



API-Integration mit Drittanbieterkomponenten

Alfabet-Referenzhandbuch

Dokumentationsversion Alfabet 10.9.0

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:	Konfigurieren von Integrationslösungen	6
Kapitel 2:	Allgemeine Konfigurationen mit Gültigkeit für mehrere Integrationslösungen	9
	Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen	9
	Konfigurieren der Verwendung von selbstsignierten Zertifikaten für Integrationslösungen	10
Kapitel 3:	Konfigurieren der Interoperabilität mit Skype for Business Server®	11
Kapitel 4:	Konfigurieren der Interoperabilität mit Technopedia	14
	Beschreibung der Zuordnung von Technopedia-Produkten zu Anbieterprodukten in Alfabet	16
	Konfigurieren des XML-Objekts „TechnopediaConfig“	22
	Bereitstellen der Technopedia-Funktionalität für die Anwender-Community	28
	Aktualisieren von Technopedia-Produkten in Alfabet über das ADIF-Importschema ALFABET_TECHNOPEDIA_UPDATE	31
Kapitel 5:	Konfigurieren der Interoperabilität mit CentraSite	35
	Konfigurieren des Klassenmodells für Interoperabilität mit CentraSite	36
	Konfigurieren von Verbindungen für Interoperabilität mit CentraSite	40
	Konfigurieren der Anzeige von CentraSite-Services in alfabet	44
	Konfigurieren der Aktualisierung von alfabet-Daten in CentraSite	48
	Importieren von CentraSite-Daten über ADIF-Schemata	52
Kapitel 6:	Konfiguration der Interoperabilität mit webMethods API Portal	54
	Übersicht über die für webMethods API Portal erforderliche Konfiguration	55
	Konfigurieren des Klassenmodells für Interoperabilität mit webMethods API Portal	57
	Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Portal	58
	Konfiguration der Zuordnung von API-Portal-Ressourcen zu Businessdaten	61
Kapitel 7:	Konfiguration der Interoperabilität mit webMethods API Gateway	66
	Übersicht über die für webMethods API Gateway erforderliche Konfiguration	67
	Konfigurieren des Alfabet -Klassenmodells für Interoperabilität mit webMethods API Gateway	69
	Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Gateway	70
	Konfiguration der Zuordnung von API-Gateway-Ressourcen zu Businessdaten	73
Kapitel 8:	Konfigurieren der Interoperabilität mit den Apigee API-Plattform-Tools von Google	77
	Übersicht über die für die Interoperabilität mit Apigee erforderliche Konfiguration	78
	Konfigurieren des Klassenmodells für Interoperabilität mit Apigee	80
	Konfigurieren von Verbindungen und API-Proxy-Zuordnungen für die Apigee-Integration	81
	Senden von Anfragen über einen Proxyserver an Apigee	84
	Erzeugen von Apigee-Datenverbindungen	85
	Erzeugen eines ADIF-Importschemas für den Import aus Apigee	87
Kapitel 9:	Konfigurieren der Erstellung bzw. des Exports von technischen Services auf Grundlage von WSDL- und OpenAPI-Spezifikationsdateien	91
Kapitel 10:	Konfigurieren der Integration der Daten von Amazon Web Services	94
	Konfigurieren der Verbindung mit Amazon Web Services	94
	Senden von Anforderungen über Proxyserver zu Amazon Web Services	96
	Konfigurieren der Integration von Amazon Web Services-Daten in die Alfabet-Datenbank	97
Kapitel 11:	Konfigurieren der Integration von Daten zwischen ServiceNow und Alfabet	101

Konfigurieren der Integration von Daten von ServiceNow	101
Konfigurieren der Datenübertragung von ServiceNow zu Alfabet	102
Konfigurieren der Integration von ServiceNow-Daten in die Alfabet-Datenbank	108
Ändern einer vorhandenen Konfiguration für die ServiceNow-Integration	112
Konfigurieren der Integration von Alfabet -Daten in ServiceNow	113
Definieren von zu exportierenden Daten aus der Alfabet-Datenbank über einen konfigurierten Bericht	115
Konfigurieren der Datenübertragung von Alfabet zu ServiceNow	118
Konfigurieren des ADIF-Exportschemas für den Datenexport in ServiceNow	123
Ändern einer vorhandenen Konfiguration für die ServiceNow-Integration	126
Kapitel 12: Konfigurieren der Integration mit JIRA	127
JIRA-Daten in Alfabet importieren	129
Konfigurieren von Verbindungen für den Import von JIRA-Daten in Alfabet	133
Importieren von JIRA-Daten in Alfabet mithilfe von ADIF-Schemata	135
Alfabet -Daten nach JIRA exportieren	142
Konfigurieren von Verbindungen und Zuordnungen für den Datenexport von Alfabet nach JIRA	149
Konfigurieren der primären und sekundären Berichte	153
Konfigurieren von Objektfilterberichten	158
Konfigurieren des ADIF-Exportassistenten für den Export von Alfabet-Daten nach JIRA	159
Konfigurieren der Ereignisvorlagen zum Auslösen der ADIF-Exportschemata für die Synchronisierung	163
Konfigurieren von semantischen Verbindungen zum Verknüpfen und Synchronisieren von Jira-Projekten mit Alfabet-Objekten	164
Erstellen einer JIRA-Verbindung für die projektbasierte Integration	169
Erstellen einer JIRA-Verbindung für die architekturbasierte Integration	173
Verknüpfung und Synchronisierung des JIRA-Projekts	177
Kapitel 13: Konfiguration der Interoperabilität mit Microsoft Project	180
Konfigurieren der Verbindungen für die Interoperabilität mit Microsoft Project	181
Angabe eines Assistenten für die Interoperabilität mit Microsoft Project	185
Angabe eines Release-Status für die Interoperabilität mit Microsoft Project	185
Kapitel 14: Konfigurieren der Interoperabilität mit einem Übersetzungsdienst	187
Index	189

Kapitel 1: Konfigurieren von Integrationslösungen

Mehrere in Alfabet verfügbare Schnittstellen bieten eine Interoperabilität oder Integration mit einer Vielzahl von Produkten und Services. Zur Implementierung einer Integrationslösung muss die Schnittstelle konfiguriert werden. Die grundlegende Konfiguration wird vorwiegend in XML-Objekten ausgeführt, die in Alfabet Expand verfügbar sind. Ein Großteil der Schnittstelle hängt auch von der Konfiguration und Ausführung von ADIF-Schemata für den Datenaustausch zwischen dem externen Service/Produkt und der Alfabet-Datenbank ab.



Vor dem Import werden die Daten für alle Integrationslösungen überprüft, die die Ausführung eines ADIF-Schemas erfordern. Zeichen, die nicht XML-konform sind, werden vor dem Import entfernt, um Probleme bei der Datenspeicherung in der Alfabet-Datenbank zu verhindern.

Folgende Informationen sind verfügbar:

- [Konfigurieren von Integrationslösungen](#)
- [Allgemeine Konfigurationen mit Gültigkeit für mehrere Integrationslösungen](#)
 - [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#)
 - [Konfigurieren der Verwendung von selbstsignierten Zertifikaten für Integrationslösungen](#)
 - [Aktivieren der Interoperabilität mit einer Service-Registry über das XML-Objekt „ServiceRegistryManager“](#)
- [Konfigurieren der Interoperabilität mit Skype for Business Server®](#)
- [Konfigurieren der Interoperabilität mit Technopedia](#)
 - [Beschreibung der Zuordnung von Technopedia-Produkten zu Anbieterprodukten in Alfabet](#)
 - [Konfigurieren des XML-Objekts „TechnopediaConfig“](#)
 - [Bereitstellen der Technopedia-Funktionalität für die Anwender-Community](#)
 - [Aktualisieren von Technopedia-Produkten in Alfabet über das ADIF-Importschema ALFABET_TECHNOPEDIA_UPDATE](#)
- [Konfigurieren der Interoperabilität mit CentraSite](#)
 - [Konfigurieren des Klassenmodells für Interoperabilität mit CentraSite](#)
 - [Konfigurieren von Verbindungen für Interoperabilität mit CentraSite](#)
 - [Konfigurieren der Anzeige von CentraSite-Services in alfabet](#)
 - [Konfigurieren der Aktualisierung von alfabet-Daten in CentraSite](#)
 - [Importieren von CentraSite-Daten über ADIF-Schemata](#)
- [Konfiguration der Interoperabilität mit webMethods API Portal](#)
 - [Übersicht über die für webMethods API Portal erforderliche Konfiguration](#)
 - [Konfigurieren des Klassenmodells für Interoperabilität mit webMethods API Portal](#)

- [Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Portal](#)
- [Konfiguration der Zuordnung von API-Portal-Ressourcen zu Businessdaten](#)
- [Konfiguration der Interoperabilität mit webMethods API Gateway](#)
 - [Übersicht über die für webMethods API Gateway erforderliche Konfiguration](#)
 - [Konfigurieren des Alfabet -Klassenmodells für Interoperabilität mit webMethods API Gateway](#)
 - [Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Gateway](#)
 - [Konfiguration der Zuordnung von API-Gateway-Ressourcen zu Businessdaten](#)
- [Konfigurieren der Interoperabilität mit den Apigee API-Plattform-Tools von Google](#)
 - [Übersicht über die für die Interoperabilität mit Apigee erforderliche Konfiguration](#)
 - [Konfigurieren des Klassenmodells für Interoperabilität mit Apigee](#)
 - [Konfigurieren von Verbindungen und API-Proxy-Zuordnungen für die Apigee-Integration](#)
 - [Senden von Anfragen über einen Proxyserver an Apigee](#)
 - [Erzeugen von Apigee-Datenverbindungen](#)
 - [Erzeugen eines ADIF-Importschemas für den Import aus Apigee](#)
- [Konfigurieren der Erstellung bzw. des Exports von technischen Services auf Grundlage von WSDL- und OpenAPI-Spezifikationsdateien](#)
- [Konfigurieren der Integration der Daten von Amazon Web Services](#)
 - [Konfigurieren der Verbindung mit Amazon Web Services](#)
 - [Senden von Anforderungen über Proxyserver zu Amazon Web Services](#)
 - [Konfigurieren der Integration von Amazon Web Services-Daten in die Alfabet-Datenbank](#)
- [Konfigurieren der Integration von Daten zwischen ServiceNow und Alfabet](#)
 - [Konfigurieren der Integration von Daten von ServiceNow](#)
 - [Konfigurieren der Datenübertragung von ServiceNow zu Alfabet](#)
 - [Konfigurieren der Integration von ServiceNow-Daten in die Alfabet-Datenbank](#)
 - [Ändern einer vorhandenen Konfiguration für die ServiceNow-Integration](#)
 - [Konfigurieren der Integration von Alfabet -Daten in ServiceNow](#)
 - [Definieren von zu exportierenden Daten aus der Alfabet-Datenbank über einen konfigurierten Bericht](#)
 - [Konfigurieren der Datenübertragung von Alfabet zu ServiceNow](#)
 - [Konfigurieren des ADIF-Exportschemas für den Datenexport in ServiceNow](#)
 - [Ändern einer vorhandenen Konfiguration für die ServiceNow-Integration](#)
- [Konfigurieren der Integration mit JIRA](#)

- [JIRA-Daten in Alfabet importieren](#)
 - [Konfigurieren von Verbindungen für den Import von JIRA-Daten in Alfabet](#)
 - [Importieren von JIRA-Daten in Alfabet mithilfe von ADIF-Schemata](#)
- [Alfabet -Daten nach JIRA exportieren](#)
 - [Konfigurieren von Verbindungen und Zuordnungen für den Datenexport von Alfabet nach JIRA](#)
 - [Konfigurieren der primären und sekundären Berichte](#)
 - [Konfigurieren von Objektfilterberichten](#)
 - [Konfigurieren des ADIF-Exportassistenten für den Export von Alfabet-Daten nach JIRA](#)
 - [Konfigurieren der Ereignisvorlagen zum Auslösen der ADIF-Exportschemata für die Synchronisierung](#)
- [Konfigurieren von semantischen Verbindungen zum Verknüpfen und Synchronisieren von Jira-Projekten mit Alfabet-Objekten](#)
 - [Erstellen einer JIRA-Verbindung für die projektbasierte Integration](#)
 - [Erstellen einer JIRA-Verbindung für die architekturbasierte Integration](#)
- [Verknüpfung und Synchronisierung des JIRA-Projekts](#)
- [Konfiguration der Interoperabilität mit Microsoft Project](#)
 - [Konfigurieren der Verbindungen für die Interoperabilität mit Microsoft Project](#)
 - [Angaben eines Assistenten für die Interoperabilität mit Microsoft Project](#)
 - [Angaben eines Release-Status für die Interoperabilität mit Microsoft Project](#)
- [Konfigurieren der Interoperabilität mit einem Übersetzungsdienst](#)

Kapitel 2: Allgemeine Konfigurationen mit Gültigkeit für mehrere Integrationslösungen

Die in diesem Abschnitt beschriebenen Funktionen gelten für mehrere oder alle Integrationslösungen.

Bei folgenden Konfigurationen handelt es sich um allgemeine Konfigurationen:

- [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#)
- [Konfigurieren der Verwendung von selbstsignierten Zertifikaten für Integrationslösungen](#)
- [Aktivieren der Interoperabilität mit einer Service-Registry über das XML-Objekt „ServiceRegistryManager“](#)

Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen

Durch die Definition von Servervariablen können einige XML-Attribute in den XML-Objekten im Ordner **Integrationslösungen** in Alfabet Expand in der Server-Alias-Konfiguration definiert werden. Durch das Definieren in der Konfiguration des Server-Alias statt direkt im XML-Objekt wird die Ausbreitung von Änderungen vereinfacht.

Ein Wert für ein XML-Attribut kann entweder im XML-Element als String oder über eine Servervariable definiert werden. Es ist nicht möglich, einen Wert mit einer Servervariablen zu definieren, die als Teil eines String geschrieben wurde. Der vollständige Wert des XML-Elements im XML-Objekt muss durch eine Servervariable ersetzt werden. In einem XML-Objekt muss die Servervariablendefinition durch ein Gleichheitszeichen, gefolgt vom Wert, angegeben werden, um dem XML-Attribut in doppelten Anführungszeichen zugeordnet zu werden. Beispiel: `<XML attribute>="$$SQLSERVER"`. Die Servervariable wird im relevanten XML-Attribut folgendermaßen referenziert: `$$<Servervariablenname>`. Die Servervariable `SQLSERVER` würde im XML-Objekt z. B. als `$$SQLSERVER` referenziert werden.

Die Servervariablen werden im Alfabet Administrator definiert. Definieren einer Servervariablen zur Verwendung in XML-Objekten:

- 1) Klicken Sie im Alfabet Administrator im Explorer **Administrator** auf den Knoten **Alfabet-Aliase**.
- 2) Wählen Sie in der Tabelle rechts den Server-Alias aus, für den Sie eine Servervariable definieren möchten, und klicken Sie auf die Schaltfläche **Bearbeiten** . Der Aliaseditor wird angezeigt.
- 3) Wechseln Sie auf die Registerkarte **Variablen**, und klicken Sie auf die Schaltfläche **Neu** Ein Dialogfeld wird angezeigt.
- 4) Geben Sie für die Servervariable einen eindeutigen Namen in das Feld **Variablenname** ein.



Der Name der Servervariable darf nur Buchstaben des englischen Alphabets, Zahlen und Unterstriche enthalten.

- 5) Geben Sie den Wert, der für die Servervariable verwendet werden soll, in das Feld **Wert der Variablen** ein.



Wenn der Variablenwert ein Sonderzeichen gemäß XML-Standards enthält (z. B.: &, %, ;,<,>), müssen diese Zeichen durch ihren jeweiligen XML-kompatiblen Code ersetzt werden (z. B. `&` für &)

- 6) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern. Die Servervariablendefinition erscheint in der Liste der Servervariablen.



Um die Servervariable zu bearbeiten oder zu löschen, wählen Sie sie in der Tabelle aus und klicken auf die Schaltfläche **Bearbeiten** bzw. **Löschen** unterhalb der Tabelle.

- 7) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern und den Editor zu schließen. Die Definition der Servervariablen ist nun in der Konfiguration des Server-Alias verfügbar und kann in relevanten XML-Objekten, die im Ordner **Integrationslösungen** in Alfabet Expand verfügbar sind, genutzt werden.

Konfigurieren der Verwendung von selbstsignierten Zertifikaten für Integrationslösungen

Für Integrationslösungen auf der Basis von Web Services können selbstsignierte Zertifikate für HTTPS-Verbindungen verwendet werden.

Hierfür ist folgende Konfiguration erforderlich:

- 1) Kopieren Sie die selbstsignierten Zertifikate vom Web Service des Drittanbieters in einen lokalen Ordner, für den die Alfabet-Webapplikation über Zugriffsberechtigungen verfügt.
- 2) Öffnen Sie den Alfabet Administrator.
- 3) Klicken Sie im Explorer auf **Alfabet-Aliase**. Im rechten Fenster wird eine Liste aller verfügbaren Alias-Konfigurationen angezeigt.
- 4) Wählen Sie in der Tabelle die Alias-Konfiguration der Alfabet-Webapplikation aus.
- 5) Klicken Sie in der Symbolleiste auf die Schaltfläche **Bearbeiten** . Der Editor zur Bearbeitung der Alias-Konfiguration wird angezeigt.
- 6) Öffnen Sie die Registerkarte **Server-Einstellungen > Sicherheit**.
- 7) Geben Sie im Feld **Pfad für selbstsignierte öffentliche Zertifikatsdateien** den Pfad zum Ordner ein, der die selbstsignierten Zertifikate enthält.
- 8) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern.

Kapitel 3: Konfigurieren der Interoperabilität mit Skype for Business Server®

Die Integration mit dem Skype for Business-Server® ist verfügbar, um Skype-Nachrichten und Audio- und Videoanrufe zwischen Anwendern in der Alfabet -Anwendergemeinschaft zu unterstützen. Die Verbindung kann serverbasiert oder browserbasiert sein.

Wenn für Ihr Unternehmen die Interoperabilität mit Skype for Business Server® konfiguriert und Skype für einen Anwender zulässig ist, werden neben dem Namen des Anwenders im Bereich **Attribute** der Objektprofile/Objekt-Cockpits ein Skype-Statussymbol und eine Vorschau über dessen aktuellen Skype-Status angezeigt. Der Anwender kann auf das Skype-Statussymbol klicken, um den Skype-Bildschirm zu öffnen und sich bei Fragen an den autorisierten Anwender eines Objekts zu wenden. Informationen über die Verwendung der Skype-Funktion finden Sie im Abschnitt *Skypen mit Ihren Kollegen* im Referenzhandbuch *Erste Schritte mit Alfabet*.



Die Interoperabilität mit Skype ist nur für Anwender verfügbar, für die ein gültiger Skype-Name und eine gültige Skype-ID angegeben wurden. Dies kann ein Anwender über die Option **Persönliche Informationen** im Menü < **Alfabet-Anwender** > in der Alfabet -Benutzeroberfläche oder über die Attribute **Skype-Name** und **Skype-ID** im Editor **Anwender** in der Funktionalität **Anwenderverwaltung** definieren. Informationen hierzu finden Sie im Abschnitt *Erzeugen eines Anwenders* im Referenzhandbuch *Anwender- und Lösungsverwaltung*.

Beachten Sie, dass der Kontaktstatus aktiviert werden muss, damit er dem Anwender im Abschnitt „Kontaktliste“ des Editors „Skype for Business – Optionen“ in der Applikation Skype for Business angezeigt wird. Wenn der Kontaktstatus in der Applikation Skype for Business nicht aktiviert ist, wird der Skype-Anwesenheitsstatus für den Anwender in Alfabet nicht angezeigt.

Mit dem XML-Objekt **AlfaSkypeIntegrationConfig** können Sie die Skype for Business Server®-Funktion aktivieren und einen Hauptanwender so konfigurieren, dass er den Verfügbarkeitsstatus von Skype-Anwendern abfragen kann. Wenn der Server nicht mit einer direkten Verbindung zum Skype-Server eingerichtet ist, werden die Skype-Berechtigungen des Master-Anwenders verwendet, um den Verfügbarkeitsstatus anderer Anwender abzufragen.

So konfigurieren Sie das XML-Objekt **AlfaSkypeIntegrationConfig**:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte** > **IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **AlfaSkypeIntegrationConfig**, und wählen Sie **XML bearbeiten....** Das XML-Objekt **AlfaSkypeIntegrationConfig** wird geöffnet.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist über das Attribut **XML-Vorlage** im Attributraster des XML-Objekts **AlfaSkypeIntegrationConfig** eine Vorlage verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. In diesem Fall bearbeiten Sie die nachfolgend beschriebenen XML-Elemente. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf.

- 3) Es gibt zwei verschiedene Methoden, um die Interoperabilität mit Skype festzulegen.
 - Mit dem XML-Element `SkypeInfo` kann eine serverbasierte Verbindung mit Skype for Business Service hergestellt werden. Für diese Methode muss ein administratives Anwenderkonto eingerichtet werden.

- Fügen Sie dem Root-XML-Element `AlfaSkypeIntegrationConfig` ein untergeordnetes XML-Element `SkypeInfo` hinzu.
 - `IsActive`: Geben Sie „true“ an, um die Interoperabilität mit dem Skype for Business Service zu aktivieren. Geben Sie „false“ an, um die Interoperabilität mit dem Skype for Business Service zu deaktivieren. Die Attribute **Skype-ID** und **Skype-Domäne** werden nur dann im Editor **Anwender** und in der Funktionalität **Anwenderverwaltung** angezeigt, wenn das XML-Attribut `IsActive` auf „true“ gesetzt ist.
 - `SkypeAdminUserId`: Geben Sie die Skype-Anwender-ID für einen administrativen Anwender an, der zur Kommunikation mit dem Skype-Server berechtigt ist, um den Skype-Status für die verschiedenen Anwender abzurufen, die Skype über die Alfabet -Benutzeroberfläche verwenden können.
 - `SkypeAdminUserPwd`: Geben Sie das Skype-Anwender-Passwort für den administrativen Anwender an.
 - `SkypeDomain`: Geben Sie die Domäne an, wenn sich der Server, auf dem Alfabet ausgeführt wird, außerhalb der Domäne für die Ausführung von Skype for Business Server befindet.
 - `RefreshSeconds`: Geben Sie die Anzahl der Sekunden für das Intervall an, in dem der Cache des Anwesenheitsstatus der Skype-Anwender aktualisiert werden soll. Die Standardeinstellung lautet „60“.
- Das XML-Element `SkypeWebSDK` ermöglicht den Aufbau einer browserbasierten Verbindung, die das Web SDK von Skype verwendet, um zur Laufzeit darauf zuzugreifen. Diese alternative Methode zum Herstellen einer Verbindung mit Skype wird nicht empfohlen, da der Anwender bei jeder über die Alfabet -Benutzeroberfläche initiierten Interaktion die Anmeldedaten für das Skype-Konto eingeben muss.
- Fügen Sie dem Root-XML-Element `AlfaSkypeIntegrationConfig` ein untergeordnetes XML-Element `SkypeWebSDK` hinzu.
 - `IsActive`: Geben Sie „true“ an, um die Interoperabilität mit dem Skype for Business Server zu aktivieren. Beachten Sie Folgendes:

Der Download der Skype SDK-Bibliothek ist auf Alfabet -Installationen mit aktivierter Skype-Integration beschränkt.

Die folgenden Bibliotheken werden heruntergeladen, wenn das XML-Attribut `IsActive` auf „true“ gesetzt ist. `//ajax.aspnetcdn.com/ajax/jquery/jquery-1.10.2.min.js` und `//swx.cdn.skype.com/shared/v/1.2.15/SkypeBootstrap.min.js`. Für die Dateiendungen gelten die Spezifikationen der Blacklist/Whitelist von Dateiendungen, die im XML-Objekt **FileExtensionLists** definiert sind.



Stellen Sie sicher, dass die Dateierweiterung `.js` nicht in der im XML-Objekt **FileExtensionLists** definierten Blacklist aufgeführt ist. Wenn eine Whitelist im XML-Objekt **FileExtensionLists** angegeben ist, muss die Dateierweiterung `.js` als zulässige Dateierweiterung aufgeführt sein. Informationen hierzu finden Sie im Abschnitt *Festlegen der zulässigen Dateinamenerweiterungen für das Hochladen/Herunterladen von Dateien* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Die Attribute **Skype-ID** und **Skype-Domäne** werden nur dann im Editor **Anwender** und in der Funktionalität **Anwenderverwaltung** angezeigt, wenn das XML-Attribut `IsActive` auf „true“ gesetzt ist.

- `RefreshSeconds`: Geben Sie die Anzahl der Sekunden für das Intervall an, in dem der Cache des Anwesenheitsstatus der Skype-Anwender aktualisiert werden soll. Die Standardeinstellung lautet „60“.

- 4) Klicken Sie in der Symbolleiste auf die Schaltfläche **Speichern**  , um die XML-Definition zu speichern.

Kapitel 4: Konfigurieren der Interoperabilität mit Technopedia

Software AG bietet eine Schnittstelle zu dem Technopedia®-Repository von Software- und Hardwareprodukten. Durch die Technopedia-Funktionalität können Anbieterprodukte standardisiert erfasst und die Technologiedaten im gesamten Unternehmen abgeglichen werden. Der Katalog der strukturierten Informationen über die IT-Infrastruktur gewährleistet eine vereinheitlichte Sprache und Disziplin, um das Technologieportfolio zu verwalten und zu planen. Mit Hilfe von konsistenter Namensgebung und standardisierten Daten kann die Komplexität der IT-Landschaft reduziert werden, und überlappende Technologien können eliminiert sowie vorhandene und bekannte Technologien können wiederverwendet werden. Sofern Interoperabilität mit Technopedia von Ihrem Unternehmen unterstützt wird, können Alfabet-Anwender Anbieterprodukte in Alfabet basierend auf Technopedia-Software- und/oder Hardwareprodukten erzeugen.

Technopedia® hat eine zweistufige Taxonomie aus Produktkategorien und Subkategorien, die die Technopedia-Produkte enthalten. Software AG ist nicht Eigentümer der Taxonomie von Technopedia®, und die Verwendung der Taxonomie von Technopedia® kann zu einer Struktur führen, die von der Technologiedomänenstruktur Ihres Unternehmens abweicht. In der Regel würde das gesamte Repository von Technopedia®-Produktkategorien in Alfabet importiert. Die vorhandenen, bereits in Alfabet definierten Anbieterproduktkategorien verbleiben in der Alfabet-Datenbank, werden jedoch im Technopedia-Selektor der Alfabet-Benutzeroberfläche, über den Anwender Technopedia-Hardware- und Softwareprodukte importieren können, nicht mehr angezeigt. Die oberste Ebene der Technopedia®-Produktkategorien wird in die oberste Ebene der Anbieterprodukt Hierarchie importiert, und die untergeordneten Produktkategorien werden unterhalb der übergeordneten Anbieterproduktkategorie angezeigt.

Den importierten Anbieterproduktkategorien sind zunächst keine Anbieterprodukte zugeordnet. Mit anderen Worten, die Anbieterprodukte müssen explizit basierend auf den Software-/Hardwareprodukten im Technopedia-Repository erzeugt werden. Die Software- und Hardwareprodukte im Technopedia®-Repository können kategorieweise ausgewählt werden, um Anbieterprodukte in Alfabet zu erzeugen. Die Anbieterprodukte, die basierend auf Technopedia-Produkten erzeugt wurden, werden in der Alfabet-Datenbank gespeichert und können bei Bedarf weiter definiert werden. Wenn das neue Anbieterprodukt erzeugt wurde, wird der Anbieter für dieses Technopedia-Produkt automatisch dem neuen Anbieterprodukt in Alfabet zugeordnet. Wenn der Anbieter noch nicht in der Alfabet-Datenbank existiert, wird er mit dem neuen Anbieterprodukt in Alfabet erzeugt. Wenn ein vorhandenes Anbieterprodukt bereits einem Technopedia-Software- oder Technopedia-Hardwareprodukt zugeordnet ist und der Anwender dasselbe Technopedia-Produkt auswählt und dessen Eigenschaft `id` mit der Eigenschaft `TP_ID` des vorhandenen Anbieterprodukts übereinstimmt, wird das vorhandene Anbieterprodukt aktualisiert und kein neues Anbieterprodukt erstellt.



Diese Dokumentation beschreibt in erster Linie die Erzeugung von Anbieterprodukten auf der Basis von Technopedia-Software- und Hardwareprodukten. Einige Unternehmen bevorzugen jedoch die Implementierung der Technopedia-Funktionalität, um Komponenten anstelle von Anbieterprodukten zu erzeugen. Dies hängt von der in Ihrem Unternehmen eingesetzten Methodik ab. Beachten Sie Folgendes hinsichtlich der Verwendung von Komponenten anstelle von Anbieterprodukten:

- Die Technopedia®-Produktkategorien werden Komponentenkategorien zugeordnet, und die Hardware- und Softwareprodukte werden Komponenten zugeordnet.
- Die Objektklasse `Component` muss im XML-Attribut `ClassMapping` des XML-Objekts **`TechnopediaConfig`** spezifiziert werden.
- Für jede Komponente, die in Alfabet auf Basis eines Technopedia-Software- oder Hardwareprodukts erzeugt wird, wird zunächst ein Objekt der Objektklasse `VendorProduct` und dann ein Objekt der Objektklasse `Component` erzeugt.

Informationen hierzu finden Sie im Abschnitt [Beschreibung der Zuordnung von Technopedia-Produkten zu Anbieterprodukten in Alfabet](#).

- Die entsprechenden Ansichtsseiten müssen der Anwendergemeinschaft so zur Verfügung gestellt werden, wie es im Abschnitt [Bereitstellen der Technopedia-Funktionalität für die Anwender-Community](#) beschrieben ist.
- Das vordefinierte ADIF-Importschema `ALFABET_TECHNOPEdia_UPDATE` ist nur für die Aktualisierung der Alfabet -Anbieterprodukte und ICT-Objekte relevant. Sie müssen ein ADIF-Importschema konfigurieren, um Alfabet -Komponenten mit den aktuellen Informationen in Technopedia zu synchronisieren.

Für ICT-Objekte, die auf Technopedia-Produkten basieren, können Sie ein Abonnementkonzept implementieren, um die Wartung des Technologiekatalogs des Unternehmens zu vereinfachen. Wenn das ICT-Objekt das Technopedia-Produkt abonniert hat, auf dem es basiert, werden Anbieterprodukte automatisch auf der Grundlage aller entsprechenden Release-Versionen des Technopedia-Produkts erzeugt und gewartet, auf denen das ICT-Objekt basiert. Um das Abonnementkonzept für ein ICT-Objekt zu implementieren, muss das Kontrollkästchen **Hat Abonnement** auf der Registerkarte **Technopedia** des Editors **ICT-Objekt** ausgewählt sein. Das Kontrollkästchen **Hat Abonnement** gibt an, dass das ICT-Objekt das Technopedia-Produkt abonniert hat, auf dem es basiert. Die Release-Versionsebene (**Alle Releases, Hauptversionen, Nebenversionen**) des zum Erzeugen der Anbieterprodukte verwendeten Technopedia-Produkts muss im Feld **Abonnement-Ebene** angegeben werden. Der ADIF-Prozess `ALFABET_TECHNOPEdia_UPDATE` muss von einem Anwender mit einem administrativen Anwenderprofil ausgeführt werden, um die Anbieterprodukte erzeugen zu können. Anbieterprodukte werden auf der Grundlage aller entsprechenden freigegebenen Versionen des Technopedia-Produkts erzeugt, auf denen das ICT-Objekt basiert. Die Anbieterprodukte werden den Ansichtsseiten der **Anbieterprodukte** hinzugefügt, wenn der ICT-Prozess `<Code>ALFABET_TECHNOPEdia_UPDATE</Code>` ausgeführt wird.

Anbieter können zudem in Alfabet auf der Grundlage von Technopedia-Herstellern erstellt werden.



Um auf das Technopedia-Repository zugreifen und Anbieterprodukte basierend auf den Software- und Hardwareprodukten, die in Technopedia gespeichert sind, erzeugen zu können, ist ein Abonnement der Technopedia-Services erforderlich. Darüber hinaus muss die Schnittstelle zwischen der Alfabet -Lösung und der Technopedia-API konfiguriert werden.

Mit dem XML-Objekt **TechnopediaConfig** können Sie die Technopedia®-Funktion aktivieren sowie die Anmeldedaten für den Technopedia-Service konfigurieren. Sie können auch Informationen über die Selektoren konfigurieren, die in Alfabet für die Suche und Anzeige der Technopedia-Produkte verwendet werden sollen, darüber, wie die Lebenszyklusinformationen der Technopedia-Produkte der Objektklasse `VendorProduct` zugeordnet werden sollen und welche Technopedia-Attribute den benutzerdefinierten Objektklasseneigenschaften in Alfabet zugeordnet werden sollen, die für die Objektklassen `VendorProduct`, `Component`, `ICT Object`, und `Vendor` konfiguriert wurden.

Die Daten werden von Technopedia über einen RESTful-Serviceaufruf angefordert, der eine `GET`-Anfrage an die folgenden Technopedia-APIs sendet:

- Für Software: `/api/v1/software_extended/`
- Für Hardware: `/api/v1/hardware_extended/`



Wenn Ihr Unternehmen plant, neue Anbieterprodukte auf Hardware- und Softwareprodukten von Technopedia aufzubauen, dann wird empfohlen, Objektklassenstereotypen für die Objektklasse `VendorProduct` zu konfigurieren. Weitere Informationen über das Konfigurieren von

Objektklassenstereotypen finden Sie unter *Konfigurieren von Objektklassenstereotypen für Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.



Um die auf Technopedia-Produkten basierenden Anbieterprodukte, Anbieterproduktkategorien, ICT-Objekte und Anbieter auf die aktuellen Daten im Technopedia-Repository zu aktualisieren, ist ein vordefiniertes ADIF-Importschema `ALFABET_TECHNOPEDIA_UPDATE` verfügbar. Beim Ausführen liest das ADIF-Importschema die im XML-Objekt **TechnopediaConfig** festgelegte Konfiguration aus. Weitere Informationen über die Aktualisierung von auf Technopedia-Produkten basierenden Anbieterprodukten und Anbietern sowie über die für die Aktualisierung zu erfüllenden Kriterien finden Sie im Abschnitt [Aktualisieren von Technopedia-Produkten in Alfabet über das ADIF-Importschema ALFABET_TECHNOPEDIA_UPDATE](#).

Folgende Informationen sind verfügbar:

- [Beschreibung der Zuordnung von Technopedia-Produkten zu Anbieterprodukten in Alfabet](#)
- [Konfigurieren des XML-Objekts „TechnopediaConfig“](#)
- [Bereitstellen der Technopedia-Funktionalität für die Anwender-Community](#)
- [Aktualisieren von Technopedia-Produkten in Alfabet über das ADIF-Importschema ALFABET_TECHNOPEDIA_UPDATE](#)

Beschreibung der Zuordnung von Technopedia-Produkten zu Anbieterprodukten in Alfabet

In diesem Abschnitt finden Sie ausführliche Informationen darüber, wie Technopedia-Klassen Alfabet -Objektklassen zugeordnet sind und wie die Eigenschaften dieser Technopedia-Klassen den standardmäßigen Objektklasseneigenschaften für die Klassen `VendorProduct`, `Component`, `ICT Object`, und `Vendor` zugeordnet werden. Benutzerdefinierte Eigenschaften werden so zugeordnet, wie es für die entsprechende Objektklasse im XML-Element `ClassMapping` des XML-Objekts **TechnopediaConfig** angegeben ist.

Zuordnung zwischen den Technopedia-Klassen und den Objektklassen von Alfabet:

Technopedia-Klasse	Alfabet-Objektklasse
<code>software_extended</code>	<code>VendorProduct</code> , <code>Component</code>
<code>hardware_extended</code>	<code>VendorProduct</code> , <code>Component</code>
<code>manufacturer</code>	<code>Vendor</code>
<code>taxonomy2012</code>	<code>VendorProductCategory</code> , <code>ComponentCategory</code> , <code>ICTObjectCategory</code>
<code>sw_product</code>	<code>ICTObject</code>

Technopedia-Klasse	Alfabet-Objektklasse
hw_product	ICTObject

Im Folgenden wird erläutert, wie die Eigenschaften dieser Technopedia-Klassen den Objektklasseneigenschaften für die Klassen `VendorProduct`, `Component`, `ICT Object`, und `Vendor` zugeordnet werden.



Die folgenden Informationen beschreiben lediglich die Zuordnungen zu den Alfabet -Objektklassen `VendorProduct`, `Component`, `ICT Object`, und `Vendor`. Beachten Sie Folgendes, falls Ihr Unternehmen Technopedia-Software von Hardware-Produkten der Objektklasse `Component` zuordnet:

- Für jede Komponente, die in Alfabet auf Basis eines Technopedia-Software- oder Hardwareprodukts erzeugt wird, wird zunächst ein Objekt der Objektklasse `VendorProduct` und dann ein Objekt der Objektklasse `Component` erzeugt. Mit anderen Worten, wann immer eine Komponente auf Basis eines Technopedia-Produkts erzeugt wird, werden der Alfabet-Datenbank zwei Objekte (ein Anbieterprodukt und eine Komponente) hinzugefügt.
- Das Anbieterprodukt ist das Basisobjekt für die neue Komponente, und die Komponente hat eine Referenz zu dem Anbieterprodukt. Die folgenden Objektklasseneigenschaften werden von dem Anbieterprodukt zu der Komponente kopiert: `Name`, `Version`, `StartDate`, `EndDate`, `Vendor`, und `TP_CATEGORY`. Wenn für das Anbieterprodukt die Objektklasseneigenschaft `TP_CATEGORY` angegeben wird, versucht das System, ein Komponentenkategorie-Objekt mit der gleichen `ID` zu finden und dieses als Komponentenkategorie der Komponente festzulegen.
- Das vordefinierte ADIF-Importschema `ALFABET_TECHNOPEDIA_UPDATE` aktualisiert keine Komponenten. Sie müssen ein ADIF-Importschema konfigurieren, um Komponenten in der Alfabet mit den aktuellen Informationen in Technopedia zu synchronisieren.



Bei der Zuordnung von Technopedia-Eigenschaften und Alfabet -Eigenschaften werden nur Eigenschaften des Typs `String` unterstützt.

Technopedia-Klasse	Technopedia-Attribut	Alfabet-Objektklasse	Alfabet-Objektklasseneigenschaft
software_extended		VendorProduct	
	id		TP_ID Beachten Sie, dass, wenn ein Klassenschlüssel die <code>TP_ID</code> -Eigenschaft für die Objektklasse <code>VendorProduct</code> angibt und Anbieterprodukte mit doppelten Werten für die Technopedia-IDs

Technopedia-Klasse	Technopedia-Attribut	Alfabet-Objektklasse	Alfabet-Objektklasseneigenschaft
			(Technopedia id-Attribut) importiert werden, der Klassenschlüssel verletzt wird. Wenn dies eintritt, wird der Fehler in die Protokolldatei geschrieben, und das doppelte Anbieterprodukt wird übersprungen und nicht in die Alfabet-Datenbank importiert.
	product_name		Name
	version		Version
	product_alias		Alias
	suit_name		Suite
	edition		Edition
	manufacturer		Vendor
	sub_category		TP_SubCategory
	cat_taxonomy2012_id		TP_CATEGORY
	general_availability_date		StartDate
	end_of_life_date und obsolete_date		<p>EndDate, wobei das Enddatum das Maximum von end_of_life_date oder von obsolete_date ist.</p> <p>Um Probleme bei der Anzeige der Lebenszyklusdiagramme in Alfabet zu vermeiden, werden die Datumsfelder leer gelassen, wenn die Technopedia-Produkte die Daten 1.1.1900 und 31.12.2999 aufweisen.</p>

Technopedia-Klasse	Technopedia-Attribut	Alfabet-Objektklasse	Alfabet-Objektklasseneigenschaft
hardware_extended		VendorProduct	
	id		TP_ID Bitte beachten Sie, dass Technopedia die id zusammensetzt aus model_id+product_id.
	product		Name
	model		Version
	manufacturer		Vendor Wenn das vordefinierte ADIF-Schema ALFABET_TECHNOPE-DIA_UPDATE ausgeführt wird, wird automatisch ein neuer Anbieter in der Alfabet-Datenbank für jedes Anbieterprodukt erzeugt, das auf Basis eines Hardwareprodukts von Technopedia aktualisiert wird, für welches kein Wert für die Manufacturer-Eigenschaft angegeben wurde.
	cat_taxonomy2012_id		TP_CATEGORY
	general_availability_date		StartDate
	last_availability_date und obsolete_date		EndDate, wobei das Enddatum das Maximum von last_availability_date oder von obsolete_date ist. Um Probleme bei der Anzeige der Lebenszyklusdiagramme in Alfabet zu vermeiden, werden die Datumsfelder leer gelassen, wenn die Technopedia-Produkte die Daten 1.1.1900 und 31.12.2999 aufweisen.

Technopedia-Klasse	Technopedia-Attribut	Alfabet-Objektklasse	Alfabet-Objektklasseneigenschaft
manufacturer		Vendor	
	manufacturer		Name
	symbol		ShortName
	description		Description
	country		Country
	city		City
	street		Street
	zip		Zip
	email		Email
	fax		Fax
	phone		Phone
	website		Website
sw_product		ICTObject	
	cat_sw_product_id		TP_ID
	product_name		Name
	manufacturer		Vendor

Technopedia-Klasse	Technopedia-Attribut	Alfabet-Objektklasse	Alfabet-Objektklasseneigenschaft
			Wenn das vordefinierte ADIF-Schema ALFABET_TECHNOPE-DIA_UPDATE ausgeführt wird, wird automatisch ein neuer Anbieter in der Alfabet-Datenbank für jedes Anbieterprodukt erzeugt, das auf Basis eines Hardwareprodukts von Technopedia aktualisiert wird, für welches kein Wert für die <code>Manufacturer</code> -Eigenschaft angegeben wurde.
	<code>cat_taxonomy2012_id</code>		TP_CATEGORY
	<code>create_date</code>		StartDate
<code>hw_product</code>		ICTObject	
	<code>cat_hw_product_id</code>		TP_ID
	<code>product_name</code>		Name
	<code>manufacturer</code>		Vendor Wenn das vordefinierte ADIF-Schema ALFABET_TECHNOPE-DIA_UPDATE ausgeführt wird, wird automatisch ein neuer Anbieter in der Alfabet-Datenbank für jedes Anbieterprodukt erzeugt, das auf Basis eines Hardwareprodukts von Technopedia aktualisiert wird, für welches kein Wert für die <code>Manufacturer</code> -Eigenschaft angegeben wurde.
	<code>cat_taxonomy2012_id</code>		TP_CATEGORY
	<code>create_date</code>		StartDate

Konfigurieren des XML-Objekts „TechnopediaConfig“

Mit dem XML-Objekt **TechnopediaConfig** können Sie die Technopedia®-Funktion aktivieren sowie die Anmeldedaten für den Technopedia-Service konfigurieren. Sie können außerdem die Selektoren angeben, die in Alfabet für die Suche und Anzeige der Technopedia-Produkte verwendet werden sollen, darüber, wie die Lebenszyklusinformationen der Technopedia-Produkte der Objektklasse `VendorProduct` (oder ihren Objektklassenstereotypen) zugeordnet werden sollen und welche Technopedia-Attribute den benutzerdefinierten Objektklasseneigenschaften in Alfabet zugeordnet werden sollen, die für die Objektklassen `VendorProduct` und `Vendor` konfiguriert wurden.



Beachten Sie, dass bei der Zuordnung von Eigenschaften zwischen Technopedia und Alfabet nur Eigenschaften des Typs `String` unterstützt werden.



Beachten Sie, dass im XML-Objekt **TechnopediaConfig** Servervariablen verwendet werden können, um den Wert des XML-Attributs zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation auszulesen, wenn eine Verbindung zu Technopedia hergestellt wird. Weitere Informationen zu Servervariablen finden Sie unter [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).

So bearbeiten Sie das XML-Objekt **TechnopediaConfig**:

- 1) Wechseln Sie zur Registerkarte **Präsentation**, und erweitern Sie den Ordner **XML-Objekte** sowie anschließend den Ordner **Integrationslösungen**.
- 2) Klicken Sie mit der rechten Maustaste auf **TechnopediaConfig**, und wählen Sie **XML bearbeiten...**. Das XML-Objekt **TechnopediaConfig** wird geöffnet.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist über das Attribut **XML-Vorlage** im Attributraster des XML-Objekts **TechnopediaConfig** eine Vorlage verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. In diesem Fall bearbeiten Sie die nachfolgend beschriebenen XML-Elemente. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf.

- 3) Definieren Sie unter dem Root-XML-Element `TechnopediaConfig` die folgenden XML-Attribute:
 - `active`: Geben Sie „true“ ein, um die Interoperabilität mit Technopedia zu aktivieren. Wenn das Attribut `active` auf „true“ gesetzt ist, sind in den entsprechenden Ansichten die Menüoptionen für den Import von Technopedia-Kategorien und das Erzeugen neuer Anbieterprodukte auf der Basis von Technopedia-Produkten verfügbar. Weitere Informationen darüber, welche Ansichten für die Technopedia-Funktionen relevant sind, finden Sie im Abschnitt [Bereitstellen der Technopedia-Funktionalität für die Anwender-Community](#).
 - `service`: Geben Sie Schema, Host und Anschluss der URL ein, die auf den Technopedia-Server verweist, der für den Abruf von Informationen von Technopedia als Endpunkt verwendet wird. Normalerweise ist dies <http://api.technopedia.com>.
 - `api`: Geben Sie den URL-Pfad für den Endpunkt an, von dem Informationen von Technopedia abgerufen werden sollen. Dies ist im Normalfall der Technopedia-Server, der als Endpunkt für den Abruf von Informationen von Technopedia verwendet wird. In der Regel ist dies: `/api/v1/`.
 - `authorization`: Geben Sie den String ein, der zur Autorisierung des RESTful API-Aufrufs an den Technopedia-Server verwendet wird. Sie müssen sich bei Technopedia einen

Benutzernamen und einen Schlüssel besorgen. Der String muss wie folgt aussehen: `apikey
xxxlicence:0123456890abcdef`.

- `data_portion`: Geben Sie eine Ganzzahl zwischen 10-1000 ein, die als Parameter in RESTful API-Aufrufen zur Beschränkung der Anzahl an zurückgegebenen Datensätzen verwendet wird. Der empfohlene Wert ist „50“. Dabei ist zu beachten, dass eine übermäßig große Zahl die Leistung beeinträchtigt.
- `search_limit`: Geben Sie eine Ganzzahl zwischen 10 und 1000 ein, um die maximale Zahl an Datensätzen festzulegen, die vom Selektor zurückgegeben wird, der für die Suche nach Technopedia-Produkten in der Alfabet -Benutzeroberfläche verwendet wird. Der empfohlene Wert ist „300“. Dabei ist zu beachten, dass eine übermäßig große Zahl die Leistung beeinträchtigt.
- `category_source`: Geben Sie an, ob die Produktkategorien im Feld **Technopedia-(Unter-)Kategorie** im Selektor mit Technopedia-Kategorien aus der Alfabet-Datenbank oder über einen separaten Serviceaufruf mit den Technopedia-Kategorien im Technopedia-Repository gefüllt werden. Geben Sie entweder „ALFABET“ ein, um das Feld **Technopedia-(Unter-)Kategorie** im Selektor mit Anbieterproduktkategorien/Komponentenkategorien aus der Alfabet-Datenbank zu füllen, oder „TECHNOPEDIA“, um sie über einen separaten Serviceaufruf an das Technopedia-Repository zu füllen.



Beachten Sie Folgendes:

- Der Wert `TECHNOPEDIA` muss spezifiziert sein, um die ursprüngliche Taxonomie der Technopedia-Produktkategorien zu importieren. Anwender können Anbieterprodukte/Komponenten aus dem Technopedia-Repository nur in eine Anbieterproduktkategorie/Komponentenkategorie/ein ICT-Objekt in Alfabet importieren, die denselben Namen wie die Produktkategorie trägt, der sie im Technopedia-Repository zugeordnet sind.
 - Die Auffüllung des Selektors durch Technopedia-Produktkategorien aus der Alfabet-Datenbank sorgt für eine bessere Systemleistung, es kann aber eventuell zu Diskrepanzen zwischen den Daten in der Alfabet-Datenbank und denen im Technopedia-Repository kommen.
 - Es ist zwar technisch möglich, Anbieterproduktkategorien in Alfabet zu erstellen und Kategorien von Technopedia® zu implementieren, aber es wird empfohlen, dass Ihr Unternehmen nur eine Quelle zum Erstellen von Anbieterproduktkategorien verwendet. Wenn Sie im XML-Attribut `category_source` die Option `TECHNOPEDIA` eintragen, können Anwender die in Alfabet erzeugten Anbieterproduktkategorien im Explorer **Anbieterprodukte** weder sehen noch diese Anbieterproduktkategorien in den Technopedia-Selektoren auswählen.
 - Wenn der Wert „ALFABET“ für das XML-Attribut `category_source` angegeben ist, das Abonnementkonzept implementiert ist und ICT-Objekte auf der Grundlage von Technopedia-Produkten erzeugt werden, werden Anbieterproduktkategorien verwendet, um das Feld **Technopedia-(Unter-)Kategorie** zu füllen.
- `timeout`: Geben Sie das Zeitlimit (in Sekunden) an, das auf die HTTP-Anforderung angewendet werden soll, die an den Technopedia-API-Endpunkt gesendet wird.

- `certificate_path`: Geben Sie den Pfad für selbstsignierte, von Technopedia bereitgestellte Zertifikate an.
 - `default_gadate_period`: Geben Sie eine Ganzzahl an, die zur Berechnung eines Standard-Startdatums verwendet werden soll, wenn die Eigenschaft `general_availability_date` für ein Technopedia-Software- oder Hardwareprodukt leer ist. Die Ganzzahl muss die Anzahl der Jahre darstellen, die von der Eigenschaft `end_of_life_date` für das Technopedia-Produkt abgezogen werden sollen. Die Objektklasseneigenschaft `TP_ARTIFICIAL_GADATE` wird für das Anbieterprodukt automatisch auf `True` gesetzt, wenn das Startdatum auf Basis des XML-Elements `default_gadate_period` berechnet wird.
 - `tp_sync_scope`: Geben Sie in einer kommasetrennten Liste `VendorProductCategory`, `ICTObjectCategory` und/oder `ComponentCategory` an, damit die entsprechenden Kategorien über das vordefinierte ADIF-Importschema `ALFABET_TECHNOPEDIA_UPDATE` erzeugt oder aktualisiert werden können. Wenn beispielsweise ein XML-Element `ClassMapping` für die Klasse `VendorProduct` angezeigt werden soll, fügen Sie `VendorProductCategory` dem XML-Element `tp_sync_scope` hinzu. Wenn ein XML-Element `ClassMapping` für die Klasse `ICTObject` angezeigt werden soll, fügen Sie `ICTObjectCategory` dem XML-Element `tp_sync_scope` hinzu. Für jede angegebene Klasse ist die Schaltfläche **Alle Technopedia-Kategorien importieren** in der entsprechenden Ansicht verfügbar. Wenn dieses XML-Element nicht definiert ist, werden diese Klassen nicht über das ADIF-Importschema `ALFABET_TECHNOPEDIA_UPDATE` importiert oder synchronisiert.
- 4) Mit dem XML-Element `ClassMapping` können Sie die relevante Zuordnung für die folgenden Klassen festlegen: `VendorProduct`, `Component`, `ICTObject`, `Vendor`, `VendorProductCategory`, `ICTObjectCategory` und `ComponentCategory`. Fügen Sie dem Root-XML-Element `TechnopediaConfig` für jede relevante Objektklasse ein untergeordnetes XML-Element `ClassMapping` hinzu, und geben Sie die folgenden XML-Attribute an:



Wenn für eine im XML-Attribut `class` eingegebene Klasse Klassenschlüssel definiert wurden, für die das Attribut **Eindeutig** auf `True` gesetzt ist, sollten Sie sicherstellen, dass das Attribut **Strikte Nullbehandlung** des Klassenschlüssels für die Klasse `VendorProduct` (oder `Component`) auf `False` gesetzt ist, damit sämtliche Einträge mit NULL-Werten aus eindeutigen Indizes ausgeschlossen werden. Dadurch können doppelte Einträge eingefügt werden, wenn eines der Index-Attribute auf NULL gesetzt wird. Weitere Informationen über das Konfigurieren eindeutiger Schlüssel finden Sie im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* im Abschnitt *Konfigurieren von Klassenschlüsseln für Objektklassen*.

- **So konfigurieren Sie die Zuordnung für die Klasse Anbieterprodukt (oder Komponente):**
 - `class`: Geben Sie `VendorProduct` (oder `Component`) ein.
 - `sw_stereotype` und `hw_stereotype`: Dies sind optionale Attribute, mit denen Sie Objektklassenstereotypen zuordnen können, die für die im XML-Attribut `class` angegebene Objektklasse spezifiziert sind. So können Sie verschiedene Anbieterproduktstereotype implementieren, um verschiedene Arten von Hardware- und Softwareprodukten darzustellen. Definieren Sie die XML-Attribute `sw_stereotype` und `hw_stereotype` folgendermaßen:
 - `sw_stereotype`: Geben Sie den Namen des Objektklassenstereotyps ein, das für die Objektklasse spezifiziert ist, die im XML-Attribut `class` angegeben ist. Dieses Attribut kann basierend auf Technopedia-Softwareprodukten erzeugt werden.

- `hw_stereotype`: Geben Sie den Namen des Objektklassenstereotyps ein, der für die Objektklasse spezifiziert ist, die im XML-Attribut `class` angegeben ist. Dieses Attribut kann basierend auf Technopedia-Hardwareprodukten erzeugt werden.
- `lc_phase1` und `lc_phase2`: Dies sind optionale Attribute, mit denen Sie die Lebenszyklusphasen zuordnen können, die für die im XML-Attribut `class` angegebene Objektklasse spezifiziert sind. Für manche Technopedia-Produkte ist jeweils ein Datum für den Start des Produktlebenszyklus, für das Ende des Produktlebenszyklus und für das Datum definiert, zu dem das Produkt obsolet wird. Lebenszyklusphasen werden erzeugt, wenn alle Datumsangaben im Technopedia®-Produkt sowie die XML-Attribute `lc_phase1` und `lc_phase2` definiert sind. Die Daten der ersten Lebenszyklusphase (`lc_phase1`) basieren auf dem Datumsbereich von Datum 1 bis $\min(\text{Datum 2}, \text{Datum 3})$, die Daten der zweiten Lebenszyklusphase (`lc_phase2`) auf dem Datumsbereich von $\min(\text{Datum 2}, \text{Datum 3})$ bis $\max(\text{Datum 2}, \text{Datum 3})$. Die Lebenszyklus-Attribute für Technopedia-Softwareprodukte sind wie folgt:

- Datum 1 = `general_availability_date`
- Datum 2 = `end_of_life_date`
- Datum 3 = `obsolete_date`

Bitte beachten Sie, dass die Technopedia-Lebenszyklusattribute `general_availability_date`, `end_of_life_date` und `obsolete_date` den Lebenszyklus von Technopedia-Softwareprodukten beschreiben. Die Lebenszyklus-Attribute für Technopedia-Hardwareprodukte sind wie folgt:

- Datum 1 = `general_availability_date`
- Datum 2 = `last_availability_date`
- Datum 3 = `obsolete_date`

Die XML-Attribute `lc_phase1` und `lc_phase2` sollten zwei Lebenszyklusphasen zugeordnet werden, die für die Objektklasse `VendorProduct` im XML-Objekt ***ObjectLifecycleManager*** definiert sind. Dabei ist zu beachten, dass diese Namen exakt wie im XML-Objekt ***ObjectLifecycleManager*** für die Objektklasse `VendorProduct` angegeben geschrieben werden müssen. Definieren Sie die XML-Attribute `lc_phase1` und `lc_phase2` folgendermaßen:

- `lc_phase1`: Geben Sie den Namen der Lebenszyklusphase für die Objektklasse `VendorProduct` im XML-Objekt ***ObjectLifecycleManager*** an, die Sie dem Zeitraum zuordnen möchten, der durch den Datumsbereich Datum 1 (`general_availability_date`) – den Mindestwert von Datum 2 (`end_of_life_date/last_availability_date`) oder Datum 3 (`obsolete_date`) dargestellt wird. Ist kein `obsolete_date` angegeben, wird das Datum für das `end_of_life_date/last_availability_date` verwendet.
- `lc_phase2`: Geben Sie den Namen der Lebenszyklusphase für die Objektklasse `VendorProduct` im XML-Objekt ***ObjectLifecycleManager*** an, die Sie dem Zeitraum zuordnen möchten, der durch den Mindestwert des Datumsbereichs von Datum 2 (`end_of_life_date/last_availability_date`) oder Datum 3 (`obsolete_date`) – den Höchstwert von Datum 2 (`end_of_life_date/last_availability_date`) oder Datum 3 (`obsolete_date`) dargestellt wird. Ist kein `obsolete_date` angegeben, wird das Datum für das `general_availability_date` verwendet.

- `update_mode`: Wenn mit einem Technopedia-Produkt ein Anbieterprodukt verbunden ist und das Technopedia-Projekt mit einer Technopedia-Produktversion/-Release verbunden ist, wird in Alfabet in der Regel ein ICT-Objekt auf Grundlage der referenzierten Technopedia-Produktversion/-Release erzeugt, sobald das ADIF-Schema `ALFABET_TECHNOPEDIA_UPDATE` ausgeführt wird. Mit dem XML-Attribut `update_mode` können Sie festlegen, was geschehen soll, wenn das Anbieterprodukt in Alfabet vorhanden ist, aber keine Referenz zum relevanten ICT-Objekt aufweist. Geben Sie eine der folgenden Optionen für die Klasse `VendorProduct` an, um Anbieterprodukte für die ICT-Objekte zu erzeugen und zusammenzuführen, die Alfabet ohne die entsprechenden Verweise auf Anbieterprodukte hinzugefügt wurden:
 - `Merge`: Das System versucht, ein entsprechendes ICT-Objekt zu finden, das passende Werte für angegebene Alfabet -Eigenschaften aufweist, die Technopedia-Attributen zugewiesen wurden.
 - `CreateOrMerge`: Das System versucht, ein vorhandenes ICT-Objekt zu finden, und erzeugt andernfalls ein neues ICT-Objekt.
 - `None`: Es wird kein Versuch zum Zusammenführen oder Erzeugen unternommen.
- `merge_props`: Geben Sie die Alfabet -Eigenschaften an, die verwendet werden sollen, wenn Alfabet -Objekte mit Technopedia-Objekten in Übereinstimmung gebracht werden sollen. Die standardmäßigen Zusammenführungseigenschaften für die Klasse `VendorProduct` lauten `Name`, `Version`, `Alias`, `Suite`, `Edition` und `Stereotype`. Für jede Alfabet -Eigenschaft, die im XML-Attribut `merge_props` angegeben ist, muss ein XML-Element `AttributeMapping` konfiguriert werden.
- Erzeugen Sie für jedes XML-Element `ClassMapping` ein oder mehrere XML-Elemente `AttributeMapping`, um die Attribute in Technopedia benutzerdefinierten Objektklasseneigenschaften zuzuordnen, die für die im XML Attribut `class` angegebene Objektklasse konfiguriert sind. Spezifizieren Sie für jedes XML-Element `AttributeMapping` die folgenden XML-Attribute:
 - `tp_attr`: Geben Sie die Namen der Attribute in Technopedia ein, um sie der benutzerdefinierten Objektklasseneigenschaft in Alfabet zuzuordnen, die im Attribut `alfa_attr` spezifiziert ist.
 - `alfa_attr`: Geben Sie die Namen der benutzerdefinierten Alfabet - Objektklasseneigenschaft ein, um sie dem Technopedia@-Attribut zuzuordnen, das im Attribut `tp_attr` spezifiziert ist.
- **So konfigurieren Sie die Zuordnung für die Klasse `ICTObject`**: Diese Konfiguration ist nur relevant, wenn ein Abonnementkonzept implementiert wird, bei dem Technopedia-Produkte, die einem ICT-Objekt zugeordnet sind, automatisch zu Alfabet hinzugefügt und regelmäßig aktualisiert werden können. In diesem Fall werden ICT-Objekte in Alfabet basierend auf Software- und Hardwareprodukten in Technopedia erzeugt. Die auf Technopedia-Produkten basierenden ICT-Objekte sind versionslos und können daher als Abonnement für ein Technopedia-Produkt dienen. Die Versionen des Technopedia-Produkts, auf dem das ICT-Objekt basiert, können als Anbieterprodukte in Alfabet importiert werden, wenn das ICT-Objekt aktualisiert wird. Um das Abonnementkonzept zu implementieren, muss das Kontrollkästchen **Hat Abonnement** auf der Registerkarte **Technopedia** des Editors **ICT-Objekt** (`ICTO_TP_Editor`) ausgewählt sein. Wenn diese Option ausgewählt ist, werden Anbieterprodukte für alle neuen Technopedia-Produktversionen erzeugt, die dem Technopedia-Produkt zugeordnet sind, auf dem das ICT-Objekt basiert. Die Anbieterprodukte

werden den Ansichtsseiten der **Anbieterprodukte** hinzugefügt, wenn der ADIF-Prozess ALFABET_TECHNOPEDIA_UPDATE ausgeführt wird.

- `class`: Geben Sie `ICTObject` ein.
- `sw_stereotype`: Geben Sie den Namen des Objektklassenstereotyps ein, der für die Objektklasse `ICTObject` angegeben ist. Dieses Attribut kann basierend auf Technopedia-Softwareprodukten erzeugt werden.
- `hw_stereotype`: Geben Sie den Namen des Objektklassenstereotyps ein, der für die Objektklasse `ICTObject` angegeben ist. Dieses Attribut kann basierend auf Technopedia-Hardwareprodukten erzeugt werden.
- `update_mode`: Wenn mit einem Technopedia-Produkt ein Anbieterprodukt verbunden ist und das Technopedia-Projekt mit einer Technopedia-Produktversion/-Release verbunden ist, wird in Alfabet in der Regel ein ICT-Objekt auf Grundlage der referenzierten Technopedia-Produktversion/-Release erzeugt, sobald das ADIF-Schema ALFABET_TECHNOPEDIA_UPDATE ausgeführt wird. Mit dem XML-Attribut `update_mode` können Sie festlegen, was geschehen soll, wenn das relevante ICT-Objekt in Alfabet vorhanden ist, aber keine Referenz zum Anbieterprodukt aufweist. Geben Sie eine der folgenden Optionen für die Klasse `ICTObject` an, um ICT-Objekte für die Anbieterprodukte zu erzeugen und zusammenzuführen, die Alfabet ohne die entsprechenden Verweise auf ICT-Objekte hinzugefügt wurden:
 - `Merge`: Das System versucht, ein entsprechendes ICT-Objekt zu finden, das passende Werte für angegebene Alfabet -Eigenschaften aufweist, die Technopedia-Attributen zugewiesen wurden.
 - `Create`: Das System erzeugt ein neues ICT-Objekt. Der Standardwert lautet `Create`.
 - `CreateOrMerge`: Das System versucht, ein vorhandenes ICT-Objekt zu finden, und erzeugt andernfalls ein neues ICT-Objekt.
 - `None`: Es wird kein Versuch zum Zusammenführen oder Erzeugen unternommen.
- `merge_props`: Geben Sie die Alfabet -Eigenschaften an, die verwendet werden sollen, wenn Alfabet -Objekte mit Technopedia-Objekten in Übereinstimmung gebracht werden sollen. Die standardmäßigen Zusammenführungseigenschaften für die Klasse `ICTObject` lauten `Name` und `Stereotype`. Für jede Alfabet -Eigenschaft, die im XML-Attribut `merge_props` angegeben ist, muss ein XML-Element `AttributeMapping` konfiguriert werden.
- Erzeugen Sie ein oder mehrere XML-Elemente `AttributeMapping` als untergeordnetes Element des XML-Attributs `ClassMapping`, um die Attribute in Technopedia benutzerdefinierten Objektklasseneigenschaften zuzuordnen, die für die im XML Attribut `class` angegebene Objektklasse konfiguriert sind. Spezifizieren Sie für jedes XML-Element `AttributeMapping` die folgenden XML-Attribute:
 - `tp_attr`: Geben Sie die Namen der Attribute in Technopedia ein, um sie der benutzerdefinierten Objektklasseneigenschaft in Alfabet zuzuordnen, die im Attribut `alfa_attr` spezifiziert ist.
 - `alfa_attr`: Geben Sie die Namen der benutzerdefinierten Alfabet - Objektklasseneigenschaft ein, um sie dem Technopedia@-Attribut zuzuordnen, das im Attribut `tp_attr` spezifiziert ist.
- **So konfigurieren Sie die Zuordnung für die Klasse Anbieter:**

- `class`: Geben Sie Vendor ein.
- Erzeugen Sie für jedes XML-Element `ClassMapping` ein oder mehrere XML-Elemente `AttributeMapping`, um die Attribute in Technopedia benutzerdefinierten Objektklasseneigenschaften zuzuordnen, die für die im XML Attribut `class` angegebene Objektklasse konfiguriert sind. Spezifizieren Sie für jedes XML-Element `AttributeMapping` die folgenden XML-Attribute:
 - `tp_attr`: Geben Sie die Namen der Attribute in Technopedia ein, um sie der benutzerdefinierten Objektklasseneigenschaft in Alfabet zuzuordnen, die im Attribut `alfa_attr` spezifiziert ist.
 - `alfa_attr`: Geben Sie die Namen der benutzerdefinierten Alfabet - Objektklasseneigenschaft ein, um sie dem Technopedia@-Attribut zuzuordnen, das im Attribut `tp_attr` spezifiziert ist.
- **So fügen Sie den zugehörigen *Category-Objekten in Alfabet (VendorProductCategory, ComponentCategory und ICTObjectCategory) die Beschreibungsinformationen für Technopedia-Kategorien und -Unterkategorien hinzu:**
 - Stellen Sie sicher, dass die Kategorieklasse (`VendorProductCategory`, `ICTObjectCategory` und/oder `ComponentCategory`) im XML-Element `tp_sync_scope` angegeben ist.
 - Fügen Sie dem Root-XML-Element `TechnopediaConfig` für jede relevante Kategorieklasse, die im XML-Element `tp_sync_scope` angegeben ist, ein untergeordnetes XML-Element `ClassMapping` hinzu: Spezifizieren Sie folgende XML-Attribute:
 - Klasse: Geben Sie die entsprechende Kategorieklasse ein (`VendorProductCategory`, `ICTObjectCategory` oder `ComponentCategory`).
 - Fügen Sie dem XML-Element `ClassMapping` ein untergeordnetes XML-Element `AttributeMapping` hinzu.

