde.

Folgende Informationen sind verfügbar:

- [Konfigurieren eines ADIF-Exportschemas zur Verwendung von Parametern in Exportdefinitionen](#)
- [Definieren von Parameterwerten bei der Ausführung von ADIF-Schemata](#)
- [Definieren von Parameterwerten während des Tests des ADIF-Schemas](#)

Konfigurieren eines ADIF-Exportschemas zur Verwendung von Parametern in Exportdefinitionen

Parameter müssen im ADIF-Schema definiert werden. Nur Parametern, die im ADIF-Schema definiert sind, können während der Ausführung des ADIF-Schemas Werte zugewiesen werden.

So definieren Sie die Parameter für ein ADIF-Schema:

- 1) Klicken Sie im Explorer auf das ADIF-Exportschema, das Sie mit Parametern starten möchten.

2) Definieren Sie im Attributfenster die für Parameter relevanten Attribute:

- **Parameter – Abwärtskompatibilitätsmodus:** Stellen Sie sicher, dass der Parameter auf `False` eingestellt ist. Für neue ADIF-Schemata ist dies der Standardwert.



Für ADIF-Schemata, die vor Alfabet 10.4 definiert wurden, ist der Kompatibilitätsmodus auf `True` eingestellt. Wenn Sie den Kompatibilitätsmodus für diese Schemata zurücksetzen, müssen Sie Parameterdefinitionen zu den ADIF-Exporteinträgen im Schema hinzufügen und die Art und Weise, wie Parameter in den SQL-Befehlen definiert werden, überprüfen.

Bitte beachten Sie: Für ADIF-Schemata, bei denen **Parameter – Abwärtskompatibilitätsmodus** auf `True` eingestellt ist und die Parameterdefinitionen enthalten, kann die Ausführung nicht über die Funktionalität **Prozesszeitplan** geplant werden.

- **Argument-Tabellenname:** Zu Beginn eines ADIF-Exports werden die Parameterwerte in eine temporäre Datenbanktabelle geschrieben. Der Name der temporären Datenbanktabelle kann im ADIF-Exportschema konfiguriert werden. Standardmäßig wird sie `ADIF_ARGS` genannt.



Für jeden Parameter wird in der Datenbanktabelle eine Zeile mit zwei Spalten angezeigt:

- `ARG_NAME` zum Speichern des Parameternamens.
- `ARG_VALUE` zum Speichern des Parameterwerts.

3) Klicken Sie für jeden Parameter, den Sie in einem der Exporteinträge im Schema verwenden möchten, mit der rechten Maustaste auf den untergeordneten Knoten **Parameter** des ADIF-Exportschemas, und wählen Sie **Parameter erzeugen** aus der Dropdown-Liste aus.

4) Klicken Sie auf den neu hinzugefügten Knoten **@Parameter**, und definieren Sie im Attributfenster folgende Attribute:

- **Name:** Definieren Sie einen eindeutigen Namen für den Parameter. Der Parametername muss mit "@" beginnen und darf keine anderen Sonderzeichen oder Leerzeichen enthalten. Er kann in den SQL-Befehlen und Attributen des ADIF-Exportschemas verwendet werden und wird während der Ausführung des ADIF-Exportschemas durch den aktuellen Wert des Parameters ersetzt.
- **Parametertyp:** Wählen Sie den Datentyp des Parameterwerts aus der Dropdown-Liste aus.



Die Parametertypen "StringArray" und "ReferenceArray" können verwendet werden, wenn eine Abfrage definieren soll, dass eine Zeichenfolge oder der REFSTR eines referenzierten Objekts innerhalb eines zulässigen Bereichs von Zeichenfolgen- oder REFSTR-Werten liegen soll.

- **Standardwert:** Definieren Sie optional einen Standardwert, der verwendet werden soll, wenn während der Ausführung des ADIF-Schemas kein Parameterwert angegeben wird. Klicken Sie auf die Schaltfläche **Durchsuchen**  rechts neben dem Attributfeld, um den Editor zum Definieren des Standardwerts zu öffnen. Beachten Sie Folgendes bezüglich der Spezifikation von Werten:

- Geben Sie für die Definition von Datumsangaben das Datum in das Feld **Wert** in dem Format ein, das in der Landeseinstellung der Sprache definiert ist, die Sie derzeit zum Anzeigen der Alfabet Expand -Benutzeroberfläche verwenden, oder klicken Sie auf die Schaltfläche **Auswählen**, und wählen Sie ein Datum aus dem Kalender aus.

- Wenn der Parameter ein Platzhalter für eine Zeichenfolge ist und in einer Abfrage in einer `WHERE`-Bedingung verwendet wird, die überprüft, ob ein Wert sich in einem Bereich befindet, muss der **Parametertyp** `StringArray` sein. Das Feld **Wert** im Editor des Attributs **Standardwert** zeigt für `StringArray`-Parameter ein Textfeld an. Definieren Sie den Bereich der zulässigen Werte im Textfeld, wobei jeder Wert in einer separaten Zeile stehen muss. In der Abfrage, die auf das Array verweist, muss der Parameter in Klammern eingefügt werden:


```
WHERE app.OBJECTSTATE IN (@appState)
```
- Wenn ein Zeichenfolgenwert ein einzelnes Anführungszeichen enthält (z. B. in „O'Hara“), wird das einfache Anführungszeichen automatisch durch ein zweites einfaches Anführungszeichen geschützt, wenn der Parameterwert verarbeitet wird. Das Schützen eines einzelnen Anführungszeichens mit einem zweiten einfachen Anführungszeichen ist eine Voraussetzung für die Integration in SQL-Abfragen. Aus Sicherheitsgründen werden zweite einfache Anführungszeichen zusätzlich in Zeichenfolgen eingefügt, um SQL-Injection zu verhindern.
- "%" kann als Platzhalter in Zeichenfolgen und Texten verwendet werden. Es ist nicht zulässig, einen Platzhalter in einem Wert eines Zeichenfolgenarrays zu definieren.
- Boolesche Werte können aus einer Dropdown-Liste ausgewählt werden. Beachten Sie, dass Objektklasseneigenschaften des Typs `Boolean`, die weder obligatorisch sind noch einen definierten Standardwert haben, entweder auf "True", "False" oder "NULL" gesetzt werden können. Wenn der Standardwert `False` ist, werden nur Eigenschaftswerte der Objektklasse zurückgegeben, die als "False" definiert sind, während NULL-Werte nicht zurückgegeben werden. Dieses Verhalten kann über die SQL-Befehle im ADIF-Schema gehandhabt werden.
- Für `ReferenceArray`-Eigenschaften können keine Standardwerte definiert werden.
- **Obligatorisch:** Wählen Sie `True` aus, wenn die ADIF-Ausführung mit einer Fehlermeldung fehlschlagen soll, wenn für die aktuelle Ausführung kein Wert definiert ist. Wählen Sie "False" aus, wenn die ADIF-Ausführung auch dann ausgeführt werden soll, wenn der Parameter für die aktuelle Ausführung nicht definiert ist. Der Parameter wird dann durch den im Attribut **Standardwert** definierten Wert ersetzt. Wenn kein Standardwert angegeben wird, wird der Parameter durch `NULL` ersetzt.



Bei der ADIF-Ausführung werden `WHERE`-Klauseln in SQL-Befehlen, die einen Parameter enthalten, nicht aus der Abfrage entfernt, wenn kein Wert angegeben wird. Um sicherzustellen, dass das Abfrageergebnis sinnvoll ist, wenn der Parameterwert während der ADIF-Ausführung nicht angegeben und der Parameter daher auf `NULL` gesetzt wird, sollten Sie die Abfrage so definieren, dass `NULL`-Werte akzeptiert werden. Um beispielsweise sicherzustellen, dass Ergebnisse zurückgegeben werden, wenn ein Vergleich eines Datums mit dem Startdatum einer Applikation ohne Ergebnis bleibt, sollte die `WHERE`-Klausel folgendermaßen verfasst werden:

```
WHERE APPLICATION.STARTDATE >= @StartDate OR @StartDate IS
NULL
```

- 5) Verwenden Sie je nach Bedarf die Parameter, die im Ordner **Parameter** in den ADIF-Exporteinträgen des ADIF-Exportschemas in den folgenden Einstellungen definiert sind:
- In den SQL-Abfragen, die für einen beliebigen SQL-Befehl definiert sind.
 - Im Attribut **Meldung** von SQL-Befehlen vom **Ergebnistyp** `DebugMessage`.
 - Im Attribut **Verbindungsstring** eines Datenbankexportsatzes.

- Im Attribut **Überschriftentext** von Exporteinträgen für den Export in Microsoft® Excel®-Dateien.

Um einen Parameter einzuschließen, ersetzen Sie den tatsächlichen Wert in der Abfrage oder der Textzeichenfolge durch den Parameternamen. Beachten Sie, dass Parameter ohne einfache Anführungszeichen um den Parameternamen eingeschlossen werden sollen. Wenn der Parametertyp die Definition von einfachen Anführungszeichen um den Wert erfordert, wird dies von der Software während der Ersetzung des Parameternamens durch den Parameterwert gehandhabt.



Eine Abfrage, die beispielsweise den Zeichenfolgenparameter @orga verwendet, kann folgendermaßen geschrieben werden:

```
SELECT app.NAME, app.VERSION
FROM APPLICATION app, ORGAUNIT org
WHERE app.RESPONSIBLEORGANIZATION = org.REFSTR
AND org.NAME LIKE @orga;
```

Wenn er während der Laufzeit den Wert Trade% annimmt, lautet die Abfrage:

```
SELECT app.NAME, app.VERSION
FROM APPLICATION app, ORGAUNIT org
WHERE app.RESPONSIBLEORGANIZATION = org.REFSTR
AND org.NAME LIKE 'Trade%';
```

Eine Debugmeldung, die im Attribut **Meldung** eines SQL-Befehls mit dem **Ergebnistyp** DebugMessage definiert ist und die während des Exports der Applikationsdaten in die Exportprotokolldatei geschrieben werden soll, kann auch auf die aktuelle Organisation verweisen:

```
Export of applications with responsible organization @orga.
```

Während der Laufzeit wird sie konvertiert in:

```
Export of applications with responsible organization Trade%.
```

Die Parameter "StringArray" oder "ReferenceArray" müssen wie folgt in die Abfrage eingeschlossen werden:

```
WHERE APPLICATION.STATUS IN (@StringArray)
```

Definieren von Parameterwerten bei der Ausführung von ADIF-Schemata

Parameterwerte werden bei der Ausführung des ADIF-Imports oder -Exports während der Laufzeit gesetzt. Bei Parameternamen muss die Groß-/Kleinschreibung beachtet werden. Wenn Sie Parameter für die Ausführung des ADIF-Imports oder -Exports definieren, müssen Sie den Parameternamen genau so verwenden, wie er im ADIF-Schema angegeben ist, einschließlich des Zeichens „@“ am Anfang. Beachten Sie Folgendes bezüglich der Definition von Werten für die Parameter:

- Wenn ein Zeichenfolgenwert ein einzelnes Anführungszeichen enthält (z. B. in „O'Hara“), wird das einfache Anführungszeichen automatisch durch ein zweites einfaches Anführungszeichen geschützt, wenn der Parameterwert verarbeitet wird. Das Schützen eines einzelnen Anführungszeichens mit einem zweiten einfachen Anführungszeichen ist eine Voraussetzung für

die Integration in SQL-Abfragen. Aus Sicherheitsgründen werden zweite einfache Anführungszeichen zusätzlich in Zeichenfolgen eingefügt, um SQL-Injection zu verhindern.

- Parameterwerte für Datentypen wie Zeichenfolgen oder Datumswerte werden ohne einfache Anführungszeichen am Anfang und am Ende definiert. Wenn in der Abfrage für den Datentyp einfache Anführungszeichen erforderlich sind, werden diese automatisch von den ADIF-Mechanismen hinzugefügt.
- Boolesche Werte müssen definiert werden als 1 (True) oder 0 (False):

```
@MyBooleanValue 1
```

- Werte für "StringArray" und "ReferenceArray" müssen durch "\r\n" getrennt werden:

```
@MyArrayValue First\r\nSecond
```

Ob und wie Befehlszeilenparameter zur Laufzeit definiert werden können, hängt von der Art und Weise ab, wie das ADIF-Schema ausgeführt wird:

- **Starten von ADIF über die Konsolenapplikation**

Um Parameter in der Befehlszeile der ADIF-Konsolenapplikation anzugeben, starten Sie die Konsolenapplikation mit der Befehlszeilenoption `-<Parametername> <Parameterwert>`. Für jeden Parameter muss eine Befehlszeilenoption angegeben werden.



Um die Parameter `@lastupdate` und `@orga` festzulegen, kann die Befehlszeile beispielsweise wie folgt definiert werden:

```
ADIF_Console.exe -export -msalias Alfabet -alfaLoginName
UserName -alfaLoginPassword pw1234 -scheme AppExport -
exportfile C:\Export\Applications.zip -@lastupdate 09/06/2018
-@orga ITGroup
```



Allgemeine Informationen zum Starten von ADIF über die Konsolenapplikation finden Sie unter [Ausführen von ADIF über ein Befehlszeilen-Tool](#).

- **Starten von ADIF über einen RESTful-Serviceaufruf an die RESTful API der Alfabet-Web-Applikation**

Parameter werden in den JSON-Nutzdaten der Anfrage in dem Feld `UserArgs` definiert, das ein JSON-Objekt mit einem Feld für jeden Parameter in folgendem Format zurückgibt: `"UserArgs": {" arg1name ":" arg1value "," arg2name ":" arg2value "}`. Der Feldname muss mit dem Variablennamen übereinstimmen, und der Feldwert definiert den Variablenwert für die aktuelle Ausführung des ADIF-Exportschemas.



Um die Parameter `@lastupdate` und `@orga` festzulegen, können die JSON-Nutzdaten der Anfrage beispielsweise wie folgt definiert werden:

```
{
  "Scheme": "AppExport",
  "UserArgs": {"@lastupdate":"09/06/2018","@orga":"ITGroup"},
  "Verbose": false,
  "Synchron": true
}
```



Wenn die Spezifikation der Parameter im Aufruf nicht korrekt ist, wird der ADIF-Prozess gestartet, die Ausführung kann aber fehlschlagen. Es werden an den RESTful-Serviceaufruf keine Informationen über den Erfolg der ADIF-Prozessausführung zurückgegeben. Der Rückgabewert des RESTful-Serviceaufrufs informiert über den letzten erfolgreichen Ausführungsschritt, also dem Start des ADIF-Prozesses. Der Erfolg der ADIF-Prozessausführung kann dann z. B. über die Funktionalität **ADIF-Prozessverwaltung** überprüft werden.

Weitere Informationen über die Funktionalität "ADIF-Prozessverwaltung" finden Sie unter [Ausführen und Überprüfen von ADIF über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" in der Alfabet -Benutzeroberfläche.](#)

- **Starten von ADIF über die Funktionalität "ADIF-Prozessverwaltung" in der Alfabet-Benutzeroberfläche**

Die Funktionalität "ADIF-Prozessverwaltung" bietet keinen Mechanismus zum Festlegen von Parameterwerten. Wenn ein ADIF-Prozess über die **ADIF-Prozessverwaltung** gestartet werden soll, darf er keine Parameter enthalten. Wenn das ADIF-Schema Parameterdefinitionen enthält, kann es über die Funktionalität **ADIF-Prozessverwaltung** nur mit den im ADIF-Schema für die Parameter definierten Standardwerten gestartet werden. Das Schema darf keine obligatorischen Parameterdefinitionen enthalten.



Allgemeine Informationen zum Starten von ADIF über die Funktionalität **ADIF-Prozessverwaltung** finden Sie unter [Ausführen und Überprüfen von ADIF über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" in der Alfabet -Benutzeroberfläche.](#)

- **Starten von ADIF über ein Ereignis**

Wenn Sie den ADIF-Prozess über ein Ereignis starten, können Parameterwerte während der Ausführung des Ereignisses über eine in der Ereignisvorlage definierte Abfrage übergeben werden. In den Attributen der Ereignisvorlage müssen die folgenden Attribute festgelegt werden:

- **Name der Variablen:** Wenn das ADIF-Schema Parameterdefinitionen enthält, müssen die Namen der Parameter als durch Kommas getrennte Liste in dieses Attribut geschrieben werden. Bei der Spezifikation muss zwischen Groß- und Kleinschreibung unterschieden werden. Die Werte für die Parameter müssen dann mit den Attributen **Werte für Variablen über Abfrage / Werte für Variablen über Abfrage als Text** angegeben werden.
- **Werte für Variablen über Abfrage / Werte für Variablen über Abfrage als Text:** Die Abfrage muss die Werte, die im Attribut **Namen der Variablen** für die Parameter definiert sind, in der gleichen Reihenfolge zurückgeben. Die Abfragen können entweder über Native-SQL-Abfragen oder über Alfabet -Abfragen definiert werden. Verwenden Sie das Attribut **Werte für Variablen über Abfrage**, um eine Alfabet -Abfrage zu definieren. Verwenden Sie das Attribut **Werte für Variablen über Abfrage als Text**, um eine Native-SQL-Abfrage zu definieren. In der Abfrage können Parameter der Alfabet -Abfragesprache verwendet werden. Das über den Parameter `BASE` der Alfabet -Abfragesprache zurückgegebene Basisobjekt hängt davon ab, wie ein Ereignis ausgelöst wurde. Für Ereignisse, die über einen Workflow oder Assistenten ausgelöst wurden, gibt `BASE` den REFSTR des Objekts zurück, an dem der Anwender im Schritt des Assistenten oder Workflows gearbeitet hat, der das Ereignis auslöst. Bei Ereignissen, die über ein REST API-Aufrufereignis ausgelöst werden, entspricht das `BASE`-Objekt dem Objekt für das Ereignis, welches das Ereignis auslöst.

Die Abfrage muss einen Datensatz zurückgeben, dessen Spaltennamen mit den Parameternamen, die mit dem Attribut **Name der Variablen** definiert wurden, identisch sind.



Wenn **Name der Variablen** beispielsweise als @AppName, @AppVersion definiert ist, kann die Abfrage folgendermaßen definiert werden:

```
SELECT REFSTR, NAME AS '@AppName', VERSION AS
 '@AppVersion'
FROM APPLICATION
WHERE REFSTR = @BASE
```



Allgemeine Informationen zum Starten von ADIF über die RESTful-Services von Alfabet finden Sie im Referenzhandbuch *Alfabet RESTful API*.

- **Starten von ADIF über eine Schaltfläche im Filterbereich eines konfigurierten Berichts oder in der Symbolleiste einer Alfabet-Ansicht**

Wenn ein ADIF-Prozess über eine Schaltfläche im Filterbereich eines konfigurierten Berichts oder in der Symbolleiste einer Objektansicht gestartet wird, können die Werte für die Parameterdefinition im konfigurierten Bericht wie folgt festgelegt werden:

- Wenn die Schaltfläche für eine Objektansicht oder einen konfigurierten Bericht definiert ist, der über das Attribut **Auf Klasse anwenden** einer Basisobjektklasse zugeordnet ist, werden die Informationen über das Basisobjekt, für das die Objektansicht oder der konfigurierte Bericht geöffnet wird, als @BASE zurückgegeben. Wenn das ADIF-Schema über eine Parameterdefinition verfügt, deren **Name** auf @BASE und deren **Parametertyp** auf Reference eingestellt ist, wird dieser Parameter bei der Ausführung des ADIF-Schemas durch den REFSTR des aktuellen Objekts ersetzt.
- Wenn die Schaltfläche für einen konfigurierten Bericht mit Filterfeldern definiert ist, werden die in den Filterfeldern festgelegten Werte als Parameterwerte an das ADIF-Schema übergeben, wenn der ADIF-Prozess über die Schaltfläche gestartet wird. Im ADIF-Schema muss der **Name** des Parameters mit dem Attribut **Name** des Filterfelds identisch sein, und der **Parametertyp** muss mit dem vom Filterfeld zurückgegebenen Datentyp identisch sein.

Beachten Sie, dass der Name des Filterfelds mit einem „@“ beginnen muss. Standardfilterfelder, die standardmäßig in Berichten des Typs Query, die auf einer Alfabet -Abfrage basieren, automatisch generiert werden, beginnen mit einem Doppelpunkt anstelle von „@“ und müssen manuell angepasst werden.

Bei Vergleichen zwischen Filterfeldern und Parameternamen wird zwischen Groß- und Kleinschreibung unterschieden.

Wenn die Filter eines konfigurierten Berichts bei der Ausführung des ADIF-Prozesses nicht festgelegt sind, wird kein Wert für den ADIF-Schemaparameter bereitgestellt. Daher sollten ADIF-Schemaparameter, die mit Werten aus Filterfeldern gefüllt werden, die im konfigurierten Bericht nicht obligatorisch sind, im ADIF-Schema nicht als obligatorisch definiert werden. Entweder sollte ein Standardwert für die Parameter definiert werden, oder in den SQL-Befehlen des ADIF-Schemas sollte die Verarbeitung von NULL-Werten berücksichtigt werden.



Allgemeine Informationen zum Starten von ADIF über eine Schaltfläche finden Sie unter [Ausführen von ADIF über eine Schaltfläche in der Alfabet-Benutzeroberfläche](#).

Definieren von Parameterwerten während des Tests des ADIF-Schemas

Wenn Sie ein ADIF-Schema mit dem ADIF-Debugger debuggen, kann keine Befehlszeile definiert werden. Zum Festlegen von Parametern für Tests können Parameterwerte für einen Test in das ADIF-Schema geschrieben werden.

So legen Sie Parameterwerte für Tests im ADIF-Schema fest:

- 1) Klicken Sie im ADIF-Explorer auf das ADIF-Schema, für das Sie Parameterwerte angeben möchten.
- 2) Legen Sie im Attributfenster des ADIF-Schemas die folgenden Attribute fest:

- **Debug-Argumente:** Definieren Sie die Werte für die verwendeten Parameter in einer kommasetrennten Zeichenfolge. Das Format der Zeichenfolge muss so aussehen: <Parametername>=<Wert>. Bitte beachten Sie, dass die Werte nicht in einfache Anführungszeichen geschrieben werden dürfen. Wenn in der Abfrage für den Parameterdatentyp, z. B. für Zeichenfolgen, einfache Anführungszeichen erforderlich sind, werden diese automatisch gesetzt, wenn der Wert in der Abfrage ersetzt wird.

Obligatorische Parameter müssen in den **Debug-Argumenten** definiert werden, um das ADIF-Schema über den ADIF-Debugger zu testen. Wenn der Parameter nicht obligatorisch ist und kein Wert für den Parameter in den **Debug-Argumenten** definiert ist, verwendet der ADIF-Debugger den für den Parameter definierten Standardwert. Wenn kein Standardwert definiert ist, wird der Parameter durch NULL ersetzt.



Um zum Beispiel den Export von Applikationen anzugeben, bei dem die für die Applikation verantwortliche Organisation und das letzte Aktualisierungsdatum für die Applikationsdaten Variablen sind, kann das Attribut **Debug-Argumente** wie folgt angegeben werden:

```
@lastupdate=09/10/2012,@orga=ITGroup
```

Konfigurieren von Protokollierungsparametern

Während des Exports über die ADIF-Konsolenanwendung werden Protokollnachrichten in folgende Speicherorte geschrieben:

- Eine temporäre Datenbanktabelle in der Alfabet-Datenbank.
- Die Protokolldatei der ADIF-Konsolenanwendung. Standardmäßig ist dies die Protokolldatei ADIF_Console.log im Arbeitsverzeichnis der ADIF-Konsolenanwendung.

Protokollierungsinformationen werden im folgenden Format in eine Protokolldatei geschrieben:

```
<date and time> <message type> <message text>
```



Beispiel:

```
2010-11-30T10:43:24.31Z DEBUG_INFO Query returns 314 records
```

Protokollmeldungen werden in der Sprache in die Protokolldatei geschrieben, die in den Landeseinstellungen der Alfabet-Datenbank angegeben ist. Wenn die ADIF-Konsolenanwendung mit einem Serveralias ausgeführt wird, wird die Landeseinstellung verwendet, die in der Konfiguration von Alfabet als Standard angegeben ist. Wird die ADIF-Konsolenanwendung mit einem Remote-Alias ausgeführt, wird die Landeseinstellung der Remotealiaskonfiguration verwendet.



Mehr Informationen über das Konfigurieren von Landeseinstellungen für die Alfabet-Datenbank finden Sie unter *Geben Sie die für Ihr Unternehmen relevanten Landeseinstellungen an.* im Kapitel *Lokalisierung und Mehrsprachenunterstützung für die Alfabet -Benutzeroberfläche* des Referenzhandbuchs *Konfigurieren von Alfabet mit Alfabet Expand.*

Informationen über das Konfigurieren der Landeseinstellung des Remote-Alias finden Sie im Referenzhandbuch *Systemadministration* unter *Konfigurationsattribute für die Alfabet -Komponenten.*

Der Zeitstempel richtet sich nach UTC-Zeit (Coordinated Universal Time) und kann sich daher von der Zeit in Ihrer Zeitzone unterscheiden. Der Zeitstempel wird nach ISO 8601 im kombinierten Format aus Datum und Uhrzeit wie folgt geschrieben: "Jahr-Monat-TagTStunden:Minuten:SekundenZ" geschrieben.

Der Nachrichtentyp kann wie folgt lauten:

- **FEHLER:** Es ist ein Fehler aufgetreten. Die Nachricht beschreibt die Art des Fehlers.
- **WARNUNG:** Es sind Probleme bei der Ausführung des Hilfsprogramms aufgetreten, die nicht so schwerwiegend sind wie ein Fehler. Der Prozess wurde ausgeführt, das Ergebnis sollte aber überprüft werden. Die Nachricht beschreibt das Problem.
- **INFO:** Enthält Informationen zur normalen Ausführung des Hilfsprogramms.
- **DEBUGINFO:** Enthält Informationen über den Importvorgang. Diese Informationen werden auch in der Alfabet-Datenbank in einer temporären Datenbanktabelle gespeichert.



In den Debug-Informationen zur Ausführung von SQL-Befehlen wird die Anzahl der Datensätze zurückgegeben, die bei der Ausführung des SQL-Befehls beteiligt sind. Diese Zahl kann höher ausfallen als die Anzahl der aufgrund von ausgelösten Datenbank-Triggern exportierten Objekte.

Während des Importvorgangs können E-Mails mit den Protokollinformationen in einem definierten Zeitintervall an eine konfigurierbare E-Mail-Adresse gesendet werden.



Die E-Mail ist als Alive-Meldung des Systems gedacht, mit der der Empfänger über die Verfügbarkeit eines aktiven ADIF-Prozesses informiert wird. Der Inhalt der E-Mail ist auf die eine Zeile beschränkt, die in dem Moment in die Protokolldatei geschrieben wird, in dem die E-Mail generiert wird, und ist daher nicht für das Debuggen geeignet.

Konfigurieren der Speicherung und Handhabung von Protokolldateien

Das Speichern von Protokollinformationen und das Senden von E-Mails mit prozessbezogenen Informationen kann entweder im ADIF-Importschema oder in der Befehlszeile konfiguriert werden, wenn die ADIF-Konsolenanwendung gestartet wird. Wenn eine Konfiguration sowohl im ADIF-Importschema als auch in der Befehlszeile vorgenommen werden kann, überschreibt die Spezifikation über die Befehlszeilenoptionen die Konfiguration im ADIF-Importschema.

Die folgende Tabelle bietet eine Übersicht über die Konfigurationsoptionen:

Konfigurierbares Protokollierungsverhalten	Attribut des ADIF-Schemas	Befehlszeilenoption	Beschreibung
Zeitintervall für das Senden von E-Mails mit aktuellen Protokollinformationen	Debug-Heart-Beat	-heartbeat <Zeit in Minuten>	Geben Sie das Zeitintervall beim Senden von E-Mails mit dem aktuellen Inhalt der Protokolldatei der ADIF-Konsolenanwendung während des Imports in Minuten an. Der Debug-Heart-Beat ist standardmäßig auf -1 gesetzt. Das bedeutet, dass keine E-Mails gesendet werden. HINWEIS: Wenn der Importprozess abgeschlossen ist, bevor das Zeitintervall zum ersten Mal verstrichen ist, wird keine E-Mail gesendet.
E-Mail-Adresse des Empfängers der Protokolldaten	Empfänger-E-Mail	-recipient-mail <E-Mail-Adresse>	Geben Sie die E-Mail-Adresse der Person an, die den Inhalt der Protokolldatei der ADIF-Konsolenanwendung per E-Mail im konfigurierten Zeitintervall erhalten soll.
Als Absenderadresse in den E-Mails mit Protokolldaten verwendete E-Mail-Adresse	Absender-E-Mail	-sendermail <E-Mail-Adresse>	Geben Sie die E-Mail-Adresse an, die als Absenderadresse für E-Mails dienen soll, die mithilfe der ADIF-Konsolenanwendung gesendet werden.
Name der Protokolldatei der ADIF-Konsolenanwendung	-	-logfile <Dateiname>	Geben Sie den Namen der Protokolldatei der ADIF-Konsolenanwendung an. Zulässige Dateierweiterungen sind LOG und TXT. Der Standardname lautet ADIF_Console.log.
Speicherort der Protokolldatei in der ADIF-Konsolenanwendung	-	-logpath <Pfad>	Geben Sie den Pfad zur Protokolldatei in der ADIF-Konsolenanwendung an. Standardmäßig wird die Protokolldatei im Arbeitsverzeichnis der ADIF-Konsolenanwendung gespeichert. Der ADIF-Konsolenanwendungsprozess muss über Schreibberechtigungen für das angegebene Verzeichnis verfügen.
Menge der in die Protokolldatei geschriebenen Daten	-	-logverbose	Ist diese Option eingestellt, werden alle Informationen über den Prozess in die Protokolldatei geschrieben. Ist diese Option nicht eingestellt, werden nur Fehlermeldungen und Informationen über den Prozessstart und das Prozessende in die Protokolldatei geschrieben.

Konfigurierbares Protokollierungsverhalten	Attribut des ADIF-Schemas	Be-fehlszeilenoption	Beschreibung
Entfernen alter Protokollmeldungen aus der Protokolldatei	-	<code>-logclear <Anzahl Tage></code>	<p>Wenn der Parameter <code>-nologappend</code> festgelegt ist, wird jedes Mal eine neue Protokolldatei erzeugt, wenn das Hilfsprogramm mit derselben Spezifikation von <code>-logfile</code> und <code>-logpath</code> verwendet wird. Der Name der Protokolldatei erhält eine Erweiterung, die aus einem Zeitstempel besteht, der die aktuelle UTC-Zeit angibt.</p> <p>HINWEIS: Der Prüfvorgang kann zu Leistungseinbußen führen.</p>
Erstellen einer neuen Protokolldatei für jeden gestarteten Prozess	-	<code>-nologappend</code>	<p>Diese Option kann nur verwendet werden, wenn <code>-nologappend</code> nicht festgelegt ist. Wenn <code>-nologappend</code> festgelegt ist, wird die Einstellung <code>-logclear</code> ignoriert.</p> <p>Wenn der Parameter <code>-nologappend</code> nicht definiert ist, werden die Protokollierungsoptionen bei jeder Verwendung der ausführbaren Datei der bereits bestehenden Protokolldatei hinzugefügt.</p> <p>Wenn der Parameter <code>-nologappend</code> nicht definiert ist, werden die Protokollierungsoptionen bei jeder Verwendung der ausführbaren Datei der bereits bestehenden Protokolldatei hinzugefügt.</p> <p>HINWEIS: Zur Beschränkung der Dateigröße können Sie die Option <code>-logclear</code> definieren, um alte Protokollmeldungen zu löschen.</p>

So legen Sie die Parameter im ADIF-Exportschema fest:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche auf das Exportschema, das Sie konfigurieren möchten. Die Attribute des ADIF-Exportschemas werden im Attributfenster auf der rechten Seite angezeigt.
- 2) Legen Sie im Attributfenster die Parameter, die in der Tabelle darüber aufgelistet sind, als anwendbar fest.

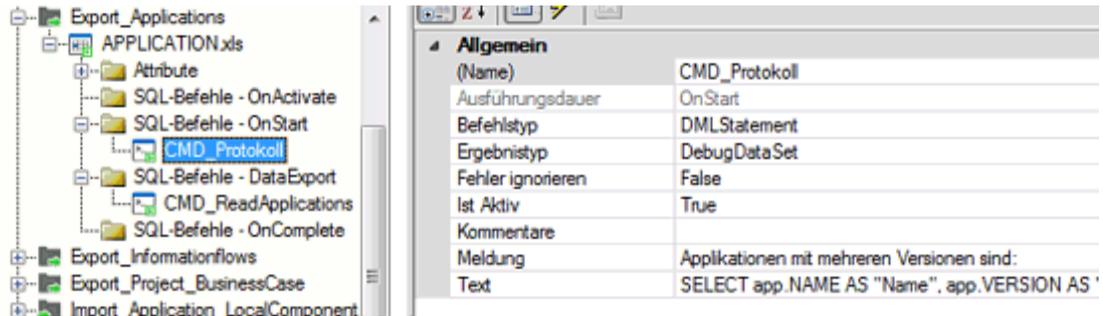
Konfigurieren des Inhalts von Protokollmeldungen

ADIF-Importschemata können so konfiguriert werden, dass sie die benutzerdefinierten Protokollmeldungen und Datensätze enthalten, die während der Ausführung des ADIF-Schemas mithilfe von SQL-Befehlen in den Protokolldateien gefunden wurden. Inhalte von Protokolldateien können dem ADIF-Exportschema mithilfe von SQL-Befehlselementen hinzugefügt werden. SQL-Befehlselemente für Debugmeldungen sind in jedem Ordner **SQL-Befehle – < Aktionstyp >** zu finden, mit Ausnahme des **Ordners SQL-Befehle – DataBind**.

Während der Ausführung des Exports werden die SQL-Befehle im Schema in der Reihenfolge ausgeführt, in der sie durch die Position des Befehls in der XML-Definition angegeben sind. Wenn der SQL-Befehl an der Reihe ist, wird in der Protokollmeldung die Meldung angezeigt, die im SQL-Befehlselement definiert ist und, falls angegeben, das Ergebnis der SQL-Abfrage als Datensatz.



In einem ADIF-Exportschema, mit dem Daten über Anwendungen in eine CSV-Datei exportiert werden, sollen die Informationen über alle Anwendungen, für die in der Datenbank mehr als eine Version existiert, vor der Exportausführung in die Protokolldatei geschrieben werden. Dem Ordner **SQL-Befehle - OnStart** des Exporteintrags wird ein SQL-Befehl vom **Ergebnistyp** `DebugDataSet` hinzugefügt, mit dem der Export der Anwendungsdaten in die CSV-Datei ausgelöst wird:



Im Attributfenster wird das Attribut **Ergebnistyp** auf `DebugDataSet` gesetzt. Der Datensatz, der in der Protokolldatei angezeigt werden soll, wird durch die SQL-Abfrage bestimmt, die im Attribut `Text` definiert ist. Vor dem Datensatz soll in der Protokolldatei ein Text angezeigt werden, der den Datensatz erläutert. Daher wird dem Attribut **Meldung** ein erklärender Text hinzugefügt.

Wenn der Export ausgeführt wird, werden der Text **Meldung** und der aus der SQL-Abfrage resultierende Datensatz der Protokolldatei hinzugefügt, wenn der SQL-Befehl in der Verarbeitungsreihenfolge des ADIF-Schemas an der Reihe ist.

```

2016-09-20T11:51:08.7163369Z      DEBUG_INFO      Das ADIF-Schema Export_Applications wurde initialisiert
2016-09-20T11:51:15.1300165Z      DEBUG_INFO      Exportprozess starten. Datenbank: alfabet_Standard. Tr
2016-09-20T11:51:15.1300165Z      DEBUG_INFO      Datenbank vorbereiten...
2016-09-20T11:51:15.1360177Z      DEBUG_INFO      Prozesseintrag APPLICATION.xls
2016-09-20T11:51:15.1370179Z      DEBUG_INFO      Datensatz mittels CMD_Protokoll finden:
SELECT app.NAME AS "Name", app.VERSION AS "Version"
FROM APPLICATION app
INNER JOIN APPLICATION app1
ON app.NAME=app1.NAME AND app.VERSION != app1.VERSION
WHERE app.NAME LIKE 'A%'
2016-09-20T11:51:15.1410187Z      DEBUG_INFO      Applikationen mit mehreren Versionen sind:
2016-09-20T11:51:15.1420189Z      Name|Version
2016-09-20T11:51:15.1420189Z      ACCOUNT|1.2
2016-09-20T11:51:15.1420189Z      ACCOUNT|1
2016-09-20T11:51:15.1420189Z      AF Enterprise ConTrol|4.0
2016-09-20T11:51:15.1420189Z      AF Enterprise ConTrol|3.1
2016-09-20T11:51:15.1420189Z      AF Good Buy|3.0
2016-09-20T11:51:15.1420189Z      AF Good Buy|2.0
2016-09-20T11:51:15.1420189Z      AF HR Online|3.0
2016-09-20T11:51:15.1420189Z      AF HR Online|2.2
2016-09-20T11:51:15.1420189Z      Anno-Fact|2.0
2016-09-20T11:51:15.1420189Z      Anno-Fact|1.0
2016-09-20T11:51:15.1430191Z      DEBUG_INFO      Bereichsabfrage für Attribut Responsible User durchführ
SELECT TECH_NAME
FROM PERSON
    
```



Debugmeldungen und Datensätze können über Variablen, die von Argumenten in der Befehlszeile der ADIF-Konsolenapplikation beim Starten des Exports abhängen, auf die aktuellen Importbedingungen verweisen. Informationen zur Verwendung von Variablen, um auf die in der

Befehlszeile definierten Bedingungen zu verweisen, finden Sie unter [Konfigurieren der Ausführung des Importschemas in Abhängigkeit von aktuellen Parametern](#).

Einfügen einer statischen Textmeldung

So fügen Sie der Protokolldatei eine statische Textmeldung hinzu:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf den Ordner **SQL-Befehle - < Aktionstyp >**, der dann in der Reihenfolge der Befehle ausgeführt wird, wenn die Textmeldung angezeigt werden soll.
- 2) Wählen Sie im Kontextmenü **SQL-Befehl erzeugen** aus. Dem Ordner **SQL-Befehle - < Aktionstyp >** wird ein neues SQL-Befehlselement als untergeordneter Knoten hinzugefügt.
- 3) Klicken Sie auf den neuen SQL-Befehlsknoten, und geben Sie die folgenden Attribute im Attributfenster an:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität angibt, die mit dem SQL-Befehl implementiert wird.
 - **Ergebnistyp:** Wählen Sie `DebugMessage` aus.
 - **Meldung:** Geben Sie einen Text ein, der an jener Position in die Protokolldatei geschrieben wird, an der dieser SQL-Befehl ausgeführt wird.
 - **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.

Einfügen eines Datensatzes

So beziehen Sie Daten aus der Datenbank in die Protokolldatei ein:

- 1) Klicken Sie im Explorer der ADIF-Konfigurationsbenutzeroberfläche mit der rechten Maustaste auf den Ordner **SQL-Befehle - < Aktionstyp >**, der dann in der Reihenfolge der Befehle ausgeführt wird, wenn die Textmeldung angezeigt werden soll.
- 2) Wählen Sie im Kontextmenü **SQL-Befehl erzeugen** aus. Dem Ordner **SQL-Befehle - < Aktionstyp >** wird ein neues SQL-Befehlselement als untergeordneter Knoten hinzugefügt.
- 3) Klicken Sie auf den neuen SQL-Befehlsknoten, und geben Sie die folgenden Attribute im Attributfenster an:
 - **Name:** Geben Sie einen aussagekräftigen Namen für den SQL-Befehl ein.
 - **Kommentare:** Geben Sie einen Kommentar ein, der Informationen bezüglich der Funktionalität angibt, die mit dem SQL-Befehl implementiert wird.
 - **Ergebnistyp:** Wählen Sie `DebugDataSet` aus.
 - **Meldung:** Geben Sie einen Text ein, der an jener Position in die Protokolldatei geschrieben wird, an der dieser SQL-Befehl ausgeführt wird, und vor dem Datensatz, der aus der definierten SQL-Abfrage resultiert.

- **Text:** Definieren Sie eine SQL-Abfrage mit einer `SELECT`-Anweisung, die einen Datensatz zurückgibt. Der resultierende Datensatz wird an der Position in die Protokolldatei geschrieben, die die Ausführung des SQL-Befehls angibt. Wenn ein Text auch in dem Attribut **Meldung** definiert ist, wird dieser Text vor dem Datensatz in die Protokolldatei geschrieben. Schreiben Sie die SQL-Abfrage entweder direkt in das Attributfeld oder klicken Sie auf die Schaltfläche **Durchsuchen** , um einen Editor für die Definition der SQL-Abfrage zu öffnen. Definieren Sie die SQL-Abfrage auf der Registerkarte **SQL-Text** des Editors.
- **Meldung:** Geben Sie einen Text ein, der als Einführung in den Datensatz in die Protokolldatei geschrieben werden soll.
- **Fehler ignorieren:** Wählen Sie `True` aus, wenn der Export auch dann ausgeführt werden soll, wenn die mit dem Attribut **Text** angegebene SQL-Anweisung zu einer Ausnahme führt. Wählen Sie `False` aus, wenn der Import im Falle eines Fehlers in der SQL-Anweisung nicht ausgeführt werden soll.
- **Ist aktiv:** Wählen Sie `True` aus, um den SQL-Befehl zu aktivieren. Wählen Sie `False` aus, um den SQL-Befehl zu deaktivieren.

Kapitel 6: Debugging einer ADIF-Konfiguration

Die ADIF-Konfigurationsbenutzeroberfläche beinhaltet ein Debugging-Tool. Mithilfe der Debugging-Oberfläche kann der Datenimport oder -export als Test ohne bleibende Auswirkung auf die Alfabet-Datenbank ausgeführt werden, sodass Sie Schritt für Schritt überprüfen können, ob die Befehle ordnungsgemäß ausgeführt werden. Sie können auf das Debugging-Tool über das Kontextmenü der ADIF-Import- oder Export-schemata zugreifen.



Aus Sicherheitsgründen sollten die Konfiguration und das Debugging von ADIF in einer Testumgebung durchgeführt werden, um sicherzustellen, dass die Datenbank in der Produktionsumgebung nicht beschädigt wird, wenn die Konfiguration des ADIF-Schemas oder die Importdaten, die für den Import verwendet wurden, Fehler enthalten.

Aufbau der Debugger-Benutzeroberfläche

Wenn der Debugger für ein ausgewähltes ADIF-Schema gestartet wird, wird ein neues Fenster geöffnet, in dem die Struktur des ausgewählten ADIF-Schemas in einem Explorer angezeigt wird. Rechts neben dem Explorer wird ein Debug-Fenster geöffnet. In diesem Fenster werden Protokollmeldungen und Informationen zu dem Prozessschritt angezeigt, der aktuell ausgeführt wird.



Der ADIF-Debugger zeigt nicht nur die Protokollinformationen in der Benutzeroberfläche an, sondern speichert die Protokollmeldungen auch in einer Protokolldatei im Verzeichnis \$APP-DATA/Temp/Alfabet/<server name>/Runtime/temp.

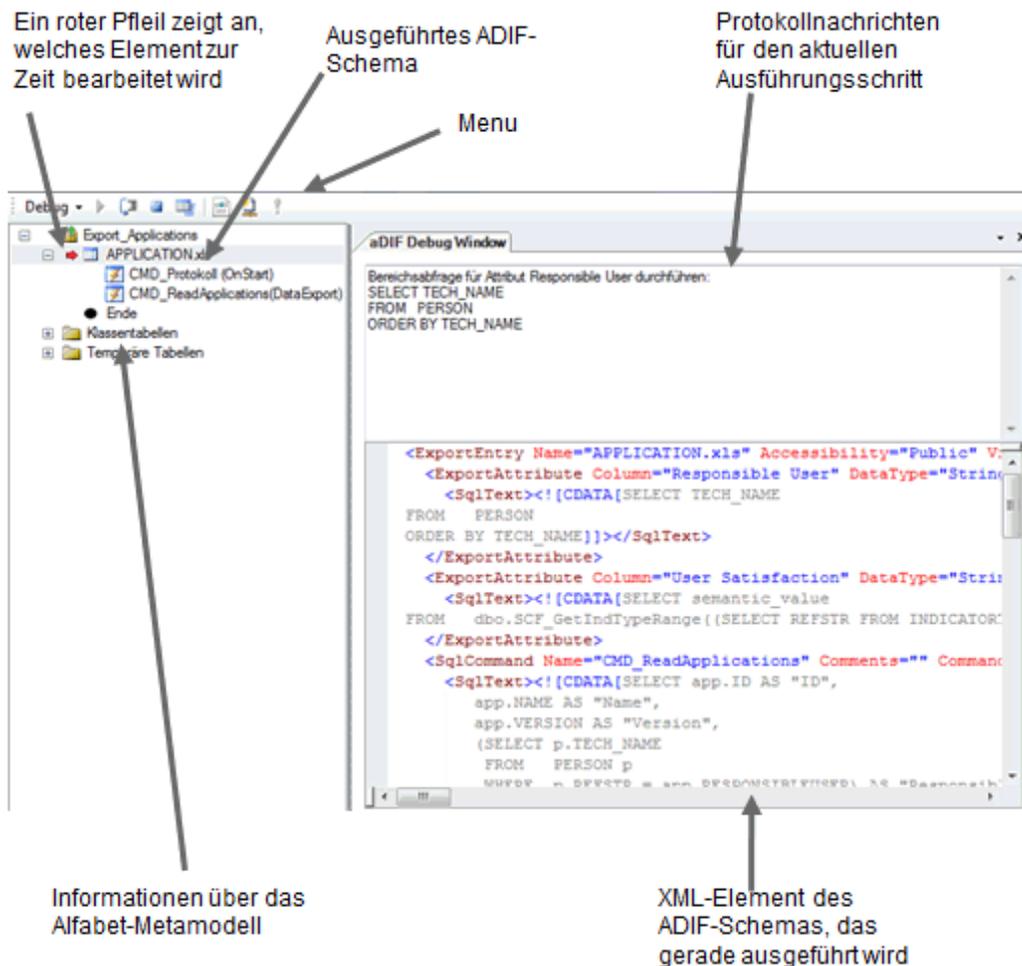


Abbildung: Benutzeroberfläche des ADIF-Debugger

Oben im Explorer ist ein Menü verfügbar, mit dem Sie den Debugging-Prozess steuern können.

Im Explorer wird unter dem ADIF-Schemaknoten ein **Klassentabellen**-Knoten angezeigt. Alle relevanten Datenbanktabellen in der Alfabet-Datenbank sind unter dem **Klassentabellen**-Knoten aufgelistet. Sie können die Tabellen mit SQL-Abfragen durchsuchen oder Informationen über die für die Datenbanktabelle geltende Metamodellstruktur anzeigen. Falls das ADIF-Schema die Definition temporärer Tabellen enthält, werden diese Tabellen im Explorer ebenfalls aufgelistet, in einem eigenen Knoten **Temporäre Tabellen**. Einzelheiten zu den über die Datenbanktabellenknoten abrufbaren Informationen finden Sie in *Durchführen eines Debug-Durchlaufs*.

Ablauf des Debugging-Prozesses



Für das Debugging ist der folgende Workflow erforderlich:

- Wenn Ihre Datenbank in einer XML-Datei konfiguriert ist, laden Sie sie in die Datenbank hoch. (optional)
- Konfigurieren Sie Ihr ADIF-Schema so, dass der Import/Export nach dem Testdurchlauf zurückgenommen wird.

- Starten Sie den Debugger.
- Starten Sie den Debug-Prozess und befolgen Sie den Prozess Schritt für Schritt, während Sie die Ergebnisse über Protokollmeldungen und die daraus resultierenden Änderungen in den Datenbanktabellen kontrollieren.
- Korrigieren Sie bei Bedarf eventuelle Fehler in den Konfigurationen, und starten Sie den Debugging-Prozess neu, bis das ADIF-Schema zur gewünschten Datenmanipulation führt.
- Halten Sie den Debugger an.
- Konfigurieren Sie das ADIF-Schema so, dass Daten dauerhaft in die Datenbank geschrieben werden.
- Exportieren Sie Ihr ADIF-Schema, wenn es über eine XML-Datei verfügbar sein soll. (optional)

Folgende Informationen sind verfügbar:

- [Erforderliche Konfiguration des ADIF-Schemas vor dem Debugging](#)
- [Durchführen eines Debug-Durchlaufs](#)
- [Erforderliche Konfiguration des ADIF-Schemas nach dem Debuggen](#)

Erforderliche Konfiguration des ADIF-Schemas vor dem Debugging



Es können nur ADIF-Schemata debuggt werden, die in die Alfabet-Datenbank hochgeladen sind. Wenn Sie Ihr ADIF-Schema mit einem externen XML-Editor definiert haben, müssen Sie diese über die ADIF-Konfigurationsbenutzeroberfläche in die Alfabet-Datenbank importieren. Informationen über das Erzeugen und Aktualisieren des Indexes finden Sie unter *Externe Editoren im Konfigurieren von ADIF-Schemata*.

Der Debugging-Prozess führt alle in einem ADIF-Schema konfigurierten Schritte aus, um Ihnen zu helfen, die Konfigurationsschritte zu identifizieren, die zu Fehlern oder unerwünschter Datenverarbeitung führen, und diese zu korrigieren. Der Debugging-Prozess ist so etwas wie ein Testlauf und darf daher die bestehenden Daten in der Alfabet-Datenbank oder der externen Datenbank, die als Ziel für den Datenexport vorgesehen ist, nicht verändern. Aus diesem Grund muss das ADIF-Schema so konfiguriert sein, dass für alle beim Testen durchgeführten Schritte ein Rollback durchgeführt wird:

- 1) Klicken Sie im ADIF-Explorer auf das ADIF-Schema, das Sie debuggen möchten.
- 2) Setzen Sie im Attribut Fenster des ADIF-Schemas das Attribut **Änderungen beibehalten** auf "False".



Die folgenden Einschränkungen gelten beim Setzen des Attributs **Änderungen beibehalten** auf "False":

- **Änderungen beibehalten** betrifft nur Datenbanktransaktionen. Wenn Sie Daten in eine Datei exportieren, wird die Exportdatei erstellt und die Daten werden auch dann der Datei hinzugefügt, wenn **Änderungen beibehalten** auf "False" gesetzt ist.

- Wenn das Attribut **Änderungen beibehalten** auf "False" gesetzt wird, werden dadurch alle Änderungen, die durch DML-Anweisungen verursacht wurden (Änderungen an Datensätzen in vorhandenen Tabellen), zurückgenommen. Das Erstellen oder Löschen von Tabellen ist nicht im Rollback enthalten. Das heißt, wenn Sie zum Beispiel ein ADIF-Schema testen, das dafür konfiguriert ist, temporäre Tabellen dauerhaft in die Datenbank zu schreiben, werden diese temporären Tabellen dauerhaft erzeugt, selbst wenn **Änderungen beibehalten** auf "False" gesetzt ist. SQL-Befehle des Typs OnActivate sind ebenfalls vom Rollback ausgeschlossen.
- Wenn während eines Importauftrags neue Objekte erzeugt werden, weist der Datenbindungsmechanismus REFSTR-Werte für die neuen Objekte zu. Wenn **Änderungen beibehalten** auf False gesetzt ist, werden die Objekte nicht in der Datenbank erstellt, die REFSTR-Werte werden jedoch trotzdem als verwendet betrachtet und bei der nächsten ADIF-Ausführung nicht zur Datenbindung verwendet, wenn nicht der Alfabet-Server oder die Alfabet Expand -Applikation, die zur Verarbeitung des ADIF-Auftrags verwendet wird, neu gestartet wird.
- Von OnActivate-Befehlen ausgelöste Änderungen werden nicht zurückgenommen, wenn die Option **Änderungen beibehalten** im Importschema auf False gesetzt ist.

ADIF-Schemata können nicht so konfiguriert werden, dass Variablen in SQL-Abfragen enthalten sind. Die Werte der Variablen werden während der Ausführung des Imports oder Exports in der Befehlszeile der ADIF-Konsolenanwendung bereitgestellt. Es können Standardwerte für die Variablen im ADIF-Schema definiert werden. Während des Import- oder Exportprozesses werden die Werte für die Variablen aus der Befehlszeile oder, wenn sie nicht aus der Befehlszeile verfügbar sind, aus der Konfiguration des ADIF-Schemas gelesen und zur Verwendung während des Imports in eine temporäre Datenbanktabelle geschrieben. Der Name der temporären Datenbanktabelle kann konfiguriert werden.



Informationen zum Definieren von Parametern finden Sie unter [Konfigurieren des Imports in Abhängigkeit von Parametern](#) oder [Konfigurieren des Imports in Abhängigkeit von Parametern](#).

Der Debugger lässt nicht zu, dass während der Debug-Ausführung Variablenwerte gesetzt werden. Deshalb müssen die Variablen im ADIF-Schema definiert werden:

- 1) Klicken Sie im ADIF-Explorer auf das ADIF-Schema, das Sie debuggen möchten.
- 2) Legen Sie im Attributfenster des ADIF-Schemas die folgenden Attribute fest:
 - **Debug-Argumente:** Definieren Sie die Werte für die verwendeten Variablen in einer kommasetrennten Zeichenfolge. Das Format der Zeichenfolge muss so aussehen: <Variablenname>=<Wert>.



Beachten Sie Folgendes beim Angeben von Variablen:

- Obligatorische Parameter müssen in den **Debug-Argumenten** definiert werden, um das ADIF-Schema über den ADIF-Debugger zu testen. Wenn der Parameter nicht obligatorisch ist und kein Wert für den Parameter in den **Debug-Argumenten** definiert ist, verwendet der ADIF-Debugger den für den Parameter definierten Standardwert. Wenn kein Standardwert definiert ist, wird der Parameter durch NULL ersetzt.
- Die Werte dürfen nicht in einfache Anführungszeichen geschrieben werden. Wenn in der Abfrage für den Parameterdatentyp, z. B. für Zeichenfolgen,

einfache Anführungszeichen erforderlich sind, werden diese automatisch gesetzt, wenn der Wert in der Abfrage ersetzt wird.

- Bei der Angabe der Variablen- und Wertnamen wird die Groß-/Kleinschreibung beachtet.
- **Argument-Tabellenname:** Geben Sie optional einen Namen für die temporäre Datenbanktabelle an, in der die Variablennamen und -werte während des Im-/Exportvorgangs gespeichert werden. Der Standardname der Tabelle lautet ADIF_ARGS.

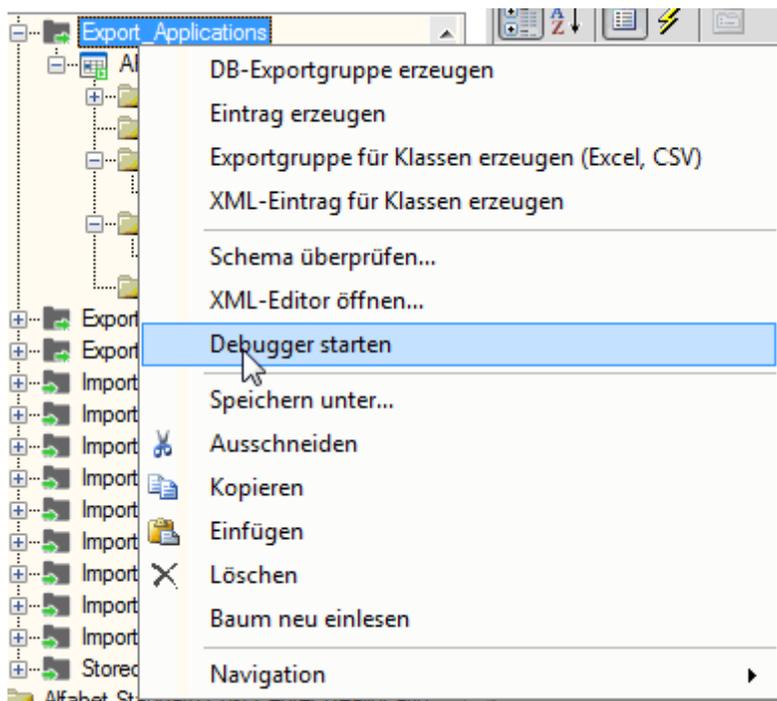
Durchführen eines Debug-Durchlaufs

Das Debugging wird im Debugger-Fenster der ADIF-Konfigurationsbenutzeroberfläche durchgeführt.