`tp_attr`: Geben Sie Folgendes ein DESCRIPTION

`alfa_attr`: Geben Sie Folgendes ein Description

5) In the toolbar, click the **Save**  button to save the XML definition.

Bereitstellen der Technopedia-Funktionalität für die Anwender-Community

Sie müssen sicherstellen, dass die entsprechenden Ansichtsseiten, die die Technopedia-Funktionalität unterstützen, für die Anwender-Community zur Verfügung stehen. Die Seitenaufrufe sollten nach Bedarf in den Objektansichten verfügbar sein, die den Anwenderprofilen der Anwender zugehörig sind, die für das Definieren und Pflegen von Anbieterprodukten (oder Komponenten) zuständig sind. Informationen hierzu finden Sie in den Kapiteln *Konfigurieren von Objektansichten* und *Konfigurieren von Anwenderprofilen für die Anwender-Community* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Wenn Ihr Unternehmen Objektklassenstereotypen für die Klasse `VendorProduct` implementiert hat, können Sie in der benutzerdefinierten Objektansicht die Menüoptionen ausblenden, die für den Anbieterproduktstereotyp nicht relevant sind. Wenn Ihr Unternehmen die Verwendung sowohl von Hardware- als auch

von Softwareprodukten nicht unterstützt, ist es ratsam, die nicht relevanten Menüoptionen ebenfalls auszublenden. Weitere Informationen zur Durchführung finden Sie im Abschnitt *Ausblenden von Funktionalitäten in einer Seitenansicht oder einem konfigurierten Bericht* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Die folgenden Alfabet -Ansichten unterstützen die Erzeugung von Anbieterprodukten auf der Basis von Technopedia-Software- und Hardwareprodukten. Welche Ansichten Sie in ein Anwenderprofil aufnehmen müssen, hängt von der in Ihrem Unternehmen eingesetzten Methodik und der Konfiguration des XML-Objekts **TechnopediaConfig** ab:

- Die folgenden Ansichten ermöglichen das Erzeugen von Anbieterprodukten auf der Basis von Technopedia Software- und Hardwareprodukten. Für die Suche nach Technopedia-Softwareprodukten oder Technopedia-Hardwareprodukten ist jeweils ein unterschiedlicher Selektor implementiert:
 - Ansichtsseite **Anbieterprodukte** (VPC_Products) im Objektprofil „Anbieterproduktkategorie“
 - Ansichtsseite **Anbieterprodukte** (VDR_VendorProducts) im Objektprofil „Anbieter“
 - Ansichtsseite **Anbieterprodukte** (DOM_VendorProducts) im Objektprofil „Domäne“
 - Ansichtsseite **Anbieterprodukte** (ICTO_VendorProducts) im Objektprofil „ICT-Objekt“
 - Ansichtsseite **Anbieterprodukte** (COM_VendorProduct) im Objektprofil „Komponente“
 - Funktionalität **Anbieterprodukte erfassen** (VP_CaptureVendorProduct und COM_CaptureCVendorProduct_Ex)
 - Funktionalität **Anbieterprodukte dokumentieren** (Document_VendorProduct)
 - Objektprofil **Anbieterprodukt** (VP_ObjectView): In dieser Ansicht können Sie vorhandene Anbieterprodukte mit Anbieterprodukten im Technopedia-Repository zusammenführen.
 - Ansichtsseite **Root-Anbieterproduktkategorien** (VPC_RootCategories), verfügbar im Root-Knoten des Explorers **Anbieterprodukte**. Auf dieser Ansichtsseite können Sie das gesamte Repository an Technopedia®-Produktkategorien als Anbieterproduktkategorien importieren.
- Es kann auch ein Abonnementkonzept implementiert werden, bei dem Technopedia-Produkte, die einem ICT-Objekt zugeordnet sind, automatisch zu Alfabet hinzugefügt und regelmäßig aktualisiert werden können. In diesem Fall werden ICT-Objekte in Alfabet basierend auf Software- und Hardwareprodukten in Technopedia erzeugt. Die auf Technopedia-Produkten basierenden ICT-Objekte sind versionslos und können daher als Abonnement für eine Technopedia-Release dienen. Die Release-Versionen des Technopedia-Produkts, auf dem das ICT-Objekt basiert, können als Anbieterprodukte in Alfabet importiert werden, wenn das ICT-Objekt aktualisiert wird.

Die folgenden Ansichten ermöglichen das Erzeugen von ICT-Objekten auf der Basis von Technopedia Software- und Hardwareprodukten. Für die Suche nach Technopedia-Softwareprodukten oder Technopedia-Hardwareprodukten ist jeweils ein unterschiedlicher Selektor implementiert:

- Ansichtsseite **ICT-Objekte** (ICTOC_ICTObjects) im Objektprofil „ICT-Objektkategorie“
- Ansichtsseite **ICT-Objekte** (ICTOG_ICTObjects) im Objektprofil „ICT-Objektgruppe“
- Ansichtsseite **ICT-Objekte** (DOM_ICTObjects) im Objektprofil „Domäne“
- Ansichtsseite **ICT-Objekte** (VD_ICTObjects) im Objektprofil „Anbieter“

- Funktionalität **ICT-Objekte erfassen** (ICTO_CaptureICTObjects und ICTO_CaptureICTObjects_Ex)
- Funktionalität **ICT-Objekte dokumentieren** (Document_ICTObjects)
- Beachten Sie, dass der Editor **ICT-Objekt** (ICTO_TP_Editor) für die Implementierung des Abonnementkonzepts erforderlich ist und den relevanten Klasseneinstellungen für die Klasse „ICT-Objekt“ zugeordnet werden muss. Dieser Editor enthält das Kontrollkästchen **Hat Abonnement** auf der Registerkarte **Technopedia**. Wenn es aktiviert ist, bedeutet das, dass das ICT-Objekt das Technopedia-Produkt abonniert hat, auf dem es basiert. Mithilfe des Felds **Abonnement-Ebene** kann die Release-Versionsebene (**Alle Releases, Hauptversionen, Nebenversionen**) des zum Erzeugen der Anbieterprodukte verwendeten Technopedia-Produkts angegeben werden. Anbieterprodukte werden auf der Grundlage aller entsprechenden freigegebenen Versionen des Technopedia-Produkts erzeugt, auf denen das ICT-Objekt basiert. Nachdem das ICT-Objekt in Alfabet erzeugt wurde, muss der ADIF-Prozess ALFABET_TECHNOPEDIA_UPDATE ausgeführt werden, um die Anbieterprodukte zu erzeugen. Die Anbieterprodukte werden den Ansichtsseiten der **Anbieterprodukte** hinzugefügt, wenn der ICT-Prozess `<Code>ALFABET_TECHNOPEDIA_UPDATE</Code>` ausgeführt wird.
- Objektprofil **ICT-Objekt** (ICTO_ObjectView): In dieser Ansicht können Sie vorhandene ICT-Objekte mit ICT-Objekten im Technopedia-Repository zusammenführen.
- Ansichtssseite **Root-ICT-Objektkategorien** (ICTOC_Roots), verfügbar im Root-Knoten des Explorers **ICT-Objekte nach Kategorie**. In dieser Ansicht können Sie das gesamte Repository an Technopedia-Produktkategorien als ICT-Objektkategorien importieren.
- Anbieter können basierend auf einem Technopedia-Hersteller in den folgenden Ansichten erzeugt werden:
 - **Explorer „Anbieter“** (VDR_Explorer)
 - Ansichtssseite **Anbieter** (VDRG_Vendors) im Objektprofil „Anbietergruppe“
 - Funktionalität **Anbieter erfassen** (COM_CaptureVendor und COM_CaptureVendor_Ex)



Wenn Ihr Unternehmen die Technopedia-Funktionalität implementiert, Komponenten anstelle von Anbieterprodukten zu erzeugen, unterstützen die folgenden Alfabet -Ansichten das Erzeugen von Komponenten basierend auf Technopedia-Software- und Hardwareprodukten:

- Ansichtssseite **Root-Kategorien** (COMC_RootCategories), verfügbar im Root-Knoten des Explorers **Komponenten**. In dieser Ansicht können Sie das gesamte Repository an Technopedia@-Produktkategorien als Komponentenkategorien importieren.
- Die folgenden Ansichten ermöglichen das Erzeugen von Technopedia Software- und Hardwareprodukten. Für die Suche nach Technopedia-Softwareprodukten oder Technopedia-Hardwareprodukten ist jeweils ein unterschiedlicher Selektor implementiert:
 - Ansichtssseite **Komponenten** (COMC_Components) im Objektprofil „Komponentenkategorie“
 - Ansichtssseite **Komponenten** (COMG_Components) im Objektprofil „Komponentengruppe“
 - Ansichtssseite **Komponenten** (DOM_Components) im Objektprofil „Domäne“

- Funktionalität **Komponenten erfassen** (COM_CaptureComponent und COM_CaptureComponent_Ex)
- Funktionalität **Komponenten dokumentieren** (Document_Component)

Aktualisieren von Technopedia-Produkten in Alfabet über das ADIF-Importschema ALFABET_TECHNOPE- DIA_UPDATE

Ein vordefiniertes ADIF-Importschema steht zur Verfügung, um die auf Technopedia-Software- und Hardwareprodukten basierenden Anbieterprodukte, ICT-Objekte und Anbieter sowie deren Anbieter mit den aktuellen Daten im Technopedia-Repository zu aktualisieren. Das ADIF-Importschema ALFABET_TECHNOPE-DIA_UPDATE ist ein geschütztes ADIF-Importschema und kann nicht bearbeitet werden. Beim Ausführen liest das ADIF-Importschema die im XML-Objekt **TechnopediaConfig** festgelegte Konfiguration aus. Die im Technopedia-Repository erfolgten Änderungen werden in der Alfabet-Datenbank aktualisiert. Wenn an den relevanten Produkten im Technopedia-Repository keine Änderungen erfolgt sind, wird eine entsprechende Meldung angezeigt.

Die Technopedia-Produkte können von einem Anwender mit einem administrativen Anwenderprofil in der Funktionalität *ADIF-Prozessverwaltung* synchronisiert werden. Informationen hierzu finden Sie im Abschnitt *Ausführen und Steuern von ADIF-Prozessen* im Referenzhandbuch *Anwender- und Lösungsverwaltung*.

Im Folgenden werden Informationen zur Aktualisierung der Daten in der Alfabet über das vordefinierte ADIF-Importschema ALFABET_TECHNOPE-DIA_UPDATE beschrieben. Das Update wird mit Daten in der Alfabet mit den aktuellen Informationen im Technopedia-Repository übereinstimmen. Eine Übersicht über die Attributzuordnung finden Sie im Abschnitt [Beschreibung der Zuordnung von Technopedia-Produkten zu Anbieterprodukten in Alfabet](#).

Beachten Sie Folgendes:

- **Anbieterprodukte:**
 - Jedes Anbieterprodukt wird in der Alfabet-Datenbank aktualisiert, wenn folgende Kriterien erfüllt sind:
 - Die TP_ID-Eigenschaft des Anbieterprodukts ist definiert (was den Bezeichner des product_name-Wertes in Technopedia bildet, auf dem das Anbieterprodukt basiert).
 - Der TP_UPDATE-Wert liegt vor dem Attribut last_modified_date des Produkts im Technopedia-Repository. Der TP_UPDATE-Wert ist der Zeitstempel der letzten Aktualisierung des Anbieterprodukts mit dem entsprechenden Produkt im Technopedia-Repository.
 - Alle anderen Anbieterprodukte, die die Kriterien nicht erfüllen, bleiben unverändert.
 - Die folgenden Objektklasseneigenschaften werden für Softwareprodukte über das ADIF-Importschema aktualisiert ALFABET_TECHNOPE-DIA_UPDATE:
 - Name
 - Version
 - Vendor

- Alias
- Suite
- Edition
- StartDate: Das StartDate wird aktualisiert, wenn das general_availability_date definiert ist.
- EndDate: Das EndDate wird aktualisiert, wenn ein bestimmtes Datum für eines der folgenden Daten in der folgenden Abfolge bestimmt ist: obsolete_date, end_of_life_date, general_availability_date.
- Lebenszyklusphasen werden erzeugt, wenn alle relevanten Datumsangaben im Technopedia®-Produkt sowie die XML-Attribute lc_phase1 und lc_phase2 im XML-Objekt **TechnopediaConfig** definiert sind. Beachten Sie Folgendes:
 - Die erste Lebenszyklusphase besteht aus general_availability_date minus den Minimalwert von entweder end_of_life_date oder obsolete_date>.
 - Die letzte Lebenszyklusphase besteht aus <Minimalwert von end_of_life_date last_availability_date oder obsolete_date Minus Maximalwert von end_of_life_date oder obsolete_date>.
- Alle anderen Eigenschaften werden wie im XML-Attribut AttributeMapping für die Klasse VendorProduct des XML-Objekts **TechnopediaConfig** zugeordnet.
- Die folgenden Objektklasseneigenschaften werden für Hardwareprodukte über das ADIF-Importschema aktualisiert ALFABET_TECHNOPEDIA_UPDATE:
 - Name
 - Version
 - Vendor
 - Alias
 - StartDate: Das StartDate wird aktualisiert, wenn das general_availability_date definiert ist.
 - EndDate: Das EndDate wird aktualisiert, wenn ein bestimmtes Datum für eines der folgenden Daten in der folgenden Abfolge bestimmt ist: obsolete_date, last_availability_date, general_availability_date.
 - Lebenszyklusphasen werden erzeugt, wenn alle relevanten Datumsangaben im Technopedia®-Produkt sowie die XML-Attribute lc_phase1 und lc_phase2 im XML-Objekt **TechnopediaConfig** definiert sind. Beachten Sie Folgendes:
 - Die erste Lebenszyklusphase besteht aus general_availability_date minus den Minimalwert von entweder last_availability_date oder obsolete_date>.
 - Die letzte Lebenszyklusphase besteht aus <Minimalwert von end_of_life_date last_availability_date oder obsolete_date Minus Maximalwert von end_of_life_date oder obsolete_date>.
 - Alle anderen Eigenschaften werden wie im XML-Attribut AttributeMapping für die Klasse VendorProduct des XML-Objekts **TechnopediaConfig** zugeordnet.

- **Anbieter:** Beachten Sie Folgendes:
 - Bitte beachten Sie, dass jeder Anbieter in der Alfabet-Datenbank aktualisiert wird, wenn folgende Kriterien erfüllt sind:
 - Die `TP_ID`-Eigenschaft des Anbieters ist definiert (was den Bezeichner des `manufacturer`-Wertes in Technopedia bildet, auf dem der Anbieter basiert).
 - Der `TP_UPDATE`-Wert liegt vor dem Attribut `last_modified_date` des Produkts im Technopedia-Repository. Der `TP_UPDATE`-Wert ist der Zeitstempel der letzten Aktualisierung des Anbieterprodukts mit dem entsprechenden Produkt im Technopedia-Repository.
 - Ein neuer Anbieter wird automatisch in der Alfabet-Datenbank für jedes Anbieterprodukt erzeugt, das auf einem Hardwareprodukt von Technopedia basiert, für welches kein Wert für die `manufacturer`-Eigenschaft angegeben wurde, automatisch erzeugt.
 - Die folgenden Objektklasseneigenschaften werden über das ADIF-Importschema aktualisiert `ALFABET_TECHNOPEDIA_UPDATE`:
 - Name
 - ShortName
 - Description
 - Country
 - City
 - Street
 - Zip
 - Email
 - Fax
 - Phone
 - Website
 - Alle anderen Eigenschaften werden wie im XML-Attribut `AttributeMapping` für die Klasse `Vendor` des XML-Objekts **TechnopediaConfig** zugeordnet.
- **Komponenten:** Die Komponenten werden nicht über das vordefinierte ADIF-Importschema `ALFABET_TECHNOPEDIA_UPDATE` aktualisiert. Sie müssen ein ADIF-Importschema konfigurieren, um Alfabet -Komponenten mit den aktuellen Informationen in Technopedia zu synchronisieren.
- **ICT-Objekte:** Beachten Sie Folgendes:
 - Wenn die Eigenschaft `TP_Subscribed` des ICT-Objekts auf `True` gesetzt ist, erzeugt die Ausführung des vordefinierten ADIF-Importschemas `ALFABET_TECHNOPEDIA_UPDATE` neue Anbieterprodukte, wie über die Eigenschaft `TP_SUBSCRIPTIONLEVEL` des ICT-Objekts angegeben.
 - Der Verweis eines Anbieterprodukts zu einem ICT-Objekt wird über die Ausführung des ADIF-Importschemas `ALFABET_TECHNOPEDIA_UPDATE` 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 vom ICT-Objekt referenzierten Technopedia-Produkt unterscheidet. Mit dem XML-Attribut `update_mode` können Sie festlegen, was geschehen soll, wenn das relevante ICT-Objekt in Alfabet vorhanden ist, aber keine Referenz zum Anbieterprodukt aufweist.

- `StartDate` und `EndDate` sowie die Lebenszyklus-Informationen werden nicht vom Technopedia-Produkt in das ICT-Objekt kopiert.
- Über das XML-Attribut `AttributeMapping` zugeordnete Eigenschaften für die Klasse `ICTObject` im XML-Objekt ***TechnopediaConfig*** können über das ADIF-Importschema `ALFABET_TECHNOPEDIA_UPDATE` aktualisiert werden.
- **Anbieterproduktkategorien:** Anbieterproduktkategorien werden über das vordefinierte ADIF-Importschema `ALFABET_TECHNOPEDIA_UPDATE` aktualisiert, wenn der Wert `VendorProductCategory` für das XML-Element `tp_sync_scope` im XML-Objekt ***TechnopediaConfig*** angegeben ist. Wenn ein Anbieterprodukt auf der Grundlage eines Technopedia-Produkts erzeugt wird und die zugehörige Kategorie in Alfabet nicht vorhanden ist, wird die Anbieterproduktkategorie bei der Ausführung des ADIF-Importschemas `ALFABET_TECHNOPEDIA_UPDATE` in Alfabet erzeugt.
- **ICT-Objektkategorien:** ICT-Objektkategorien werden über das vordefinierte ADIF-Importschema `ALFABET_TECHNOPEDIA_UPDATE` aktualisiert, wenn der Wert `ICTObjectCategory` für das XML-Element `tp_sync_scope` im XML-Objekt ***TechnopediaConfig*** angegeben ist. Wenn ein ICT-Objekt auf der Grundlage eines Technopedia-Produkts erzeugt wird und die zugehörige Kategorie in Alfabet nicht vorhanden ist, wird die ICT-Objektkategorie bei der Ausführung des ADIF-Importschemas `ALFABET_TECHNOPEDIA_UPDATE` in Alfabet erzeugt.
- **Komponentenkategorien:** Komponentenkategorien werden über das vordefinierte ADIF-Importschema `ALFABET_TECHNOPEDIA_UPDATE` aktualisiert, wenn der Wert `ComponentCategory` für das XML-Element `tp_sync_scope` im XML-Objekt ***TechnopediaConfig*** angegeben ist. Wenn eine Komponente auf der Grundlage eines Technopedia-Produkts erzeugt wird und die zugehörige Kategorie in Alfabet nicht vorhanden ist, wird die Komponentenkategorie bei der Ausführung des ADIF-Importschemas `ALFABET_TECHNOPEDIA_UPDATE` in Alfabet erzeugt.

Kapitel 5: Konfigurieren der Interoperabilität mit CentraSite

Durch die Interoperabilität mit CentraSite können relevante Assets in CentraSite® als technische Services in Alfabet erzeugt werden, sodass diese Informationen verwendet werden können, um Business-Funktionen, die in Alfabet geplant und dokumentiert werden, am operativen Kontext auszurichten. Außerdem können technische Services, die auf Basis der Business-Anforderungen in Alfabet geplant werden, zur operativen Umsetzung und Steuerung zu CentraSite übertragen werden. Alfabet kann bei Bedarf Verbindungen zu mehreren CentraSite-Instanzen herstellen.

Folgendes wird von der Alfabet -Benutzeroberfläche unterstützt:

- Auf den Ansichtsseiten **Technische Service-Registry-Services – gefiltert** (COMSR_ServicesExt) und **Technische Service-Registry-Services** (COMSR_Services) werden relevante CentraSite-Assets angezeigt. Dort können Sie auch neue technische Services in Alfabet erzeugen, die auf diesen CentraSite-Assets basieren. Jedes Mal, wenn die Ansichtsseite **Technische Service-Registry-Services – gefiltert** oder die Ansichtsseite **Technische Service-Registry-Services** geladen wird, wird jede im XML-Objekt **CentraSiteManager** konfigurierte Verbindung hergestellt und alle auf CentraSite-Assets basierenden technischen Services werden mit den Daten im CentraSite-Repository synchronisiert.
- Über die Ansichtsseite **Technische Services** (COM_TechServices) können Assets in CentraSite mit den Informationen aktualisiert werden, die für ihre zugehörigen technischen Services in Alfabet definiert sind. Wenn der Anwender einen ausgewählten technischen Service auf der Ansichtsseite **Technische Services** (COM_TechServices) mit dem entsprechenden CentraSite-Asset synchronisieren möchte, muss die für die Synchronisierung zu verwendende relevante Verbindungsdefinition im Feld **Verbindung** auf der Registerkarte **Service-Registry** des Editors **Technischer Service** ausgewählt werden. Nur ein gültiger technischer Service kann im CentraSite-Repository aktualisiert werden.

Im Folgenden finden Sie eine kurze Übersicht über die allgemeinen Konfigurationsanforderungen für Interoperabilität mit CentraSite:

- Für die Objektklassen `Service` und `OrgaUnit` müssen Objektklassenstereotypen konfiguriert werden.
- Für die Objektklassen `Person` und `OrgaUnit` müssen benutzerdefinierte Eigenschaften konfiguriert werden.
- Für die Objektklasse `Service` und die Objektklassenstereotypen im XML-Objekt **ReleaseStatusDefs** müssen Release-Status konfiguriert werden. Diese Release-Status sollten den Lebenszyklusstatus in CentraSite entsprechen. Das XML-Objekt **ReleaseStatusDefs** ist in Alfabet Expand verfügbar.
- Im XML-Objekt **CentraSiteManager** müssen eine oder mehrere Verbindungen zu CentraSite konfiguriert werden. Das XML-Objekt **CentraSiteManager** ist in Alfabet Expand verfügbar. Informationen zum Konfigurieren von XML-Objekten in Alfabet Expand finden Sie im Abschnitt *Arbeiten mit XML-Objekten* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- In der Ansicht **CentraSite-Verbindungen** der Funktionalität **Integrationslösungskonfigurationen** müssen CentraSite-Verbindungsobjekte erzeugt werden, damit ein Anwender auf der Ansichtsseite **Technische Services** (COM_TechServices) die Verbindung zur relevanten CentraSite-Instanz für die Synchronisierung eines technischen Services angeben kann. Ein CentraSite-Verbindungsobjekt sollte für jede konfigurierte Verbindung im XML-Objekt erstellt werden **CentraSiteManager**

- Die relevanten Anwenderprofile müssen konfiguriert werden, um die Ansichtsseiten **Technische Service-Registry-Services – gefiltert** (COMSR_ServicesExt) und **Technische Services** (COM_TechServices) zu enthalten. Die Ansichtssseite **Technische Service-Registry-Services** (COMSR_Services) sollte für Anwenderprofile, die auf das CentraSite-Repository zugreifen, nicht sichtbar sein.
- Die Attribute **Assembly** und **Assembly-Klasse**, die für den Lösungsmanager **ServiceRegistryManager** verfügbar sind, müssen spezifiziert werden, um Interoperabilität zu gewährleisten. Das XML-Objekt **ServiceRegistryManager** ist in Alfabet Expand verfügbar. Folgende Werte müssen spezifiziert werden:
 - **Assembly:** ITPlan
 - **Assembly-Klasse:** ITPlanSolution.GenericServiceRegistryManager
- Wenn zum Konfigurieren der Verbindung im XML-Objekt **CentraSiteManager** Servervariablen verwendet werden, müssen die Servervariablen für den Server-Alias im Alfabet Administrator spezifiziert werden.
- Die ADIF-Schemata `Alfabet_CentraSite_Organization_Synchronization` und `Alfabet_CentraSite_User_Synchronization` sollten bei dem ersten Konfigurieren der Interoperabilität mit CentraSite ausgeführt werden, um die Eigentümerorganisationen und Eigentümer, die den relevanten Assets in CentraSite zugeordnet sind, in Alfabet zu importieren.
- Das ADIF-Schema `Alfabet_CentraSite_Asset_Synchronization` sollte regelmäßig ausgeführt werden, um die Daten der CentraSite-Assets in Alfabet zu aktualisieren.

Folgende Informationen sind verfügbar:

- [Konfigurieren des Klassenmodells für Interoperabilität mit CentraSite](#)
- [Konfigurieren von Verbindungen für Interoperabilität mit CentraSite](#)
- [Konfigurieren der Anzeige von CentraSite-Services in alfabet](#)
- [Konfigurieren der Aktualisierung von alfabet-Daten in CentraSite](#)
- [Importieren von CentraSite-Daten über ADIF-Schemata](#)

Konfigurieren des Klassenmodells für Interoperabilität mit Centra-Site

Die folgenden Konfigurationen für die Objektklasse `Service` werden empfohlen oder sind erforderlich, um Interoperabilität mit CentraSite zu bieten.

- Konfigurieren von Objektklassenstereotypen für die Objektklasse `Service`. Jeder Asset-Typ in CentraSite, der in Alfabet angezeigt werden soll, muss einem Objektklassenstereotyp der Klasse `Service` im XML-Element **AssetTypeMappings** des XML-Objekts **CentraSiteManager** zugeordnet werden. Daher müssen Sie die relevanten Objektklassenstereotypen für die Klasse `Service` konfigurieren, bevor Sie das XML-Objekt **CentraSiteManager** spezifizieren können. Weitere Informationen über das Erstellen und Zuordnen von Objektklassen-Stereotypen für technische Services finden Sie unter *Konfigurieren von Objektklassenstereotypen für technische Services* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Die Objektklassenstereotypen für die Objektklasse `Service` könnten beispielsweise folgendermaßen angegeben werden:

```
<ClassStereotypes>
  <Stereotype Name="SOAPService" Caption="SOAP Service"
    CaptionPlural="SOAP Services" Comments="" HasMandates="false" />
  <Stereotype Name="DataService" Caption="DataService"
    CaptionPlural="Data Services" Comments="" HasMandates="false" />
  <Stereotype Name="RESTService" Caption="REST Service"
    CaptionPlural="REST Services" Comments="" HasMandates="false" />
  <Stereotype Name="XMLService" Caption="XML Service"
    CaptionPlural="XML Services" Comments="" HasMandates="false" />
</ClassStereotypes>
```

- Konfigurieren Sie Release-Status für die Objektklasse `Service` sowie auch alle relevanten Objektklassenstereotypen für die Objektklasse `Service`. Jeder Asset-Typ in CentraSite wird über einen Lebenszyklus verfügen, der aus Lebenszyklusstatus besteht. Daher sollte jeder Objektklassenstereotyp über entsprechende Release-Status verfügen, die den Lebenszyklusstatus des Asset-Typs entsprechen, die ihm zugeordnet werden. Bitte beachten Sie Folgendes:
 - Der Satz an Release-Status für jeden relevanten Objektklassenstereotyp muss im XML-Objekt **ReleaseStatusDefs** konfiguriert werden. Für die Objektklasse `Service` als Ganzes sowie für die einzelnen Objektklassenstereotypen muss eine Release-Status-Definition erzeugt werden. Die Release-Status-Definition für die Objektklasse muss den kompletten Satz an Release-Status enthalten, die für ihre Objektklassenstereotypen konfiguriert wurden. Die Release-Status-Definition für einen Objektklassenstereotyp darf nur die Release-Status enthalten, die für den Objektklassenstereotyp relevant sind, sowie eine Abfolge der Release-Status, die zum Erreichen eines bestimmten Ziel-Release-Status verfügbar sind. Weitere Informationen zum Konfigurieren von Release-Status finden Sie im Abschnitt *Konfigurieren von Release-Status-Definitionen für Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

```

<ReleaseStatusDef
  ClassNames = "Service"

  StatusSet = "Requested, Designing, Designed, Design-Approved,
    Handedover-to-production, Retired"
  RetiredStatusSet = "Retired"
  EditableStatusSet = "Requested, Designing, Designed, Design-Approved,
    Handedover-to-production, Retired"
  DefaultStatus = "Requested"
  ApprovedStatus = "Design-Approved">

  <StatusTransition ToStatus="Requested" FromStatuses="" />
  <StatusTransition ToStatus="Designing" FromStatuses="Requested, Designed" />
  <StatusTransition ToStatus="Designed" FromStatuses="Designing" />
  <StatusTransition ToStatus="Design-Approved" FromStatuses="Designed" />
  <StatusTransition ToStatus="Handedover-to-production"
    FromStatuses="Design-Approved" />
  <StatusTransition ToStatus="Retired" FromStatuses="any" />

  <Status Name="Requested"
    Hint="Service has been requested and is now detailed." />
  <Status Name="Designing"
    Hint="The Service is currently designed by service provider." />
  <Status Name="Designed"
    Hint="Service is designed and is waiting for approval." />
  <Status Name="Design-Approved"
    Hint="Service design is approved." />
  <Status Name="Handedover-to-production"
    Hint="Service is handed over to production." />
  <Status Name="Retired"
    Hint="Service is retired and not productively usable any more." />

</ReleaseStatusDef>

```

- Die Release-Status in Alfabet sollten in der primären Sprache spezifiziert werden, die der Definition der Basis-Landeseinstellungen der standardmäßigen Landeseinstellung entspricht. Wenn ein Anwender versucht, Daten in Alfabet mit CentraSite-Assets zu synchronisieren, und für ein oder mehrere dieser CentraSite-Assets Lebenszyklusstatus in einer anderen Sprache als der primären Sprache in Alfabet definiert sind, wird eine Fehlermeldung angezeigt.
- Für jeden für einen Asset-Typ verfügbaren Lebenszyklusstatus muss für den relevanten Objektstereotyp ein zugehöriger Release-Status definiert sein. Der Name des Release-Status muss mit dem Namen des Lebenszyklusstatus übereinstimmen.
- Der für den Objektklassenstereotyp verfügbare Satz von Release-Status kann Release-Status beinhalten, die für die Verwaltung der technischen Services in alfabet relevant sind. Solche Release-Status werden als Alfabet -Eigentum betrachtet, da sie nur in Alfabet definiert werden.
- Die in CentraSite definierten Lebenszyklusstatus, die zugleich auch CentraSite-Eigentum sind, sollten dieselben Namen aufweisen, wie die in Alfabet definierten Release-Status.
- Der Satz an Release-Status, die CentraSite-Eigentum sind, sollte nicht in Alfabet bearbeitbar sein. Diese Lebenszyklusstatus (wie Designbeschreibung, Implementierung, Validierung oder Laufzeit) sollten nur in CentraSite definiert werden.
- Einer der Release-Status, die im Release-Status-Satz konfiguriert sind, wird im XML-Attribut `HandoverStatus` des XML-Objekts **CentraSiteManager** als der Release-Status spezifiziert, über den der technische Service verfügen muss, um im CentraSite-Repository aktualisiert zu werden. Falls erforderlich kann für jeden Objektklassenstereotyp ein anderer Übergabestatus spezifiziert werden.

```

<AssetTypeMappings>
  <AssetTypeMapping
    CentraSiteAssetType="Service"
    HandoverStatus="Requested"
    AlfabetStereotype="SOAPService"/>
  <AssetTypeMapping
    CentraSiteAssetType="REST service"
    HandoverStatus="Designing"
    AlfabetStereotype="RESTService"/>
  <AssetTypeMapping
    CentraSiteAssetType="OData service"
    HandoverStatus="Requested"
    AlfabetStereotype="DataService"/>
  <AssetTypeMapping
    CentraSiteAssetType="XMLService"
    HandoverStatus="Requested"
    AlfabetStereotype="XMLService"/>
</AssetTypeMappings>

```



Beachten Sie, dass das CentraSite-Asset vom Anfangsstatus in CentraSite direkt zu dem Lebenszyklusstatus in CentraSite, der dem Übergabestatus zugeordnet ist, wechseln muss, wenn der konfigurierte Übergabestatus nicht dem Anfangslebenszyklusstatus in CentraSite zugeordnet ist. Die technischen Services in Alfabet werden nicht übertragen, wenn die zugehörigen CentraSite-Assets über andere dazwischenliegende Lebenszyklusstatus in CentraSite zum Übergabestatus wechseln.

- Konfigurieren Sie für die Objektklasse `Person` eine benutzerdefinierte Eigenschaft. Ihr Unternehmen kann sich entscheiden, eine benutzerdefinierte Eigenschaft für die Klasse `Person` zu konfigurieren, um anzugeben, welche Anwendertypen in das CentraSite-Repository importiert werden können.
- Die benutzerdefinierte Eigenschaft sollte in einem benutzerdefinierten Editor für die Klasse `Person` verfügbar sein, damit für jeden Anwender Werte definiert werden können. Die benutzerdefinierte Eigenschaft und zulässige Werte werden im XML-Attribut `UserFilterProperty` und im XML-Element **`UserFilterPropertyValues`** des XML-Objekts **`CentraSiteManager`** spezifiziert. Beachten Sie, dass die Standardeigenschaft `EXTERNAL_SOURCE`, die für die Objektklasse `Person` verfügbar ist, anstelle einer benutzerdefinierten Eigenschaft verwendet werden kann. Weitere Informationen über das Konfigurieren benutzerdefinierter Eigenschaften finden Sie im Abschnitt *Konfigurieren benutzerdefinierter Eigenschaften für geschützte oder öffentliche Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- Konfigurieren Sie für die Objektklasse `OrgaUnit` eine benutzerdefinierte Eigenschaft des Typs `String` mit einer benutzerdefinierten Aufzählung. Über die benutzerdefinierte Eigenschaft kann die Organisation in Alfabet identifiziert werden, die einer Organisation in der CentraSite-Instanz zugeordnet werden soll. Die benutzerdefinierte Eigenschaft sollte für jede Verbindung, die Sie konfigurieren möchten, einen zulässigen Wert für die Objektklasse `OrgaUnit` enthalten. Der Name der benutzerdefinierten Eigenschaft wird in das XML-Attribut `InstanceSeparatingOrgProperty` des Root-XML-Elements **`CentraSiteConfig`** eingegeben. Der zulässige Wert für die benutzerdefinierte Eigenschaft, die im XML-Attribut `InstanceSeparatingOrgProperty` spezifiziert wurde, wird dann in das XML-Attribut `InstanceSeparatingOrgPropertyValue` des relevanten XML-Elements **`Connection`** eingegeben.

- Konfigurieren von Objektklassenstereotypen für die Objektklasse `OrgaUnit`. Dies ist erforderlich, wenn Sie planen, Alfabet -Daten in CentraSite zu aktualisieren. Jedes Asset in CentraSite muss über eine gültige Eigentümerorganisation verfügen. Wenn die Interoperabilität mit CentraSite erstmals konfiguriert wird, sollte das ADIF-Schema

`Alfabet_CentraSite_Organization_Synchronization` ausgeführt werden, um alle Organisationen, die Eigentümer von relevanten Assets sind, nach Alfabet zu importieren. Die Organisationen, die zu Beginn über das ADIF-Schema importiert werden, werden als der Organisation, die im XML-Attribut `AlfabetParentOrganization` des XML-Objekts **CentraSiteManager** identifiziert wird, untergeordnete Organisationen hinzugefügt. Hinweis: Wenn Sie in das XML-Objekt **CentraSiteManager** einen String eingeben möchten, der Sonderzeichen enthält (z. B. die Symbole für größer als (>) oder kleiner als (<)), müssen Sie diese durch den entsprechenden kompatiblen HTML-Code ersetzen. Informationen zum Konfigurieren von XML-Objekten in Alfabet Expand finden Sie im Abschnitt *Arbeiten mit XML-Objekten* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

```
<ClassStereotypes>
  <Stereotype Name="SharedServiceCenter" Caption="Shared Service Center"
    CaptionPlural="Shared Service Centers" Comments=""
    HasMandates="false" />
  <Stereotype Name="ExternalOrganization" Caption="External Organization"
    CaptionPlural="External Organizations" Comments=""
    HasMandates="false" />
  <Stereotype Name="OperatingEntity" Caption="Operating Entity"
    CaptionPlural="Operating Entities" Comments=""
    HasMandates="false" />
  <Stereotype Name="ITServiceCenter" Caption="IT Service Center"
    CaptionPlural="IT Service Centers" Comments=""
    HasMandates="false" />
</ClassStereotypes>
```

- Konfigurieren Sie bei Bedarf Rollentypen für die Objektstereotypen, die für die Objektklasse `Service` konfiguriert sind, um den Eigentümer und die Eigentümerorganisation des technischen Services zu identifizieren. Rollentypen werden im Modul **Konfiguration** konfiguriert. Informationen hierzu finden Sie im Abschnitt *Konfigurieren von Rollentypen zur Definition von Rollen auf der Ansichtssseite „Verantwortlichkeiten“*.

Konfigurieren von Verbindungen für Interoperabilität mit CentraSite

Um den Anforderungen von Partner-Organisationen gerecht zu werden, kann Alfabet Verbindungen zu mehreren CentraSite-Repositories herstellen.



Vor Alfabet -Release 10.0 konnte im XML-Objekt **CentraSiteManager** nur eine Verbindung zum CentraSite-Repository konfiguriert werden. Ab Alfabet -Release 10.0 können mehrere Verbindungen konfiguriert werden, um Verbindungen zu mehreren CentraSite-Repositories herzustellen. Durch diese Erweiterung hat sich auch die Struktur des XML-Objekts **CentraSiteManager** geändert, um die Konfiguration mehrerer Verbindungen zu unterstützen. Beachten Sie, dass das XML-Element **Connection** weiterhin ein untergeordnetes Element des Root-XML-Elements **CentraSiteConfig** ist, jetzt jedoch das übergeordnete XML-Element von vorhandenen untergeordneten XML-Elementen **AssetTypeMapping**, **OrganizationMapping** und **UserMapping** ist. Mehrere untergeordnete Elemente **Connection** können für das Root-XML-Element **CentraSiteConfig** konfiguriert werden, wobei jedes XML-Element **Connection** eine Verbindung von Alfabet zu einem CentraSite-Repository sowie die Zuordnung von für die Verbindung relevanten Asset-Typen, Organisationen und Anwendern spezifiziert. Außerdem verfügt das Root-XML-Element

CentraSiteConfig über ein neues XML-Attribut `InstanceSeparatingOrgProperty` und ein neues untergeordnetes Element **Proxy**.

Beachten Sie bitte, dass Ihre vorhandene Konfiguration des XML-Objekts **CentraSiteManager** nicht automatisch der neuen XML-Struktur angepasst wird, wenn Sie das XML-Objekt **CentraSiteManager** vor Alfabet -Release 10.0 konfiguriert haben. Wenn Ihr Unternehmen mehrere Verbindungen konfigurieren möchte, müssen Sie die Struktur des XML-Objekts **CentraSiteManager** manuell auf Basis der Definition im Attribut **XML-Vorlage** und des Attributs **XML-XSD** des XML-Objekts **CentraSiteManager** überarbeiten.

Die Konfiguration der Verbindungen wird im XML-Objekt **CentraSiteManager** über das Root-XML-Element **CentraSiteConfig** und dem diesem untergeordneten XML-Element **Connection** vorgenommen. Sobald Sie die Konfiguration des XML-Objekts **CentraSiteManager** abgeschlossen haben, müssen Sie in der Ansicht **CentraSite-Verbindungen**, die in der Funktionalität **Integrationslösungskonfigurationen** in der Alfabet -Benutzeroberfläche verfügbar ist, Verbindungsobjekte erzeugen.

Die allgemeine Struktur der XML im XML-Objekt sieht wie folgt aus:

```
<?xml version="1.0" encoding="utf-8"?>
<CentraSiteConfig InstanceSeparatingOrgProperty="CSINSTANCE">
  <Proxy url="" user="" password="" domain="">
    <AdditionalProxies>
      <AdditionalProxy Name="" url="" user="" password="" domain="" />
    </AdditionalProxies>
  </Proxy>
  <Connection Name="CS1">
    Proxy="None"
    InstanceSeparatingOrgPropertyValue="CentraSite1"
    CentraSiteURL="$CentraSiteURL"
    CentraSiteUsername="$CentraSiteUsername"
    CentraSitePassword="$CentraSitePassword"
    ServiceViewURL="$ServiceViewURL"
    ExcludedLifecycles=""
    CentraSiteApiMaturityStatus=""
    Timeout="300"
    DefaultPageSize="1000"
    OwnerDefinition="SELECT top 1 p.REFSTR FROM PERSON p
                    INNER JOIN ROLE r ON r.RESPONSIBLE=p.REFSTR
                    INNER JOIN SERVICE svc ON r.OBJECT=svc.REFSTR
                    WHERE svc.REFSTR=@BASE"
    OrganizationDefinition="SELECT ou.REFSTR FROM ORGAUNIT ou
                           INNER JOIN ICTOBJECT icto ON icto.OWNER=ou.REFSTR
                           INNER JOIN COMPONENT com ON com.ICTOBJECT=icto.REFSTR
                           INNER JOIN SERVICE svc ON svc.OWNER=com.REFSTR
                           WHERE svc.REFSTR=@BASE">
  <AssetTypeMappings>
```

Das Root-XML-Element **CentraSiteConfig** verfügt über das XML-Attribut `InstanceSeparatingOrgProperty`. Dieses XML-Attribut identifiziert die Organisation in Alfabet, die einer Organisation in der CentraSite-Instanz zugeordnet wird, und wird verwendet, um zu identifizieren, zu welcher CentraSite-Verbindung die Organisation gehört. Der eingegebene Wert sollte eine benutzerdefinierte Eigenschaft der Objektklasse `OrgaUnit` spezifizieren.



Wenn Sie in das XML-Objekt **CentraSiteManager** einen String eingeben möchten, der Sonderzeichen enthält (z. B. die Symbole für größer als (>) oder kleiner als (<)), müssen Sie diese durch den entsprechenden kompatiblen HTML-Code ersetzen, zum Beispiel:

- `>` für >
- `<` für <
- `"` für "
- `[` für [

- `]` für]

Das Root-XML-Element **CentraSiteConfig** verfügt über zwei untergeordnete Elemente:

- **Proxy:** Über dieses XML-Element können Sie konfigurieren, wie Anforderungen über einen Proxyserver an eine CentraSite-Instanz gesendet werden. Die Konfiguration des XML-Elements **Proxy** wird im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#) beschrieben.
- **Connection:** Über dieses XML-Element können Sie die Verbindung zu einer CentraSite-Instanz konfigurieren. Für jede Verbindung zu einer CentraSite-Instanz sollte ein XML-Element **Connection** konfiguriert werden. Jedes XML-Element verfügt über drei untergeordnete Elemente. Diese untergeordneten Elemente sind **AssetTypeMappings**, **OrganizationMapping** und **UserMapping**. Die Konfiguration dieser XML-Elemente wird in den folgenden Abschnitten beschrieben:
 - [Konfigurieren der Anzeige von CentraSite-Services in alfabet](#)
 - [Konfigurieren der Aktualisierung von alfabet-Daten in CentraSite](#)

Sie können mehrere Verbindungen von Alfabet zu mehreren CentraSite-Repositorys konfigurieren. So definieren Sie eine Verbindung zu einem CentraSite-Repository:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte** > **IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf das XML-Objekt **CentraSiteManager**, und wählen Sie **XML bearbeiten** aus. Das XML-Objekt wird geöffnet.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist eine Vorlage über das Attribut **XML-Vorlage** des XML-Objekts verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf. Bei einer Beispielformatierung müssen Sie eher XML-Elemente bearbeiten, als neue hinzuzufügen.

- 3) Fügen Sie dem Root-XML-Element **CentraSiteConfig** das XML-Attribut **InstanceSeparatingOrgProperty** hinzu, und spezifizieren Sie die benutzerdefinierte Eigenschaft, die für die Objektklasse `OrgaUnit` definiert ist, über die wiederum die Organisation in Alfabet identifiziert werden kann, die einer Organisation in der CentraSite-Instanz zugeordnet werden soll.
- 4) Fügen Sie dem Root-XML-Element **CentraSiteConfig** ein untergeordnetes XML-Element **Connection** hinzu.
- 5) Konfigurieren Sie eine Verbindung zu CentraSite, indem Sie die folgenden XML-Attribute für das XML-Element **Connection** festlegen:
 - **Proxy:** Wenn Sie Proxys konfigurieren, fügen Sie jedem XML-Element **Connection**, das einen der zusätzlichen Proxys verwenden soll, ein XML-Attribut `Proxy` hinzu. Der Wert des XML-Attributs `Proxy` muss mit dem Wert des XML-Attributs `Name` des XML-Elements **AdditionalProxy** übereinstimmen. Die Konfiguration des XML-Elements **Proxy** wird im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#) beschrieben.
 - **InstanceSeparatingOrgPropertyValue:** Geben Sie den relevanten Wert der benutzerdefinierten Eigenschaft ein, die für die Objektklasse `OrgaUnit` definiert ist, die im XML-Attribut **InstanceSeparatingOrgProperty** des Root-XML-Elements **CentraSiteConfig**

spezifiziert wurde. Der Wert der benutzerdefinierten Eigenschaft identifiziert, zu welcher CentraSite-Verbindung die Organisation gehört.

- **Name:** Geben Sie einen eindeutigen Namen für die Verbindung zu CentraSite ein.
- **Active:** Geben Sie „true“ ein, um CentraSite mit den technischen Services von Alfabet zu aktualisieren und Services von CentraSite in Alfabet anzuzeigen.
- **CentraSiteURL:** Geben Sie die URL für den Zugriff auf CentraSite oder den relevanten Servervariablenamen ein. Beispiel:
`http://centrasite.company.com:53307/BusinessUI/#assetdetail`



In einigen XML-Attributen können Servervariablen verwendet werden, um den Wert des Attributs zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation auszulesen, wenn eine Verbindung zu CentraSite hergestellt wurde. Weitere Informationen über die Verwendung von Servervariablen finden Sie im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).

- **CentraSiteUserName:** Geben Sie den Anwendernamen Ihres Unternehmens für den Zugriff auf CentraSite oder den relevanten Servervariablenamen ein.
 - **CentraSitePassword:** Geben Sie das Passwort Ihres Unternehmens für den Zugriff auf CentraSite oder den relevanten Servervariablenamen ein.
 - **ServiceViewURL:** Geben Sie das Präfix für den vollständigen Link ein, der bei einem Klick eines Anwenders auf die Schaltfläche **Ansicht des Services in Service-Registry öffnen** auf der Ansichtsseite **Technische Service-Registry-Services – gefiltert** zum Navigieren von Alfabet zu CentraSite verwendet wird. Geben Sie die URL für den Zugriff auf die Serviceansicht in CentraSite oder den relevanten Servervariablenamen ein. Beispiel:
`http://centrasite.company.com:53307/BusinessUI/#assetdetail`
 - **ExcludedLifecycles:** Geben Sie einen oder mehrere CentraSite-Lebenszyklusstatus ein, die nicht auf der Alfabet -Benutzeroberfläche angezeigt werden sollen. Auf der Ansichtsseite **Technische Service-Registry-Services – gefiltert** werden keine CentraSite-Asset angezeigt, die einen der angegebenen Lebenszyklusstatus aufweisen.
 - **CentraSiteApiMaturityStatus:** Geben Sie bei Bedarf den Status der Asset-Eigenschaft „API-Reife“ in CentraSite ein, um die Assets, die in alfabet angezeigt werden, auf die mit dem angegebenen API-Reifestatus zu beschränken.
 - **Timeout:** Geben Sie die Anzahl von Minuten Anwenderinaktivität ein, nach der die Verbindung zu CentraSite beendet werden soll.
 - **DefaultPageSize:** Geben Sie eine Ganzzahl zwischen 10-1000 ein, um die maximale Zahl an Datensätzen festzulegen, die auf der Ansichtsseite **Service-Registry-Services – gefiltert** der Alfabet -Benutzeroberfläche zurückgegeben wird. Der empfohlene Wert ist „10“. Dabei ist zu beachten, dass eine übermäßig große Zahl (beispielsweise alles über 11) die Leistung beeinträchtigt.
- 6) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Konfigurieren der Anzeige von CentraSite-Services in alfabet

Wenn Ihr Unternehmen plant, CentraSite-Daten in Alfabet anzuzeigen, um neue auf CentraSite-Assets basierende technische Services in Alfabet zu erzeugen, müssen Sie im XML-Objekt **CentraSiteManager** Folgendes konfigurieren:

- Konfigurieren Sie im XML-Objekt **CentraSiteManager** ein XML-Element **Connection**, um die Autorisierung für den CentraSite-Server und die Verbindung zu diesem zu spezifizieren. Dies wird im Abschnitt [Konfigurieren von Verbindungen für Interoperabilität mit CentraSite](#) beschrieben. Beachten Sie, dass die XML-Attribute `OrganizationDefinition` und `OwnerDefinition` für den Import von Daten aus CentraSite nach Alfabet nicht erforderlich sind.
- Konfigurieren Sie im XML-Objekt **CentraSiteManager** das XML-Element **AssetTypeMappings**, um die Zuordnung von Asset-Typen in CentraSite zu den relevanten Objektklassenstereotypen der Klasse `Service` zu spezifizieren.

```
<AssetTypeMappings>
  <AssetTypeMapping
    CentraSiteAssetType="Service"
    HandoverStatus="Requested"
    AlfabetStereotype="SOAPService"/>
  <AssetTypeMapping
    CentraSiteAssetType="REST service"
    HandoverStatus="Designing"
    AlfabetStereotype="RESTService"/>
  <AssetTypeMapping
    CentraSiteAssetType="OData service"
    HandoverStatus="Requested"
    AlfabetStereotype="DataService"/>
  <AssetTypeMapping
    CentraSiteAssetType="XMLService"
    HandoverStatus="Requested"
    AlfabetStereotype="XMLService"/>
</AssetTypeMappings>
```

- Konfigurieren Sie im XML-Objekt **CentraSiteManager** das XML-Element **OrganizationMapping**, um zu spezifizieren, wie Eigentümerorganisationen von Assets in CentraSite in Alfabet behandelt werden sollen, wenn technische Services auf Basis von CentraSite-Assets erzeugt werden.

```
<OrganizationMapping CreateMissingOrganizationInAlfabet="true"
  AlfabetOrganizationStereotype="ITServiceCenter"
  AlfabetParentOrganization="Corporate IT">
  <AttributeMapping CentraSiteAttribute="Name" IsIdentifier="true"
    AlfabetProperty="Name" />
  <AttributeMapping CentraSiteAttribute="desc" IsIdentifier="false"
    AlfabetProperty="Description" />
</OrganizationMapping>
```

- Konfigurieren Sie im XML-Objekt **CentraSiteManager** das XML-Element **UserMapping**, um zu spezifizieren, wie Eigentümer von Assets in CentraSite in Alfabet behandelt werden sollen, wenn technische Services auf Basis von CentraSite-Assets erzeugt werden.

```

<UserMapping IgnoredDomainNames="INTERNAL">
  <AlfabetUserSynchronization CreateMissingUserInCentraSite="false"
    MaxRecordCount="100"
    UserFilterProperty="Registry">
    <UserFilterPropertyValues>
      <Value>CentraSite</Value>
      <Value>Partner</Value>
    </UserFilterPropertyValues>
  </AlfabetUserSynchronization>

```

So bearbeiten Sie das XML-Objekt **CentraSiteManager**:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf das XML-Objekt **CentraSiteManager**, und wählen Sie **XML bearbeiten** aus. Der XML-Editor wird im Arbeitsbereich in der Mitte angezeigt. Allgemeine Informationen zum Bearbeiten von XML-Objekten in Alfabet Expand finden Sie im Abschnitt *Arbeiten mit XML-Objekten*.
- 3) Erzeugen und definieren Sie das XML-Element **Connection**, wie im Abschnitt [Konfigurieren von Verbindungen für Interoperabilität mit CentraSite](#) beschrieben.
- 4) Jedes Asset in CentraSite verfügt über einen Asset-Typ. Sie müssen die Zuordnung von CentraSite-Asset-Typen zu den relevanten Objektklassenstereotypen, die für die Klasse `Service` in Alfabet konfiguriert sind, spezifizieren. Erzeugen Sie im XML-Element **AssetTypeMappings** für jeden CentraSite-Asset-Typ, der auf der Alfabet -Benutzeroberfläche angezeigt werden soll, ein XML-Element **AssetTypeMapping** mit folgenden XML-Attributen:



Beachten Sie, dass das ADIF-Schema `Alfabet_CentraSite_Asset_Synchroniza-tion` regelmäßig ausgeführt werden sollte, um die Daten von CentraSite-Asset-Typen in Alfabet zu aktualisieren.

- `CentraSiteAssetType`: Geben Sie den Namen des CentraSite-Asset-Typs ein, der dem im XML-Element `AlfabetStereotype` spezifizierten Objektklassenstereotyp zugeordnet werden soll.
 - `AlfabetStereotype`: Geben Sie den Namen des Objektklassenstereotyps der Klasse `Service` in Alfabet ein, der dem im XML-Element `CentraSiteAssetType` spezifizierten CentraSite-Asset-Typ zugeordnet werden soll.
 - `HandoverStatus`: Dieses XML-Attribut muss nicht spezifiziert werden, wenn Ihr Unternehmen nur CentraSite-Assets nach Alfabet importiert. (Das XML-Element `HandoverStatus` ist nur erforderlich, wenn Sie planen, Alfabet -Daten in CentraSite zu aktualisieren.)
- 5) Jedes Asset in CentraSite verfügt über eine Eigentümerorganisation. Konfigurieren Sie, wie die Definition der Eigentümerorganisation in Alfabet behandelt werden soll, indem Sie die folgenden XML-Attribute für das XML-Element **OrganizationMapping** festlegen:
 - `CreateMissingOrganizationInAlfabet`: Geben Sie „true“ ein, um zuzulassen, dass für Assets in CentraSite definierte Organisationen in Alfabet erzeugt werden, wenn diese Organisation in Alfabet noch nicht bereits vorhanden ist. Der Name der CentraSite-Organisation wird in der Alfabet-Datenbank auf Basis der Spezifikationen in den XML-Attributen `AlfabetOrganizationStereotype` und `AlfabetParentStereotype` erzeugt. Geben Sie „false“ ein, wenn die Eigentümerorganisation des Assets in CentraSite nicht in Alfabet erzeugt werden soll.

- **AlfabetOrganizationStereotype:** Wenn Sie das XML-Attribut `CreateMissingOrganizationInAlfabet` auf „true“ festlegen, geben Sie den Namen des Objektklassenstereotyps in Alfabet ein, dem die aus dem CentraSite-Repository importierten Organisationen zugeordnet werden sollen.
- **AlfabetParentOrganization:** Wenn Sie das XML-Attribut `CreateMissingOrganizationInAlfabet` auf „true“ festlegen, geben Sie den Namen des Objektklassenstereotyps in Alfabet ein, der als der Stereotyp der übergeordneten Organisation spezifiziert werden soll, dem aus dem CentraSite-Repository importierte Organisationen zugeordnet werden sollen. Wenn Sie dieses XML-Attribut nicht definieren, werden alle Organisationen zur obersten Ebene der Organisationshierarchie in Alfabet hinzugefügt.



Wenn Sie einen String eingeben möchten, der Sonderzeichen enthält (z. B. die Symbole für größer als (>) oder kleiner als (<)), müssen Sie diese durch den entsprechenden kompatiblen HTML-Code ersetzen, zum Beispiel:

- `>` für >
- `<` für <
- `"` für "
- `[` für [
- `]` für]



Beachten Sie, dass das ADIF-Schema `Alfabet_CentraSite_Oganization_Synchronization` zu Beginn ausgeführt werden sollte, um die Eigentümerorganisationen von Services in CentraSite nach Alfabet zu importieren.

- 6) Wenn Sie das XML-Attribut `CreateMissingOrganizationInAlfabet` auf „true“ festlegen, geben Sie an, welche für Eigentümerorganisationen in CentraSite definierten Attribute welchen Standard- oder benutzerdefinierten Eigenschaften des im XML-Attribut `AlfabetOrganizationStereotype` spezifizierten Organisationsstereotyps zugeordnet werden sollen. Erzeugen Sie im XML-Element **OrganizationMapping** für jede Kombination aus Alfabet - Eigenschaft/CentraSite-Attribut ein XML-Element **AttributeMapping** mit folgenden XML-Attributen:
- **CentraSiteAttribute:** Geben Sie den Namen eines Attributs für Eigentümerorganisationen in CentraSite ein, das der Standard- oder benutzerdefinierten Eigenschaft im folgenden XML-Attribut zugeordnet werden soll: `AlfabetProperty`.
 - **IsIdentifier:** Geben Sie „true“ ein, wenn das Attribut die eindeutige ID der Organisation in CentraSite ist. Geben Sie „false“ ein, wenn das Attribut nicht die eindeutige ID der Organisation in CentraSite ist.
 - **AlfabetProperty:** Geben Sie den Namen der Standard- oder benutzerdefinierten Eigenschaft ein, die für den relevanten Organisationsstereotyp in Alfabet verfügbar ist, um dem im XML-Attribut `CentraSiteAttribute` spezifizierten Attribut zugeordnet zu werden.
- 7) Jedes Asset in CentraSite verfügt über einen Eigentümer. Konfigurieren Sie, wie die Definition des Eigentümers in Alfabet behandelt werden soll, indem Sie die folgenden XML-Attribute für das XML-Element **UserMapping** festlegen:
- **IgnoreDomainName:** Spezifizieren Sie einen oder mehrere Domännennamen, die bei der Anzeige von Eigentümerinformationen in Alfabet ignoriert werden sollen.



Beachten Sie, dass das ADIF-Schema `Alfabet_CentraSite_User_Synchronisation` zu Beginn ausgeführt werden sollte, um die Eigentümer von Assets in CentraSite nach Alfabet zu importieren.

- 8) Erzeugen Sie für das XML-Element **UserMapping** ein untergeordnetes XML-Element `CentraSiteUserSynchronisation`, um die Synchronisierung der Eigentümerdefinition eines CentraSite-Assets nach Alfabet zu konfigurieren.
- `CreateMissingUserInAlfabet`: Geben Sie „true“ ein, um zuzulassen, dass für Assets in CentraSite definierte Anwender in Alfabet erzeugt werden, wenn dieser Anwender in Alfabet noch nicht bereits vorhanden ist.
 - `MaxRecordCount`: Geben Sie eine Ganzzahl zwischen 10-100 ein, um die maximale Zahl an Anwendern festzulegen, die von CentraSite nach Alfabet importiert werden. Beachten Sie, dass sich eine höhere Zahl auf die Leistung auswirkt.



Das untergeordnete XML-Element `AlfabetUserSynchronisation` des XML-Elements **UserMapping** muss nicht spezifiziert werden, wenn Ihr Unternehmen nur CentraSite-Assets nach Alfabet importiert (und Alfabet -Daten nicht in CentraSite aktualisiert werden.)

- 9) Wenn Sie das XML-Attribut `CreateMissingUserInAlfabet` auf „true“ festlegen, spezifizieren Sie, welche Standard- oder benutzerdefinierte Eigenschaften der Objektklasse `Person` automatisch mit Werten ausgefüllt werden sollen, wenn ein neuer Anwender in Alfabet erzeugt wird. Erzeugen Sie im XML-Element **CentraSiteUserSynchronisation** ein XML-Element **UserPropertyValues**. Erzeugen Sie für jede Eigenschaft, die automatisch für die aus CentraSite importierten Anwender festgelegt werden soll, ein XML-Element **PropertyValue**.

```
<CentraSiteUserSynchronisation CreateMissingUserInAlfabet="true"
MaxRecordCount="100">
  <UserPropertyValues>
    <PropertyValue AlfabetProperty="USER_NAME" Value="CentraSite"/>
  </UserPropertyValues>
</CentraSiteUserSynchronisation>
<AttributeMapping CentraSiteAttribute="userId" IsIdentifier="true"
AlfabetProperty="USER_NAME" />
<AttributeMapping CentraSiteAttribute="address" IsIdentifier="false"
AlfabetProperty="EMail" />
<AttributeMapping CentraSiteAttribute="firstName" IsIdentifier="false"
AlfabetProperty="FirstName" />
<AttributeMapping CentraSiteAttribute="lastName" IsIdentifier="false"
AlfabetProperty="Name" />
```

- `AlfabetProperty`: Geben Sie den Namen der für die Objektklasse `Person` in Alfabet verfügbaren Standard- oder benutzerdefinierten Eigenschaft ein, für die automatisch ein Wert definiert werden soll.
 - `Value`: Geben Sie den Wert ein, der automatisch für das XML-Attribut `AlfabetProperty` definiert werden soll.
- 10) Wenn Sie das XML-Attribut `CreateMissingUserInAlfabet` auf „true“ festlegen, geben Sie an, welche für Eigentümer in CentraSite definierten Attribute welchen Standard- oder benutzerdefinierten Eigenschaften der Objektklasse `Person` zugeordnet werden sollen. Erzeugen Sie im XML-Element **UserMapping** für jede Kombination aus CentraSite-Attribut/ Alfabet - Eigenschaft ein XML-Element **AttributeMapping** mit folgenden XML-Attributen:

- **CentraSiteAttribute:** Geben Sie den Namen eines Attributs für Eigentümer in CentraSite ein, das der Standard- oder benutzerdefinierten Eigenschaft im folgenden XML-Attribut zugeordnet werden soll: `AlfabetProperty`.
- **IsIdentifier:** Geben Sie „true“ ein, wenn das Attribut die eindeutige ID des Anwenders in CentraSite ist. Geben Sie „false“ ein, wenn das Attribut nicht die eindeutige ID des Anwenders in CentraSite ist.
- **AlfabetProperty:** Geben Sie den Namen der Standard- oder benutzerdefinierten Eigenschaft ein, die für die Objektklasse `Person` in Alfabet verfügbar ist, um dem im XML-Attribut `CentraSiteAttribute` spezifizierten Attribut zugeordnet zu werden.

11) In the toolbar, click the **Save**  button to save the XML definition.

Konfigurieren der Aktualisierung von alfabet-Daten in CentraSite

Beachten Sie Folgendes beim Aktualisieren von Alfabet -Daten in CentraSite:

- Konfigurieren Sie im XML-Objekt **CentraSiteManager** das XML-Element **Connection**, um die Autorisierung für den CentraSite-Server und die Verbindung zu diesem zu spezifizieren. Dies wird im Abschnitt [Konfigurieren von Verbindungen für Interoperabilität mit CentraSite](#) beschrieben. Spezifizieren Sie die Abfrage `OwnerDefinition`, um den Eigentümer des technische Services in CentraSite zu definieren, und spezifizieren Sie die Abfrage `OrganizationDefinition`, um eine gültige Eigentümerorganisation in CentraSite zu definieren. Im CentraSite muss bereits eine Organisation vorhanden sein, die denselben Namen wie die über diese Abfrage gefundene Organisation aufweist. In CentraSite muss bereits ein Anwender vorhanden sein, der denselben Namen wie der über diese Abfrage gefundene Anwender aufweist.
- Jedes Asset in CentraSite muss über eine gültige Eigentümerorganisation verfügen. Daher müssen Sie eine Organisation für den technischen Service definieren, die im CentraSite-Repository als gültig betrachtet wird, um einen in Alfabet erzeugten technischen Service mit dem CentraSite-Repository synchronisieren zu können. Die Eigentümerorganisation des technischen Services in Alfabet muss über eine in im XML-Attribut `OrganizationDefinition` des XML-Objekts **CentraSiteManager** spezifizierte Abfrage gefunden werden. Beachten Sie Folgendes bezüglich der Definition der Eigentümerorganisation:

```
<OrganizationMapping CreateMissingOrganizationInAlfabet="true"
    AlfabetOrganizationStereotype="ITServiceCenter"
    AlfabetParentOrganization="Corporate IT">
  <AttributeMapping CentraSiteAttribute="Name" IsIdentifier="true"
    AlfabetProperty="Name" />
  <AttributeMapping CentraSiteAttribute="desc" IsIdentifier="false"
    AlfabetProperty="Description" />
</OrganizationMapping>
```

- Wenn die Interoperabilität mit CentraSite erstmals implementiert wird, sollte das ADIF-Schema `Alfabet_CentraSite_Organization_Synchronization` ausgeführt werden, um alle Organisationen, die Eigentümer von relevanten Assets sind, nach Alfabet zu importieren.
- Die Organisationen, die zu Beginn über das ADIF-Schema importiert werden, werden als der Organisation, die im XML-Attribut `AlfabetParentOrganization` des XML-Objekts **CentraSiteManager** identifiziert wird, untergeordnete Organisationen hinzugefügt.