Ein Debug-Durchlauf ist wie ein Testexport oder -import von Daten. Wenn Sie den Import von Daten testen möchten, müssen Sie Daten bereitstellen, die in einer ZIP-Datei in das lokale Dateisystem importiert werden, oder Sie müssen den Zugriff auf die Datenbank ermöglichen, aus der Sie Daten importieren möchten.

- 1) Klicken Sie im ADIF-Explorer mit der rechten Maustaste auf das ADIF-Schema, das Sie debuggen möchten, und wählen Sie **Debugger starten**.

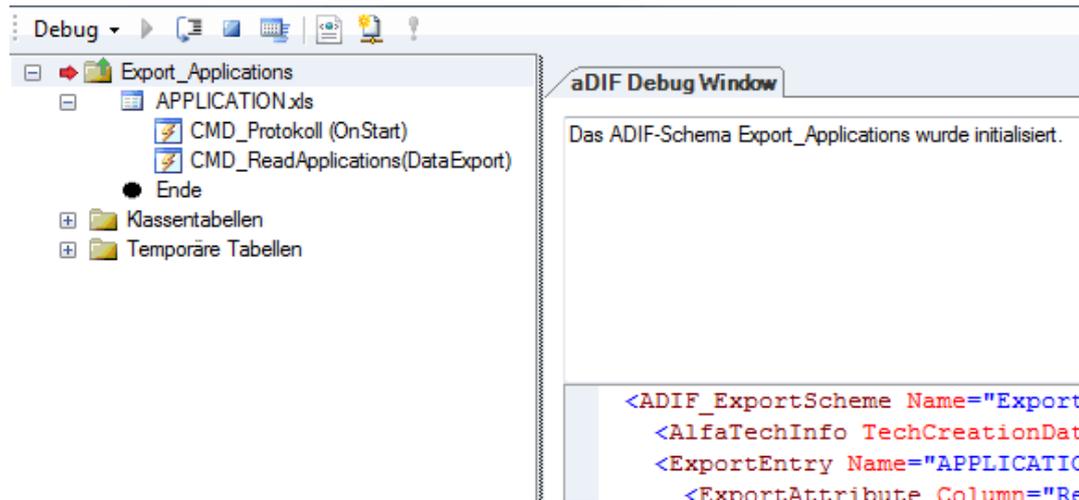


Der Debugger wird geöffnet, und das ausgewählte ADIF-Schema wird im Explorer angezeigt.

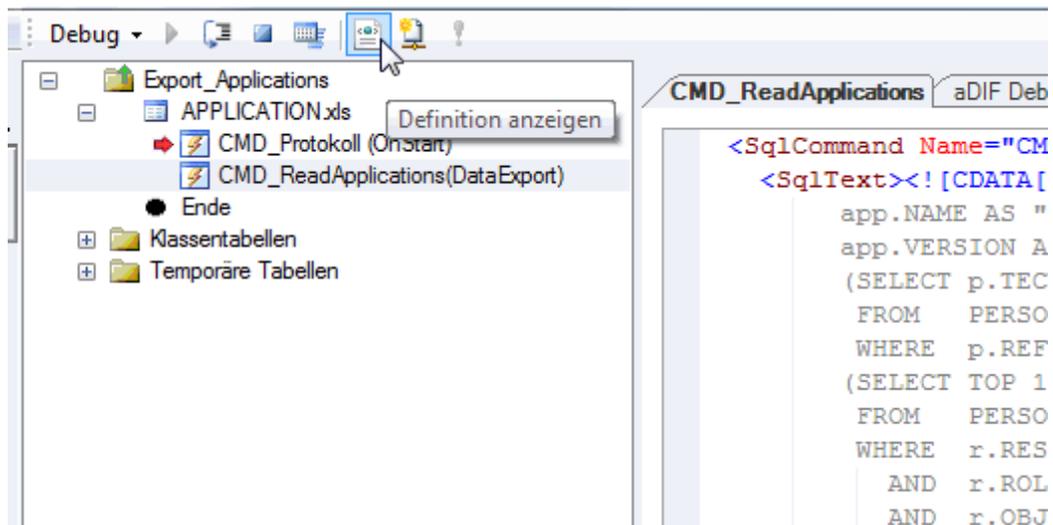
- 2) Klicken Sie im Menü auf die Schaltfläche **Start**
- 3) Wenn der Import aus Dateien oder der Export in Dateien im ADIF-Schema konfiguriert wird, wird ein Explorer-Fenster geöffnet.

- Geben Sie für Exportschemata den Namen und Speicherort der ZIP-Datei ein, in die Daten exportiert werden sollen. Wenn die Datei nicht existiert, wird sie während des Prozesses erzeugt. Wenn sie vorhanden ist, wird sie überschrieben.
- Geben Sie für Importschemata den Namen und Speicherort der ZIP-Datei ein, die die Dateien enthält, in die Daten importiert werden sollen.

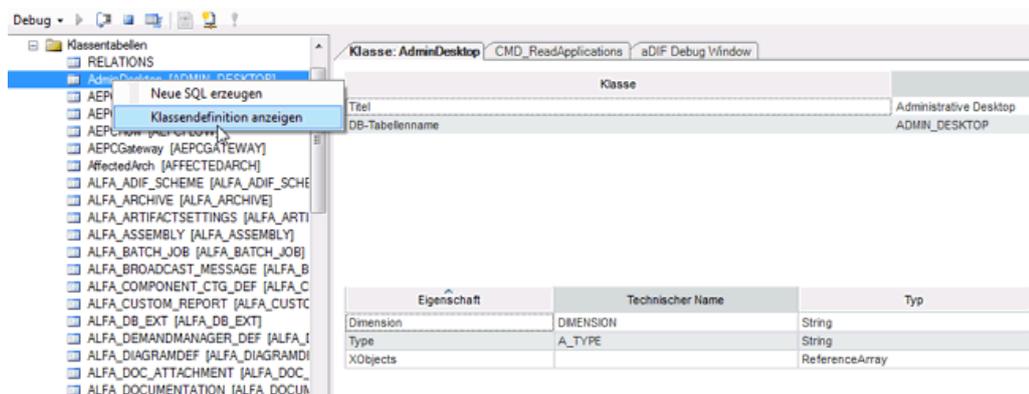
Im Explorer wird vor dem ADIF-Schema ein roter Pfeil angezeigt, der den Startpunkt für den Import-/Exportprozess zeigt. Ein Debug-Fenster wird geöffnet, das darüber informiert, dass das ADIF-Schema initialisiert wurde, und den XML Strichcode des aktuell verarbeiteten Elements des ADIF-Schemas anzeigt.



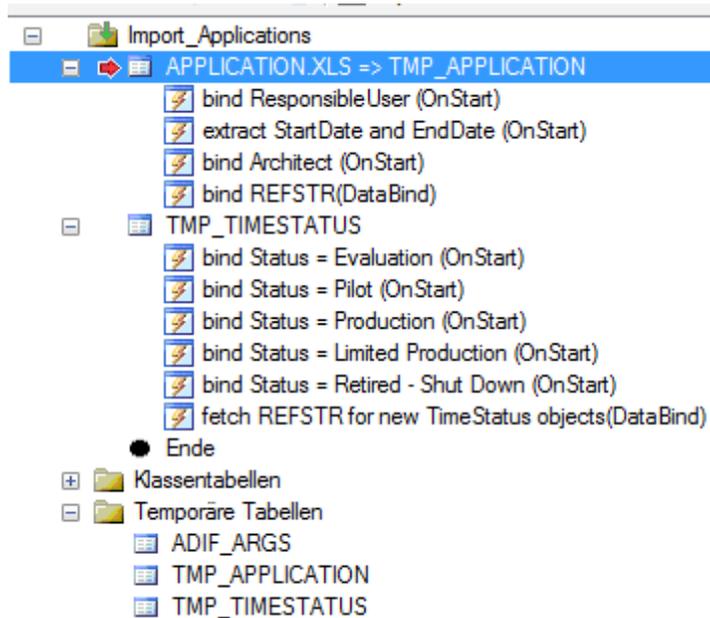
- 4) Klicken Sie im Menü auf die Schaltfläche **Überspringen** . Der nächste Schritt des Import- oder Exportprozesses wird ausgeführt. Die Meldung im **ADIF-Debug-Fenster** ändert sich und liefert Informationen zum Erfolg des nächsten Schritts.
- 5) Wiederholen Sie Schritt 4 für jeden Prozessschritt. Sie können die folgenden Mechanismen verwenden, um zu überprüfen, ob das Ergebnis Ihrer Konfiguration korrekt ist:
 - Überprüfen Sie die Meldung im ADIF-Debug-Fenster, um über den Erfolg der Schrittausführung benachrichtigt zu werden. Wenn ein Fehler auftritt, zeigt der rote Pfeil im Explorer an, welcher Teil des Importschemas betroffen ist. Der XML-Code für das Element wird in der Debug-Meldung angezeigt, sodass Sie den Code auf Fehler überprüfen können.
 - Um den XML-Code eines anderen Elements des ADIF-Schemas zu überprüfen, klicken Sie im Explorer auf das Element und anschließend im Menü des ADIF-Debuggers auf die Schaltfläche **Definition anzeigen** .



- Um zu prüfen, ob unerwartete Ergebnisse durch die falsche Interpretation des Alfabet - Metamodells verursacht werden, können Informationen zu den Objektklassen des Alfabet - Metamodells und ihren Eigenschaften angezeigt werden. Um eine Dokumentation der Anforderungen für den Datenimport in die Datenbanktabelle einer Objektklasse anzuzeigen, klicken Sie mit der rechten Maustaste auf die Klasse unter dem untergeordneten Knoten **Klassentabellen** des Explorers des ADIF-Debuggers und wählen Sie **Klassendefinition anzeigen** aus.



- Wenn als Ergebnis eines Schritts eine temporäre Tabelle erzeugt wird, wird die temporäre Tabelle im Explorer des ADIF-Debuggers unter dem untergeordneten Knoten **Temporäre Tabellen** aufgeführt.



- Wenn Daten in eine temporäre Tabelle oder eine Datenbanktabelle des Alfabet -Metamodells geschrieben werden, überprüfen Sie die daraus resultierenden Datensätze in der Tabelle. Um den Inhalt einer Datenbanktabelle zu überprüfen, klicken Sie entweder unter dem untergeordneten Knoten **Klassentabellen** oder dem untergeordneten Knoten **Temporäre Tabellen** des Explorers auf die Datenbanktabelle, und klicken Sie anschließend im Menü auf die Schaltfläche **Neue SQL** . Es wird ein neues Fenster mit einer vorgegebenen Abfrage angezeigt, die die ersten hundert Zeilen in der Datenbanktabelle findet. Sie können diese Abfrage ändern, um die Datenbanktabelle nach den erwarteten Veränderungen zu durchsuchen. Um das Ergebnis Ihrer Abfrage anzuzeigen, klicken Sie im Menü auf die Schaltfläche **SQL ausführen** .



Die **RELATIONS**-Tabelle wird nicht als untergeordneter Knoten des Knotens **Klassentabellen** angezeigt. Trotzdem kann sie kontrolliert werden. Klicken Sie hierfür

auf die Schaltfläche **Neue SQL** , und definieren Sie ein **SELECT**-Statement, dessen Ziel die **RELATIONS**-Tabelle ist.



Es wird empfohlen, dass Sie die Änderungen an Datenbanktabellen auch dann überprüfen, wenn ein Schritt, bei dem Tabelleneinträge erzeugt werden, nicht zu einer Fehlermeldung führt. Fehlermeldungen in einem späteren Schritt können durch eine fehlende, unvollständige oder falsche Datenerzeugung in einem vorherigen Schritt verursacht werden.

	ID	NAME	VERSION	R
1	APP-3217	ACCOUNT	1	Alf
2	APP-3329	ACCOUNT	1.2	Jol
3	APP-2787	Administrative General Ledger	1.0	Ad
4	APP-3275	AF Enterprise ConTrol	3.1	Jol
5	APP-3269	AF Enterprise ConTrol	4.0	Jol
6	APP-3276	AF Good Buy	2.0	Jol
7	APP-3268	AF Good Buy	3.0	Jol
8	APP-3277	AF HR Online	2.2	Jol
9	APP-3306	AF HR Online	3.0	Jol
10	APP-3285	AF HR Online EU	2.2 Var.	Jol



Wenn die Alfabet-Datenbank sich auf einem Oracle® Datenbankserver befindet, werden Ganzzahlen als Zahlen des Datentyps Real angezeigt. Oracle® Datenbankserver speichern Ganzzahlen im Datentyp NUMBER und geben als Ergebnis einer SQL-Suche Dezimalzahlen zurück.

- 6) Bei einem Fehler beim Fortschritt des Datenimports oder -exports klicken Sie im Menü auf die Schaltfläche **Stoppen**, schließen Sie den Debugger und korrigieren Sie über den ADIF-Explorer den Fehler in der Konfiguration des ADIF-Schemas. Fahren Sie anschließend mit Schritt 1 des Debugging-Prozesses fort.



Beim Schließen des Debuggers wird unabhängig vom Attribut **Änderungen beibehalten** ein Rollback der Daten durchgeführt.

- 7) Nach der erfolgreichen Durchführung des Debuggers wird eine Protokolldatei angezeigt, die Informationen zum Gesamtfortschritt des Prozesses gibt. Beenden Sie den Debugger.

Erforderliche Konfiguration des ADIF-Schemas nach dem Debuggen

Nach dem erfolgreichen Debugging muss das ADIF-Schema so konfiguriert werden, dass es Daten dauerhaft in die Alfabet-Datenbank schreibt:

- 1) Klicken Sie im ADIF-Explorer auf das ADIF-Schema, bei dem das Debugging durchgeführt wurde.
- 2) Setzen Sie im Attribut Fenster des ADIF-Schemas das Attribut **Änderungen beibehalten** auf "True".



Wenn das ADIF-Schema als XML-Datei verfügbar sein soll, müssen Sie es über die ADIF-Konfigurationsbenutzeroberfläche aus der Alfabet-Datenbank exportieren. Informationen über das Erzeugen und Aktualisieren des Indexes finden Sie unter *Externe Editoren im Konfigurieren von ADIF-Schemata*.

Kapitel 7: Starten von Datenimport, Datenexport oder Datenänderung mit ADIF

Nachdem Sie die ADIF-Schnittstelle konfiguriert und Ihre Konfiguration mit dem ADIF-Debugger kontrolliert haben, können Sie das konfigurierte ADIF-Schema entweder auf Abruf oder auf regelmäßiger Basis für den Import, Export oder die Manipulation von Daten verwenden.

Es gibt verschiedene Möglichkeiten, den ADIF-Import, -Export oder die Manipulation von Daten durchzuführen:

- Software AG bietet ein Windows®-Befehlszeilentool `ADIF_Console.exe` zur Durchführung des Imports, Exports oder der Manipulation von Daten.
- Der Alfabet RESTful Service bietet die Möglichkeit, den ADIF-Import oder -Export über einen RESTful-Serviceaufruf durchzuführen.
- Der ADIF-Import oder -Export kann über ein Ereignis ausgelöst werden, das im Hintergrund gestartet wird, wenn ein Anwender einen Assistenten oder Workflowschritt aufruft oder beendet.
- Der ADIF-Import oder -Export kann vom Anwender über eine benutzerdefinierte Schaltfläche ausgelöst werden, die der Symbolleiste einer Alfabet -Ansicht auf der Alfabet -Benutzeroberfläche hinzugefügt wurde.
- Der ADIF-Import oder -Export kann über die Alfabet -Benutzeroberfläche in der Funktionalität **ADIF-Prozessverwaltung** überprüft oder ausgelöst werden.
- Der ADIF-Import oder -Export kann über die Funktionalität **Job-Planer** für die Ausführung in der Zukunft geplant werden.



ADIF-Importe können eine hohe Anzahl von einzelnen Transaktionen zum Einfügen und Löschen in einer Datenbank auslösen. Dies kann sich negativ auf die Fragmentierung von Indizes auswirken. Dadurch wird die CPU-Auslastung erhöht und die Leistung verringert. Es wird empfohlen, Indizes für Objektklassen, die dem ADIF-Import unterliegen, regelmäßig neu zu erzeugen. Dies kann mithilfe des Befehlszeilentools `AlfaAdministratorConsole.exe` erfolgen. Informationen hierzu finden Sie unter *Neuerzeugen von Indizes für Datenbanktabellen* im Referenzhandbuch *Systemadministration*.

Folgende Informationen sind verfügbar:

- [Ausführen von ADIF über ein Befehlszeilen-Tool](#)
- [Ausführen von ADIF über RESTful-Serviceaufrufe](#)
- [Ausführen von ADIF über ein Ereignis](#)
- [Ausführen von ADIF über eine Schaltfläche in der Alfabet-Benutzeroberfläche](#)
 - [Konfigurieren des ADIF-Schemas zur Ausführung über die Benutzeroberfläche](#)
 - [Hinzufügen einer Schaltfläche zur ADIF-Ausführung zu einem konfigurierten Bericht](#)
 - [Hinzufügen einer Schaltfläche zur ADIF-Ausführung zu einer Objektansicht](#)
- [Ausführen und Überprüfen von ADIF über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" in der Alfabet -Benutzeroberfläche](#)
- [Konfigurieren der Verfügbarkeit von ADIF-Ausführung und -Überprüfung in der Benutzeroberfläche](#)

- [Konfigurieren der asynchronen Ausführung](#)
- [Überprüfen des Erfolgs von ausgeführten ADIF-Prozessen](#)
- [Ausführen von ADIF](#)
- [Testen der ADIF-Schema-Ausführung](#)
- [Planen von ADIF-Prozessen für die Ausführung zu einem definierten Zeitpunkt mit der Funktionalität "Prozesszeitplan"](#)
- [Voraussetzungen zum Verwenden des Job-Planers](#)
 - [Konfigurieren des ADIF-Schemas zur Ausführung über die Funktionalität "Job-Planer"](#)
 - [Konfigurieren von Zugriffsberechtigungen für Ordner in der internen Dokumentenauswahl für den Prozesszeitplan](#)
 - [Konfigurieren eines Anwenders zur Ausführung selbstreflektierender Ereignisse](#)
 - [Ändern des Intervalls für die Überprüfung der Warteschlangen mit geplanten Ereignissen und ADIF-Prozessen](#)
- [Erzeugen eines Prozesszeitplans für das Löschen alter ADIF-Sitzungsinformationen per Batch-Prozess](#)

Ausführen von ADIF über ein Befehlszeilen-Tool

Nachdem Sie die ADIF-Schnittstelle konfiguriert und Ihre Konfiguration mit dem ADIF-Debugger kontrolliert haben, können Sie das konfigurierte ADIF-Schema entweder auf Abruf oder auf regelmäßiger Basis für den Import, Export oder die Manipulation von Daten verwenden.

Software AG bietet ein Windows®-Befehlszeilentool `ADIF_Console.exe` zur Durchführung des Imports, Exports oder der Manipulation von Daten.



Aus Gründen der Sicherheit sollten beim Importieren von Daten in die Alfabet-Datenbank oder beim Exportieren von Daten in externe Datenbanken die folgenden Empfehlungen beachtet werden:

- Testen Sie vor jedem über die ADIF-Konsolenanwendung durchgeführten Import in eine Produktionsumgebung den Import von Daten zuerst in einer Testumgebung, um sicherzustellen, dass die richtigen Änderungen angewendet werden.
- Sichern Sie vor dem Import der Daten die Produktionsdatenbank.

Ausführbare Datei: `ADIF_Console.exe` befindet sich im Unterordner **Programs** des Alfabet-Komponenten -Installationsverzeichnisses

Voraussetzungen: Die Konfigurationsdatei `AlfabetMS.xml` ist mit einer Server-Alias-Konfiguration verfügbar, in der die Verbindung zu der Alfabet-Datenbank definiert ist, und die jeweilige Funktionalität ist konfiguriert.

Protokollierung: Die Protokolldaten werden in eine Protokolldatei und in die Alfabet-Datenbank geschrieben. Die Protokollierung ist über das verwendete ADIF-Schema oder die Befehlszeile konfigurierbar. Weitere Informationen zur Protokollierung finden Sie unter *Konfigurieren von Protokollierungsparametern* und in der nachfolgenden Liste der Befehlszeilenoptionen.

Be- Starten Sie die ausführbare Datei mit `-h` oder `-help`
fehlszeilenhilfe:

Ausführbare Dateien für Batchaufträge können über eine Befehlszeile oder einen Windows®-Batchauftrag gestartet werden.

Bei Ausführung eines Windows-Batchauftrags wird dessen Ausführungszeit mithilfe des Windows-Planers für Batchaufträge definiert. Wenn eine ausführbare Datei mit einer Befehlszeile gestartet wird, wird der Batchauftrag sofort ausgeführt.

Die Einstellung auf der Registerkarte **Anwendungsserver** muss in allen Server-Alias Konfigurationen sämtlicher Alfabet -Komponenten einschließlich Batch-Hilfsprogrammen identisch sein.

Bereitstellen relevanter Daten und Konfiguration

Für die Ausführung der ADIF-Konsole ist der Zugriff auf die folgenden Dateien erforderlich:

Ein vollständig konfiguriertes ADIF-Schema

Das ADIF-Import- oder -Exportschema muss vollständig konfiguriert und debugged sein, bevor der Datenimport oder -export gestartet wird. Das ADIF-Schema kann als XML-Datei auf dem lokalen Host oder als Teil der Alfabet-Datenbank bereitgestellt werden.



Wenn Sie ein ADIF-Schema aus einer XML-Datei verwenden, müssen Sie sicherstellen, dass kein ADIF-Schema mit einem Attribut **Name** in der Alfabet-Datenbank vorhanden ist, das mit dem Attribut **Name** des ADIF-Schemas in der XML-Datei identisch ist.

Das ADIF-Schema in der XML-Datei wird während des Imports temporär in die Alfabet-Datenbank geladen und nach dem Import-/Exportprozess aus der Alfabet-Datenbank gelöscht. Wenn beim Starten der ADIF-Konsole ein ADIF-Schema mit demselben **Namen** in der Alfabet-Datenbank verfügbar ist, wird dieses ADIF-Schema mit dem ADIF-Schema in der XML-Datei überschrieben und nach dem Import-/Exportprozess gelöscht.

Eine Konfigurationsdatei "AlfabetMS.xml"

Um auf die Alfabet-Datenbank zugreifen zu können, muss die Konfigurationsdatei `AlfabetMS.xml` mit einer Alias-Konfiguration vorhanden sein.

Die Ausführung von Batch-Prozessen hängt von der Einstellung der Attribute auf der Registerkarte „Applikationsserver“ der Server-Alias-Konfigurationen der Alfabet -Komponenten ab:

- Wenn die Option **Ereigniswarteschlange für alle Prozesse verwenden** ausgewählt ist, werden sämtliche Serverprozesse über die Ereigniswarteschlange geplant. Die ADIF-Konsolenapplikation plant einen Prozess zur Ausführung des ADIF-Prozesses in der Alfabet-Datenbank. Der Alfabet-Server überprüft regelmäßig die Warteschlange zu verarbeitender Prozesse und verarbeitet die eingehenden Prozesse. Über verschiedene Threads können mehrere ADIF-Prozesse parallel

verarbeitet werden. Der Alfabet-Server startet Prozesse nur dann parallel, wenn sie keine Daten in dieselbe Datenbanktabelle in der Alfabet-Datenbank schreiben.

Dies ist die empfohlene Option und wird so in zukünftigen Releases beibehalten.

- Wenn **Applikationsserver und Net-Remoting-Service verwenden** ausgewählt ist, übergibt die ADIF-Konsolenapplikation den Prozess zur Verarbeitung an den Alfabet-Server. Die ADIF-Konsolenapplikation wartet, bis der Alfabet-Server die Ausführung beendet hat, und ruft anschließend vom Alfabet-Server Protokoll und Erfolgsstatus ab. Der Alfabet-Server verarbeitet immer nur einen ADIF-Prozess gleichzeitig. Wenn Sie versuchen, einen ADIF-Prozess zu starten, während ein anderer Prozess ausgeführt wird, wird gemäß Standardeinstellung der neue Prozess beendet und eine Fehlermeldung angezeigt. Sie können die ADIF-Konsole aber auch so konfigurieren, dass die auszuführenden Prozesse in eine Warteschlange gestellt werden. Damit der Prozess in die Warteschlange aufgenommen wird, muss in der Befehlszeile eine maximale Zeit für die Warteschlange definiert sein. Wenn eine Zeit für die Warteschlange definiert ist und ein Prozess gestartet wird, während noch ein anderer Prozess ausgeführt wird, wird der neue Prozess in die Warteschlange gestellt und ausführt, sobald der erste Prozess abgeschlossen ist. Wenn der Prozess sich länger in der Warteschlange befindet, als in der maximalen Zeit für die Warteschlange angegeben ist, wird er ohne Ausführung aus der Warteschlange entfernt.

Wenn zur Ausführung der ADIF-Konsolenapplikation ein Remote-Alias angegeben ist, muss der entsprechende Alfabet-Server laufen. Wird ein Server-Alias genutzt, müssen alle anderen Alfabet-Applikationen, die auf die Alfabet-Datenbank zugreifen, angehalten werden.

Diese Methode wird zukünftig wegfallen, da sie nicht mehr mit der neuen.NET-Core-Komponente kompatibel ist.

Beim Starten der ADIF-Konsolenanwendung muss der Name der Alias-Konfiguration in der Befehlszeile angegeben sein. Befindet sich die Konfigurationsdatei `AlfabetMS.xml` mit der Alias-Konfiguration nicht im Arbeitsverzeichnis der ADIF-Konsolenanwendung, muss der Pfad zu der Datei zusätzlich in der Befehlszeile angegeben werden.

Für den Import: Zugriff auf die zu importierenden Daten

Wenn die Daten aus Dateien importiert werden sollen, müssen sich die Dateien, die diese Daten enthalten, entweder in einer einzigen *.zip-Datei oder in einem einzigen Importverzeichnis befinden. Wenn sich die Dateien in einem Importverzeichnis befinden, erstellt die ADIF-Konsolenanwendung als Teil des Importprozesses zur Verwendung beim Datenimport eine ZIP-Datei mit allen Dateien, die sich im Importverzeichnis befinden.



Wenn der Import von einer XML-Datei durchgeführt werden soll, die auf eine DTD-Datei verweist, muss die DTD-Datei ebenfalls in der zu importierenden ZIP-Datei bzw. im Importverzeichnis enthalten sein.

Die Dateien werden in die Alfabet-Datenbank hochgeladen und nach der Durchführung des Imports aus der Datenbank gelöscht. Es hängt von der Konfiguration des Importschemas ab, ob die Importdateien auf dem lokalen Host durch den Importprozess geändert werden. Informationen hierzu finden Sie unter *Konfigurieren von Datenimporten mit ADIF*.

Wenn Daten aus mehreren Importdateien in einem ADIF-Import importiert werden und eine der angegebenen Importdateien nicht vorhanden ist, wird der Import aller anderen Dateien durchgeführt. Zudem wird in der Protokolldatei eine Warnung verzeichnet, dass aus der fehlenden Datei keine Daten importiert werden konnten.

Starten der ADIF-Konsolenanwendung

Starten Sie `ADIF_Console.exe` mit den in der nachfolgenden Tabelle aufgeführten Befehlszeilenoptionen, um einen Datenimport auf der Basis eines ADIF-Importschemas durchzuführen. Wenn eine Befehlszeilenoption eine Parameterdefinition erfordert und der Parameter Leerzeichen enthält, muss der Parameter in Anführungszeichen gesetzt werden. Beispiel:

```
ADIF_Console.exe -import -msalias Alfabet -scheme "Application Import"
```

Für den Import von Daten muss beim Start der `ADIF_Console.exe` mindestens Folgendes vorhanden sein:

- eine Spezifikation der Alias-Konfiguration, die den Zugriff auf einen Alfabet-Server oder eine Alfabet-Datenbank definiert,
- das Importschema und, falls Daten aus Dateien importiert werden,
- der Speicherort der Importdateien im lokalen Dateisystem:



```
ADIF_Console.exe -import -msalias <alias name> [-scheme <scheme name> -importfile <file name>.zip]
```

Für den Export von Daten muss beim Start der `ADIF_Console.exe` mindestens Folgendes vorhanden sein:

- eine Spezifikation der Alias-Konfiguration, die den Zugriff auf einen Alfabet-Server oder eine Alfabet-Datenbank definiert,
- das Exportschema und, falls Daten in Dateien exportiert werden,
- der Speicherort im lokalen Dateisystem, an dem die Exportdateien gespeichert werden sollen.



```
ADIF_Console.exe -export -msalias <alias name> -alfaLoginName <user name> -alfaLoginPassword <user password> [-scheme <scheme name> -importfile <file name>.zip]
```

Befehlszeilenoption	Obligato- risch/Standard	Erklärung
-<Aktion>	Obligatorisch	<p>Um Daten zu importieren, starten Sie die Konsolenanwendung mit</p> <pre>-import</pre> <p>Um Daten zu exportieren, starten Sie die Konsolenanwendung mit</p> <pre>-export</pre> <p>In der Befehlszeilenhilfe, die mit der Befehlszeilenoption <code>-h</code> aufgerufen werden kann, wird auch ein Aktionstyp <code>-update</code> neben <code>-import</code> und <code>-export</code> aufgeführt. Diese Option ist nur zur internen Verwendung durch Software AG und darf nicht verwendet werden.</p>

Befehlszeilenoption	Obligatorisch/Standard	Erklärung
<code>-msalias <Alias-name></code>	Obligatorisch	Geben Sie den Alias-Namen so an, wie er in der Konfigurationsdatei <code>AlfabetMS.xml</code> spezifiziert wurde.
<code>-msaliasesfile <Pfad zur Alfabet-Konfigurationsdatei></code>	Optional	Wenn die Konfigurationsdatei <code>AlfabetMS.xml</code> , in der der Alias angegeben ist, sich nicht im gleichen Verzeichnis wie die EXE-Datei befindet, muss der Pfad zur Datei <code>AlfabetMS.xml</code> mit diesem Parameter angegeben werden.
<code>-alfaLoginName <Anwendername></code>	Obligatorisch	Anwendername für den Zugriff auf die Alfabet-Datenbank. Der durch diesen Anwendernamen definierte Benutzer wird als Verantwortlicher für die Datenänderung in den Überwachungsverlauf geänderter Objekte geschrieben.  Ein Anwender kann einen Batch-Prozess nur ausführen, wenn das Kontrollkästchen Kann Batch-Prozess ausführen (=True) für den Anwender aktiviert ist.
<code>-alfaLoginPassword <Anwenderpasswort></code>	Optional	Passwort für den Zugriff auf die Alfabet-Datenbank.
<code>-scheme</code>	Optional	Geben Sie den Namen des in der Alfabet-Datenbank gespeicherten ADIF-Schemas an, das die Konfiguration des Datenimports/-exports enthält. HINWEIS: Es ist obligatorisch, diese Option oder die Option <code>-schemefile</code> anzugeben. Es kann nur eine der Optionen in einer Befehlszeile angegeben werden. Welche der Optionen anzuwenden ist, hängt vom Speicherort des ADIF-Schemas ab.
<code>-schemefile</code>	Optional	Geben Sie den Namen und den Pfad der XML-Datei des ADIF-Schemas an, das die Konfiguration des Datenimports/-exports enthält. HINWEIS: Es ist obligatorisch, diese Option oder die Option <code>-scheme</code> anzugeben. Es kann nur eine der Optionen in einer Befehlszeile angegeben werden. Welche der Optionen anzuwenden ist, hängt vom Speicherort des ADIF-Importschemas ab.  Wenn bei der Ausführung des Auftrags ein Schema mit demselben Namen wie das Schema in der externen XML-Datei in der Alfabet-Datenbank vorhanden ist, wird das Schema in der Alfabet-Datenbank gelöscht. Dieser Mechanismus wird implementiert, um sicherzustellen, dass nur eine Version eines Schemas vorhanden ist. Wenn das

Befehlszeilenoption	Obligato- risch/Standard	Erklärung
		Schema über eine externe XML-Datei aufgerufen wird, wird davon ausgegangen, dass das gleichnamige Schema in der Alfabet-Datenbank veraltet ist und nicht mehr verwendet wird.
-waittime	Optional	<p>Nur anwendbar, wenn Alfabet -Komponenten zur Verarbeitung von ADIF-Prozessen Ereigniswarteschlangen verwenden.</p> <p>Geben Sie die Zeit in Minuten an, die der Prozess abwarten soll, falls ein anderer ADIF-Prozess ausgeführt wird. Es können nicht zwei Prozesse parallel laufen. Ein neuer Prozess wird nur dann in die Warteschlange zur Ausführung gesetzt, wenn die Befehlszeilenoption <code>-waittime</code> gesetzt wurde. Wenn der aktuell ausgeführte Prozess während der konfigurierten Wartezeit nicht abgeschlossen wird, wird der Prozess in der Warteschlange gelöscht.</p> <p>Wenn die Option <code>-waittime</code> nicht gesetzt wurde und ein Prozess gestartet wird, während ein anderer Prozess aktuell ausgeführt wird, wird der neue Prozess beendet, und es wird eine Fehlermeldung ausgegeben.</p>
-idletime	Optional	<p>Nur anwendbar, wenn Alfabet -Komponenten zur Verarbeitung von ADIF-Prozessen Ereigniswarteschlangen verwenden.</p> <p>Das Batch-Hilfsprogramm wartet auf Alive-Meldungen des Alfabet-Server -Servers, und wenn binnen 60 Sekunden keine Meldungen eingehen, wird der Auftrag abgebrochen. Die zulässige Wartezeit für Alive-Meldungen kann geändert werden. Definieren Sie hierzu mittels der Befehlszeilenoption <code>-idletime</code> ein neues Zeitintervall in Sekunden.</p>
-synchronously	Optional	<p>Nur anwendbar, wenn Alfabet -Komponenten zur Verarbeitung von ADIF-Prozessen Ereigniswarteschlangen verwenden. Wenn dieser Parameter der Befehlszeile hinzugefügt wird, überprüft die ADIF-Konsolenapplikation die Ereigniswarteschlange auf den Status des geplanten Ereignisses, bis die Verarbeitung des Ereignisses abgeschlossen ist, und gibt anschließend den Status der beendeten ADIF-Prozessausführung zurück.</p>
-<Variablenname>	Optional	<p>Wenn das ADIF-Schema für die Verwendung von Variablen konfiguriert ist, können die Variablen in der Befehlszeile angegeben werden. Für jede einzelne Variable muss eine separate Befehlszeilenoption spezifiziert werden, und zwar als</p> <p style="text-align: center;"><code>-<variable name> <value></code></p>

Befehlszeilenoption	Obligato- risch/Standard	Erklärung
		Beispiel: <pre>-@fiscalyear 2009</pre> Weitere Info zur Verwendung von Variablen finden Sie in der Beschreibung der Exportkonfiguration unter <i>Konfigurieren der Ausführung in Abhängigkeit von aktuellen Parametern</i> oder in der Beschreibung der Importkonfiguration unter <i>Konfigurieren des Imports in Abhängigkeit von Parametern</i> .

Importspezifische Befehlszeilenoptionen

<code>-importfile</code>	Optional	Geben Sie Namen und Pfad der ZIP-Datei an, die die Dateien enthält, aus denen Daten importiert werden sollen. HINWEIS: Wenn die Konfiguration des Importschemas den Import aus Dateien konfiguriert, ist es obligatorisch, diese Option oder die Option <code>-importdir</code> anzugeben. Es kann nur eine der Optionen in einer Befehlszeile angegeben werden.
<code>-idletime</code>	Optional	Geben Sie Namen und Pfad des Verzeichnisses an, das die Dateien enthält, aus denen Daten importiert werden sollen. HINWEIS: Wenn die Konfiguration des Importschemas den Import aus Dateien konfiguriert, ist es obligatorisch, diese Option oder die Option <code>-importfile</code> anzugeben. Es kann nur eine der Optionen in einer Befehlszeile angegeben werden.

Exportspezifische Befehlszeilenoptionen

<code>-exportfile</code>	Optional	Geben Sie Namen und Pfad der ZIP-Datei an, in die die Daten geschrieben werden. Wenn die Datei nicht existiert, wird sie erzeugt; wenn sie bereits existiert, wird sie überschrieben. Wenn der Pfad nicht vorhanden ist, kann der ADIF-Export nicht ausgeführt werden. Die Exportmechanismen können zwar Dateien, aber keine Verzeichnisse erstellen. HINWEIS: Wenn die Konfiguration des ADIF-Exportschemas den Export in Dateien konfiguriert, ist es obligatorisch, diese Option anzugeben.
<code>-unzip <true false></code>	Optional	Wenn diese Option auf <code>true</code> gesetzt ist, wird die ZIP-Datei, in die die Exportdaten geschrieben werden, nach dem Export entpackt. Die Dateien mit den Exportdaten befinden

Befehlszeilenoption	Obligato- risch/Standard	Erklärung
		<p>sich dann in dem Verzeichnis, das als Speicherort für die ZIP-Dateien angegeben wurde, sowie in der ZIP-Datei.</p> <p>Wenn diese Option auf <code>false</code> gesetzt ist, wird die ZIP-Datei, in die die Exportdaten geschrieben werden, nicht entpackt, und die Daten sind nur im gezippten Format verfügbar.</p>

Protokollspezifische Befehlszeilenooptionen

<code>-logpath <Protokolldateipfad></code>	Die Protokolldatei wird im Arbeitsverzeichnis der ausführbaren Datei gespeichert.	Angabe eines Pfads zu einem Verzeichnis zum Speichern der Protokolldatei.
<code>-logfile <Protokolldateiname></code>	<code><Executable>.log</code>	Angabe des Namens der Protokolldatei. Zulässige Dateierweiterungen: LOG und TXT.
<code>-logverbose</code>		<p>Wenn <code>-logverbose</code> definiert ist, werden zusätzliche Informationen zum laufenden Prozess protokolliert. Der Inhalt der zusätzlichen Informationsmeldungen hängt vom verwendeten Hilfsprogramm ab.</p> <p>HINWEIS: Eine ausführliche Protokollierung ist in den meisten Fällen nicht erforderlich. Sie kann zu Leistungseinbußen führen.</p>
<code>-nologappend</code>		<p>Wenn <code>-nologappend</code> festgelegt ist, wird jedes Mal eine neue Protokolldatei erzeugt, wenn das Hilfsprogramm mit derselben Spezifikation von <code>-logfile</code> und <code>-logpath</code> verwendet wird. Der Name der Protokolldatei erhält eine Erweiterung, die aus einem Zeitstempel besteht, der die aktuelle UTC-Zeit angibt.</p> <p>Wenn <code>-nologappend</code> nicht definiert ist, werden die Protokolldaten bei jeder Verwendung der ausführbaren Datei an die bereits vorhandene Protokolldatei angefügt.</p> <p>HINWEIS: Zur Beschränkung der Dateigröße können Sie die Option <code>-logclear</code> definieren, um alte Protokollmeldungen zu löschen.</p>
<code>-logclear <Anzahl Tage></code>	Unendlich	<p>Diese Option kann nur verwendet werden, wenn <code>-nologappend</code> nicht festgelegt ist. Wenn <code>-nologappend</code> festgelegt ist, wird die Einstellung <code>-logclear</code> ignoriert.</p> <p>Während der Protokollierung wird die Protokolldatei auf Protokollmeldungen überprüft, die einen Zeitstempel</p>

Befehlszeilenoption	Obligato- risch/Standard	Erklärung
		<p>haben, der älter ist als die mit <code>-logclear</code> angegebene Anzahl an Tagen. Diese Meldungen werden dann gelöscht.</p> <p>HINWEIS: Der Prüfungsvorgang kann zu Leistungseinbußen führen.</p>
-heartbeat	-1	<p>Angabe des Zeitabstands beim Senden von E-Mails mit dem aktuellen Inhalt der Protokolldatei an den mit der Option <code>-recipientmail</code> angegebenen Empfänger. Falls <code>-1</code> angegeben wurde oder kein Takt definiert ist, wird keine E-Mail gesendet und die Protokollinformationen sind ausschließlich über die Protokolldatei der ADIF-Konsolenanwendung verfügbar.</p> <p>HINWEIS: Die E-Mail ist als Alive-Meldung des Systems gedacht, mit der der Empfänger über die Verfügbarkeit eines aktiven ADIF-Prozesses informiert wird. Der Inhalt der E-Mail ist auf die eine Zeile beschränkt, die in dem Moment in die Protokolldatei geschrieben wird, in dem die E-Mail generiert wird, und ist daher nicht für das Debuggen geeignet.</p> <p>HINWEIS: Das Intervall für das Senden der E-Mails kann auch mit dem Attribut Debug-Heart-Beat des ADIF-Schemas angegeben werden, das mit der Option <code>-scheme</code> oder <code>-scheme file</code> festgelegt wurde. Eine Spezifikation in der Befehlszeile überschreibt eine Spezifikation im ADIF-Schema.</p> <p>Mitteilung: Auch die E-Mail-Adressen von Empfänger und Absender müssen entweder in der Befehlszeile oder in dem mit der Option <code>-scheme</code> oder <code>-scheme file</code> angegebenen ADIF-Schema angegeben werden, um Informationen zum Protokollierungsprozess per E-Mail zu senden.</p>
-sendermail		<p>Spezifikation der E-Mail-Adresse, die als Absenderadresse in E-Mails mit Protokollinformationen verwendet wird, die in dem mit <code>-heartbeat</code> angegebenen Intervall an die mit <code>-recipientmail</code> angegebene Empfängeradresse gesendet werden.</p> <p>HINWEIS: Die E-Mail-Adresse des Absenders kann auch mit dem Attribut Absender-E-Mail des ADIF-Schemas angegeben werden, das mit der Option <code>-scheme</code> oder <code>-scheme file</code> festgelegt wurde. Eine Spezifikation in der Befehlszeile überschreibt eine Spezifikation im ADIF-Schema.</p>
-recipientmail		<p>Spezifikation der E-Mail-Adresse, die in E-Mails mit Protokollinformationen als Empfängeradresse verwendet wird, die in dem mit <code>-heartbeat</code> angegebenen Intervall gesendet werden.</p>

Befehlszeilenoption	Obligato- risch/Standard	Erklärung
		HINWEIS: Die E-Mail-Adresse des Empfängers kann auch mit dem Attribut Empfänger-E-Mail des ADIF-Schemas angegeben werden, das mit der Option <code>-scheme</code> oder <code>-scheme:file</code> festgelegt wurde. Eine Spezifikation in der Befehlszeile überschreibt eine Spezifikation im ADIF-Schema.

Ausführen von ADIF über RESTful-Serviceaufrufe

Der ADIF-Export oder -Import kann über einen RESTful-Serviceaufruf an die RESTful-API der Alfabet-Webapplikation ausgelöst werden.

Für den Serviceaufruf muss die RESTful-Serviceschnittstelle von Alfabet aktiviert sein, und alle Zugriffsberechtigungen müssen entsprechend eingestellt sein.

Die Ausführung des ADIF-Imports oder ADIF-Exports über einen RESTful-Serviceaufruf kann entweder durch eine externe Applikation ausgelöst werden, von der die Anfrage an die RESTful API der Alfabet-Webapplikation gesendet wird, oder über ein Ereignis, das automatisch aus einem Schritt eines Workflows oder Assistenten heraus ausgelöst wird.

Informationen zu den erforderlichen Einstellungen und zum Serviceaufruf für die Ausführung von ADIF-Import oder ADIF-Export von einem RESTful-Serviceclient aus finden Sie im Referenzhandbuch *Alfabet RESTful API*.

Informationen zur Konfiguration von Ereignissen finden Sie unter *Konfigurieren von Ereignissen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Ausführen von ADIF über ein Ereignis

Die Ereignisverwaltung bietet die Möglichkeit, aufgrund einer definierten Anwenderaktion auf der Alfabet - Benutzeroberfläche automatisch einen ADIF-Import- oder ADIF-Exportprozess auszuführen. Der ADIF-Import oder -Export kann entweder mit derselben oder mit einer anderen Alfabet-Datenbank ausgeführt werden.

Die folgenden Anwenderaktivitäten können angegeben werden, um Ereignisse auszulösen:

- Wenn ein Assistentenschritt begonnen, beendet oder abgebrochen wird.
- Wenn ein Workflowschritt begonnen, abgelehnt oder beendet wird oder abgelaufen ist.
- Wenn eine Ereignisvorlage, die für das Senden eines RESTful-Serviceaufrufs konfiguriert ist, abgeschlossen ist.

Die Ausführung des Ereignisses erfolgt über einen RESTful-Serviceaufruf an die RESTful API der Alfabet-Webapplikation, die die ADIF-Schemaausführung wie im Ereignis definiert auslöst.

Informationen zur Konfiguration von Ereignissen finden Sie unter *Konfigurieren von Ereignissen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Ausführen von ADIF über eine Schaltfläche in der Alfabet-Benutzeroberfläche

ADIF-Prozesse können über eine benutzerdefinierte Schaltfläche im Filterbereich eines konfigurierten Berichts oder in der Symbolleiste von benutzerdefinierten Objektansichten gestartet werden. Die Symbolleistschaltfläche ist dann für das Objektprofil und alle Objekt-Cockpits verfügbar, die für die Objektansicht definiert sind.

Das über den ADIF-Prozess ausgelöste ADIF-Schema kann Parameter enthalten, die sich auf die Filterfelder im Filterbereich des konfigurierten Berichts oder auf das Basisobjekt des konfigurierten Berichts oder des Objekts, mit dem der Anwender in der Objektansicht arbeitet, beziehen. Informationen zum Einschließen von Parametern in ADIF-Schemata finden Sie unter [Konfigurieren der Ausführung des Importschemas in Abhängigkeit von aktuellen Parametern](#) im Kapitel [Konfigurieren von Datenexporten mit ADIF](#) oder unter [Konfigurieren des Imports in Abhängigkeit von Parametern](#) im Kapitel [Konfigurieren von Datenimporten mit ADIF](#).

Wenn der ADIF-Prozess Daten in eine Datei exportiert, werden die exportierten Dateien in einer ZIP-Datei mit dem Namen <ADIF-Schemaname>_<Zeitstempel>.zip zum Herunterladen bereitgestellt.

Wenn der ADIF-Prozess Daten aus einer Datei importiert, wird der Anwender aufgefordert, eine Datei aus dem lokalen Dateisystem auszuwählen.

Der Erfolg der Ausführung kann über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" überprüft werden. Informationen hierzu finden Sie unter [Ausführen und Überprüfen von ADIF über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" in der Alfabet -Benutzeroberfläche](#).

Um einen ADIF-Prozess über eine Schaltfläche in einem konfigurierten Bericht oder einer Objektansicht zu starten, müssen sowohl das ADIF-Schema, das der Prozess ausführt, als auch der konfigurierte Bericht oder die Objektansicht wie in den folgenden Abschnitten beschrieben konfiguriert werden:

- [Konfigurieren des ADIF-Schemas zur Ausführung über die Benutzeroberfläche](#)
- [Hinzufügen einer Schaltfläche zur ADIF-Ausführung zu einem konfigurierten Bericht](#)
- [Hinzufügen einer Schaltfläche zur ADIF-Ausführung zu einer Objektansicht](#)

Konfigurieren des ADIF-Schemas zur Ausführung über die Benutzeroberfläche

So konfigurieren Sie ein ADIF-Schema, das über die Alfabet -Benutzeroberfläche ausführbar ist:

- 1) Klicken Sie im Explorer auf der Registerkarte **ADIF** von Alfabet Expand auf das ADIF-Schema, das über die Schaltflächeninteraktion ausgeführt werden soll.
- 2) Setzen Sie im Attributfenster das Attribut **Verhalten der Alfabet-Benutzeroberfläche** auf `VisibleExecutable`.

Hinzufügen einer Schaltfläche zur ADIF-Ausführung zu einem konfigurierten Bericht

So erstellen Sie eine Schaltfläche für die ADIF-Ausführung im Filterbereich eines vorhandenen konfigurierten Berichts mit einem Filterbereich:

- 1) Erweitern Sie im Explorer der Registerkarte **Berichte** von Alfabet Expand den Knoten des konfigurierten Berichts, dem Sie eine Schaltfläche hinzufügen möchten.
- 2) Doppelklicken Sie auf den Knoten der Berichtsansicht unter dem Knoten des konfigurierten Berichts. Die Berichtsansicht wird im mittleren Bereich angezeigt.
- 3) Klicken Sie in der Symbolleiste des mittleren Bereichs auf das Steuerelement **Schaltfläche**  und dann in den Filterbereich. Dem Filterbereich wird eine neue Schaltfläche hinzugefügt. Deren Position und Größe können Sie bei Bedarf ändern.
- 4) Definieren Sie im Attributfenster die folgenden Attribute für die neue Schaltfläche:
 - **Name:** Optional können Sie hier den technischen Namen der Schaltfläche in einen aussagekräftigen Namen ändern. Der Name ist ein technischer Name und sollte keine Leerzeichen oder Sonderzeichen enthalten.
 - **Titel:** Ändern Sie den Titel der Schaltfläche in den Text, der dem Anwender auf der Alfabet - Benutzeroberfläche in der Schaltfläche angezeigt werden soll.
 - **Hinweis:** Geben Sie einen Text für eine QuickInfo ein. Die QuickInfo wird angezeigt, wenn der Anwender mit der Maus auf die Schaltfläche zeigt.
 - **Untergeordneter Typ:** Wählen Sie `ADIF_Button` aus.
 - **XML-Definition:** Klicken Sie auf die Schaltfläche **Durchsuchen** , und geben Sie den folgenden XML-Code in den Editor ein, der geöffnet wird:


```
<ADIF_Button Type="Prozesstyp" Scheme="Schemaname">
</ADIF_Button>
```

Legen Sie das XML-Attribut `Type` auf "Export" fest, um einen ADIF-Export auszulösen, oder auf "Import", um einen ADIF-Import auszulösen.

Geben Sie den Namen des auszuführenden ADIF-Schemas in das XML-Attribut `Scheme` ein.
- 5) Klicken Sie in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Hinzufügen einer Schaltfläche zur ADIF-Ausführung zu einer Objektansicht

So erstellen Sie eine Schaltfläche für die ADIF-Ausführung in der Symbolleiste einer vorhandenen benutzerdefinierten Objektansicht:

- 1) Navigieren Sie im Explorer auf der Registerkarte **Darstellungen** von Alfabet Expand zu der benutzerdefinierten Objektansicht, der Sie eine Schaltfläche hinzufügen möchten.
- 2) Klicken Sie mit der rechten Maustaste auf die benutzerdefinierte Objektansicht , und wählen Sie **Schaltfläche erzeugen** aus. Die neue Schaltfläche  wird unterhalb des Knotens der benutzerdefinierten Objektansicht angezeigt.
- 3) Klicken Sie auf die neue Schaltfläche, um das Attributfenster zu aktivieren, und definieren Sie die folgenden Attribute nach Bedarf:
 - **Schaltflächentyp:** Wählen Sie `Action` aus.

- **Name:** Geben Sie einen eindeutigen Namen für die Schaltfläche ein. Sonderzeichen sind nicht zulässig. Dieser Wert wird in der Benutzeroberfläche der Lösung angezeigt, wenn das Attribut **Titel** nicht definiert wurde.
 - **Titel:** Geben Sie den Titel ein, der auf der Schaltfläche angezeigt werden soll. Wenn kein Titel definiert ist, wird der Wert im Attribut **Name** auf der Schaltfläche angezeigt.
 - **Operation:** Wählen Sie `ADIF_Operation` aus.
 - **Operation anwenden auf:** Wenn das ADIF-Schema den Parameter `@BASE` verwendet, um auf das aktuelle Objekt zu verweisen, legen Sie diesen Parameter auf `Base Instance` fest, sodass die Schaltfläche Informationen über das Basisobjekt bereitstellt, mit dem der Anwender aktuell arbeitet.
 - **Im Ansichtsmodus deaktivieren:** Wählen Sie `True` aus, wenn die Schaltfläche nur für Anwender mit Lese-/Schreibzugriffsrechten auf das Objekt verfügbar sein soll. Wählen Sie `False` aus, wenn die Schaltfläche allen Anwendern zugänglich sein soll, die auf das Objektprofil des Objekts zugreifen.
 - **Symbol:** Wählen Sie ein Symbol aus, das auf der Schaltfläche angezeigt werden soll. Das Symbol darf nicht größer als 22 x 22 Pixel sein.
 - **Hinweis:** Geben Sie einen Text für eine QuickInfo ein. Die QuickInfo wird angezeigt, wenn der Anwender mit der Maus auf die Schaltfläche zeigt.
- 4) Klicken Sie zum Sortieren der Schaltflächen des Objektprofils auf die Objektansicht , für die die Schaltflächen erzeugt wurden, um so das Attributfenster zu aktivieren. Klicken Sie im Attribut **Schaltflächen** auf die Schaltfläche **Durchsuchen** . Daraufhin wird ein Editor geöffnet, in dem Sie die Abfolge der Schaltflächen in der Symbolleiste mittels der Schaltfläche **Nach unten/Nach oben**  definieren können. Sortieren Sie die Schaltflächen, und klicken Sie zum Schließen des Editors auf **OK**.
- 5) Klicken Sie in der Symbolleiste auf die Schaltfläche **Speichern** .

Ausführen und Überprüfen von ADIF über die Funktionalitäten "ADIF-Prozessverwaltung" und "Eigene ADIF-Prozesse" in der Alfabet -Benutzeroberfläche

Sie können den ADIF-Import und -Export starten und den Erfolg der Ausführung direkt in der Alfabet -Benutzeroberfläche überprüfen.

Für die Ausführung und Überprüfung von ADIF-Prozessen stehen zwei Business-Funktionen zur Verfügung:

- Die Business-Funktion **ADIF-Prozessverwaltung** (`ADMIN_AdifJobs`) zum Ausführen und Testen des ADIF-Imports oder -Exports und zum Überprüfen des Erfolgs von ADIF-Importen und ADIF-Exporten.
- Die Business-Funktion **Eigene ADIF-Prozesse** (`USER_AdifJobs`) zur Überprüfung des Erfolgs des ADIF-Imports oder -Exports, der vom aktuellen Anwender ausgeführt wurde.