Wenn Sie einen String eingeben möchten, der Sonderzeichen enthält (z. B. die Symbole für größer als (>) oder kleiner als (<)), müssen Sie diese durch den entsprechenden kompatiblen HTML-Code ersetzen, zum Beispiel:

- > für >
 - < für <
 - " für "
 - [für [
 -] für]
- Da in CentraSite keine neuen Eigentümerorganisationen erzeugt werden können, muss die Abfrage, die Sie im XML-Attribut `OrganizationDefinition` definieren, Organisationen finden, die bereits in CentraSite vorhanden sind. Anders formuliert, findet die Abfrage normalerweise die Organisationen, die über das ADIF-Schema `Alfabet_CentraSite_Organization_Synchronization` importiert wurden.
 - Der Alfabet -Anwender, der die technischen Services von Alfabet in CentraSite aktualisiert, muss in CentraSite über die Berechtigungen verfügen, Assets für die Organisation zu erzeugen.
 - Jedes Asset in CentraSite muss über einen gültigen Eigentümer verfügen. Daher müssen Sie einen Anwender definieren, der im CentraSite-Repository als gültig betrachtet wird, um einen in Alfabet erzeugten technischen Service mit dem CentraSite-Repository synchronisieren zu können. Der Eigentümer des technischen Services in Alfabet muss über eine in im XML-Attribut `OwnerDefinition` des XML-Objekts **CentraSiteManager** spezifizierte Abfrage gefunden werden. Beachten Sie Folgendes bezüglich der Definition des Eigentümers:

```
<UserMapping IgnoredDomainNames="INTERNAL">
  <AlfabetUserSynchronization CreateMissingUserInCentraSite="false"
    MaxRecordCount="100"
    UserFilterProperty="Registry">
    <UserFilterPropertyValues>
      <Value>CentraSite</Value>
      <Value>Partner</Value>
    </UserFilterPropertyValues>
  </AlfabetUserSynchronization>
</UserMapping>
```

- Wenn die Interoperabilität mit CentraSite erstmals implementiert wird, sollte das ADIF-Schema `Alfabet_CentraSite_User_Synchronization` ausgeführt werden, um alle Anwender, die Eigentümer von relevanten Assets sind, nach Alfabet zu importieren.
- Anders als bei Organisationen ist es möglich, in CentraSite neue Eigentümer zu erzeugen, indem das XML-Attribut `CreateMissingUserInCentraSite` im XML-Objekt **CentraSiteManager** auf „true“ festgelegt wird.
- Wenn Sie das XML-Attribut `CreateMissingUserInCentraSite` im XML-Attribut **CentraSiteManager** auf „false“ festgelegt haben, muss die von Ihnen im XML-Attribut `OwnerDefinition` definierte Abfrage Anwender finden, die bereits in CentraSite vorhanden sind. In diesem Fall findet die Abfrage normalerweise die Anwender, die über das ADIF-Schema `Alfabet_CentraSite_User_Synchronization` importiert wurden.
- Konfigurieren Sie im XML-Objekt **CentraSiteManager** das XML-Element **AssetTypeMappings**, um die Zuordnung von Asset-Typen in CentraSite zu den relevanten Objektklassenstereotypen der

Klasse `Service` zu spezifizieren. Für jede Zuordnung Objektklassenstereotyp/Asset-Typ muss im XML-Attribut `HandoverStatus` ein Release-Status spezifiziert sein. Nur technische Services mit spezifiziertem `HandoverStatus` können im CentraSite-Repository aktualisiert werden.

```
<AssetTypeMappings>
  <AssetTypeMapping
    CentraSiteAssetType="Service"
    HandoverStatus="Requested"
    AlfabetStereotype="SOAPService"/>
  <AssetTypeMapping
    CentraSiteAssetType="REST service"
    HandoverStatus="Designing"
    AlfabetStereotype="RESTService"/>
  <AssetTypeMapping
    CentraSiteAssetType="OData service"
    HandoverStatus="Requested"
    AlfabetStereotype="DataService"/>
  <AssetTypeMapping
    CentraSiteAssetType="XMLService"
    HandoverStatus="Requested"
    AlfabetStereotype="XMLService"/>
</AssetTypeMappings>
```

So bearbeiten Sie das XML-Objekt **CentraSiteManager**:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte** > **IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf das XML-Objekt **CentraSiteManager**, und wählen Sie **XML bearbeiten** aus. Der XML-Editor wird im Arbeitsbereich in der Mitte angezeigt. Allgemeine Informationen zum Bearbeiten von XML-Objekten in Alfabet Expand finden Sie im Abschnitt *Arbeiten mit XML-Objekten*.
- 3) Erzeugen und definieren Sie das XML-Element **Connection**, wie im Abschnitt [Konfigurieren von Verbindungen für Interoperabilität mit CentraSite](#) beschrieben.
- 4) Jedes Asset in CentraSite muss über eine gültige Eigentümerorganisation verfügen. Daher müssen Sie sicherstellen, dass eine Organisation in Alfabet als die Eigentümerorganisation des technischen Services identifiziert wird, bevor ein technischer Service in Alfabet in CentraSite aktualisiert werden kann. Definieren Sie im XML-Attribut `OrganizationDefinition` des XML-Elements **Connection** eine alfabet-Abfrage oder Native-SQL-Abfrage, um nach der Organisation zu suchen. Beachten Sie beim Aktualisieren von technischen Services in CentraSite, dass die Informationen zur Eigentümerorganisation für alle Anwender mit Zugriff auf CentraSite verfügbar sind.



Beachten Sie Folgendes bezüglich der Definition der Eigentümerorganisation:

- Wenn die Interoperabilität mit CentraSite erstmals implementiert wird, sollte das ADIF-Schema `Alfabet_CentraSite_Organization_Synchronization` ausgeführt werden, um alle Organisationen, die Eigentümer von relevanten CentraSite-Assets sind, nach Alfabet zu importieren.
- Die Organisationen, die zu Beginn über das ADIF-Schema importiert werden, werden als der Organisation, die im XML-Attribut `AlfabetParentOrganizatio` des XML-Objekts **CentraSiteManager** identifiziert wird, untergeordnete Organisationen hinzugefügt. Hinweis: Wenn Sie in das XML-Objekt **CentraSiteManager** einen String eingeben möchten, der Sonderzeichen enthält (z. B. die Symbole für größer als (>) oder kleiner als (<)), müssen Sie diese durch den entsprechenden kompatiblen HTML-Code ersetzen.

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

- Da in CentraSite keine neuen Eigentümerorganisationen erzeugt werden können, muss die Abfrage, die Sie im XML-Attribut `OrganizationDefinition` definieren, Organisationen finden, die bereits in CentraSite vorhanden sind. Die Abfrage findet daher normalerweise die Organisationen, die über das ADIF-Schema `Alfabet_CentraSite_Organization_Synchronization` importiert wurden.
- 5) Jedes Asset in CentraSite muss über einen Eigentümer verfügen. Daher müssen Sie sicherstellen, dass ein Anwender in Alfabet als der Eigentümer des technischen Services identifiziert wird, bevor ein technischer Service in Alfabet in CentraSite aktualisiert werden kann. Definieren Sie hinsichtlich des Eigentümers Folgendes:
- Definieren Sie im XML-Attribut `OwnerDefinition` des XML-Elements **Connection** eine alfabet-Abfrage oder Native-SQL-Abfrage, um nach dem Anwender zu suchen. Beachten Sie beim Aktualisieren von technischen Services in CentraSite, dass die Informationen zum Eigentümer für alle Anwender mit Zugriff auf CentraSite verfügbar sind.



Beachten Sie Folgendes bezüglich der Definition des Eigentümers:

- Wenn die Interoperabilität mit CentraSite erstmals implementiert wird, sollte das ADIF-Schema `Alfabet_CentraSite_User_Synchronization` ausgeführt werden, um alle Anwender, die Eigentümer von relevanten Assets sind, nach Alfabet zu importieren.
 - Anders als bei Organisationen ist es möglich, in CentraSite neue Eigentümer zu erzeugen, indem das XML-Attribut `CreateMissingUserInCentraSite` im XML-Objekt **CentraSiteManager** auf „true“ festgelegt wird.
 - Wenn Sie das XML-Attribut `CreateMissingUserInCentraSite` im XML-Attribut **CentraSiteManager** auf „false“ festgelegt haben, muss die von Ihnen im XML-Attribut `OwnerDefinition` definierte Abfrage Anwender finden, die bereits in CentraSite vorhanden sind. In diesem Fall findet die Abfrage normalerweise die Anwender, die über das ADIF-Schema `Alfabet_CentraSite_User_Synchronization` importiert wurden.
- Konfigurieren Sie, wie die Definition des Eigentümers in Alfabet behandelt werden soll, indem Sie die folgenden XML-Attribute für das XML-Element **UserMapping** festlegen:



Beachten Sie, dass das ADIF-Schema `Alfabet_CentraSite_User_Synchronization` regelmäßig ausgeführt werden sollte, um die Daten zu Eigentümern von Assets in CentraSite in Alfabet zu aktualisieren.

- `IgnoreDomainName`: Spezifizieren Sie einen oder mehrere Domännennamen, die bei der Anzeige von Eigentümerinformationen in Alfabet ignoriert werden sollen.
- Erzeugen Sie für das XML-Element **UserMapping** ein untergeordnetes XML-Element **AlfabetUserSynchronization**, um die Aktualisierung von Eigentümerorganisationen in Alfabet nach CentraSite zu konfigurieren. Spezifizieren Sie folgende XML-Attribute:
 - `CreateMissingUserInCentraSite`: Geben Sie „true“ ein, um zuzulassen, dass für technische Services in Alfabet definierte Eigentümer nach CentraSite aktualisiert werden.

- **UserFilterProperty**: Geben Sie eine gültige Standard- oder benutzerdefinierte Eigenschaft der Klasse `Person` ein, um anzugeben, welche Anwender in CentraSite erzeugt werden dürfen. Zum Beispiel kann zum Herausfiltern der Anwender, die zu CentraSite hinzugefügt werden sollen, entweder die Standardeigenschaft `EXTERNAL_SOURCE` verwendet werden, oder eine benutzerdefinierte Eigenschaft, die so konfiguriert wurde, dass sie die relevanten Eigenschaften wiedergibt.
 - Erzeugen Sie für das XML-Element **AlfabetUserSynchronization** ein untergeordnetes XML-Element **UserFilterPropertyValues**, um zu spezifizieren, welche Werte für die im XML-Element **UserFilterProperty** spezifizierte Filtereigenschaft ausgewählt werden können. Erzeugen Sie für jeden Wert, der im Filter angezeigt werden soll, ein XML-Element **Value**.
- 6) Jedes Asset in CentraSite verfügt über einen Asset-Typ. Sie müssen die Zuordnung des CentraSite-Asset-Typs zum relevanten Objektklassenstereotyp, der für die Klasse `Service` in Alfabet konfiguriert ist, spezifizieren. Erzeugen Sie im XML-Element **AssetTypeMappings** für jeden CentraSite-Asset-Typ, der auf der Alfabet -Benutzeroberfläche angezeigt werden soll, ein XML-Element **AssetTypeMapping** mit folgenden XML-Attributen:
- **CentraSiteAssetType**: Geben Sie den Namen des CentraSite-Asset-Typs ein, der dem im XML-Element `AlfabetStereotype` spezifizierten Objektklassenstereotyp zugeordnet werden soll.
 - **AlfabetStereotype**: Geben Sie den Namen des Objektklassenstereotyps der Klasse `Service` in Alfabet ein, der dem im XML-Element `CentraSiteAssetType` spezifizierten CentraSite-Asset-Typ zugeordnet werden soll.
 - **HandoverStatus**: Spezifizieren Sie den Release-Status, über den der auf dem Objektklassenstereotyp im XML-Attribut `AlfabetStereotype` basierende technische Service in Alfabet verfügen muss, um in CentraSite aktualisiert zu werden. Weitere Informationen zum Konfigurieren von Release-Status für die CentraSite-Interoperabilität finden Sie im Abschnitt *Konfigurieren von Release-Status-Definitionen für Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- 7) In the toolbar, click the **Save**  button to save the XML definition.

Importieren von CentraSite-Daten über ADIF-Schemata

Zur Synchronisierung von Alfabet mit CentraSite hinsichtlich der Organisationen und Anwender, die Eigentümer von CentraSite-Assets sind, wie auch zur Datenerfassung für CentraSite-Assets stehen mehrere vordefinierte ADIF-Importschemata zur Verfügung. Die folgenden ADIF-Schemata sind für die Interoperabilität zwischen Alfabet und CentraSite relevant:

- **Alfabet_CentraSite_Organization_Synchronization**: Dieses ADIF-Schema sollte ausgeführt werden, wenn die Interoperabilität mit CentraSite erstmals implementiert wird, um alle Eigentümerorganisationen von relevanten CentraSite-Assets in Alfabet zu importieren.
- **Alfabet_CentraSite_User_Synchronization**: Dieses ADIF-Schema sollte ausgeführt werden, wenn die Interoperabilität mit CentraSite erstmals implementiert wird, um alle Eigentümer von relevanten CentraSite-Assets in Alfabet zu importieren.

- `Alfabet_CentraSite_Asset_Synchronization`: Dieses ADIF-Schema sollte in regelmäßigen Abständen ausgeführt werden, um die Informationen zu CentraSite-Asset-Typen in Alfabet zu aktualisieren.

Detaillierte Informationen zum Ausführen vordefinierter ADIF-Schemata zur Synchronisierung des CentraSite-Repositorys mit Alfabet finden Sie im Abschnitt *Vordefinierte ADIF-Schemata* im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

Kapitel 6: Konfiguration der Interoperabilität mit webMethods API Portal

Software AG stellt die Interoperabilität zwischen den Alfabet -Produkten und webMethods® API Portal zur Verfügung, damit Ihr Unternehmen APIs in der Alfabet -Benutzeroberfläche als geplante technische Services überprüfen und testen kann. Das Produkt webMethods API Portal dient als Portal zum Entdecken und Bewerten von APIs. Sie können die Interoperabilität zwischen Alfabet und dem webMethods API Portal konfigurieren und so integrierte Funktionen bereitstellen, um APIs als geplante technische Services in Alfabet zu importieren, diese geplanten technischen Services zu dokumentieren und zu beschreiben und die geänderte Definition des technischen Services wieder in das API-Portal zu exportieren.

APIs können über die Ansicht **Technische Service-Registry-Services** (COMSR_Services) oder die Ansicht **Technische Service-Registry-Services - gefiltert** (COMSR_ServicesExt) in die Benutzeroberfläche von Alfabet importiert werden. Sobald die technischen Services einen bestimmten Release-Status haben, können Sie in der Benutzeroberfläche über die Ansicht **Technische Services** (COM_TechServices) von Alfabet in das API-Portal exportiert werden. Per Klick auf eine Schaltfläche können Anwender von Alfabet zu webMethods API Portal navigieren und die API-Definition direkt in der Benutzeroberfläche des API-Portals anzeigen.



APIs können auch mithilfe eines ADIF-Importschemas in Alfabet importiert werden. Technische Services können nicht über ein ADIF-Schema in webMethods API Portal 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. Weitere Informationen zum Ausführen des vordefinierten ADIF-Schemas `Alfabet_APIportal_Synchronization` und zum Verarbeiten der Daten in den temporären Tabellen finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

APIs und Ihre Referenzen werden vordefinierten Objektklassen in Alfabet zugeordnet. Wenn ein API über die Ansichten in der Alfabet -Benutzeroberfläche in Alfabet importiert wird, wird in Alfabet ein technischer Service erzeugt. Wenn `Operations`, `Resources`, `Methods` und `Method Parameters` für das API angegeben sind, werden entsprechend die Operation des technischen Services, Businessdaten, die Operationsmethode des technischen Services und die Parameter der Operationsmethode des technischen Services in Alfabet erzeugt. Die Zuordnung wird in der nachfolgenden Tabelle beschrieben:

API-Portal-Objekt	Alfabet-Objektklasse
API	Technischer Service (<code>Service</code>) Bitte beachten Sie, dass die zusätzliche Spezifikation der Zuordnung von APIs zu technischen Services im XML-Objekt APIportalConfig konfiguriert werden muss.
Operation	Operation des technischen Service (<code>ServiceOperation</code>)
Resource	Businessdaten (<code>BusinessData</code>)

API-Portal-Objekt	Alfabet-Objektklasse
	Bitte beachten Sie, dass die zusätzliche Spezifikation der Zuordnung von <code>Resources</code> zu Businessdaten im XML-Objekt <code>ServiceResourceMapping</code> konfiguriert werden muss.
Method	Operationmethode des technischen Service (<code>ServiceOperationMethod</code>)
Method Parameter	Parameter der Operationmethode des technischen Service (<code>ServiceOperationMethodParameter</code>)

Um die Interoperabilität mit webMethods API Portal zu konfigurieren, stehen Ihnen die folgenden Informationen zur Verfügung:

- [Übersicht über die für webMethods API Portal erforderliche Konfiguration](#)
- [Konfigurieren des Klassenmodells für Interoperabilität mit webMethods API Portal](#)
- [Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Portal](#)
- [Konfiguration der Zuordnung von API-Portal-Ressourcen zu Businessdaten](#)

Übersicht über die für webMethods API Portal erforderliche Konfiguration

Nachfolgend finden Sie einen Überblick über alle Konfigurationsschritte, die für die Interoperabilität mit webMethods® API Portal erforderlich sind:

- 1) Konfigurieren Sie Objektklassenstereotypen für die Objektklassen `Service`, `ServiceOperation` und `ServiceOperationMethod`. Später ordnen Sie die Objektklassenstereotypen der Klasse `Service` den im API-Portal verfügbaren API-Assets zu. Dies wird unter [Konfigurieren des Klassenmodells für Interoperabilität mit webMethods API Portal](#) im Detail beschrieben.
- 2) Ordnen Sie Objektklassenstereotypen für die Objektklassen `Service`, `ServiceOperation` und `ServiceOperationMethod` im XML-Objekt **`TechServiceManager`** zu. Geben Sie für jeden technischen Service-Stereotyp an, welche Operationen des technischen Services zulässig sind. Geben Sie für jeden Operations-Stereotyp des technischen Services an, welche Methoden der Operation des technischen Services zulässig sind. Informationen dazu finden Sie unter *Konfigurieren von Objektklassenstereotypen für technische Services* im Kapitel *Konfigurieren von in der Lösungsumgebung implementierten alfabet-Funktionalitäten* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- 3) Konfigurieren Sie Release-Status für die Objektklasse `Service` sowie die Objektklassenstereotypen im XML-Objekt **`ReleaseStatusDefs`**. Diese Release-Status sollten den in webMethods API Portal verwendeten Status entsprechen. Bitte beachten Sie, dass der Release-Status `HandoverStatus` angegeben werden muss, damit die technischen Services in Alfabet in webMethods API Portal exportiert werden können. Dies wird unter [Konfigurieren des Klassenmodells für Interoperabilität mit webMethods API Portal](#) im Detail beschrieben.

- 4) Geben Sie im XML-Objekt **APIPortalConfig** eine oder mehrere Verbindungen zu webMethods API Portal an. Hier können Sie auch angeben, ob der Import über die Alfabet Benutzeroberfläche oder über ein ADIF-Importschema erfolgt, wie die API-Assets den für die Objektklasse `Service` konfigurierten Objektklassenstereotypen zugeordnet werden und welchen Übergabestatus die technischen Services haben müssen, um in webMethods API Portal exportiert zu werden. Dies wird unter [Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Portal](#) im Detail beschrieben.
- 5) Wenn zum Konfigurieren der Verbindung im XML-Objekt **APIPortalConfig** Servervariablen verwendet werden, müssen die Servervariablen für den Server-Alias im Alfabet Administrator spezifiziert werden. Weitere Informationen zum Konfigurieren von Servervariablen finden Sie im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).
- 6) Geben Sie die Zuordnung der den `Resources` zugehörigen Daten zu den Objekten der Klasse **Businessdaten** an. Dies wird im XML-Objekt **ServiceResourceMapping** angegeben. Dies wird unter [Konfiguration der Zuordnung von API-Portal-Ressourcen zu Businessdaten](#) im Detail beschrieben.
- 7) Die Attribute **Assembly** und **Assembly-Klasse**, die für den Lösungsmanager **ServiceRegistryManager** verfügbar sind, müssen spezifiziert werden, um Interoperabilität zu gewährleisten. Das XML-Objekt **ServiceRegistryManager** ist in Alfabet Expand verfügbar. Folgende Werte müssen spezifiziert werden:
 - **Assembly:** `ITPlan`
 - **Assembly-Klasse:** `ITPlanSolution.GenericServiceRegistryManager`
- 8) Geben Sie an, welche an technische Services angehängten Dokumenttypen in Alfabet ebenfalls in das API-Portal exportiert werden können. Mit der geschützten Aufzählung **APIPortalDocTypes** können Sie die Dateierweiterungen anpassen, die in der Funktionalität **Integrationslösungen** im Editor **API-Portal-Verbindung** im Feld **Gemeinsam genutzte Dokumententypen** verfügbar sind. Für die von Ihnen definierten Dateierweiterungen gelten die Spezifikationen der Blacklist/Whitelist von Dateierweiterungen, die im XML-Objekt **FileExtensionLists** definiert sind. Informationen hierzu finden Sie im Abschnitt *Festlegen der zulässigen Dateinamenerweiterungen für das Hochladen/Herunterladen von Dateien* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- 9) Erzeugen Sie in der Ansicht **API-Portal-Datenbankverbindung** der Funktionalität **Konfiguration von Integrationslösungen** in der Alfabet -Benutzeroberfläche API-Portal-Verbindungsobjekte. Für jede im XML-Objekt **APIPortalConfig** konfigurierte API-Portal-Verbindung sollte ein API-Portal-Verbindungsobjekt erzeugt werden. Die API-Portal-Verbindungsobjekte ermöglichen es einem Anwender, die Verbindung zu der relevanten API-Portal-Instanz anzugeben, die beim Exportieren/Synchronisieren eines technischen Service auf der Ansichtseite **Technische Services** (`COM_TechServices`) verwendet werden soll. Sie müssen außerdem angeben, ob Alfabet die technischen Services automatisch synchronisieren soll, wenn die Ansichtseite **Technische Services** geladen wird. Das Erzeugen von API-Portal-Verbindungsobjekten wird im Abschnitt *Konfigurieren von semantischen Verbindungen für Integrationslösungen* im Referenzhandbuch *Konfigurieren von Bewertungen und Referenzdaten in Alfabet* beschrieben.
- 10) Die relevanten Anwenderprofile müssen so konfiguriert werden, dass sie die Ansichtseite **Technische Service-Registry-Services – gefiltert** (`COMSR_ServicesExt`), die im Root-Knoten des Explorers **Komponenten** (`COM_Explorer`) verfügbar ist, und die Ansichtseite **Technische Services** (`COM_TechServices`), die in der Objektansicht **Komponenten** verfügbar ist, enthalten. (Diese Ansicht ist nur für Komponenten des Typs `Service` verfügbar.) Die Ansichtseite **Technische Service-Registry-Services** (`COMSR_Services`), die im Root-Knoten des Explorers **Komponenten** (`COM_Explorer`) verfügbar ist, sollte für die Anwenderprofile, die auf webMethods API Portal zugreifen, nicht sichtbar sein. Weitere Informationen über das Konfigurieren von

Anwenderprofilen finden Sie im Kapitel *Konfigurieren von Anwenderprofilen für die Anwender-Community* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

- 11) Wenn APIs mit dem vordefinierten ADIF-Importschema `Alfabet_APIPortal_Synchronization` in Alfabet importiert werden sollen, müssen Sie das vordefinierte ADIF-Schema `Alfabet_APIPortal_Synchronization` auslösen, um die APIs in temporäre Datenbanktabellen zu importieren. Ihr Unternehmen muss die Aktualisierung der Alfabet-Datenbank -Tabellen mit den Daten aus den temporären Datenbanktabellen konfigurieren. Dies wird im Referenzhandbuch *Alfabet-Datenintegrationsframework* ausführlicher beschrieben.

Konfigurieren des Klassenmodells für Interoperabilität mit webMethods API Portal

Folgende Konfiguration ist für die Interoperabilität mit webMethods® API Portal erforderlich:

- Konfigurieren von Objektklassenstereotypen für die Objektklasse `Service`. Jeder API-Asset-Typ, der in Alfabet angezeigt werden soll, muss einem Objektklassenstereotyp der Klasse `Service` im XML-Element ***AssetTypeMappings*** des XML-Objekts ***APIPortalConfig*** zugeordnet werden. Daher müssen Sie die relevanten Objektklassenstereotypen für die Klasse `Service` konfigurieren, bevor Sie das XML-Objekt ***APIPortalConfig*** spezifizieren können. Ausführlichere Informationen zum Konfigurieren von Objektklassenstereotypen finden Sie im Abschnitt *Konfigurieren von Objektklassenstereotypen für Objektklassen* im Kapitel *Konfigurieren des Klassenmodells* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Die Objektklassenstereotypen für die Objektklasse `Service` könnten beispielsweise folgendermaßen angegeben werden:

```
<ClassStereotypes>
  <Stereotype Name="SOAPService" Caption="SOAP Service"
    CaptionPlural="SOAP Services" Comments="" HasMandates="false" />
  <Stereotype Name="DataService" Caption="DataService"
    CaptionPlural="Data Services" Comments="" HasMandates="false" />
  <Stereotype Name="RESTService" Caption="REST Service"
    CaptionPlural="REST Services" Comments="" HasMandates="false" />
  <Stereotype Name="XMLService" Caption="XML Service"
    CaptionPlural="XML Services" Comments="" HasMandates="false" />
</ClassStereotypes>
```

- Konfigurieren von Objektklassenstereotypen für die Objektklasse `ServiceOperation`. Jede einem importierten API zugehörige `Operation` wird in Alfabet als Operation des technischen Services erzeugt.
- Konfigurieren von Objektklassenstereotypen für die Objektklasse `ServiceOperationMethod`. Jede einem importierten API zugehörige `Method` wird in Alfabet als Operationsmethode des technischen Services erzeugt.
- Konfigurieren Sie die Zuordnung der Stereotypen der technischen Services (basierend auf der Klasse `Service`), der Stereotypen der Operationen des technischen Services (basierend auf der Klasse `ServiceOperation`) und die Stereotypen der Operationsmethoden des technischen Services (basierend auf der Klasse `ServiceOperationMethod`) im XML-Objekt ***TechServiceManager***. Geben Sie für jeden technischen Service-Stereotyp an, welche Operationen des technischen Services zulässig sind. Geben Sie für jeden Operations-Stereotyp des technischen Services an, welche Methoden der Operation des technischen Services zulässig sind. Weitere Informationen zur Zuordnung von Objektklassenstereotypen, die für technische Services

und Operationen des technischen Services definiert wurden, finden Sie im Abschnitt *Konfigurieren von Objektklassenstereotypen für technische Services* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

- Konfigurieren Sie Release-Status für die Objektklasse `Service` sowie auch alle relevanten Objektklassenstereotypen für die Objektklasse `Service`. Jeder Asset-Typ in API Portal hat einen Status. Daher sollte jedes Objektklassenstereotyp über entsprechende Release-Status verfügen, die den Status des Asset-Typs entsprechen, die ihm zugeordnet werden. Bitte beachten Sie Folgendes:
 - Der Satz an Release-Status für jeden relevanten Objektklassenstereotyp muss im XML-Objekt **ReleaseStatusDefs** konfiguriert werden. Für die Objektklasse `Service` als Ganzes sowie für die einzelnen Objektklassenstereotypen muss eine Release-Status-Definition erzeugt werden. Die Release-Status-Definition für die Objektklasse muss den kompletten Satz an Release-Status enthalten, die für ihre Objektklassenstereotypen konfiguriert wurden. Die Release-Status-Definition für einen Objektklassenstereotyp darf nur die Release-Status enthalten, die für den Objektklassenstereotyp relevant sind, sowie eine Abfolge der Release-Status, die zum Erreichen eines bestimmten Ziel-Release-Status verfügbar sind. Weitere Informationen zum Konfigurieren von Release-Status finden Sie im Abschnitt *Konfigurieren von Release-Status-Definitionen für Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
 - Die Release-Status in Alfabet sollten in der primären Sprache spezifiziert werden, die der Definition der Basis-Landeseinstellungen der standardmäßigen Landeseinstellung entspricht. Wenn ein Anwender versucht, Daten in Alfabet mit API-Portal-Assets zu synchronisieren, und für ein oder mehrere dieser API-Portal-Assets Status in einer anderen Sprache als der primären Sprache in Alfabet definiert sind, wird eine Fehlermeldung angezeigt.
 - Für jeden für einen Asset-Typ verfügbaren Status muss für das relevante Objektstereotyp ein zugehöriger Release-Status definiert sein. Der Name des Release-Status muss mit dem Namen des API-Portal-Status übereinstimmen.
 - Der für den Objektklassenstereotyp verfügbare Satz von Release-Status kann Release-Status beinhalten, die für die Verwaltung der technischen Services in alfabet relevant sind. Solche Release-Status werden als Alfabet -Eigentum betrachtet, da sie nur in Alfabet definiert werden.
 - Die in API Portal definierten Status, die auch Eigentum von API Portal sind, sollten dieselben Namen aufweisen, wie die in Alfabet definierten Release-Status.
 - Der Satz der Release-Status, die API-Portal-Eigentum sind, sollte in Alfabet nicht bearbeitbar sein. Diese Status (wie Entwerfen, Entwurf genehmigt, Test usw.) sollten nur in API Portal definiert werden.

Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Portal

Es können mehrere Verbindungen zu webMethods® API Portal definiert werden, um den Anforderungen von Partner-Organisationen gerecht zu werden. Über das XML-Objekt **APIPortalConfig** können Sie die Verbindungen zu webMethods API Portal-Instanzen konfigurieren und festlegen, wie die APIs in Alfabet importiert werden sollen. Sie können die erforderlichen Informationen konfigurieren, damit Anwender APIs

über die Alfabet -Benutzeroberfläche in Alfabet importieren und exportieren können, und/oder Sie können konfigurieren, wie die APIs über ein ADIF-Importschema importiert werden. Beachten Sie Folgendes:

- APIs können über die Ansicht **Technische Service-Registry-Services** (COMSR_Services) oder die Ansicht **Technische Service-Registry-Services – gefiltert** (COMSR_ServicesExt) importiert werden. Wenn der Import von APIs über die Alfabet -Benutzeroberfläche erfolgen soll, müssen Sie die API-Assets den relevanten Objektklassenstereotypen zuordnen, die für die Klasse `Service` im XML-Objekt **APIPortalConfig** konfiguriert sind. Sie müssen auch den relevanten Release-Status `HandoverStatus` angeben, den ein technischer Service haben muss, damit er aus Alfabet in das API-Portal exportiert wird.
- APIs können auch mithilfe eines ADIF-Importschemas in Alfabet importiert werden. Technische Services können nicht über ein ADIF-Schema in webMethods API Portal 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. Weitere Informationen zum Konfigurieren des vordefinierten ADIF-Schemas `Alfabet_APIPortal_Synchronization` finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.



Wenn Sie in das XML-Objekt **APIPortalConfig** eine Zeichenfolge eingeben möchten, die Sonderzeichen enthält, müssen Sie diese durch den entsprechenden kompatiblen HTML-Code ersetzen, zum Beispiel:

- `>` für `>`
- `<` für `<`
- `"` für `"`
- `[` für `[`
- `]` für `]`

Die Konfiguration der Verbindungen wird im XML-Objekt **APIPortalConfig** über das Root-XML-Element **APIPortalConfig** und dem diesem untergeordneten XML-Element **APIPortalConnections** vorgenommen. Sobald Sie die Konfiguration des XML-Objekts **APIPortalConfig** abgeschlossen haben, müssen Sie in der Ansicht **API-Portal-Datenbankverbindung**, die in der Funktionalität **Integrationslösungskonfigurationen** in der Alfabet -Benutzeroberfläche verfügbar ist, ebenfalls Verbindungsobjekte erzeugen.

Das Root-XML-Element **APIPortalConfig** verfügt über zwei untergeordnete Elemente:

- **Proxy**: Über dieses XML-Element können Sie konfigurieren, wie Anforderungen über einen Proxyserver an Instanzen von webMethods API Portal gesendet werden. Die Konfiguration des XML-Elements **Proxy** wird im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#) beschrieben.
- **APIPortalConnections**: Über dieses XML-Element können Sie die Verbindung zu Instanzen von webMethods API Portal konfigurieren. Für jede Verbindung zu einer webMethods API Portal-Instanz sollte ein XML-Element **APIPortalConnection** konfiguriert werden. Jedes XML-Element verfügt über zwei untergeordnete Elemente. Bei diesen untergeordneten Elementen handelt es sich um das Element **AssetTypeMappings**, mit dem Sie die Zuordnung von APIs aus dem API-Portal zu Alfabet angeben können, wenn sie über die Alfabet -Benutzeroberfläche importiert

werden, und um das Element **DataConnectivityInfo**, mit dem Sie den Import von APIs aus dem API-Portal in Alfabet angeben können, wenn sie über ein ADIF-Importschema importiert werden.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist eine Vorlage über das Attribut **XML-Vorlage** des XML-Objekts verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf. Bei einer Beispielformatierung müssen Sie die vorhandenen XML-Elemente bearbeiten anstatt sie der XML-Konfiguration hinzuzufügen.

Sie können mehrere Verbindungen von Alfabet zu mehreren Instanzen von webMethods API Portal konfigurieren. So definieren Sie eine Verbindung zu einer Instanz von webMethods API Portal.

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte** > **IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf das XML-Objekt **APIPortalConfig**, und wählen Sie **XML bearbeiten** aus. Das XML-Objekt wird geöffnet.



Über das XML-Element **Proxy** können Sie konfigurieren, wie Anforderungen über einen Proxyserver an Instanzen von webMethods API Portal gesendet werden. Dies ist optional. Wenn Sie mehrere Verbindungen konfiguriert haben und jede Verbindung einen anderen Proxy verwenden soll, können Sie Ihrer Proxykonfiguration zusätzliche Proxys hinzufügen und die entsprechende Proxydefinition in der Verbindungskonfiguration für das API-Portal angeben. Die Konfiguration des XML-Elements **Proxy** wird im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#) beschrieben.

- 3) Fügen Sie dem Root-XML-Element **APIPortalConfig** ein untergeordnetes XML-Element **APIPortalConnections** hinzu.
- 4) Fügen Sie dem XML-Element **APIPortalConnections** für jede Datenbankverbindungsinstanz, die Sie definieren, ein untergeordnetes XML-Element **APIPortalConnection** hinzu.
- 5) Konfigurieren Sie eine Verbindung zu jeder API-Portal-Instanz, indem Sie die folgenden XML-Attribute für das XML-Element **APIPortalConnection** festlegen:
 - **Name**: Geben Sie einen eindeutigen Namen für die Verbindung zum API-Portal ein.
 - **ServerURL**: Geben Sie die URL für den Zugriff auf das API-Portal oder den relevanten Servervariablenamen ein.



In einigen XML-Attributen können Servervariablen verwendet werden, um den Wert des Attributs zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation auszulesen, wenn eine Verbindung zum API-Portal hergestellt wurde. Weitere Informationen über die Verwendung von Servervariablen finden Sie im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).

- **UserName**: Geben Sie den Anwendernamen Ihres Unternehmens für den Zugriff auf das API-Portal oder den relevanten Servervariablenamen ein.
- **Password**: Geben Sie das Passwort Ihres Unternehmens für den Zugriff auf das API-Portal oder den relevanten Servervariablenamen ein.
- **ServiceViewURL**: Geben Sie das Präfix für den vollständigen Link ein, der bei einem Klick eines Anwenders auf die Schaltfläche **Ansicht des Services in Service-Registry öffnen** auf der Ansichtssseite **Technische Service-Registry-Services – gefiltert** zum Navigieren von Alfabet zum API-Portal verwendet wird. Geben Sie die URL für den Zugriff auf die

Serviceansicht im API-Portal oder den relevanten Servervariablenamen ein. Beispiel:

`http://centrasite.company.com:53307/BusinessUI/#assetdetail`)

- **TransactionHistoryPeriod:** Geben Sie die Anzahl der Tage ein, für die die Transaktionen abgerufen werden. Dies ist nur bei Massenimporten über ADIF erforderlich.
 - **Timeout:** Geben Sie die Anzahl von Minuten Anwenderinaktivität ein, nach der die Verbindung zum API-Portal beendet werden soll.
- 6) Wenn APIs über die Alfabet -Benutzeroberfläche in Alfabet importiert werden sollen, fügen Sie jedem XML-Element **APIPortalConnection** ein XML-Element `AssetTypeMappings` hinzu. Geben Sie für jeden Asset-Typ, den Sie in Alfabet importieren möchten, ein untergeordnetes XML-Element `AssetTypeMapping` an.
- 7) Konfigurieren Sie die Zuordnung jedes Asset-Typs, indem Sie die folgenden XML-Attribute für das XML-Element `AssetTypeMapping` festlegen:
- **APIPortalAssetType:** Geben Sie den Namen des webMethods API Portal-Asset-Typs ein, der dem im XML-Element `AlfabetStereotype` spezifizierten Objektklassenstereotyp zugeordnet werden soll.
 - **HandoverStatus:** Spezifizieren Sie den Release-Status, über den der auf dem Objektklassenstereotyp im XML-Attribut `AlfabetStereotype` basierende technische Service in Alfabet verfügen muss, um in der Instanz von webMethods API Portal aktualisiert zu werden. Weitere Informationen zum Konfigurieren von Release-Status für die Interoperabilität mit webMethods API Portal finden Sie im Abschnitt *Konfigurieren von Release-Status-Definitionen für Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
 - **AlfabetStereotype:** Geben Sie den Namen des Objektklassenstereotyps der Klasse `Service` in Alfabet ein, der dem im XML-Element `APIPortalAssetType` spezifizierten webMethods API Portal-Asset-Typ zugeordnet werden soll.
- 8) Falls APIs über ein ADIF-Importschema in Alfabet importiert werden sollen, fügen Sie jedem XML-Element **APIPortalConnection** ein XML-Element `DataImport` mit einem untergeordneten XML-Element `DataConnectivityInfo` hinzu.
- 9) Geben Sie für jede Importdefinition, die Sie konfigurieren möchten, ein untergeordnetes XML-Element `DataConnection` und ein XML-Attribut `Name` an. Geben Sie den Namen der Datenverbindung ein.
- 10) Geben Sie für jeden Importtyp in webMethods API Portal, den Sie über ADIF importieren möchten, ein XML-Element `Import` mit einem XML-Attribut `Type` ein. Geben Sie den Importtyp ein, der über das ADIF-Importschema importiert werden kann. Zulässige Importtypen: `API`, `PROVIDER`, `EVENT`, `PACKAGE` und `PLAN`.
- 11) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die

Schaltfläche **Speichern**  .

Konfiguration der Zuordnung von API-Portal-Ressourcen zu Businessdaten

Ein API, das aus dem webMethods API Portal in Alfabet importiert wird, wird auf der Basis des im XML-Objekt **APIPortalConfig** konfigurierten Stereotyps als technischer Service erzeugt. Beim Importieren des APIs

können auch die zugehörigen Ressourcen importiert werden. In diesem Fall kann für jede `Resource` eine Instanz der Objektklasse `BusinessData` aktualisiert oder erzeugt werden. Dazu müssen Sie die zu verwendende Logik zuordnen, um die entsprechenden Businessdaten in Alfabet zu finden und diese Businessdaten mit Informationen über die `Resource` zu aktualisieren oder, wenn nötig, neue Businessdaten zu erzeugen und Informationen über die `Resource` hineinzuschreiben. Für die Suche nach Businessdaten bzw. deren Erzeugung können mehrere Zuordnungslogiken konfiguriert werden.



Wenn `Methods` und `Parameters` für eine `Resource` definiert sind, werden sie auch zu den Businessdaten in Alfabet hinzugefügt. Für jede `Method`, die für eine `Resource` angegeben ist, wird eine Methode der Operation des technischen Services (Instanz der Klasse `ServiceOperationMethod`) erzeugt, und für jeden `Parameter` wird ein Parameter für die Methode der Operation des technischen Services (Instanz der Klasse `ServiceOperationMethodParameter`) erzeugt. Beachten Sie, dass die Methode der Operation des technischen Services (Instanz der Klasse `ServiceOperationMethod`) auf Basis eines Objektklassenstereotypen erzeugt wird, der im XML-Objekt **TechServiceManager** angegeben ist. Informationen hierzu finden Sie im Abschnitt *Konfigurieren von Objektklassenstereotypen für technische Services* im Kapitel *Konfigurieren von in der Lösungsumgebung implementierten alfabet-Funktionalitäten*.

Das Folgende muss im XML-Objekt **ServiceResourceMapping** konfiguriert werden.

- Geben Sie eine oder mehrere `ResourceMapping`-Definitionen an, um vorhandene Businessdaten zu finden, oder um vorhandene Business-Objekte zu finden, für die neue Businessdaten erzeugt werden. Die Zuordnungslogik ist in hohem Maße konfigurierbar.



Sie könnten zum Beispiel folgende Zuordnungslogiken eingeben:

- **Priorität 1:** Finden Sie einen vorhandenen Businessdatensatz, der allen Eigenschaften entspricht, die in der Ressourcen-Zuordnung angegeben sind, und aktualisieren Sie diesen. Finden Sie zum Beispiel einen Businessdatensatz mit der gleichen Eigenschaft **Name** wie die `Resource`.
 - **Priorität 2:** Finden Sie einen vorhandenen Businessdatensatz, der einigen der Eigenschaften entspricht, die in der Ressourcen-Zuordnung angegeben sind, und aktualisieren Sie diesen. Finden Sie zum Beispiel einen Businessdatensatz mit teilweise der gleichen Eigenschaft **Name** wie die `Resource`.
 - **Priorität 3:** Erzeugen Sie einen neuen Businessdatensatz für ein Business-Objekt, das allen Eigenschaften in der Ressourcen-Zuordnung entspricht. Finden Sie zum Beispiel ein vorhandenes Business-Objekt mit der gleichen Eigenschaft **Name** wie die `Resource` und erzeugen Sie einen Businessdatensatz für dieses Business-Objekt.
 - **Priorität 4:** Erzeugen Sie einen neuen Businessdatensatz für ein Business-Objekt, das einigen Eigenschaften in der Ressourcen-Zuordnung entspricht. Finden Sie zum Beispiel ein vorhandenes Business-Objekt mit teilweise der gleichen Eigenschaft **Name** wie die `Resource` und erzeugen Sie einen Businessdatensatz für dieses Business-Objekt.
- Geben Sie an, wie die Datentypen (wie beispielsweise `string`, `integer` usw.) für `Resources` den Eigenschaftstypen für die Objektklasseneigenschaftstypen für die Klasse `BusinessData` zugeordnet werden. Im Alfabet -Klassenmodell hat jede Objektklasseneigenschaft ein Attribut **Eigenschaftstyp** mit einem Wert wie z. B. `String`, `StringArray`, `Integer`, `Real`, `Text` usw. Zum Beispiel haben die Eigenschaften **Name** und **Version** beide den **Eigenschaftstyp** = `String`. Sie müssen diese Zuordnung für die Businessdaten angeben, die im Kontext jedes

Objektklassenstereotypen der bei der Integration des API-Portals verwendeten Klasse `Service` erzeugt/aktualisiert wurden.

- Geben Sie an, in welcher Weise die Datentypen (wie beispielsweise `string`, `integer` usw.) für `Resources`, die den Attributtypen der Businessdaten zugeordnet sind, für Alfabet -Businessdaten relevant sind. Businessdaten-Attribute sind Attribute, die die Businessdaten beschreiben. Businessdaten-Attribute werden in der Eigenschaft `Attributes` der Klasse `BusinessData` erfasst. Das Attribut `Type` der Klasse `BusinessDataAttribute` wird von Ihrem Lösungsentwickler für ihre Lösungskonfiguration über die geschützte Aufzählung **`BusinessDataAttributeType`** konfiguriert. Jedes Aufzählungselement, das für die geschützte Aufzählung **`BusinessDataAttributeType`** definiert ist, stellt einen Typen eines Businessdaten-Attributs dar.

So spezifizieren Sie das XML-Objekt **`ServiceResourceMapping`** für die Integration mit webMethods API Portal:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf das XML-Objekt **`ServiceResourceMapping`**, und wählen Sie **XML bearbeiten** aus. Das XML-Objekt wird geöffnet.
- 3) Die Namen der `Resource` in webMethods API Portal können ein oder mehrere Präfixe enthalten, die Sie vielleicht aus der Zuordnung ausschließen möchten. Fügen Sie dem Root-XML-Element **`ServiceResourceMapping`** ein untergeordnetes XML-Element **`PrefixesToIgnoreOnResourceMapping`** hinzu, um die zu ignorierenden Präfixe anzugeben.
- 4) Fügen Sie dem XML-Element **`PrefixesToIgnoreOnResourceMapping`** für jedes Präfix, das Sie angeben möchten, ein untergeordnetes XML-Element **`Value`** hinzu. Geben Sie das Präfix, das ignoriert werden soll, in das XML-Element **`Value`** ein.



Eine `Resource` mit dem Namen `Employee` kann beispielsweise einen Namen haben, der den Pfad im API-Portal enthält, auf dem sich die `Resource` befindet. Der Name kann beispielsweise wie ein Ressourcenpfad aussehen: `<StoreName>/<API-Name>/Employee`. In diesem Fall möchten Sie vielleicht den `ResourceEmployee` als Businessdaten zu A hinzufügen, dabei aber die Präfixe `Store` und `API` ignorieren. In einem solchen Fall würden Sie Folgendes angeben:

```
...
<PrefixesToIgnoreOnResourceMapping>
  <Value>store</Value>
  <Value>api</Value>
</PrefixesToIgnoreOnResourceMapping>
```

- 5) Für jede Regel, um die `Resource` Businessdaten zuzuordnen, erzeugen Sie ein XML-Element **`ResourceMapping`**. Konfigurieren Sie für das XML-Element **`ResourceMapping`** die folgenden XML-Attribute:
 - **`Class`**: Geben Sie `BusinessData` ein, um nach vorhandenen Businessdaten für die Zuordnung zu suchen. Geben Sie `BusinessObject` ein, um nach vorhandenen Business-Objekten zu suchen, für die neue Businessdaten erzeugt werden sollen. Im Kontext dieser Konfiguration können keine neuen Business-Objekte erzeugt werden.
 - **`Priority`**: Geben Sie eine Ganzzahl ein, um die Priorität dieser Ressourcenzuordnung festzulegen. Zuerst wird nach einer Ressourcenzuordnungsdefinition mit `Priority="1"` gesucht, als Zweites wird nach `Priority,="2"` gesucht usw.

- **Properties:** Geben Sie eine oder mehrere Objektklasseneigenschaften der Businessdaten/des Business-Objekts an, die für eine Übereinstimmung gesucht werden müssen. Beim Namen der `Resource` wird im Normalfall die folgende Syntax verwendet: `{Name}\{Property1}\{Property2}`. Sie können mehrere Objektklassen angeben, um die Suche nach einer Übereinstimmung zu erleichtern. Zum Beispiel: `Properties="{Name}/{Version}"`.
 - **MatchType:** Geben Sie `FullString` ein, wenn alle Werte im XML-Attribut `Properties` übereinstimmen müssen, um die Businessdaten oder das Business-Objekt zu finden. Geben Sie `SubString` ein, wenn nur einige der Werte im XML-Attribut `Properties` übereinstimmen müssen, um die Businessdaten oder das Business-Objekt zu finden. Wird keine Übereinstimmung gefunden, wird die `Resource` ignoriert, und die Businessdaten werden nicht aktualisiert/erzeugt.
- 6) Um die Informationen zu Dateneigenschaften für Businessdaten zuzuordnen, erstellen Sie ein XML-Element ***DataTypeMappings***.
 - 7) Erzeugen Sie für jeden Objektklassenstereotyp der Klasse `Service`, der im XML-Element `AlfabetStereotype` des XML-Objekts ***APIPortalConfig*** angegeben ist, ein XML-Element ***DataTypeMapping***, das dem XML-Element ***DataTypeMappings*** untergeordnet ist.
 - 8) Erzeugen Sie im XML-Element ***DataTypeMapping*** ein XML-Attribut `AlfabetStereotype`, und geben Sie den Namen des für die Klasse `Service` konfigurierten Objektklassenstereotyps ein, das Sie angeben möchten.
 - 9) Erzeugen Sie im XML-Element ***DataTypeMapping*** für jeden Eigenschaftstyp, der für das im XML-Attribut angegebene `AlfabetStereotype` Objektklassenstereotyp relevant ist, ein XML-Element ***DataType***. Alfabet -Eigenschaftstypen sind `String`, `Boolean`, `Date`, `Integer`, `Text` usw. Der Eigenschaftstyp wird im Attribut ***Eigenschaftstyp*** für jede relevante Eigenschaft der Objektklasse `Service` angegeben. (Die Eigenschaften ***Name*** und ***Version*** für die Klasse `BusinessData` haben beispielsweise beide den ***Eigenschaftstyp*** = `String`.) Konfigurieren Sie für das XML-Element ***DataType*** die folgenden XML-Attribute:
 - ***AlfabetPropertyType***: Geben Sie den Typ für die Alfabet -Objektklasseneigenschaft (`String`, `Boolean`, `Date`, `Integer`, `Text` usw.) ein, der dem Datentyp des im XML-Attribut `ExternalPropertyType` angegebenen APIs zugeordnet werden soll.
 - ***ExternalPropertyType***: Geben Sie den Datentyp für das API ein, der dem im XML-Attribut `AlfabetPropertyType` definierten Alfabet -Objektklasseneigenschaftstyp zugeordnet werden soll.
 - 10) Erzeugen Sie im XML-Element ***DataTypeMapping*** für jeden Typ von Businessdaten-Attributen, der für Alfabet -Businessdaten relevant ist, ein XML-Element ***AttributeDataType***. Businessdaten-Attribute sind Attribute, die die Businessdaten beschreiben. Businessdaten-Attribute werden über die Eigenschaft `Attributes` der Klasse `BusinessData` erfasst. Das Attribut `Type` der Klasse `BusinessDataAttribute` wird von Ihrem Lösungsentwickler für Ihre Lösungskonfiguration über die geschützte Aufzählung ***BusinessDataAttributeType*** konfiguriert. Jedes Aufzählungselement, das für die geschützte Aufzählung ***BusinessDataAttributeType*** definiert ist, stellt einen Typen eines Businessdaten-Attributs dar. Konfigurieren Sie für das XML-Element ***AttributeDataType*** die folgenden XML-Attribute:
 - ***AlfabetAttributeType***: Geben Sie das für die geschützte Aufzählung ***BusinessDataAttributeType*** angegebene Aufzählungselement ein, das dem Datentyp `Resource` zugeordnet werden soll, der im Attribut `ExternalPropertyType` spezifiziert ist.
 - ***ExternalPropertyType***: Geben Sie den Datentyp `Resource` ein, der dem im XML-Attribut `AlfabetPropertyType` angegebenen Businessdaten-Attributtyp zugeordnet werden soll.

- 11) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Kapitel 7: Konfiguration der Interoperabilität mit webMethods API Gateway

Software AG stellt die Interoperabilität zwischen Alfabet und webMethods® API Gateway zur Verfügung, damit Ihr Unternehmen APIs in der Alfabet -Benutzeroberfläche als operative technische Dienste verwalten und überwachen kann. Das Produkt webMethods API Gateway bietet ein Richtlinien-Framework, um APIs zu verwalten und zu sichern. Sie können die Interoperabilität zwischen Alfabet und dem webMethods API Gateway konfigurieren und so integrierte Funktionen bereitstellen, um APIs als operative technische Services in Alfabet zu importieren, diese technischen Services zu dokumentieren und zu beschreiben und die geänderte Definition des technischen Services wieder in das API-Gateway zu exportieren.

APIs können über die Ansicht **Technische Service-Registry-Services** (COMSR_Services) oder die Ansicht **Technische Service-Registry-Services - gefiltert** (COMSR_ServicesExt) in die Benutzeroberfläche von Alfabet importiert werden. Sobald die technischen Services einen bestimmten Release-Status haben, können Sie in der Benutzeroberfläche über die Ansicht **Technische Services** (COM_TechServices) aus Alfabet in das API Gateway exportiert werden. Per Klick auf eine Schaltfläche können Anwender von Alfabet zu webMethods API Gateway navigieren und die API-Definition direkt in der Benutzeroberfläche des API-Gateways anzeigen.



APIs können auch mithilfe eines ADIF-Importschemas in Alfabet 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 ein ADIF-Schema 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. Weitere Informationen zu Konfiguration und Durchführung eines ADIF-Schemas finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

APIs und Ihre Referenzen werden vordefinierten Objektklassen in Alfabet zugeordnet. Wenn ein API über die Ansichten in der Alfabet -Benutzeroberfläche in Alfabet importiert wird, wird in Alfabet ein technischer Service erzeugt. Wenn `Operations`, `Resources`, `Methods` und `Method Parameters` für das API angegeben sind, werden entsprechend die Operation des technischen Services, Businessdaten, die Operationsmethode des technischen Services und die Parameter der Operationsmethode des technischen Services in Alfabet erzeugt. Die Zuordnung wird in der nachfolgenden Tabelle beschrieben:

API-Gateway-Objekt	Alfabet-Objektklasse
API	<p>Technischer Service (Service)</p> <p>Bitte beachten Sie, dass die zusätzliche Spezifikation der Zuordnung von APIs zu technischen Services im XML-Objekt APIGatewayConfig konfiguriert werden muss.</p>
Operation	<p>Operation des technischen Service (ServiceOperation)</p>
Resource	<p>Businessdaten (BusinessData)</p> <p>Bitte beachten Sie, dass die zusätzliche Spezifikation der Zuordnung von Resources zu Businessdaten im XML-Objekt ServiceResourceMapping konfiguriert werden muss.</p>

API-Gateway-Objekt	Alfabet-Objektklasse
Method	Operationmethode des technischen Service (<code>ServiceOperationMethod</code>)
Method Parameter	Parameter der Operationmethode des technischen Service (<code>ServiceOperationMethodParameter</code>)

Um die Interoperabilität mit webMethods API Gateway zu konfigurieren, stehen Ihnen die folgenden Informationen zur Verfügung:

- [Übersicht über die für webMethods API Gateway erforderliche Konfiguration](#)
- [Konfigurieren des Alfabet -Klassenmodells für Interoperabilität mit webMethods API Gateway](#)
- [Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Gateway](#)
- [Konfiguration der Zuordnung von API-Gateway-Ressourcen zu Businessdaten](#)

Übersicht über die für webMethods API Gateway erforderliche Konfiguration

Nachfolgend finden Sie einen Überblick über alle Konfigurationsschritte, die für die Interoperabilität mit webMethods® API Gateway erforderlich sind:

- 1) Konfigurieren Sie Objektklassenstereotypen für die Objektklassen `Service`, `ServiceOperation` und `ServiceOperationMethod`. Später ordnen Sie die Objektklassenstereotypen der Klasse `Service` den im API-Gateway verfügbaren API-Assets zu. Dies wird unter [Konfigurieren des Alfabet -Klassenmodells für Interoperabilität mit webMethods API Gateway](#) im Detail beschrieben.
- 2) Ordnen Sie Objektklassenstereotypen für die Objektklassen `Service`, `ServiceOperation` und `ServiceOperationMethod` im XML-Objekt **TechServiceManager** zu. Geben Sie für jeden technischen Service-Stereotyp an, welche Operationen des technischen Services zulässig sind. Geben Sie für jeden Operations-Stereotyp des technischen Services an, welche Methoden der Operation des technischen Services zulässig sind. Informationen dazu finden Sie unter *Konfigurieren von Objektklassenstereotypen für technische Services* im Kapitel *Konfigurieren von in der Lösungsumgebung implementierten alfabet-Funktionalitäten* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- 3) Konfigurieren Sie Release-Status für die Objektklasse `Service` sowie die Objektklassenstereotypen im XML-Objekt **ReleaseStatusDefs**. Diese Release-Status sollten den in webMethods API Gateway verwendeten Status entsprechen. Bitte beachten Sie, dass der Release-Status `HandoverStatus` angegeben werden muss, damit die technischen Services in Alfabet in webMethods API Gateway exportiert werden können. Dies wird unter [Konfigurieren des Alfabet -Klassenmodells für Interoperabilität mit webMethods API Gateway](#) im Detail beschrieben.
- 4) Geben Sie im XML-Objekt **APIGatewayConfig** eine oder mehrere Verbindungen zu webMethods API Gateway an. Hier können Sie auch angeben, ob der Import über die Alfabet -Benutzeroberfläche oder über ein ADIF-Importschema erfolgt, wie die API-Assets den für die Objektklasse `Service` konfigurierten Objektklassenstereotypen zugeordnet werden und welchen

Übergabestatus die technischen Services haben müssen, um in webMethods API Gateway exportiert zu werden. Dies wird unter [Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Gateway](#) im Detail beschrieben.

- 5) Wenn zum Konfigurieren der Verbindung im XML-Objekt **APIGatewayConfig** Servervariablen verwendet werden, müssen die Servervariablen für den Server-Alias im Alfabet Administrator spezifiziert werden. Weitere Informationen zum Konfigurieren von Servervariablen finden Sie im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).
- 6) Geben Sie die Zuordnung der den `Resources` zugehörigen Daten zu den Objekten der Klasse **Businessdaten** an. Dies wird im XML-Objekt **ServiceResourceMapping** angegeben. Dies wird unter [Konfiguration der Zuordnung von API-Gateway-Ressourcen zu Businessdaten](#) im Detail beschrieben.
- 7) Die Attribute **Assembly** und **Assembly-Klasse**, die für den Lösungsmanager **ServiceRegistryManager** verfügbar sind, müssen spezifiziert werden, um Interoperabilität zu gewährleisten. Das XML-Objekt **ServiceRegistryManager** ist in Alfabet Expand verfügbar. Folgende Werte müssen spezifiziert werden:
 - **Assembly:** `ITPlan`
 - **Assembly-Klasse:** `ITPlanSolution.GenericServiceRegistryManager`
- 8) Erzeugen Sie in der Ansicht **API-Gateway-Datenbankverbindung** der Funktionalität **Konfiguration von Integrationslösungen** in der Alfabet -Benutzeroberfläche API-Gateway-Verbindungsobjekte. Für jede im XML-Objekt **APIGatewayConfig** konfigurierte API-Gateway-Verbindung sollte ein API-Gateway-Verbindungsobjekt erzeugt werden. Die API-Gateway-Verbindungsobjekte ermöglichen es einem Anwender, die Verbindung zu der relevanten API-Gateway-Instanz anzugeben, die beim Exportieren/Synchronisieren eines technischen Service auf der Ansichtseite **Technische Services** (`COM_TechServices`) verwendet werden soll. Sie müssen außerdem angeben, ob Alfabet die technischen Services automatisch synchronisieren soll, wenn die Ansichtseite **Technische Services** geladen wird. Das Erzeugen von API-Gateway-Verbindungsobjekten wird im Abschnitt *Konfigurieren von semantischen Verbindungen für Integrationslösungen* im Referenzhandbuch *Konfigurieren von Bewertungen und Referenzdaten in Alfabet* beschrieben.
- 9) Die relevanten Anwenderprofile müssen so konfiguriert werden, dass sie die Ansichtseite **Technische Service-Registry-Services – gefiltert** (`COMSR_ServicesExt`), die im Root-Knoten des Explorers **Komponenten** (`COM_Explorer`) verfügbar ist, und die Ansichtseite **Technische Services** (`COM_TechServices`), die in der Objektansicht **Komponenten** verfügbar ist, enthalten. (Diese Ansicht ist nur für Komponenten des Typs `Service` verfügbar.) Die Ansichtseite **Technische Service-Registry-Services** (`COMSR_Services`), die im Root-Knoten des Explorers **Komponenten** (`COM_Explorer`) verfügbar ist, sollte für die Anwenderprofile, die auf webMethods API Gateway zugreifen, nicht sichtbar sein. Weitere Informationen über das Konfigurieren von Anwenderprofilen finden Sie im Kapitel *Konfigurieren von Anwenderprofilen für die Anwender-Community* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- 10) Wenn APIs mit einem ADIF-Importschema in Alfabet importiert werden sollen, müssen Sie mit dem **API-Gateway-Assistenten** ein ADIF-Schema erstellen, um die APIs in temporäre Datenbanktabellen zu importieren. Ihr Unternehmen muss die Aktualisierung der Alfabet-Datenbank -Tabellen mit den Daten aus den temporären Datenbanktabellen konfigurieren. Dies wird im Referenzhandbuch *Alfabet-Datenintegrationsframework* ausführlicher beschrieben.

Konfigurieren des Alfabet -Klassenmodells für Interoperabilität mit webMethods API Gateway

Folgende Konfiguration ist für die Interoperabilität mit webMethods® API Gateway erforderlich:

- Konfigurieren von Objektklassenstereotypen für die Objektklasse `Service`. Jeder API-Asset-Typ, der in Alfabet angezeigt werden soll, muss einem Objektklassenstereotyp der Klasse `Service` im XML-Element **AssetTypeMappings** des XML-Objekts **APIGatewayConfig** zugeordnet werden. Daher müssen Sie die relevanten Objektklassenstereotypen für die Klasse `Service` konfigurieren, bevor Sie das XML-Objekt **APIGatewayConfig** spezifizieren können. Ausführlichere Informationen zum Konfigurieren von Objektklassenstereotypen finden Sie im Abschnitt *Konfigurieren von Objektklassenstereotypen für Objektklassen* im Kapitel *Konfigurieren des Klassenmodells* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Die Objektklassenstereotypen für die Objektklasse `Service` könnten beispielsweise folgendermaßen angegeben werden:

```
<ClassStereotypes>
  <Stereotype Name="SOAPService" Caption="SOAP Service"
    CaptionPlural="SOAP Services" Comments="" HasMandates="false" />
  <Stereotype Name="DataService" Caption="DataService"
    CaptionPlural="Data Services" Comments="" HasMandates="false" />
  <Stereotype Name="RESTService" Caption="REST Service"
    CaptionPlural="REST Services" Comments="" HasMandates="false" />
  <Stereotype Name="XMLService" Caption="XML Service"
    CaptionPlural="XML Services" Comments="" HasMandates="false" />
</ClassStereotypes>
```

- Konfigurieren von Objektklassenstereotypen für die Objektklasse `ServiceOperation`. Jede einem importierten API zugehörige `Operation` wird in Alfabet als Operation des technischen Services erzeugt.
- Konfigurieren von Objektklassenstereotypen für die Objektklasse `ServiceOperationMethod`. Jede einem importierten API zugehörige `Method` wird in Alfabet als Operationsmethode des technischen Services erzeugt.
- Konfigurieren Sie die Zuordnung der Stereotypen der technischen Services (basierend auf der Klasse `Service`), der Stereotypen der Operationen des technischen Services (basierend auf der Klasse `ServiceOperation`) und die Stereotypen der Operationsmethoden des technischen Services (basierend auf der Klasse `ServiceOperationMethod`) im XML-Objekt **TechServiceManager**. Geben Sie für jeden technischen Service-Stereotyp an, welche Operationen des technischen Services zulässig sind. Geben Sie für jeden Operations-Stereotyp des technischen Services an, welche Methoden der Operation des technischen Services zulässig sind. Weitere Informationen zur Zuordnung von Objektklassenstereotypen, die für technische Services und Operationen des technischen Services definiert wurden, finden Sie im Abschnitt *Konfigurieren von Objektklassenstereotypen für technische Services* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- Konfigurieren Sie Release-Status für die Objektklasse `Service` sowie auch alle relevanten Objektklassenstereotypen für die Objektklasse `Service`. Jeder Asset-Typ in API Gateway hat einen Status. Daher sollte jedes Objektklassenstereotyp über entsprechende Release-Status verfügen, die den Status des Asset-Typs entsprechen, die ihm zugeordnet werden. Bitte beachten Sie Folgendes:

- Der Satz an Release-Status für jeden relevanten Objektklassenstereotyp muss im XML-Objekt **ReleaseStatusDefs** konfiguriert werden. Für die Objektklasse `Service` als Ganzes sowie für die einzelnen Objektklassenstereotypen muss eine Release-Status-Definition erzeugt werden. Die Release-Status-Definition für die Objektklasse muss den kompletten Satz an Release-Status enthalten, die für ihre Objektklassenstereotypen konfiguriert wurden. Die Release-Status-Definition für einen Objektklassenstereotyp darf nur die Release-Status enthalten, die für den Objektklassenstereotyp relevant sind, sowie eine Abfolge der Release-Status, die zum Erreichen eines bestimmten Ziel-Release-Status verfügbar sind. Weitere Informationen zum Konfigurieren von Release-Status finden Sie im Abschnitt *Konfigurieren von Release-Status-Definitionen für Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- Die Release-Status in Alfabet sollten in der primären Sprache spezifiziert werden, die der Definition der Basis-Landeseinstellungen der standardmäßigen Landeseinstellung entspricht. Wenn ein Anwender versucht, Daten in Alfabet mit API-Gateway-Assets zu synchronisieren, und für ein oder mehrere dieser API-Gateway-Assets Status in einer anderen Sprache als der primären Sprache in Alfabet definiert sind, wird eine Fehlermeldung angezeigt.
- Für jeden für einen Asset-Typ verfügbaren Status muss für das relevante Objektstereotyp ein zugehöriger Release-Status definiert sein. Der Name des Release-Status muss mit dem Namen des API-Gateway-Status übereinstimmen.
- Der für den Objektklassenstereotyp verfügbare Satz von Release-Status kann Release-Status beinhalten, die für die Verwaltung der technischen Services in alfabet relevant sind. Solche Release-Status werden als Alfabet -Eigentum betrachtet, da sie nur in Alfabet definiert werden.
- Die in API Gateway definierten Status, die auch Eigentum von API Gateway sind, sollten dieselben Namen aufweisen wie die in Alfabet definierten Release-Status.
- Einer der Release-Status, die im Release-Status-Satz konfiguriert sind, wird im XML-Attribut `HandoverStatus` des XML-Objekts **APIGatewayConfig** als der Release-Status spezifiziert, über den der technische Service verfügen muss, damit er im API-Gateway-Repository aktualisiert wird. Falls erforderlich kann für jeden Objektklassenstereotyp ein anderer Übergabestatus spezifiziert werden. Dies wird im Abschnitt [Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Gateway](#) beschrieben.