Folgende Informationen sind verfügbar:

- [Konfigurieren der Verfügbarkeit von ADIF-Ausführung und -Überprüfung in der Benutzeroberfläche](#)
- [Konfigurieren der asynchronen Ausführung](#)
 - [Konfigurieren von Zugriffsberechtigungen für Ordner in der internen Dokumentenauswahl für den asynchronen Export in eine Datei](#)
 - [Konfigurieren eines Anwenders zur Ausführung selbstreflektierender Ereignisse](#)
- [Überprüfen des Erfolgs von ausgeführten ADIF-Prozessen](#)
- [Ausführen von ADIF](#)
- [Testen der ADIF-Schema-Ausführung](#)

Konfigurieren der Verfügbarkeit von ADIF-Ausführung und -Überprüfung in der Benutzeroberfläche

Die folgende Konfiguration ist erforderlich, um ADIF-Prozesse in der Alfabet -Benutzeroberfläche zu verwalten und/oder auszuführen:

- Stellen Sie dem Anwender die Funktionalitäten über das Anwenderprofil, die Navigationsseite oder die Konfiguration der Navigationsansicht zur Verfügung.



Weitere Informationen zum direkten Hinzufügen von Business-Funktionen zu Anwenderprofilen finden Sie unter *Funktionalitäten für Anwenderprofile zugänglich machen*. Informationen zum Hinzufügen von Business-Funktionen zu Anwenderprofilen über Navigationsseiten oder Navigationsansichten finden Sie im Referenzhandbuch *Entwerfen von Navigationsseiten für Alfabet*.

- Die Verfügbarkeit von ADIF-Prozessen in den Funktionalitäten wird pro ADIF-Schema gesteuert. Standardmäßig sind ein ADIF-Schema und die für das ADIF-Schema ausgeführten ADIF-Prozesse in der Alfabet -Benutzeroberfläche nicht sichtbar. Setzen Sie das Attribut **Verhalten der Alfabet-Benutzeroberfläche** auf einen der folgenden Werte, um die Ausführung und Verwaltung oder nur die Verwaltung über die Alfabet -Benutzeroberfläche zu ermöglichen:
 - `VisibleNotExecutable`: Informationen über die Ausführung von ADIF-Prozessen basierend auf dem ADIF-Schema werden in den Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** angezeigt. Wenn der Anwender das ADIF-Schema in der Funktionalität **ADIF-Prozessverwaltung** auswählt, werden die Schaltflächen für die Ausführung des ADIF-Imports oder ADIF-Exports deaktiviert.
 - `VisibleExecutable`: Informationen über die Ausführung von ADIF-Prozessen basierend auf dem ADIF-Schema werden in den Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** angezeigt. Der Anwender kann einen ADIF-Import oder ADIF-Export über die Schaltflächeninteraktion in der **ADIF-Prozessverwaltung** ausführen.



Es ist nicht möglich, Parameter für ADIF-Prozesse zu definieren, die über die Benutzeroberfläche ausgeführt werden. ADIF-Prozesse, die von Parametern abhängen, die während der Ausführung übergeben werden, sollten nicht auf `VisibleExecutable` eingestellt werden.

- ADIF-Prozesse können synchron oder asynchron über die Funktionalität "ADIF-Prozessverwaltung" konfiguriert werden. Die asynchrone Ausführung wird empfohlen, um Probleme zu vermeiden, die durch lang laufende ADIF-Prozesse verursacht werden, die die aktuelle Anwendersitzung blockieren. Die asynchrone Ausführung erfordert die Umsetzung einer Reihe von Vorbedingungen, die im nachfolgenden Abschnitt beschrieben sind.

Konfigurieren der asynchronen Ausführung

ADIF-Prozesse können asynchron über die Funktionalität **ADIF-Prozessverwaltung** oder **Eigene ADIF-Prozesse** ausgeführt werden. Die synchrone Ausführung blockiert die aktuelle Anwendersitzung, solange der ADIF-Prozess ausgeführt wird. Es wird daher empfohlen, die asynchrone Ausführung zu verwenden.

Zusätzlich zur in diesem Abschnitt beschriebenen Konfiguration gelten folgende Vorbedingungen zur Aktivierung der asynchronen Ausführung:

- Ein ausgeführter Alfabet-Server muss mit derselben Datenbank verbunden sein wie die Alfabet-Webapplikation, die die Funktionalität der asynchronen ADIF-Ausführung auf der Benutzeroberfläche bereitstellt.
- Die RESTful-Services der Alfabet-Webapplikation müssen aktiviert und konfiguriert werden wie im Kapitel *Aktivieren des Alfabet RESTful API auf Serverseite* im Referenzhandbuch *Alfabet RESTful API* beschrieben, wobei im Server-Alias von Alfabet-Webapplikation die Berechtigungen **Hat den ADIFAPIInvocation-Zugriff** erteilt wurden.
- [Konfigurieren von Zugriffsberechtigungen für Ordner in der internen Dokumentenauswahl für den asynchronen Export in eine Datei](#)
- [Konfigurieren eines Anwenders zur Ausführung selbstreflektierender Ereignisse](#)

Konfigurieren von Zugriffsberechtigungen für Ordner in der internen Dokumentenauswahl für den asynchronen Export in eine Datei

Der ADIF-Export in eine Datei über die asynchrone Ausführung kann ausschließlich die **Interne Dokumentenauswahl** als Ziel haben. Der Export von Daten in das lokale Dateisystem ist für die asynchrone Ausführung nicht verfügbar.

Der asynchrone Export erfordert die Einstellung expliziter Zugriffsberechtigungen für Ordner der **internen Dokumentenauswahl**.

So definieren Sie IDOC-Ordner, auf die über die asynchrone Ausführung zugegriffen werden kann:

- 1) Erweitern Sie in Alfabet Expand auf der Registerkarte **Präsentation** den Explorer-Knoten **XML-Objekte > Verwaltung**.
- 2) Klicken Sie im Explorer mit der rechten Maustaste auf das XML-Objekt **IDocManagerConfiguration**, und wählen Sie im Kontextmenü **XML bearbeiten** aus. Das XML-Objekt wird im Bereich in der Mitte angezeigt.
- 3) Geben Sie den folgenden Code ein:

```
<IDocManager>
  <UseCase Name="JobSchedule" >
```

```
<Folder Name="FolderShowName" Path="IDOC:\Path" />
</UseCase>
</IDocManager>
```

Das XML-Element `UseCase` kann über mehrere untergeordnete Elemente `Folder` verfügen, wobei jedes Element einen Ordner definiert, auf den über den Prozesszeitplan zugegriffen werden kann. Definieren Sie den Ordner mit den folgenden XML-Attributen des XML-Elements `Folder`:

- **Name:** Definieren Sie einen Titel, der für den Ordner im Editor zur Auswahl der Zielformat für den ADIF-Export in eine Datei angezeigt werden soll. Der Name muss nicht mit dem Namen des Ordners in der **Internen Dokumentenauswahl** identisch sein..
- **Path:** Definieren Sie den Pfad zum Ordner in der **internen Dokumentenauswahl**. Der Pfad muss mit `IDOC:\` beginnen und mit dem Namen des ausgewählten Ordners enden. Zwischen Ordnernamen müssen umgekehrte Schrägstriche verwendet werden. Der ADIF-Exportmechanismus kann im definierten Ordner und in allen untergeordneten Ordnern des definierten Ordners lesen und schreiben. Im Editor zur Auswahl der Zielformat für den ADIF-Export in eine Datei werden auch alle untergeordneten Ordner zur Auswahl aufgelistet.

4) Klicken Sie in der Symbolleiste auf die Schaltfläche **Speichern**  .

Konfigurieren eines Anwenders zur Ausführung selbstreflektierender Ereignisse

Der für die Ausführung von Ereignissen des Typs `SelfReflective` ausgewählte Anwender wird sowohl für Wakeup-Ereignisse bei der Prozessplanung als auch bei allen anderen ereignisbasierten Aktivitäten, die eine Ausführung über die RESTful API beinhalten, zur Authentifizierung von RESTful-Serviceaufrufen an die RESTful API der Alfabet-Webapplikation verwendet.



Weitere Informationen zum Auslösen von Aktionen über Ereignisvorlagen finden Sie im Kapitel *Konfigurieren von Ereignissen*.

Es wird empfohlen, einen Anwender zu erzeugen, der ausschließlich für das Ausführen von Ereignissen über die RESTful-Services verwendet wird. Der Anwender kann vom Zugriff auf beliebige Objekte ausgeschlossen werden. Hiervon ausgenommen sind das Ausführen von Prozesszeitplänen, das asynchrone Starten von ADIF-Prozessen über die Benutzeroberfläche und das Starten von Workflows, RESTful-Serviceaufrufen und/oder das Ausführen von ADIF-Prozessen über Ereignisse auf der Basis von Ereignisvorlagen. Sie können aber auch einen vorhandenen Anwender verwenden.

Der Anwender muss ein benannter Anwender mit mindestens einem Anwenderprofil sein. Das Anwenderprofil wird nicht zur Bewertung von Zugriffsberechtigungen verwendet. Ein schreibgeschütztes Anwenderprofil reicht aus, um die RESTful-Services im Kontext von Ereignissen, asynchroner ADIF-Ausführung und Prozesszeitplanung auszuführen.

Es kann nur ein Anwender zur Ausführung von Ereignissen des Typs `SelfReflective` ausgewählt werden. Wenn Sie diese Funktion einem Anwender zuweisen und bereits ein anderer Anwender zur Ausführung von Selbstreflexionsereignissen ausgewählt wurde, wird die Einstellung von diesem Anwender entfernt, wenn sie für jenen Anwender festgelegt wird, dem Sie die Einstellung aktuell zuweisen. Wenn also ein Anwender bereits bei der Aktivierung einer anderen Funktionalität definiert wurde, ist es sinnvoller, diesem Anwender die unten beschriebenen benötigten Zugriffsberechtigungen zu erteilen, als einen weiteren Anwender zu

erstellen. Dadurch können Sie sicherstellen, dass die Berechtigungen für die Funktionalität, für die der Anwender erstellt wurde, beibehalten werden.

So erzeugen Sie in der Alfabet -Benutzeroberfläche in der Funktionalität **Anwenderverwaltung** einen Anwender für die Ausführung von selbstreflektierenden Ereignissen:



Die gleiche Funktionalität ist auch über den Alfabet Administrator verfügbar. Weitere Information zum Zugreifen auf die Funktionalität zur Anwenderverwaltung in Alfabet Administrator finden Sie im Referenzhandbuch *Systemadministration* unter *Über den erweiterten Explorer des verbundenen Alias verfügbare Funktionalitäten*.

- 1) Klicken Sie in der Ansicht **Anwenderverwaltung** auf **Neu > Neuen Anwender erzeugen**. Ein Editor wird geöffnet.
- 2) Definieren Sie im Editor die folgenden Optionen:

Registerkarte **Basisdaten**:

- **Name:** Geben Sie einen aussagekräftigen Namen für den Anwender ein. Der Anwender ist ein technischer Anwender. Sie können ihm entweder den Namen einer vorhandenen Person zuweisen oder ihm einen Namen geben, der darauf hinweist, dass es sich um eine virtuelle Person handelt, die für die Ausführung einer Funktionalität definiert wurde.
- **Anwendername:** Geben Sie einen Anwendernamen ein. Der Anwendername wird von den RESTful-Services zur Identifizierung des Anwenders verwendet.
- **Typ:** Wählen Sie `NamedUser` aus.

Registerkarte **API-Berechtigungen**:

- **Hat Zugriff auf API V2:** Aktivieren Sie das Kontrollkästchen.
 - **API-Zugriffsoptionen:** Stellen Sie sicher, dass die folgenden Berechtigungen ausgewählt sind:
 - **Hat den ADIFAPIInvocation-Zugriff** für die asynchrone Ausführung von ADIF-Schemata über die Benutzeroberfläche.
 - **API-Passwort generieren:** Klicken Sie auf die Schaltfläche. Das Feld "API-Passwort" wird ausgefüllt. Kopieren Sie bei Ereignissen des Typs `Query` den **API-Anwendernamen** und das **API-Passwort**. Diese müssen in die Spezifikation des XML-Objekts `AlfabetIntegrationConfig` eingegeben werden, das im Abschnitt *Konfigurieren der Verbindungsparameter im XML-Objekt AlfabetIntegrationConfig* unten beschrieben wird.
- 3) Klicken Sie zum Schließen des Editors auf **OK**. Der neue Anwender wird der Tabelle in der Ansicht **Anwenderverwaltung** hinzugefügt.
 - 4) Wählen Sie den Benutzer in der Tabelle aus, und wählen Sie in der Symbolleiste **Aktion > Anwender als berechtigt für "Selbstreflektierende Ereignisse ausführen" festlegen** aus.
 - 5) Wählen Sie den Anwender in der Tabelle aus, und klicken Sie dann auf die Schaltfläche **Navigieren**
 - 6) Klicken Sie im Objektprofil des Anwenders auf **Zugeordnete Anwenderprofile**.
 - 7) Klicken Sie in der Symbolleiste der Ansicht **Alle Anwenderprofile** auf **Neu > Anwenderprofile zuordnen**.

- 8) Wählen Sie im Selektor ein Anwenderprofil aus, und klicken Sie auf **OK**, um es dem Anwender zuzuordnen. Bei Anwendern, die ausschließlich für Ereignisse verwendet werden, wird empfohlen, ein schreibgeschütztes Anwenderprofil zu verwenden.

Die oben beschriebene Konfiguration ist die Konfiguration, die nur für die asynchrone Ausführung von ADIF-Prozessen erforderlich ist. Darüber hinaus können Sie beliebige weitere Eigenschaften des Anwenders festlegen. Aus Sicherheitsgründen können Sie dem Anwender auch ein Anmeldepasswort zuweisen, obwohl das Anwenderpasswort für die Funktionalität **Job-Planer** nicht erforderlich ist. Weitere Informationen zu den verfügbaren Konfigurationsoptionen finden Sie im Referenzhandbuch *Anwender- und Lösungsverwaltung* unter *Definieren und Verwalten von Anwendern*.

Überprüfen des Erfolgs von ausgeführten ADIF-Prozessen

In den Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** werden ausgeführte ADIF-Prozesse in einem erweiterbaren Datensatz aufgelistet, der Informationen zur ADIF-Prozessausführung in der Struktur anzeigt, die für die ADIF-Schemata in Alfabet Expand definiert ist:

- Auf der ersten Ebene wird der Root-Knoten angezeigt. Dort finden Sie Informationen zur Gesamtzahl der ausgeführten Ereignisse, die den Suchbedingungen entsprechen, die im Filter der Ansicht definiert sind.
- Auf der zweiten Ebene werden ADIF-Schema-Gruppen und ADIF-Schemata angezeigt, die sich direkt unter dem Root-Knoten befinden.
- Die dritte Ebene zeigt die ADIF-Schemata in der ADIF-Gruppe oder, für ADIF-Schemata direkt unter dem Root-Knoten, die Liste der ausgeführten Prozesse an.
- Die vierte Ebene zeigt die Ausführung von ADIF-Prozessen für ADIF-Schemata an, die einer Gruppe zugewiesen sind. Die Ebene weist die beiden Abschnitte **Ausgeführte(r) Prozess(e)** und **Gestartete(r) Prozess(e)** auf. Wenn ein ADIF-Prozess asynchron gestartet wird, wird der ADIF-Prozess während der Ausführung im Abschnitt **Gestartete(r) Prozess(e)** aufgeführt. Nach Abschluss der Ausführung wird er in den Abschnitt **Ausgeführte(r) Prozess(e)** verschoben. Hinweis: Asynchrone ADIF-Prozesse werden in keinem der Abschnitte aufgeführt, wenn sie zur Ausführung über den Server geplant sind, die Ausführung aber noch nicht begonnen hat. Eine synchrone Ausführung von ADIF-Schemata verhindert ein Neuladen der Ansicht, sodass die Prozesse zu keinem Zeitpunkt im Abschnitt **Gestartete(r) Prozess(e)** angezeigt werden.

	1	2	3	4	ADIF-Schema-Gruppe	ADIF-Schemata	ADIF-Schema-Typ	Aktueller Status
Root-Knoten	ADIF-Prozesse 75(75)							
ADIF-Schemagruppe	aDIF examples							
ADIF-Schema	Applikationsumfrage Export							
Information über Ausführungen	Ausgeführte(r) Prozess(e): 2				aDIF examples	AppSurvey Export	Exportieren	Erfolg

Wenn ADIF-Prozesse für ein ADIF-Schema ein- oder mehrmals ausgeführt wurden, wird jede Ausführung als separate Zeile unter der Überschrift **Ausgeführte(r) Prozess(e)** mit den folgenden Informationen zur ADIF-Prozessausführung aufgelistet:

- **ADIF-Schema-Gruppe:** Der Name der ADIF-Schema-Gruppe, der das ausgewählte ADIF-Schema zugeordnet ist.
- **ADIF-Schemata:** Der Titel des ADIF-Schemas, das der ADIF-Prozess ausführt.

- **ADIF-Schema-Typ:** Zeigt **EXPORT** für die Ausführung eines ADIF-Exportschemas oder **IMPORT** für die Ausführung eines ADIF-Importschemas an.
- **Aktueller Status:** Zeigt den Gesamtstatus der ADIF-Prozessausführung an:
 - **Gestartet:** Der ADIF-Prozess wird derzeit ausgeführt. Der ADIF-Prozess wird im Abschnitt **Gestartete(r) Prozess(e)** aufgeführt.
 - **Erfolg:** Der ADIF-Prozess wurde erfolgreich ausgeführt.
 - **Warnung:** Der ADIF-Prozess wurde ausgeführt, es wurde jedoch eine Warnung in die Protokolldatei geschrieben, weil eine nicht-kritische Funktion fehlgeschlagen ist.
 - **Fehlgeschlagen:** Der ADIF-Prozess hat während der Ausführung einen Fehler zurückgegeben.
- **Prozesssitzung:** Die eindeutige Sitzungs-ID des ADIF-Prozesses.
- **Start:** Der Zeitpunkt, an dem die ADIF-Prozessausführung gestartet wurde.
- **Ende:** Der Zeitpunkt, an dem die ADIF-Prozessausführung beendet wurde.
- **Ausgeführt von:** Die Art und Weise, wie die ADIF-Prozessausführung gestartet wurde. In der Spalte wird eine der folgenden Optionen zurückgegeben:
 - `Expand ADIF Debugger`: Der ADIF-Prozess wurde zu Testzwecken über den ADIF-Debugger von Alfabet Expand Alfabet Expand ausgeführt.
 - `User Interface`: Die ADIF-Prozessausführung wurde über die Alfabet Alfabet - Benutzeroberfläche über eine Schaltflächeninteraktion gestartet. Dies wird auch für die Ausführung von Testprozessen in dieser Ansicht über die Option **Prozess ausführen > Nicht persistenter Test-Prozess** angezeigt.
 - `Rest API`: Der ADIF-Prozess wurde über einen RESTful-Serviceaufruf an die RESTful API von Alfabet ausgelöst.
 - `Meta-Model Update`: Der ADIF-Prozess wurde während einer Metamodellaktualisierung automatisch ausgeführt.
 - `Database Restore`: Der ADIF-Prozess wurde im Rahmen einer Datenbankwiederherstellung aus einer ADBZ-Datei von Alfabet automatisch ausgeführt.
- **Anwendername:** Der Anwendername des Anwenders, der den ADIF-Prozess gestartet hat. Bitte beachten Sie Folgendes hinsichtlich des Anwenders, der den ADIF-Prozess startet:
 - Für ADIF-Prozesse, die über eine Metamodell-Aktualisierung oder eine Datenbankwiederherstellung ausgeführt werden und die über den Alfabet Administrator gestartet wurden, sind keine Anwenderinformationen verfügbar, da für den Alfabet Administrator keine Anwenderanmeldung erforderlich ist.
 - Bei ADIF-Prozessen, die über einen RESTful-Serviceaufruf an den Endpunkt `adifimport` oder `adifexport` der RESTful API von Alfabet ausgeführt werden, ist der Authentifizierungsanwender für den RESTful-Serviceaufruf mit dem Anwender identisch, der den ADIF-Prozess ausführt. Dies gilt für RESTful-Serviceaufrufe von einem externen RESTful-Client sowie für RESTful-Serviceaufrufe, die über Ereignisse ausgelöst werden.

Es ist ein Filter verfügbar, um den Datensatz auf relevante Inhalte zu reduzieren. Legen Sie die folgenden Filterfelder oben in der Tabelle fest, und klicken Sie auf **Aktualisieren**, um nur Daten anzeigen zu lassen, die Ihren Filtereinstellungen entsprechen:

- **ADIF-Schema-Typ:** Wählen Sie **IMPORT** aus, um ADIF-Importschemata und ADIF-Prozesse nur für ADIF-Importschemata anzeigen zu lassen, oder **EXPORT**, um ADIF-Exportschemata und ADIF-Prozesse nur für ADIF-Exportschemata anzeigen zu lassen.
- **ADIF-Schemata:** Wählen Sie ein oder mehrere ADIF-Schemata aus dem Dropdown-Listefeld mit Mehrfachauswahl aus, um die Anzeige auf die ausgewählten ADIF-Schemata und die für die Schemata ausgeführten ADIF-Prozesse zu beschränken.
- **ADIF-Prozessstatus:** Wählen Sie eine der folgenden Optionen aus:
 - **Gestartet**, um nur aktuell ausgeführte ADIF-Prozesse anzeigen zu lassen.
 - **Erfolg**, um nur erfolgreich ausgeführte ADIF-Prozesse anzeigen zu lassen.
 - **Warnung**, um nur ADIF-Prozesse anzuzeigen, die mit einer Warnmeldung ausgeführt wurden.
 - **Fehlgeschlagen**, um nur ADIF-Prozesse anzeigen zu lassen, die mit einem Fehler beendet wurden.
 - **Beenden von Ausführung erzwungen**, um nur die ADIF-Prozesse anzuzeigen, deren Durchführung gescheitert ist, weil während der Ausführung der ausführende Alfabet-Server entweder geplant oder gezwungen heruntergefahren wurde oder der Thread für die Ausführung gezwungenermaßen beendet wurde.
- **Startdatum nach:** Wählen Sie ein Datum aus dem Kalender aus, um nur ADIF-Prozesse anzeigen zu lassen, die am oder nach dem ausgewählten Datum gestartet wurden.
- **Startdatum vor:** Wählen Sie ein Datum aus dem Kalender aus, um nur ADIF-Prozesse anzeigen zu lassen, die am oder vor dem ausgewählten Datum gestartet wurden.

So zeigen Sie die Protokollinformationen zu einem ausgeführten ADIF-Prozess an:

- 1) Klicken Sie in der Symbolleiste auf die Schaltfläche **Protokoll anzeigen**.
- 2) Klicken Sie im daraufhin angezeigten Fenster auf die Schaltfläche **Herunterladen**. Die Datei wird über die Downloadmechanismen Ihres Browsers heruntergeladen.



Die Schaltfläche **Prozessdetails** ist nur für die ServiceNow®-Integration verfügbar. Sie finden dort Informationen über die IDs der Protokollinformationen, die von ServiceNow® während der Ausführung des ADIF-Prozesses im ausführlichen Modus zurückgegeben werden. Die Protokollinformationen werden in einer Tabelle `ALFA_ADIF_SESSION_DETAIL` in der Alfabet-Datenbank gespeichert, die auf der Alfabet -Benutzeroberfläche oder in Alfabet Expand nicht angezeigt wird. Um die Protokollinformationen anzeigen zu lassen, können Sie einen einfachen konfigurierbaren Bericht erstellen, der auf der folgenden Abfrage basiert, und die Informationen exportieren.

```
SELECT REFSTR, SESSION_ID, SCHEME_NAME, DETAIL_ID, IN_CONTENT, OUT_CONTENT
FROM ALFA_ADIF_SESSION_DETAIL
```

Sie können Informationen zur ADIF-Prozessausführung löschen, wenn sie nicht mehr benötigt werden, um die Menge der in der zugrunde liegenden Datenbanktabelle gespeicherten Informationen klein zu halten und die Informationen in den Funktionalitäten **Prozessverwaltung** und **Eigene ADIF-Prozesse** klarer zu gestalten.

Um Informationen zu ADIF-Prozesssitzungen zu löschen, wählen Sie eine oder mehrere ADIF-Prozessinformationen in der Tabelle aus, und klicken Sie auf die Schaltfläche **Löschen** . Die ausgewählten Informationen zu ADIF-Prozesssitzungen werden unwiderruflich aus der Alfabet-Datenbank gelöscht.

Ausführen von ADIF

Sie können ADIF-Prozesse, die als ausführbar konfiguriert sind und bei denen zur Ausführung keine obligatorischen Parameter gesetzt sein müssen, über die Benutzeroberfläche in der Funktionalität **ADIF-Prozessverwaltung** starten.

ADIF-Prozesse werden asynchron ausgeführt. Sie werden zur Ausführung per Alfabet-Server in die Warteschlange gestellt. Die Ausführung kann durch andere in der Warteschleife befindliche ADIF-Prozesse verzögert werden. In der aktuellen Sitzung können Sie in der Alfabet -Benutzeroberfläche weiterarbeiten, während der ADIF-Prozess in die Warteschlange gestellt und ausgeführt wird. Solange er sich in der Warteschlange befindet, ist der Prozess nicht in der Tabelle der **ADIF-Prozessverwaltung** aufgeführt. Während der Ausführung ist der Prozess im Abschnitt **Gestartete(r) Prozess(e)** und nach Abschluss im Abschnitt **Ausgeführte(r) Prozess(e)** aufgeführt.

So starten Sie die Ausführung eines ADIF-Prozesses:

- 1) Klicken Sie in der Tabelle auf das ADIF-Schema, das Sie ausführen möchten.
- 2) Wählen Sie in der Symbolleiste eine der folgenden Optionen aus:
 - **Prozess asynchron über Server ausführen > Importprozess asynchron ausführen:** Der ADIF-Importprozess wird asynchron ausgeführt, und nur Fehler und Warnungen sowie die Start- und Endzeiten der Ausführung werden in die Protokolldatei eingetragen.
 - **Prozess asynchron über Server ausführen > Importprozess asynchron mit ausführlicher Protokollierung ausführen:** Der ADIF-Importprozess wird asynchron ausgeführt, und während der Ausführung werden detaillierte Informationen in die Protokolldatei eingetragen.
 - **Prozess asynchron über Server ausführen > Exportprozess asynchron ausführen:** Der ADIF-Exportprozess wird asynchron ausgeführt, und nur Fehler und Warnungen sowie die Start- und Endzeiten der Ausführung werden in die Protokolldatei eingetragen.
 - **Prozess asynchron über Server ausführen > Exportprozess asynchron mit ausführlicher Protokollierung ausführen:** Der ADIF-Exportprozess wird asynchron ausgeführt, und während der Ausführung werden detaillierte Informationen in die Protokolldatei eingetragen.



Wenn die Schaltflächenoptionen deaktiviert sind, ist das ausgewählte ADIF-Schema so konfiguriert, dass es auf der Alfabet -Benutzeroberfläche sichtbar, aber nicht ausführbar ist.

- 3) Wenn Sie ein ADIF-Importschema ausführen, das Daten aus einer Datei importiert, wird ein Dateiauswahlfenster geöffnet. Wählen Sie die Datei aus, die die hochzuladenden Daten enthält, und klicken Sie auf **Hochladen**.
- 4) Wenn Sie ein ADIF-Exportschema ausführen, das Daten in eine Datei exportiert, wird die Exportdatei in die **Interne Dokumentenauswahl** hochgeladen und steht über die Funktionalität **Interne Dokumente** zur Verfügung. In der internen Dokumentenauswahl wird ein Dialogfeld zur Auswahl der zulässigen Dateispeicherorte geöffnet. Markieren Sie in der Ordnerliste der Tabelle **IDOC-Ordner für Export** das Kontrollkästchen des Zielordners, und geben Sie optional im Feld **Name der Exportdatei** einen Dateinamen an. Der Dateiname muss die Erweiterung ZIP aufweisen. Wenn Sie keinen Dateinamen angeben, wird bei der Ausführung eine ZIP-Datei mit dem Namen <NameDesADIFSchemas>_<unformatierter_Zeitstempel>.zip erstellt.

Testen der ADIF-Schema-Ausführung

Sie können alle als ausführbar konfigurierten ADIF-Prozesse über die Benutzeroberfläche in der Funktionalität **ADIF-Prozessverwaltung** im Testmodus starten. Im Testmodus sind Informationen, die in Datenbanktabellen geschrieben werden, nicht persistent, aber Änderungen werden erst als letzte Aktion der ADIF-Prozessausführung zurückgesetzt. Die Folgen des Imports können über die Protokolldatei ohne Risiko für Ihre Daten getestet werden. Für ADIF-Prozesse, die Daten in Dateien exportieren, ist der Testmodus nicht verfügbar. Mit den Testoptionen wird die exportierte Datei wie im normalen Ausführungsmodus erstellt.

- 1) Klicken Sie in der Tabelle auf das ADIF-Schema, das Sie ausführen möchten.
- 2) Wählen Sie in der Symbolleiste eine der folgenden Optionen aus:
 - **Prozess asynchron über Server ausführen > Asynchron nicht persistenter Import-Test-Prozess:** Der ADIF-Importprozess wird asynchron ausgeführt, ohne dass die Daten in der Datenbank dauerhaft geändert werden; nur Fehler und Warnungen sowie die Start- und Endzeiten der Ausführung werden in die Protokolldatei eingetragen.
 - **Prozess asynchron über Server ausführen > Asynchron nicht persistenter Import-Test-Prozess mit ausführlicher Protokollierung:** Der ADIF-Importprozess wird asynchron ausgeführt, ohne dass die Daten in der Datenbank dauerhaft geändert werden; während der Ausführung werden detaillierte Informationen in die Protokolldatei eingetragen.
 - **Prozess asynchron über Server ausführen > Asynchron nicht persistenter Export-Test-Prozess:** Der ADIF-Importprozess wird asynchron ausgeführt, ohne dass die Daten in der Datenbank dauerhaft geändert werden; nur Fehler und Warnungen sowie die Start- und Endzeiten der Ausführung werden in die Protokolldatei eingetragen.
 - **Prozess asynchron über Server ausführen > Asynchron nicht persistenter Export-Test-Prozess mit ausführlicher Protokollierung:** Der ADIF-Importprozess wird asynchron ausgeführt, ohne dass die Daten in der Datenbank dauerhaft geändert werden; während der Ausführung werden detaillierte Informationen in die Protokolldatei eingetragen.
- 3) Wenn Sie die Ausführung eines ADIF-Importschemas testen, das Daten aus einer Datei importiert, wird ein Datei-Auswahlfenster geöffnet. Wählen Sie die Datei aus, die die hochzuladenden Daten enthält, und klicken Sie auf **Hochladen**.
- 4) Wenn Sie ein ADIF-Exportschema ausführen, das Daten in eine Datei exportiert, unterscheidet sich der Testmodus nicht vom normalen Ausführungsmodus.
 - Die Ergebnisse werden in eine persistente Datei geschrieben. Die Exportdatei wird in die **Interne Dokumentenauswahl** hochgeladen und steht über die Funktionalität **Interne Dokumente** zur Verfügung. In der **internen Dokumentenauswahl** wird ein Dialogfeld zur Auswahl der zulässigen Dateispeicherorte geöffnet. Markieren Sie in der Ordnerliste der Tabelle **IDOC-Ordner für Export** das Kontrollkästchen des Zielordners, und geben Sie optional im Feld **Name der Exportdatei** einen Dateinamen an. Der Dateiname muss die Erweiterung ZIP aufweisen. Wenn Sie keinen Dateinamen angeben, wird bei der Ausführung eine ZIP-Datei mit dem Namen <NameDesADIFSchemas>_<unformatierter_Zeitstempel>.zip erstellt.

Planen von ADIF-Prozessen für die Ausführung zu einem definierten Zeitpunkt mit der Funktionalität "Prozesszeitplan"

Die Funktionalität **Job-Planer** (`JobSchedule`) kann für administrative Anwender auf der Alfabet -Benutzeroberfläche zur Verfügung gestellt werden. Mit dieser Funktionalität können Anwender ADIF-Prozesse

entweder für eine einmalige Ausführung an einem definierten Datum und zu einer festgelegten Uhrzeit oder für eine wiederholte Ausführung zu definierten Zeiten und Intervallen in einem definierten Zeitraum planen.

Darüber hinaus kann die Funktionalität **Prozesszeitplan** verwendet werden, um alte ADIF-Sitzungsinformationen aus der Tabelle `ADIF_SESSION` in der Alfabet -Datenbank im Batch zu löschen, um die Größe der Datenbank zu verringern und die Übersichtlichkeit der Informationen in den Funktionalitäten **ADIF-Prozessverwaltung** und **Eigene ADIF-Prozesse** zu verbessern.

Der Prozesszeitplan erfordert, dass ein ausgeführter Alfabet-Server mit derselben Datenbank verbunden ist wie die Alfabet-Webapplikation, die den Prozesszeitplan auf der Benutzeroberfläche bereitstellt.

Wenn ein Anwender einen Prozesszeitplan erzeugt, werden die Informationen über die Planung als Objekt der Objektklasse " Alfabet -Prozesszeitplan" gespeichert und für die nächste Ausführung des Prozesses wird ein Wakeup-Ereignis gespeichert.

Der Alfabet-Server durchsucht die geplanten Ereignisse in regelmäßigen Abständen nach geplanten Prozessen, die ausgeführt werden müssen. Wenn ein Prozess fällig ist, stellt der Alfabet-Server den Prozess in die Warteschlange der auszuführenden Prozesse.

Die Funktionalität muss zunächst über die Konfiguration in Alfabet Expand aktiviert werden.

Folgende Informationen sind verfügbar:

- [Voraussetzungen zum Verwenden des Job-Planers](#)
- [Konfigurieren des ADIF-Schemas zur Ausführung über die Funktionalität "Job-Planer"](#)
- [Konfigurieren von Zugriffsberechtigungen für Ordner in der internen Dokumentenauswahl für den Prozesszeitplan](#)
- [Konfigurieren eines Anwenders zur Ausführung selbstreflektierender Ereignisse](#)
- [Ändern des Intervalls für die Überprüfung der Warteschlangen mit geplanten Ereignissen und ADIF-Prozessen](#)
- [Erzeugen eines Prozesszeitplans für das Löschen alter ADIF-Sitzungsinformationen per Batch-Prozess](#)

Voraussetzungen zum Verwenden des Job-Planers

Zusätzlich zur in diesem Abschnitt beschriebenen Konfiguration gelten folgende Voraussetzungen:

- Der Job Scheduler erfordert, dass ein ausgeführter Alfabet-Server mit derselben Datenbank verbunden ist wie die Alfabet-Webapplikation, die die Funktionalität **Prozesszeitplan** auf der Benutzeroberfläche bereitstellt.
- Die RESTful-Services der Alfabet-Webapplikation müssen aktiviert und konfiguriert werden, wie im Kapitel *Aktivieren des Alfabet RESTful API auf Serverseite* im Referenzhandbuch *Alfabet RESTful API* beschrieben. Der Server-Alias der Alfabet-Webapplikation muss die Berechtigung **Hat den ADIFAPIInvocation-Zugriff** gewähren, um ADIF-Prozesse über die Funktionalität **Prozesszeitplan** zu starten.

Die folgenden Konfigurationen sind für die Aktivierung und Konfiguration des Job-Planers relevant:

- [Konfigurieren des ADIF-Schemas zur Ausführung über die Funktionalität "Job-Planer"](#)

- [Konfigurieren von Zugriffsberechtigungen für Ordner in der internen Dokumentenauswahl für den Prozesszeitplan](#)
- [Konfigurieren eines Anwenders zur Ausführung selbstreflektierender Ereignisse](#)
- [Ändern des Intervalls für die Überprüfung der Warteschlangen mit geplanten Ereignissen und ADIF-Prozessen](#)

Konfigurieren des ADIF-Schemas zur Ausführung über die Funktionalität "Job-Planer"

Der ADIF-Prozess muss die folgenden Voraussetzungen erfüllen, damit er über die Funktionalität **Job-Planer** geplant werden kann:

- Er muss einer der Kategorien zugeordnet werden, die für den Anwendungsfall `JobScheduler` und den Bereich `ADIF` im XML-Objekt ***UseCaseCategories*** definiert sind.



Informationen über das Definieren von Kategorienamen im XML-Objekt ***UseCaseCategories*** finden Sie unter *Erstellen von Kategorien für den Anwendungsfall „Prozesszeitplan“*.

- Das ADIF-Schema darf keine Parameter enthalten, die im Kompatibilitätsmodus oder mit dem Datentyp `StringArray` oder `ReferenceArray` definiert sind.

So aktivieren Sie die Prozessplanung für ein ADIF-Importschema:

- 1) Klicken Sie auf der Registerkarte **ADIF** von Alfabet Expand auf das ADIF-Schema.
- 2) Wählen Sie im Attributfenster eine der Kategorien aus, die für den Anwendungsfall `JobScheduler` und den Bereich `ADIF` im XML-Objekt ***UseCaseCategories*** definiert sind.

Konfigurieren von Zugriffsberechtigungen für Ordner in der internen Dokumentenauswahl für den Prozesszeitplan

Ein über den Job-Planer geplanter ADIF-Import kann aus einer Datei in der **internen Dokumentenauswahl** der Alfabet-Datenbank ausgeführt werden. Diese Methode ist optional. Der Import kann auch aus einer Datei im lokalen Dateisystem ausgeführt werden. Der ADIF-Export in eine Datei über den Job-Planer kann ausschließlich die **interne Dokumentenauswahl** als Ziel haben. Der Export von Daten in das lokale Dateisystem ist für geplante Prozesse nicht verfügbar.

Der Job-Planer kann nur auf Ordner in der **internen Dokumentenauswahl** mit expliziten Zugriffsberechtigungen für den Job-Planer zugreifen.

So definieren Sie IDOC-Ordner, auf die über geplante ADIF-Prozesse zugegriffen werden kann:

- 1) Erweitern Sie in Alfabet Expand auf der Registerkarte **Präsentation** den Explorer-Knoten **XML-Objekte > Verwaltung**.
- 2) Klicken Sie im Explorer mit der rechten Maustaste auf das XML-Objekt ***IDocManagerConfiguration***, und wählen Sie im Kontextmenü **XML bearbeiten** aus. Das XML-Objekt wird im Bereich in der Mitte angezeigt.
- 3) Geben Sie den folgenden Code ein:

```
<IDocManager>
```

```

<UseCase Name="JobSchedule" >
    <Folder Name="AnzuzeigenderOrdnername" Path="IDOC:\Pfad" />
</UseCase>
</IDocManager>

```

Das XML-Element `UseCase` kann über mehrere untergeordnete Elemente `Folder` verfügen, wobei jedes Element einen Ordner definiert, auf den über den Job-Planer zugegriffen werden kann. Definieren Sie den Ordner mit den folgenden XML-Attributen des XML-Elements `Folder`:

- **Name:** Definieren Sie einen Titel, der für den Ordner im Editor zur Planung von ADIF-Prozessen angezeigt werden soll. Der Name muss nicht mit dem Namen des Ordners in der **Internen Dokumentenauswahl** identisch sein.
- **Path:** Definieren Sie den Pfad zum Ordner in der **internen Dokumentenauswahl**. Der Pfad muss mit `IDOC:\` beginnen und mit dem Namen des ausgewählten Ordners enden. Zwischen Ordnernamen müssen umgekehrte Schrägstriche verwendet werden. Der Job-Planer kann im definierten Ordner und in allen untergeordneten Ordnern des definierten Ordners lesen und schreiben.

- 4) Klicken Sie in der Symbolleiste auf die Schaltfläche **Speichern**  .

Konfigurieren eines Anwenders zur Ausführung selbstreflektierender Ereignisse

Der für die Ausführung von Ereignissen des Typs `SelfReflective` ausgewählte Anwender wird sowohl für Wakeup-Ereignisse bei der Prozesszeitplanung als auch bei allen anderen ereignisbasierten Aktivitäten, die eine Ausführung über die RESTful API beinhalten, zur Authentifizierung von RESTful-Serviceaufrufen an die RESTful-API der Alfabet-Webapplikation verwendet.



Weitere Informationen zum Auslösen von Aktionen über Ereignisvorlagen finden Sie im Kapitel *Konfigurieren von Ereignissen*.

Es wird empfohlen, einen Anwender zu erzeugen, der ausschließlich für das Ausführen von Ereignissen über die RESTful-Services verwendet wird. Der Anwender kann vom Zugriff auf beliebige Objekte ausgeschlossen werden, außer für das Ausführen von Prozesszeitplänen, das asynchrone Starten von ADIF-Prozessen über die Benutzeroberfläche und das Starten von Workflows, RESTful-Serviceaufrufen und/oder das Ausführen von ADIF-Prozessen über Ereignisse auf der Basis von Ereignisvorlagen. Sie können aber auch einen vorhandenen Anwender verwenden.

Der Anwender muss ein benannter Anwender mit mindestens einem Anwenderprofil sein. Das Anwenderprofil wird nicht zur Bewertung von Zugriffsberechtigungen verwendet. Ein schreibgeschütztes Anwenderprofil reicht aus, um die RESTful-Services im Kontext von Ereignissen auszuführen.

Es kann nur ein Anwender zur Ausführung von Ereignissen des Typs `SelfReflective` ausgewählt werden. Wenn Sie diese Funktion einem Anwender zuweisen und bereits ein anderer Anwender zur Ausführung von Selbstreflexionsereignissen ausgewählt wurde, wird die Einstellung von diesem Anwender entfernt, wenn sie für jenen Anwender festgelegt wird, dem Sie die Einstellung aktuell zuweisen. Wenn also ein Anwender bereits bei der Aktivierung einer anderen Funktionalität definiert wurde, ist es sinnvoller, diesem Anwender die unten beschriebenen benötigten Zugriffsberechtigungen zu erteilen, als einen weiteren Anwender zu erstellen. Dadurch können Sie sicherstellen, dass die Berechtigungen für die Funktionalität, für die der Anwender erstellt wurde, beibehalten werden.

So erzeugen Sie in der Alfabet -Benutzeroberfläche in der Funktionalität **Anwenderverwaltung** einen Anwender für die Ausführung von selbstreflektierenden Ereignissen:



Die gleiche Funktionalität ist auch über den Alfabet Administrator verfügbar. Weitere Information zum Zugreifen auf die Funktionalität zur Anwenderverwaltung in Alfabet Administrator finden Sie im Referenzhandbuch *Systemadministration* unter *Über den erweiterten Explorer des verbundenen Alias verfügbare Funktionalitäten*.

- 1) Klicken Sie in der Ansicht **Anwenderverwaltung** auf **Neu > Neuen Anwender erzeugen**. Ein Editor wird geöffnet.
- 2) Definieren Sie im Editor die folgenden Optionen:

Registerkarte **Basisdaten**:

- **Name:** Geben Sie einen aussagekräftigen Namen für den Anwender ein. Der Anwender ist ein technischer Anwender. Sie können ihm entweder den Namen einer vorhandenen Person zuweisen oder ihm einen Namen geben, der darauf hinweist, dass es sich um eine virtuelle Person handelt, die für die Ausführung einer Funktionalität definiert wurde.
- **Anwendername:** Geben Sie einen Anwendernamen ein. Der Anwendername wird von den RESTful-Services zur Identifizierung des Anwenders verwendet.
- **Typ:** Wählen Sie `NamedUser` aus.

Registerkarte **API-Berechtigungen**:

- **Hat Zugriff auf API V2:** Aktivieren Sie das Kontrollkästchen.
 - **API-Zugriffsoptionen:** Stellen Sie sicher, dass die folgenden Berechtigungen ausgewählt sind:
 - **Hat den ADIFAPIInvocation-Zugriff** für die Ausführung von ADIF-Schemata über den **ADIF-Import-Prozesszeitplan** und den **ADIF-Export-Prozesszeitplan**.
 - **Hat Zugriff auf Batch-Hilfesprogramme-API** für die Ausführung aller weiteren Prozesszeitpläne.
 - **API-Passwort generieren:** Klicken Sie auf die Schaltfläche. Das Feld „API-Passwort“ wird ausgefüllt. Kopieren Sie bei Ereignissen des Typs `Query` den **API-Anwendernamen** und das **API-Passwort**. Diese müssen in die Spezifikation des XML-Objekts `AlfabetIntegrationConfig` eingegeben werden, das im Abschnitt *Konfigurieren der Verbindungsparameter im XML-Objekt AlfabetIntegrationConfig* unten beschrieben wird.
- 3) Klicken Sie zum Schließen des Editors auf **OK**. Der neue Anwender wird der Tabelle in der Ansicht **Anwenderverwaltung** hinzugefügt.
 - 4) Wählen Sie den Benutzer in der Tabelle aus, und wählen Sie in der Symbolleiste **Aktion > Anwender als berechtigt für „Selbstreflektierende Ereignisse ausführen“ festlegen** aus.
 - 5) Wählen Sie den Anwender in der Tabelle aus, und klicken Sie dann auf die Schaltfläche **Navigieren**
 - 6) Klicken Sie im Objektprofil des Anwenders auf **Zugeordnete Anwenderprofile**.
 - 7) Klicken Sie in der Symbolleiste der Ansicht **Alle Anwenderprofile** auf **Neu > Anwenderprofile zuordnen**.

- 8) Wählen Sie im Selektor ein Anwenderprofil aus, und klicken Sie auf **OK**, um es dem Anwender zuzuordnen. Bei Anwendern, die ausschließlich für Ereignisse verwendet werden, wird empfohlen, ein schreibgeschütztes Anwenderprofil zu verwenden.

Die oben beschriebene Konfiguration ist die Konfiguration, die nur für die Funktionalität **Job-Planer** erforderlich ist. Darüber hinaus können Sie beliebige weitere Eigenschaften des Anwenders festlegen. Aus Sicherheitsgründen können Sie dem Anwender auch ein Anmeldepasswort zuweisen, obwohl das Anwenderpasswort für die Funktionalität **Job-Planer** nicht erforderlich ist. Weitere Informationen zu den verfügbaren Konfigurationsoptionen finden Sie im Referenzhandbuch *Anwender- und Lösungsverwaltung* unter *Definieren und Verwalten von Anwendern*.

Ändern des Intervalls für die Überprüfung der Warteschlangen mit geplanten Ereignissen und ADIF-Prozessen

Diese Konfiguration ist optional. Das Alfabet-Server -Standardintervall für die Überprüfung der Wakeup-Ereignisse, die für die Ausführung von Prozesszeitplänen für ADIF-Prozesse erstellt wurden, beträgt 500 Millisekunden. Das Alfabet-Server -Standardintervall für die Überprüfung der ADIF-Prozesswarteschlange und die Ausführung des nächsten ADIF-Prozesses beträgt 3.000 Millisekunden. Optional können Sie diese Werte an Ihre Anforderungen anpassen.

Die Konfiguration muss im Tool Alfabet Administrator durchgeführt werden:

- 1) Klicken Sie im Alfabet Administrator -Explorer auf den Knoten **Alfabet Aliases**.
- 2) Wählen Sie in der Tabelle die Serveralias-Konfiguration aus, mit der der Alfabet-Server gestartet wird.
- 3) Klicken Sie in der Symbolleiste auf die Schaltfläche „Bearbeiten“ . Der Serveralias-Editor wird angezeigt.
- 4) Wechseln Sie zur Registerkarte **Server-Einstellungen > Allgemein**, und bearbeiten Sie die folgenden Attribute:
 - **Ruhezustandszeit des ADIF-Prozessservers (Millisekunden):** Geben Sie das Intervall zwischen den Überprüfungen der Warteschlange mit den auszuführenden ADIF-Prozessen ein.
 - **Ruhezustandszeit des Ereignisservers (Millisekunden):** Geben Sie das Intervall zwischen den Überprüfungen der Ereigniswarteschlange mit geplanten Wakeup-Ereignissen ein, die ausgeführt werden sollen.
- 5) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern.

Erzeugen eines Prozesszeitplans für das Löschen alter ADIF-Sitzungsinformationen per Batch-Prozess

ADIF-Sitzungsinformationen, die älter als eine festgelegte Anzahl an Tagen sind, können für ein definiertes ADIF-Schema über einen Prozesszeitplan im Batch gelöscht werden. Die Informationen über die Prozessausführung können entweder komplett gelöscht werden, oder der Löschvorgang kann auf den Inhalt der Protokolldatei beschränkt werden, während die Informationen, dass der Prozess ausgeführt wurde, ebenso wie der sich daraus ergebende Ausführungsstatus beibehalten werden.

So planen Sie einen Prozess zur Löschung von ADIF-Sitzungsinformationen:

- 1) Klicken Sie in der Symbolleiste auf **Neu > Zeitplan erzeugen**.
- 2) Wählen Sie im angezeigten Dialogfeld **Stereotyp-Auswahl** die Option **Prozesszeitplan für Löschung von ADIF-Sitzungsinhalt**.
- 3) Definieren Sie im Editor die folgenden Attribute:

Registerkarte **Basisdaten**:

- **Name:** Definieren Sie einen Namen für den Prozesszeitplan. Der Prozesszeitplan wird in der Tabelle der Funktionalität **Prozesszeitplan** mit diesem Namen aufgeführt. Der Name muss eindeutig sein. Es können keine zwei Prozesszeitpläne gleichen Namens definiert werden, auch wenn sie unterschiedlichen Typs sind. Dieses Attribut ist obligatorisch.
- **Beschreibung:** Geben Sie eine aussagekräftige Beschreibung ein, die den Zweck des Prozesszeitplans erläutert.
- **ADIF-Schemata:** Wählen Sie in der Alfabet-Datenbank das ADIF-Schema aus, für das die Informationen zur Prozessausführung gelöscht werden sollen. Wenn kein ADIF-Schema ausgewählt ist, werden bei sämtlichen ADIF-Schemata die Informationen zur Prozessausführung gelöscht.
- **Tagesanzahl:** Aufbewahrungszeit der Informationen zur Prozessausführung in Tagen. Hierbei muss es sich um eine positive Ganzzahl handeln. Bei der Eingabe von 1 werden die Informationen über die Prozessausführung für den aktuellen Tag und den Vortag beibehalten. Bei der Eingabe von 0 werden sämtliche Informationen über die Prozessausführung, einschließlich der Informationen für den aktuellen Tag, gelöscht.
- **Ebene:** Wählen Sie **Protokoll**, damit nur der in der Datenbank gespeicherte Inhalt der Protokolldatei gelöscht wird, während die Informationen über das Datum der Prozessausführung und den Ausführungsstatus beibehalten werden. Wählen Sie **Alle**, um sämtliche Informationen über die Prozessausführung zu löschen.

Registerkarte **Zeitplan**:

- **Ausführungszeit:** Geben Sie die Startzeit für die Prozessausführung in das Feld **Startzeit** ein.
- **Recurrence Pattern:** Select one of the following check boxes and provide additional data, if applicable:
 - **Täglich:** Das Intervall zwischen Prozessausführungen ist die Anzahl der Tage, die im Feld **Tag(e)** als Ganzzahl angegeben sind.
 - **Wöchentlich:** Das Intervall zwischen Prozessausführungen ist die Anzahl der Wochen, die im Feld **Woche(n) am** als Ganzzahl angegeben sind. Aktivieren Sie außerdem das Kontrollkästchen des Wochentags, an dem der Prozess für jede Wiederholung ausgeführt werden soll.
 - **Monthly:** There are tow methods to select from. If you select the checkbox **Date-Based**, define a day of the month in the **day** field behind the checkbox. For example if you want the job to be executed on the fifteenth of each month, enter 15. If you select the checkbox **Weekday-Based**, you can schedule the job for a specific day of a specific week in the month. In the first field behind the checkbox, select the week of the month the job shall be executed. In the second field behind the checkbox, select the day of the week the job shall be executed. The interval between job executions is the number of weeks specified in the field **month(s)** field of the selected scheduling method.
 - **Einmal:** Der Prozess wird nur einmal ausgeführt.

- **Range of Recurrence:** This definition is mandatory. Select the start date for the job schedule from the calendar in the **Start Date** field. For the definition of the end of the recurrence period, either select **End after:** and define the maximum number of occurrences in the **Occurrences** field or select **End by:** and select the end date for the job scheduling from the calendar in the field next to the checkbox entry. If you have selected **Once** in the **Recurrence Pattern** field, only define the date of execution via the **Start Date** field.

Registerkarte **Ausführungsinformation:**

Diese Registerkarte bietet die Möglichkeit, die Prozesszeitplanung an die Umgebungsbedingungen anzupassen, etwa die Zeitplanung anderer Prozesse oder aber Wartungsfenster, während derer die Ausführung nicht stattfinden soll.

- **Maximale Wartezeit [Min.]:** Geben Sie eine maximale Wartezeit in Minuten ein, um die die Prozessausführung verschoben werden kann, wenn eine der folgenden Bedingungen erfüllt ist:
 - Die Ausführung von Prozessen für den aktuellen Prozesszeitplan hängt davon ab, dass die Ausführung eines oder mehrerer anderer Prozesse abgeschlossen ist, weil das Ausführungsergebnis eines Prozesses zur Verfügung stehen muss, um diesen Prozess ausführen zu können, oder weil beide Prozesse miteinander in Konflikt stehende Aktionen durchführen. Wenn ein Prozess des aktuellen Zeitplans ausgeführt werden soll, während noch ein im Feld **Abhängige Prozesse** definierter Prozess ausgeführt wird, wird der Prozess nicht ausgeführt, sondern für eine Ausführung fünf Minuten später eingeplant, wenn die abhängigen Prozesse abgeschlossen sind.
 - Ein Prozess für den aktuellen Prozesszeitplan kann nicht ausgeführt werden, weil die aktuelle Ausführungszeit in einem Wartungsfenster liegt. Bei Wartungsfenstern handelt es sich um Zeiträume, die gesperrt sind, um eine Unterbrechung des Prozesses durch Wartungsarbeiten zu verhindern. Wenn ein Prozess innerhalb eines Wartungsfensters fällig ist, wird er zunächst nicht ausgeführt. Er wird so neugeplant, dass er eine Minute nach Ende des Wartungsfensters ausgeführt wird.



Für die Alfabet-Komponenten müssen Wartungsfenster konfiguriert sein, damit die Prozesszeitplanung für das Wartungsfenster über die Einstellung **Maximale Wartezeit [Min.]** angepasst werden kann. Informationen über das Definieren von Wartungsfenstern finden Sie unter *Definieren von Wartungsfenstern* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* oder unter *Definieren von Wartungsfenstern für geplante Prozesse* im Referenzhandbuch *Systemadministration*.

Wenn der Zeitraum zwischen der ursprünglich fälligen Prozessausführung und dem Ende des Wartungsfensters oder dem Ende der Ausführung eines abhängigen Prozesses die maximale Wartezeit überschreitet, wird die Prozessausführung abgebrochen und eine Fehlermeldung in die Protokolldatei eingetragen. Die nächste Ausführung des Prozesses wird gemäß den Einstellungen in der Registerkarte **Zeitplan** des Editors Prozesszeitplan geplant.

- **Abhängige Prozesse:** Wenn die Ausführung von Prozessen für den aktuellen Prozesszeitplan verschoben werden soll, solange noch ein anderer Prozess ausgeführt wird, klicken Sie auf **Neu** > Neue Prozessabhängigkeit erzeugen. Wählen Sie in der Liste den Prozesszeitplan aus, von dem der aktuelle Prozesszeitplan abhängig ist, und klicken Sie dann auf **OK**. Wenn ein Prozess eines der abhängigen Prozesszeitpläne ausgeführt wird, wird die Ausführung des Prozesses aus dem aktuellen Prozesszeitplan um so viele Minuten verschoben, wie es dauert, bis die Ausführung sämtlicher abhängiger Prozesse abgeschlossen ist. Wenn die unter **Maximale Wartezeit [Min.]** definierte maximale Wartezeit überschritten ist und noch

abhängige Prozesse ausgeführt werden, wird die nächste Ausführung des Prozesses gemäß den Einstellungen in der Registerkarte **Zeitplan** des Editors Prozesszeitplan geplant.