Konfigurieren von Verbindungen und API-Asset-Zuordnungen für webMethods API Gateway

Es können mehrere Verbindungen zu webMethods® API Gateway definiert werden, um den Anforderungen von Partner-Organisationen gerecht zu werden. Über das XML-Objekt **APIGatewayConfig** können Sie die Verbindungen zu webMethods API Gateway-Instanzen konfigurieren und festlegen, wie die APIs in Alfabet importiert werden sollen. Sie können die erforderlichen Informationen konfigurieren, damit Anwender APIs über die Alfabet -Benutzeroberfläche in Alfabet importieren und exportieren können, und/oder Sie können konfigurieren, wie die APIs über ein ADIF-Importschema importiert werden. Beachten Sie Folgendes:

- APIs können über die Ansicht **Technische Service-Registry-Services** (`COMSR_Services`) oder die Ansicht **Technische Service-Registry-Services - gefiltert** (`COMSR_ServicesExt`) importiert werden. Wenn der Import von APIs über die Alfabet -Benutzeroberfläche erfolgen soll, müssen Sie die API-Assets den relevanten Objektklassenstereotypen zuordnen, die für die Klasse `Service` im XML-Objekt **APIGatewayConfig** konfiguriert sind. Sie müssen auch den relevanten Release-

Status `HandoverStatus` angeben, den ein technischer Service haben muss, damit er aus Alfabet in das API-Gateway exportiert wird.

- APIs können auch mithilfe eines ADIF-Importschemas in Alfabet 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 ADIF-Schema 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. Weitere Informationen zum Konfigurieren eines ADIF-Schemas finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.



Wenn Sie in das XML-Objekt **APIGatewayConfig** eine Zeichenfolge eingeben möchten, die Sonderzeichen enthält, müssen Sie diese durch den entsprechenden kompatiblen HTML-Code ersetzen, zum Beispiel:

- `>`; für `>`
- `<`; für `<`
- `"`; für `"`
- `[`; für `[`
- `]`; für `]`

Die Konfiguration der Verbindungen wird im XML-Objekt **APIGatewayConfig** über das Root-XML-Element **APIGatewayConfig** und dem diesem untergeordneten XML-Element **APIGatewayConnections** vorgenommen. Sobald Sie die Konfiguration des XML-Objekts **APIGatewayConfig** abgeschlossen haben, müssen Sie in der Ansicht **API-Gateway-Datenbankverbindung**, die in der Funktionalität **Integrationslösungskonfigurationen** in der Alfabet -Benutzeroberfläche verfügbar ist, ebenfalls Verbindungsobjekte erzeugen.

Das Root-XML-Element **APIGatewayConfig** verfügt über zwei untergeordnete Elemente:

- **Proxy:** Über dieses XML-Element können Sie konfigurieren, wie Anforderungen über einen Proxyserver an Instanzen von webMethods API Gateway gesendet werden. Die Konfiguration des XML-Elements **Proxy** wird im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#) beschrieben.
- **APIGatewayConnections:** Über dieses XML-Element können Sie die Verbindung zu Instanzen von webMethods API Gateway konfigurieren. Für jede Verbindung zu einer webMethods API Gateway-Instanz sollte ein XML-Element **APIGatewayConnection** konfiguriert werden. Jedes XML-Element verfügt über zwei untergeordnete Elemente. Bei diesen untergeordneten Elementen handelt es sich um das Element **AssetTypeMappings**, mit dem Sie die Zuordnung von APIs aus dem API-Gateway zu Alfabet angeben können, wenn sie über die Alfabet -Benutzeroberfläche importiert werden, und um das Element **DataConnectivityInfo**, mit dem Sie den Import von APIs aus dem API-Gateway in Alfabet angeben können, wenn sie über ein ADIF-Importschema importiert werden.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist eine Vorlage über das Attribut **XML-Vorlage** des XML-Objekts verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf. Bei einer Beispielformatierung müssen Sie die vorhandenen XML-Elemente bearbeiten anstatt sie der XML-Konfiguration hinzuzufügen.

Sie können mehrere Verbindungen von Alfabet zu mehreren Instanzen von webMethods API Gateway konfigurieren. So definieren Sie eine Verbindung zu einer Instanz von webMethods API Gateway.

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf das XML-Objekt **APIGatewayConfig**, und wählen Sie **XML bearbeiten** aus. Das XML-Objekt wird geöffnet.



Über das XML-Element **Proxy** können Sie konfigurieren, wie Anforderungen über einen Proxyserver an Instanzen von webMethods API Gateway gesendet werden. Dies ist optional. Wenn Sie mehrere Verbindungen konfiguriert haben und jede Verbindung einen anderen Proxy verwenden soll, können Sie Ihrer Proxykonfiguration zusätzliche Proxys hinzufügen und die entsprechende Proxydefinition in der Verbindungskonfiguration für das API-Gateway angeben. Die Konfiguration des XML-Elements **Proxy** wird im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#) beschrieben.

- 3) Fügen Sie dem Root-XML-Element **APIGatewayConfig** ein untergeordnetes XML-Element **APIGatewayConnections** hinzu.
- 4) Fügen Sie dem XML-Element **APIGatewayConnections** für jede Datenbankverbindungsinstanz, die Sie definieren, ein untergeordnetes XML-Element **APIGatewayConnection** hinzu.
- 5) Konfigurieren Sie eine Verbindung zu jeder API-Gateway-Instanz, indem Sie die folgenden XML-Attribute für das XML-Element **APIGatewayConnection** festlegen:

- **Name:** Geben Sie einen eindeutigen Namen für die Verbindung zum API-Gateway ein.
- **ServerURL:** Geben Sie die URL für den Zugriff auf das API-Gateway oder den relevanten Servervariablenamen ein.



In einigen XML-Attributen können Servervariablen verwendet werden, um den Wert des Attributs zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation auszulesen, wenn eine Verbindung zum API-Gateway hergestellt wurde. Weitere Informationen über die Verwendung von Servervariablen finden Sie im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).

- **UserName:** Geben Sie den Anwendernamen Ihres Unternehmens für den Zugriff auf das API-Gateway oder den relevanten Servervariablenamen ein.
- **Password:** Geben Sie das Passwort Ihres Unternehmens für den Zugriff auf das API-Gateway oder den relevanten Servervariablenamen ein.
- **ServiceViewURL:** Geben Sie das Präfix für den vollständigen Link ein, der bei einem Klick eines Anwenders auf die Schaltfläche **Ansicht des Services in Service-Registry öffnen** auf der Ansichtssseite **Technische Service-Registry-Services – gefiltert** zum Navigieren von Alfabet zum API-Gateway verwendet wird. Geben Sie die URL für den Zugriff auf die Serviceansicht im API-Gateway oder den relevanten Servervariablenamen ein. Beispiel:
`http://<APIGatewayServer>:<Port>/#default/apiDetails)`
- **TransactionHistoryPeriod:** Geben Sie die Anzahl der Tage ein, für die die Transaktionen abgerufen werden. Dies ist nur bei Massenimporten über ADIF erforderlich.
- **Timeout:** Geben Sie die Anzahl von Minuten Anwenderinaktivität ein, nach der die Verbindung zum API-Gateway beendet werden soll.

- 6) Wenn APIs über die Alfabet -Benutzeroberfläche in Alfabet importiert werden sollen, fügen Sie jedem XML-Element **APIGatewayConnection** ein XML-Element `AssetTypeMappings` hinzu. Geben Sie für jeden Asset-Typ, den Sie in Alfabet importieren möchten, ein untergeordnetes XML-Element `AssetTypeMapping` an.
- 7) Konfigurieren Sie die Zuordnung jedes Asset-Typs, indem Sie die folgenden XML-Attribute für das XML-Element `AssetTypeMapping` festlegen:
 - `APIGatewayAssetType`: Geben Sie den Namen des webMethods API Gateway-Asset-Typs ein, der dem im XML-Element `AlfabetStereotype` spezifizierten Objektklassenstereotyp zugeordnet werden soll.
 - `HandoverStatus`: Spezifizieren Sie den Release-Status, über den der auf dem Objektklassenstereotyp im XML-Attribut `AlfabetStereotype` basierende technische Service in Alfabet verfügen muss, um in der Instanz von webMethods API Gateway aktualisiert zu werden. Weitere Informationen zum Konfigurieren von Release-Status für die Interoperabilität mit webMethods API Gateway finden Sie im Abschnitt *Konfigurieren von Release-Status-Definitionen für Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
 - `AlfabetStereotype`: Geben Sie den Namen des Objektklassenstereotyps der Klasse `Service` in Alfabet ein, das dem im XML-Element `APIGatewayAssetType` spezifizierten webMethods API Gateway-Asset-Typ zugeordnet werden soll.
- 8) Falls APIs über ein ADIF-Importschema in Alfabet importiert werden sollen, fügen Sie jedem XML-Element **APIGatewayConnection** ein XML-Element `DataImport` mit einem untergeordneten XML-Element `DataConnectivityInfo` hinzu.
- 9) Geben Sie für jede Importdefinition, die Sie konfigurieren möchten, ein untergeordnetes XML-Element `DataConnection` und ein XML-Attribut `Name` an. Geben Sie den Namen der Datenverbindung ein.
- 10) Geben Sie für jeden Importtyp in webMethods API Gateway, den Sie über ADIF importieren möchten, ein XML-Element `Import` mit einem XML-Attribut `Type` ein. Geben Sie den Importtyp ein, der über das ADIF-Importschema importiert werden kann. Zulässige Importtypen: `API`, `POLICY`, `APPLICATION`, `TRANSACTION`, `TRANSACTIONDETAIL`, `PACKAGE`, und `PLAN`:
- 11) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Konfiguration der Zuordnung von API-Gateway-Ressourcen zu Businessdaten

Ein API, das aus dem webMethods API Gateway in Alfabet importiert wird, wird auf der Basis des im XML-Objekt **APIGatewayConfig** konfigurierten Stereotyps als technischer Service erzeugt. Beim Importieren des APIs können auch die zugehörigen `Resources` importiert werden. In diesem Fall kann für jede `Resource` eine Instanz der Objektklasse `BusinessData` aktualisiert oder erzeugt werden. Dazu müssen Sie die zu verwendende Logik zuordnen, um die entsprechenden Businessdaten in Alfabet zu finden und diese Businessdaten mit Informationen über die `Resource` zu aktualisieren oder, wenn nötig, neue Businessdaten zu erzeugen und Informationen über die `Resource` hineinzuschreiben. Für die Suche nach Businessdaten bzw. deren Erzeugung können mehrere Zuordnungslogiken konfiguriert werden.



Wenn `Methods` und `Parameters` für eine `Resource` definiert sind, werden sie auch zu den Businessdaten in Alfabet hinzugefügt. Für jede `Method`, die für eine `Resource` angegeben ist, wird eine Methode der Operation des technischen Services (Instanz der Klasse `ServiceOperationMethod`) erzeugt, und für jeden `Parameter` wird ein Parameter für die Methode der Operation des technischen Services (Instanz der Klasse `ServiceOperationMethodParameter`) erzeugt. Beachten Sie, dass die Methode der Operation des technischen Services (Instanz der Klasse `ServiceOperationMethod`) auf Basis eines Objektklassenstereotypen erzeugt wird, der im XML-Objekt **TechServiceManager** angegeben ist. Informationen hierzu finden Sie im Abschnitt *Konfigurieren von Objektklassenstereotypen für technische Services* im Kapitel *Konfigurieren von in der Lösungsumgebung implementierten alfabet-Funktionalitäten*.

Das Folgende muss im XML-Objekt **ServiceResourceMapping** konfiguriert werden.

- Geben Sie eine oder mehrere `ResourceMapping`-Definitionen an, um vorhandene Businessdaten zu finden, oder um vorhandene Business-Objekte zu finden, für die neue Businessdaten erzeugt werden. Die Zuordnungslogik ist in hohem Maße konfigurierbar.



Sie könnten zum Beispiel folgende Zuordnungslogiken eingeben:

- **Priorität 1:** Finden Sie einen vorhandenen Businessdatensatz, der allen Eigenschaften entspricht, die in der Ressourcen-Zuordnung angegeben sind, und aktualisieren Sie diesen. Finden Sie zum Beispiel einen Businessdatensatz mit der gleichen Eigenschaft **Name** wie die `Resource`.
 - **Priorität 2:** Finden Sie einen vorhandenen Businessdatensatz, der einigen der Eigenschaften entspricht, die in der Ressourcen-Zuordnung angegeben sind, und aktualisieren Sie diesen. Finden Sie zum Beispiel einen Businessdatensatz mit teilweise der gleichen Eigenschaft **Name** wie die `Resource`.
 - **Priorität 3:** Erzeugen Sie einen neuen Businessdatensatz für ein Business-Objekt, das allen Eigenschaften in der Ressourcen-Zuordnung entspricht. Finden Sie zum Beispiel ein vorhandenes Business-Objekt mit der gleichen Eigenschaft **Name** wie die `Resource` und erzeugen Sie einen Businessdatensatz für dieses Business-Objekt.
 - **Priorität 4:** Erzeugen Sie einen neuen Businessdatensatz für ein Business-Objekt, das einigen Eigenschaften in der Ressourcen-Zuordnung entspricht. Finden Sie zum Beispiel ein vorhandenes Business-Objekt mit teilweise der gleichen Eigenschaft **Name** wie die `Resource` und erzeugen Sie einen Businessdatensatz für dieses Business-Objekt.
- Geben Sie an, wie die Datentypen (wie beispielsweise `string`, `integer` usw.) für `Resources` den Eigenschaftstypen für die Objektklasseneigenschaftstypen für die Klasse `BusinessData` zugeordnet werden. Im Alfabet -Klassenmodell hat jede Objektklasseneigenschaft ein Attribut **Eigenschaftstyp** mit einem Wert wie z. B. `String`, `StringArray`, `Integer`, `Real`, `Text` usw. Zum Beispiel haben die Eigenschaften **Name** und **Version** beide den **Eigenschaftstyp** = `String`. Sie müssen diese Zuordnung für die Businessdaten angeben, die im Kontext jedes Objektklassenstereotypen der bei der Integration des API-Portals verwendeten Klasse `Service` erzeugt/aktualisiert wurden.
 - Geben Sie an, in welcher Weise die Datentypen (wie beispielsweise `string`, `integer` usw.) für `Resources`, die den Attributtypen der Businessdaten zugeordnet sind, für Alfabet -Businessdaten relevant sind. Businessdaten-Attribute sind Attribute, die die Businessdaten beschreiben. Businessdaten-Attribute werden in der Eigenschaft `Attributes` der Klasse `BusinessData`

erfasst. Das Attribut `Type` der Klasse `BusinessDataAttribute` wird von Ihrem Lösungsentwickler für ihre Lösungskonfiguration über die geschützte Aufzählung **`BusinessDataAttributeType`** konfiguriert. Jedes Aufzählungselement, das für die geschützte Aufzählung **`BusinessDataAttributeType`** definiert ist, stellt einen Typen eines Businessdaten-Attributs dar.

So spezifizieren Sie das XML-Objekt **`ServiceResourceMapping`** für die Integration mit webMethods API Gateway:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf das XML-Objekt **`ServiceResourceMapping`**, und wählen Sie **XML bearbeiten** aus. Das XML-Objekt wird geöffnet.
- 3) Die Namen der `Resource` in webMethods API Gateway können ein oder mehrere Präfixe enthalten, die Sie vielleicht aus der Zuordnung ausschließen möchten. Fügen Sie dem Root-XML-Element **`ServiceResourceMapping`** ein untergeordnetes XML-Element **`PrefixesToIgnoreOnResourceMapping`** hinzu, um die zu ignorierenden Präfixe anzugeben.
- 4) Fügen Sie dem XML-Element **`PrefixesToIgnoreOnResourceMapping`** für jedes Präfix, das Sie angeben möchten, ein untergeordnetes XML-Element **`Value`** hinzu. Geben Sie das Präfix, das ignoriert werden soll, in das XML-Element **`Value`** ein.



Eine `Resource` mit dem Namen `Employee` kann beispielsweise einen Namen haben, der den Pfad im API-Gateway enthält, an dem sich die `Resource` befindet. Der Name kann beispielsweise wie ein Ressourcenpfad aussehen: `<StoreName>/<API-Name>/Employee`. In diesem Fall möchten Sie vielleicht den `ResourceEmployee` als Businessdaten zu A hinzufügen, dabei aber die Präfixe `Store` und `API` ignorieren. In einem solchen Fall würden Sie Folgendes angeben:

```
...
<PrefixesToIgnoreOnResourceMapping>
  <Value>store</Value>
  <Value>api</Value>
</PrefixesToIgnoreOnResourceMapping>
```

- 5) Für jede Regel, um die `Resource` Businessdaten zuzuordnen, erzeugen Sie ein XML-Element **`ResourceMapping`**. Konfigurieren Sie für das XML-Element **`ResourceMapping`** die folgenden XML-Attribute:
 - **Class:** Geben Sie `BusinessData` ein, um nach vorhandenen Businessdaten für die Zuordnung zu suchen. Geben Sie `BusinessObject` ein, um nach vorhandenen Business-Objekten zu suchen, für die neue Businessdaten erzeugt werden sollen. Im Kontext dieser Konfiguration können keine neuen Business-Objekte erzeugt werden.
 - **Priority:** Geben Sie eine Ganzzahl ein, um die Priorität dieser Ressourcenzuordnung festzulegen. Zuerst wird nach einer Ressourcenzuordnungsdefinition mit `Priority="1"` gesucht, als Zweites wird nach `Priority,="2"` gesucht usw.
 - **Properties:** Geben Sie eine oder mehrere Objektklasseneigenschaften der Businessdaten/des Business-Objekts an, die für eine Übereinstimmung gesucht werden müssen. Beim Namen der `Resource` wird im Normalfall die folgende Syntax verwendet: `{Name}\{Property1}\{Property2}`. Sie können mehrere Objektklassen angeben, um die Suche nach einer Übereinstimmung zu erleichtern. Zum Beispiel: `Properties="{Name}/{Version}"`.

- **MatchType:** Geben Sie `FullString` ein, wenn alle Werte im XML-Attribut `Properties` übereinstimmen müssen, um die Businessdaten oder das Business-Objekt zu finden. Geben Sie `SubString` ein, wenn nur einige der Werte im XML-Attribut `Properties` übereinstimmen müssen, um die Businessdaten oder das Business-Objekt zu finden. Wird keine Übereinstimmung gefunden, wird die `Resource` ignoriert, und die Businessdaten werden nicht aktualisiert/erzeugt.
- 6) Um die Informationen zu Dateneigenschaften für Businessdaten zuzuordnen, erstellen Sie ein XML-Element ***DataTypeMappings***.
 - 7) Erzeugen Sie für jeden Objektklassenstereotyp der Klasse `Service`, der im XML-Element `AlfabetStereotype` des XML-Objekts ***APIGatewayConfig*** angegeben ist, ein XML-Element ***DataTypeMapping***, das dem XML-Element ***DataTypesMappings*** untergeordnet ist.
 - 8) Erzeugen Sie im XML-Element ***DataTypeMapping*** ein XML-Attribut `AlfabetStereotype`, und geben Sie den Namen des für die Klasse `Service` konfigurierten Objektklassenstereotyps ein, das Sie angeben möchten.
 - 9) Erzeugen Sie im XML-Element ***DataTypeMapping*** für jeden Eigenschaftstyp, der für das im XML-Attribut angegebene `AlfabetStereotype` Objektklassenstereotyp relevant ist, ein XML-Element ***DataType***. Alfabet -Eigenschaftstypen sind `String`, `Boolean`, `Date`, `Integer`, `Text` usw. Der Eigenschaftstyp wird im Attribut ***Eigenschaftstyp*** für jede relevante Eigenschaft der Objektklasse `Service` angegeben. (Die Eigenschaften ***Name*** und ***Version*** für die Klasse `BusinessData` haben beispielsweise beide den ***Eigenschaftstyp*** = `String`.) Konfigurieren Sie für das XML-Element ***DataType*** die folgenden XML-Attribute:
 - ***AlfabetPropertyType***: Geben Sie den Typ für die Alfabet -Objektklasseneigenschaft (`String`, `Boolean`, `Date`, `Integer`, `Text` usw.) ein, der dem Datentyp des im XML-Attribut `ExternalPropertyType` angegebenen APIs zugeordnet werden soll.
 - ***ExternalPropertyType***: Geben Sie den Datentyp für das API ein, der dem im XML-Attribut `AlfabetPropertyType` definierten Alfabet -Objektklasseneigenschaftstyp zugeordnet werden soll.
 - 10) Erzeugen Sie im XML-Element ***DataTypeMapping*** für jeden Typ von Businessdaten-Attributen, der für Alfabet -Businessdaten relevant ist, ein XML-Element ***AttributeDataType***. Businessdaten-Attribute sind Attribute, die die Businessdaten beschreiben. Businessdaten-Attribute werden über die Eigenschaft `Attributes` der Klasse `BusinessData` erfasst. Das Attribut `Type` der Klasse `BusinessDataAttribute` wird von Ihrem Lösungsentwickler für Ihre Lösungskonfiguration über die geschützte Aufzählung ***BusinessDataAttributeType*** konfiguriert. Jedes Aufzählungselement, das für die geschützte Aufzählung ***BusinessDataAttributeType*** definiert ist, stellt einen Typen eines Businessdaten-Attributs dar. Konfigurieren Sie für das XML-Element ***AttributeDataType*** die folgenden XML-Attribute:
 - ***AlfabetAttributeType***: Geben Sie das für die geschützte Aufzählung ***BusinessDataAttributeType*** angegebene Aufzählungselement ein, das dem Datentyp `Resource` zugeordnet werden soll, der im Attribut `ExternalPropertyType` spezifiziert ist.
 - ***ExternalPropertyType***: Geben Sie den Datentyp `Resource` ein, der dem im XML-Attribut `AlfabetPropertyType` angegebenen Businessdaten-Attributtyp zugeordnet werden soll.
 - 11) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Kapitel 8: Konfigurieren der Interoperabilität mit den Apigee API-Plattform-Tools von Google

Software AG Bietet die Interoperabilität zwischen den Alfabet -Produkten und dem Apigee™-Produkt von Google, damit Ihr Unternehmen APIs in der Alfabet -Benutzeroberfläche als geplante technische Services überprüfen und testen kann. Das Apigee-Produkt dient als Plattform zum Entwerfen, Überwachen und Veröffentlichenden von APIs. Sie können die Interoperabilität zwischen Alfabet und Apigee konfigurieren und so integrierte Funktionen bereitstellen, um APIs als geplante technische Services in Alfabet zu importieren, diese geplanten technischen Services zu dokumentieren und zu beschreiben und die geänderte Definition des technischen Services wieder in Apigee zu exportieren.

APIs können über die Ansicht **Technische Service-Registry-Services** (COMSR_Services) oder die Ansicht **Technische Service-Registry-Services - gefiltert** (COMSR_ServicesExt) in die Benutzeroberfläche von Alfabet importiert werden. Sobald die technischen Services einen bestimmten Release-Status haben, können Sie in der Benutzeroberfläche über die Ansicht **Technische Services** (COM_TechServices) aus Alfabet in Apigee exportiert werden. Per Klick auf eine Schaltfläche können Anwender von Alfabet zu Apigee navigieren und die API-Definition direkt in der Benutzeroberfläche von Apigee anzeigen.



APIs können auch mithilfe eines ADIF-Importschemas in Alfabet importiert werden. Technische Services können nicht über ein ADIF-Schema in Apigee exportiert werden. Wenn APIs über ADIF importiert werden sollen, muss das ADIF-Schema über einen Assistenten erzeugt werden. Der Assistent definiert den korrekten Import von Daten aus Apigee in temporäre Tabellen in ADIF. Eine weitere Konfiguration ist erforderlich, um die Alfabet-Datenbank -Tabellen mit den Daten der temporären Datenbanktabellen zu aktualisieren. Der Import kann Zuordnungen enthalten, die in der Standardzuordnung über die Funktionalitäten auf der Benutzeroberfläche nicht enthalten sind. Weitere Informationen hierzu finden Sie unter [Erzeugen eines ADIF-Importschemas für den Import aus Apigee](#).

Apigee API-Vertreter und ihre Referenzen werden vordefinierten Objektklassen in Alfabet zugeordnet. Wenn ein API über die Ansichten in der Alfabet -Benutzeroberfläche in Alfabet importiert wird, wird in Alfabet ein technischer Service erzeugt. Wenn für das API Endpunkte oder Methoden angegeben sind, werden Objekte der in der nachfolgenden Tabelle angegebenen Objektklassen in Alfabet erzeugt:

Apigee-Objekt	Alfabet-Objektklasse
API-Vertreter	Technischer Service (Service) Bitte beachten Sie, dass die zusätzliche Spezifikation der Zuordnung von APIs zu technischen Services im XML-Objekt APIRepositoryConfig konfiguriert werden muss.
Endpunkt	Operation des technischen Service (ServiceOperation)
Zielendpunkt	Businessdaten (BusinessData) Bitte beachten Sie, dass die zusätzliche Spezifikation der Zuordnung von Zielendpunkten zu Businessdaten im XML-Objekt ServiceResourceMapping konfiguriert werden muss.

Apigee-Objekt	Alfabet-Objektklasse
Methode	Operationsmethode des technischen Service (<code>ServiceOperationMethod</code>)
Methodenparameter	Parameter der Operationsmethode des technischen Service (<code>ServiceOperationMethodParameter</code>)

Um die Interoperabilität mit Apigee zu konfigurieren, stehen Ihnen die folgenden Informationen zur Verfügung:

- [Übersicht über die für die Interoperabilität mit Apigee erforderliche Konfiguration](#)
- [Konfigurieren des Klassenmodells für Interoperabilität mit Apigee](#)
- [Konfigurieren von Verbindungen und API-Proxy-Zuordnungen für die Apigee-Integration](#)
- [Senden von Anfragen über einen Proxyserver an Apigee](#)
- [Erzeugen von Apigee-Datenverbindungen](#)
- [Erzeugen eines ADIF-Importschemas für den Import aus Apigee](#)

Übersicht über die für die Interoperabilität mit Apigee erforderliche Konfiguration

Nachfolgend finden Sie eine Übersicht über alle Konfigurationsschritte, die für die Interoperabilität mit Apigee erforderlich sind:

- 1) Konfigurieren Sie Objektklassenstereotypen für die Objektklassen `Service`, `ServiceOperation` und `ServiceOperationMethod`. Später ordnen Sie die Objektklassenstereotypen der Klasse `Service` den in Apigee verfügbaren API-Vertretern zu. Dies wird unter [Konfigurieren des Klassenmodells für Interoperabilität mit Apigee](#) im Detail beschrieben.
- 2) Ordnen Sie Objektklassenstereotypen für die Objektklassen `Service`, `ServiceOperation` und `ServiceOperationMethod` im XML-Objekt **`TechServiceManager`** zu. Geben Sie für jeden technischen Service-Stereotyp an, welche Operationen des technischen Services zulässig sind. Geben Sie für jeden Operations-Stereotyp des technischen Services an, welche Methoden der Operation des technischen Services zulässig sind. Informationen dazu finden Sie unter *Konfigurieren von Objektklassenstereotypen für technische Services* im Kapitel *Konfigurieren von in der Lösungsumgebung implementierten alfabet-Funktionalitäten* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- 3) Konfigurieren Sie Release-Status für die Objektklasse `Service` sowie die Objektklassenstereotypen im XML-Objekt **`ReleaseStatusDefs`**. Bitte beachten Sie, dass der Release-Status `HandoverStatus` angegeben sein muss, damit die technischen Services in Alfabet in Apigee exportiert werden können. Dies wird unter [Konfigurieren des Klassenmodells für Interoperabilität mit Apigee](#) im Detail beschrieben.
- 4) Geben Sie im XML-Objekt **`APIRepositoryConfig`** eine oder mehrere Verbindungen zu Apigee an. Hier legen Sie auch fest, wie API-Vertreter den Objektklassenstereotypen zugeordnet werden, die

für die Objektklasse `Service` konfiguriert sind, und welcher `HandoverStatus` für die technischen Services erforderlich ist, um sie in Apigee zu exportieren. Dies wird unter [Konfigurieren von Verbindungen und API-Proxy-Zuordnungen für die Apigee-Integration](#) im Detail beschrieben.

- 5) Wenn zum Konfigurieren der Verbindung im XML-Objekt ***APIRepositoryConfig*** Servervariablen verwendet werden, müssen die Servervariablen für den Server-Alias im Alfabet Administrator spezifiziert werden. Weitere Informationen zum Konfigurieren von Servervariablen finden Sie im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).
- 6) Geben Sie die Zuordnung der den Zielendpunkten zugehörigen Daten zu den Objekten der Klasse ***Businessdaten*** an. Dies wird im XML-Objekt ***ServiceResourceMapping*** angegeben. Dies wird unter [Konfigurieren von Verbindungen und API-Proxy-Zuordnungen für die Apigee-Integration](#) im Detail beschrieben.
- 7) Die Attribute ***Assembly*** und ***Assembly-Klasse***, die für den Lösungsmanager ***ServiceRegistryManager*** verfügbar sind, müssen spezifiziert werden, um Interoperabilität zu gewährleisten. Das XML-Objekt ***ServiceRegistryManager*** ist in Alfabet Expand verfügbar. Folgende Werte müssen spezifiziert werden:
 - ***Assembly***: `ITPlan`
 - ***Assembly-Klasse***: `ITPlanSolution.GenericServiceRegistryManager`
- 8) Erzeugen Sie in der Ansicht ***Apigee-Datenverbindung*** der Funktionalität ***Konfiguration von Integrationslösungen*** in der Alfabet -Benutzeroberfläche API-Datenverbindungsobjekte. Für jede im XML-Objekt ***APIRepositoryConfig*** konfigurierte Apigee-Verbindung sollte eine Apigee-Datenverbindung erzeugt werden. Die Apigee-Datenverbindungsobjekte ermöglichen es einem Anwender, die Verbindung zu der relevanten Apigee-Instanz anzugeben, die beim Exportieren/Synchronisieren eines technischen Services auf der Ansichtseite ***Technische Services*** (`COM_TechServices`) verwendet werden soll. Das Erzeugen von Apigee-Datenverbindungsobjekten wird im Abschnitt [Erzeugen von Apigee-Datenverbindungen](#) beschrieben.
- 9) Die relevanten Anwenderprofile müssen so konfiguriert werden, dass sie die Ansichtseite ***Technische Service-Registry-Services - gefiltert*** (`COMSR_ServicesExt`), die im Root-Knoten des Explorers ***Komponenten*** (`COM_Explorer`) verfügbar ist, und die Ansichtseite ***Technische Services*** (`COM_TechServices`), die in der Objektansicht ***Komponenten*** verfügbar ist, enthalten. (Diese Ansicht ist nur für Komponenten des Typs `Service` verfügbar.) Die Ansichtseite ***Technische Service-Registry-Services*** (`COMSR_Services`), die im Root-Knoten des Explorers ***Komponenten*** (`COM_Explorer`) verfügbar ist, sollte für die Anwenderprofile, die auf `webMethods API Portal` zugreifen, nicht sichtbar sein. Weitere Informationen über das Konfigurieren von Anwenderprofilen finden Sie im Kapitel *Konfigurieren von Anwenderprofilen für die Anwender-Community* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- 10) Wenn APIs mit einem ADIF-Import in Alfabet importiert werden sollen, müssen Sie mit dem `ApigeeImport_Assistant` ein ADIF-Importschema erzeugen, um die APIs in temporäre Datenbanktabellen zu importieren. Ihr Unternehmen muss die Aktualisierung der Alfabet-Datenbank -Tabellen mit den Daten aus den temporären Datenbanktabellen konfigurieren. Der ADIF-Import muss dann über eine der verfügbaren Importmethoden ausgeführt werden. Weitere Informationen zum Erzeugen des ADIF-Importschemas über den Assistenten finden Sie unter [Erzeugen eines ADIF-Importschemas für den Import aus Apigee](#). Die Konfiguration von ADIF-Importschemata und die Ausführung von ADIF-Importen werden ausführlich im Referenzhandbuch *Alfabet-Datenintegrationsframework* beschrieben.

Konfigurieren des Klassenmodells für Interoperabilität mit Apigee

Folgende Konfiguration ist für die Interoperabilität mit Apigee erforderlich:

- Konfigurieren von Objektklassenstereotypen für die Objektklasse `Service`. Jeder API-Proxy-Typ, der in Alfabet angezeigt werden soll, muss einem Objektklassenstereotyp der Klasse `Service` im XML-Element **`AssetTypeMappings`** des XML-Objekts **`APIRepositoryConfig`** zugeordnet werden. Daher müssen Sie die relevanten Objektklassenstereotypen für die Klasse `Service` konfigurieren, bevor Sie das XML-Objekt **`APIRepositoryConfig`** spezifizieren können. Ausführlichere Informationen zum Konfigurieren von Objektklassenstereotypen finden Sie im Abschnitt *Konfigurieren von Objektklassenstereotypen für Objektklassen* im Kapitel *Konfigurieren des Klassenmodells* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Die Objektklassenstereotypen für die Objektklasse `Service` könnten beispielsweise folgendermaßen angegeben werden:

```
<ClassStereotypes>
  <Stereotype Name="SOAPService" Caption="SOAP Service"
    CaptionPlural="SOAP Services" Comments="" HasMandates="false" />
  <Stereotype Name="DataService" Caption="DataService"
    CaptionPlural="Data Services" Comments="" HasMandates="false" />
  <Stereotype Name="RESTService" Caption="REST Service"
    CaptionPlural="REST Services" Comments="" HasMandates="false" />
  <Stereotype Name="XMLService" Caption="XML Service"
    CaptionPlural="XML Services" Comments="" HasMandates="false" />
</ClassStereotypes>
```

- Konfigurieren von Objektklassenstereotypen für die Objektklasse `ServiceOperation`. Jeder einem importierten API zugehörige Endpunkt wird in Alfabet als Operation des technischen Services erzeugt.
- Konfigurieren von Objektklassenstereotypen für die Objektklasse `ServiceOperationMethod`. Jede einem importierten API zugehörige Methode wird in Alfabet als Operationsmethode des technischen Services erzeugt.
- Konfigurieren Sie die Zuordnung der Stereotypen der technischen Services (basierend auf der Klasse `Service`), der Stereotypen der Operationen des technischen Services (basierend auf der Klasse `ServiceOperation`) und die Stereotypen der Operationsmethoden des technischen Services (basierend auf der Klasse `ServiceOperationMethod`) im XML-Objekt **`TechServiceManager`**. Geben Sie für jeden technischen Service-Stereotyp an, welche Operationen des technischen Services zulässig sind. Geben Sie für jeden Operations-Stereotyp des technischen Services an, welche Methoden der Operation des technischen Services zulässig sind. Weitere Informationen zur Zuordnung von Objektklassenstereotypen, die für technische Services und Operationen des technischen Services definiert wurden, finden Sie im Abschnitt *Konfigurieren von Objektklassenstereotypen für technische Services* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- Konfigurieren Sie Release-Status für die Objektklasse `Service` sowie auch alle relevanten Objektklassenstereotypen für die Objektklasse `Service`. Bitte beachten Sie, dass der Release-Status `HandoverStatus` angegeben sein muss, damit die technischen Services in Alfabet in Apigee exportiert werden können. Bitte beachten Sie Folgendes:
 - Der Satz an Release-Status für jeden relevanten Objektklassenstereotyp muss im XML-Objekt **`ReleaseStatusDefs`** konfiguriert werden. Für die Objektklasse `Service` als Ganzes sowie für die einzelnen Objektklassenstereotypen muss eine Release-Status-Definition erzeugt werden. Die Release-Status-Definition für die Objektklasse muss den kompletten Satz an Release-

Status enthalten, die für ihre Objektklassenstereotypen konfiguriert wurden. Die Release-Status-Definition für einen Objektklassenstereotyp darf nur die Release-Status enthalten, die für den Objektklassenstereotyp relevant sind, sowie eine Abfolge der Release-Status, die zum Erreichen eines bestimmten Ziel-Release-Status verfügbar sind. Weitere Informationen zum Konfigurieren von Release-Status finden Sie im Abschnitt *Konfigurieren von Release-Status-Definitionen für Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Konfigurieren von Verbindungen und API-Proxy-Zuordnungen für die Apigee-Integration

Es können mehrere Verbindungen zu Apigee definiert werden, um den Anforderungen von Partner-Organisationen gerecht zu werden. Über das XML-Objekt **APIRepositoryConfig** können Sie die Verbindungen zu Apigee-Instanzen konfigurieren und festlegen, wie die APIs in Alfabet importiert werden sollen. Sie können die erforderlichen Informationen konfigurieren, damit Anwender APIs über die Alfabet -Benutzeroberfläche in Alfabet importieren und exportieren können, und/oder Sie können konfigurieren, wie die APIs über ein ADIF-Importschema importiert werden. Beachten Sie Folgendes:

- Apigee-API-Vertreter können über die Ansicht **Technische Service-Registry-Services** (COMSR_Services) oder die Ansicht **Technische Service-Registry-Services – gefiltert** (COMSR_ServicesExt) importiert werden. Wenn der Import von APIs über die Alfabet -Benutzeroberfläche erfolgen soll, müssen Sie die API-Proxy-Typen den relevanten Objektklassenstereotypen zuordnen, die für die Klasse `Service` im XML-Objekt **APIRepositoryConfig** konfiguriert sind. Sie müssen auch den relevanten Release-Status `HandoverStatus` angeben, den ein technischer Service haben muss, damit er aus Alfabet in Apigee exportiert wird.



Wenn Sie in das XML-Objekt **APIRepositoryConfig** eine Zeichenfolge eingeben möchten, die Sonderzeichen enthält, müssen Sie diese durch den entsprechenden kompatiblen HTML-Code ersetzen, zum Beispiel:

- `>` für `>`
- `<` für `<`
- `"` für `"`
- `[` für `[`
- `]` für `]`

Die Konfiguration der Verbindungen wird im XML-Objekt **APIRepositoryConfig** über das Root-XML-Element **APIRepositoryConfig** und dem diesem untergeordneten XML-Element **RepositoryConnections** vorgenommen. Sobald Sie die Konfiguration des XML-Objekts **APIRepositoryConfig** abgeschlossen haben, müssen Sie in der Ansicht **Apigee-Datenverbindung**, die in der Funktionalität **Integrationslösungskonfigurationen** in der Alfabet -Benutzeroberfläche verfügbar ist, ebenfalls Verbindungsobjekte erzeugen.

Das Root-XML-Element **APIRepositoryConfig** verfügt über zwei untergeordnete Elemente:

- **Proxy**: Über dieses XML-Element können Sie konfigurieren, wie Anforderungen über einen Proxyserver an eine Apigee-Instanz gesendet werden. Die Konfiguration des XML-Elements **Proxy** wird im Abschnitt beschrieben.

- **RepositoryConnections:** Über dieses XML-Element können Sie mehrfache Verbindungen zu Apigee-Repositorys konfigurieren. Für jede Verbindung zu einem Apigee-Repository sollte ein untergeordnetes XML-Element des Typs **RepositoryConnection** konfiguriert werden. Jedes XML-Element **RepositoryConnection** verfügt über Attribute für die Definition der Verbindungsdaten, um die Verbindung zum Apigee-Repository und einem untergeordneten Element **AssetTypeMappings** herzustellen. Damit können Sie die Zuordnung von API-Vertreter-Typen von Apigee zu Alfabet festlegen, wenn diese über die Alfabet -Benutzeroberfläche importiert werden.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist eine Vorlage über das Attribut **XML-Vorlage** des XML-Objekts verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf. Bei einer Beispielformatierung müssen Sie die vorhandenen XML-Elemente bearbeiten anstatt sie der XML-Konfiguration hinzuzufügen.

Sie können mehrere Verbindungen von Alfabet zu mehreren Apigee-Repositorys konfigurieren. So definieren Sie eine Verbindung zu einem Apigee-Repository:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf das XML-Objekt **APIRepositoryConfig**, und wählen Sie **XML bearbeiten** aus. Das XML-Objekt wird geöffnet.



Über das XML-Element **Proxy** können Sie konfigurieren, wie Anforderungen über einen Proxyserver an Instanzen von Apigee gesendet werden. Dies ist optional. Wenn Sie mehrere Verbindungen konfiguriert haben und jede Verbindung einen anderen Proxy verwenden soll, können Sie Ihrer Proxykonfiguration zusätzliche Proxys hinzufügen und die entsprechende Proxydefinition in der Verbindungskonfiguration für Apigee angeben. Die Konfiguration des XML-Elements **Proxy** wird im Abschnitt [Senden von Anfragen über einen Proxyserver an Apigee](#) beschrieben.

- 3) Fügen Sie dem Root-XML-Element **APIRepositoryConfig** ein untergeordnetes XML-Element **RepositoryConnections** hinzu.
- 4) Fügen Sie dem XML-Element **RepositoryConnections** für jede Repository-Verbindung, die Sie definieren, ein untergeordnetes XML-Element **RepositoryConnection** hinzu.
- 5) Konfigurieren Sie eine Verbindung zu jedem Apigee-Repository, indem Sie die folgenden XML-Attribute für das XML-Element **RepositoryConnection** festlegen:
 - **Name:** Geben Sie einen eindeutigen Namen für die Verbindung zum Apigee-Repository ein.
 - **Repository:** Geben Sie den Namen des Apigee-Repositorys in der Apigee-Instanz ein, auf die Sie über diese Verbindung zugreifen möchten.
 - **ServerURL:** Geben Sie die URL des API-Servers für die Apigee Edge RESTful-Verwaltung ohne Angabe des Endpunkts ein, oder geben Sie den entsprechenden Servervariablenamen ein.



In einigen XML-Attributen können Servervariablen verwendet werden, um den Wert des Attributs zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation auszulesen, wenn eine Verbindung zum API-Portal hergestellt wurde. Weitere Informationen über die Verwendung von Servervariablen finden Sie im Abschnitt [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).

- **UserName:** Geben Sie die E-Mail-Adresse Ihres Unternehmens für den Zugriff auf das Apigee Edge RESTful-Verwaltungs-API oder den relevanten Servervariablenamen ein.

- **Password:** Geben Sie das Passwort Ihres Unternehmens für den Zugriff auf das Apigee Edge RESTful-Verwaltungs-API oder den relevanten Servervariablenamen ein.
 - **ServiceViewURL:** Geben Sie das Präfix für den vollständigen Link ein, der bei einem Klick eines Anwenders auf die Schaltfläche **Ansicht des Services in Service-Registry öffnen** auf der Ansichtseite **Technische Service-Registry-Services – gefiltert** zum Navigieren von Alfabet zur Apigee-Benutzeroberfläche verwendet wird. Geben Sie die URL für den Zugriff auf die Serviceansicht in Apigee oder den relevanten Servervariablenamen ein. Beispiel:
`https://apigee.com/platform`)
 - **MaxThreadCount:** Geben Sie die maximale Anzahl paralleler Threads für Aufrufe von Apigee ein. Diese Angabe ist optional. Die Standardeinstellungen sorgen in der Regel für ein optimales Importverhalten. Die Einstellung sollte nur geändert werden, wenn während der Integration Probleme auftreten.
 - **PageSize:** Die maximale Anzahl von API-Vertreter-Datenimporten über einen Thread. Diese Angabe ist optional. Die Standardeinstellungen sorgen in der Regel für ein optimales Importverhalten. Die Einstellung sollte nur geändert werden, wenn während der Integration Probleme auftreten.
 - **Timeout:** Geben Sie die Anzahl von Minuten Anwenderinaktivität ein, nach der die Verbindung zum API-Portal beendet werden soll.
- 6) Wenn API-Vertreter über die Alfabet -Benutzeroberfläche in Alfabet importiert werden sollen, fügen Sie jedem XML-Element **RepositoryConnection** ein XML-Element `AssetTypeMappings` hinzu. Geben Sie für jeden Asset-Typ, den Sie in Alfabet importieren möchten, ein untergeordnetes XML-Element `AssetTypeMapping` an.
- 7) Konfigurieren Sie die Zuordnung jedes Asset-Typs, indem Sie die folgenden XML-Attribute für das XML-Element `AssetTypeMapping` festlegen:
- **RepositoryAssetType:** Geben Sie den Namen des Apigee API-Proxy-Typs ein, der dem im XML-Element `AlfabetStereotype` spezifizierten Objektklassenstereotyp zugeordnet werden soll.
 - **HandoverStatus:** Spezifizieren Sie den Release-Status, über den der auf dem Objektklassenstereotyp im XML-Attribut `AlfabetStereotype` basierende technische Service in Alfabet verfügen muss, um im Apigee-Repository aktualisiert zu werden. Weitere Informationen zum Konfigurieren von Release-Status für die Apigee-Interoperabilität finden Sie im Abschnitt *Konfigurieren von Release-Status-Definitionen für Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
 - **AlfabetStereotype:** Geben Sie den Namen des Objektklassenstereotyps der Klasse `Service` in Alfabet ein, der dem im XML-Element `RepositoryAssetType` spezifizierten `webMethods` API Portal-Asset-Typ zugeordnet werden soll.
- 8) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Senden von Anfragen über einen Proxyserver an Apigee

Optional können Sie die Apigee-Integrationsschnittstelle dazu konfigurieren, Anforderungen an Apigee über einen Proxyserver zu senden. Dies erfordert für die oben beschriebene Repository-Verbindungskonfiguration die folgende zusätzliche Konfiguration im XML-Objekt **APIRepositoryConfig**:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **APIRepositoryConfig**, und wählen Sie **XML bearbeiten....** Das XML-Objekt mit der Datenimport-Definition für ServiceNow wird geöffnet.
- 3) Fügen Sie dem XML-Element `APIRepositoryConfig` ein untergeordnetes XML-Element `Proxy` hinzu.
- 4) Definieren Sie im XML-Element `Proxy` die folgenden XML-Attribute:
 - `url`: Definieren Sie die URL des Proxyservers.
 - `user`: Geben Sie bei Bedarf den Anwendernamen für den Zugriff auf den Proxyserver ein. Der Domänenname für die Authentifizierung wird separat über das XML-Attribut „Domäne“ definiert und muss nicht als Teil des Anwendernamens angegeben werden.
 - `password`: Geben Sie bei Bedarf das Passwort für den Zugriff auf den Proxyserver ein.
 - `domain`: Definieren Sie bei Bedarf den Domännennamen, der als Teil des Anwendernamens zur Authentifizierung am Proxyserver verwendet werden soll.
- 5) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Wenn Sie mehrere Repository-Verbindungen konfiguriert haben und jede Repository-Verbindung einen anderen Proxy verwenden soll, können Sie Ihrer Proxykonfiguration zusätzliche Proxys hinzufügen und in der Repository-Verbindungskonfiguration auf einen der Proxys verweisen. Die obige Proxydefinition wird als Standard verwendet, wenn einer Datenverbindung kein Proxy zugeordnet ist. Zur Verwendung eines zusätzlichen Proxys ist die folgende Konfiguration erforderlich:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **APIRepositoryConfig**, und wählen Sie **XML bearbeiten....** Das XML-Objekt mit der Apigee-Repository-Verbindungsdefinition wird geöffnet.
- 3) Fügen Sie dem XML-Element `Proxy` ein untergeordnetes XML-Element `AdditionalProxies` hinzu.
- 4) Fügen Sie dem XML-Element `AdditionalProxies` für jeden zusätzlichen Proxy, den Sie definieren möchten, ein untergeordnetes XML-Element `AdditionalProxy` hinzu, und definieren Sie die folgenden XML-Attribute für das XML-Element `AdditionalProxy`:
 - `Name`: Definieren Sie einen eindeutigen Namen für den zusätzlichen Proxy. Der Name wird verwendet, um in der Datenverbindungskonfiguration auf den Proxy zu verweisen.
 - `url`: Definieren Sie die URL des Proxyservers.
 - `user`: Geben Sie bei Bedarf den Anwendernamen für den Zugriff auf den Proxyserver ein. Der Domänenname für die Authentifizierung wird separat über das XML-Attribut „Domäne“ definiert und muss nicht als Teil des Anwendernamens angegeben werden.

- `password`: Geben Sie bei Bedarf das Passwort für den Zugriff auf den Proxyserver ein.
 - `domain`: Definieren Sie bei Bedarf den Domännennamen, der als Teil des Anwendernamens zur Authentifizierung am Proxyserver verwendet werden soll.
- 5) Fügen Sie jedem XML-Element **RepositoryConnection**, das einen der zusätzlichen Proxys verwenden soll, ein XML-Attribut `Proxy` hinzu. Der Wert des XML-Attributs `Proxy` muss mit dem Wert des XML-Attributs `Name` des XML-Elements **AdditionalProxy** übereinstimmen.
 - 6) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Erzeugen von Apigee-Datenverbindungen

In der Ansicht **Apigee-Datenverbindung** können Sie semantische Definitionen für alle relevanten Apigee-Verbindungen erzeugen, die im XML-Objekt **APIRepositoryConfig** konfiguriert wurden. Die Apigee-Datenverbindungsdefinitionen sind erforderlich, um zu definieren, mit welchen Umgebungen in einer Organisation, die in Apigee verwaltet wird, die Verbindung hergestellt wird. Benutzer können festlegen, welche konfigurierte Verbindung zur Synchronisierung technischer Alfabet -Services mit Apigee-Assets verwendet werden soll. Für jede im XML-Objekt **APIRepositoryConfig** konfigurierte Verbindung sollte mindestens eine Apigee-Datenverbindung erstellt werden.

Jedes Mal, wenn eine Ansichtsseite aus **Technische Services aus Repositories** (`COMSR_ServicesExt` oder `COMSR_Services`) geladen wird, wird jede im XML-Objekt **APIPortalConfig** konfigurierte Verbindung hergestellt, und die auf Apigee-Assets basierenden technischen Services werden mit den Daten im Apigee-Repository synchronisiert. Wenn der Anwender einen ausgewählten technischen Service auf der Ansichtsseite **Technische Services** (`COM_TechServices`) mit dem entsprechenden Apigee-Asset synchronisieren möchte, muss die für die Synchronisierung zu verwendende relevante Verbindungsdefinition im Feld **Master-Repository-Verbindung** auf der Registerkarte **Betriebs-Repositories** des Editors **Technischer Service** ausgewählt werden. Wenn Alfabet die technischen Services automatisch auf Basis von APIs aus dem Apigee-Repository synchronisieren soll, wenn die Ansichtsseite **Technische Services** (`COM_TechServices`) geladen wird, müssen Sie das Kontrollkästchen **Services automatisch aktualisieren** im Editor der **API-Portal-Datenbankverbindung** aktivieren. Wenn Sie das Kontrollkästchen nicht aktivieren, muss der Anwender einen ausgewählten technischen Service explizit aktualisieren.

Konfigurieren einer Apigee-Datenverbindung zur Anzeige im Feld **Verbindung** auf der Registerkarte **Service-Registry** des Editors **Technischer Service**:

- 1) Rufen Sie die Funktionalität **Konfiguration von Integrationslösungen** auf, und klicken Sie dann auf den Knoten **Apigee-Datenverbindung** im Explorer **Konfiguration von Integrationslösungen**.
- 2) Klicken Sie in der Ansicht auf **Neu > Apigee-Datenverbindung erzeugen**.



Wenn Sie bereits eine ähnliche Verbindung definiert haben und die Einstellungen dieser Verbindung für Ihre neue Verbindung übernehmen möchten, können Sie alternativ auf **Neu > Apigee-Datenverbindung als Kopie erzeugen** klicken und in der sich öffnenden Auswahl die vorhandene Verbindung auswählen, auf der die neue Verbindung basieren soll. Der Editor für die neue Verbindung wird dann mit allen Einstellungen identisch mit denen der kopierten Verbindung geöffnet, und der Name wird auf "Kopie von <Basisverbindungsname>" gesetzt.

- 3) Definieren Sie im Editor **Apigee-Datenverbindung** ggf. die folgenden Felder.

Registerkarte **Basisdaten**:

- **ID**: Alfabet weist jeder Apigee-Datenverbindung eine eindeutige Identifikationsnummer zu. Diese Nummer kann nicht bearbeitet werden.
- **Name**: Geben Sie einen eindeutigen Namen für die Apigee-Datenverbindung ein. Der Name sollte dem Anwender, der den technischen Service synchronisiert, helfen, das Apigee-Repository zu identifizieren, zu dem eine Verbindung hergestellt wird.
- **Release-Status**: Wählen Sie den aktuellen Release-Status der Apigee-Datenverbindung aus.
- **Beschreibung**: Geben Sie eine aussagekräftige Beschreibung ein, die den Zweck der Apigee-Datenverbindung verdeutlicht.
- **Hintergrundfarbe**: Geben Sie die Hintergrundfarbe an, die für technische Services verwendet werden soll, die mit dieser Apigee-Datenverbindung verknüpft sind. In den Ansichten **Technische Services aus Repositories** stehen die farbigen Zeilen für Assets, für die in Alfabet technische Services erstellt wurden. In der Ansicht **Technische Services** stehen die farbigen Zeilen für technische Services, die in Alfabet vorhanden sind und in das operative Repository exportiert wurden.
- **Vordergrundfarbe**: Geben Sie die Vordergrundfarbe an, die in Verbindung mit der Hintergrundfarbe verwendet werden soll.

Registerkarte **Autorisierter Zugriff**:

- **Autorisierter Anwender**: Klicken Sie auf das **Suchen**-Symbol, um der ausgewählten Apigee-Verbindung einen autorisierten Anwender zuzuordnen. Der autorisierte Anwender verfügt über Schreib-/Lesezugriffsrechte für das Objekt und ist für dessen Pflege verantwortlich.
- **Autorisierte Anwendergruppen**: Aktivieren Sie ein oder mehrere Kontrollkästchen, um allen Anwendern in den ausgewählten Anwendergruppen Schreib-/Lesezugriffsrechte zuzuordnen.

Registerkarte **Verbindung**:

- **Apigee-Verbindung**: Wählen Sie die im XML-Element **RepositoryConnection** des XML-Objekts **APIRepositoryConfig** konfigurierte relevante Verbindung aus, die hergestellt wird, wenn der Anwender im Feld **Master-Repository-Verbindung** auf der Registerkarte **Operative Repositories** des Editors **Technischer Service** diese Apigee-Datenverbindung auswählt.
- **Organisation**: Geben Sie in Apigee den Namen der Organisation ein, zu der die Verbindung hergestellt werden soll.
- **Installationsumgebungen**: Nachdem Sie im Feld **Organisation** die Organisation ausgewählt haben, klicken Sie auf die Schaltfläche **Umgebungen abrufen**, um alle implementierten Apigee-Umgebungen in der Apigee-Organisation anzuzeigen. Wählen Sie die relevante Umgebung aus, die die API-Proxys enthält, die über die Apigee-Datenverbindung in die technischen Services in Alfabet integriert werden sollen.
- **Revision berücksichtigen**: Aktivieren Sie das Kontrollkästchen, wenn Sie eine andere als die neueste Version von API-Proxys importieren möchten. Wenn das Kontrollkästchen aktiviert ist, kann ein Anwender zwischen allen Revisionen in den ausgewählten Umgebungen eine zur Integration auswählen. Durch den Export in Apigee wird ein API-Proxy mit Version 1 erzeugt. Wenn das Kontrollkästchen nicht aktiviert ist, wird die neueste Version eines API-Proxys in den ausgewählten Umgebungen für die Integration verwendet.

- **Vorlage für den Vertreternamen:** Technische Services, die in Alfabet auf der Grundlage von API-Proxys in Apigee erzeugt werden, werden entsprechend der in diesem Feld definierten Namenskonvention benannt. Die Definition kann die folgenden Variablen enthalten:

- {Name} wird durch den API-Proxy-Namen ersetzt.
- {Version} wird durch die API-Proxy-Version ersetzt.

Standardmäßig gilt ohne Berücksichtigung der Revisionen die folgende Namenskonvention: {Name}_V{Version}. Wenn Versionen berücksichtigt werden, wird der angegebene Name ergänzt mit "r.RevisionNumber".

- 4) Wiederholen Sie dies für alle im XML-Objekt **APIPortalConfig** konfigurierten Elemente **Connection**, die Anwender im Feld **Verbindung** auf der Registerkarte **Operative Repositories** des Editors **Technischer Service** auswählen können sollen.

Erzeugen eines ADIF-Importschemas für den Import aus Apigee

Daten von Apigee können über ADIF-Importprozesse importiert werden, um alle Daten im Batch zu importieren. Mit ADIF-Importprozessen können zusätzliche Daten aus Apigee, wie Richtlinien oder Statistiken, integriert werden, oder die Apigee-Daten können auf eine Art und Weise integriert werden, die von der Standardintegration über die Standardfunktionen zur Integration technischer Servicedaten abweicht.

Ein Teil des ADIF-Importschemas muss wie nachfolgend beschrieben über einen Assistenten erstellt werden. Vor dem Erstellen des ADIF-Importschemas müssen die folgenden Konfigurationsschritte ausgeführt werden:

- [Konfigurieren von Verbindungen und API-Proxy-Zuordnungen für die Apigee-Integration](#)
- [Erzeugen von Apigee-Datenverbindungen](#)

So erstellen Sie einen ADIF-Importprozess über den Assistenten:

- 1) Klicken Sie in der Registerkarte **ADIF** in Alfabet Expand mit der rechten Maustaste auf den Root-Knoten **ADIF-Schemata** im Explorer oder in einem beliebigen Unterordner, und wählen Sie **Importschema erzeugen** 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 Berichtverzeichnisses in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.
 - **Assistent:** Wählen Sie `ApigeeImport_Assistant` aus dem Dropdown-Listefeld aus.
 - **Import-Datei erforderlich:** Stellen Sie sicher, dass der Wert `False` festgelegt wurde.
 - **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 vom Typ **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.
- 3) Klicken Sie im Explorer mit der rechten Maustaste auf den Knoten des neuen ADIF-Importschemas, und wählen Sie **Erzeugen der ADIF-Schema-Details mithilfe von Apigee-Importassistent** aus. Der Assistent wird in Ihrem Standard-Webbrowser geöffnet.
- 4) Geben Sie auf der Registerkarte **Eintrag importieren** Folgendes an:
- **Eintragsname importieren:** Geben Sie einen eindeutigen Namen für den Importeintrag ein, der im ADIF-Schema für den Import der Daten aus Apigee erzeugt wird.
 - **Apigee-Verbindung:** Wählen Sie aus der Dropdown-Liste der verfügbaren Apigee-Datenverbindungen die Apigee-Datenverbindung aus, die für den Datenimport verwendet werden soll.

- **Zu importierende Daten:** Wählen Sie die Daten aus, die Sie von Apigee importieren möchten. Sie können API-Vertreter, Applikation, Statistik, Richtlinien, API-Produkte, Entwickler und Unternehmen importieren.
- 5) Wenn Sie „Statistik“ für den Import ausgewählt haben, müssen Sie die Art und Weise, wie Statistiken importiert werden, auf der Registerkarte **Statistikfilter** weiter definieren:
- **Umgebung:** Wählen Sie die entsprechende Apigee-Umgebung für die Generierung von Statistikdaten aus.
 - **Dimensionen:** Wählen Sie die Dimensionen aus, die in die Daten eingeschlossen werden sollen.
 - **Metrik:** Wählen Sie die Metriken aus, die für die ausgewählten Dimensionen berechnet werden sollen.
 - **Funktionen:** Wählen Sie eine Aggregierungsfunktion aus, um die Daten zu berechnen. Sie können den Durchschnitt (AVG), den Höchstwert (Max.), den Mindestwert (Min.) oder die Summe aller Daten zurückgeben.
 - **Zeiteinheit:** Wählen Sie das Intervall aus, das für die Berechnung von Zeiten und Daten verwendet werden soll.
 - **Transaktionshistorie-Zeitraum:** Geben Sie die Anzahl der Tage in der Vergangenheit ein, für die die Statistiken importiert werden sollen.
 - **Filterabfrage:** Geben Sie ggf. eine Apigee-Filterabfrage ein, um nach API-Vertretern zu filtern, die in die Statistiken eingeschlossen werden sollen.
- 6) Wechseln Sie zur Registerkarte **Eintrag importieren**, und klicken Sie auf **Importeintrag speichern**. Der Importeintrag wird im Feld **Zu bearbeitenden/entfernenden Importeintrag auswählen** aufgelistet.



Wenn Sie auf der Registerkarte **Statistikfilter** Daten hinzugefügt haben, müssen Sie nicht zur Registerkarte **Eintrag importieren** zurückkehren. Sie können den aus Ihrer Definition resultierenden ADIF-Importeintrag direkt über die Schaltfläche **Importeintrag** auf der Registerkarte **Statistikfilter** speichern.

- 7) Sie können folgende Aktionen ausführen:
- **Neuen Importeintrag erzeugen:** Geben Sie die Daten wie oben in den entsprechenden Feldern beschrieben mit einem anderen Namen im Feld **Eintragsname importieren** an, und klicken Sie auf **Importeintrag speichern**, um den nächsten Importeintrag zu speichern.
 - **Vorhandenen Importeintrag bearbeiten:** Aktivieren Sie das Kontrollkästchen neben dem Namen des Importeintrags. Ändern Sie die Daten in den entsprechenden Feldern des Editors, und klicken Sie auf **Importeintrag speichern**, um die Änderungen für den vorhandenen Importeintrag zu speichern.
 - **Vorhandenen Importeintrag löschen:** Aktivieren Sie das Kontrollkästchen neben dem Namen des Importeintrags. Klicken Sie auf **Ausgewählten Importeintrag entfernen**.
- 8) Klicken Sie auf die Schaltfläche **OK**, damit das ADF-Schema die angegebenen Inhalte aus Apigee abrufen und der **Apigee-Importassistent** geschlossen wird. Die temporären Tabellen werden generiert und im Explorer unter dem Importschema-Knoten angezeigt. Oder klicken Sie auf **Abbrechen**, um den Editor zu schließen, ohne die Ausführung des Importschemas auszulösen. Wenn Sie auf **Abbrechen** klicken, werden Ihre Einstellungen im **Apigee-Importassistenten** nicht gespeichert.

- 9) Schließen Sie das Browserfenster, und führen Sie in Alfabet Expand **Metamodell > Erneut aus Datenbank lesen** aus. Sie sehen dann alle automatisch generierten ADIF-Elemente im ADIF-Schema. Jeder ADIF-Importeintrag verfügt über Datenimporte in temporären Tabellen, die über die Attributelemente im Ordner **Attribute** definiert werden. Der Import in die Alfabet-Datenbank - Standardtabellen ist nicht in der Konfiguration enthalten.
- 10) Um die Integration der Daten in die Alfabet-Datenbank abzuschließen, ist eine zusätzliche Konfiguration des Importschemas erforderlich, um die importierten Apigee-Daten in den temporären Tabellen den für die entsprechenden Objektklassen und Objekten von Alfabet erstellten Tabellen zuzuordnen. Welche Konfiguration für diesen letzten Schritt des Imports erforderlich ist, hängt davon ab, wie Ihr Unternehmen die Apigee-Daten in Alfabet implementieren möchte. Weitere Informationen zu den über ADIF-Importschemata verfügbaren Konfigurationsoptionen finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

Nachdem Sie das ADIF-Schema definiert haben, müssen Sie es mit einer der Methoden ausführen, die im Kapitel *Konfigurieren von ADIF-Schemata* des Referenzhandbuchs *Alfabet-Datenintegrationsframework* beschrieben sind.

Kapitel 9: Konfigurieren der Erstellung bzw. des Exports von technischen Services auf Grundlage von WSDL- und OpenAPI-Spezifikationsdateien

Technische Services können basierend auf WSDL-Dateien und Swagger-Dateien gemäß OpenAPI-Spezifikation erstellt werden. Die Ansicht **Technische Services** (`COM_TechServices`) enthält die Optionen **Neue(n) technische(n) Services aus WSDL erzeugen** und **Technischen Service aus der OpenAPI-Spezifikation erzeugen**. Technische Services in Alfabet können mit den Optionen **In WSDL exportieren** und **In die OpenAPI-Spezifikation exportieren** auch in die Formate WSDL und JSON exportiert werden.



Alfabet unterstützt nur Swagger 2.0 JSON für die Erzeugung technischer Services. Frühere Versionen von Swagger werden nicht unterstützt.



Informationen zur Konfiguration von technischen Services finden Sie im Abschnitt *Konfigurieren von Objektklassenstereotypen für technische Services* im Kapitel *Konfigurieren von in der Lösungsumgebung implementierten alfabet-Funktionalitäten* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Beachten Sie die folgenden Konfigurationsanforderungen, damit dies möglich ist.

- Das XML-Objekt **TechServiceManager** muss folgendermaßen konfiguriert werden: Zwecks Zuordnung für die Klasse relevanter Stereotypen wurde der Service erweitert und umfasst jetzt auch die Erstellung technischer Services auf Grundlage von WSDL-Dateien und mit den OpenAPI-Spezifikationen kompatiblen Swagger-Dateien. Dem XML-Element `Stereotype`, mit dem das Klassenstereotyp für die Klasse `Service` festgelegt werden kann, wurde das neue XML-Element `FileExtension` hinzugefügt. Der Wert `".wsdl"` sollte in das XML-Element `FileExtensions` eingegeben werden, das für ein relevantes Objektklassenstereotyp festgelegt ist, damit in der Ansicht **Technische Services** (`COM_TechServices`) die Optionen **Neue(n) technische(n) Services aus WSDL erzeugen** und **In WSDL exportieren** aktiviert werden. Der Wert `".json"` sollte festgelegt werden, damit in der Ansicht **Technische Services** die Optionen **Technischen Service aus der OpenAPI-Spezifikation erzeugen** und **In die OpenAPI-Spezifikation exportieren** aktiviert werden.
- Geben Sie für jeden technischen Service-Stereotyp an, welche Operationen des technischen Services zulässig sind.
- Erzeugen Sie unter dem Root-XML-Element **TechServiceManager** ein untergeordnetes XML-Element `Stereotype`, und geben die folgenden XML-Attribute an:



Das nachfolgende Beispiel zeigt das Mapping für den Stereotyp `SOAPService`, der für die Objektklasse `Service` konfiguriert ist:

```
<TechServiceManager>
  <Stereotype ClassName="Service" Name="SOAPService"
    FileExtension=".wsdl">
    <Stereotype ClassName="ServiceOperation"
      Name="SOAPOperation">
      <Stereotype ClassName="ServiceOperationMethod"
        Name="AHTTPPOST" MethodType="HTTPPOST">
```

```
<Stereotype ClassName="ServiceOperationMethod"
Name="AHTTPGET" MethodType="HTTPGET" />

<Stereotype ClassName="ServiceOperationMethod"
Name="ASOAP12" MethodType="SOAP12"/>

<Stereotype ClassName="ServiceOperationMethod"
Name="ASOAP" MethodType="SOAP"/>

</Stereotype>

</Stereotype>

</TechServiceManager>
```

- **ClassName:** Geben Sie `Service` (oder `SolutionService`) ein, um das parallele Mapping für die Lösungsplanung zu erzeugen:
- **Name:** Geben Sie den technischen Namen des Objektklassenstereotyps ein, der für die Objektklasse `Service` (oder `SolutionService`) definiert wurde. Dies ist der Name, der im XML-Attribut `Name` des Attributs **Stereotypen** der Objektklasse definiert ist.



Sie müssen den technischen Namen des technischen Service-Stereotyps genau in der Schreibweise eingeben, wie er im Attribut **Stereotypen** der entsprechenden Objektklasse angegeben ist.

- **FileExtension:** Geben Sie für ein relevantes Objektklassenstereotyp `".wsdl"` ein, um in der Ansicht **Technische Services** (`COM_TechServices`) die Optionen **Neue(n) technische(n) Services aus WSDL erzeugen** und **In WSDL exportieren** zu aktivieren. Geben Sie für ein relevantes Objektklassenstereotyp `".json"` ein, um in der Ansicht **Technische Services** die Optionen **Technischen Service aus der OpenAPI-Spezifikation erzeugen** und **In die OpenAPI-Spezifikation exportieren** zu aktivieren.
- Geben Sie für jeden Operations-Stereotyp des technischen Services an, welche Methoden der Operation des technischen Services zulässig sind.
- Wenn Ihr Unternehmen die Lösungsplanung unterstützt und technische Services sowie Operationen des technischen Services Bestandteil der Soll-Architektur sein sollen, müssen Sie für die entsprechenden Lösungsobjektklassen `SolutionService`, `SolutionServiceOperation` und `SolutionServiceMethod` eine identische Konfiguration erzeugen.
- Die Dateierweiterungen `.wsdl` und `.json` dürfen nicht in der Blacklist enthalten sein, die im XML-Objekt **FileExtensions** festgelegt ist. Wenn eine Whitelist mit Dateierweiterungen aktiviert ist, müssen die Dateierweiterungen in die Whitelist aufgenommen werden. Der Download einer ZIP-Datei mit mehreren WSDL- oder JSON-Dateien ist nicht möglich. Weitere Informationen zum Konfigurieren der Zulässigkeit von Dateierweiterungen finden Sie im Abschnitt *Konfiguration der Zulässigkeit von Dateien und Weblinks in Alfabet* des Referenzhandbuchs *Konfigurieren von Alfabet mit Alfabet Expand*.
- In der relevanten Klasseneinstellung für die Klasse `Service` muss im Attribut **Assistent** ein geschützter oder benutzerdefinierter Assistent festgelegt werden, damit ein technischer Service auf Basis einer WSDL- oder JSON-Datei erzeugt werden kann. (Beachten Sie, dass diese Definition NICHT für den Objektklassenstereotyp für die Klasse `Service` vorgenommen werden sollte.) Die Spezifikation des Assistenten ist erforderlich, damit gewährleistet ist, dass ein dem Assistenten vorgelagerter Schritt implementiert wird, mit dem die relevante Datei in Alfabet importiert werden und der technische Service somit erzeugt werden kann.

- In der relevanten Klasseneinstellung für die Klasse `Service` kann im Attribut **Assistent** ein geschützter oder benutzerdefinierter Assistent festgelegt werden, damit ein technischer Service auf Basis einer WSDL- oder Open API-Spezifikation erzeugt werden kann.

Kapitel 10: Konfigurieren der Integration der Daten von Amazon Web Services

Alfabet 10.9 stellt eine neue Schnittstelle mit Amazon Web Services bereit. Über die Schnittstelle können Daten über die Gesamtheit der Applikationen und Services, die Ihr Unternehmen in der Infrastruktur der Amazon Web Services betreibt (zum Beispiel die Komponenten, die Amazon Web Services Ihrem Unternehmen bereitstellt), eingeholt und auf Basis dieser Daten Objekte in der Alfabet-Datenbank erzeugt und aktualisiert werden. Wie die Daten in die Alfabet-Datenbank integriert werden ist nicht vorkonfiguriert. Jeder Kunde definiert die Importlogik nach Bedarf. Die Schnittstelle mit den Amazon Web Services (AWS-Schnittstelle) erfordert die folgende Konfiguration:

- Die Verbindung mit den Amazon Web Services muss im XML-Objekt **AmazonWebServicesConfig** in Alfabet Expand definiert werden.
- Welche Daten importiert werden sollen, muss im XML-Objekt **AmazonWebServicesConfig** in Alfabet Expand näher definiert werden.
- Der Import von Daten aus Amazon Web Services wird über ein ADIF-Importschema verwaltet, das über einen Assistenten zur Implementierung der erforderlichen Basiskonfiguration erstellt wird. Der Assistent erstellt eine Reihe von ADIF-Importsätzen im ADIF-Importschema. Die Importsätze sind dazu konfiguriert, Daten aus Amazon Web Services in temporäre Datenbanktabellen in der Alfabet-Datenbank zu importieren. Das ADIF-Importschema muss mit der kundendefinierten Importlogik ergänzt werden, über die die Daten aus den temporären Dateien in die Datenbanktabellen für das Objektklassenmodell in der Alfabet importiert werden.

Um den Datenimport aus Amazon Web Services auszuführen, müssen Sie den vom ADIF-Importschema ausgelösten ADIF-Import starten. Beim ADIF-Import werden dann sowohl der Datenabruf aus Amazon Web Services als auch die Datenintegration in die Alfabet-Datenbank ausgeführt. Der ADIF-Import sollte in regelmäßigen Abständen ausgeführt werden, um die Datenkonsistenz zwischen Amazon Web Services und Alfabet zu gewährleisten.



Informationen zur Ausführung von ADIF-Schemata finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

In den folgenden Abschnitten werden die Konfigurationsschritte für die Schnittstelle mit Amazon Web Services in der erforderlichen Reihenfolge der Ausführung beschrieben:

- [Konfigurieren der Verbindung mit Amazon Web Services](#)
- [Senden von Anforderungen über Proxyserver zu Amazon Web Services](#)
- [Konfigurieren der Integration von Amazon Web Services-Daten in die Alfabet-Datenbank](#)

Konfigurieren der Verbindung mit Amazon Web Services

Die Verbindung mit den Amazon Web Services muss im XML-Objekt **AmazonWebServicesConfig** definiert werden. Dabei können mehrere Verbindungen mit Amazon Web Services konfiguriert werden. Daten werden aus allen definierten Amazon Web Services über das gleiche ADIF-Importschema importiert.

Die allgemeine Struktur der XML im XML-Objekt **AmazonWebServicesConfig** sieht wie folgt aus:

```
<AmazonWebServicesConfig>
```

```
<DataConnections>
  <DataConnection
    Name="Account1"
    IsActive="true"
    AWSSecretKey="$AWSSecretKey"
    AWSAccessKey="$AWSAccessKey"
    AWSAccount="$AWSAccountNumber">
    <Regions>
      <Region>eu-west-1</Region>
      <Region>us-west-1</Region>
      <Region>ap-southeast-1</Region>
    </Regions>
  </DataConnection>
</DataConnections>
</AmazonWebServicesConfig>
```

Das Root-XML-Element `AmazonWebServicesConfig` verfügt über ein untergeordnetes XML-Element ***DataConnections***. Das XML-Element ***DataConnections*** kann mehrere untergeordnete XML-Elemente ***DataConnection*** enthalten. Jedes von ihnen definiert eine Verbindung zu einem anderen Zugang zu den Amazon Web Services.

So konfigurieren Sie die Verbindung mit den Amazon Web Services:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **AmazonWebServicesConfig**, und wählen Sie **XML bearbeiten....** Das XML-Objekt wird geöffnet.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist eine Vorlage über das Attribut `XML Template` des XML-Objekts verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf. Bei einer Beispielformatierung müssen Sie XML-Elemente bearbeiten, anstatt neue hinzuzufügen.

- 3) Fügen Sie dem Root-XML-Element `AmazonWebServicesConfig` ein untergeordnetes XML-Element `DataConnections` hinzu.
- 4) Fügen Sie dem XML-Element `DataConnections` ein untergeordnetes XML-Element `DataConnection` hinzu, und definieren Sie die folgenden XML-Attribute für das Element `DataConnection`:
 - **Name:** Der Name der Verbindung zu den Amazon Web Services, mit dem die Verbindung in der Konfiguration des ADIF-Schemas identifiziert wird.
 - **IsActive:** Auf `true` setzen, wenn Daten aus der Amazon Web Services-Instanz importiert werden sollen. Auf `false` setzen, wenn Daten derzeit nicht aus der Amazon Web Services-Instanz importiert werden sollen.
 - **AWSAccessKey:** Die Zugriffsschlüssel-ID des Zugriffsschlüssels für die Verbindung mit den Amazon Web Services.

- **AWSecretKey:** Der geheime Zugriffsschlüssel für die Verbindung mit den Amazon Web Services.
- **AWSAccount:** Geben Sie die ID Ihres Kontos bei Amazon Web Services ein.



Beachten Sie Folgendes:

- In den XML-Elementen zur Definition der Verbindung mit den Amazon Web Services können Servervariablen verwendet werden, um den Wert des XML-Attributs zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation auszulesen, wenn eine Verbindung zu Amazon Web Services hergestellt wurde. Weitere Informationen zu Servervariablen finden Sie unter [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).
 - Amazon empfiehlt, den Zugriffsschlüssel in regelmäßigen Abständen zu ändern. Aus Sicherheitsgründen sollten Sie Ihre Schlüssel ändern und die Konfiguration der XML-Attribute `AWSecretKey` und `AWSAccessKey` entsprechend aktualisieren.
- 5) Fügen Sie dem XML-Element `DataConnection` optional ein untergeordnetes XML-Element `Regions` hinzu, um die Region zu definieren, in der sich die Amazon Web Services befinden. Fügen Sie dem XML-Element `Regions` für jede Region ein untergeordnetes XML-Element `Region` hinzu, und geben Sie den Amazon-Namen der Region als Text in dieses XML-Element ein.
- 6) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Senden von Anforderungen über Proxyserver zu Amazon Web Services

Optional können Sie die Integrationsschnittstelle für Amazon Web Services dazu konfigurieren, Anforderungen an die Amazon Web Services über einen Proxyserver zu senden. Dies erfordert für die oben beschriebene Datenimport-Konfiguration die folgende zusätzliche Konfiguration im XML-Objekt ***AmazonWebServicesConfig***:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf ***AmazonWebServicesConfig***, und wählen Sie **XML bearbeiten....** Das XML-Objekt mit der Datenimport-Definition für die Amazon Web Services wird geöffnet.
- 3) Fügen Sie dem XML-Element `AmazonWebServicesConfig` ein untergeordnetes XML-Element `Proxy` hinzu.
- 4) Definieren Sie im XML-Element `Proxy` die folgenden XML-Attribute:
 - **Url:** Definieren Sie die URL des Proxyservers.
 - **UserName:** Geben Sie bei Bedarf den Anwendernamen für den Zugriff auf den Proxyserver ein. Der Domänenname für die Authentifizierung wird separat über das XML-Attribut „Domäne“ definiert und muss nicht als Teil des Anwendernamens angegeben werden.
 - **Password:** Geben Sie bei Bedarf das Passwort für den Zugriff auf den Proxyserver ein.

- **Domain:** Definieren Sie bei Bedarf den Domännennamen, der als Teil des Anwendernamens zur Authentifizierung am Proxyserver verwendet werden soll.
- 5) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Wenn Sie mehrere Datenverbindungen konfiguriert haben und jede Datenverbindung einen anderen Proxy verwenden soll, können Sie Ihrer Proxykonfiguration zusätzliche Proxys hinzufügen und in der Datenverbindungskonfiguration auf einen der Proxys verweisen. Die obige Proxydefinition wird als Standard verwendet, wenn einer Datenverbindung kein Proxy zugeordnet ist. Zur Verwendung eines zusätzlichen Proxys ist die folgende Konfiguration erforderlich:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **AmazonWebServicesConfig**, und wählen Sie **XML bearbeiten....** Das XML-Objekt mit der Datenimport-Definition für die Amazon Web Services wird geöffnet.
- 3) Fügen Sie dem XML-Element `Proxy` ein untergeordnetes XML-Element `AdditionalProxies` hinzu.
- 4) Fügen Sie dem XML-Element `AdditionalProxies` für jeden zusätzlichen Proxy, den Sie definieren möchten, ein untergeordnetes XML-Element `AdditionalProxy` hinzu, und definieren Sie die folgenden XML-Attribute für das XML-Element `AdditionalProxy`:
 - **Name:** Definieren Sie einen eindeutigen Namen für den zusätzlichen Proxy. Der Name wird verwendet, um in der Datenverbindungskonfiguration auf den Proxy zu verweisen.
 - **Url:** Definieren Sie die URL des Proxyservers.
 - **UserName:** Geben Sie bei Bedarf den Anwendernamen für den Zugriff auf den Proxyserver ein. Der Domännename für die Authentifizierung wird separat über das XML-Attribut „Domäne“ definiert und muss nicht als Teil des Anwendernamens angegeben werden.
 - **Password:** Geben Sie bei Bedarf das Passwort für den Zugriff auf den Proxyserver ein.
 - **Domain:** Definieren Sie bei Bedarf den Domännennamen, der als Teil des Anwendernamens zur Authentifizierung am Proxyserver verwendet werden soll.
- 5) Fügen Sie jedem XML-Element **DataConnection**, das einen der zusätzlichen Proxys verwenden soll, ein XML-Attribut `Proxy` hinzu. Der Wert des XML-Attributs `Proxy` muss mit dem Wert des XML-Attributs `Name` des XML-Elements **AdditionalProxy** übereinstimmen.
- 6) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Konfigurieren der Integration von Amazon Web Services-Daten in die Alfabet-Datenbank

Der Datenimport aus Amazon Web Services wird durchgeführt, indem ein ADIF-Importschema ausgeführt wird. Es löst die Datenübertragung von Amazon Web Services auf der Grundlage der konfigurierten Verbindungsangaben aus, speichert die eingehenden Daten auf vordefinierte Art in temporären Tabellen und

integriert die Daten in die Standard-Objektklassentabellen der Alfabet-Datenbank, wie sie vom Kunden im ADIF-Importschema konfiguriert wurden.

Auf der Registerkarte „ADIF“ von Alfabet Expand kann das Importschema mithilfe eines Assistenten konfiguriert werden. Beim Öffnen des Assistenten können Sie eine oder mehrere Datenverbindungen auswählen, die im XML-Objekt **AmazonWebServicesConfig** definiert sind. Beim Schließen des Assistenten wird das ADIF-Schema automatisch mit den erforderlichen Einträgen ausgefüllt, um die Standarddaten aus Amazon Web Services in temporäre Datenbanktabellen einzulesen. Um die Integration der Daten in die Alfabet-Datenbank zu definieren, müssen die Einträge aus der halbautomatischen Erzeugung der ADIF-Importschema-Elemente mit Informationen dazu ergänzt werden, wie Daten aus den temporären Tabellen in die Alfabet-Datenbank integriert werden sollen. Welche Konfiguration für diesen letzten Schritt des Imports erforderlich ist, hängt stark von Ihren Anforderungen ab.



In diesem Abschnitt wird nur die Handhabung des Amazon Web Services-Assistenten beschrieben. Ausführliche Informationen über die Konfiguration eines ADIF-Importschemas, um Daten aus den temporären Datenbanktabellen in die Alfabet-Datenbank zu schreiben, finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*, sie werden hier nicht wiederholt.

Die Nutzung des Amazon Web Services-Assistenten ermöglicht nicht nur die halbautomatische Konfiguration des ADIF-Importschemas. Es werden auch die erforderlichen Mechanismen eingerichtet, um die Verbindung zu Amazon Web Services zu öffnen und die Daten zu importieren, wenn der ADIF-Import auf der Basis des ADIF-Importschemas ausgeführt wird.

Nach der Definition einer gültigen Konfiguration für den Datenimport im XML-Objekt `AmazonWebServicesConfig` müssen Sie ein neues ADIF-Importschema definieren, um die Datenintegration mithilfe des Amazon Web Services-Importassistenten zu konfigurieren:

- 1) Klicken Sie in der Registerkarte **ADIF** in Alfabet Expand mit der rechten Maustaste auf den Root-Knoten **ADIF-Schemata** im Explorer oder in einem beliebigen Unterordner, und wählen Sie **Importschema erzeugen** 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 Berichtverzeichnisses in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.
 - **Assistent:** Wählen Sie `AWSImport_Assistant` aus dem Dropdown-Listenfeld aus.
 - **Import-Datei erforderlich:** Wählen Sie `False` aus.
 - **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 vom Typ **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.
- 3) Klicken Sie im Explorer mit der rechten Maustaste auf den Knoten des neuen ADIF-Importschemas, und wählen Sie **Erzeugen der ADIF-Schema-Details mithilfe von AWS-Importassistent** aus. Eine Warnmeldung wird angezeigt.



Wenn Sie den Assistenten mit einem vorhandenen, bereits konfigurierten ADIF-Importschema verwenden, werden dadurch alle automatisch generierten Teile des ADIF-Schemas überschrieben. Falls Änderungen an diesen Importeinträgen vorgenommen wurden, gehen diese Änderungen verloren.

- 4) Klicken Sie auf **Ja**. Der Assistent wird in Ihrem Standard-Webbrowser geöffnet.
- 5) Im Feld **Auszuführende Datenverbindungen auswählen** werden alle XML-Elemente `DataConnection` in Ihrem XML-Objekt `AmazonWebServicesConfig` angezeigt. Klicken Sie auf jede Datenverbindung, für die Daten in dieses ADIF-Importschema integriert werden sollen.
- 6) Klicken Sie nach der Auswahl aller relevanten `DataConnection`-XML-Elemente auf die Schaltfläche **OK** unter dem Feld. Eine neue Seite wird angezeigt, die Sie daran erinnert, die ADIF-Struktur in Alfabet Expand neu abzufragen.

- 7) Schließen Sie das Browserfenster, und kehren Sie zur Registerkarte **ADIF** von Alfabet Expand zurück.
- 8) Klicken Sie mit der rechten Maustaste auf das ADIF-Importschema, und wählen Sie im Kontextmenü die Option **Struktur neu abfragen** aus. Sie sehen dann alle automatisch generierten ADIF-Elemente im ADIF-Schema. Jeder ADIF-Importeintrag verfügt über Datenimporte in temporären Tabellen, die über die Attributelemente im Ordner **Attribute** definiert werden. Der Import in die Alfabet-Datenbank -Standardtabellen ist nicht in der Konfiguration enthalten.
- 9) Konfigurieren Sie die Datenintegration in die Alfabet-Datenbank -Standardtabellen entsprechend Ihren Anforderungen. Weitere Informationen zu den über ADIF-Importschemata verfügbaren Konfigurationsoptionen finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

Kapitel 11: Konfigurieren der Integration von Daten zwischen ServiceNow und Alfabet

Alfabet bietet eine Schnittstelle mit ServiceNow, um die in ServiceNow verfügbaren Daten in Alfabet zu importieren und umgekehrt. Mit dieser Funktionalität können Sie Daten über gehostete ServiceNow-Services in die Alfabet-Datenbank importieren, um sie in Ihre IT-Verwaltungs- und Planungsprozesse einzubeziehen und Daten über geplante und genehmigte Services in ServiceNow zu importieren.

Die Datenübertragung erfolgt über RESTful-API-Aufrufe, die durch Alfabet ADIF-Aufträge ausgelöst werden. Der Import und der Export werden zwar durch Alfabet ausgelöst, ServiceNow muss aber trotzdem dazu konfiguriert werden, die erforderlichen Zugriffsrechte bereitzustellen und, beim Import von Daten von ServiceNow in Alfabet, die Daten im erforderlichen Format bereitzustellen. Bitte beachten Sie, dass die Konfiguration des Produkts ServiceNow in dieser Dokumentation nicht erläutert wird. Die durch Alfabet bereitgestellte Dokumentation behandelt die Konfiguration, die in Alfabet zur Integration von Daten zwischen ServiceNow und Alfabet erforderlich ist. Informationen zum Konfigurieren des ServiceNow-Produkts finden Sie in der im Lieferumfang des Produkts enthaltenen Dokumentation.



Die Integration von Alfabet in ServiceNow funktioniert nicht, wenn Authentifizierungsanforderungen von Alfabet auf eine SSO-Plattform umgeleitet werden.

Folgende Informationen sind verfügbar:

- [Konfigurieren der Integration von Daten von ServiceNow](#)
 - [Konfigurieren der Datenübertragung von ServiceNow zu Alfabet](#)
 - [Senden einer Anforderung über Proxyserver an ServiceNow](#)
 - [Konfigurieren der Integration von ServiceNow-Daten in die Alfabet-Datenbank](#)
 - [Ändern einer vorhandenen Konfiguration für die ServiceNow-Integration](#)
- [Konfigurieren der Integration von Alfabet -Daten in ServiceNow](#)
 - [Definieren von zu exportierenden Daten aus der Alfabet-Datenbank über einen konfigurierten Bericht](#)
 - [Konfigurieren der Datenübertragung von Alfabet zu ServiceNow](#)
 - [Senden einer Anforderung über Proxyserver an ServiceNow](#)
 - [Konfigurieren des ADIF-Exportschemas für den Datenexport in ServiceNow](#)
 - [Ändern einer vorhandenen Konfiguration für die ServiceNow-Integration](#)

Konfigurieren der Integration von Daten von ServiceNow

Wenn Sie ServiceNow verwenden, können Sie Daten über Ihre gehosteten ServiceNow-Services in die Alfabet-Datenbank importieren, um sie in Ihre IT-Verwaltungs- und Planungsprozesse zu integrieren. Es gibt keine vordefinierte Objektklasse zur Speicherung der eingehenden Daten. Die Objektklassen und Objekt-klassenattribute, die für die Speicherung der Informationen relevant sind, werden über die Integrations-schnittstelle vollständig vom Kunden konfiguriert.

Daten über gehostete ServiceNow-Services können aus ServiceNow-Datenbanktabellen, ServiceNow-Datenbankansichten und ServiceNow-Berichten importiert werden. Zur Integration von Daten aus ServiceNow in die Alfabet-Datenbank muss die vorhandene Schnittstelle mit den folgenden Informationen konfiguriert werden:

- Zugriffsinformationen für die Herstellung der Verbindung mit ServiceNow. Die Datenübertragung erfolgt über RESTful-Webdienste.
- Angaben zur Quelle und Art der zu importierenden Daten und zu den HTTP-Übertragungsdetails.
- Zuordnung zwischen ServiceNow- und Alfabet -Datentypen.
- Die Objektklassen und Objektklasseneigenschaften, die das Ziel des Imports in die Alfabet-Datenbank sind, und die Art, wie Objektdaten in die vorhandenen Objektdaten in der Alfabet-Datenbank integriert werden.

Der Zugriff auf ServiceNow und die Spezifikation des Datenimports werden im XML-Objekt `ServiceNowImportConfig` konfiguriert.

Diese Konfiguration ist als Voraussetzung für den zweiten Konfigurationsschritt erforderlich, in dem die Erstellung und Aktualisierung von Objektdaten in der Alfabet-Datenbank definiert werden. Diese Datenintegration erfolgt über einen ADIF-Import, und gemäß den Anforderungen müssen eines oder mehrere ADIF-Importschemata erzeugt und konfiguriert werden.

Um den Datenimport aus ServiceNow auszuführen, müssen Sie den vom ADIF-Importschema ausgelösten ADIF-Import starten. Beim ADIF-Import werden sowohl der Datenabruf aus ServiceNow als auch die Datenintegration in die Alfabet-Datenbank ausgeführt. Der ADIF-Import sollte in regelmäßigen Abständen ausgeführt werden, um die Datenkonsistenz zwischen ServiceNow und Alfabet zu gewährleisten.



Informationen zur Ausführung von ADIF-Schemata finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

In den folgenden Abschnitten werden die Konfigurationsschritte für die Schnittstelle mit ServiceNow in der erforderlichen Reihenfolge der Ausführung beschrieben:

- [Konfigurieren der Datenübertragung von ServiceNow zu Alfabet](#)
- [Senden einer Anforderung über Proxyserver an ServiceNow](#)
- [Konfigurieren der Integration von ServiceNow-Daten in die Alfabet-Datenbank](#)
- [Ändern einer vorhandenen Konfiguration für die ServiceNow-Integration](#)

Konfigurieren der Datenübertragung von ServiceNow zu Alfabet

Die Konfiguration des Datenimports erfolgt im XML-Objekt **`ServiceNowImportConfig`**. In dem XML-Objekt können Verbindungen mit mehreren ServiceNow-Quellen definiert werden. Nachfolgend wird die allgemeine Struktur der XML im XML-Objekt **`ServiceNowImportConfig`** angezeigt:

```
<ServiceNowConfig>
  <DataTransferMappings>
    <DataTransferMapping>
      <ServiceAccessInfo>...</ServiceAccessInfo>
```

```

        <DataConnectivityInfo>...</DataConnectivityInfo>
    </DataTransferMapping>
</DataTransferMappings>
<DataTypeMappings>...</DataTypeMappings>

```

Das Root-XML-Element `ServiceNowConfig` der XML-Objekte enthält zwei untergeordnete Elemente:

- **DataTransferMappings:** Das XML-Element `DataTransferMappings` kann ein oder mehrere untergeordnete Elemente `DataTransferMapping` enthalten. Jedes untergeordnete Element `DataTransferMapping` enthält die Definition einer Verbindung zu einem anderen ServiceNow-Server. Die Definition beinhaltet in einem untergeordneten Element `ServiceAccessInfo` die Spezifikation der Verbindung zum REST-API des Webdienstes des ServiceNow-Servers und in einem untergeordneten Element `DataConnectivityInfo` die Spezifikation der zu übertragenden Daten. Im Element `DataConnectivityInfo` können unterschiedliche Datenübertragungsbedingungen für unterschiedliche Datentypen definiert werden.
- **DataTypeMappings:** Das XML-Element `DataTypeMappings` enthält die Zuordnung der importierten Datentypen zu den in der Alfabet-Datenbank gültigen Datentypen. Die Daten werden während des Datenimports in die angegebenen Datentypen konvertiert. Die Datentypdefinitionen gelten für alle Datenverbindungen in allen Datenübertragungszuordnungen.

So definieren Sie den Datenimport von einem ServiceNow-Server:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **ServiceNowImportConfig**, und wählen Sie **XML bearbeiten...** Das XML-Objekt wird geöffnet.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist eine Vorlage über das Attribut **XML-Vorlage** des XML-Objekts **ServiceNowImportConfig** verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf. Wenn Sie mit einer Beispielformatierung beginnen, müssen Sie die XML-Elemente bearbeiten und nicht neu hinzufügen.

- 3) Fügen Sie dem Root-XML-Element `ServiceNowConfig` ein untergeordnetes XML-Element `DataTransferMappings` hinzu.
- 4) Fügen Sie dem XML-Element `DataTransferMappings` ein untergeordnetes XML-Element `DataTransferMapping` hinzu, und definieren Sie das folgende XML-Attribut für das XML-Element `DataTransferMapping`:
 - **Name:** Der Name der Verbindung zum ServiceNow-Server, mit dem die Verbindung in der Konfiguration des ADIF-Schemas identifiziert wird.
 - **IsActive:** Auf `true` setzen, wenn Daten aus der ServiceNow-Instanz importiert werden sollen. Auf `false` setzen, wenn Daten derzeit nicht aus der ServiceNow-Instanz importiert werden sollen.
- 5) Fügen Sie dem XML-Element `DataTransferMapping` ein untergeordnetes XML-Element `ServiceAccessInfo` hinzu, und definieren Sie die Informationen, die zum Einrichten einer Verbindung mit dem ServiceNow-Server erforderlich sind. Das XML-Element `ServiceAccessInfo` hat keine XML-Attribute. Die Informationen, die zum Einrichten einer Verbindung mit dem REST-API der Webdienste des ServiceNow-Servers erforderlich sind, müssen als Zeichenfolgen den

folgenden untergeordneten XML-Elementen hinzugefügt werden, die dem XML-Element `ServiceAccessInfo` hinzugefügt werden müssen:

- `service`: Die URL der ServiceNow-Instanz.
- `apipath`: Der Pfad zum REST-API in der ServiceNow-Instanz. Geben Sie `/api/now/v1/` ein.
- `oauthpath`: OAuth ist zur Authentifizierung bei ServiceNow erforderlich. Sie müssen den Pfad zu dem OAuth-Endpoint mit diesem XML-Element angeben.
- `username`: Der Anwendername für den Zugriff auf ServiceNow.



Der Anwender muss die erforderlichen Zugriffsberechtigungen zur Durchführung folgender API-Aufrufe haben:

- Authentifizierungsaufruf: `https://<BaseAddress>/oauth_token.do`
- Lesen der Tabellen: `https://<BaseAddress>/<tableName>.do?SCHEMA`
- Lesen von Datenbankansichten:
`https://<BaseAddress>/<databaseviewname>.do?CSV&sysparm_record_count=1`
- Lesen von Berichten:
`https://<BaseAddress>/sys_report_template.do?CSV&sysparm_record_count=1&jvar_report_id=<reportid>`
- Lesen von Berichten:
`https://<BaseAddress>/api/now/v1/table/sys_report/<reportID>`

Folgende Zugriffsrechte würden beispielsweise ausreichen:

- `rest_service` für den Lesezugriff auf Daten über einen RESTful-API-Aufruf.
 - `soap_query` für den Lesezugriff auf die Schema-Informationen von ServiceNow-Datenbanktabellen oder Datenbankansichten. Diese Berechtigung ist für den Import aus Berichten nicht erforderlich.
 - `itil` für den Lesezugriff auf die Beziehungstabellen `cmdb_rel_ci` und `cmdb_rel_type`.
 - `itil_admin`
- `password`: Das Passwort für den Zugriff auf ServiceNow.
 - `client_id`: Geben Sie die Client-ID für den OAuth-Endpoint bei ServiceNow ein.
 - `client_secret`: Geben Sie den Clientschlüssel für den OAuth Endpoint bei ServiceNow in dieses XML-Element ein.



Zum Definieren der ServiceNow-Serververbindung können Servervariablen genutzt werden. Die Servervariablen lesen den Wert des XML-Elements zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation aus, wenn eine Verbindung zu ServiceNow hergestellt wurde. Weitere Informationen zu Servervariablen finden Sie unter [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).

- 6) Fügen Sie dem XML-Element `DataTransferMapping` ein untergeordnetes XML-Element `DataConnectivityInfo` hinzu, und definieren Sie die Informationen, die für die Datenanforderung vom ServiceNow-Server erforderlich sind. Das XML-Element `DataConnectivityInfo`. Definieren Sie die Standard-Datenübertragungsparameter für die Datenanforderungen, indem Sie die folgenden XML-Attribute für das XML-Element `DataConnectivityInfo` festlegen:
- `DefaultPageSize`: Für den Import aus Datenbanktabellen und Datenbankansichten mit vielen Datensätzen können Sie einen Grenzwert für die Daten festlegen, die gleichzeitig über die Datenverbindung übertragen werden sollen. Geben Sie die Anzahl der Datensätze ein, die gleichzeitig übertragen werden sollen. Dieser Wert gilt für alle Datenverbindungen, bei denen kein Attribut `PageSize` definiert wurde. Wenn weder für die Datenverbindung noch über das XML-Attribut `DefaultPageSize` eine Datenbeschränkung definiert wurde, wird standardmäßig der Wert 100 festgelegt. Das XML-Attribut `DefaultPageSize` wird für den Import aus Berichten nicht unterstützt.
 - `DefaultTimeout`: Definieren Sie das Standard-Zeitlimit für HTTP-Anforderungen, das für alle Datenverbindungen verwendet werden soll, für die kein Zeitlimit über das XML-Attribut `Timeout` des XML-Elements `DataConnection` definiert wurde. Wenn weder für die Datenverbindung noch über das XML-Attribut `DefaultTimeout` ein Zeitlimit definiert wurde, wird der für die Alfabet-Webapplikation definierte Standardwert verwendet.
 - `DefaultMaxRecordCount`: Definieren Sie die maximale Anzahl von Datensätzen, die für eine Anforderung an ServiceNow über eine Datenverbindung übertragen werden sollen, wenn die Daten aus einem ServiceNow-Bericht ausgelesen werden. Der hier definierte Wert wird als Standardwert für alle Datenverbindungen verwendet, für die keine maximale Anzahl von zu übertragenden Datensätzen über das XML-Attribut `MaxRecordCount` des XML-Elements `DataConnection` definiert wurde. Wenn weder für die Datenverbindung noch über das XML-Attribut `DefaultMaxRecordCount` eine maximale Anzahl von Datensätzen definiert wurde, wird standardmäßig der Wert 1000 festgelegt. Das XML-Attribut `DefaultMaxRecordCount` wird für den Import aus Datenbanktabellen und Datenbankansichten nicht unterstützt.
- 7) Fügen Sie dem XML-Element `DataConnectivityInfo` ein oder mehrere untergeordnete XML-Elemente `DataConnection` hinzu. Jedes XML-Element `DataConnection` bündelt Daten, die über eine Anforderung an den ServiceNow-Server importiert werden sollen. Definieren Sie die Datenübertragungsparameter für die Datenanforderung, indem Sie die folgenden XML-Attribute für das XML-Element `DataConnection` festlegen:
- `DataConnectionName`: Geben Sie einen eindeutigen Namen für die Datenverbindung ein. Dieser Name dient dazu, die Datenverbindung in der Konfiguration des ADIF-Schemas zu identifizieren.
 - `FetchType`: Geben Sie den Typ der Datenquelle an, die für den Abruf der Daten über die Verbindung verwendet wird. Zulässige Werte:
 - `Table`: Zum Auslesen von Daten aus einer ServiceNow-Datenbanktabelle.
 - `DatabaseView`: Zum Auslesen von Daten aus einer ServiceNow-Datenbankansicht.
 - `Report`: Zum Auslesen von Daten aus einem ServiceNow-Bericht.
 - `Timeout`: Definieren Sie das Zeitlimit für HTTP-Anforderungen für die Datenverbindung. Wenn kein Zeitlimit definiert wird, wird das im XML-Element `DataConnectivityInfo` definierte Standard-Zeitlimit verwendet.

- **PageSize:** Geben Sie die maximale Anzahl von Datensätzen an, die gleichzeitig auf der Datenverbindung übertragen werden sollen. Dieses Attribut wird für den Import aus Berichten und Datenbankansichten nicht unterstützt.



Es wird empfohlen, das XML-Attribut `PageSize` auf einen Wert unter 500 zu setzen. Eine zu hohe Anzahl von Datensätzen kann zu Übertragungsproblemen führen.

- **MaxRecordCount:** Definieren Sie die maximale Anzahl von Datensätzen, die für eine Anforderung an ServiceNow beim Datenimport aus Berichten übertragen werden sollen. Wenn das XML-Attribut `PageSize` nicht festgelegt wurde oder nicht unterstützt wird, sollte der Wert für das XML-Attribut `MaxRecordCount` maximal 10.000 betragen. Dieses Attribut wird für den Import aus Datenbanktabellen und Datenbankansichten nicht unterstützt.
 - **IncludeDisplayValue:** Setzen Sie das XML-Attribut `IncludeDisplayValue` auf `true`, wenn Sie zusätzlich zur Ganzzahl den definierten Anzeigewert für Ganzzahlen in ServiceNow aufnehmen möchten, beispielsweise die Priorität von Änderungsanforderungen. Der Anzeigewert wird als Zeichenfolge in eine separate Spalte der für den Datenimport erzeugten temporären Tabelle geschrieben. Der Ganzzahlwert und der Anzeigewert werden in zwei unterschiedliche Spalten der temporären Tabelle importiert, mit den Erweiterungen „_IV“ (Ganzzahlwert) und „_DV“ (Anzeigewert) für den Spaltennamen. Falls dieses XML-Attribut nicht gesetzt wird oder auf `false` gesetzt ist, wird nur der Ganzzahlwert importiert.
 - **Prefix:** Der Datenimport wird durch ein ADIF-Importschema ausgelöst, das halbautomatisch über einen Assistenten erstellt wird. Der Assistent generiert die Namen der temporären Tabellen und der Spalten dieser Tabellen aus den in diesem Attribut definierten Zeichenfolgen, gefolgt von den von ServiceNow importierten Spaltennamen. Es wird empfohlen, das XML-Attribut `Prefix` zu setzen, um sicherzustellen, dass der Spaltenname nicht mit einem reservierten SQL-Schlüsselwort identisch ist (Beispiel: `Order`).
- 8) Definieren Sie für jedes XML-Element `DataConnection` die Quelle, aus der die Daten importiert werden sollen. Fügen Sie dazu ein XML-Element `Entry` als untergeordnetes XML-Element des XML-Elements `DataConnection` hinzu. Ein XML-Element `DataConnection` kann mehrere untergeordnete XML-Elemente `Entry` enthalten. Jedes XML-Element `Entry` entspricht einem ADIF-Eintrag im ADIF-Importschema, das für die Integration der Daten in die Alfabet-Datenbank konfiguriert wurde. Definieren Sie die Datenquelle mit den folgenden XML-Attributen des XML-Elements `Entry`:
- **Name:** Geben Sie die ServiceNow-`sys_id` des Berichts oder den Namen der Datenbanktabelle/Datenbankansicht ein, die die Quelle des Datenimports in ServiceNow sind.
-
- Um die `sys_id` eines Berichts abzurufen, können Sie im Kontextmenü den Befehl **SYS_ID kopieren** verwenden, der für die Einträge der Listenansichtsdatensätze in ServiceNow verfügbar ist.
- **Id:** Geben Sie einen Namen ein, der in Alfabet in der Konfiguration des ADIF-Importschemas verwendet werden soll.
- 9) Fügen Sie dem Root-XML-Element `ServiceNowConfig` ein XML-Element `DataTypeMappings` hinzu. Fügen Sie das XML-Attribut `UnknownServiceNowTypeAsString` hinzu. Geben Sie `true` ein, um alle Datentypen, die in den untergeordneten Elementen des XML-Elements nicht ausdrücklich als Datentypen definiert sind, als Zeichenfolge zu importieren. Wenn Sie `false` eingeben und ein importierter Datentyp in der Spezifikation fehlt, schlägt der Import fehl.

- 10) Fügen Sie dem XML-Element `DataTypeMappings` für jeden Datentyp, der nicht als Zeichenfolge importiert werden soll, ein untergeordnetes XML-Element `DataType` hinzu, und definieren Sie die folgenden XML-Attribute für das XML-Element `DataType`:
 - `ServiceNowType`: Geben Sie den ServiceNow-Datentyp ein, der während des Imports in einen bestimmten Alfabet -Datentyp konvertiert werden soll.
 - `ADIFType`: Geben Sie den Alfabet -spezifischen Datentyp ein, der verwendet werden soll, um die Daten in die temporären Tabellen zu schreiben, die während des Imports über das ADIF-Importschema erzeugt wurden.
- 11) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Senden einer Anforderung über Proxyserver an ServiceNow

Optional können Sie die ServiceNow-Integrationschnittstelle dazu konfigurieren, Anforderungen an eine ServiceNow-Instanz über einen Proxyserver zu senden. Dies erfordert für die oben beschriebene Datenimport-Konfiguration die folgende zusätzliche Konfiguration im XML-Objekt **ServiceNowImportConfig**:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **ServiceNowImportConfig**, und wählen Sie **XML bearbeiten...** Das XML-Objekt mit der Datenimport-Definition für ServiceNow wird geöffnet.
- 3) Fügen Sie ein untergeordnetes XML-Element `Proxy` zu dem XML-Element `ServiceAccessInfo` des XML-Elements `DataTransferMappings` hinzu, für das Sie die Datenübertragung über einen Proxy definieren möchten.
- 4) Definieren Sie im XML-Element `Proxy` die folgenden XML-Attribute:
 - `url`: Definieren Sie die URL des Proxyservers.
 - `user`: Geben Sie bei Bedarf den Anwendernamen für den Zugriff auf den Proxyserver ein. Der Domänenname für die Authentifizierung wird separat über das XML-Attribut „Domäne“ definiert und muss nicht als Teil des Anwendernamens angegeben werden.
 - `password`: Geben Sie bei Bedarf das Passwort für den Zugriff auf den Proxyserver ein.
 - `domain`: Definieren Sie bei Bedarf den Domännennamen, der als Teil des Anwendernamens zur Authentifizierung am Proxyserver verwendet werden soll.
- 5) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Wenn Sie mehrere Datenverbindungen konfiguriert haben und jede Datenverbindung einen anderen Proxy erfordert, können Sie Ihrer Proxykonfiguration zusätzliche Proxys hinzufügen und in der Datenverbindungskonfiguration auf einen der Proxys verweisen. Die obige Proxydefinition wird als Standard verwendet, wenn einer Datenverbindung kein Proxy zugeordnet ist. Zur Verwendung eines zusätzlichen Proxys ist die folgende Konfiguration erforderlich:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.

- 2) Klicken Sie mit der rechten Maustaste auf **ServiceNowImportConfig**, und wählen Sie **XML bearbeiten...**. Das XML-Objekt mit der Datenimport-Definition für die Amazon Web Services wird geöffnet.
- 3) Fügen Sie dem XML-Element `Proxy` ein untergeordnetes XML-Element `AdditionalProxies` hinzu.
- 4) Fügen Sie dem XML-Element `AdditionalProxies` für jeden zusätzlichen Proxy, den Sie definieren möchten, ein untergeordnetes XML-Element `AdditionalProxy` hinzu, und definieren Sie die folgenden XML-Attribute für das XML-Element `AdditionalProxy`:
 - `Name`: Definieren Sie einen eindeutigen Namen für den zusätzlichen Proxy. Der Name wird verwendet, um in der Datenverbindungskonfiguration auf den Proxy zu verweisen.
 - `url`: Definieren Sie die URL des Proxyservers.
 - `user`: Geben Sie bei Bedarf den Anwendernamen für den Zugriff auf den Proxyserver ein. Der Domänenname für die Authentifizierung wird separat über das XML-Attribut „Domäne“ definiert und muss nicht als Teil des Anwendernamens angegeben werden.
 - `password`: Geben Sie bei Bedarf das Passwort für den Zugriff auf den Proxyserver ein.
 - `domain`: Definieren Sie bei Bedarf den Domännennamen, der als Teil des Anwendernamens zur Authentifizierung am Proxyserver verwendet werden soll.
- 5) Fügen Sie jedem XML-Element **DataConnection**, das einen der zusätzlichen Proxys verwenden soll, ein XML-Attribut `Proxy` hinzu. Der Wert des XML-Attributs `Proxy` muss mit dem Wert des XML-Attributs `Name` des XML-Elements **AdditionalProxy** übereinstimmen.
- 6) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Konfigurieren der Integration von ServiceNow-Daten in die Alfabet-Datenbank

Der Datenimport aus ServiceNow erfolgt durch Ausführen eines ADIF-Importschemas. In ADIF können Schemata mithilfe eines Assistenten konfiguriert werden. Wenn Sie den Assistenten öffnen, wird eine Verbindung zu der ServiceNow-Instanz hergestellt, für die Sie Daten definieren möchten. Sie können dann sehen, welche Daten zum Import in die Datenbanktabelle, die Datenbankansicht oder den Bericht verfügbar sind. Sie können die Daten im Assistenten auswählen. Beim Schließen des Assistenten wird das ADIF-Schema dann automatisch mit den erforderlichen Einträgen ausgefüllt, um die Daten in temporäre Datenbanktabellen einzulesen. Um die Integration der Daten in die Alfabet-Datenbank zu definieren, müssen die Einträge aus der halbautomatischen Erzeugung der ADIF-Importschema-Elemente mit Informationen dazu ergänzt werden, wie Daten aus den temporären Tabellen in die Alfabet-Datenbank integriert werden sollen. Welche Konfiguration für diesen letzten Schritt des Imports erforderlich ist, hängt stark von Ihren Anforderungen ab.



In diesem Abschnitt wird nur die Handhabung des ServiceNow-Assistenten beschrieben. Ausführliche Informationen über die Konfiguration eines ADIF-Importschemas, um Daten aus den temporären Datenbanktabellen in die Alfabet-Datenbank zu schreiben, finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*, sie werden hier nicht wiederholt.

Die Nutzung des ServiceNow-Assistenten ermöglicht nicht nur die halbautomatische Konfiguration des ADIF-Importschemas. Es werden auch die erforderlichen Mechanismen eingerichtet, um die Verbindung zu

ServiceNow zu öffnen und die Daten zu importieren, wenn der ADIF-Import auf der Basis des ADIF-Importschemas ausgeführt wird.

Innerhalb des ADIF-Importschemas entspricht ein XML-Element `DataConnection` im XML-Objekt `ServiceNowImportConfig` einem ADIF-Importeintrag, wenn der Import für Berichte im CSV-Format erfolgt, oder einem ADIF-Importsatz, wenn der Import aus Datenbanktabellen oder Datenbankansichten erfolgt und deshalb XML als Datenübertragungsformat verwendet wird. Jeder ADIF-Eintrag kann einzeln deaktiviert oder aktiviert werden. Deshalb können Sie den Datenimport über ADIF auf Teilmengen der im XML-Objekt `ServiceNowImportConfig` konfigurierten Daten beschränken, ohne die XML-Objektkonfiguration zu ändern.

Nach der Definition einer gültigen Konfiguration für den Datenimport im XML-Objekt `ServiceNowImportConfig` müssen Sie ein neues ADIF-Importschema definieren, um die Datenintegration mithilfe des ServiceNow-Importassistenten zu konfigurieren:

- 1) Klicken Sie in der Registerkarte **ADIF** in Alfabet Expand mit der rechten Maustaste auf den Root-Knoten **ADIF-Schemata** im Explorer, und wählen Sie **Importschema erzeugen** 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 Berichtsverzeichnisses in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.
 - **Assistent:** Wählen Sie `ServiceNowImport_Assistant` aus dem Dropdown-Listefeld aus.
 - **Import-Datei erforderlich:** Wählen Sie `False` aus.
 - **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 vom Typ **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.
- 3) Klicken Sie im Explorer mit der rechten Maustaste auf den Knoten des neuen ADIF-Importschemas, und wählen Sie **Erzeugen der ADIF-Schema-Details mithilfe von ServiceNow-Importassistent** aus. Der Assistent wird in Ihrem Standard-Webbrowser geöffnet.
 - 4) Im Feld **(1) Erzeugen Sie ADIF-Einträge für Datenverbindungseinträge** werden alle XML-Elemente `Entry` angezeigt, die für alle in Ihrem XML-Objekt `ServiceNowImportConfig` enthaltenen XML-Elemente `DataConnection` definiert wurden. Für jedes XML-Element `Entry` wird der Name des `Entry`, gefolgt vom Namen der `DataConnection` in Klammern, angezeigt. Klicken Sie auf jede Datenverbindung, für die mit diesem ADIF-Importschema Daten integriert werden sollen.
 - 5) Wenn Sie alle relevanten XML-Elemente `Entry` ausgewählt haben, klicken Sie auf die Schaltfläche **Einträge erzeugen** unter dem Feld, um die automatische Erzeugung der ADIF-Einträge/ADIF-Importsätze in Ihrem ADIF-Importschema auszulösen. Wenn der Vorgang abgeschlossen wurde, wird in der unteren linken Ecke des Assistenten-Fensters die Meldung **Erzeugen der Einträge abgeschlossen** angezeigt.
 - 6) Wählen Sie in der Dropdown-Liste des Felds **(2) Wählen Sie einen zu bearbeitenden Eintrag aus** eines der XML-Elemente `Entry` aus, für das Sie die ADIF-Konfiguration bearbeiten möchten. Eine Reihe von neuen Feldern wird angezeigt.
 - 7) Optional können Sie den Import so definieren, dass er nur Daten aus einer Teilmenge der Spalten in einer Datenbanktabelle, Datenbankansicht oder einem Bericht enthält. Standardmäßig ist im Feld **(3) Wählen Sie zu importierende ServiceNow-Spalten aus** die Option **Alle Spalten für Import auswählen** ausgewählt. Wenn Sie Daten aus einer einzelnen Spalte vom Import ausschließen möchten, können Sie stattdessen die Option **Spalten für Import manuell auswählen** verwenden.



Die Funktion zum Ausschließen von Spalten vom Import kann verwendet werden, wenn das Datenvolumen für den Import zu groß wird. Es besteht jedoch eine Einschränkung, die zu berücksichtigen ist: Beim manuellen Definieren der zu importierenden Spalten muss die an ServiceNow gesendete Anforderung zum Importieren der Daten alle definierten Spaltennamen beinhalten, anstatt alle Daten aus einer angegebenen Tabelle anzufordern. Je nach Länge des Spaltennamens besteht die Möglichkeit, dass die Größenbeschränkung für API-Aufrufe überschritten wird.

- 8) Das Feld **(4) Bearbeiten Sie den ausgewählten Eintrag** enthält eine Tabelle mit allen Daten, die importiert werden können. Für jede Spalte in einer Datenbanktabelle, Datenbankansicht oder einem Bericht wird in der Tabelle eine Zeile angezeigt. Der **Spaltenname** der ersten Spalte zeigt den Namen der Datenbankspalte, der Datenbankansichtspalte oder der Spaltenüberschrift im Bericht an. Gegebenenfalls können Sie den Standardimport von ServiceNow durch Bearbeitung der Tabellendaten ändern:
- **Einbeziehen:** Standardmäßig werden ADIF-Importeinträge/ADIF-Importsätze für alle Daten definiert, die in einer Datenbanktabelle, Datenbankansicht oder einem Bericht in ServiceNow verfügbar sind. Sie können Daten vom Import ausschließen, indem Sie das Kontrollkästchen in der entsprechenden Tabellenzeile deaktivieren. Der entsprechende ADIF-Importeintrag/ADIF-Importsatz wird aus dem ADIF-Importschema entfernt. Er wird außerdem aus dieser Tabelle entfernt, da er nicht die XML-Elemente `Entry` des XML-Objekts **ServiceNowImportConfig** auflistet, sondern die verfügbare Konfiguration des ADIF-Importeintrags im ADIF-Importschema. Diese Spalte kann nur dann bearbeitet werden, wenn das Feld **(3) Wählen Sie zu importierende ServiceNow-Spalten aus** auf **Spalten für Import manuell auswählen** gesetzt ist.
 - **Typ:** Überprüfen Sie, ob der Datentyp für die Integration der Daten in die Objektklassen-Datenbanktabellen der Alfabet-Datenbank korrekt ist. Er sollte mit dem Datentyp der Objektklasseneigenschaft identisch sein, in der die Daten gespeichert werden. Passen Sie die Definition bei Bedarf an.
 - **Größe:** Wenn im Feld **Typ** die Option `String` ausgewählt ist, überprüfen Sie, ob die definierte Zeichenfolgenlänge, die in dieser Spalte angezeigt wird, der Größendefinition der Ziel-Objektklasseneigenschaft in der Alfabet-Datenbank entspricht. Aus technischen Gründen muss der Wert **Größe** auf `<Größe in der Zieldatenbanktabelle>+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.
- 9) Optional können Sie den Import so konfigurieren, dass er auf Objekte begrenzt ist, die seit dem letzten Datenimport mit dem gleichen ADIF-Importschema geändert wurden. In der Dropdown-Liste des Feldes **(5) Wählen Sie Netchange ServiceNow-Spalten aus** werden alle Spalten für den Eintrag aufgeführt, die Daten zurückgeben. Wählen Sie eine oder mehrere Spalten aus, um den Import der Nettoänderungen auf der Basis dieser Daten auszuführen. Die Daten in den ausgewählten Spalten werden mit den Daten des letzten ADIF-Imports verglichen. Die Daten werden nur dann importiert, wenn eines der Daten nach dem Datum des letzten ADIF-Imports liegt. Der Import der Nettoänderungen wird auf Datenebene ausgeführt. Die Importzeit ist für den Vergleich nicht relevant.
-  Wenn Sie den Typ der Eigenschaft im Feld **(4) Bearbeiten Sie den ausgewählten Eintrag** auf Datum ändern, wird nur die Eigenschaft in der Dropdown-Liste des Feldes **(5) Wählen Sie Netchange ServiceNow-Spalten aus** angezeigt, nachdem Sie auf die Schaltfläche **Einträge aktualisieren** geklickt haben.
- 10) Klicken Sie auf die Schaltfläche **Einträge aktualisieren**, um Ihre Änderungen auf die ADIF-Importeinträge/ADIF-Importsätze im ADIF-Importschema anzuwenden. Wenn der Vorgang abgeschlossen wurde, wird eine Meldung angezeigt, dass die Aktualisierung abgeschlossen ist. Die Meldung wird in der unteren linken Ecke des Assistenten angezeigt.
- 11) Wiederholen Sie die Schritte 6) bis 11) für alle Datenverbindungseinträge.
- 12) Im Feld **(6) Auswählen, um die Datenverbindung zu aktivieren/deaktivieren** können Sie die Ausführung einzelner ADIF-Importeinträge/ADIF-Importsätze in Ihrem ADIF-Importschema optional deaktivieren. Standardmäßig sind alle ADIF-Elemente aktiviert und werden mit einem

Häkchen in der Liste angezeigt. Sie können die Aktivierung oder Deaktivierung ausführen, indem Sie in der Dropdown-Liste auf einen Namen klicken, um die Häkchen-Einstellung zu ändern.

- 13) Klicken Sie auf **OK**, um die Änderungen in das ADIF-Importschema zu schreiben.
- 14) Schließen Sie das Browserfenster, um den Assistenten zu beenden und zur Registerkarte „ADIF“ von Alfabet Expand zurückzukehren.
- 15) Klicken Sie mit der rechten Maustaste auf das ADIF-Importschema, und wählen Sie im Kontextmenü die Option **Struktur neu abfragen** aus. Sie sehen dann alle automatisch generierten ADIF-Elemente im ADIF-Schema. Jeder ADIF-Importeintrag verfügt über Datenimporte in temporären Tabellen, die über die Attributelemente im Ordner **Attribute** definiert werden. Der Import in die Alfabet-Datenbank -Standardtabellen ist nicht in der Konfiguration enthalten.



Der Assistent generiert die Namen der temporären Tabellen und der Spalten dieser temporären Tabellen aus der Zeichenfolge, die im Attribut `Prefix` des XML-Elements `Data-Connection` in Ihrem XML-Objekt `ServiceNowImportConfig` definiert wurde, gefolgt von den von ServiceNow importierten Spaltennamen. Die Namen werden nach 30 Zeichen abgeschnitten.

- 16) Konfigurieren Sie die Datenintegration in die Alfabet-Datenbank -Standardtabellen entsprechend Ihren Anforderungen. Weitere Informationen zu den über ADIF-Importschemata verfügbaren Konfigurationsoptionen finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

Ändern einer vorhandenen Konfiguration für die ServiceNow-Integration

Während der Ausführung des ADIF-Datenimports aus ServiceNow wird die Konfiguration im XML-Objekt `ServiceNowImportConfig` verwendet, um die Verbindung herzustellen und die zu importierenden Daten zu finden. Bei Änderungen im XML-Objekt, die sich darauf auswirken, wie Daten importiert werden, muss das ADIF-Schema entsprechend geändert werden. Beim Importieren von Ganzzahlwerten mit Anzeigewerten sind beispielsweise zwei Datenbankspalten in den während des ADIF-Imports erzeugten temporären Tabellen zum Speichern der Daten erforderlich. Dies wird beim Erstellen des ADIF-Schemas über den ServiceNow-Assistenten berücksichtigt. Wenn Sie die Importoptionen später ändern, sodass nur Ganzzahlwerte importiert werden, enthält die temporäre Datenbanktabelle immer noch die Spalten mit „_IV“ und „_DV“, die für den Import sowohl von Ganzzahl- als auch Anzeigewerten geändert wurden. Der Importmechanismus versucht, die Daten in eine Datenbankspalte ohne die Suffixe zu schreiben.

Es ist daher erforderlich, das ADIF-Schema an Änderungen der Konfiguration im XML-Objekt `ServiceNowImportConfig` anzupassen. Dies muss über den ServiceNow-Assistenten ausgeführt werden, um sicherzustellen, dass der Importmechanismus korrekt an die Änderungen angepasst wird.

Änderungen der ServiceNow-Daten werden im ServiceNow-Importassistenten angezeigt. Wenn Sie überprüfen möchten, ob Ihre Konfiguration immer noch den ServiceNow-Daten entspricht, wählen Sie in der Dropdown-Liste des Feldes **(2) Wählen Sie einen zu bearbeitenden Eintrag aus** das XML-Element `Entry`, und überprüfen Sie die Symbole, die im Feld **(3) Bearbeiten Sie den ausgewählten Eintrag** angezeigt werden.

- : Für diese Daten hat sich nichts geändert.
- : Die Daten sind in ServiceNow nicht mehr verfügbar.
- : Die Daten wurden in ServiceNow hinzugefügt.



Wenn Sie die Zuordnungstabelle anklicken, wird eine unverankerte Symbolleiste angezeigt, über die Sie auf eine Legende zur Erläuterung der Symbole zugreifen können.

Aktualisierungen können über das XML-Element `Entry` in einem XML-Element `DataConnection` im XML-Objekt `ServiceNowImportConfig` ausgeführt werden. Sie können eine der folgenden Aktionen ausführen:

- Um einen neuen ADIF-Importsatz/ADIF-Importeintrag für ein neues XML-Element `Entry` hinzuzufügen, ohne die übrige Konfiguration zu ändern, wählen Sie im Feld **(1) Erzeugen Sie ADIF-Einträge für Datenverbindungseinträge** nur den neuen `Entry` aus, klicken Sie auf die Schaltfläche **Einträge erzeugen** und dann auf **OK**, um Ihre Änderungen auf das ADIF-Schema anzuwenden. Der ADIF-Importsatz oder ADIF-Importeintrag wird erzeugt, ohne die vorhandene Importkonfiguration für andere XML-Elemente `Entry` zu ändern.
- Wenn Sie einen vorhandenen ADIF-Importsatz/ADIF-Importeintrag mit neuen Spalten für ein geändertes XML-Element `Entry` aktualisieren, den Rest der Konfiguration aber nicht verändern möchten, wählen Sie das XML-Element `Entry` im Dropdown-Listenfeld des Feldes **(2) Wählen Sie einen zu bearbeitenden Eintrag aus** aus, und bearbeiten Sie je nach Bedarf die Daten im Feld **(3) Bearbeiten Sie den ausgewählten Eintrag**. Klicken Sie danach auf die Schaltfläche **Einträge aktualisieren** und dann auf **OK**, um Ihre Änderungen auf das ADIF-Schema anzuwenden.

Beachten Sie bitte, dass der ADIF-Importsatz/ADIF-Importeintrag durch diese Aktion vollständig überschrieben wird. Jede manuell durchgeführte Konfiguration im ADIF-Schema wird entfernt.

- Um eine vorhandene ADIF-Eintragskonfiguration für ein vorhandenes und unverändertes XML-Element `Entry` zu bearbeiten, wählen Sie das XML-Element `Entry` im Dropdown-Listenfeld des Feldes **(2) Wählen Sie einen zu bearbeitenden Eintrag aus** aus und bearbeiten die Daten im Feld **(3) Bearbeiten Sie den ausgewählten Eintrag**. Klicken Sie danach auf die Schaltfläche **Einträge aktualisieren** und dann auf **OK**, um Ihre Änderungen auf das ADIF-Schema anzuwenden. Die Änderungen werden selektiv angewendet, ohne andere Konfigurationen zu ändern.
- Wenn Sie einen ADIF-Importsatz/ADIF-Importeintrag löschen müssen, weil das entsprechende XML-Element `Entry` entfernt wurde, müssen Sie den ADIF-Importsatz/ADIF-Importeintrag in Alfabet Expand auf der Registerkarte **ADIF** im Explorer **ADIF-Schemata** manuell entfernen. Diese Aktion wird vom ServiceNow-Assistenten nicht unterstützt.

Beachten Sie bitte, dass alle Änderungen erst angewendet werden, nachdem Sie im Assistenten auf die Schaltfläche **OK** geklickt haben. Die Änderungen sind im Explorer ADIF-Schemata in Alfabet Expand nicht automatisch sichtbar. Klicken Sie mit der rechten Maustaste auf das ADIF-Importschema, das Sie über den ServiceNow-Assistenten bearbeitet haben, und wählen Sie **Baum neu einlesen**, um die Änderungen im ADIF-Importschema sichtbar zu machen.

Konfigurieren der Integration von Alfabet -Daten in ServiceNow

Wenn Sie ServiceNow verwenden, können Sie Daten zu den in Alfabet geplanten und genehmigten Services aus der Alfabet-Datenbank in ServiceNow importieren, um die Daten zur Verwaltung der gehosteten Services weiterzuverwenden. Es gibt keine vordefinierte Methode zum Exportieren der Daten. Welche Daten aus Alfabet und ServiceNow einbezogen werden, wird vollständig vom Kunden konfiguriert.

Die Daten aus verschiedenen Tabellen aus der Alfabet-Datenbank werden zuerst in einem konfigurierten Bericht gesammelt, der einen einzigen Tabellendatensatz darstellt. Die Daten aus diesem konfigurierten Bericht müssen einer in ServiceNow definierten Importsatztafel entsprechen. Während des Exports

werden die Daten aus dem konfigurierten Bericht in Alfabet in die entsprechenden Spalten der Importsatz-tabelle in ServiceNow geschrieben. Danach werden sie durch einen vom Kunden konfigurierten Transformationsprozess in die ServiceNow-Datenbanktabellen importiert.

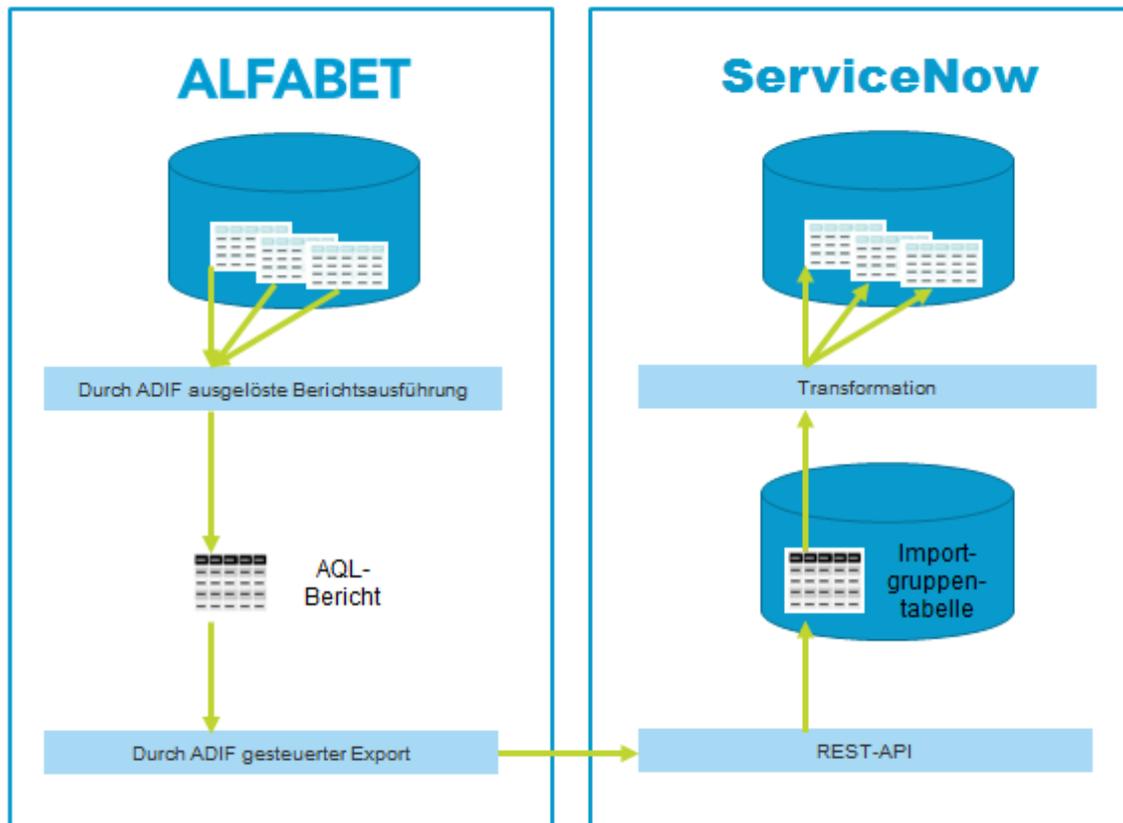


Abbildung: Übersicht über die Datensammlung von Alfabet in ServiceNow

Folgende Konfigurationsschritte sind erforderlich, um Daten von der Alfabet-Datenbank in ServiceNow zu exportieren:

- In Alfabet muss ein konfigurierter Bericht des **Typs** Query definiert werden, der die Daten aus den Datenbanktabellen der Alfabet-Datenbank in einem einfachen Datensatz zusammenfasst, um sie an ServiceNow zu exportieren. Für den Bericht können Filter definiert werden, in die beim Export Werte über die Befehlszeile des ADIF-Exportschemas eingefügt werden, das den Datenexport auslöst. Für das Attribut **Kategorie** des konfigurierten Berichts muss ein String festgelegt werden, der dann im XML-Objekt **ServiceNowExportConfig** für die Datenverbindung zum Exportieren der Daten aus dem Bericht definiert wird.
- Die Verbindung zu ServiceNow und die Zuordnung der Datentypen in Alfabet müssen im neuen XML-Objekt **ServiceNowExportConfig** definiert werden. Die Struktur und die Konfiguration des XML-Objekts **ServiceNowExportConfig** sind weitgehend mit dem XML-Objekt `ServiceNowImportConfig` identisch. Es bestehen lediglich geringfügige Unterschiede in der Konfiguration des XML-Elements **DataConnection**. Das XML-Attribut `FetchType` ist nicht verfügbar, da für den Import keine unterschiedlichen Ziele vorhanden sind. Vielmehr werden nur Daten aus Staging-Tabellen importiert. Das XML-Attribut `ReportCategory` steht zur Verfügung, für das ein String festgelegt werden muss, der der Einstellung des Attributs **Kategorie** des für den Datenimport verwendeten konfigurierten Berichts entspricht.
- In Alfabet muss ein ADIF-Exportschema erstellt werden, dessen Attribut „Assistant“ auf `ServiceNowExport_Assistant` gesetzt ist. Die gesamte Definition des Exportschemas wird vom

Assistenten durchgeführt. Eine manuelle Anpassung ist nicht nötig. Die Zuordnung von Daten des Ergebnisdatensatzes des konfigurierten Berichts zu den Spalten der Staging-Tabelle des ServiceNow-Imports wird im Assistenten vorgenommen. Ein Export der Nettoänderung kann ebenfalls konfiguriert werden.

- In ServiceNow muss ein Service mit der Importmethode HTTP und mit einer Importsatztabelle erstellt werden, in die während des Importvorgangs die Daten aus Alfabet eingefügt werden. Informationen zu diesem Konfigurationsschritt finden Sie in der Dokumentation von ServiceNow. Der Import aus der Importsatztabelle in die Tabellen von ServiceNow muss über eine Umwandlungszuordnung in ServiceNow konfiguriert werden. Beachten Sie, dass bei der Änderung von Datentabellen in ServiceNow Geschäftsregeln aufgerufen werden können, die zu Leistungsproblemen oder Fehlern bei Importvorgängen führen können. Die Menge der für eine Anforderung zu importierenden Daten sollte sorgfältig abgewogen werden, um diese Probleme zu vermeiden. Informationen zu diesem Konfigurationsschritt finden Sie in der Dokumentation von ServiceNow.
- In ServiceNow muss das Plug-In insertMultiple aktiviert sein, da hierdurch die SOAP-Methode aktiviert wird, die zum Import von Daten in den Service dient.

Um den Datenimport von Alfabet in ServiceNow auszuführen, müssen Sie den vom ADIF-Exportschema ausgelösten ADIF-Export starten. Beim ADIF-Export werden sowohl der Datenabruf aus dem konfigurierten Bericht als auch die Datenübertragung zum ServiceNow-Service ausgeführt. Der ADIF-Export sollte in regelmäßigen Abständen ausgeführt werden, um die Datenkonsistenz zwischen ServiceNow und Alfabet zu gewährleisten.



Informationen zur Ausführung von ADIF-Schemata finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

In den folgenden Abschnitten werden die Konfigurationsschritte für die Schnittstelle mit ServiceNow auf Alfabet -Seite in der erforderlichen Reihenfolge der Ausführung beschrieben:

- [Definieren von zu exportierenden Daten aus der Alfabet-Datenbank über einen konfigurierten Bericht](#)
- [Konfigurieren der Datenübertragung von Alfabet zu ServiceNow](#)
- [Senden einer Anforderung über Proxyserver an ServiceNow](#)
- [Konfigurieren des ADIF-Exportschemas für den Datenexport in ServiceNow](#)
- [Ändern einer vorhandenen Konfiguration für die ServiceNow-Integration](#)

Definieren von zu exportierenden Daten aus der Alfabet-Datenbank über einen konfigurierten Bericht

Alle Daten, die exportiert werden sollen, müssen in einem konfigurierten Bericht gesammelt werden, der auf einer Alfabet -Abfrage beruht und einen einfachen, tabellarischen Datensatz ausgibt.

Der konfigurierte Bericht wird im Konfigurationswerkzeug Alfabet Expand erzeugt:

- 1) Klicken Sie auf der Registerkarte **Berichte** von Alfabet Expand mit der rechten Maustaste auf den Ordner **Berichte** oder einen Unterordner, und wählen Sie die Option **Neuen Bericht erzeugen**. Der neue konfigurierte Bericht wird unter dem ausgewählten Berichtsordner angezeigt.

2) Definieren Sie im Eigenschaftsfenster die folgenden Attribute für den konfigurierten Bericht:

- **Name:** Optional können Sie hier den Namen des konfigurierten Berichts ändern. Der technische Name des konfigurierten Berichts muss eindeutig sein.



Der technische Name dient der Identifikation des konfigurierten Berichts im Kontext technischer Prozesse, wie z. B. das Zusammenführen der Berichtskonfiguration mit einer als XML-Datei gespeicherten Berichtskonfiguration. Daher müssen bei der Festlegung eines Namens für einen konfigurierten Bericht folgende Bedingungen erfüllt sein:

- Der Name muss eindeutig sein. Der Name muss unabhängig vom Berichtordner, in dem er sich befindet, eindeutig sein.
- Der Name darf nach Erstellung des konfigurierten Berichts nicht mehr geändert werden.
- Beim Festlegen eines technischen Namens wird die Syntax über einen Prüfmechanismus auf Korrektheit geprüft. Darüber hinaus ist zu beachten, dass der Name nicht mit einem Leerschritt beginnen und keine Sonderzeichen enthalten darf. Eine Liste der unzulässigen Sonderzeichen finden Sie im Abschnitt *Definieren von Attributen für Konfigurationsobjekte* im Kapitel *Erste Schritte mit Alfabet Expand*.
- Wenn Sie den Namen des konfigurierten Berichts ändern, wird er auch in anderen Konfigurationsobjekten, die während der Entwurfszeit auf das geänderte Objekt verweisen, korrekt aktualisiert.

Beachten Sie, dass der Name eines geänderten Konfigurationsobjekts nicht auf Navigationsseiten oder in Navigationsansichten aktualisiert wird. Wenn Sie planen, den Namen eines Konfigurationsobjekts anzupassen, sollten Sie erst den Verweis auf der Navigationsseite und dann den Namen des Konfigurationsobjekts ändern. Im Kontextmenü des konfigurierten Berichts können Sie die Funktionalität **Nutzung anzeigen** verwenden, um zu bewerten, ob der konfigurierte Bericht zu einer Navigationsansicht oder Navigationsseite hinzugefügt wurde.

- **Titel:** Geben Sie einen Titel für den konfigurierten Bericht ein. Der Titel, den Sie hier definieren, wird im Modul **Verwaltung** in der Funktionalität **Berichtverwaltung** sowie in den Ansichten **Konfigurierte Berichte** der Alfabet –Benutzeroberfläche angezeigt. Wenn der konfigurierte Bericht einer Objektansicht als Ansichtsseite zugeordnet ist, wird der Text in der Objektansicht als Titel der Ansichtsseite angezeigt. Der Titel des konfigurierten Berichts kann länger sein als die üblicherweise zulässigen 64 Zeichen.
- **Typ:** Wählen Sie *Query* aus. Für neue konfigurierte Berichte ist dies der Standardwert.
- **Kategorie:** Geben Sie einen eindeutigen String ein, mit dem der Bericht in der Datenverbindungsdefinition identifiziert wird, wenn die Datenübertragung an ServiceNow konfiguriert wird. Sie können mehrere Berichte definieren, die die gleiche Einstellung für die **Kategorie** haben. Wenn Sie ein ADIF-Exportschema für den Datenexport über die Datenverbindung mit Bezug auf die Berichtskategorie konfigurieren, können Sie den konfigurierten Bericht, der für den Export verwendet werden soll, aus allen konfigurierten Berichten mit der definierten Einstellung der **Kategorie** auswählen.



Die Zeichenfolgen für die **Kategorie** müssen sich vollständig voneinander unterscheiden, da im Dropdown-Listefeld für die Berichte im ADIF-Exportschema-

Assistenten aus technischen Gründen nicht nur genaue Übereinstimmungen der Zeichenfolge gefunden werden, die in der Konfiguration der Datenverbindung definiert wurde, sondern auch Berichte, deren Einstellung für die **Kategorie** die definierte Zeichenfolge enthält. Wenn die Datenverbindung beispielsweise so konfiguriert ist, dass sie mit konfigurierten Berichten der **Kategorie** `ExportApplications` arbeitet, werden konfigurierte Berichte mit einer Kategorie wie etwa `ExportApplicationGroups` ebenfalls gefunden.

- **Alfabet-Abfrage:** Klicken Sie auf die Schaltfläche **Durchsuchen** , um den **Alfabet Query Builder** zu öffnen und die Alfabet -Abfrage für den konfigurierten Bericht zu definieren. Die Abfrage muss einen einfachen tabellarischen Datensatz zurückgeben, der die Informationen enthält, die an den ServiceNow-Dienst übertragen werden sollen. Beachten Sie bezüglich der Konfiguration der Abfrage Folgendes:
 - Für den Bericht können Filter definiert werden. Die Filtereinstellungen werden beim Durchführen des ADIF-Exports dann über die Befehlszeileneinstellungen ausgeführt.
 - Wenn Sie einen Import von Nettoänderungen definieren möchten, sollte der Datensatz mindestens eine Spalte enthalten, die ein Datum zurückgibt, mit dem bei einem Datenexport von Nettoänderungen nur Daten exportiert werden können, deren Datum nach der Ausführung des letzten identischen Exports liegt.
 - Der Abfrage über **Anzeige-Eigenschaften** des Typs `RoleType` oder `Indicator` hinzugefügte Indikatoren und Rollen werden als Zeichenfolge verarbeitet. Einstellungen für **Kennzahlen als Symbol darstellen** werden ignoriert.



Informationen über das Arbeiten mit dem **Alfabet Query Builder** finden Sie im Kapitel *Definieren von Abfragen*.

Eine Beschreibung zur Definition von Berichtsfiltern in einer Alfabet -Abfrage finden Sie unter *Definieren von Filtern für einen auf Alfabet -Abfragen basierten konfigurierten Bericht* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

- **Auf Klasse anwenden:** Definieren Sie optional eine Objektklasse. Wenn das Attribut **Auf Klasse anwenden** festgelegt ist, muss die Alfabet -Abfrage für den konfigurierten Bericht eine `WHERE`-Klausel beinhalten, die den Alfabet -Parameter:`BASE` verwendet, um das aktuelle Objekt zu referenzieren. Der Parameter:`BASE` muss dann über die Befehlszeile des ADIF-Jobs, der den konfigurierten Bericht ausführt, mit dem REFSTR eines Objekts der ausgewählten Klasse ausgefüllt werden.
- 3) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste auf die Schaltfläche **Speichern** .
 - 4) Klicken Sie im Explorer mit der rechten Maustaste auf den konfigurierten Bericht, und wählen Sie **Bericht prüfen** aus. Der konfigurierte Bericht wird in einem Webbrowser geöffnet. Wenn die Ergebnisse nicht Ihren Erwartungen entsprechen, können Sie die Alfabet -Abfrage so lange bearbeiten, bis sie Ihren Erwartungen entspricht.
 - 5) Klicken Sie im Explorer mit der rechten Maustaste auf den konfigurierten Bericht, und wählen Sie im Kontextmenü **Status in „Active“ ändern** aus.

Konfigurieren der Datenübertragung von Alfabet zu ServiceNow

Die Konfiguration des Datenexports erfolgt im XML-Objekt **ServiceNowExportConfig**. In dem XML-Objekt können Verbindungen mit mehreren ServiceNow-Quellen definiert werden. Die allgemeine Struktur der XML im XML-Objekt sieht wie folgt aus:

```
<ServiceNowConfig>
  <DataTransferMappings>
    <DataTransferMapping>
      <ServiceAccessInfo>...</ServiceAccessInfo>
      <DataConnectivityInfo>...</DataConnectivityInfo>
    </DataTransferMapping>
  </DataTransferMappings>
  <DataTypeMappings>...</DataTypeMappings>
```

Das Root-XML-Element `ServiceNowConfig` der XML-Objekte enthält zwei untergeordnete Elemente:

- **DataTransferMappings:** Dieses XML-Element kann ein oder mehrere untergeordnete Elemente `DataTransferMapping` enthalten, die jeweils die Definition einer Verbindung zu einem anderen ServiceNow-Server enthalten. Die Definition enthält in einem untergeordneten Element `ServiceAccessInfo` die Spezifikation der Verbindung zum REST-API des Webdienstes des ServiceNow-Servers und in einem untergeordneten Element `DataConnectivityInfo` die Spezifikation der zu übertragenden Daten. Innerhalb des Elements `DataConnectivityInfo` können unterschiedliche Datenübertragungsbedingungen für unterschiedliche Datentypen definiert werden.
- **DataTypeMappings:** Das XML-Element enthält die Zuordnung der exportierten Datentypen zu den in ServiceNow gültigen Datentypen. Die Daten werden während des Datenimports in die angegebenen Datentypen konvertiert. Die Datentypdefinitionen gelten für alle Datenverbindungen in allen Datenübertragungszuordnungen.

So definieren Sie den Datenimport von einem ServiceNow-Server:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **ServiceNowExportConfig**, und wählen Sie **XML bearbeiten....** Das XML-Objekt wird geöffnet.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist eine Vorlage über das XML-Attribut `XML Template` des XML-Objekts verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf. Bei einer Beispielformatierung müssen Sie XML-Elemente bearbeiten, anstatt neue hinzuzufügen.

- 3) Fügen Sie dem Root-XML-Element `ServiceNowConfig` ein untergeordnetes XML-Element `DataTransferMappings` hinzu.
- 4) Fügen Sie dem XML-Element `DataTransferMappings` ein untergeordnetes XML-Element `DataTransferMapping` hinzu, und definieren Sie das folgende XML-Attribut für das XML-Element `DataTransferMapping`:

- **Name:** Der Name der Verbindung zum ServiceNow-Server, mit dem die Verbindung in der Konfiguration des ADIF-Schemas identifiziert wird.
 - **IsActive:** Auf `true` setzen, wenn Daten in die ServiceNow-Instanz exportiert werden sollen. Auf `false` setzen, wenn Daten derzeit nicht in die ServiceNow-Instanz exportiert werden sollen.
- 5) Fügen Sie dem XML-Element `DataTransferMapping` ein untergeordnetes XML-Element `ServiceAccessInfo` hinzu, und definieren Sie die Informationen, die zum Einrichten einer Verbindung mit dem ServiceNow-Server erforderlich sind. Das XML-Element `ServiceAccessInfo` hat keine XML-Attribute. Die Informationen, die zum Herstellen einer Verbindung mit dem REST-API der Webdienste des ServiceNow-Servers erforderlich sind, müssen als Zeichenfolgen den folgenden untergeordneten XML-Elementen hinzugefügt werden, die dem XML-Element `ServiceAccessInfo` hinzugefügt werden müssen:
- **service:** Die URL der ServiceNow-Instanz.
 - **apipath:** Der Pfad zum REST-API in der ServiceNow-Instanz. Geben Sie `/api/now/v1/` ein.
 - **oauthpath:** OAuth ist für die Authentifizierung erforderlich. Geben Sie den Pfad zu dem OAuth-Endpunkt mit dem XML-Element `oauthpath` an.
 - **username:** Der Anwendername für den Zugriff auf ServiceNow.



Der Anwender muss die erforderlichen Zugriffsberechtigungen zur Durchführung folgender REST API-Aufrufe an ServiceNow haben:

- `https://<BaseAddress>/oauth_token.do`
- `https://<BaseAddress>/<databaseviewname>.do?WSDL`
- `https://<BaseAddress>/api/now/v1/table/sys_dictionary?sysparm_query=GOTOname=<tableName>`
- `https://<BaseAddress>/api/now/v1/table/sys_transform_map?source_table=<tableName>`
- `https://<BaseAddress>/api/now/v1/table/sys_transform_entry?map=<map id>`
- `https://<BaseAddress>/api/now/v1/table/sys_db_object?name=<tableName>`

Mit folgenden Berechtigungen können Sie alle erforderlichen Serviceaufrufe durchführen:

- `import_admin` zur Durchführung der erforderlichen Änderungen an den Importsatztabellen.
- `soap` zur Durchführung von SOAP-Vorgängen
- `personalize_dictionary` für den Zugriff auf die Tabellen `sys_dictionary` und `sys_db_object`, in denen Sie bei einem Import in ServiceNow zusätzliche Informationen zu den einzelnen Tabellen und die Definitionen für alle Spalten der einzelnen Tabellen finden.

- `Rest_service` für das Recht, über REST API-Aufrufe auf die erforderlichen Datenbanktabellen zuzugreifen.
- `itil`, wenn CMDB CI-Tabellen bei einem Import in ServiceNow wiederverwendet werden.
- `password`: Das Passwort für den Zugriff auf ServiceNow.
- `client_id`: Geben Sie die Client-ID für den OAuth-Endpunkt bei ServiceNow ein.
- `client_secret`: Geben Sie den Clientschlüssel für den OAuth-Endpunkt bei ServiceNow ein.



In den XML-Elementen zur Definition der ServiceNow-Serververbindung können Servervariablen verwendet werden, um den Wert des XML-Elements zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation auszulesen, wenn eine Verbindung zu ServiceNow hergestellt wurde. Weitere Informationen zu Servervariablen finden Sie unter [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).

- 6) Fügen Sie dem XML-Element `DataTransferMapping` ein untergeordnetes XML-Element `DataConnectivityInfo` hinzu, und definieren Sie die Informationen, die für die Datenübertragung zum ServiceNow-Server erforderlich sind. Das XML-Element `DataConnectivityInfo`. Definieren Sie die Standard-Datenübertragungsparameter für Datenexporte, indem Sie die folgenden XML-Attribute für das XML-Element `DataConnectivityInfo` festlegen:
- `DefaultPageSize`: Sie können einen Grenzwert für die Daten definieren, die gleichzeitig auf der Datenverbindung übertragen werden sollen. Geben Sie die Anzahl der Datensätze ein, die gleichzeitig übertragen werden sollen. Dieser Wert gilt für alle Datenverbindungen, bei denen kein Attribut `PageSize` definiert wurde. Wenn dies weder für die Datenverbindung noch über dieses XML-Attribut definiert wurde, wird ein Standardwert von 100 festgelegt.
 - `DefaultTimeout`: Definieren Sie das Standard-Zeitlimit für HTTP-Anforderungen, das für alle Datenverbindungen verwendet werden soll, für die kein Zeitlimit über das XML-Attribut `Timeout` des XML-Elements `DataConnection` definiert wurde. Wenn weder für die Datenverbindung noch über dieses XML-Attribut ein Zeitlimit definiert wurde, wird der für die Alfabet-Webapplikation definierte Standardwert festgelegt.
 - `DefaultMaxThreadCount`: Definieren Sie die maximale Anzahl paralleler Threads für die Datenübertragung an ServiceNow auf einer Datenverbindung. Der hier definierte Wert wird als Standardwert für alle Datenverbindungen verwendet, für die keine maximale Anzahl von zu übertragenden Datensätzen über das XML-Attribut `MaxRecordCount` des XML-Elements `DataConnection` definiert wurde.
- 7) Fügen Sie dem XML-Element `DataConnectivityInfo` ein oder mehrere untergeordnete XML-Elemente `DataConnection` hinzu. Jedes XML-Element `DataConnection` bündelt Daten, die über eine Übertragung zum ServiceNow-Server exportiert werden sollen. Jedes XML-Element `DataConnection` entspricht einem Service von ServiceNow für den Datenimport. Das bedeutet, dass alle zu exportierenden Informationen für ein XML-Element `DataConnection` das gleiche Exportformat und die gleiche Datenstruktur haben müssen. Definieren Sie die Datenübertragungsparameter für die Datenanforderung, indem Sie die folgenden XML-Attribute für das XML-Element `DataConnection` festlegen:
- `DataConnectionName`: Geben Sie einen eindeutigen Namen für die Datenverbindung ein. Dieser Name dient dazu, die Datenverbindung in der Konfiguration des ADIF-Schemas zu identifizieren.

- **ReportCategory:** Geben Sie die Zeichenfolge ein, die im Attribut **Kategorie** des konfigurierten Berichts definiert ist, der für den Datenexport über diese Verbindung verwendet wird.
- **Timeout:** Definieren Sie das Zeitlimit für HTTP-Anforderungen für die Datenverbindung. Wenn kein Zeitlimit definiert wird, wird das im XML-Element `DataConnectivityInfo` definierte Standard-Zeitlimit verwendet.
- **PageSize:** Geben Sie die maximale Anzahl von Datensätzen an, die gleichzeitig auf der Datenverbindung übertragen werden sollen.



Es wird empfohlen, das XML-Attribut `PageSize` auf einen Wert unter 500 zu setzen. Eine zu hohe Anzahl von Datensätzen kann zu Übertragungsproblemen führen.

- **MaxthreadCount:** Definieren Sie die maximale Anzahl paralleler Threads für die Datenübertragung an ServiceNow.
- 8) Definieren Sie für jedes XML-Element `DataConnection` die Quelle, aus der die Daten exportiert werden sollen. Fügen Sie dazu ein XML-Element `Entry` als untergeordnetes XML-Element des XML-Elements `DataConnection` hinzu. Ein XML-Element `DataConnection` kann mehrere untergeordnete XML-Elemente `DataConnectionEntry` enthalten. Beachten Sie bitte, dass die über die definierten Quellen exportierte Datenstruktur innerhalb eines XML-Elements `DataConnection` konsistent sein muss. Definieren Sie die Datenquelle mit den folgenden XML-Attributen des XML-Elements `Entry`:
 - **Name:** Geben Sie einen eindeutigen Namen ein, um den Eintrag in der Konfiguration des ADIF-Exportschemas zu identifizieren.
 - **Id:** Geben Sie den Bezeichner des ServiceNow-Services ein, der das Ziel des Exports ist.
 - 9) Fügen Sie dem Root-XML-Element `ServiceNowConfig` ein XML-Element `DataTypeMappings` hinzu. Fügen Sie das XML-Attribut `UnknownServiceNowTypeAsString` hinzu. Geben Sie `true` ein, um alle Datentypen, die in den untergeordneten Elementen des XML-Elements nicht ausdrücklich als Datentypen definiert sind, als Zeichenfolge zu exportieren. Wenn Sie `false` eingeben und ein exportierter Datentyp in der Spezifikation fehlt, schlägt der Export fehl.
 - 10) Fügen Sie dem XML-Element `DataTypeMappings` für jeden Datentyp, der nicht als Zeichenfolge exportiert werden soll, ein untergeordnetes XML-Element `DataType` hinzu, und definieren Sie die folgenden XML-Attribute für das XML-Element `DataType`:
 - **ServiceNowType:** Geben Sie den ServiceNow-Datentyp ein, der an ServiceNow übertragen wird.
 - **ADIFType:** Geben Sie den Alfabet -spezifischen Datentyp ein, der in den ServiceNow-Datentyp umgewandelt wird, der mit dem XML-Attribut `ServiceNowType` definiert wurde.
 - 11) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Senden einer Anforderung über Proxyserver an ServiceNow

Optional können Sie die ServiceNow-Integrationsschnittstelle dazu konfigurieren, Anforderungen an eine ServiceNow-Instanz über einen Proxyserver zu senden. Dies erfordert für die oben beschriebene Datenexport-Konfiguration die folgende zusätzliche Konfiguration im XML-Objekt **ServiceNowExportConfig**:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte** > **IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **ServiceNowExportConfig**, und wählen Sie **XML bearbeiten...**. Das XML-Objekt mit der Datenimport-Definition für ServiceNow wird geöffnet.
- 3) Fügen Sie ein untergeordnetes XML-Element `Proxy` zu dem XML-Element `ServiceAccessInfo` des XML-Elements `DataTransferMappings` hinzu, für das Sie die Datenübertragung über einen Proxy definieren möchten.
- 4) Definieren Sie im XML-Element `Proxy` die folgenden XML-Attribute:
 - `url`: Definieren Sie die URL des Proxyservers.
 - `user`: Geben Sie bei Bedarf den Anwendernamen für den Zugriff auf den Proxyserver ein. Der Domänenname für die Authentifizierung wird separat über das XML-Attribut „Domäne“ definiert und muss nicht als Teil des Anwendernamens angegeben werden.
 - `password`: Geben Sie bei Bedarf das Passwort für den Zugriff auf den Proxyserver ein.
 - `domain`: Definieren Sie bei Bedarf den Domännennamen, der als Teil des Anwendernamens zur Authentifizierung am Proxyserver verwendet werden soll.
- 5) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Wenn Sie mehrere Datenverbindungen konfiguriert haben und jede Datenverbindung einen anderen Proxy verwenden soll, können Sie Ihrer Proxykonfiguration zusätzliche Proxys hinzufügen und in der Datenverbindungskonfiguration auf einen der Proxys verweisen. Die obige Proxydefinition wird als Standard verwendet, wenn einer Datenverbindung kein Proxy zugeordnet ist. Zur Verwendung eines zusätzlichen Proxys ist die folgende Konfiguration erforderlich:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte** > **IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **ServiceNowExportConfig**, und wählen Sie **XML bearbeiten...**. Das XML-Objekt mit der Datenimport-Definition für die Amazon Web Services wird geöffnet.
- 3) Fügen Sie dem XML-Element `Proxy` ein untergeordnetes XML-Element `AdditionalProxies` hinzu.
- 4) Fügen Sie dem XML-Element `AdditionalProxies` für jeden zusätzlichen Proxy, den Sie definieren möchten, ein untergeordnetes XML-Element `AdditionalProxy` hinzu, und definieren Sie die folgenden XML-Attribute für das XML-Element `AdditionalProxy`:
 - `Name`: Definieren Sie einen eindeutigen Namen für den zusätzlichen Proxy. Der Name wird verwendet, um in der Datenverbindungskonfiguration auf den Proxy zu verweisen.
 - `url`: Definieren Sie die URL des Proxyservers.

- **user:** Geben Sie bei Bedarf den Anwendernamen für den Zugriff auf den Proxyserver ein. Der Domänenname für die Authentifizierung wird separat über das XML-Attribut „Domäne“ definiert und muss nicht als Teil des Anwendernamens angegeben werden.
 - **password:** Geben Sie bei Bedarf das Passwort für den Zugriff auf den Proxyserver ein.
 - **domain:** Definieren Sie bei Bedarf den Domännennamen, der als Teil des Anwendernamens zur Authentifizierung am Proxyserver verwendet werden soll.
- 5) Fügen Sie jedem XML-Element **DataConnection**, das einen der zusätzlichen Proxys verwenden soll, ein XML-Attribut `Proxy` hinzu. Der Wert des XML-Attributs `Proxy` muss mit dem Wert des XML-Attributs `Name` des XML-Elements **AdditionalProxy** übereinstimmen.
 - 6) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Konfigurieren des ADIF-Exportschemas für den Datenexport in ServiceNow

Der Datenexport in ServiceNow erfolgt durch Ausführen eines ADIF-Exportschemas. In ADIF können Exportschemata mithilfe eines Assistenten konfiguriert werden. Wenn Sie den Assistenten öffnen, wird eine Verbindung zu der ServiceNow-Instanz hergestellt, in die Sie Daten exportieren möchten. Sie können die Datenzuordnung direkt ausführen. Sie können die Daten im Assistenten auswählen. Beim Schließen des Assistenten wird das ADIF-Schema dann automatisch mit den erforderlichen Einträgen ausgefüllt, um die Daten zu exportieren.

Die Nutzung des ServiceNow-Assistenten ermöglicht nicht nur die halbautomatische Konfiguration des ADIF-Exportschemas. Es werden auch die erforderlichen Mechanismen eingerichtet, um die Verbindung zu ServiceNow zu öffnen, den konfigurierten Alfabet -Bericht auszuführen und die Daten zu exportieren, wenn der ADIF-Export auf der Basis des ADIF-Exportschemas ausgeführt wird.

Innerhalb des ADIF-Exportschemas entspricht ein XML-Element **Entry** in einem XML-Element `Data-Connection` des XML-Objekts `ServiceNowImportConfig` einem ADIF-Exporteintrag. Jeder ADIF-Eintrag kann einzeln deaktiviert oder aktiviert werden. Deshalb können Sie den Datenexport über ADIF auf Teilmengen der im XML-Objekt `ServiceNowImportExportConfig` konfigurierten Daten beschränken, ohne die XML-Objektkonfiguration zu ändern.

Nach der Definition einer gültigen Konfiguration für den Datenexport im XML-Objekt `ServiceNowExport-Config` müssen Sie ein neues ADIF-Exportschema definieren, um die Datenintegration mithilfe des ServiceNow-Importassistenten zu konfigurieren:

- 1) Klicken Sie in der Registerkarte **ADIF** in Alfabet Expand mit der rechten Maustaste auf den Root-Knoten **ADIF-Schemata** im Explorer, und wählen Sie **Exportschema erzeugen** aus. Das neue ADIF-Exportschema wird dem Explorer hinzugefügt. Das Attributfenster des neuen ADIF-Exportschemas wird rechts 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-Exportschemas in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.
 - **Assistent:** Wählen Sie `ServiceNowExport_Assistant` aus dem Dropdown-Listenfeld aus.

- **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.
- 3) Klicken Sie im Explorer mit der rechten Maustaste auf den Knoten des neuen ADIF-Exportschemas, und wählen Sie **Erzeugen der ADIF-Schema-Details mithilfe von ServiceNow-Importassistent** aus.
 - 4) Bestätigen Sie die Warnmeldung. Der Assistent wird in Ihrem Standard-Webbrowser geöffnet.
 - 5) Wählen Sie im Feld **(1) Wählen Sie eine Datenverbindung aus** eine im XML-Objekt **ServiceNowExportConfig** konfigurierte Datenverbindung aus dem Dropdown-Listefeld aus. Die Eintrags Elemente für das Element „Datenverbindung“ werden im Feld **(2) Datenverbindungseinträge** aufgelistet. Nach jedem Element „Eintrag“ wird die Information (Eintrag nicht erstellt) angezeigt, da die Einträge noch nicht im ADIF-Exportfirma enthalten sind.
 - 6) Wählen Sie im Feld **(3) Wählen Sie einen zu bearbeitenden/aktualisierenden Eintrag aus** einen der verfügbaren Einträge aus, um einen ADIF-Exporteintrag dafür zu erstellen.
 - 7) Wählen Sie im Feld **(4) Ordnen Sie den Alfabet-Bericht dem ausgewählten Eintrag zu** einen der konfigurierten Berichte aus, die der Datenverbindung über ihre Einstellung **Kategorie** zugeordnet sind, um ihn zum Export über diesen **Entry** auszuführen.
 - 8) Optional können Sie im Feld **(5) Wählen Sie eine Anzeigeeoption für Alfabet-Berichtseigenschaften aus** Festlegungen vornehmen. Wählen Sie eine der folgenden Optionen aus:
 - **Berichtsspalten wie im Bericht definiert sortieren:** Die Spaltennamen im konfigurierten Bericht werden anstelle der Namen der Objekteigenschaftsklassen angezeigt. Die Sortierreihenfolge ist identisch mit der Reihenfolge der Spalten im konfigurierten Bericht.

- **Berichtsspalten mit lexikografischer Sortierung verwenden:** Die Spaltennamen im konfigurierten Bericht werden anstelle der Namen der Objekteigenschaftsklassen angezeigt. Die Sortierreihenfolge ist alphanumerisch.
 - **Eigenschaftsspalten wie im Bericht definiert sortieren:** Die Objekteigenschaftsklassen werden als <ObjectClassName>.<PropertyName> angezeigt. Die Sortierreihenfolge ist identisch mit der Reihenfolge der Spalten im konfigurierten Bericht.
 - **Eigenschaftsspalten mit lexikographischer Sortierung sortieren:** Die Objekteigenschaftsklassen werden als <ObjectClassName>.<PropertyName> angezeigt. Die Sortierreihenfolge ist alphanumerisch.
- 9) Optional können Sie den Export so konfigurieren, dass er auf Objekte begrenzt ist, die seit dem letzten Datenexport mit dem gleichen ADIF-Exportschema und den gleichen Befehlszeilenparametern geändert wurden. In der Dropdown-Liste des Felds **(6) Wählen Sie Netchange-Berichtsspalten aus** werden alle Spalten des ausgewählten Berichts aufgeführt, die Daten zurückgeben. Wählen Sie eine oder mehrere Spalten aus, um den Export von Nettoänderungen auf der Basis dieser Daten auszuführen. Die Daten in der ausgewählten Spalte werden mit den Daten der letzten ADIF-Exportausführung verglichen. Die Daten werden nur dann exportiert, wenn das Datum nach dem Datum des letzten ADIF-Exports liegt. Der Export von Nettoänderungen wird auf Datenebene ausgeführt. Die Exportzeit ist für den Vergleich nicht relevant.
- 10) Klicken Sie auf die Schaltfläche **Attribute erzeugen**. Wenn der Vorgang abgeschlossen wurde, wird in der unteren linken Ecke des Assistenten-Fensters die Meldung angezeigt, dass die Attributerzeugung abgeschlossen ist, und im Feld **(7) Ordnen Sie Alfabet-Klasseneigenschaften ServiceNow Web Service-Feldern zu** wird eine Tabelle angezeigt.
- 11) In der Tabelle des Feldes **(7) Ordnen Sie Alfabet-Klasseneigenschaften ServiceNow Web Service-Feldern zu** werden alle Felder aus der ServiceNow-Importsatztabelle in der Spalte ServiceNow Web Service-Feld aufgelistet. Wählen Sie für jede Zeile die entsprechende Spalte des konfigurierten Berichts von Alfabet aus der Dropdownliste aus. In den zusätzlichen Spalten der Tabelle finden Sie Informationen zu den Zieldaten in der ServiceNow-Standardtabelle, in die die Daten importiert werden. Sie erhalten den ServiceNow-Eigenschaftsnamen und -typ sowie Informationen zu wichtigen Merkmalen:
- *C = Die Eigenschaft ist der eindeutige Schlüssel, mit dem die Daten während des Imports den ServiceNow-Daten zugeordnet werden.
 - *R = Die Eigenschaft wird in ServiceNow als Anzeigewert festgelegt. Dies bedeutet, dass der Wert bei der Referenzierung des Datensatzes in der Benutzeroberfläche von ServiceNow angezeigt wird.
- 12) Klicken Sie auf **Eintrag erzeugen**.
- 13) Wiederholen Sie die Schritte 6) bis 11) für alle Datenverbindungseinträge. Wenn Sie einen ADIF-Exporteintrag für das gleiche XML-Element `Entry`, aber mit einem anderen konfigurierten Bericht erstellen möchten, klicken Sie auf **Bericht ändern**, um die Tabelle aus dem Assistenten zu entfernen, und fahren Sie mit Schritt 7) fort.
- 14) Klicken Sie auf **OK**, um die Änderungen in das ADIF-Importschema zu schreiben.
- 15) Schließen Sie das Browserfenster, um den Assistenten zu beenden und zu Alfabet Expand zurückzukehren.
- 16) Wählen Sie im Menü von Alfabet Expand die Option **Metamodell > Metamodell neu lesen** aus. Das ADIF-Schema wird mit den Einstellungen aktualisiert, die im Assistenten vorgenommen wurden.

Die ADIF-Exporteinträge, die auf Basis der Einstellungen erstellt wurden, werden über die Attributelemente im Ordner **Attribute** definiert.

Ändern einer vorhandenen Konfiguration für die ServiceNow-Integration

Sie können eine vorhandene Konfiguration ändern, indem Sie beispielsweise den Exportbericht ändern, um verschiedene Daten oder neue, seitens ServiceNow hinzugefügte Eigenschaften zu integrieren. Bei Änderungen der Konfiguration muss immer auch über den **ServiceNow-Exportassistenten** das ADIF-Exportschema geändert werden.

Wenn Sie den **ServiceNow-Exportassistenten** eines bereits vorhandenen und für den Datenexport in ADIF konfigurierten ADIF-Schemas starten und einen geänderten Datenverbindungseintrag zur Aktualisierung auswählen, vergleicht der Exportassistent die aktuelle Konfiguration in Alfabet und ServiceNow mit der vorhandenen Konfiguration im ADIF-Schema. Im Feld für die Zuordnung der Eigenschaften des **ServiceNow-Exportassistenten** werden die Änderungen mit Symbolen markiert, damit Sie Ihre Änderungen sofort finden können. Den Symbolen können Sie folgende Informationen entnehmen:

-  : Die Konfiguration wurde für diese Zuordnung nicht geändert.
-  : Die Eigenschaft wurde hinzugefügt.
-  : Die Eigenschaft wurde entfernt.
-  : Die Eigenschaft wurde in ServiceNow geändert.
-  : Die Eigenschaft wurde in Alfabet geändert.



Wenn Sie die Zuordnungstabelle anklicken, wird eine unverankerte Symbolleiste angezeigt, über die Sie auf eine Legende zur Erläuterung der Symbole zugreifen können.

Sie können eine der folgenden Aktionen ausführen:

- Um Änderungen in der Eigenschaftszuordnung eines vorhandenen Eintrags anzuwenden, ändern Sie die Einstellungen im Fenster **(7) Ordnen Sie Alfabet-Klasseneigenschaften ServiceNow Web Service-Feldern zu**, und klicken Sie auf **Eintrag aktualisieren**.
- Wenn der Import auf einem vollständig neuen Bericht beruhen soll, klicken Sie auf **Bericht ändern**. Das Feld **(4) Ordnen Sie den Alfabet-Bericht dem ausgewählten Eintrag zu** wird aktiviert, und der Inhalt der Zuordnungstabelle wird nicht mehr angezeigt. Klicken Sie auf **Attribute erstellen**, um die Zuordnungstabelle mit den relevanten Informationen aus dem neuen Bericht zu füllen, ordnen Sie die Eigenschaften im Fenster **(7) Ordnen Sie Alfabet-Klasseneigenschaften ServiceNow Web Service-Feldern zu** zu, und klicken Sie auf **Eintrag erzeugen**.

Kapitel 12: Konfigurieren der Integration mit JIRA

Alfabet unterstützt den Datenimport von JIRA® in Alfabet und den Export von Alfabet -Daten in JIRA. Mit der Funktion, Jira-Daten in Alfabet zu importieren, kann das Unternehmen Sachverhalte, Projekte, Kategorien und Boards, die in Jira dokumentiert sind, in festgelegte Entitäten in Alfabet importieren, etwa Applikationen, Installationen, Standardplattformen, Projekte usw. Dadurch wird eine Brücke zwischen agilen Entscheidungen auf Unternehmensebene und den betrieblichen Entscheidungen von SCRUM-Teams, Projektteams usw. geschlagen.

Die Funktion zum Export von Alfabet -Daten in Jira unterstützt die Planung und Priorisierung der operativen Entwicklungsarbeit und bietet Unterstützung für eine bimodale IT und agile Projekte im Kontext der Portfoliobewertung. Das Ziel des Jira-Exports ist es, in Jira neue Sachverhalte auf der Grundlage von Sachverhalten zu erzeugen, die über Anforderungen, Funktionen, Tickets, Steuerungselemente usw. in Alfabet erfasst wurden.



Details zu den Einheiten in JIRA, z. B. Projekten, Sachverhalten, Boards und Sprints, sowie allgemeine Informationen zum Produkt JIRA finden Sie in der im Lieferumfang des Produkts enthaltenen Dokumentation.

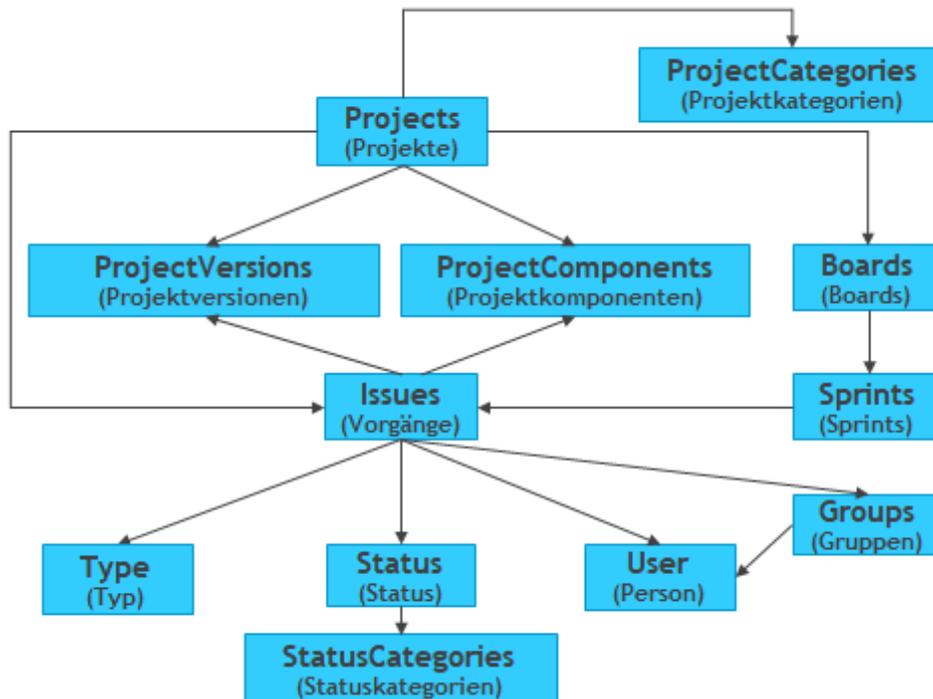
Welche JIRA-Inhalte integriert werden können, hängt von der von Ihrem Unternehmen verwendeten JIRA-Version ab.

- Für JIRA-Versionen vor JIRA 5 ist keine Integration verfügbar, da JIRA 5 als erste Version das REST-API zur Einrichtung der Integration unterstützt.
- Für die JIRA-Versionen 5 bis 7 ist nur das JIRA Server-API verfügbar. Damit können Sachverhalten zugeordnete Sachverhalte, Projekte und Sprints abgerufen werden.
- Für Jira-Version 7 sind sowohl das Jira Server-API als auch das Jira Agile-API verfügbar. Zusätzlich zu den vom Jira Server-API bereitgestellten Problemen und Projekten stellt das Jira Agile-API sicher, dass Scrum-Boards und alle Sprints abgerufen werden. Beachten Sie bitte, dass bestimmte Funktionalitäten möglicherweise nur in Nebenversionen von Jira verfügbar sind. Wenn ein Abruf (beispielsweise einer Statuskategorie) erfolglos ist, wird er im ausführlichen ADIF-Protokoll vermerkt, und der Import wird mit den verbleibenden Abrufen fortgesetzt.



JIRA verwendet OAuth für die Authentifizierung. Da es sich bei der von JIRA implementierten OAuth-Version jedoch um OAuth 1.0a 3-Legged (3LO) handelt und eine Anwenderinteraktion erforderlich ist, wird für die Integration eine Basis-Authentifizierung verwendet, um das JIRA Server-API und das JIRA Agile-API zu authentifizieren.

Die folgende Abbildung zeigt das in JIRA verwendete Metamodell. Sie sollten sich mit diesen Informationen vertraut machen, um die Integration in JIRA zu konfigurieren und JIRA-Objekte Alfabet -Objekten zuzuordnen.



Folgende Informationen sind verfügbar:

- [JIRA-Daten in Alfabet importieren](#)
- [Konfigurieren von Verbindungen für den Import von JIRA-Daten in Alfabet](#)
- [Importieren von JIRA-Daten in Alfabet mithilfe von ADIF-Schemata](#)
- [Alfabet -Daten nach JIRA exportieren](#)
- [Konfigurieren von Verbindungen und Zuordnungen für den Datenexport von Alfabet nach JIRA](#)
- [Konfigurieren der primären und sekundären Berichte](#)
- [Konfigurieren von Objektfilterberichten](#)
- [Konfigurieren des ADIF-Exportassistenten für den Export von Alfabet-Daten nach JIRA](#)
- [Konfigurieren der Ereignisvorlagen zum Auslösen der ADIF-Exportschemata für die Synchronisierung](#)
- [Konfigurieren von semantischen Verbindungen zum Verknüpfen und Synchronisieren von Jira-Projekten mit Alfabet-Objekten](#)
- [Erstellen einer JIRA-Verbindung für die projektbasierte Integration](#)
- [Erstellen einer JIRA-Verbindung für die architekturbasierte Integration](#)
- [Verknüpfung und Synchronisierung des JIRA-Projekts](#)

JIRA-Daten in Alfabet importieren

Je nachdem, welche Version von JIRA® Sie verwenden, kann Ihr Unternehmen JIRA-Projekte (z. B. auch Projektkategorien und Projektkomponenten), Sachverhalte, Boards und Sprints importieren und den entsprechenden Alfabet -Objektklassen zuordnen. Sie können verschiedene Sachverhaltfelder auswählen, die importiert werden sollen. Optional können Sie auswählen, ob die Daten für Sprints oder Sprints & Boards importiert werden sollen. Alle anderen Daten, wie z. B. Projects, Project Categories usw., werden standardmäßig mit den Sachverhalten importiert. Die Jira-Objekte können relevanten Alfabet -Objektklassen zugeordnet und als Applikationen, Komponenten, Installationen, Standardplattformen und Projekte importiert werden. So schlagen sie eine Brücke zwischen portfolio- und prioritätsbezogenen Entscheidungen auf Unternehmensebene und den operativen Entscheidungen von Scrum-Teams, Projektteams usw.



Beispielsweise kann die IT-Abteilung eines Bankunternehmens Bugs und Änderungsanforderungen für Ihre Handelssoftware in JIRA dokumentieren und ihre Betriebspläne in Alfabet erfassen. In diesem Fall könnten die Jira-Sachverhalte zum Beispiel als Funktionen, die für Applikationen geplant sind, deren Besitz und finanzielle Verwaltung über ICT-Objekte erfolgen, in Alfabet importiert werden. Das folgende Zuordnungsschema zeigt, wie Jira-Projekte einer ICT-Projektstruktur in Alfabet zugeordnet werden können:

JIRA-Klasse	Alfabet-Klasse
Projektkategorie	ICT-Objektgruppe
Projekt	ICT-Objekt
ProjectVersion	Applikation
Sachverhalt	Funktion

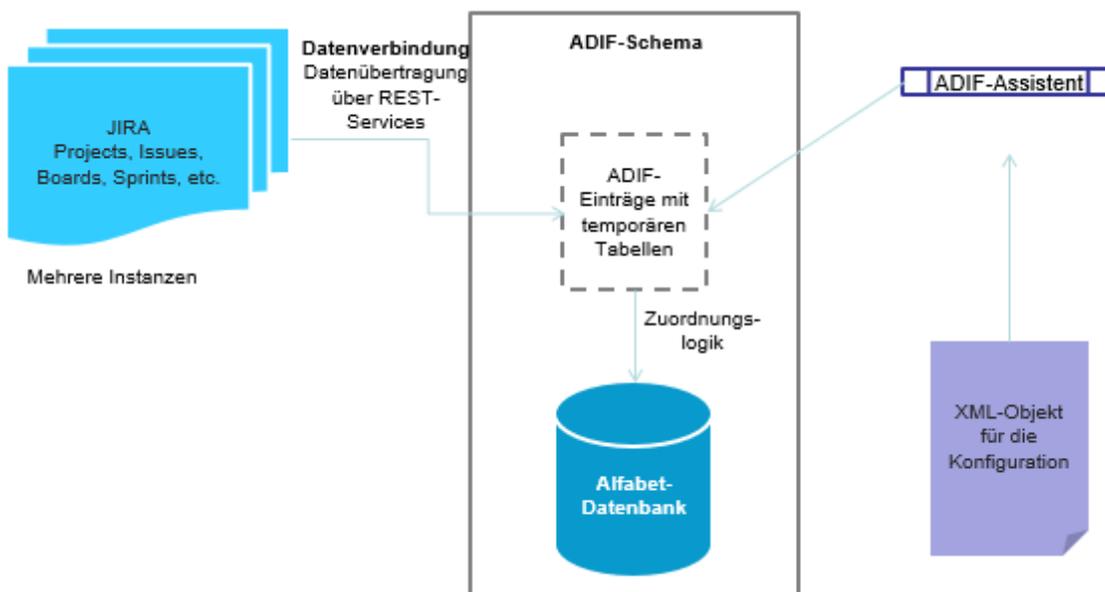


Abbildung: Übersicht über den Import aus Jira

Die folgende Konfiguration ist erforderlich, um Daten aus Jira in Alfabet zu importieren:

- 1) Die Verbindung mit JIRA muss im XML-Objekt **JIRAIntegrationConfig** in Alfabet Expand definiert werden.
- 2) Die zu importierenden Daten müssen über den **JIRA-Importassistenten** für ein ADIF-Importschema konfiguriert werden. Im **JIRA-Importassistenten** konfigurieren Sie, welche Verbindung für den Import verwendet werden soll und welche Projekte, Sachverhaltstypen, Sachverhaltstatus und standardmäßigen und benutzerdefinierten Sachverhaltfelder importiert werden sollen. Bei der Ausführung des Importschemas werden temporäre Tabellen für Jira-Sachverhalte und die relevanten Jira-Sachverhaltattribute erstellt. Je nachdem, mit welcher Version von JIRA Sie arbeiten, können Projekte, Sprints und Boards ebenfalls importiert werden. Dies hängt von der JIRA-Version und der Konfiguration des **JIRA-Importassistenten** ab.
- 3) Es erfordert eine zusätzliche Konfiguration des Importschemas, wenn Sie die importierten Sachverhaltattribute von JIRA temporären Tabellen zuordnen möchten, die für die entsprechenden Alfabet -Objektklassen erstellt werden. Ausführliche Informationen zur Konfiguration des ADIF-Importschemas, um Daten aus den temporären Datenbanktabellen in die Alfabet-Datenbank zu schreiben, finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework* beschrieben.
- 4) Die semantischen Verbindungen müssen konfiguriert werden, die das Integrationsmuster angeben, das für die im XML-Objekt **JIRAIntegrationConfig** konfigurierte Datenbankverbindung verwendet werden soll. Für jede im XML-Objekt **JIRAIntegrationConfig** konfigurierte Datenverbindung sollte eine semantische Definition erzeugt werden. Sie können mehrere semantische Datenverbindungen für jede Datenverbindung konfigurieren, die im XML-Element **DataConnection** des XML-Objekts **JIRAIntegrationConfig** konfiguriert ist. Für jede von Ihnen erstellte semantische Verbindung müssen Sie das Integrationsmuster angeben, damit die Objekte in der Alfabet -Objekthierarchie dem Jira-Projekt, der Jira-Projektversion und der Jira-Projektkomponente zugeordnet werden können. Sie können zwischen einer architekturbasierten und einer projektbasierten Zuordnung der Integrationsmuster wählen. Sie können eine beliebige Ebene in der Jira-Projekthierarchie angeben, um mit der Integration zu beginnen. Sie können beispielsweise Alfabet -Applikationen Jira-Projektversionen zuordnen.

Für einen projektbasierten Ansatz können die folgenden Integrationsmuster angegeben werden:

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
Jira-Projekte, -Projektversionen und -Projektkomponenten	Projekt -Stereotyp 1	Projekt -Stereotyp 2	Projekt -Stereotyp 3
Jira-Projekte und -Projektversionen	Projekt -Stereotyp 1	Projekt -Stereotyp 2	
Jira-Projekte und -Projektkomponenten	Projekt -Stereotyp 1		Projekt -Stereotyp 3

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
JIRA-Projekte	Projekt -Stereotyp 1		
Jira-Projektversionen		Projekt -Stereotyp 2	
Jira-Projektversionen und -Projektkomponenten		Projekt -Stereotyp 2	Projekt -Stereotyp 3
Jira-Projektkomponenten			Projekt -Stereotyp 3

Für einen architekturbasierten Ansatz können die folgenden Integrationsmuster angegeben werden:

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
Jira-Projekte, -Projektversionen und -Projektkomponenten	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Jira-Projekte und -Projektversionen	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	
JIRA-Projekte	ICT-Objekt (oder Stereotyp)		
Projektversionen		Applikation (oder Stereotyp)	
		Komponente (oder Stereotyp)	
		Standardplattform (oder Stereotyp)	
Projektversionen und Projektkomponenten	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Jira-Projekte, -Projektversionen und -Projektkomponenten		Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
Projektkomponenten		Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)

Folgende Informationen sind verfügbar:

- [Konfigurieren von Verbindungen für den Import von JIRA-Daten in Alfabet](#)
- [Importieren von JIRA-Daten in Alfabet mithilfe von ADIF-Schemata](#)

Konfigurieren von Verbindungen für den Import von JIRA-Daten in Alfabet

Es können mehrere Verbindungen mit JIRA® konfiguriert werden. Daten werden aus allen definierten JIRA-Instanzen über das gleiche ADIF-Importschema importiert. Die Verbindungen mit Jira® müssen im XML-Objekt **JIRAIntegrationConfig** definiert werden.

Die allgemeine Struktur der XML im XML-Objekt **JIRAIntegrationConfig** sieht wie folgt aus:

```
<JIRAIntegrationConfig>
  <Proxy Url="" UserName="" Password="" Domain="" />
  <DataConnections>
    <DataConnection
      Name="JiraInstance1"
      MajorVersion="7"
      ServerUrl="$JiraServer"
      UserName="$JiraUser"
      Password="$JiraPassword"
      PageSize="-1">
      <IssueRank Enabled="true" RankFieldId="10005"/>
    </DataConnection>
  </DataConnections>
</JIRAIntegrationConfig>
```

Das Root-XML-Element **JIRAIntegrationConfig** verfügt über ein untergeordnetes XML-Element **DataConnections**.



Beachten Sie, dass im XML-Objekt **JIRAIntegrationConfig** Servervariablen verwendet werden können, um den Wert des XML-Attributs zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation auszulesen, wenn eine Verbindung zu Jira hergestellt wurde. Weitere

Informationen zu Servervariablen finden Sie unter [Konfigurieren von Servervariablen für Integrations- und Interabilitätslösungen](#).

So definieren Sie eine Verbindung mit JIRA:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte** > **IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **JIRAIntegrationConfig**, und wählen Sie **XML bearbeiten....** Das XML-Objekt **JIRAIntegrationConfig** wird geöffnet.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist über das Attribut **XML-Vorlage** im Attributraster des XML-Objekts **JIRAIntegrationConfig** eine Vorlage verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. In diesem Fall bearbeiten Sie die nachfolgend beschriebenen XML-Elemente. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf.

- 3) Fügen Sie dem Root-XML-Element `JIRAIntegrationConfig` ein untergeordnetes XML-Element `DataConnections` hinzu.
- 4) Fügen Sie dem XML-Element `DataConnections` ein untergeordnetes XML-Element `DataConnection` hinzu. Das XML-Element `DataConnections` kann mehrere untergeordnete XML-Elemente `DataConnection` enthalten. Jedes von ihnen definiert eine Verbindung zu einer anderen JIRA-Instanz.
- 5) Spezifizieren Sie für jedes XML-Element `DataConnection` die folgenden XML-Attribute:
 - **Name:** Der Name der Verbindung zu JIRA, mit dem die Verbindung in der Konfiguration des ADIF-Schemas identifiziert wird. Für die Verbindung muss ein eindeutiger Name festgelegt werden.
 - **MajorVersion:** Geben Sie die JIRA-Hauptversion (5, 6, 7) an, um das entsprechende JIRA-API aufzurufen (für JIRA V. 5 und 6 das JIRA Server-API; für JIRA V. 7 das JIRA Server-API und das JIRA Agile-API). Sie sollten nur die Hauptversion (5, 6, 7) und nicht die Nebenversionsnummer der Jira-Instanz angeben. (Geben Sie beispielsweise `MajorVersion = 5` an, nicht `MajorVersion = 5.3.2`).
 - **ServerURL:** Geben Sie die URL zur JIRA-Instanz für die Verbindung an.
 - **UserName:** Geben Sie den Anwendernamen des Jira-Anwenders für die Verbindung an.
 - **Password:** Geben Sie das Passwort des JIRA-Anwenders für die Verbindung an.
 - **MaxThreadCount:** Die Anzahl der parallelen Threads, die für die gleichzeitige Datenübertragung verwendet werden können, ist im XML-Attribut `MaxThreadCount` standardmäßig auf `-1` gesetzt. Dieser Standardwert kann bei Bedarf geändert werden. Die Anzahl der CPU-Kerne ist in der Regel ein guter Ausgangswert für `MaxThreadCount`.



Bitte beachten Sie Folgendes hinsichtlich des Multithreading im Kontext der Jira-Integration:

- Wenn das XML-Attribut `MaxThreadCount` nicht definiert ist, gilt folgendes Standardverhalten:

- Wenn im **Jira-Importassistenten** die Option `Import Data Using JIRA Query Language Definition` ausgewählt ist, erfolgt kein Multithreading.
 - Wenn im **Jira-Importassistenten** die Option `Import Data Using List Box` ausgewählt ist, basiert die Anzahl der Threads auf der maximalen Anzahl von Optionen, die in den Feldern **Jira-Projekte**, **Jira-Sachverhaltstypen** oder **Jira-Sachverhaltstatus** ausgewählt wurde. Wenn der Anwender beispielsweise 5 Projekte, 3 Sachverhaltstypen und 2 Sachverhaltstatus auswählt, dann werden 5 Threads erzeugt (einer für jedes Projekt). Wenn im **JIRA-Importassistenten** die Option `Import Data Using List Box` ausgewählt ist und in den Feldern **Jira-Projekte**, **Jira-Sachverhaltstypen** oder **Jira-Sachverhaltstatus** keine Optionen ausgewählt wurden, dann basiert das Multithreading auf dem Feld mit den meisten Optionen. Wenn in den Feldern also 20 Projekte, 8 Sachverhaltstypen und 5 Sachverhaltstatus verfügbar sind, gibt es 20 Threads (einen für jedes Projekt).
 - Die Spezifikation des XML-Attributs `MaxThreadCount` setzt das Standardverhalten außer Kraft.
- `PageSize`: Gibt die Antwortseitengröße an und ist standardmäßig auf `-1` gesetzt. Dieser Standardwert wird vom Standardwert für die Seitengröße von der Jira-Instanz übernommen, er kann jedoch bei Bedarf geändert werden. Bitte sehen Sie sich in der API-Dokumentation die Informationen zu den Paginierungsfunktionen der verschiedenen APIs in JIRA an, bevor Sie den Standardwert `-1` ändern.
 - `HTTPTimeout`: Geben Sie bei Bedarf einen Zeitlimit-Wert in Sekunden an. Andernfalls wird das HTTP-Standardzeitlimit verwendet.
- 6) Fügen Sie dem XML-Element `DataConnection` optional ein untergeordnetes XML-Element `IssueRank` hinzu, um anzugeben, ob das Ranking der Vorgänge zusammen mit den Vorgängen importiert werden soll. Dies ist nur für aktive Sprints relevant. Um Sachverhalte mit ihren Rankingwerten auszufüllen, fügen Sie dem XML-Element `IssueRank` die folgenden XML-Attribute hinzu:
- `Enabled`: Auf `True` setzen.
 - `IssueRankID`: Geben Sie die ID des Felds ein, das für das Ranking von Vorgängen im Jira-Sprint verwendet wird. Das Ranking der Vorgänge wird dann im Attribut „RANK“ der temporären/Zuordnungstabelle „ISSUE_SPRINT“ im ADIF- Schema für alle aktiven Sprints ausgefüllt.
- 7) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Importieren von JIRA-Daten in Alfabet mithilfe von ADIF-Schemata

Der Datenimport aus JIRA wird durchgeführt, indem ein ADIF-Importschema ausgeführt wird. Es löst die Datenübertragung aus JIRA auf der Grundlage der konfigurierten Verbindungsangaben aus und speichert die eingehenden Daten auf vordefinierte Art in temporären Tabellen. Auf der Registerkarte „ADIF“ von

Alfabet Expand kann das Importschema mithilfe des **JIRA-Importassistenten** konfiguriert werden. Mit dem **JIRA-Importassistenten** können Sie auswählen, welche im XML-Objekt **JIRAIntegrationConfig** definierte Datenverbindung verwendet werden soll und welche Daten importiert werden sollen.

Sobald das ADF-Schema zum Abrufen der entsprechenden Inhalte mit dem **JIRA-Importassistenten** konfiguriert wurde, muss die Integration mit nachfolgenden Einträgen vervollständigt werden, um die importierten Artefakte in den temporären Tabellen den entsprechenden Objektklassen und Objekten von Alfabet zuzuordnen. Um die Integration der Daten in die Alfabet-Datenbank abzuschließen, ist deshalb eine zusätzliche Konfiguration des Importschemas erforderlich, um die importierten Artefakte in den temporären Tabellen den entsprechenden Objektklassen und Objekten von Alfabet zuzuordnen. Welche Konfiguration für diesen letzten Schritt des Imports erforderlich ist, hängt davon ab, wie Ihr Unternehmen die JIRA-Daten in Alfabet implementieren möchte. Schließlich müssen Ereignisvorlagen so konfiguriert werden, dass sie die konfigurierten ADIF-Importschemata für die Synchronisierungsfunktionen auslösen.



In diesem Abschnitt wird nur die Handhabung des **JIRA-Importassistenten** beschrieben. Ausführliche Informationen über die Konfiguration des ADIF-Importschemas, um Daten aus den temporären Datenbanktabellen in die Alfabet-Datenbank zu schreiben, finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*, sie werden hier nicht wiederholt. Informationen zur Konfiguration von Ereignisvorlagen finden Sie im Kapitel *Konfigurieren von Ereignissen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*. Ein Anwender mit einem administrativen Anwenderprofil kann den Erfolg der ausgelösten Ereignisse in der Funktionalität *Ereignisverwaltung* überprüfen. Informationen hierzu finden Sie im Abschnitt *Verwalten von Ereignissen* im Referenzhandbuch *Anwender- und Lösungsverwaltung*.

Die folgende Tabelle zeigt, welche JIRA-Objekte in Alfabet importiert werden und welche besonderen Einstellungen im **JIRA-Importassistenten** benötigt werden, um das JIRA-Objekt zu importieren:

JIRA-Objekt	Inhalt des Imports	Kontext des Imports
Issues	Im Feld Zu importierende Jira-Standard-Sachverhaltfelder ausgewählte Optionen.	Immer importiert
IssueCustomFields	Im Feld Zu importierende benutzerdefinierte Jira-Vorgangsfelder ausgewählte Optionen.	Immer importiert
IssueComponents	ProjectComponents die mit den importierten Issues verbunden sind.	Wird importiert, wenn Components im Feld Zu importierende benutzerdefinierte Jira-Vorgangsfelder ausgewählt ist.
IssueSprints	Sprints die mit den importierten Issues verbunden sind.	Wird importiert, wenn Sprints oder Sprints and Boards im Feld Jira-Sprint-Importoption ausgewählt ist. Nur relevant für JIRA-Versionen 5-7.

JIRA-Objekt	Inhalt des Imports	Kontext des Imports
Issue-FixVersions	ProjectVersions bei denen die zugeordneten Issues behoben wurden.	Wird importiert, wenn Fix Versions im Feld Zu importierende Jira-Standard-Sachverhaltfelder ausgewählt ist.
IssueAffectsVersions	ProjectVersions mit Auswirkungen auf die zugehörigen Issues.	Wird importiert, wenn Affects Versions im Feld Zu importierende Jira-Standard-Sachverhaltfelder ausgewählt ist.
IssueLinks	IssueLinks die mit den importierten Issues verbunden sind.	Wird importiert, wenn Linked Versions im Feld Zu importierende Jira-Standard-Sachverhaltfelder ausgewählt ist.
Projects	Projects die im Feld Jira-Projekte ausgewählt sind.	Immer importiert
ProjectVersions	ProjectVersions in JIRA.	Immer importiert
ProjectComponents	ProjectComponents in JIRA.	Immer importiert
ProjectCategories	ProjectCategories in JIRA.	Hängt von der JIRA-Version ab.
StatusCategories	Kategorisierung von Statuses, die mit Issues verbunden sind.	Hängt von der JIRA-Version ab.
Sprints	Sprints in JIRA.	Wird importiert, wenn Sprints oder Sprints and Boards im Feld Jira-Sprint-Importoption ausgewählt ist. Nur relevant für JIRA-Versionen 5-7. Wenn die JIRA-Version älter als Version 7 ist, werden nur Sprints importiert, die mit Issues verbunden sind.
GroupMembers	Groups und die zugehörigen Personen.	Hängt von der JIRA-Version ab.
Boards	Scrum-Boards in JIRA.	Wird importiert, wenn Sprints and Boards im Feld Jira-Sprint-Importoption ausgewählt ist. Nur relevant für JIRA-Version 7.

JIRA-Objekt	Inhalt des Imports	Kontext des Imports
Board-Sprints	Sprints die mit den importierten Boards verbunden sind.	Wird importiert, wenn Sprints and Boards im Feld Jira-Sprint-Importoption ausgewählt ist. Nur relevant für JIRA-Version 7.
Board-Projects	Projects die mit den importierten Boards verbunden sind.	Wird importiert, wenn Sprints and Boards im Feld Jira-Sprint-Importoption ausgewählt ist. Nur relevant für JIRA-Version 7.



Der JIRA-Anwender, der den JIRA-Import durchführt, sollte seine/ihre Sprache in JIRA auf die Hauptsprache von Alfabet (Englisch) eingestellt haben. Darüber hinaus muss der Anwender die erforderlichen Berechtigungen zum Importieren der folgenden Daten haben.

- Vorgänge (mithilfe des Search-Issues-API)
- Projekte
- Projektkategorien
- Projektkomponenten
- Projektversionen
- Statuskategorien
- Gruppenmitglieder
- Felder
- Vorgangstyp
- Vorgangstatus
- Boards
- Sprints pro Board
- Projekte pro Board



Beim Import von Jira-Inhalten in Alfabet können aufgrund von Zeichen, die für die XML- oder JSON-Verarbeitung nicht kompatibel sind, Fehler auftreten. Dies kann beispielsweise beim Importieren von JIRA- Fehlertickets auftreten, deren Beschreibungen kopierte Fehler-Speicherabbilder oder Code-Snippets enthalten. Daher wird dringend empfohlen, die Aussagekraft der Daten zu berücksichtigen, die von Jira in Alfabet importiert werden. Bei Jira-Issues für Fehlerbehebungen kann beispielsweise die Anzahl der Fehlerbehebungen für das Portfolio-Management relevant sein, während die detaillierten Beschreibungen der Fehlerbehebung normalerweise nicht von Bedeutung sind. Bitte beachten Sie dies, wenn Sie das entsprechende JIRA-Importschema mit dem **JIRA-Importassistenten** konfigurieren.

Nach der Definition des XML-Objekts **JIRAIntegrationConfig** müssen Sie ein neues ADIF-Importschema definieren, um die Datenintegration mithilfe des **JIRA-Importassistenten** zu konfigurieren:

- 1) Klicken Sie in der Registerkarte **ADIF** in Alfabet Expand mit der rechten Maustaste auf den Root-Knoten **ADIF-Schemata** im Explorer oder in einem beliebigen Unterordner, und wählen Sie **Importschema erzeugen** 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 Berichtverzeichnisses in technischen Prozessen. Der Name muss eindeutig sein und darf keine Leerzeichen oder Sonderzeichen enthalten.

- **Assistent:** Wählen Sie `JiraImport_Assistant` aus dem Dropdown-Listefeld aus.
- **Import-Datei erforderlich:** Stellen Sie sicher, dass der Wert `False` festgelegt wurde.
- **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 vom Typ **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.
- 3) Klicken Sie im Explorer mit der rechten Maustaste auf den Knoten des neuen ADIF-Importschemas, und wählen Sie **Erzeugen der ADIF-Schema-Details mithilfe von Jira-Importassistent** aus. Eine Warnmeldung wird angezeigt.



Wenn Sie den Assistenten zur Änderung eines vorhandenen, bereits konfigurierten ADIF-Importschemas verwenden, werden dadurch alle automatisch generierten Teile des ADIF-Schemas überschrieben. Falls Änderungen an diesen Importeinträgen vorgenommen wurden, gehen diese Änderungen verloren.

- 4) Klicken Sie auf **Ja**. Der Assistent wird in Ihrem Standard-Webbrowser geöffnet.
- 5) Im Feld **Datenverbindung** werden alle XML-Elemente `DataConnection` in Ihrem XML-Objekt **JIRAIntegrationConfig** angezeigt. Klicken Sie auf die Datenverbindung, um die Jira-Daten mithilfe des ausgewählten ADIF-Importschemas zu importieren, und klicken Sie dann auf die Schaltfläche **OK** unter dem Feld. Eine Meldung wird angezeigt, die Sie daran erinnert, das Metamodell in Alfabet Expand neu zu lesen.
- 6) Klicken Sie dazu in der Symbolleiste von Alfabet Expand das Menü **Metamodell** an, und wählen Sie die Option **Metamodell neu lesen** aus.
- 7) Kehren Sie zu dem Importschema zurück, das Sie für den Jira-Import erstellt haben, und klicken Sie mit der rechten Maustaste auf **Ändern der ADIF-Schema-Details mithilfe von Jira-Importassistent**.
- 8) Der **Jira-Importassistent** wird geöffnet. Wenn keine Filterkriterien bereitgestellt und keine ausdrücklichen Felder ausgewählt sind, werden standardmäßig alle Probleme zusammen mit ihren Standard- und benutzerdefinierten Feldern aus der ausgewählten Jira-Verbindung importiert. Definieren Sie die folgenden Felder im **JIRA-Importassistenten**:
- 9) Aktivieren Sie das Kontrollkästchen **Nettoänderungen**, wenn nur JIRA-Probleme importiert werden sollen, die seit der letzten Aktualisierung des ADIF-Schemas geändert wurden.
- 10) Definieren Sie in der Dropdownliste **Importfiltertyp**, welche Art von Import Sie implementieren möchten. Geben Sie eines der Folgenden an:
 - `Import Data Using List Box` wenn die zu importierenden Sachverhalte über bestimmte vordefinierte Filterkriterien gefiltert werden können. Bei diesen Filterkriterien wird entweder nach ausgewählten Projekten und/oder Sachverhaltstypen und/oder Sachverhaltstatus gefiltert. Definieren Sie folgende Felder:



Beachten Sie bitte, dass die folgenden Filterkriterien-Einstellungen kumulativ sind. Wenn also einige Projekte ausgewählt sind und der Anwender ein anderes Projekt sucht und auswählt, bleiben die vorher definierten Filterkriterien ausgewählt, solange sie nicht ausdrücklich deaktiviert werden. Beispiel: Wenn Sie ein Projekt auswählen, die Felder **Jira-Sachverhaltstypen** und **Jira-Sachverhaltstatus** definieren und anschließend im Feld **Jira-Projekte** ein zweites Projekt auswählen, bleiben die in den Feldern **Jira-Sachverhaltstypen** und **Jira-Sachverhaltstatus** für das erste Projekt festgelegten Einstellungen aktiviert, es sei denn, sie werden ausdrücklich deaktiviert.

- **Jira-Projekte:** Wählen Sie ein oder mehrere Jira-Projekte zum Importieren aus. Sie können einen Namen oder den Teil eines Namens eingeben und auf die Schaltfläche **Suchen** klicken, um Projekte zu suchen.
- **Jira-Sachverhaltstypen:** Wählen Sie einen oder mehrere Jira-Sachverhaltstypen zum Import aus.
- **Jira-Sachverhaltstatus:** Wählen Sie einen oder mehrere Jira-Sachverhaltstatus zum Import aus.
- `Import Data Using JIRA Query Language Definition`. um festzulegen, dass die zu importierenden Jira-Vorgänge gefiltert werden. Die Abfrage muss in der Jira-Abfragesprache

geschrieben und ins Feld **JQL-Abfrage eingeben** eingegeben werden. Weitere Informationen zur Jira-Abfragesprache finden Sie in der im Lieferumfang des Produkts enthaltenen Dokumentation.

- 11) Geben Sie im Feld **Jira-Sprint-Importoption** an, ob nur Sprints oder Sprints und Boards importiert werden sollen. Die Verfügbarkeit dieser Optionen hängt davon ab, welche Jira-Version für die Verbindung angegeben wurde.



Beachten Sie bitte, dass die Einstellungen für die Vorgangsauswahl kumulativ sind. Wenn also einige Vorgangsfelder ausgewählt sind und der Anwender ein anderes Projekt sucht und auswählt, bleiben die vorher ausgewählten Vorgangsfelder ausgewählt, solange sie nicht ausdrücklich deaktiviert werden.

- 12) Sie können festlegen, welche Informationen für die Projekte (und, falls relevant, Sprints und/oder Boards) importiert werden sollen:

- Geben Sie im Feld **Zu importierende Jira-Standard-Sachverhaltfelder** ein oder mehrere Standard-Sachverhaltfelder an, die importiert werden sollen. Wenn kein Standard-Sachverhaltfeld ausgewählt wird, werden alle Standard-Sachverhaltfelder importiert. Die temporären Tabellen in ADIF werden für Standard-Sachverhalte automatisch erstellt.
- Geben Sie im Feld **Zu importierende benutzerdefinierte Jira-Vorgangsfelder** ein oder mehrere benutzerdefinierte Sachverhaltfelder an, die importiert werden sollen. Sie können die benutzerdefinierten Sachverhaltfelder, die im Kombinationslistenfeld angezeigt werden, filtern, indem Sie in der Dropdownliste einen benutzerdefinierten Feldtyp (Zeichenfolge, Nummer usw.) auswählen. Wenn kein benutzerdefiniertes Sachverhaltfeld ausgewählt wird, werden alle benutzerdefinierten Sachverhaltfelder importiert.



Beachten Sie, dass die Felder für benutzerdefinierte Sachverhalte in eine temporäre Tabelle `IssueCustomFields` importiert werden. Benutzerdefinierte Sachverhaltfelder können skalare (z. B. `FieldName`, `Value` oder `FieldID`) oder referenzierte Werte (z. B. `Group`) aufweisen. Bei referenzierten Werten wird der temporären Tabelle die `REFID` hinzugefügt. Wenn für das benutzerdefinierte Sachverhaltfeld keine `REFID` verfügbar ist, wird dem temporären Feld `REFNAME` hinzugefügt.



Bitte beachten Sie, dass die Definition der Einstellungen des benutzerdefinierten Feldtyps (`String`, `Integer` usw.) kumulativ ist. Wenn also einige benutzerdefinierte Feldtypen (`String`, `Integer` usw.) ausgewählt sind und der Anwender ein anderes Projekt sucht und auswählt, bleiben die vorher ausgewählten benutzerdefinierten Feldtypen (`String`, `Integer` usw.) ausgewählt, solange sie nicht ausdrücklich deaktiviert werden.

- 13) Klicken Sie auf die Schaltfläche **OK**, damit das ADF-Schema die angegebenen Inhalte aus Jira abrufen und der **JIRA-Importassistent** geschlossen wird. Die temporären Tabellen werden generiert und im Explorer unter dem Importschema-Knoten angezeigt. Oder klicken Sie auf **Abbrechen**, um den Editor zu schließen, ohne die Ausführung des Importschemas auszulösen. Wenn Sie auf **Abbrechen** klicken, werden Ihre Einstellungen im **JIRA-Importassistenten** nicht gespeichert.
- 14) Schließen Sie das Browserfenster, und führen Sie in Alfabet Expand **Metamodell > Erneut aus Datenbank lesen** aus. Sie sehen dann alle automatisch generierten ADIF-Elemente im ADIF-Schema. Jeder ADIF-Importeintrag verfügt über Datenimporte in temporären Tabellen, die über die Attributelemente im Ordner **Attribute** definiert werden. Der Import in die Alfabet-Datenbank - Standardtabellen ist nicht in der Konfiguration enthalten.

- 15) Um die Integration der Daten in die Alfabet-Datenbank abzuschließen, ist eine zusätzliche Konfiguration des Importschemas erforderlich, um die importierten JIRA-Daten in den temporären Tabellen den für die entsprechenden Objektklassen und Objekten von Alfabet erstellten Tabellen zuzuordnen. Welche Konfiguration für diesen letzten Schritt des Imports erforderlich ist, hängt davon ab, wie Ihr Unternehmen die JIRA-Daten in Alfabet implementieren möchte. Weitere Informationen zu den über ADIF-Importschemata verfügbaren Konfigurationsoptionen finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

Alfabet -Daten nach JIRA exportieren

Der Export von Alfabet -Daten in JIRA® wird bereitgestellt, um die Planung und Priorisierung der operativen Entwicklungsarbeit zu unterstützen und um Unterstützung für eine bimodale IT und agile Projekte im Kontext der Portfoliobewertung zu bieten. Das Ziel des Jira-Exports ist es, in Jira neue Sachverhalte auf der Grundlage von Sachhalten zu erzeugen, die über Anforderungen, Funktionen, Tickets, Steuerungselemente usw. in Alfabet erfasst wurden. In JIRA werden Sachverhalte normalerweise einem Projekt zugewiesen, und für Projekte wird in der Regel eine `ProjectVersion` und möglicherweise eine `ProjectComponent` definiert.



Beispielsweise kann ein Finanzinstitut Anforderungen erfassen, um die Änderung seiner Handelsfunktionen zu planen. Die Anforderungen werden Projekten in Alfabet zugeordnet. Der Projektstereotyp „Programm“ in Alfabet kann `Projects` in der JIRA-Instanz zugeordnet werden. Die Anforderungen werden als Sachverhalte in JIRA exportiert, um die Projekte zu implementieren und die erforderlichen Änderungen umzusetzen. Der untergeordnete Projektstereotyp kann der `JIRA-ProjectVersion` zugeordnet werden, und der Projektstereotyp „Projektschritt“ würde der `JIRA-ProjectComponent` entsprechen. Das folgende Zuordnungsschema zeigt, wie JIRA-Projekte einer Projektstruktur in Alfabet zugeordnet werden können:

Alfabet-Klasse	JIRA-Klasse
Projekt-Stereotyp 1	<code>Project</code>
Projekt-Stereotyp 2	<code>ProjectVersion</code>
Projekt-Stereotyp 3	<code>ProjectComponent</code>
Anforderung	<code>Issue</code>



Eine Übersicht über alle Jira-Objekte, die durch den Import von Alfabet -Daten in Jira importiert werden können, finden Sie im Abschnitt [JIRA-Daten in Alfabet importieren](#).

Da Sachverhalte in JIRA in der Regel JIRA-Projekten zugeordnet sind, muss Ihr Unternehmen zuerst die folgenden Fragen in Betracht ziehen, um ein Konzept dafür zu entwickeln, wie die Unternehmensdaten aus Alfabet in JIRA zugeordnet werden:

- Welchen Jira-Projekten werden die exportierten Issues zugeordnet? So könnte ein Jira-Projekt auf ein Produkt wie „FD Trading“ ausgelegt sein. Die Projekte in Jira können entweder durch die

Angabe des Projektschlüssels der Projekte oder durch die Ermittlung der relevanten Projekte anhand zentraler Bezeichner ermittelt werden.

- Welchem Jira-Issue-Typ werden die Informationen aus Alfabet zugeordnet? Beispiel: die Jira-Issue-Typen „Fehler“ oder „Änderungsanträge“.
- Welche Objekte aus Alfabet werden Jira als Issues zugeordnet? In der Regel wäre dies eine Anforderung, ein Issue, eine Funktion oder ein Steuerungselement, andere Klassen könnten jedoch ebenfalls zugeordnet werden. So könnte die Anforderung, das Risikomanagement zu integrieren, als Änderungsantrag in Jira exportiert werden.
- Werden die bestehenden Issues in Jira mit Daten aus den Alfabet -Objekten aktualisiert, oder werden in Jira neue Issues erstellt?
- Sobald klar ist, welche Alfabet -Objekte in welche Jira-Projekte exportiert werden, muss das Integrationsmuster angegeben werden, damit die entsprechenden Alfabet -Objekte Jira-Projekten, -Projektversionen und -Projektkomponenten zugeordnet werden. Sie können eine beliebige Ebene in der Jira-Projekthierarchie angeben, um mit der Integration zu beginnen. Es ist möglich, mindestens ein vordefiniertes Integrationsmuster zu implementieren. Zum Beispiel können Sie bei einem der Integrationsmuster für ICT-Objekte ein ausgewähltes ICT-Objektstereotyp dem Jira-Projekt, eine Applikation (bzw. das Stereotyp) der Jira-Projektversion und eine lokale Komponente (bzw. das Stereotyp) der Jira-Projektkomponente zuordnen. Ob Ihr Unternehmen ein Integrationsmuster für ICT-Objekte oder für Projekte wählt, hängt von der Methode ab, wie Sie Jira einsetzen. Wenn Ihr Unternehmen in Alfabet eine Zuordnung zwischen Anforderungen und Produkten vornimmt, kann das Integrationsmuster für ICT-Objekte passender sein, aber wenn Anforderungen Projekten zugeordnet werden, ist das Integrationsmuster einer Projekthierarchie möglicherweise besser geeignet.
- Mit welcher Jira-Instanz werden Sie eine Verbindung herstellen, um die Alfabet -Objekte den Issues für die spezifischen Projekte zuordnen zu können? Bei der Integration von Alfabet mit Jira werden Szenarien mit mehreren Instanzen unterstützt, wie sie bei Großunternehmen typisch sind.

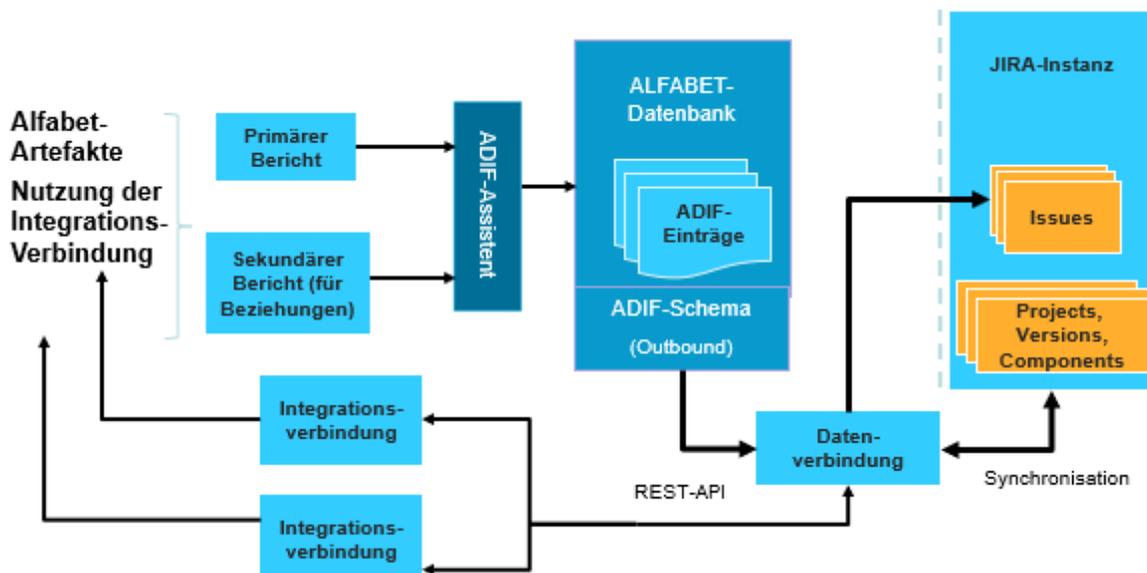


Abbildung: Übersicht über den Export in Jira

In der nachstehenden Übersicht werden die erforderlichen Konfigurationsschritte für den Datenexport in Jira erläutert.

- 1) Konfigurieren Sie benutzerdefinierte Eigenschaften des Typs `String`, um in JIRA Werte für Projekte, Projektversionen und Projektkomponenten einzupflegen. Für JIRA-Bezeichner wie `Name.ID` und `Key` sind benutzerdefinierte Eigenschaften erforderlich. Beachten Sie die folgenden Zeichenbeschränkungen für die JIRA-Benutzeroberfläche:

Werte in Jira	JIRA-Projekte	JIRA-Projektversionen/Projektkomponenten	JIRA-Sachverhalte
Name	80	255	nicht relevant
ID	18	18	18
Key	10	nicht relevant	10
Summary	nicht relevant	nicht relevant	255

- 2) Konfigurieren Sie das XML-Objekt ***JIRAIntegrationConfig***. Das XML-Objekt ***JIRAIntegrationConfig***, mit dem Sie Daten aus JIRA in Alfabet importieren können, wird auch zum Konfigurieren der Verbindung für den Export von Alfabet -Daten in JIRA verwendet. Im XML-Objekt ***JIRAIntegrationConfig*** muss Folgendes definiert werden:
- Das XML-Element `DataConnection`: Geben Sie Datenverbindungen zu mindestens einer Jira-Instanz an.
 - Das XML-Element `ExportSettings`: Geben Sie die Alfabet -Berichtskategorien an, anhand derer die primären und sekundären konfigurierten Berichte vom ADIF-Assistenten aufgelistet werden, in denen die Objekte und Referenzen in Alfabet identifiziert werden, die als Sachverhalte in Jira exportiert werden.
 - Das XML-Element `JIRADataTypeIdentifierTypes`: Geben Sie die für die Suche nach Projekten und Sachverhalten in JIRA benötigten Bezeichner an. Dies beinhaltet einen Bezeichner für die Suche nach Projekten (z. B. `Key`) und einen Bezeichner für die Suche nach Sachverhaltstypen (z. B. `Name`).
 - Das XML-Element `JIRAProjectFilters`: Legen Sie Filter fest, mit denen Sie nach Projekten suchen können, während Sie diese Alfabet -Artefakten zuordnen. Beispielsweise könnte der Filter angeben, dass nur Projekte ausgegeben werden, bei denen der Projekttyp „Software“ oder „Business“ ist.
 - Das XML-Element `AlfabetClassMappingStandardPropertySettings`: Geben Sie die Standard-Objektklasseneigenschaften des Typs `String` an, mit denen die Felder in den Editoren **Jira-Verbindung für projektbasierte Integrationen** und **Jira-Verbindung für architekturbasierte Integrationen** ausgefüllt werden sollen, die in der Ansicht **Jira-Verbindung** der Funktionalität **Konfiguration von Integrationslösungen** verfügbar sind.
- 3) Konfigurieren Sie die Berichte, anhand derer die Alfabet -Objekte identifiziert werden, die als Issues exportiert werden. Sie können zwei konfigurierte Berichte konfigurieren: Ein primärer Bericht ermöglicht die Suche nach Alfabet -Objekten wie Anforderungen, Issues, Funktionen oder Steuerungselementen auf Basis von skalaren Attributen. Der sekundäre Bericht dient dazu, die

Issue-bezogenen Informationen durch Referenz-Arrays, Zeichenfolgen-Arrays usw. (n:n-Beziehungen) zu ergänzen.

- 4) Konfigurieren Sie einen Bericht, um die Alfabet -Objekte zu finden, die in Jira-Projekte integriert werden können. Die Objekte sind in der Regel über die Stereotypspezifikation im Feld **Stereotypfilter** der Editoren **Jira-Verbindung für projektbasierte Integrationen** und **Jira-Verbindung für architekturbasierte Integrationen** zu finden, die in der Funktionalität **Konfiguration von Integrationslösungen** in der Ansicht **JIRA-Verbindung** verfügbar sind. Der konfigurierte Bericht kann in den Editoren im Feld **Objektfilterbericht** als zusätzlicher Mechanismus zum Suchen von Objekten ausgewählt werden. Der Bericht muss eine Liste der Alfabet -REFSTRs der geeigneten Projekte bereitstellen, die zur Integration in Jira-Projekte zulässig sind. Die Editor-Option **Mit Jira-Projekt verknüpfen** ist im Menü **Aktion** in den Objektprofilen der Projekte verfügbar, die über die Definition des Felds **Stereotypfilter** oder des Felds **Objektfilterbericht** gefunden werden.
- 5) Legen Sie fest, welche Informationen aus Alfabet zum Erzeugen von JIRA-Sachverhalten verwendet werden sollen. Definieren Sie dazu das ADIF-Exportschema mithilfe des **JIRA-Exportassistenten**. Der **JIRA-Exportassistent** ist auf zwei Alfabet -Berichte konzentriert, die für die Suche nach den zu exportierenden Alfabet -Objekten (primärer Bericht) und den zu exportierenden Referenzen (sekundärer Bericht) konfiguriert sind. Folgendes muss definiert sein:



Der JIRA-Export beruht auf dem Prinzip der Konsistenz zwischen JIRA-Benutzeroberfläche und JIRA-API. Die JIRA-Benutzeroberfläche und die JIRA-API können jedoch verschiedene Formate haben. Dies ist zum Beispiel bei Datumsfeldern der Fall. Bitte beachten Sie, dass sich Anwender bei der Konfiguration des ADIF-Exportschemas nicht ausschließlich auf die Filterfelder in der JIRA-Benutzeroberfläche verlassen sollten, sondern wissen müssen, welches Format die JIRA-API erwartet, um das ADIF-Exportschema entsprechend zu konfigurieren. Wenn die Formate inkonsistent sind, bringt der Export von Alfabet nach JIRA möglicherweise nicht die richtigen Ergebnisse.

- Wählen Sie auf der Registerkarte **Ordnen Sie primäre Alfabet-Berichtsspalten den JIRA-Problemfeldern zu** die zu verwendende Datenverbindung aus.
 - Wählen Sie ein Projekt und einen Issue-Typ aus, für das/den mindestens ein Issue definiert ist und das/der eine Jira-Issue-Struktur aufweist, die für den vorgesehenen Export repräsentativ ist. Die Zuordnungstabelle wird vorab mit den Feldern ausgefüllt, die für die Aktionen zum Erstellen und Aktualisieren mit Blick auf das ausgewählte Projekt und den ausgewählten Sachverhaltstyp festgelegt sind. Alle Felder, die nicht im Zuge des Schema-Abrufs zurückgegeben werden, können einzeln hinzugefügt werden. Die Felder der JIRA-Sachverhalte werden den entsprechenden Spalten des primären Berichts zugeordnet.
 - Wählen Sie den primären Bericht für die Suche nach den Alfabet -Objekten aus.
 - Ordnen Sie für jedes im Datensatz angezeigte Feld das relevante Alfabet -Projekt unter **Name der Berichtsspalte** zu. Der Datensatz zeigt, welche JIRA-Felder zum Erstellen und Aktualisieren von Problemen obligatorisch sind.
 - Wählen Sie auf der Registerkarte **Ordnen Sie sekundäre Alfabet-Berichtsspalten den JIRA-Problemfeldern zu** mindestens einen sekundären Bericht aus, um die Zuordnung für mehrere Werte umfassende Attribute von Jira-Issues festzulegen, etwa die Zuordnung von Jira-Projektversionen und Jira-Projektcomponenten.
- 6) Nach Abschluss der Zuordnung müssen Ereignisvorlagen so konfiguriert werden, dass sie die ADIF-Exportschemata auslösen, wenn Anwender die Funktionalität **Mit Projektdaten synchronisieren** in den entsprechenden Objektprofilen ausführen. Für den Export aus Alfabet sollte die Ereignisvorlage einen Satz von Objekten zurückgeben, die Jira-Sachverhalten

zugeordnet werden können. Die von der Ereignisvorlage zurückgegebenen Menge an Objekten muss eine Teilmenge der Objekte sein, die vom primären Bericht des ADIF-Prozesses zurückgegeben werden, auf den die Ereignisvorlage verweist. Nur die zurückgegebenen Objekte, die sowohl der Ereignisvorlage als auch dem primären Bericht gemeinsam sind, werden in Jira exportiert. Informationen zum Konfigurieren von Ereignisvorlagen für den Export aus Alfabet finden Sie im Abschnitt [Konfigurieren der Ereignisvorlagen zum Auslösen der ADIF-Exportschemata für die Synchronisierung](#).

- 7) Die semantischen Verbindungen müssen konfiguriert werden, die das Integrationsmuster angeben, das für die im XML-Objekt **JIRAIntegrationConfig** konfigurierte Datenbankverbindung verwendet werden soll. Für jede im XML-Objekt **JIRAIntegrationConfig** konfigurierte Datenverbindung sollte eine semantische Definition erzeugt werden. Sie können mehrere semantische Datenverbindungen für jede Datenverbindung konfigurieren, die im XML-Element **DataConnection** des XML-Objekts **JIRAIntegrationConfig** konfiguriert ist. Für jede von Ihnen erstellte semantische Verbindung müssen Sie das Integrationsmuster angeben, damit die Objekte in der Alfabet -Objekthierarchie dem Jira-Projekt, der Jira-Projektversion und der Jira-Projektkomponente zugeordnet werden können. Sie können zwischen einer architekturbasierten und einer projektbasierten Zuordnung der Integrationsmuster wählen. Sie können eine beliebige Ebene in der Jira-Projekthierarchie angeben, um mit der Integration zu beginnen. Sie können beispielsweise Alfabet -Applikationen Jira-Projektversionen zuordnen.

Für einen projektbasierten Ansatz können die folgenden Integrationsmuster angegeben werden:

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
Jira-Projekte, -Projektversionen und -Projektkomponenten	Projekt -Stereotyp 1	Projekt -Stereotyp 2	Projekt -Stereotyp 3
Jira-Projekte und -Projektversionen	Projekt -Stereotyp 1	Projekt -Stereotyp 2	
Jira-Projekte und -Projektkomponenten	Projekt -Stereotyp 1		Projekt -Stereotyp 3
JIRA-Projekte	Projekt -Stereotyp 1		
Jira-Projektversionen		Projekt -Stereotyp 2	
Jira-Projektversionen und -Projektkomponenten		Projekt -Stereotyp 2	Projekt -Stereotyp 3
Jira-Projektkomponenten			Projekt -Stereotyp 3

Jira-Sachverhalte werden der entsprechenden Alfabet -Objektklasse (z. B. **Anforderung, Feature, Sachverhalt, Steuerungselement** usw.) zugeordnet, die im entsprechenden ADIF-Schema für die Jira-Verbindung angegeben ist.

Für einen architekturbasierten Ansatz können die folgenden Integrationsmuster angegeben werden:

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
Jira-Projekte, -Projektversionen und -Projektkomponenten	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Jira-Projekte und -Projektversionen	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	
JIRA-Projekte	ICT-Objekt (oder Stereotyp)		
Projektversionen		Applikation (oder Stereotyp)	
		Komponente (oder Stereotyp)	

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
		Standardplattform (oder Stereotyp)	
Projektversionen und Projektkomponenten	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Jira-Projekte, -Projektversionen und -Projektkomponenten		Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Projektkomponenten		Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)

Jira-Sachverhalte werden der entsprechenden Alfabet -Objektklasse (z. B. **Anforderung, Feature, Sachverhalt, Steuerungselement** usw.) zugeordnet, die im entsprechenden ADIF-Schema für die Jira-Verbindung angegeben ist.

Folgende Informationen sind verfügbar:

- [Konfigurieren von Verbindungen und Zuordnungen für den Datenexport von Alfabet nach JIRA](#)
- [Konfigurieren der primären und sekundären Berichte](#)
- [Konfigurieren von Objektfilterberichten](#)
- [Konfigurieren des ADIF-Exportassistenten für den Export von Alfabet-Daten nach JIRA](#)

- [Konfigurieren der Ereignisvorlagen zum Auslösen der ADIF-Exportschemata für die Synchronisierung](#)

Konfigurieren von Verbindungen und Zuordnungen für den Datenexport von Alfabet nach JIRA

Das Folgende muss im XML-Objekt **JIRAIntegrationConfig** konfiguriert werden:

- Geben Sie Datenverbindungen zu mindestens einer JIRA-Instanz im XML-Element `DataConnection` an.
- Geben Sie die Alfabet -Berichtskategorien an, anhand derer die primären und sekundären konfigurierten Berichte vom ADIF-Assistenten aufgelistet werden, in denen die Objekte und Referenzen in Alfabet identifiziert werden, die als Sachverhalte in Jira exportiert werden. Dies geschieht über das XML-Element `ExportSettings`.
- Geben Sie die für die Suche nach Projekten und Sachverhalten in Jira benötigten Bezeichner über das XML-Element `JIRADataTypeIdentifiers` an. Dies beinhaltet einen Bezeichner für die Suche nach Projekten (z. B. `Key`) und einen Bezeichner für die Suche nach Sachverhaltstypen (z. B. `Name`).
- Geben Sie Filter an, um die Jira-Projekte, -Projektversionen und -Projektkomponenten einzuschränken, die im Editor **Mit Jira-Projekt verknüpfen** verfügbar sind, wenn Anwender ein Alfabet -Objekt mit Jira verknüpfen. Dies geschieht über das XML-Element `JIRAProjectFilters`. Beispielsweise könnte der Filter angeben, dass nur Projekte ausgegeben werden, bei denen der Projekttyp „Software“ oder „Business“ ist.
- Geben Sie die Standard-Objektklasseneigenschaften des Typs `String` an, mit denen die Jira-Eigenschaftsfelder in den Editoren **Jira-Verbindung für projektbasierte Integrationen** und **Jira-Verbindung für architekturbasierte Integrationen** ausgefüllt werden sollen, die in der Ansicht **Jira-Verbindung** der Funktionalität **Konfiguration von Integrationslösungen** verfügbar sind. Dies geschieht über das XML-Element `AlfabetClassMappingStandardPropertySettings`.



Ein Beispiel für die Spezifikation des XML-Objekts **JIRAIntegrationConfig** finden Sie im Attribut **XML-Vorlage**.

Die Verbindung mit Jira® muss im XML-Objekt **JIRAIntegrationConfig** definiert werden. Es können mehrere Verbindungen mit JIRA konfiguriert werden. Daten werden aus allen definierten JIRA-Instanzen über das gleiche ADIF-Importschema importiert.



Beachten Sie, dass im XML-Objekt **JIRAIntegrationConfig** Servervariablen verwendet werden können, um den Wert eines XML-Elements zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation auszulesen, wenn eine Verbindung zu JIRA hergestellt wurde. Weitere Informationen zu Servervariablen finden Sie unter [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).

So definieren Sie eine Verbindung mit JIRA:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte** > **IntegrationSolutions**.

- 2) Klicken Sie mit der rechten Maustaste auf **JIRAIntegrationConfig**, und wählen Sie **XML bearbeiten...**. Das XML-Objekt **JIRAIntegrationConfig** wird geöffnet.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist über das Attribut **XML-Vorlage** im Attributraster des XML-Objekts **JIRAIntegrationConfig** eine Vorlage verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. In diesem Fall bearbeiten Sie die nachfolgend beschriebenen XML-Elemente. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf.

- 3) Fügen Sie dem Root-XML-Element `JIRAIntegrationConfig` ein untergeordnetes XML-Element `DataConnections` hinzu.
- 4) Fügen Sie dem XML-Element `DataConnections` ein untergeordnetes XML-Element `DataConnection` hinzu. Das XML-Element `DataConnections` kann mehrere untergeordnete XML-Elemente `DataConnection` enthalten. Jedes von ihnen definiert eine Verbindung zu einer anderen JIRA-Instanz.
- 5) Spezifizieren Sie für jedes XML-Element `DataConnection` die folgenden XML-Attribute:
- **Name:** Der Name der Verbindung zu JIRA, mit dem die Verbindung in der Konfiguration des ADIF-Schemas identifiziert wird. Für die Verbindung muss ein eindeutiger Name festgelegt werden. Der Name der Datenverbindung kann im Feld **Datenverbindung** der Editoren **Jira-Verbindung für projektbasierte Integrationen** und **Jira-Verbindung für architekturbasierte Integrationen** einer semantischen Jira-Verbindung zugewiesen werden.
 - **MajorVersion:** Geben Sie die JIRA-Hauptversion (5, 6, 7) an, um das entsprechende JIRA-API aufzurufen (für JIRA V. 5 und 6 das JIRA Server-API; für JIRA V. 7 das JIRA Server-API und das JIRA Agile-API). Sie sollten nur die Hauptversion (5, 6, 7) und nicht die Nebenversionsnummer der Jira-Instanz angeben. (Geben Sie beispielsweise `MajorVersion = 5` an, nicht `MajorVersion = 5.3.2`).
 - **ServerURL:** Geben Sie die URL zur JIRA-Instanz für die Verbindung an.
 - **UserName:** Geben Sie den Anwendernamen des Jira-Anwenders für die Verbindung an.
 - **Password:** Geben Sie das Passwort des JIRA-Anwenders für die Verbindung an.
 - **MaxThreadCount:** Die Anzahl der parallelen Threads, die für die gleichzeitige Datenübertragung verwendet werden können, ist im XML-Attribut `MaxThreadCount` standardmäßig auf -1 gesetzt. Dieser Standardwert kann bei Bedarf geändert werden. Die Anzahl der CPU-Kerne ist in der Regel ein guter Ausgangswert für `MaxThreadCount`. Die Spezifikation des XML-Attributs `MaxThreadCount` setzt das Standardverhalten außer Kraft.
 - **HTTPTimeout:** Geben Sie bei Bedarf einen Zeitlimit-Wert in Sekunden an. Andernfalls wird das HTTP-Standardzeitlimit verwendet.
- 6) Fügen Sie dem Root-XML-Element `JIRAIntegrationConfig` ein untergeordnetes XML-Element `ExportSettings` hinzu:
- Geben Sie die Alfabet -Berichtskategorien an, die die konfigurierten Berichte enthalten, die im **ADIF-Exportassistenten** verfügbar sein sollen. Die Berichtskategorien sollten die relevanten primären und sekundären konfigurierten Berichte enthalten, in denen die Objekte und Referenzen identifiziert werden, die als Probleme in Alfabet exportiert werden. Es können sowohl für die primären als auch für die sekundären Berichte die gleichen Berichtskategorien angegeben werden:

- **PrimaryAlfabetReportCategory:** Geben Sie eine oder mehrere relevante Berichtskategorien der konfigurierten Berichte zur Ermittlung der Alfabet -Objekte, die exportiert werden sollen, an. Wenn mehrere Berichtskategorien angegeben werden, müssen sie als durch Kommas getrennte Werte aufgelistet werden. Alle konfigurierten Berichte, die dem angegebenen Kategoriewert zugeordnet sind, werden im Feld **Primärer Alfabet-Bericht** in der Registerkarte **Ordnen Sie primäre Alfabet-Berichtsspalten den Jira-Sachverhaltfeldern zu** im **ADIF-Exportassistenten** aufgelistet.
- **SecondaryAlfabetReportCategory:** Wählen Sie auf der Registerkarte „Ordnen Sie sekundäre Alfabet-Berichtsspalten den JIRA-Problemfeldern zu“ mindestens einen sekundären Bericht aus, um die Zuordnung für mehrere Werte umfassende Attribute von JIRA-Sachverhalten festzulegen, etwa die Zuordnung von JIRA-Projektversionen und JIRA-Projektcomponenten. Wenn mehrere Berichtskategorien angegeben werden, müssen sie als durch Kommas getrennte Werte aufgelistet werden. Alle konfigurierten Berichte, die dem angegebenen Kategoriewert zugeordnet sind, werden im Feld **Wählen Sie einen sekundären Bericht aus, um die Zuordnung zu definieren** in der Registerkarte **Ordnen Sie sekundäre Alfabet-Berichtsspalten den JIRA-Problemfeldern zu** im **ADIF-Exportassistenten** aufgelistet.
- **UpdateJIRAObjectsOnStructureSync:** Geben Sie `True` ein, wenn JIRA-Projekte beim Synchronisieren der Architektur/Projektstruktur von Alfabet aktualisiert werden sollen. Geben Sie `False` ein, wenn JIRA-Projekte beim Synchronisieren der Architektur/Projektstruktur von Alfabet nicht aktualisiert werden sollen. Die Standardeinstellung des XML-Attributs `UpdateJIRAObjectsOnStructureSync` ist `True`.
- Geben Sie Folgendes an, um die Leistung beim Importieren einer großen Anzahl von Jira-Sachverhalten in Alfabet zu verbessern:
 - **BulkCreateMode:** Geben Sie `True` für die Synchronisierung großer Datenmengen mit einzelnen Jira-Instanzen an. Dies ist nicht relevant, wenn eine große Anzahl von Sachverhalten über eine große Anzahl von Jira-Instanzen hinweg synchronisiert wird. Der Standardwert ist `False`.
 - **BulkCreateBatchSize:** Geben Sie eine Batch-Größe für die Erzeugung von Sachverhalten im verbundenen Jira-System an. Der Standardwert lautet „200“.
 - **JIRAFIELDIDToMapAlfabetObject:** Geben Sie das benutzerdefinierte Feld in Jira an, in dem die Objekt-ID des Alfabet -Objekts für den Sachverhalt in Jira gespeichert wird. Es wird empfohlen, dass dieses Feld in Jira schreibgeschützt ist. Beachten Sie, dass das benutzerdefinierte Feld in allen Jira-Instanzen vorhanden sein sollte, die zum Erzeugen und Aktualisieren/Bearbeiten von Sachverhalten in Jira verwendet werden, damit das Feld im zugehörigen Jira-API verfügbar ist. Dieses Feld muss durch den voll qualifizierten Bezeichner dargestellt werden (z. B. `customfield_10202`, wobei die fünfstellige Ganzzahl die Variable ist).
 - **BulkCreateMaxThreadCount:** Geben Sie die Anzahl der parallelen Threads an, die beim Erzeugen von Sachverhalten in Jira erzeugt werden sollen. Diese Zahl ist eine Funktion verschiedener Aspekte der Jira-Installation (z. B. Maschinengröße und Konfiguration des Apache-Webservers, der als Applikationsserver für Jira verwendet wird) sowie der Größe, Infrastruktur und Konfiguration der Jira-Datenbank (insbesondere der Größe des verfügbaren Verbindungspools). Wenn nicht anders angegeben, wird der Eigenschaftswert `MaxThreadCount` verwendet.
- 7) Geben Sie die Bezeichner (`ID`, `Key` oder `Name`) an, die benötigt werden, um Jira-Projekte zu finden, wenn die Struktur von Alfabet (Projekthierarchie oder ICT-Objektstruktur) mit der Projekthierarchie

von Jira synchronisiert wird oder Sachverhalte nach Jira exportiert werden. JIRA-Bezeichner sind zum Beispiel `ID`, `Key` oder `Name` für Projekte und `Key` oder `Name` für Sachverhalte. Fügen Sie zu diesem Zweck dem Root-XML-Element `JIRAIntegrationConfig` ein untergeordnetes XML-Element `JIRADatatypeIdentifierTypes` hinzu, und legen Sie folgende XML-Attribute für das XML-Element `DefaultDataTypeIdentifierTypes` fest:

- `DefaultDataTypeIdentifierType`: Der Standardbezeichner, wenn über die anderen angegebenen Datentypen keine Übereinstimmungen gefunden werden. Dies ist in der Regel `ID`.
 - `IssueDataTypeIdentifierType`: Geben Sie einen Bezeichner an, um Sachverhalte in JIRA zu finden. Jira-Sachverhaltbezeichner sind zum Beispiel `ID` oder `Key`.
 - `ProjectDataTypeIdentifierType`: Geben Sie einen Bezeichner an, um Projekte in JIRA zu finden. JIRA-Projektbezeichner sind zum Beispiel `ID`, `Key` oder `Name`.
 - `IssueTypeDataTypeIdentifierType`: Geben Sie einen Bezeichner an, um Sachverhalte in JIRA über die Feldwerte `IssueType` zu finden.
 - `PriorityDataTypeIdentifierType`: Geben Sie einen Bezeichner an, um Projekte in JIRA über die Feldwerte `Priority` zu suchen.
 - `ComponentDataTypeIdentifierType`: Geben Sie einen Bezeichner an, um Projekte in JIRA über die Feldwerte `Component` zu suchen.
 - `VersionDataTypeIdentifierType`: Geben Sie einen Bezeichner an, um Projekte in JIRA über die Feldwerte `Version` zu suchen.
- 8) Geben Sie Filter an, um die Anzahl der Projekte zu begrenzen, die im Editor **Mit Jira-Projekt verknüpfen** angezeigt werden, der im Objektprofil eines entsprechenden Alfabet -Projekts verfügbar ist. Sie können für jeden Filter mehrere Bedingungen angeben. Der Filter kann im Feld **Projektfiler** in den Editoren **Jira-Verbindung für projektbasierte Integrationen** und **Jira-Verbindung für architekturbasierte Integrationen** einer semantischen Jira-Verbindung zugewiesen werden. Dieses Editoren sind in der Funktionalität **Konfiguration von Integrationslösungen** in der Benutzeroberfläche von Alfabet verfügbar. Fügen Sie zu diesem Zweck dem Root-XML-Element `JIRAIntegrationConfig` ein untergeordnetes XML-Element `JIRAProjectFilters` hinzu, und fügen Sie für jede benötigte Filterdefinition ein untergeordnetes XML-Element `Filter` hinzu.
- Legen Sie für das XML-Element `Filter` die folgenden XML-Attribute fest.
 - `Name`: Geben Sie einen Namen für den Filter ein.
 - `Desc`: Geben Sie einen Text mit Informationen über die benötigten Filterkriterien ein. Diese werden im Feld **Beschreibung des Projektfilters** in den Editoren **JIRA-Verbindung für projektbasierte Integrationen erzeugen** und **JIRA-Verbindung für architekturbasierte Integrationen erzeugen** angezeigt, die in der Ansicht **JIRA-Verbindung** in der Funktionalität **Integration Solutions Configuration** verfügbar sind.
 - Fügen Sie dem XML-Element `Filter` ein untergeordnetes XML-Element `Condition` hinzu, und legen Sie das folgende XML-Attribut für das XML-Element `Filter` fest.
 - `PropertyName`: Geben Sie die Eigenschaft des JIRA-Projekts an, die als Filterkriterium verwendet werden soll. Sie können folgende Angaben machen: `Key`, `Name`, `Type`
 - `Operation`: Geben Sie die Operation für die Eigenschaft an. Sie können folgende Operationen angeben: `StartsWith`, `Contains`, `Equals`.

- **Value:** Geben Sie den Wert an, der zur Erfüllung der Suchkriterien verwendet werden soll.
- 9) Geben Sie die Standard-Objektklasseneigenschaften des Typs `String` an, mit denen die relevanten Felder für die Zuordnung von Jira-Eigenschaften in den Editoren **Jira-Verbindung für projektbasierte Integrationen** und **Jira-Verbindung für architekturbasierte Integrationen** ausgefüllt werden sollen. Diese Editoren sind in der Ansicht **Jira-Verbindung** in der Funktionalität **Konfiguration von Integrationslösungen** verfügbar. Benutzerdefinierte Eigenschaften des Typs `String` werden automatisch zu den Editoren hinzugefügt. Fügen Sie zu diesem Zweck für jede benötigte Klassendefinition dem Root-XML-Element `JIRAIntegrationConfig` ein untergeordnetes XML-Element `AlfabetClassMappingStandardPropertySettings` hinzu. Beachten Sie Folgendes:
- Wenn in Ihrem Unternehmen eine projektbasierte Integration verwendet wird, geben Sie für das XML-Element `AlfabetClassMappingStandardPropertySettings` die folgenden XML-Attribute an.
 - **Class:** Geben Sie `Project` ein.
 - **PermittedStandardProperties:** Geben Sie die Standardeigenschaften an, die in den verschiedenen Editorfeldern des Editors **JIRA-Verbindung für projektbasierte Integrationen erzeugen** verfügbar sein sollen. In dem Feld können Sie die Eigenschaft von Alfabet auswählen, die zum Speichern bestimmter Attribute für Projekte in JIRA verwendet wird.
 - Wenn in Ihrem Unternehmen eine architekturbasierte Integration verwendet wird, geben Sie für das XML-Element `AlfabetClassMappingStandardPropertySettings` die folgenden XML-Attribute an. Je nachdem, welches Integrationsmuster angegeben wurde, sind folgende Klassen relevant:
 - **Class:** Geben Sie den Namen der Klasse ein, die für die architekturbasierte Integration relevant ist. Wenn Sie beispielsweise das Integrationsmuster **ICT-Objekt** `>Application >Local Component` auswählen, müssen Sie einen Eintrag `Class` für jede Klasse außer **ICT-Objekt** erstellen.
-  Für die Klasse **ICT-Objekt** kann kein Eintrag erstellt werden, da in JIRA die Werte für `Project`, wie z. B. `ID`, `Key` und `Name`, nur in benutzerdefinierten Eigenschaften gespeichert werden.
- **PermittedStandardProperties:** Geben Sie die Standardeigenschaften an, die in den verschiedenen Editorfeldern des Editors **JIRA-Verbindung für architekturbasierte Integrationen erzeugen** verfügbar sein sollen. In dem Feld können Sie die Eigenschaft von Alfabet auswählen, die zum Speichern bestimmter Attribute für Projekte in JIRA verwendet wird.
- 10) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Konfigurieren der primären und sekundären Berichte

Für jeden Eintrag, der im ADIF-Exportschema erstellt wird, sollte ein primärer konfigurierter Bericht und möglicherweise ein sekundärer konfigurierter Bericht erstellt werden, um die Alfabet -Objekte zu finden, die als Sachverhalte exportiert werden. Sowohl der primäre als auch der sekundäre Bericht müssen vom Typ

Query oder NativeSQL sein. Bitte beachten Sie, dass nur der primäre konfigurierte Bericht obligatorisch ist. Der sekundäre Bericht ist nur erforderlich, wenn andere Sachverhaltfelder exportiert werden müssen, die von n:n-Beziehungen stammen oder untergeordnete Eigenschaften haben.

Der primäre Bericht ermöglicht die Suche nach Alfabet -Objekten wie Anforderungen, Sachverhalten, Funktionen oder Steuerungselementen auf Basis von skalaren Attributen. Der primäre konfigurierte Bericht muss definiert werden, um die JIRA-Sachverhaltfelder aufzufüllen, die nur einen einzelnen skalaren Wert erfordern. Typische Werte wären Summary, Description, Project, Issue Type usw. Der sekundäre Bericht wird verwendet, um die Sachverhaltinformationen mit Referenz-Arrays, String-Arrays usw. zu ergänzen (n:n-Beziehungen). Seine Definition dient daher zum Einpflegen der JIRA-Sachverhaltfelder, die möglicherweise mehr als einen Wert (Arrays) erfordern oder untergeordnete Eigenschaften wie Issue Link oder Attachment enthalten. Zu den Feldern, die mit mehreren Werten gefüllt werden können, gehören beispielsweise Fix Versions, Affects Versions, Components usw. Der sekundäre Bericht ist außerdem erforderlich, um Werte von Feldern mit untergeordneten Eigenschaften anzugeben. Beispielsweise erfordert das Feld Issue Link die Werte für den übergeordneten Sachverhalt und die Art der Beziehung.



Weitere Informationen zum Definieren von konfigurierten Berichten finden Sie im Kapitel *Berichte konfigurieren* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.



Alle Sachverhalte werden erstellt, indem sie Datensatz für Datensatz exportiert werden. Bitte beachten Sie, dass aufgrund der Beschränkungen von JIRA kein Rollback möglich ist.

Beachten Sie die folgenden Informationen zur Konfiguration und Ausführung des Berichts:

- Die für die primären und sekundären Berichte relevanten Bezeichner müssen im XML-Element `JIRADataTypeIdentifierTypes` im XML-Objekt ***JIRAIntegrationConfig*** angegeben werden.
- Es können mehrere primäre und sekundäre Berichte definiert werden. Das Attribut **Kategorie** der Berichte muss definiert werden. Die Spezifikation des XML-Elements `PrimaryAlfabetReportCategory` im XML-Objekt ***JIRAIntegrationConfig*** bestimmt, welche Berichte für eine ausgewählte Datenverbindung im Feld **Primärer Alfabet-Bericht** in der Registerkarte **Ordnen Sie primäre Alfabet-Berichtsspalten den Jira-Sachverhaltfeldern zu** im **ADIF-Exportassistenten** verfügbar sind. Ebenso bestimmt das XML-Element `SecondaryAlfabetReportCategory` die sekundären Berichte, die in der Registerkarte **Ordnen Sie sekundäre Alfabet-Berichtsspalten den JIRA-Sachverhaltfeldern zu** verfügbar sind.
- Mindestens ein primärer konfigurierter Bericht muss als Quelle der Exportdaten definiert werden. Dieser Bericht muss die zu exportierenden Alfabet -Objekte identifizieren.



Die folgende SQL-Abfrage ist ein Beispiel für die Abfrage, die für einen primären Bericht angegeben wird:

```
SELECT j.REFSTR, j.REFSTR AS 'ConnID', (SELECT EXTERNALID
FROM INTEGRATIONCONNECTIONUSAGE WHERE A_OBJECT = ftr.REFSTR)
AS 'ID', ftr.REFSTR AS 'Object', ftr.DESCRPTION, ftr.STATUS
'Status', 'New Feature' 'IssueType', ftr.NAME, 'Customer' AS
'User', icto.SC_JIRAKEY 'Project'

FROM FEATURE ftr, APPLICATION app, ICTOBJECT icto,
JIRA_DBCONNECTION j, INTEGRATIONCONNECTIONUSAGE ictou

WHERE ftr.OBJECT = app.REFST

AND app.ICTOBJECT = icto.REFSTR

AND j.NAME = 'ICT-App-LCom'
```

```
AND ictou.CONNECTION = j.REFSTR
AND ictou.A_OBJECT = icto.REFSTR
```

Die Ausgabe des Berichts pflegt JIRA-Sachverhaltfelder ein, die nur einen einzelnen skalaren Wert erfordern, z. B. `Summary`, `Description`, `Project`, `Issue Type` usw. Der primäre konfigurierte Bericht muss folgende Spalten ausgeben:

- **Alfabet -Integrationsverbindungs-ID:** Der `REFSTR` der für den Export verwendeten JIRA-Datenbankverbindung.
- **Alfabet -Objekt-ID:** Der `REFSTR`-Wert des Alfabet -Objekts, das als JIRA-Sachverhalt exportiert und zugeordnet werden soll.
- **JIRA-Project ID oder Key:** Die ID oder der Key des JIRA-Projekts muss im Bericht enthalten sein. Sie können sie bzw. ihn entweder als Zeichenfolge angeben oder in einer benutzerdefinierten Eigenschaft einer JIRA-Projekten zugeordneten Alfabet -Objektklasse erfassen und bei der Definition des Berichts aus dieser Objektklasse lesen.



Ob ID oder Key angegeben wird, hängt von der Spezifikation im XML-Element `JIRA-DataTypeIdentifierTypes` im XML-Objekt ***JIRAIntegrationConfig*** ab. Dies gilt für die Spalten `Project ID oder Key`, `Issue ID oder Key` und `Jira Issue Type ID oder Key`.

- **JIRA-Issue ID oder Key:** Beachten Sie, dass dies eine optionale Spalte ist. Wenn ein Wert für `Issue ID oder Key` angegeben wird, wird der entsprechende JIRA-Sachverhalt mit den neuen Werten aktualisiert. Wenn kein Wert für `Issue ID oder Key` angegeben wird, wird in JIRA ein neuer Sachverhalt für diesen Datensatz erstellt. Bei Sachverhalten, die bereits aus Alfabet exportiert wurden, kann die `JIRA-Issue ID` aus der Objektklasseneigenschaft `ExternalID` der Objektklasse `IntegrationConnectionUsage` gelesen werden.
- `Jira Issue Type ID oder Key`
- Weitere relevante Felder mit Einzelwerten in JIRA, z. B. `Summary`, `Description`, `Assignee` usw.
- Es muss ein sekundärer Konfigurationsbericht definiert werden, um die Referenzen zu finden und die JIRA-Sachverhaltfelder einzupflegen, die nicht skalar sind und möglicherweise mehr als einen Wert (Arrays) erfordern oder untergeordnete Eigenschaften haben, wie z. B. `Issue Link oder Attachment`. Dazu gehören beispielsweise `Fix Versions`, `Affects Versions`, `Components` usw.



- Die Namen der für die `Issue Field`-Spalten in JIRA relevanten sekundären Berichtspalten müssen im XML-Element `JIRADatatypeIdentifierTypes` im XML-Objekt ***JIRAIntegrationConfig*** angegeben werden. Beim Export werden nur diese Spalten eingepflegt, und alle anderen Spalten werden ignoriert. Beachten Sie Folgendes:
 - Alle nicht skalaren Felder (mit untergeordneten Eigenschaften) und Felder mit Arrays, die Teil des Sachverhaltschemas sind, können über den sekundären Bericht exportiert werden.
 - Um Sachverhaltfelder des Typs `Array` zu exportieren, müssen Anwender ihre Werte über 3 Spalten im Bericht angeben: eine Spalte für `Issue Field`

ID, eine Spalte für `Issue Field Name` und eine Spalte für `Issue Field Value`.

- Um Sachverhaltfelder mit untergeordneten Eigenschaften, wie z. B. Sachverhaltverknüpfung und Anlage, fehlerfrei zu exportieren, müssen Anwender Berichtsspalten für jede einzelne untergeordnete Eigenschaft der entsprechenden Spalte `Issue Field` zuordnen.
- Es ist eine einzigartige Kombination aus Alfabet -Integrationsverbindungs-ID, JIRA-Project ID und Alfabet -Objekt-ID erforderlich, wenn die JIRA-Issue ID nicht von den primären oder sekundären Berichten genannt oder gekannt wird. Wenn mehrere Sachverhalte dieselbe `Issue ID` und Alfabet -Integrationsverbindungs-ID oder dieselbe Kombination aus Alfabet -Integrationsverbindungs-ID, JIRA-Project ID und Alfabet -Objekt-ID aufweisen, überschreibt der zuletzt exportierte ID-Wert die anderen ID-Werte und wird in die `IntegrationConnectionUsage`-Datenbanktabelle geschrieben. Es werden jedoch nur die im primären Bericht angegebenen Attribute überschrieben, und die Aktualisierung kann daher kumulativ sein. Beachten Sie bezüglich des sekundären Berichts Folgendes:
 - Wenn die Kombination aus Alfabet -Integrationsverbindungs-ID, JIRA-Project ID und Alfabet -Objekt-ID für mehrere Zeilen gleich ist (und `Issue ID/Key` leer ist), gelten alle Zeilen für einen Sachverhalt, der neu erstellt wird.
 - Wenn die Kombination aus Alfabet -Integrationsverbindungs-ID, JIRA-Project ID und Alfabet -Objekt-ID für mehrere Zeilen gleich ist (und `Issue ID/Key` nicht leer ist), gelten alle Zeilen für denselben vorhandenen Sachverhalt, der aktualisiert wird.
- Eine Alfabet-Eigenschaft des Typs `StringArray` kann einer JIRA-Spalte `Issue Field` zugeordnet werden, wenn ein Mehrfachauswahlwert festgelegt werden soll. Jeder Wert des `String-Arrays` muss als einzelne Option in der entsprechenden JIRA-Spalte `Issue Field` verfügbar sein.

Der sekundäre konfigurierte Bericht sollte die folgenden Spalten ausgeben:

- **Alfabet -Objekt-ID:** Der `REFSTR`-Wert des Alfabet -Objekts, das als JIRA-Sachverhalt exportiert und zugeordnet werden soll.
- **JIRA-Issue field ID oder Issue Field Name:** Bei benutzerdefinierten Feldern sollte die `Issue field ID` angegeben werden, da die Namen für benutzerdefinierte Felder möglicherweise nicht eindeutig sind. Standardfelder können entweder über `Issue Field ID` oder `Issue Field Name` identifiziert werden.
- **JIRA-Issue Field Value:** Es wird eine neue Zeile für jedes Feld erstellt, das untergeordnete Eigenschaften enthält, z. B. `Issue Link` oder `Attachment`. Beispielsweise werden für ein Feld `Issue Link` die Spalten `Issue Link Outward Issue` und `Issue Link Relation` der Standardliste der Spalten mit Sachverhaltfeldern hinzugefügt.
- Beachten Sie folgende zusätzliche Konfigurationsanforderungen bezüglich des Exports von Web-Links und Dokumenten.
 - So exportieren Sie Web-Links von Alfabet nach JIRA.

- 1) Vergewissern Sie sich in JIRA, dass ein benutzerdefiniertes Feld vom Typ `Text Field (multi-line)` verfügbar ist. Wenn dies nicht der Fall ist, muss ein benutzerdefiniertes Feld des Typs `Text Field (multi-line)` erstellt werden.
 - 2) Navigieren Sie zur Feldkonfiguration für das benutzerdefinierte Feld.
 - 3) Navigieren Sie zu dem Feld, und klicken Sie auf `Renderers`.
 - 4) Wählen Sie `Wiki Style Renderer` aus.
 - 5) Geben Sie im `Alfabet`-Bericht den Wert für die Web-Links an, und verwenden Sie dabei das Format `[neuer Link|http://beispiel.com]`.
- Zum Exportieren von Dokumenten nach JIRA.
 - 1) `23540`JIRA kann keine eindeutigen Bezeichner für angehängte Dokumente bereitstellen. Wenn dieselbe Anlage für einen Sachverhalt erneut gesendet wird, enthält der Sachverhalt daher mehrmals das gleiche Dokument. Dies ist eine Beschränkung von JIRA, und es wird empfohlen, Dokumente nur dann zu exportieren, wenn es unumgänglich ist. Der konfigurierte Bericht sollte angeben, dass Dokumente dem Ergebnisdatensatz nur dann hinzugefügt werden, wenn die Eigenschaften `CREATION_DATE` und/oder `CHANGE_DATE` der Klasse `ALFA_IDOCUMENT` ein Datum ausgeben, das später als das Ausführungsdatum der letzten JIRA-Sachverhaltsynchronisierung liegt.
 - 2) Im konfigurierten Bericht muss eine Datensatzanweisung `RetrieveIDOCPath` definiert werden, um den Export von Anlagen in den Laufzeitordner von `Alfabet`-Webapplikation auszulösen. Die Konfiguration der Datensatzanweisung wird im Abschnitt *Exportieren von Anlagen in einen Runtime-Ordner während der Berichtsausführung* im Kapitel *Definieren von Abfragen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* beschrieben.
 - Wenn Probleme in JIRA auf Grundlage eines `Alfabet`-Exports erstellt werden, wird die bzw. der für den Export verwendete `Issue ID` bzw. `Issue Key` in der Datenbanktabelle `IntegrationConnectionUsage` aktualisiert. Wenn die JIRA-`Issue ID` nicht durch die primären oder sekundären Berichte genannt oder gekannt wird, ist eine einmalige Kombination aus `Alfabet`-Integrationsverbindungs-ID, `JIRA-Project ID` und `Alfabet`-Objekt-ID erforderlich. Wenn mehrere Sachverhalte dieselbe Sachverhalt-ID und `Alfabet`-Integrationsverbindungs-ID oder dieselbe Kombination aus `Alfabet`-Integrationsverbindungs-ID, `JIRA-Project ID` und `Alfabet`-Objekt-ID aufweisen, überschreibt der zuletzt exportierte `ID`-Wert die anderen `ID`-Werte. Es werden jedoch nur die im primären Bericht angegebenen Attribute überschrieben, und die Aktualisierung kann daher kumulativ sein.
 - Wenn die Kombination aus `Alfabet`-Integrationsverbindungs-ID, `JIRA-Project ID` und `Alfabet`-Objekt-ID für mehrere Zeilen gleich ist (und `Issue ID/Key` leer ist), gelten alle Zeilen für einen Sachverhalt, der neu erstellt wird.
 - Wenn die Kombination aus `Alfabet`-Integrationsverbindungs-ID, `JIRA-Project ID` und `Alfabet`-Objekt-ID für mehrere Zeilen gleich ist (und `Issue ID/Key` nicht leer ist), gelten alle Zeilen für denselben vorhandenen Sachverhalt, der aktualisiert wird.
 - Die Berichtsdefinitionen werden intern als Teil des ADIF-Schemas gespeichert. Bei einer Aktualisierung des primären oder sekundären konfigurierten Berichts muss das ADIF-Schema aktualisiert werden. Zu diesem Zweck muss das ADIF-Schema über den **ADIF-Exportassistenten** für Änderungen geöffnet, im Feld **Wählen Sie einen zu bearbeitenden Eintrag** der entsprechende Eintrag ausgewählt und die Schaltfläche **Bearbeiten** sowie anschließend die Schaltfläche **Temporären Eintrag speichern** angeklickt werden. Gehen Sie zur zweiten Registerkarte, wählen Sie einen entsprechenden Bericht im Feld **Wählen Sie einen sekundären Bericht aus, um die**

Zuordnung zu definieren, und klicken Sie auf die Schaltfläche **Bearbeiten** sowie anschließend auf die Schaltfläche **Temporären Eintrag speichern**. Wiederholen Sie diesen Vorgang für jeden sekundären Bericht, der gespeichert werden muss. Wenn Sie fertig sind, klicken Sie auf die Schaltfläche **OK**, um die Änderungen an Alfabet-Datenbank zu speichern und den **ADIF-Exportassistenten** zu schließen.

Konfigurieren von Objektfilterberichten

Optional können Sie einen Bericht konfigurieren, um die Alfabet -Objekte zu finden, die in Jira-Projekte integriert werden dürfen. Die Objekte sind in der Regel über die Stereotypspezifikation im Feld **Stereotypfilter** der Editoren **Jira-Verbindung für projektbasierte Integrationen** und **Jira-Verbindung für architekturbasierte Integrationen** zu finden, die in der Funktionalität **Konfiguration von Integrationslösungen** in der Ansicht **JIRA-Verbindung** verfügbar sind. Konfigurierte Berichte sind jedoch ein zusätzlicher Mechanismus, um Objekte zu finden, die in Jira-Projekte, Jira-Projektversionen und Jira-Projektkomponenten integriert werden können.

Die konfigurierten Berichte können im Feld **Objektfilterbericht** der Editoren **Jira-Verbindung für projektbasierte Integrationen** und **Jira-Verbindung für architekturbasierte Integrationen** einer semantischen Jira-Verbindung zugeordnet werden. Die Editor-Option **Mit Jira-Projekt verknüpfen** ist im Menü **Aktion** in den Objektprofilen der Objekte verfügbar, die über die Definition des Felds **Stereotypfilter** oder des Felds **Objektfilterbericht** gefunden werden.

Beachten Sie bezüglich des konfigurierten Berichts Folgendes:

- Der Bericht muss in der ersten Spalte der entsprechenden Projekte eine Liste von Alfabet -REFSTR zurückgeben, die in Jira-Projekte integriert werden dürfen.
- Der REFSTR muss für dieselbe Klasse sein, für die das Feld **Objektfilterbericht** definiert ist.



Wenn ein Alfabet -Objekt mit Jira synchronisiert wird, werden die Objekte nur dann synchronisiert, wenn sie im Rückgabesatz des zugehörigen, im Feld **Objektfilterbericht** definierten Berichts oder im zurückgegebenen Satz des im Feld **Stereotypfilter** angegebenen Objektstereotyps enthalten sind. Beispielsweise wurde ein Bericht, der als Objektfilter für die Klasse „Applikation“ konfiguriert wurde, so konfiguriert, dass nur Applikationen zurückgegeben werden, deren Status auf „Genehmigt“ gesetzt ist. Daher darf eine neue Applikation mit dem Status „Entwurf“ nicht zum Erzeugen einer neuen Version im zugehörigen Jira-Projekt verwendet werden. In diesem Fall ist die Editoroption **Mit Jira-Projekt verknüpfen** im Menü **Aktion** im Objektprofil der neuen Applikation mit dem Status „Entwurf“ verfügbar.

- Das Attribut **Kategorie** des konfigurierten Berichts muss auf `JIRAIIntegration` gesetzt sein. Beachten Sie, dass im Feld **Objektfilterbericht** nur die konfigurierten Berichte angezeigt werden, für die das Attribut **Kategorie** auf `JIRAIIntegration` festgelegt wurde.



Weitere Informationen zum Definieren von konfigurierten Berichten finden Sie im Kapitel *Berichte konfigurieren* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Konfigurieren des ADIF-Exportassistenten für den Export von Alfabet-Daten nach JIRA

Nachdem die Struktur von Alfabet und die JIRA-Projektstruktur zugeordnet und verknüpft wurden, müssen Sie den Export von Problemen nach JIRA auf Grundlage der primären und sekundären konfigurierten Berichte und der Konfiguration eines entsprechenden ADIF-Schemas konfigurieren. Dazu müssen Sie die Felder in JIRA zuordnen, um Informationen über Sachverhalte in den entsprechenden Spalten der konfigurierten Berichte zu erfassen. Mit dem primären Konfigurationsbericht können Sie skalare JIRA-Sachverhaltfelder mit einem einzelnen Wert erfassen, und mit dem sekundären Bericht können Sie Felder erfassen, die mehr als einen Wert (Arrays) erfordern oder untergeordnete Eigenschaften wie z. B. `Issue Link` oder `Attachment` haben. Nachdem die Zuordnung abgeschlossen wurde, kann das ADIF-Exportschema über einen Batchauftrag ausgeführt werden, und die relevanten Alfabet -Objekte wie Anforderungen, Funktionen usw. werden als Sachverhalte nach JIRA exportiert.

Wenn Sie den **JIRA-Exportassistenten** definieren, muss der Datensatz im Assistenten zunächst mit den entsprechenden Sachverhaltfeldern ausgefüllt werden, die in JIRA zur Erfassung von Informationen über die Sachverhalte verwendet werden. Die verschiedenen Sachverhaltfelder müssen den entsprechenden Spalten des konfigurierten Berichts zugeordnet werden. Dazu müssen Sie für das Schema einen Abruf definieren, der die Sachverhaltfelder in den Datensatz einpflegt. Wenn durch das Schema nicht alle Sachverhaltfelder abgerufen werden, können Sie die restlichen Sachverhaltfelder manuell hinzufügen. Der Abruf des Schemas erfordert die folgende Definition:

- Eine Datenverbindung zum Abrufen der Informationen von der JIRA-Instanz
- Ein JIRA-Projekt, in dem vorhandene Sachverhalte definiert sind. Beachten Sie, dass dies nicht unbedingt dem eigentlichen Projekt entsprechen muss, dem die Sachverhalte zugewiesen werden.
- Der Sachverhaltstyp, für den Sie die Sachverhaltfelder zuordnen möchten.



Die JIRA-Zielinstanz muss mindestens einen Sachverhalt für jeden Problemtyp im ausgewählten Projekt haben, damit die Schema-Informationen abgerufen werden können. Wenn ein Sachverhaltstyp ausgewählt wird, für den im ausgewählten Projekt kein Sachverhalt vorliegt, können die Sachverhaltfelder nicht abgerufen werden, und der Datensatz wird nicht eingepflegt.



Der JIRA-Export beruht auf dem Prinzip der Konsistenz zwischen JIRA-Benutzeroberfläche und JIRA-API. Die JIRA-Benutzeroberfläche und die JIRA-API können jedoch verschiedene Formate haben. Dies ist zum Beispiel bei Datumsfeldern der Fall. Bitte beachten Sie, dass sich Anwender bei der Konfiguration des ADIF-Exportschemas nicht ausschließlich auf die Filterfelder in der JIRA-Benutzeroberfläche verlassen sollten, sondern wissen müssen, welches Format die JIRA-API erwartet, um das ADIF-Exportschema entsprechend zu konfigurieren. Wenn die Formate inkonsistent sind, bringt der Export von Alfabet nach JIRA möglicherweise nicht die richtigen Ergebnisse.

Jeder Sachverhalt ist mit einem Projekt verknüpft. Alfabet kann Objekte (Anforderungen) haben, die mehreren Projekten zugeordnet sind.

- 1) Erstellen Sie ein ADIF-Exportschema mit mindestens einem Eintrag. Bitte beachten Sie, dass die Daten über das gleiche ADIF-Importschema in alle definierten JIRA-Instanzen exportiert werden. Weitere Informationen dazu, wie Sie das ADIF-Exportschema und den Eintrag erstellen, finden Sie im Referenzhandbuch *Alfabet-Datenintegrationsframework*.
- 2) Setzen Sie das Attribut **Alfabet-Anwenderschnittstellenverhalten** des Exportschemas auf `VisibleExecutable`, und klicken Sie auf die Schaltfläche **Speichern**.

- 3) Klicken Sie mit der rechten Maustaste auf das ADIF-Exportschema, und wählen Sie **Erzeugen der ADIF-Schema-Details mithilfe von JIRA-Exportassistent**. Der **JIRA-Exportassistent** wird geöffnet. Der **JIRA-Exportassistent** verfügt über zwei Registerkarten. Zunächst muss die Registerkarte **Ordnen Sie primäre Alfabet-Berichtsspalten den JIRA-Problemfeldern zu** definiert werden, um den primären Bericht auszulösen und den Datensatz im Editor mit den skalaren JIRA-Sachverhaltfeldern zu füllen, die einen einzelnen Wert erfordern. Anschließend kann die Registerkarte **Ordnen Sie sekundäre Alfabet-Berichtsspalten den JIRA-Problemfeldern zu** definiert werden, um die JIRA-Sachverhaltfelder anzugeben, die möglicherweise mehr als einen Wert (Arrays) erfordern. Nähere Informationen über das Konfigurieren der erforderlichen Berichte finden Sie im Abschnitt [Konfigurieren der primären und sekundären Berichte](#).
- 4) Machen Sie in der Registerkarte **Ordnen Sie primäre Alfabet-Berichtsspalten den JIRA-Problemfeldern zu** folgende Angaben:

- **Namen des Exporteintrags eingeben:** Zur Erstellung eines neuen Exporteintrags geben Sie den Namen des Eintrags ein, der für das ADIF-Exportschema definiert wurde.



Alternativ können Sie einen vorhandenen Exporteintrag bearbeiten, indem Sie im Feld **Wählen Sie einen zu bearbeitenden Eintrag** den Exporteintrag auswählen, den Sie bearbeiten möchten. Die Assistentenfelder werden mit der vorhandenen Definition aktualisiert. Klicken Sie auf die Schaltfläche **Bearbeiten**, um die im Folgenden beschriebenen Felder zu bearbeiten.

- **Datenverbindung:** Wählen Sie die entsprechende Datenverbindung aus, die zur Verbindung mit der JIRA-Instanz verwendet werden soll.
- **Jira-Projekt für Schemaabruf:** Wählen Sie ein repräsentatives Projekt aus, um das Sachverhaltschema abzurufen. Dieses Projekt muss nicht das Projekt sein, an das der Sachverhalt gesendet werden soll. Es werden alle in der JIRA-Instanz verfügbaren Projekte angezeigt, die Ziel der Verbindung sind.
- **Jira-Sachverhaltstyp für Schemaabruf:** Wählen Sie einen repräsentativen Sachverhaltstyp aus, um Sachverhalte abzurufen.



Sie müssen ein Projekt und einen Sachverhaltstyp auswählen, für das/den mindestens ein Sachverhalt abgerufen werden kann und das/der eine JIRA-Sachverhaltstruktur aufweist, die für den vorgesehenen Export repräsentativ ist. Die Zuordnungstabelle wird mit den für die gefundenen Sachverhalte verfügbaren Feldern vorausgefüllt. Alle Felder, die nicht im Zuge des Schema-Abrufs zurückgegeben werden, können einzeln hinzugefügt werden. Wenn ein Sachverhaltstyp ausgewählt wird, für den im ausgewählten Projekt kein Sachverhalt vorliegt, können die Sachverhaltfelder nicht abgerufen werden, und der Datensatz wird nicht eingepflegt. Außerdem wird in diesem Fall eine Fehlermeldung angezeigt.

- **Primärer Alfabet-Bericht:** Geben Sie den primären konfigurierten Bericht zur Erfassung von skalaren JIRA-Sachverhaltfeldern mit einem einzigen Wert an. Es werden alle Berichte, die der für die ausgewählte Datenverbindung im XML-Objekt **JIRAIntegrationConfig** definierten Berichtskategorie zugewiesen sind, in der Dropdown-Liste angezeigt.
- **Anzeigeoption für Alfabet-Berichtseigenschaften:** Wählen Sie eine der folgenden Sortierungsoptionen für die im Datensatz angezeigten Daten:
 - **Berichtsspalten wie im Bericht definiert sortieren:** Die Spaltennamen im konfigurierten Bericht werden anstelle der Namen der Objekteigenschaftsklassen angezeigt. Die Sortierreihenfolge ist identisch mit der Reihenfolge der Spalten im konfigurierten Bericht.

- **Berichtsspalten mit lexikografischer Sortierung verwenden:** Die Spaltennamen im konfigurierten Bericht werden anstelle der Namen der Objekteigenschaftsklassen angezeigt. Die Sortierreihenfolge ist alphanumerisch.
 - **Eigenschaftsspalten wie im Bericht definiert sortieren:** Die Objekteigenschaftsklassen werden als <ObjectClassName>.<PropertyName> angezeigt. Die Sortierreihenfolge ist identisch mit der Reihenfolge der Spalten im konfigurierten Bericht.
 - **Eigenschaftsspalten mit lexikographischer Sortierung sortieren:** Die Objekteigenschaftsklassen werden als <ObjectClassName>.<PropertyName> angezeigt. Die Sortierreihenfolge ist alphanumerisch.
- 5) Klicken Sie auf die Schaltfläche **Temporären Eintrag speichern**. Im Abschnitt **Ordnen Sie JIRA-Sachverhaltfelder den Berichtsspalten zu** werden alle JIRA-Sachverhaltfelder angezeigt, die über die Definition des Exporteintrags gefunden wurden. Wenn Sie weitere JIRA-Sachverhaltfelder hinzufügen möchten, klicken Sie auf die Schaltfläche **Weitere Felder hinzufügen**, und wählen Sie in der Auswahl **JIRA-Felder** die entsprechenden Felder. Klicken Sie auf **OK**, um dem Datensatz die Felder hinzuzufügen.
- **JIRA-Sachverhaltfeld-ID:** Zeigt die ID des JIRA-Sachverhaltfelds an.
 - **JIRA-Sachverhaltfeldname:** Zeigt den Namen des JIRA-Sachverhaltfelds an. Der Datensatz zeigt, welche JIRA-Felder zum Erstellen und Aktualisieren von Problemen obligatorisch sind. Alle Felder, die obligatorisch sind, werden in Rot angezeigt.
 - **JIRA-Sachverhaltfeldtyp:** Zeigt den Typ des JIRA-Sachverhaltfelds an.
 - ***C:** Zeigt ein **M** für Obligatorisch („Mandatory“) an, wenn beim Export das Feld **Name der Berichtsspalte** zugeordnet werden muss, um in JIRA einen Sachverhalt zu erstellen.
 - ***U:** Zeigt ein **M** für Obligatorisch („Mandatory“) an, wenn beim Export das Feld **Name der Berichtsspalte** zugeordnet werden muss, um in JIRA einen Sachverhalt zu aktualisieren.
 - **Name der Berichtsspalte:** Wählen Sie die Alfabet -Eigenschaft aus, der das JIRA-Sachverhaltfeld zugeordnet werden soll. Einige Zeilen können erst nach der Ausführung des primären Berichts angegeben werden.



Bitte beachten Sie Folgendes hinsichtlich des Sachverhaltfelds `Status` in JIRA:

- JIRA-Sachverhaltfelder können nicht aktualisiert werden, wenn der Wert `Status` für den JIRA-Sachverhalt nicht korrekt festgelegt werden kann.
- Um einen Sachverhalt in JIRA auf Grundlage eines Alfabet -Objekts zu erstellen, muss der neue Sachverhalt entweder dem für den Sachverhaltstyp zulässigen Anfangsstatus oder dem Status, der dem Anfangsstatus folgt, zugewiesen werden. Beachten Sie Folgendes:
 - Wenn der Alfabet -Status nicht mit dem anfänglichen JIRA-Status identisch ist, prüft das System, ob ein Statusübergang zwischen den beiden Statuswerten möglich ist.
 - Wenn der Statusübergang nicht möglich ist, wird der neue Sachverhalt gelöscht.

- Wenn der Statusübergang möglich ist, wird der Sachverhalt erstellt, und die Sachverhaltfelder, die im aktuellen Status des Alfabet -Objekts vor dem Statusübergang aktualisiert werden können.
 - Das System prüft dann, ob der aktuelle JIRA-Status einen Übergang zu sich selbst hat. Wenn der Statusübergang möglich ist, wird er durchgeführt, und die Sachverhaltfelder, die im aktuellen JIRA-Sachverhalt aktualisiert werden können, werden aktualisiert.
 - Um ein Problem in JIRA auf Grundlage eines Alfabet -Objekts zu aktualisieren, prüft das System, ob der Alfabet -Status mit dem JIRA-Status identisch ist. Beachten Sie Folgendes:
 - Wenn der Alfabet -Status nicht mit dem JIRA-Status identisch ist, prüft das System, ob ein Statusübergang zwischen den beiden Statuswerten möglich ist.
 - Wenn der Statusübergang nicht möglich ist, wird eine Fehlermeldung angezeigt, und die Aktualisierung des Sachverhalts wird übersprungen.
 - Wenn der Statusübergang möglich ist, wird die Aktualisierung für die Sachverhaltfelder durchgeführt, die im aktuellen Status des Alfabet -Objekts vor dem Statusübergang aktualisiert werden können.
 - Das System prüft dann, ob der aktuelle JIRA-Status einen Übergang zu sich selbst hat. Wenn der Statusübergang möglich ist, wird er durchgeführt, und die Sachverhaltfelder, die im aktuellen JIRA-Sachverhalt aktualisiert werden können, werden aktualisiert.
- 6) Klicken Sie auf die Schaltfläche **Temporären Eintrag speichern**. Eine Meldung wird angezeigt, wenn der Exporteintrag gespeichert wurde.
- 7) Gehen Sie zur Registerkarte **Ordnen Sie sekundäre Alfabet-Berichtsspalten den JIRA-Problemfeldern zu**, und machen Sie folgende Zuordnungsangaben für mehrere Werte umfassende Attribute, die mehr als einen Wert (Arrays) erfordern:
- **Namen des Exporteintrags eingeben:** Zeigt den Namen des in der Registerkarte ausgewählten Exporteintrags oder den für einen neuen Eintrag eingegebenen Namen an.
 - **Master-Gruppe für sekundäre Alfabet-Berichte:** Geben Sie die sekundären konfigurierten Berichte an, um mehrere Werte umfassende Attribute zu erfassen. Es werden alle Berichte, die für die ausgewählte Datenverbindung im XML-Objekt **JIRAIntegrationConfig** definierten Berichtskategorie zugewiesen sind, in der Dropdown-Liste angezeigt.
-  Alternativ dazu können Sie im Feld **Wählen Sie einen sekundären Bericht aus, um die Zuordnung zu definieren** die sekundären Berichte auswählen, die Sie bearbeiten möchten, wenn Sie die vorhandene Definition auf Grundlage eines oder mehrerer Bericht(e) bearbeiten möchten. Die Assistentenfelder werden mit der vorhandenen Definition aktualisiert. Klicken Sie auf die Schaltfläche **Bearbeiten**, um die im Folgenden beschriebenen Felder zu bearbeiten.
- **Sekundärer Bericht, um die Zuordnung zu definieren:** Wählen Sie nötigenfalls zusätzliche sekundäre Berichte aus, und führen Sie sie aus, um mehrere Werte umfassende Attribute zu finden.

- **Anzeigeoption für Alphabet-Berichteigenschaften:** Wählen Sie eine der folgenden Sortierungsoptionen für die im Datensatz angezeigten Daten:
 - **Berichtsspalten wie im Bericht definiert sortieren:** Die Spaltennamen im konfigurierten Bericht werden anstelle der Namen der Objekteigenschaftsklassen angezeigt. Die Sortierreihenfolge ist identisch mit der Reihenfolge der Spalten im konfigurierten Bericht.
 - **Berichtsspalten mit lexikografischer Sortierung verwenden:** Die Spaltennamen im konfigurierten Bericht werden anstelle der Namen der Objekteigenschaftsklassen angezeigt. Die Sortierreihenfolge ist alphanumerisch.
 - **Eigenschaftsspalten wie im Bericht definiert sortieren:** Die Objekteigenschaftsklassen werden als <ObjectClassName>.<PropertyName> angezeigt. Die Sortierreihenfolge ist identisch mit der Reihenfolge der Spalten im konfigurierten Bericht.
 - **Eigenschaftsspalten mit lexikographischer Sortierung sortieren:** Die Objekteigenschaftsklassen werden als <ObjectClassName>.<PropertyName> angezeigt. Die Sortierreihenfolge ist alphanumerisch.
- 8) Klicken Sie auf die Schaltfläche **Temporären Eintrag speichern**. Im Abschnitt **Ordnen Sie sekundäre Alphabet-Berichtsspalten den JIRA-Problemfeldern zu** werden alle über die sekundären Berichte gefundenen JIRA-Sachverhaltfelder angezeigt.
- **JIRA-Sachverhaltfeldname:** Zeigt den Namen des JIRA-Sachverhaltfelds an. Der Datensatz zeigt, welche JIRA-Felder zum Erstellen und Aktualisieren von Problemen obligatorisch sind. Alle Felder, die obligatorisch sind, werden in Rot angezeigt.
 - **JIRA-Sachverhaltfeldtyp:** Zeigt den Typ des JIRA-Sachverhaltfelds an.
 - ***C:** Zeigt ein M für Obligatorisch („Mandatory“) an, wenn beim Export das Feld **Name der Berichtsspalte** zugeordnet werden muss, um in JIRA einen Sachverhalt zu erstellen.
 - ***U:** Zeigt ein M für Obligatorisch („Mandatory“) an, wenn beim Export das Feld **Name der Berichtsspalte** zugeordnet werden muss, um in JIRA einen Sachverhalt zu aktualisieren.
 - **Name der Berichtsspalte:** Wählen Sie die Alphabet -Eigenschaft aus, der das JIRA-Sachverhaltfeld zugeordnet werden soll. Einige Zeilen können erst nach der Ausführung des sekundären Berichts angegeben werden.
- 9) Klicken Sie auf die Schaltfläche **Temporären Eintrag speichern**.
- 10) Klicken Sie zum Schließen des Editors auf **OK**. Das ADIF-Exportschema kann über einen Batchauftrag ausgeführt werden, und die relevanten Alphabet -Objekte wie Anforderungen, Funktionen usw. werden als Sachverhalte nach JIRA exportiert.

Konfigurieren der Ereignisvorlagen zum Auslösen der ADIF-Exportschemata für die Synchronisierung

Nach Abschluss der Zuordnung müssen Ereignisvorlagen so konfiguriert werden, dass sie die ADIF-Exportschemata auslösen, die den entsprechenden konfigurierten Berichten zugeordnet sind. Der konfigurierte Bericht, der dem ADIF-Exportschema zugeordnet ist, sollte einen Satz von Objekten zurückgeben, die Jira-Sachverhalten zugeordnet werden können. Die von der Ereignisvorlage zurückgegebenen Menge an Objekten muss eine Teilmenge der Objekte sein, die vom primären Bericht des ADIF-Prozesses zurückgegeben werden, auf den die Ereignisvorlage verweist. Nur die zurückgegebenen Objekte, die sowohl der Ereignisvorlage als auch dem primären Bericht gemeinsam sind, werden in Jira exportiert. Die Ausführung der

Ereignisvorlage kann wie nachfolgend beschrieben parametrisiert werden, um die Anzahl der zurückgegebenen Objekte zu begrenzen.

Die Ereignisvorlagen können im Feld **Ereignisvorlagen für die Synchronisierung** der Editoren **Jira-Verbindung für projektbasierte Integrationen** und **Jira-Verbindung für architekturbasierte Integrationen** einer semantischen Jira-Verbindung zugeordnet werden. Die Ereignisvorlage wird ausgelöst, wenn die Anwender die Funktionalität **Mit Projektdaten synchronisieren** in den entsprechenden Objektprofilen ausführen.

Beachten Sie Folgendes bezüglich der Konfiguration von Ereignisvorlagen für die ADIF-Exportschemata, die für die Jira-Integration konfiguriert sind:

- Für die Berichte, die für das ADIF-Exportschema angegeben sind, ist kein Parameter erforderlich. Wenn ein Parameter angegeben ist, wird er ignoriert.
- Für die Ereignisvorlage müssen die folgenden Attribute definiert sein:
 - **ADIF-Schema:** Wählen Sie das ADIF-Exportschema aus, das die Ereignisvorlage auslösen soll.
 - **Werte für Variablen über Abfrage:** Geben Sie eine Abfrage an, um die vom Bericht zurückgegebenen Objekte zu begrenzen. Wenn der Anwender beispielsweise eine Applikation mit einer Jira-Projektversion synchronisiert, werden nur die für die zu synchronisierende Applikation relevanten Funktionen in Jira exportiert. Beachten Sie, dass der REFSTR den Namen `ObjectRef` wie in der Beispielabfrage gezeigt haben muss:

```
SELECT NULL; REFSTRA AS 'ObjectRef'
FROM FEATURE
WHERE OBJECT = @BASE
```



Für den Import aus Jira sollte die Ereignisvorlage einen Satz von Objekten zurückgeben, die mit den relevanten Jira-Strukturobjekten verknüpft sind.

Informationen zur Konfiguration von Ereignisvorlagen finden Sie im Kapitel *Konfigurieren von Ereignissen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*. Ein Anwender mit einem administrativen Anwenderprofil kann den Erfolg der ausgelösten Ereignisse in der Funktionalität *Ereignisverwaltung* überprüfen. Informationen hierzu finden Sie im Abschnitt *Verwalten von Ereignissen* im Referenzhandbuch *Anwender- und Lösungsverwaltung*.

Konfigurieren von semantischen Verbindungen zum Verknüpfen und Synchronisieren von Jira-Projekten mit Alfabet-Objekten

Die Ansicht **Jira-Verbindung** in der Funktionalität **Konfiguration von Integrationslösungen** in der Alfabet -Benutzeroberfläche ermöglicht es Ihnen, eine oder mehrere semantische Definitionen für alle relevanten Jira@-Verbindungen zu erstellen, die im XML-Objekt **JIRAIntegrationConfig** konfiguriert wurden, um Alfabet -Daten in Jira zu exportieren. Für jede im XML-Objekt **JIRAIntegrationConfig** konfigurierte Datenverbindung sollte eine JIRA-Verbindungsdefinition erstellt werden. Sie können mehrere semantische Datenverbindungen für jede Datenverbindung konfigurieren, die im XML-Element **DataConnection** des XML-Objekts **JIRAIntegrationConfig** konfiguriert ist.

Über eine semantische Jira-Verbindung können Sie die Alfabet -Objektklassen der Jira-Projektstruktur zuordnen und die Stereotypen und eventuell einen konfigurierten Bericht angeben, der den Satz von Alfabet -Objekten zurückgeben sollen, die in Jira integriert werden können. Für jede von Ihnen erstellte Jira-

Verbindung müssen Sie das Integrationsmuster angeben, damit die Objekte in der Alfabet -Objekthierarchie dem Jira-Projekt, der Jira- Projektversion oder der Jira-Projektkomponente zugeordnet werden können. Sie können für die Zuordnung zwischen einem architekturbasierten und einem projektbasierten Integrationsmuster wählen.

Ob Ihr Unternehmen die architekturbasierte oder projektbasierte Integration wählt, hängt von der Methode ab, wie Sie Jira einsetzen. Wenn Ihr Unternehmen in Alfabet eine Zuordnung zwischen Anforderungen und Produkten vornimmt, kann das Integrationsmuster für ICT-Objekte passender sein. Wenn jedoch Anforderungen Projekten zugeordnet werden, ist ein Integrationsmuster einer Projekthierarchie möglicherweise besser geeignet.



Im Folgenden finden Sie Beispiele für potenzielle Anwendungsfälle für die Integration in Jira. Das erste Beispiel basiert auf einem projektbasierten Integrationsmuster und das zweite auf einem architekturbasierten Integrationsmuster:

- Die IT-Abteilung eines Bankunternehmens dokumentiert Bugs und Änderungsanforderungen für ihre Handelssoftware in Jira und erfasst ihre Betriebspläne in Alfabet. In diesem Fall könnten die Jira-Sachverhalte zum Beispiel als Funktionen, die für Applikationen geplant sind, deren Besitz und finanzielle Verwaltung über ICT-Objekte erfolgen, in Alfabet importiert werden. Das folgende Zuordnungsschema zeigt, wie JIRA-Projekte einer Projektstruktur in Alfabet zugeordnet werden können:

Alfabet-Klasse	JIRA-Klasse
ICT-Objekt	Project
Applikation	ProjectVersion
Lokale Komponente	ProjectComponent
Funktion	Issue

- Ein Finanzinstitut erfasst Anforderungen, um die Änderung seiner Handelsfunktionen zu planen. Die Anforderungen werden Projekten in Alfabet zugeordnet. Der Projektstereotyp „Programm“ in Alfabet kann `Projects` in der JIRA-Instanz zugeordnet werden. Die Anforderungen werden als Sachverhalte in JIRA exportiert, um die Projekte zu implementieren und die erforderlichen Änderungen umzusetzen. Der untergeordnete Projektstereotyp kann der `JIRA-ProjectVersion` zugeordnet werden, und der Projektstereotyp „Projektschritt“ würde der `JIRA-ProjectComponent` entsprechen. Das folgende Zuordnungsschema zeigt, wie Jira-Projekte einer ICT-Objekthierarchie in Alfabet zugeordnet werden können:

Alfabet-Klasse	JIRA-Klasse
Projekt-Stereotyp 1	Project
Projekt-Stereotyp 2	ProjectVersion
Projekt-Stereotyp 3	ProjectComponent
Anforderung	Issue

Für einen projektbasierten Ansatz können die folgenden Integrationsmuster angegeben werden:

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
Jira-Projekte, -Projektversionen und -Projektkomponenten	Projekt -Stereotyp 1	Projekt -Stereotyp 2	Projekt -Stereotyp 3
Jira-Projekte und -Projektversionen	Projekt -Stereotyp 1	Projekt -Stereotyp 2	
Jira-Projekte und -Projektkomponenten	Projekt -Stereotyp 1		Projekt -Stereotyp 3
JIRA-Projekte	Projekt -Stereotyp 1		
Jira-Projektversionen		Projekt -Stereotyp 2	
Jira-Projektversionen und -Projektkomponenten		Projekt -Stereotyp 2	Projekt -Stereotyp 3
Jira-Projektkomponenten			Projekt -Stereotyp 3

Für einen architekturbasierten Ansatz können die folgenden Integrationsmuster angegeben werden:

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
Jira-Projekte, -Projektversionen und -Projektkomponenten	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Jira-Projekte und -Projektversionen	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	
JIRA-Projekte	ICT-Objekt (oder Stereotyp)		
Projektversionen		Applikation (oder Stereotyp)	
		Komponente (oder Stereotyp)	
		Standardplattform (oder Stereotyp)	
Projektversionen und Projektkomponenten	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Jira-Projekte, -Projektversionen und -Projektkomponenten		Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Projektkomponenten		Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)

Sie können eine beliebige Ebene in der Jira-Projektstruktur angeben, um mit der Integration zu beginnen. Sie können beispielsweise Alfabet -Applikationen den Jira-Projektversionen und die lokalen Komponenten der Applikation den Jira-Projektkomponenten zuordnen. Abhängig vom für die relevante Datenverbindung angegebenen Integrationsmuster können Anwender zum Objektprofil des entsprechenden ICT-Objekts, Projekts, der Applikation, Komponente, Standardplattform usw. in der Alfabet -Benutzeroberfläche navigieren und das Objekt über die Option **Mit Jira-Projekt verknüpfen** des Menüs **Aktion** mit dem/der entsprechenden Jira-Projekt/-Projektversion/-Projektkomponente verknüpfen. Die Option **Mit Jira-Projekt verknüpfen** ist nur für Objekte verfügbar, die für die Jira-Integration über die Stereotypdefinition des konfigurierten Berichts zulässig sind.

Nachdem das Alfabet -Objekt in der Benutzeroberfläche mit dem Jira-Projekt verknüpft wurde, können Anwender das Objekt über die Option **Mit Jira-Projektstruktur synchronisieren** regelmäßig mit der Jira-Projektstruktur synchronisieren. Die Projektversion und die Projektkomponente werden nur synchronisiert, wenn Alfabet -Objekte für die relevanten Objektklassen/Objektklassenstereotypen vorhanden sind. Darüber hinaus werden die Alfabet -Objekte nur dann mit der vorhandenen Version und Komponente des Projekts synchronisiert, wenn die für die Zuordnung angegebenen Eigenschaften denselben Wert haben. Wenn das Integrationsmuster beispielsweise angibt, dass Applikationen für JIRA-Projektversionen auf Basis der Alfabet -Eigenschaft `Name` zugeordnet werden sollen, werden alle Applikationen mit demselben Namen, die Eigentum des ausgewählten ICT-Objekts sind, derselben JIRA-Projektversion zugeordnet.

Folgende Informationen sind verfügbar:

- [Erstellen einer JIRA-Verbindung für die projektbasierte Integration](#)
- [Erstellen einer JIRA-Verbindung für die architekturbasierte Integration](#)

Erstellen einer JIRA-Verbindung für die projektbasierte Integration

In der Ansicht **Jira-Verbindung** in der Funktionalität **Konfiguration von Integrationslösungen** können Sie Jira-Verbindungen definieren, die Projektstereotypen in Alfabet zuordnen, die für die Jira-Integration relevant sind. Sie können eine Integrationszuordnung der Projekte in der Projekthierarchie, die für Ihre Alfabet-Lösung konfiguriert sind, zum Jira-Projekt, der Jira-Projektversion und der Jira-Projektkomponente festlegen. Beispielsweise könnte der Projektstereotyp Agile Release Train Jira-Projekten zugeordnet werden, und der Projektstereotyp Programminkrement kann der Jira-Projektversion zugeordnet werden.

Sie können mehrere semantische Jira-Verbindungen für jede Datenverbindung konfigurieren, die im XML-Element **DataConnection** des XML-Objekts **JiraIntegrationConfig** konfiguriert ist. Für jede Ebene in der Jira-Projektstruktur, mit der die Integration beginnen kann, sollte eine semantische Jira-Verbindung konfiguriert werden.

Für einen projektbasierten Ansatz können die folgenden Integrationsmuster angegeben werden:

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
Jira-Projekte, -Projektversionen und -Projektkomponenten	Projekt -Stereotyp 1	Projekt -Stereotyp 2	Projekt -Stereotyp 3
Jira-Projekte und -Projektversionen	Projekt -Stereotyp 1	Projekt -Stereotyp 2	
Jira-Projekte und -Projektkomponenten	Projekt -Stereotyp 1		Projekt -Stereotyp 3
JIRA-Projekte	Projekt -Stereotyp 1		
Jira-Projektversionen		Projekt -Stereotyp 2	
Jira-Projektversionen und -Projektkomponenten		Projekt -Stereotyp 2	Projekt -Stereotyp 3
Jira-Projektkomponenten			Projekt -Stereotyp 3

So konfigurieren Sie eine Jira-Verbindung für eine projektbasierte Integration:

- 1) Rufen Sie die Funktionalität **Konfiguration von Integrationslösungen** auf, und klicken Sie dann auf den Knoten **Jira-Verbindung** im Explorer **Konfiguration von Integrationslösungen**.
- 2) Klicken Sie in der Ansicht auf **Neu > JIRA-Verbindung für projektbasierte Integration erzeugen**.

- 3) Definieren Sie im Editor **JIRA-Verbindung für projektbasierte Integration erzeugen** ggf. die folgenden Felder.

Registerkarte **Basisdaten**:

- **ID**: Alfabet weist jeder JIRA-Datenbankverbindung eine eindeutige Identifikationsnummer zu. Diese Nummer kann nicht bearbeitet werden.
- **Name**: Geben Sie einen eindeutigen Namen für die JIRA-Verbindung ein. Der Name sollte dem Anwender bei der Synchronisierung der JIRA-Instanz helfen, zu der eine Verbindung hergestellt werden soll.
- **Beschreibung**: Geben Sie eine aussagekräftige Beschreibung ein, die den Zweck der JIRA-Verbindung verdeutlicht.

Registerkarte **Autorisierter Zugriff**:

- **Autorisierter Anwender**: Klicken Sie auf das **Suchen** -Symbol, um der ausgewählten JIRA-Verbindung einen autorisierten Anwender zuzuordnen. Der autorisierte Anwender verfügt über Schreib-/Lesezugriffsrechte für das Objekt und ist für dessen Pflege verantwortlich.
- **Autorisierte Anwendergruppen**: Aktivieren Sie ein oder mehrere Kontrollkästchen, um allen Anwendern in den ausgewählten Anwendergruppen Schreib-/Lesezugriffsrechte zuzuordnen.

Registerkarte **Verbindung**:



Beachten Sie Folgendes hinsichtlich der Felder auf der Registerkarte **Verbindung**:

- Das Integrationsmuster, das Sie im Feld **Integrationstyp** auswählen, bestimmt, welche Felder auf der Registerkarte **Verbindung** definiert werden müssen. Hinweis: Wenn Sie das Integrationsmuster angeben, das entweder mit einer Projektversion oder einer Projektkomponente beginnt, müssen die Optionen **Eigenschaft zum Speichern der Jira-Projekt-ID**, **Eigenschaft zum Speichern des Jira-Projektnamens** und **Eigenschaft zum Speichern des Jira-Projektschlüssels** für das übergeordnete Jira-Projekt angegeben werden. Die Felder **Eigenschaft zum Speichern der Jira-Projekt-ID**, **Eigenschaft zum Speichern des Jira-Projektnamens** und **Eigenschaft zum Speichern des Jira-Projektschlüssels** werden mit allen benutzerdefinierten Eigenschaften sowie den Standardeigenschaften des Typs `String` ausgefüllt, der im XML-Element `AlfabetClassMappingStandardPropertySettings` im XML-Objekt **JiraIntegrationConfig** angegeben ist.
- Im Feld **Alfabet-Klasse an Jira-Projekt ausgerichtet** wird immer die Objektklasse `Project` angezeigt. Sie sollten das Feld **Stereotypfilter** definieren, um zu bestimmen, welcher Projektstereotyp in Jira integriert werden kann. Wenn kein Projektstereotyp ausgewählt ist, werden alle im Feld **Stereotypfilter** verfügbaren Projektstereotypen ausgewählt. Beachten Sie, dass die Projektstereotypen, die in den Feldern **Stereotypfilter** verfügbar sind, durch die Konfiguration der Projektstereotyp-Hierarchien im XML-Objekt **ProjectManager** bestimmt werden. Informationen hierzu finden Sie im Abschnitt *Erstellen von Projektstereotypen für die Projekthierarchie* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- Die Definition des Felds **Stereotypfilter** und des Felds **Objektfilterbericht** bestimmen die Alfabet -Projekte, die in Jira-Projekte integriert werden dürfen.

Die Editor-Option **Mit Jira-Projekt verknüpfen** ist im Menü **Aktion** in den Objektprofilen der Projekte verfügbar, die über die Definition des Felds **Stereotypfilter** oder des Felds **Objektfilterbericht** gefunden werden. Informationen über das Konfigurieren der Berichte, die im Feld **Objektfilterbericht** ausgewählt werden können, finden Sie im Abschnitt [Konfigurieren von Objektfilterberichten](#).

- **Datenverbindung:** Wählen Sie die Datenverbindung aus, die im XML-Element **DataConnection** des XML-Objekts **JiraIntegrationConfig** konfiguriert ist. Diese Datenverbindung wird eingerichtet, wenn der Anwender das unten definierte Integrationsmuster über die Option **Aktion > Mit Jira-Projekt verknüpfen** im Objektprofil für den entsprechenden Projektstereotyp auswählt.
- **Projektfilter:** Wählen Sie einen der im XML-Objekt **JiraIntegrationConfig** definierten JIRA-Projektfilter aus. Die Filter begrenzen die Anzahl der Jira-Projekte, die Anwender für die Integration in dem Editor auswählen können, der beim Ausführen der Option **Mit Jira-Projekt verknüpfen** für ein Projekt in Alfabet geöffnet wird. Beispielsweise könnte ein Filter angeben, dass nur Jira-Projekte ausgegeben werden, bei denen der Projekttyp „Software“ oder „Business“ entspricht.
- **Beschreibung des Projektfilters:** Zeigt Informationen zu dem im Feld **Projektfilter auswählen** ausgewählten Filter an.
- **Integrationstyp:** Wählen Sie die Ebene in der Jira-Projektstruktur aus, mit der die Integration beginnen soll. Die im Feld **Integrationsmuster** angezeigten Projektstereotypen werden auf der Grundlage des Felds **Integrationstyp** bestimmt. Sie können den spezifischen Alfabet - Projektstereotyp auswählen, der der relevanten Ebene in der Jira-Projektstruktur im jeweiligen Feld **Stereotypfilter** zugeordnet werden soll.
- **Integrationsmuster:** Wählen Sie das Architekturmuster aus, um Alfabet -Projekte der im Feld **Integrationstyp** ausgewählten Jira-Projektstruktur zuzuordnen.
- **Alfabet-Klasse an Jira-Projekt ausgerichtet:** Zeigt die Objektklasse `Project` an.
- **Stereotypfilter:** Wählen Sie einen Alfabet -Projektstereotyp aus, der Jira-Projekten zugeordnet werden soll. Wenn kein Projektstereotyp ausgewählt ist, werden alle verfügbaren Projektstereotypen ausgewählt.
- **Objektfilterbericht:** Wählen Sie den Alfabet -Bericht aus, der die relevanten Objekte zurückgibt, die in Jira-Objekte integriert werden können. Dies ist ein optionaler Bericht, mit dem Sie die für die Jira-Integration verfügbaren Projekte einschränken können.
- **Eigenschaft zum Speichern der Jira-Projekt-ID:** Wählen Sie die Alfabet - Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `ID` für Jira-Projekte verwendet werden soll.
- **Eigenschaft zum Speichern des Jira-Projektnamens:** Wählen Sie die Alfabet - Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `Name` für Jira-Projekte verwendet werden soll.
- **Eigenschaft zum Speichern des Jira-Projektschlüssels:** Wählen Sie die Alfabet - Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `Key` für Jira-Projekte verwendet werden soll.
- **Alfabet-Klasse an Jira-Projektversion ausgerichtet:** Zeigt die Objektklasse `Project` an.

- **Stereotypfilter:** Wählen Sie einen Alfabet -Projektstereotyp aus, der Projektversionen in Jira zugeordnet werden soll. Wenn kein Projektstereotyp ausgewählt ist, werden alle verfügbaren Projektstereotypen ausgewählt.
 - **Objektfilterbericht:** Wählen Sie den Alfabet -Bericht aus, der die relevanten Objekte zurückgibt, die in Jira-Projektversionen integriert werden können. Dies ist ein optionaler Bericht, mit dem Sie die für die Jira-Integration verfügbaren Projektversionen einschränken können.
 - **Eigenschaft zum Speichern der Jira-Projekt-ID:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `ID` für Jira-Projektversionen verwendet werden soll.
 - **Eigenschaft zum Speichern des Jira-Projektnamens:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `Name` für Jira-Projektversionen verwendet werden soll.
 - **Eigenschaft zum Erzeugen der Jira-Version bei Synchronisierung:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `Name` verwendet werden soll, wenn neue Jira-Projektversionen erzeugt werden.
 - **Alfabet-Klasse an Jira-Projektkomponente ausgerichtet:** Zeigt die Objektklasse `Project` an.
 - **Stereotypfilter:** Wählen Sie einen Alfabet -Projektstereotyp aus, der Jira-Projektkomponenten zugeordnet werden soll. Wenn kein Projektstereotyp ausgewählt ist, werden alle verfügbaren Projektstereotypen ausgewählt.
 - **Objektfilterbericht:** Wählen Sie den Alfabet -Bericht aus, der die relevanten Objekte zurückgibt, die in Jira-Projektversionen integriert werden können. Dies ist ein optionaler Bericht, mit dem Sie die für die Jira-Integration verfügbaren Projektversionen einschränken können.
 - **Eigenschaft zum Speichern der Jira-Projekt-ID:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `ID` für Jira-Projektversionen verwendet werden soll.
 - **Eigenschaft zum Speichern des Jira-Projektnamens:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `Name` für Jira-Projektversionen verwendet werden soll.
 - **Ereignisvorlagen für die Synchronisierung:** Wählen Sie Alfabet -Ereignisvorlagen zum Aufrufen der entsprechenden ADIF-Import-/Exportschemata aus, die ausgelöst werden sollen, wenn ein Alfabet -Objekt mit den Daten des verknüpften Jira-Objekts synchronisiert wird. Details über das Konfigurieren von Ereignisvorlagen zum Exportieren von Alfabet -Daten in Jira finden Sie im Abschnitt [Konfigurieren der Ereignisvorlagen zum Auslösen der ADIF-Exportschemata für die Synchronisierung](#).
 - **Ereignisvorlagensequenz:** Ordnen Sie die Reihenfolge neu an, in der die ausgewählten Alfabet -Ereignisvorlagen ausgelöst werden. Klicken Sie dazu im Fenster auf eine Ereignisvorlage und dann auf
- 4) Wiederholen Sie diesen Vorgang für alle erforderlichen konfigurierten **DataConnection**-Elemente des XML-Objekts **JiraIntegrationConfig**.

Erstellen einer JIRA-Verbindung für die architekturbasierte Integration

In der Ansicht **Jira-Verbindung** in der Funktionalität **Konfiguration von Integrationslösungen** können Sie Jira-Verbindungen definieren, die die Zuordnung von Objektklassen oder ihren Objektklassenstereotypen in Alfabet angeben, die für die Jira-Integration relevant sind. Sie können eine Integrationszuordnung basierend auf vordefinierten Architekturen für Objekthierarchien in Alfabet festlegen, die dem Jira-Projekt, der Projektversion und der Projektkomponente zugeordnet werden können. Für jede Ebene in der Jira-Projektstruktur, mit der die Integration beginnen kann, sollte eine semantische Definition konfiguriert werden.

Für einen architekturbasierten Ansatz können die folgenden Integrationsmuster angegeben werden:

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
Jira-Projekte, -Projektversionen und -Projektkomponenten	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Jira-Projekte und -Projektversionen	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	
JIRA-Projekte	ICT-Objekt (oder Stereotyp)		
Projektversionen		Applikation (oder Stereotyp)	
		Komponente (oder Stereotyp)	
		Standardplattform (oder Stereotyp)	

Integrationstyp	Jira-Projekt	Jira-Projektversion	Jira-Projektkomponente
Projektversionen und Projektkomponenten	ICT-Objekt (oder Stereotyp)	Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
	ICT-Objekt (oder Stereotyp)	Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Jira-Projekte, -Projektversionen und -Projektkomponenten		Applikation (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)
Projektkomponenten		Komponente (oder Stereotyp)	Lokale Komponente (oder Stereotyp)
		Standardplattform (oder Stereotyp)	Plattformelement (oder Stereotyp)

Sie können mehrere semantische Datenverbindungen für jede Datenverbindung konfigurieren, die im XML-Element **DataConnection** des XML-Objekts **JiraIntegrationConfig** konfiguriert ist.

So konfigurieren Sie eine JIRA-Datenbankverbindung für eine architekturbasierte Integration:

- 1) Rufen Sie die Funktionalität **Konfiguration von Integrationslösungen** auf, und klicken Sie dann auf den Knoten **Jira-Verbindung** im Explorer **Konfiguration von Integrationslösungen**.
- 2) Klicken Sie in der Ansicht auf **Neu > JIRA-Verbindung für architekturbasierte Integration erzeugen**.
- 3) Definieren Sie im Editor **JIRA-Verbindung für architekturbasierte Integration erzeugen** ggf. die folgenden Felder.

Registerkarte **Basisdaten**:

- **ID**: Alfabet weist jeder JIRA-Datenbankverbindung eine eindeutige Identifikationsnummer zu. Diese Nummer kann nicht bearbeitet werden.

- **Name:** Geben Sie einen eindeutigen Namen für die JIRA-Verbindung ein. Der Name sollte dem Anwender bei der Synchronisierung der JIRA-Instanz helfen, zu der eine Verbindung hergestellt werden soll.
- **Beschreibung:** Geben Sie eine aussagekräftige Beschreibung ein, die den Zweck der JIRA-Verbindung verdeutlicht.

Registerkarte **Autorisierter Zugriff:**

- **Autorisierter Anwender:** Klicken Sie auf das **Suchen** -Symbol, um der ausgewählten JIRA-Verbindung einen autorisierten Anwender zuzuordnen. Der autorisierte Anwender verfügt über Schreib-/Lesezugriffsrechte für das Objekt und ist für dessen Pflege verantwortlich.
- **Autorisierte Anwendergruppen:** Aktivieren Sie ein oder mehrere Kontrollkästchen, um allen Anwendern in den ausgewählten Anwendergruppen Schreib-/Lesezugriffsrechte zuzuordnen.

Registerkarte **Verbindung:**



Beachten Sie Folgendes hinsichtlich der Felder auf der Registerkarte **Verbindung:**

- Das Integrationsmuster, das Sie im Feld **Integrationstyp** auswählen, bestimmt, welche Felder auf der Registerkarte **Verbindung** definiert werden müssen. Hinweis: Wenn Sie das Integrationsmuster angeben, das entweder mit einer Projektversion oder einer Projektkomponente beginnt, müssen die Optionen **Eigenschaft zum Speichern der Jira-Projekt-ID**, **Eigenschaft zum Speichern des Jira-Projektnamens** und **Eigenschaft zum Speichern des Jira-Projektschlüssels** für das übergeordnete Jira-Projekt angegeben werden. Die Felder **Eigenschaft zum Speichern der Jira-Projekt-ID**, **Eigenschaft zum Speichern des Jira-Projektnamens** und **Eigenschaft zum Speichern des Jira-Projektschlüssels** werden mit allen benutzerdefinierten Eigenschaften sowie den Standardeigenschaften des Typs `String` ausgefüllt, der im XML-Element `AlfabetClassMappingStandardPropertySettings` im XML-Objekt ***JiraIntegrationConfig*** angegeben ist.
- Sie können das Feld **Stereotypfilter** definieren, um zu bestimmen, welcher Objektklassenstereotyp in Jira integriert werden kann. Wenn kein Objektklassenstereotyp ausgewählt ist, werden alle im Feld **Stereotypfilter** verfügbaren Objektklassenstereotypen ausgewählt.
- Die Definition des Felds **Stereotypfilter** und des Felds **Objektfilterbericht** bestimmen die Alfabet -Objekte, die in Jira-Projekte integriert werden dürfen. Die Editor-Option **Mit Jira-Projekt verknüpfen** ist im Menü **Aktion** in den Objektprofilen der Objekte verfügbar, die über die Definition des Felds **Stereotypfilter** oder des Felds **Objektfilterbericht** gefunden werden. Informationen über das Konfigurieren der Berichte, die im Feld **Objektfilterbericht** ausgewählt werden können, finden Sie im Abschnitt [Konfigurieren von Objektfilterberichten](#).
- **Alfabet-Klasse an Jira-Projekt ausgerichtet:** Geben Sie die Alfabet -Klasse im angegebenen Integrationsmuster an, die mit Jira-Projekten abgeglichen wird. Ob dieses Feld definiert werden kann oder nicht, hängt von der Auswahl im Feld **Integrationstyp** ab.
- **Datenverbindung:** Wählen Sie die Datenverbindung aus, die im XML-Element **DataConnection** des XML-Objekts ***JiraIntegrationConfig*** konfiguriert ist. Diese

Datenverbindung wird eingerichtet, wenn der Anwender das unten definierte Integrationsmuster über die Option **Aktion > Mit Jira-Projekt verknüpfen** im Objektprofil für das entsprechende Objekt auswählt.

- **Projektfilter:** Wählen Sie einen der im XML-Objekt *JiraIntegrationConfig* definierten JIRA-Projektfiler aus. Die Filter begrenzen die Anzahl der Jira-Projekte, die Anwender für die Integration in dem Editor auswählen können, der beim Ausführen der Option **Mit Jira-Projekt verknüpfen** für ein Objekt geöffnet wird. Beispielsweise könnte ein Filter angeben, dass nur Jira-Projekte ausgegeben werden, bei denen der Projekttyp „Software“ oder „Business“ entspricht.
- **Beschreibung des Projektfilters:** Zeigt Informationen zu dem im Feld **Projektfiler auswählen** ausgewählten Filter an.
- **Integrationstyp:** Wählen Sie die Ebene in der Jira-Projektstruktur aus, mit der die Integration beginnen soll. Die im Feld **Integrationsmuster** angezeigten Objektklassen werden auf der Grundlage des Felds **Integrationstyp** bestimmt. Sie können den spezifischen Objektstereotyp auswählen, der der relevanten Ebene in der Jira-Projektstruktur im jeweiligen Feld **Stereotypfilter** zugeordnet werden soll.
- **Integrationsmuster:** Wählen Sie das Architekturmuster aus, um Alfabet -Objekte der im Feld **Integrationstyp** ausgewählten Jira-Projektstruktur zuzuordnen.
- **Alfabet-Klasse an Jira-Projekt ausgerichtet:** Zeigt die Alfabet -Klasse im angegebenen Integrationsmuster an, die Jira-Projekten zugeordnet ist. Ob dieses Feld definiert werden kann oder nicht, hängt von der Auswahl im Feld **Integrationstyp** ab.
- **Stereotypfilter:** Wählen Sie einen Objektstereotyp aus, der Jira-Projekten zugeordnet werden soll. Wenn kein Objektklassenstereotyp ausgewählt ist, werden alle verfügbaren Objektklassenstereotypen ausgewählt.
- **Objektfilerbericht:** Wählen Sie den Alfabet -Bericht aus, der die relevanten Objekte zurückgibt, die in Jira-Projekte integriert werden können. Dies ist ein optionaler Bericht, mit dem Sie die für die Jira-Integration verfügbaren Objekte einschränken können.
- **Eigenschaft zum Speichern der Jira-Projekt-ID:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `ID` für Jira-Projekte verwendet werden soll.
- **Eigenschaft zum Speichern des Jira-Projektnamens:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `Name` für Jira-Projekte verwendet werden soll.
- **Eigenschaft zum Speichern des Jira-Projektschlüssels:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `Key` für Jira-Projekte verwendet werden soll.
- **Alfabet-Klasse an Jira-Projektversion ausgerichtet:** Zeigt die Alfabet -Klasse im angegebenen Integrationsmuster an, die Jira-Projektversionen zugeordnet ist. Ob dieses Feld definiert werden kann oder nicht, hängt von der Auswahl im Feld **Integrationstyp** ab.
- **Stereotypfilter:** Wählen Sie einen Objektklassenstereotyp aus, der Projektversionen in Jira zugeordnet werden soll. Wenn kein Objektklassenstereotyp ausgewählt ist, werden alle verfügbaren Objektklassenstereotypen ausgewählt.

- **Objektfilterbericht:** Wählen Sie den Alfabet -Bericht aus, der die relevanten Objekte zurückgibt, die in Jira-Projektversionen integriert werden können. Dies ist ein optionaler Bericht, mit dem Sie die für die Jira-Integration verfügbaren Objekte einschränken können.
 - **Eigenschaft zum Speichern der Jira-Projekt-ID:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `ID` für Jira-Projektversionen verwendet werden soll.
 - **Eigenschaft zum Speichern des Jira-Projektnamens:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `Name` für Jira-Projektversionen verwendet werden soll.
 - **Eigenschaft zum Erzeugen der Jira-Version bei Synchronisierung:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `Name` verwendet werden soll, wenn neue Jira-Projektversionen erzeugt werden.
 - **Alfabet-Klasse an Jira-Projektkomponente ausgerichtet:** Zeigt die Alfabet -Klasse im angegebenen Integrationsmuster an, die Jira-Projektkomponenten zugeordnet ist. Ob dieses Feld definiert werden kann oder nicht, hängt von der Auswahl im Feld **Integrationstyp** ab.
 - **Stereotypfilter:** Wählen Sie einen Objektklassenstereotyp aus, der Jira-Projektkomponenten zugeordnet werden soll. Wenn kein Objektklassenstereotyp ausgewählt ist, werden alle verfügbaren Objektklassenstereotypen ausgewählt.
 - **Objektfilterbericht:** Wählen Sie den Alfabet -Bericht aus, der die relevanten Objekte zurückgibt, die in Jira-Projektversionen integriert werden können. Dies ist ein optionaler Bericht, mit dem Sie die für die Jira-Integration verfügbaren Objekte einschränken können.
 - **Eigenschaft zum Speichern der Jira-Projekt-ID:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `ID` für Jira-Projektversionen verwendet werden soll.
 - **Eigenschaft zum Speichern des Jira-Projektnamens:** Wählen Sie die Alfabet -Klasseneigenschaft aus, die zum Erfassen der Eigenschaft `Name` für Jira-Projektversionen verwendet werden soll.
 - **Ereignisvorlagen für die Synchronisierung:** Wählen Sie Alfabet -Ereignisvorlagen zum Aufrufen der entsprechenden ADIF-Import-/Exportschemata aus, die ausgelöst werden sollen, wenn ein Alfabet -Objekt mit den Daten des verknüpften Jira-Objekts synchronisiert wird. Details über das Konfigurieren von Ereignisvorlagen zum Exportieren von Alfabet -Daten in Jira finden Sie im Abschnitt [Konfigurieren der Ereignisvorlagen zum Auslösen der ADIF-Exportschemata für die Synchronisierung](#).
 - **Ereignisvorlagensequenz:** Ordnen Sie die Reihenfolge neu an, in der die ausgewählten Alfabet -Ereignisvorlagen ausgelöst werden. Klicken Sie dazu im Fenster auf eine Ereignisvorlage und dann auf
- 4) Wiederholen Sie diesen Vorgang für alle erforderlichen konfigurierten **Connection**-Elemente des XML-Objekts **JiraIntegrationConfig**.

Verknüpfung und Synchronisierung des JIRA-Projekts

Sobald ein Integrationsmuster in der Funktionalität **Konfiguration der Integrationslösungen** konfiguriert wurde, können Sie das relevante Alfabet -Objekt mit einem Jira-Projekt verknüpfen. Je nach Konfiguration

des Integrationsmusters für die semantische Jira-Verbindung kann es sich bei dem Objekt, das verknüpft werden kann, um ein Projektstereotyp, ein ICT-Objekt, eine Applikation, eine Komponente, eine lokale Komponente, eine Standardplattform oder ein Standardplattformelement handeln. Die Option **Mit Jira-Projekt verknüpfen** ist im Objektprofil für alle Objekte verfügbar, die über die Stereotypdefinition oder die konfigurierte Berichtsdefinition einer semantischen Jira-Verbindung angegeben werden.



Der Alfabet-Server muss ausgeführt werden und eine Verbindung mit dem Internet herstellen können.

So stellen Sie eine Verknüpfung mit einem JIRA-Projekt her:

- 1) Navigieren Sie in der Benutzeroberfläche von Alfabet zum Objektprofil des entsprechenden Objekts, das über eine semantische Jira-Verbindung angegeben wurde.
- 2) Klicken Sie in der Symbolleiste auf **Aktion > Mit Jira-Projekt verknüpfen**. Hinweis: Wenn Sie ein Alfabet -Objekt verknüpfen, das in eine Jira-Projektversion integriert werden soll, wird die Option **Aktion > Mit Jira-Projektversion verknüpfen** angezeigt. Wenn Sie ein Alfabet -Objekt verknüpfen, das in eine Jira-Projektkomponente integriert werden soll, wird die Option **Aktion > Mit Jira-Projektkomponente verknüpfen** angezeigt. Der Editor **Mit JIRA-Projekt verknüpfen** wird geöffnet.
- 3) Wählen Sie im Feld **Integrationsverbindung** die relevante Verbindung zum Verbinden mit Jira aus. Es werden alle JIRA-Projekte angezeigt, die den angegebenen Bedingungen für die Projektfilter entsprechen, die für die ausgewählte Integrationsverbindung angegeben wurden. Die Jira-Projekte können weiter eingeschränkt werden, indem im Feld **Suchmuster** Kriterien angegeben werden.
- 4) Wählen Sie im Datensatz das relevante Jira-Projekt bzw. die relevante Jira-Projektversion/Jira-Projektkomponente aus, das/die Sie mit dem ausgewählten Projekt verknüpfen möchten, und klicken Sie auf **OK**. Mit der Aktion **Mit JIRA-Projekt verknüpfen** wird ein neuer Datensatz in der Datenbanktabelle für die Klasse `IntegrationConnectionUsage` erzeugt. In der Tabelle werden die Beziehung zwischen der ausgewählten Integrationsverbindung, dem ICT-Objekt/Projekt in Alfabet und dem Bezeichner des Jira-Projekts zugeordnet.
- 5) Um die Jira-Projektstruktur (die relevante Jira-Projektversion und die Jira-Projektkomponente) zu synchronisieren, klicken Sie auf **Aktion > Mit Jira-Projektstruktur synchronisieren**. Die Synchronisierung erstellt die entsprechenden Projektversionen und Projektkomponenten anhand des für die von Ihnen genutzte Datenverbindung angegebenen Integrationsmusters. Wenn das Alfabet -Objekt, mit dem die Integration gestartet werden soll, beispielsweise eine Applikation ist und die Applikation mit einer Jira-Projektversion verknüpft wurde, werden die lokalen Komponenten der Applikation mit den Jira-Projektkomponenten der Jira-Projektversion synchronisiert.

Die Projektversion und die Projektkomponente werden nur synchronisiert, wenn Alfabet -Objekte für die relevanten Objektklassen/Objektklassenstereotypen vorhanden sind. Beispiel: Das Integrationsmuster gibt die Zuordnung der Objektklasse „Applikation“ zu Jira-Projektversionen und die Zuordnung des Objektklassenstereotyps „Technische Komponente“ der Klasse „Lokale Komponente“ zur Jira-Projektkomponente an. Wenn die Applikation „GL Trade Products“ mit einer Jira-Projektversion verknüpft ist, werden bei der Synchronisierung alle lokalen Komponenten, die der Applikation zugeordnet sind, als Jira-Projektkomponenten für die Jira-Projektversion erstellt. Wenn jedoch keine technischen Komponenten für eine der Applikationen definiert wurden, werden keine Jira-Projektkomponenten für die lokale Komponente von Alfabet erstellt.

.Darüber hinaus werden die Alfabet -Objekte nur dann mit der vorhandenen Version und Komponente eines Projekts synchronisiert, wenn die für die Zuordnung angegebenen Eigenschaften

denselben Wert haben. Wenn das Integrationsmuster beispielsweise angibt, dass ICT-Objekte Jira-Projekten auf Basis der Alfabet -Eigenschaft `Name` zugeordnet werden sollen, werden alle Applikationen mit demselben Namen, die Eigentum des ausgewählten Alfabet -ICT-Objekts sind, derselben Jira-Projektversion zugeordnet.

- 6) Um die Ereignisvorlagen auszulösen, die für die Synchronisierung für die relevante Jira-Integrationsverbindung angegeben wurden, klicken Sie auf **Aktion > Mit Jira-Projektdaten synchronisieren**. Die Ereignisvorlagen müssen so konfiguriert werden, dass sie die ADIF-Exportschemata auslösen, die den entsprechenden konfigurierten Berichten zugeordnet sind. Der konfigurierte Bericht, der dem ADIF-Exportschema zugeordnet ist, sollte einen Satz von Objekten zurückgeben, die Jira-Sachverhalten zugeordnet werden können.

Kapitel 13: Konfiguration der Interoperabilität mit Microsoft Project

Die Interoperabilität mit Microsoft® Project ermöglicht Ihrem Unternehmen die Synchronisierung der strategischen Projektportfolioplanung mit dem operativen Projektmanagement sowie die Bewertung aller Projekte im Unternehmen im Kontext der Unternehmensstrategie und der strategischen Investitionen. Mit dieser Funktionalität kann das Unternehmen Projekte, die in Alfabet geplant sind, nach Microsoft Project exportieren, und Projekte nach Alfabet importieren, die in Microsoft