- **Erwartete Ausführungsdauer [Min.]:** Wenn in diesem Feld (in Minuten) eine maximale Zeit für die Ausführung des Prozesses definiert wird, wird in die über die Schaltfläche **Protokoll anzeigen** verfügbare Protokolldatei eine Warnung eingetragen, wenn die Ausführung eines Prozesses für den aktuellen Prozesszeitplan die maximale Wartezeit überschreitet. Der Prozess wird dessen ungeachtet weiter ausgeführt.
- **Ausführender Anwender:** Sobald ein geplanter Prozess ausgeführt ist, wird dem Anwender, der Eigentümer des Prozesszeitplans ist, im Slide-in-Fenster **Ereignis-Feedback** angezeigt, ob die aktuelle Ausführung erfolgreich verlaufen ist. Wenn ein Ergebnis zum Download verfügbar ist, wird ein Download-Link angezeigt. Standardmäßig wird die Slide-in-Meldung demjenigen Anwender angezeigt, der den Prozesszeitplan erzeugt hat, sofern er mit dem bei der Erstellung des Prozesszeitplans verwendeten Anwenderprofil angemeldet ist. Dieses Standardverhalten kann geändert werden. Wählen Sie in der Dropdown-Liste den Anwender aus, dem stattdessen die Meldung **Ereignis-Feedback** angezeigt werden soll. Damit der Anwender die Meldungen **Ereignis-Feedback** für diesen Prozesszeitplan erhalten kann, muss er mit dem im Attribut **Ausführendes Anwenderprofil** definierten Anwenderprofil angemeldet sein.
- **Ausführendes Anwenderprofil:** Damit der Anwender die Meldungen **Ereignis-Feedback** bezüglich der Ausführung dieses Prozesszeitplans erhält, muss er mit dem im Attribut **Ausführender Anwender** definierten Anwenderprofil angemeldet sein. Standardmäßig wird die Meldung demjenigen Anwender angezeigt, der den Prozesszeitplan erzeugt hat, sofern er mit dem bei der Erstellung des Prozesszeitplans verwendeten Anwenderprofil angemeldet ist. Dieses Attribut ist obligatorisch, falls **Ausführender Anwender** definiert ist.

4) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern.

Nachdem Sie den Prozesszeitplan erstellt haben, können Sie ihn als Vorlage zum Erstellen neuer Prozesszeitpläne verwenden. Wählen Sie den Prozesszeitplan in der Tabelle aus, und klicken Sie in der Symbolleiste auf **Neu > Job-Zeitplan als Kopie erzeugen**.

Vordefinierte ADIF-Schemata

Die ADIF-Import- und Exportprozesse hängen in der Regel stark von der Umgebung und den Anforderungen des Kunden ab. Nur wenige Importprozesse sind von allgemeinem Interesse, unabhängig von der Kundenumgebung.

Vordefinierte ADIF-Schemata werden von Software AG bereitgestellt, wenn die Anforderung für einen bestimmten Import von allgemeinem Interesse ist. Diese vordefinierten ADIF-Schemata sind entweder privat und können daher nicht geändert oder geschützt werden. In diesem Fall können Sie nur eine Teilmenge der Attribute der vorkonfigurierten untergeordneten Elemente des ADIF-Schemas bearbeiten. Es ist möglich, neue ADIF-Elemente zu geschützten ADIF-Schemata hinzuzufügen. Die vom Kunden hinzugefügten ADIF-Elemente sind öffentlich und vollständig editierbar. Private und geschützte ADIF-Schemata haben ein Schloss-Symbol, das ihren Bearbeitungsstatus anzeigt. Das Schloss ist orange für geschützte ADIF-Schemata und rot für private ADIF-Schemata. Bei einem geschützten ADIF-Schema ist die Editierbarkeit der Attribute nur in den Attributfenstern der untergeordneten Elemente sichtbar. Deaktivierte Attribute werden in Grau angezeigt.

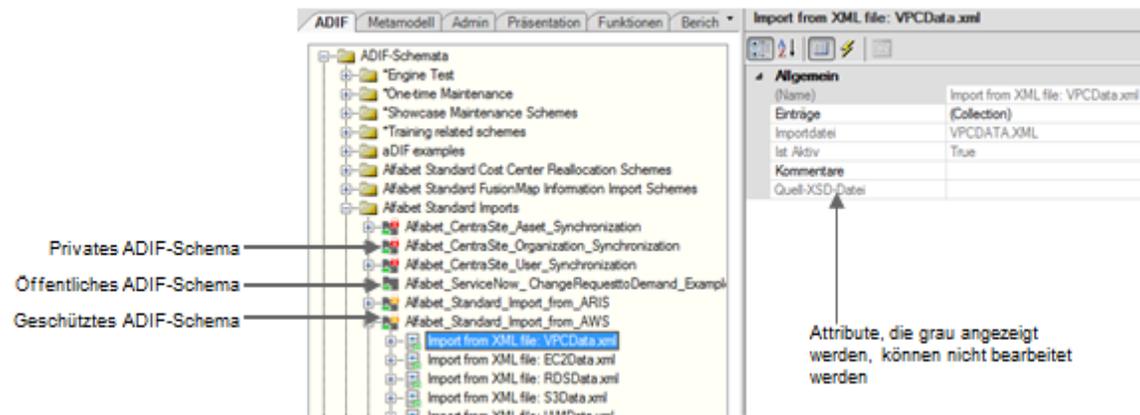


Abbildung: Private, geschützte und öffentliche ADIF-Schemata im ADIF Explorer

Vordefinierte ADIF-Schemata werden als Bestandteil der ADIF-Standardfunktionen ausgeliefert und sind als Teil der ausgelösten Funktionalität oder im Folgenden dokumentiert. Sie können vor der Verwendung an die Kundenanforderungen angepasst werden. Die Ausführung erfolgt über die ADIF-Konsolenapplikation, auf dieselbe Art wie die Ausführung eines kundendefinierten ADIF-Schemas.

Die folgenden ADIF-Schemata sind vordefiniert:

Gruppe	ADIF-Importschema	Bereitgestellte Funktionen	Weitere Informationen hierzu finden Sie unter...
Alfabet Standard Cost Center Reallocation Schemes	CC_Reallocate	ADIF kann dazu verwendet werden, die Budgetkosten und die Objektzuweisung einer Kostenstelle sowie die Kostenzuordnungsverhältnisse als Batch-Prozess zu aktualisieren. Bei allen Importen über ADIF erfolgt die Kosten-Neuzuordnung auf die den Kostenstellen zugeordneten Objekte nicht automatisch nach dem Import der geänderten Verrechnungsdaten. Die Neuzuordnung der Kosten muss entweder manuell oder über das vordefinierte ADIF-Importschema erfolgen.	Importschema für die automatische Neuzuordnung von Kostenstellenkosten
Alfabet Standard FusionMap Information Import Schemes	FUSIONMAPINFO_IMPORT		
Alfabet Standardimporte	Alfabet_CentraSite_Asset_Synchronization	Wird die Schnittstelle zur Datensynchronisation mit CentraSite verwendet, kann über dieses vordefinierte ADIF-Schema ein standardisierter Batch-Import von Informationen über gemeinsame Assets von Alfabet und CentraSite durchgeführt werden. Das ADIF-Schema ist ein geschütztes Schema und kann nicht benutzerdefiniert bearbeitet werden.	Weitere Informationen über die Datensynchronisation mit CentraSite finden Sie im Referenzhandbuch <i>API-Integration mit Drittanbieterkomponenten</i> unter <i>Konfigurieren der Interoperabilität mit CentraSite</i> .
Alfabet Standardimporte	Alfabet_CentraSite_Organization_Synchronization	Wird die Schnittstelle zur Datensynchronisation mit CentraSite verwendet, kann über dieses vordefinierte ADIF-Schema ein standardisierter Batch-Import von Informationen über gemeinsame Organisationen von Alfabet und CentraSite durchgeführt werden. Dieses Schema	Weitere Informationen über die Datensynchronisation mit CentraSite finden Sie im Referenzhandbuch <i>API-Integration mit Drittanbieterkomponenten</i> unter <i>Konfigurieren der Interoperabilität mit CentraSite</i> .

Gruppe	ADIF-Importschema	Bereitgestellte Funktionen	Weitere Informationen hierzu finden Sie unter...
		<p>ist ein geschütztes Schema und kann nicht benutzerdefiniert bearbeitet werden.</p>	
<p>Alfabet Standardimporte</p>	<p>Alfabet_CentraSite_User_Synchronisation</p>	<p>Wird die Schnittstelle zur Datensynchronisation mit CentraSite verwendet, kann über dieses vordefinierte ADIF-Schema ein standardisierter Batch-Import von Informationen über gemeinsame Benutzer von Alfabet und CentraSite durchgeführt werden. Dieses Schema ist ein geschütztes Schema und kann nicht benutzerdefiniert bearbeitet werden.</p>	<p>Weitere Informationen über die Datensynchronisierung mit CentraSite finden Sie im Referenzhandbuch <i>API-Integration mit Drittanbieterkomponenten</i> unter <i>Konfigurieren der Interoperabilität mit CentraSite</i>.</p>
<p>Alfabet Standard Technopedia Interoperability Schemes</p>	<p>ALFABET_TECHNOPEDIA_UPDATE</p>	<p>Anbieterprodukte/Komponenten sowie deren Hersteller können auf Basis der Daten in Technopedia®, einer kategorisierten Sammlung von Informationen über Unternehmenshardware und -software, aktualisiert werden. Ein Update kann durch Ausführen des Standard-ADIF-Importschemas durchgeführt werden. Das Importschema ist nicht anpassbar. Das Importschema liest die im XML-Objekt TechnopediaConfig festgelegte Konfiguration aus.</p> <p> Der Verweis eines Anbieterprodukts zu einem ICT-Objekt wird während der Ausführung der für die Alfabet Technopedia-Interoperabilität verfügbaren ADIF-Standardschemata aktualisiert, wenn sowohl das Anbieterprodukt als auch das ICT-Objekt von Technopedia stammen und die mit dem Anbieterprodukt verbundene Technopedia-Produktversion bzw. das Produkt-Release auf ein Technopedia-Produkt verweisen, das sich von dem Technopedia-Produkt</p>	<p>Weitere Informationen zur Technopedia-Interoperabilität finden Sie unter <i>Konfigurieren der Interoperabilität mit Technopedia</i> im Referenzhandbuch <i>API-Integration mit Drittanbieterkomponenten</i>.</p>

Gruppe	ADIF-Importschema	Bereitgestellte Funktionen	Weitere Informationen hierzu finden Sie unter...
		<p>unterscheidet, welches vom ICT-Objekt referenziert wird.</p>	
<p>Alfabet Standardimporte</p>	<p>Alfabet_APIGateway_Synchronization</p>	<p>APIs in webMethods API Gateway können in Alfabet über ein ADIF-Importschema importiert werden. Technische Services können nicht über ein ADIF-Schema in das webMethods API-Gateway exportiert werden. Wenn der Import von APIs über ADIF im XML-Objekt APIGatewayConfig angegeben ist, muss das vordefinierte ADIF-Schema Alfabet_APIGateway_Synchronization ausgeführt werden, um den Import auszulösen. Die Daten vom webMethods API-Gateway werden in temporäre Tabellen in ADIF importiert. Eine weitere Konfiguration ist erforderlich, um die Alfabet-Datenbank -Tabellen mit den Daten der temporären Datenbanktabellen zu aktualisieren.</p>	<p>Weitere Informationen zur Konfiguration, die zum Importieren und Zuordnen von APIs aus dem webMethods® API Gateway in Alfabet erforderlich sind, finden Sie unter <i>Konfiguration der Interoperabilität mit webMethods API Gateway</i> im Referenzhandbuch <i>API-Integration mit Drittanbieterkomponenten</i>.</p>
<p>Alfabet Standardimporte</p>	<p>Alfabet_APIPortal_Synchronization</p>	<p>APIs in webMethods API Portal können in Alfabet über ein ADIF-Importschema importiert werden. Technische Services können nicht über ein ADIF-Schema in das webMethods API-Gateway exportiert werden. Wenn der Import von APIs über ADIF im XML-Objekt APIPortalConfig angegeben ist, muss das vordefinierte ADIF-Schema Alfabet_APIPortal_Synchronization ausgeführt werden, um den Import auszulösen. Die Daten vom webMethods API Portal werden in temporäre Tabellen in ADIF importiert. Eine weitere Konfiguration ist erforderlich, um die Alfabet-Datenbank -Tabellen mit den Daten der temporären Datenbanktabellen zu aktualisieren.</p>	<p>Weitere Informationen zur Konfiguration, die zum Importieren und Zuordnen von APIs aus dem webMethods® API Portal in Alfabet erforderlich sind, finden Sie unter <i>Konfiguration der Interoperabilität mit webMethods API Portal</i> im Referenzhandbuch <i>API-Integration mit Drittanbieterkomponenten</i>.</p>

Gruppe	ADIF-Importschema	Bereitgestellte Funktionen	Weitere Informationen hierzu finden Sie unter...
<p>Automatisierte Übersetzungen der Alfabet-Instanz</p>	<p><code>Get_Instance_Automated_Translations_For_Empty_Texts</code></p>	<p>Dieses private ADIF-Schema ruft automatisierte Übersetzungen für zulässige Objektklasseneigenschaften ab, die einen Wert in der primären Sprache aufweisen und in einer der sekundären Sprachen leer sind. Dieses ADIF-Schema kann für Objektklassen ausgeführt werden, für die das Attribut Datenübersetzung aktivieren auf <code>True</code> und mindestens eine Eigenschaft im Attribut Eigenschaft der automatisierten Übersetzung auf <code>True</code> gesetzt ist.</p>	<p>Weitere Informationen zur Funktion „Automatisierte Datenübersetzungen“ sowie zur erforderlichen Konfiguration für die Ausführung automatisierter Übersetzungen finden Sie im Abschnitt <i>Konfigurieren der Übersetzung von Objektdaten</i> des Kapitels <i>Lokalisierung und Mehrsprachenunterstützung für die Alfabet -Benutzeroberfläche</i> im Referenzhandbuch <i>Konfigurieren von Alfabet mit Alfabet Expand</i>.</p>
<p>Automatisierte Übersetzungen der Alfabet-Instanz</p>	<p><code>GetAutomatedTranslations_For_Instance</code></p>	<p>Dieses private ADIF-Schema ruft automatisierte Übersetzungen für zulässige Eigenschaften der Objektklasse unter Wiederverwendung vorhandener Zeichenfolgen ab, falls möglich. Dieses ADIF-Schema kann mit einer Objektreferenzzeichenfolge parametrisiert werden. Dieses ADIF-Schema kann für Objektklassen ausgeführt werden, für die das Attribut Datenübersetzung aktivieren auf <code>True</code> und mindestens eine Eigenschaft im Attribut Eigenschaft der automatisierten Übersetzung auf <code>True</code> gesetzt ist.</p>	<p>Weitere Informationen zur Funktion „Automatisierte Datenübersetzungen“ sowie zur erforderlichen Konfiguration für die Ausführung automatisierter Übersetzungen finden Sie im Abschnitt <i>Konfigurieren der Übersetzung von Objektdaten</i> des Kapitels <i>Lokalisierung und Mehrsprachenunterstützung für die Alfabet -Benutzeroberfläche</i> im Referenzhandbuch <i>Konfigurieren von Alfabet mit Alfabet Expand</i>.</p>
<p>Automatisierte Übersetzungen der Alfabet-Instanz</p>	<p><code>GetAutoTranslationsForMM</code></p>	<p>Dieses private ADIF-Schema lädt automatisierte Übersetzungen in die Übersetzungstabellen hoch.</p> <p>Beachten Sie, dass zum Aktualisieren der Übersetzungen in Berichten und Berichtsordnern die im Root-Knoten der Registerkarte Berichte verfügbare Funktion Übersetzung aktualisieren ausgelöst werden muss. Um die Übersetzungen in Workflows zu aktualisieren, muss die Funktion Workflowübersetzung aktualisieren im Menü Globalisierung ausgelöst werden.</p>	<p>Weitere Informationen zur Funktion „Automatisierte Datenübersetzungen“ sowie zur erforderlichen Konfiguration für die Ausführung automatisierter Übersetzungen finden Sie im Abschnitt <i>Konfigurieren der Übersetzung von Objektdaten</i> des Kapitels <i>Lokalisierung und Mehrsprachenunterstützung für die Alfabet -Benutzeroberfläche</i> im Referenzhandbuch <i>Konfigurieren von Alfabet mit Alfabet Expand</i>.</p>

Gruppe	ADIF-Importschema	Bereitgestellte Funktionen	Weitere Informationen hierzu finden Sie unter...
Automatisierte Übersetzungen der Alfabet-Instanz	Import_Instance_Validated_Auto_Texts	Dieses geschützte ADIF-Schema ist die Ergänzung zum ADIF-Auftrag <code>Export_Instance_Automated_Texts</code> und dient zum Hochladen übersetzter Zeichenfolgen, die überprüft und korrigiert wurden. Dieses ADIF-Schema kann für Objektklassen ausgeführt werden, für die das Attribut Datenübersetzung aktivieren auf <code>True</code> und mindestens eine Eigenschaft im Attribut Eigenschaft der automatisierten Übersetzung auf <code>True</code> gesetzt ist.	Weitere Informationen zur Funktion „Automatisierte Datenübersetzungen“ sowie zur erforderlichen Konfiguration für die Ausführung automatisierter Übersetzungen finden Sie im Abschnitt <i>Konfigurieren der Übersetzung von Objektdaten</i> des Kapitels <i>Lokalisierung und Mehrsprachunterstützung für die Alfabet -Benutzeroberfläche</i> im Referenzhandbuch <i>Konfigurieren von Alfabet mit Alfabet Expand</i> .
Automatisierte Übersetzungen der Alfabet-Instanz	Renew_Instance_Automated_Translations_Automated_Texts	Dieses geschützte ADIF-Schema aktualisiert automatisierte Übersetzungen für zulässige Eigenschaften der Objektklasse unter Wiederverwendung vorhandener Zeichenfolgen, falls möglich. Dieses ADIF-Schema kann für Objektklassen ausgeführt werden, für die das Attribut Datenübersetzung aktivieren auf <code>True</code> und mindestens eine Eigenschaft im Attribut Eigenschaft der automatisierten Übersetzung auf <code>True</code> gesetzt ist.	Weitere Informationen zur Funktion „Automatisierte Datenübersetzungen“ sowie zur erforderlichen Konfiguration für die Ausführung automatisierter Übersetzungen finden Sie im Abschnitt <i>Konfigurieren der Übersetzung von Objektdaten</i> des Kapitels <i>Lokalisierung und Mehrsprachunterstützung für die Alfabet -Benutzeroberfläche</i> im Referenzhandbuch <i>Konfigurieren von Alfabet mit Alfabet Expand</i> .
Automatisierte Übersetzungen der Alfabet-Instanz	Rescan_Instance_Translations_To_Tmp	Dieses geschützte ADIF-Schema extrahiert vorhandene manuelle Übersetzungen in die interne Übersetzungstabelle <code>ALFA_INSTANCE_VOCABULARY</code> , die zum Abgleich neu gespeicherter Texte mit vorhandenen Übersetzungen verwendet wird. Dadurch wird die Wiederverwendung vorhandener Übersetzungen ermöglicht. Den Kunden wird empfohlen, diesen ADIF-Auftrag auszuführen, wenn sie mit der Verwendung der Funktion "Automatisierte Datenübersetzungen" beginnen. Dieses ADIF-Schema kann für Objektklassen ausgeführt werden, für die das Attribut Datenübersetzung aktivieren auf <code>True</code> und mindestens	Weitere Informationen zur Funktion „Automatisierte Datenübersetzungen“ sowie zur erforderlichen Konfiguration für die Ausführung automatisierter Übersetzungen finden Sie im Abschnitt <i>Konfigurieren der Übersetzung von Objektdaten</i> des Kapitels <i>Lokalisierung und Mehrsprachunterstützung für die Alfabet -Benutzeroberfläche</i> im Referenzhandbuch <i>Konfigurieren von Alfabet mit Alfabet Expand</i> .

Gruppe	ADIF-Importschema	Bereitgestellte Funktionen	Weitere Informationen hierzu finden Sie unter...
		eine Eigenschaft im Attribut Eigenschaft der automatisierten Übersetzung auf <code>True</code> gesetzt ist.	

Folgende Informationen sind verfügbar:

- [Importschema für die automatische Neuordnung von Kostenstellenkosten](#)
 - [Anwendungsfall](#)
 - [Beschreibung des Schemas "CC_Reallocate"](#)
 - [Anpassen des Schemas an die aktuelle Umgebung vor der Verwendung](#)
 - [Auswählen einer Teilmenge der verfügbaren Kostenstellen zur Aktualisierung](#)
 - [Berücksichtigen von anderen Währungs-IDs als "Aktuell"](#)

Importschema für die automatische Neuordnung von Kostenstellenkosten

Schematyp: Importschema

Name: CC_Reallocate

Supported Alfabet Functionality: Cost Management

Bereitgestellte Funktionen: Neuordnung von Kosten zu den Kostenstellen zugeordneten Objekten, nachdem sich die Kostenzuordnung zu Kostenstellen geändert hat.

Anwendungsfall

Die **Kostenmanagement** -Funktion unterstützt Sie bei der Verwaltung von Kosten auf Basis von Kostenstellen in Alfabet. Die Kostenstelle ist ein Mittel zur zentralen Definition von Kosten sowie für die Zuordnung der Kosten zu einer Gruppe von Architekturobjekten nach einem festgelegten Zuordnungsschema. In einer Kostenstelle werden die Kosten der Kostenstelle für verschiedene Jahre und Kostenarten, die Objekte, denen die Kosten zugeordnet werden, und die Art und Weise, in der die Zuordnung der Kosten zu den Objekten erfolgt, definiert.

Wenn die Kosten, die der Kostenstelle zugeordnet sind, sich auf der Ansichtseite **Akkumulierte Kosten** der Kostenstelle ändern, wird die Kostenzuordnung zu den Objekten automatisch aktualisiert. Wenn der Kostenstelle neue Objekte zugeordnet werden oder die Kostenverteilungsverhältnisse auf der Ansichtseite **Objekte** der Kostenstelle manuell definiert werden, muss die Kostenzuordnung manuell aktualisiert werden. Die manuelle Aktualisierung erfolgt durch Klicken auf die Schaltfläche **Kosten neu zuordnen** auf der Ansichtseite **Objekte**.



Kostenstellen müssen zunächst in der Funktionalität **Kostenstellen** in Alfabet konfiguriert werden. Weitere Informationen zu Kostenstellen finden Sie unter *Konfigurieren von Kostenstellen für Kostenmanagement-Funktionen* im Referenzhandbuch *Konfigurieren von Bewertungen und Referenzdaten in Alfabet*.

ADIF kann dazu verwendet werden, die Budgetkosten und die Objektzuweisung einer Kostenstelle sowie die Kostenzuordnungsverhältnisse als Batch-Prozess zu aktualisieren. Bei allen Importen über ADIF wird die Kostenzuordnung nicht automatisch durchgeführt, sondern muss manuell aktualisiert werden. Es gibt zwei Möglichkeiten, dies zu tun:

- Navigieren Sie in der Alfabet -Benutzeroberfläche zur Ansichtseite **Objekte** jeder Kostenstelle, und klicken Sie auf die Schaltfläche **Kosten neu zuordnen**.

oder

- Führen Sie das vorkonfigurierte ADIF-Schema "CC_Reallocate" über die AIDF-Konsolenapplikation aus, um alle Kostenzuordnungen für alle Kostenstellen als Batch-Prozess zu aktualisieren. Informationen zum Ausführen eines ADIF-Schemaauftrags finden Sie unter *Starten von Datenimport, Datenexport oder Datenänderung mit ADIF*.

Beschreibung des Schemas "CC_Reallocate"

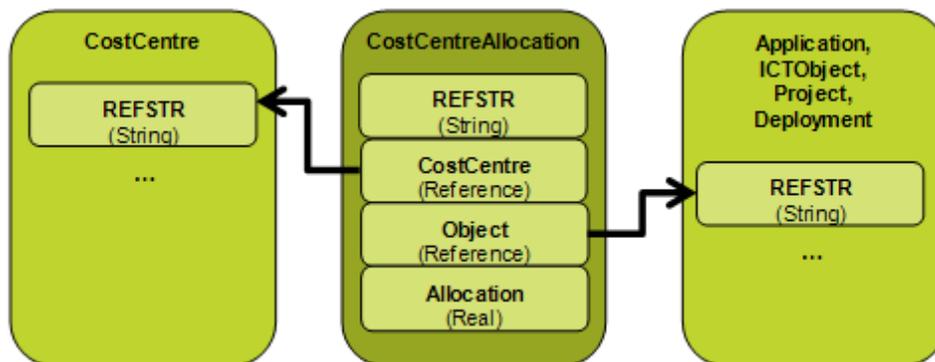
In diesem Abschnitt wird die Verarbeitung mit dem Schema "CC_Reallocate" beschrieben. Diese Informationen sind zur Ausführung des Schemas nicht erforderlich. Sie können diese Informationen lesen, wenn Sie kostenstellenbezogene Daten über ADIF importieren möchten und noch nicht mit den zugrunde liegenden Datenbankobjektstrukturen vertraut sind.

Es ist wichtig zu verstehen, wie Änderungen an der Konfiguration einer Kostenstelle in Alfabet verarbeitet werden, um das Schema **CC_Reallocate** nachvollziehen zu können.

Folgende Alfabet Objektklassen werden in den Prozess eingebunden:

Name	Titel	Beschreibung
CostCentre	Kostenstelle	Darstellung eines Kostenstellenobjekts.
CostCentreAllocation	Kostenstellenzuordnung	Unterstützende Klasse, die Informationen zu den Objektzuordnungen zu Kostenstellen speichert.
BudgetValue	Budgetwert	Unterstützende Klasse, die Informationen zu den Kostenwerten in den Kostenmanagement-Funktionen speichert.

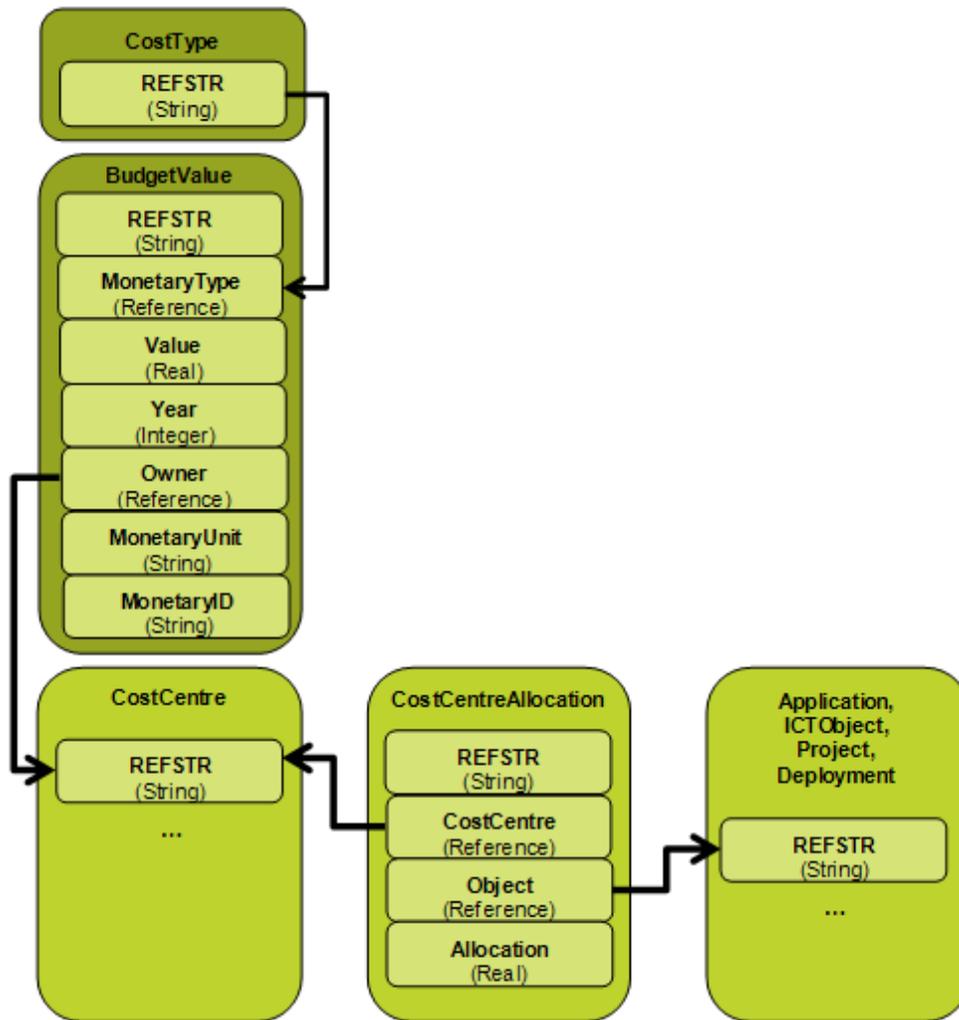
Wenn ein Objekt der Kostenstelle auf der Ansichtseite **Objekte** dieser Kostenstelle zugeordnet wird, wird eine neue Instanz der Objektklasse `CostCentreAllocation` erstellt, die sowohl auf die Kostenstelle als auch auf das Objekt verweist. Zusätzlich speichert sie den Prozentsatz der Kostenstellenkosten, die einem Objekt zugeordnet werden sollen, wenn die Kostenzuordnung (ein Prozentsatz) manuell definiert wird.

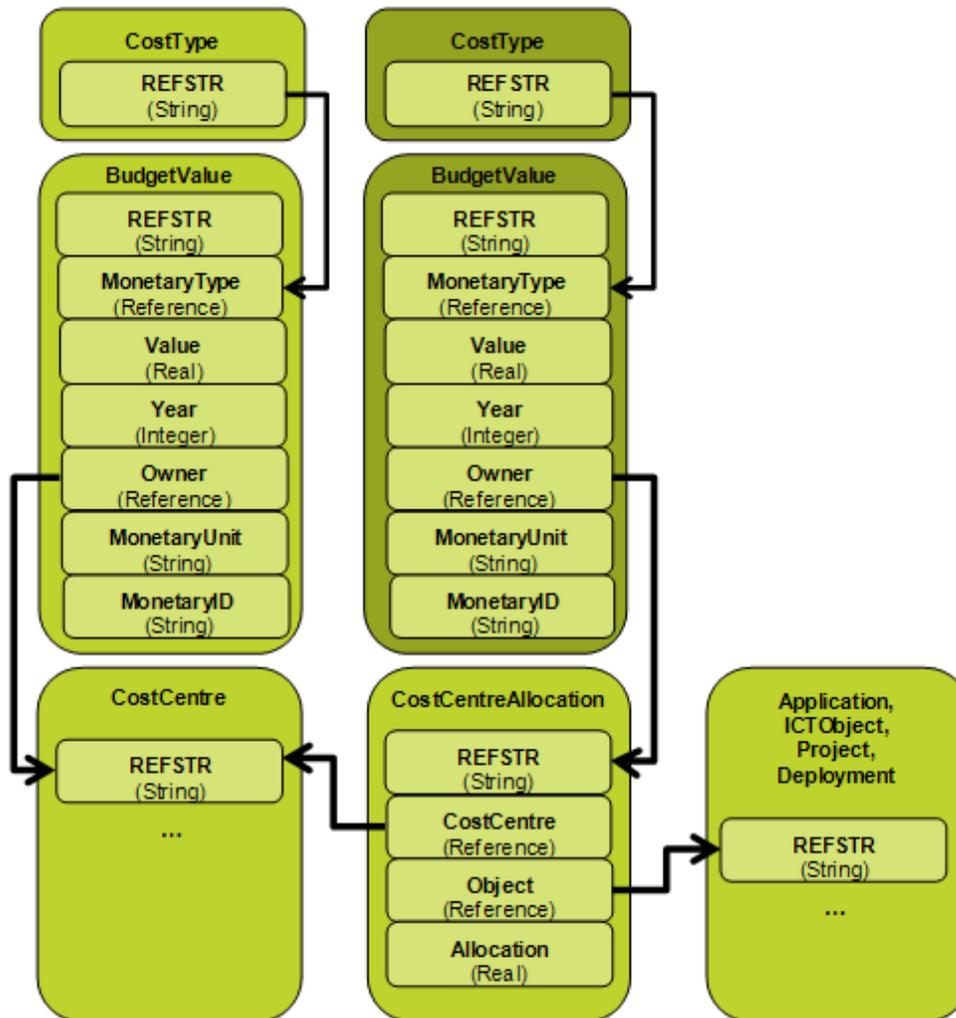




Beachten Sie Folgendes:

- Kostenstellen werden in der Funktionalität **Kostenstellen** in Alfabet erzeugt. Weitere Informationen zu Kostenstellen finden Sie unter *Konfigurieren von Kostenstellen für Kostenmanagement-Funktionen* im Referenzhandbuch *Konfigurieren von Bewertungen und Referenzdaten in Alfabet*.
- Die Bearbeitbarkeit der Kostenstellen wird im XML-Objekt **CostManagerDef** konfiguriert. Informationen über diese Konfigurationsanforderungen für Kostenstellen finden Sie im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* unter *Konfigurieren der Bearbeitbarkeit von Kosten in Kostenstellen*.
- Währungen werden in der Funktionalität **Referenzdatendefinition** in Alfabet konfiguriert. Weitere Informationen über das Konfigurieren von Währungen finden Sie im Referenzhandbuch *Konfigurieren von Bewertungen und Referenzdaten in Alfabet* im Abschnitt *Konfigurieren von Währungen und Währungswechselkursen für Kostenmanagementfunktionen*.

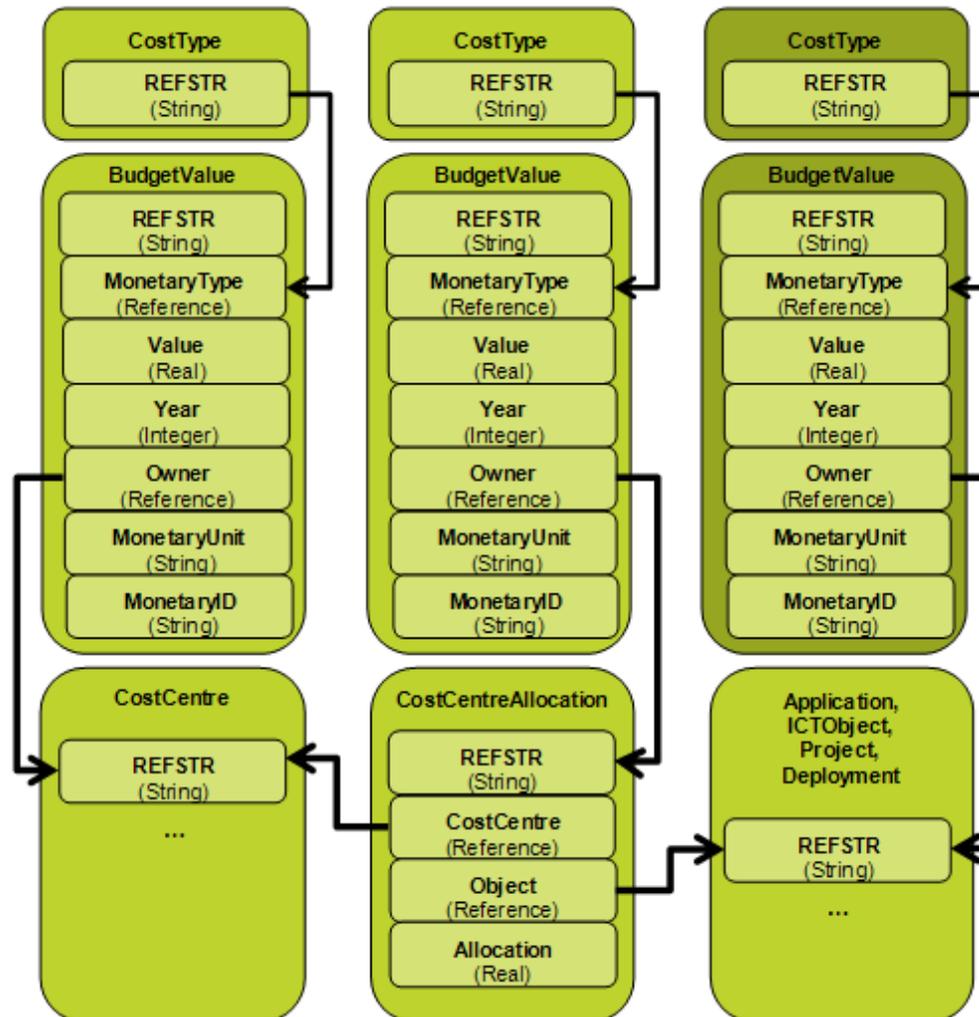




In einem zweiten Schritt wird das dem Objekt zugeordnete Budget aktualisiert. Das Budget des Objekts ist auch in Instanzen der Objektklasse `BudgetValue` definiert.

Wenn ein Objekt nur einer Kostenstelle zugeordnet ist, sind die Budgetwert-Instanzen, die dem Objekt zugeordnet sind, identisch mit den Budgetwert-Instanzen der `CostCentreAllocation`, die das Objekt referenzieren, mit der Ausnahme, dass der `Owner` das Objekt anstelle der `CostCentreAllocation` ist.

Wenn ein Objekt mehreren Kostenstellen zugeordnet ist, sind die Budgetwert-Instanzen des Objekts die Summe aller Budgetwerte, deren Besitzer eine `CostCentreAllocation` mit einer Referenz zum Objekt und einer identischen Spezifikation für Jahr, Kostenart und Währungs-ID ist.





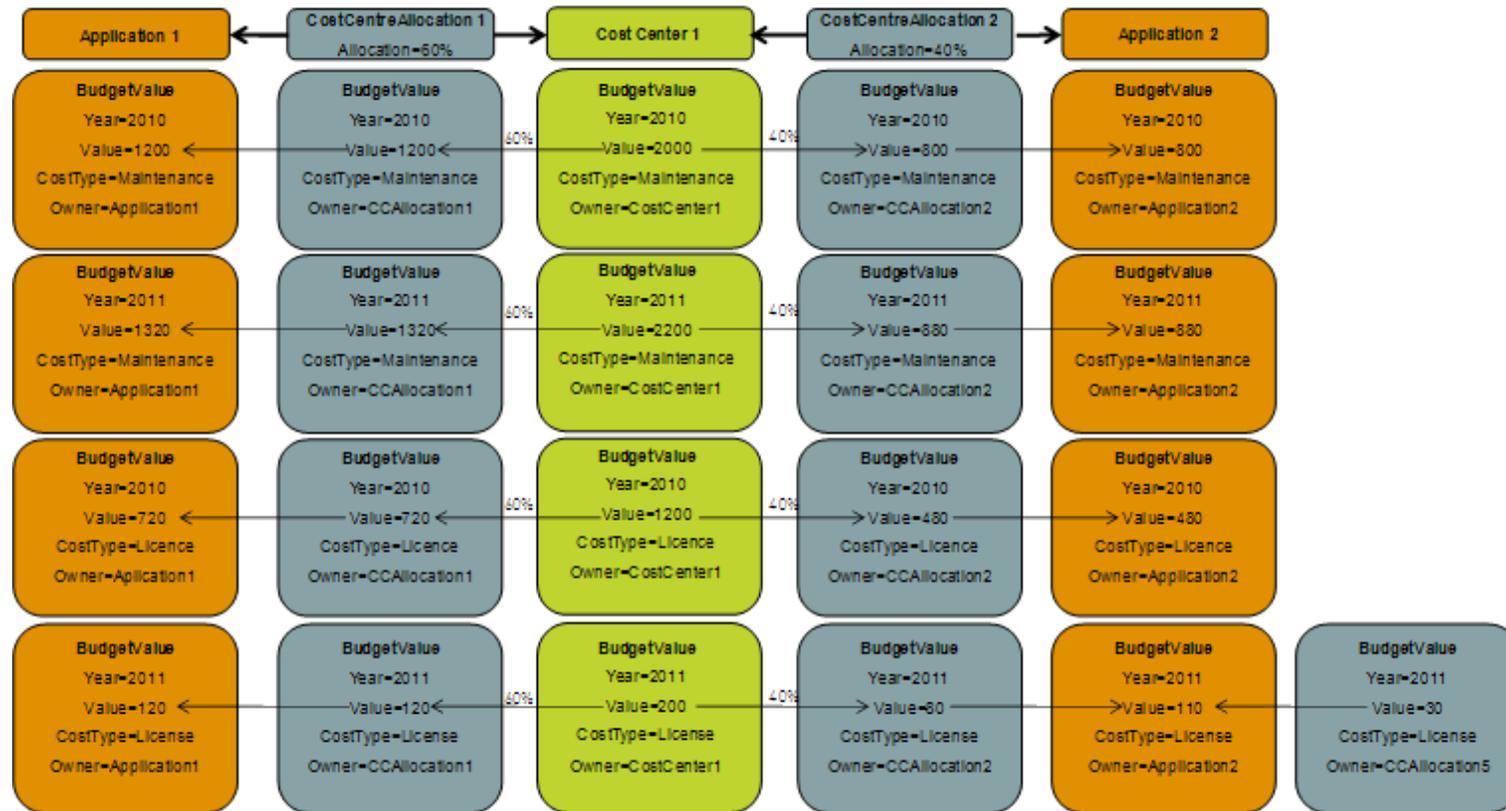
Zum Beispiel sind einer Kostenstelle zwei Objekte zugeordnet: Applikation 1 und Applikation 2. Die Kostenzuordnung wird manuell definiert. 60 % der Kosten werden Applikation 1 und 40 % der Kosten Applikation 2 zugeordnet. Diese Information wird in zwei `CostCentreAllocation`-Instanzen in der Alfabet-Datenbank gespeichert.

Die Kosten werden für die Jahre 2010 und 2011 und für die Kostenarten Wartung und Lizenz definiert. Die Währungs-ID und die Währungseinheit sind für alle Kostendefinitionen identisch und nicht im Beispiel enthalten, um es einfach zu halten.

Die Budget-Spezifikation in der Kostenstelle führt zur Generierung von vier Instanzen der Objektklasse `BudgetValue`, eine für jede Kombination aus Jahr und Kostenart.

Bei der Kostenzuordnung werden `BudgetValue`-Instanzen für die beiden `CostCentreAllocation`-Instanzen der Kostenstelle erstellt. Jeder `CostCentreAllocation` ist dieselbe Anzahl von `BudgetValue`-Instanzen zugeordnet wie der Kostenstelle. Die `BudgetValue`-Instanzen sind mit den `BudgetValue`-Instanzen der Kostenstelle identisch, mit der Ausnahme, dass der Besitzer die `CostCentreAllocation` ist und der Wert der Prozentsatz des Kostenwerts für die Kostenstelle ist, der durch das Kostenzuordnungsverhältnis definiert wird.

Für jedes Objekt, das der Kostenstellenzuordnung zugeordnet ist, werden dann dieselben `BudgetValue`-Instanzen definiert, bei denen das Objekt als Besitzer festgelegt ist. Applikation 1 werden nur Kosten von Kostenstelle 1 zugeordnet. Deshalb sind die Wert-Einstellungen der `BudgetValue`-Instanzen des Objekts mit den Wert-Einstellungen der `BudgetValue`-Instanzen der `CostCentreAllocation` 1 identisch. Die Lizenzgebühren für Applikation 2 werden über zwei unterschiedliche Kostenstellen zugeordnet. Deshalb ist der Wert, der im `BudgetValue` für Applikation 1 definiert ist, die Summe der `BudgetValue`-Instanzen von Kostenstelle 1 und der anderen Kostenstelle.



Das ADIF-Schema CC_Reallocate führt die Neuzuordnung von Kosten durch, nachdem die Konfiguration von Kostenstellen in einem Prozess aus fünf – jeweils durch einen Importeintrag ausgelöst – Schritten über ADIF geändert wurde. Die aktuellen Informationen zu verfügbaren Kostenstellen sowie CostCentreAllocation-Instanzen und BudgetValue-Instanzen, die auf die Kostenstellen verweisen, werden aus der Alfabet-Datenbank übernommen und in temporären Tabellen verarbeitet.

Nach Ausführung des ADIF-Schemas CC_Reallocate werden die Informationen in der BudgetValue-Tabelle in der Alfabet-Datenbank aktualisiert. Die Aktualisierung beschränkt sich auf BudgetValue-Instanzen der CostCentreAllocation-Instanzen und Objekte, die über Kostenzuordnungsprozesse erzeugt werden.

Die Tabelle bietet eine Übersicht über die Reihenfolge der Aktionen, die bei Ausführung des Schemas "CC_Reallocate" durchgeführt werden.

Importeintrag	Durchzuführende Aktion
<p>Aktualisierte Kostenstellen auswählen</p>	<p>Informationen zu allen <code>CostCentre</code>-Instanzen in der Alfabet-Datenbank werden in eine temporäre Tabelle <code>CC_Temp</code> geschrieben.</p> <p>HINWEIS: Standardmäßig werden Daten zu allen verfügbaren Kostenstellen in die temporäre Tabelle geschrieben. Sie können den Importeintrag ändern, sodass nur eine Teilmenge der verfügbaren Kostenstellen berücksichtigt wird. Weitere Informationen hierzu finden Sie unter Auswählen einer Teilmenge der verfügbaren Kostenstellen zur Aktualisierung.</p>
<p>Zuordnungsprozentsätze für alle Kostenstellenzuordnungs-Objekte aktualisieren</p>	<p>Informationen zu allen <code>CostCentreAllocation</code>-Instanzen in der Alfabet-Datenbank, die für die Kostenstellen in der <code>CC_Temp</code>-Tabelle relevant sind, werden in eine temporäre Tabelle <code>CCA_Temp</code> geschrieben, zusammen mit den Informationen zu der Gesamtzahl an Objekten, die einer Kostenstelle zugeordnet sind. Das Zuordnungsverhältnis wird entweder aus der Eigenschaft <code>Allocation</code> der <code>CostCentreAllocation</code>-Instanzen übernommen oder aus der Gesamtzahl an Objekten berechnet, wenn gleiche Zuordnung aktiviert ist.</p>
<p>Verwandte Währungstypen abrufen</p>	<p>Es wird eine temporäre Tabelle <code>CCOBJ_MTYPE_TMP</code> erzeugt, in der die Kostenarten aufgeführt sind, die für die Kostenstellen relevant sind. Diese Funktion ist im fünften Schritt erforderlich, um zu identifizieren, welche Teile der Kostenplanung von Applikationen, Projekten, Installationen, ICT-Objekten oder Service-Produkten über Kostenstellenzuordnung definiert werden.</p>
<p>Kostenstellenzuordnungs-Budgetwerte aktualisieren</p>	<p>Auf Grundlage der Informationen in den zuvor erzeugten temporären Tabellen und der <code>BudgetValue</code>-Informationen in der Alfabet-Datenbank werden die <code>BudgetValue</code>-Instanzen für alle aktualisierten <code>CostCentreAllocation</code>-Instanzen in der Tabelle <code>CCA_Temp</code> berechnet und in eine temporäre Tabelle <code>MV_CCA_Temp</code> geschrieben.</p> <p>Die Daten in den temporären Tabelle werden dann verwendet, um die Daten in der <code>BudgetValue</code>-Tabelle der Alfabet-Datenbank zu aktualisieren. Neue Instanzen werden hinzugefügt, vorhandene Instanzen werden aktualisiert oder, wenn sie obsolet sind, gelöscht.</p>
<p>Objekt-Budgetwerte aktualisieren</p>	<p>Basierend auf den Informationen in den vorher erzeugten temporären Tabellen werden die <code>BudgetValue</code>-Instanzen für alle Objekte, die in der Tabelle <code>CC_Temp</code> einer oder mehreren Kostenstellen zugeordnet sind, berechnet und in eine temporäre Tabelle <code>MV_Temp</code> geschrieben.</p>

Importeintrag	Durchzuführende Aktion
	<p>Die Daten in der temporären Tabelle werden dann verwendet, um die Daten in der <code>BudgetValue</code>-Tabelle der Alfabet-Datenbank zu aktualisieren. Neue Instanzen werden hinzugefügt, vorhandene Instanzen werden aktualisiert oder, wenn sie obsolet sind, gelöscht.</p> <p>Mitteilung: Die Berechnung von Budgetkostenwerten ist auf Budgetwertinstanzen beschränkt, die über eine <code>MonetaryID</code> verfügen, die auf <code>Current</code> gesetzt ist, da in der Standardkonfiguration von Alfabet nur aktuelle Kosten über Kostenstellen behandelt werden. Wurde die Konfiguration der Bearbeitbarkeit des Kostendefinitionstyps im XML-Objekt <code>CostManagerDef</code> geändert, muss das <code>CC_Reallocate</code>-Schema entsprechend angepasst werden. Informationen hierzu finden Sie unter Berücksichtigen von anderen Währungs-IDs als "Aktuell".</p>

Anpassen des Schemas an die aktuelle Umgebung vor der Verwendung

Die Standard-Aktualisierung muss an Ihre Anforderungen angepasst werden. Sie müssen folgenden Teil des Prozesses ändern:

- [Auswählen einer Teilmenge der verfügbaren Kostenstellen zur Aktualisierung](#)

Optional können Sie den folgenden Teil des Prozesses ändern:

- [Berücksichtigen von anderen Währungs-IDs als "Aktuell"](#)

Auswählen einer Teilmenge der verfügbaren Kostenstellen zur Aktualisierung

Die Abfrage im Schema, die die in die temporäre Datenbanktabelle zu schreibenden Kostenstellendaten sucht, enthält eine `WHERE`-Klausel, um nur für eine Teilmenge der verfügbaren Kostenstellen die Kosten neu zuzuordnen. Das Beispiel der `WHERE`-Klausel schränkt die Aktualisierung auf Kostenstellen ein, die mit ‚AAA‘ anfangen. Dies unterdrückt die Kosten-Neuzuordnung für beliebige Kostenstellen in den meisten Datenbanken. Vor der Verwendung des Schemas **`CC_Reallocate`** müssen Sie deshalb die Abfrage ändern, um Kostenstellen zu suchen, die Ihren Anforderungen entsprechen.



Wenn die Budgetwerte für Objekte durch mehr als eine Kostenstelle beeinflusst werden, empfehlen wir Ihnen, alle Kostenstellendaten zu aktualisieren, ohne die Aktualisierung auf eine Teilmenge der aktualisierten Kostenstellen einzuschränken. So ist sichergestellt, dass alle relevanten Aktualisierungen durchgeführt werden.

So definieren Sie die Kostenzuordnung über das Schema **CC_Reallocate** für einen definierten Satz von Kostenstellen in der Alfabet-Datenbank:

- 1) Gehen Sie im ADIF-Explorer von Alfabet Expand zu **CC_Reallocate** > **Select Updated Cost Centers** > **SQL-Befehle - OnComplete** > **Insert Updated cost centers**.
- 2) Klicken Sie im Attributfenster des SQL-Befehls **Insert Updated cost centers** im Attributfeld **Text** auf die Schaltfläche **Durchsuchen** . Die Abfrage des SQL-Befehls wird in einem Texteditor angezeigt.
- 3) Ändern Sie die **WHERE**-Klausel in der Abfrage, um zu definieren, welche Kostenstellen in den Kostenstellenzuordnungsprozess aufgenommen werden sollen, oder löschen Sie die **WHERE**-Klausel, um alle Kostenstellen zu aktualisieren.



Das folgende Beispiel zeigt die Abfrage mit einer **WHERE**-Klausel an, die die Kosten-Neuzuordnung auf Kostenstellen beschränkt, die Eigentum von „Corporate IT“ der Organisation sind. Bitte beachten Sie, dass die Definition der Bedingung es erfordert, dass zwei Objekte von der Abfrage gefunden werden, und daher die **SELECT**-Klausel geändert werden muss, um Klassenspezifikationen einzuschließen. Diese Klassenspezifikationen müssen dann aus den resultierenden Spaltennamen entfernt werden, indem ein Alias für den Spaltennamen definiert wird.

```
INSERT CC_TMP
SELECT cc.REFSTR, cc.NAME As 'NAME', cc.EQUALALLOCATION As 'EQUALALLOCATION'
FROM COSTCENTRE cc, ORGAUNIT org
WHERE cc.OWNER = org.REFSTR
      AND org.NAME = 'Corporate IT'
```

- 4) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern.

Berücksichtigen von anderen Währungs-IDs als "Aktuell"

Die Eigenschaft `MonetaryID` der Klasse `BudgetValue` definiert, ob der Budgetwert für eine Kostenanforderung, budgetierte Kosten, aktuelle Kosten, Fixkosten oder prognostizierte Kosten definiert ist. Standardmäßig ist Alfabet so konfiguriert, dass nur aktuelle Kosten über Kostenstellen verwaltet werden, und das **CC_Reallocate**-Schema behandelt daher nur `BudgetValue`-Instanzen, die Objekten zugeordnet sind, die über eine `MonetaryID` verfügen, die auf `Current` gesetzt ist.

Alfabet kann so konfiguriert werden, dass andere Kostendefinitionstypen über Kostenstellen verwaltet werden. Die Konfiguration erfolgt im XML-Objekt **CostManagerDef** im Konfigurationstool Alfabet Expand.



Weitere Informationen zum Konfigurieren des XML-Objekts **CostManagerDef** finden Sie unter *Konfigurieren von Kostenmanagement-Funktionalitäten* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Während der Großteil des **CC_Reallocate**-Schemas BudgetValue-Instanzen behandelt, die für eine von der MonetaryTypeID unabhängige Kostenstelle definiert sind, verwendet die Abfrage, die die Alfabet-Datenbank nach obsoleteren BudgetValue-Instanzen durchsucht, die MonetaryID, um zu definieren, ob eine BudgetValue-Instanz, die Eigentum eines Objekts ist, aufgrund einer Kostenzuordnung der Kostenstelle erzeugt wurde.

So beziehen Sie andere Kostendefinitionsarten in die Aktualisierung von Budgetwerten mit dem Schema **CC_Reallocate** ein:

- 1) Gehen Sie im ADIF-Explorer von Alfabet Expand zu **CC_Reallocate > Update Object's Budget Values > SQL-Befehle - OnComplete > Delete unused objects**.
- 2) Klicken Sie im Attributfenster des SQL-Befehls **Update changed budget values** im Attributfeld **Text** auf die Schaltfläche **Durchsuchen** . Die Abfrage des SQL-Befehls wird in einem Texteditor angezeigt.
- 3) Ändern Sie die WHERE-Klausel, die die Datenverarbeitung auf BudgetValue-Instanzen beschränkt, bei denen eine MonetaryID auf Current gesetzt ist, um die Datenverarbeitung auf diejenigen MonetaryIDs einzuschränken, die Sie einbeziehen möchten.



Im folgenden Beispiel wird die Abfrage mit einer WHERE-Klausel angezeigt, die die MonetaryIDs Current und Budget umfasst:

```
DELETE FROM BUDGETVALUE
WHERE (REFSTR IN
      (SELECT BV.REFSTR
      FROM BUDGETVALUE AS BV
      INNER JOIN ON BV.MONETARYTYPE = CCOBJ_MTYPE_TMP.MONETARYTYPE AND
      BV.OWNER = CCOBJ_MTYPE_TMP.OBJECT
      WHERE (BV.REFSTR NOT IN
            (SELECT DISTINCT REFSTR
            FROM MV_TMP))
```

```
AND  
( BV.MONETARYCODEID = 'Current'  
OR BV.MONETARYCODEID = 'Budget' )))
```

- 4) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern.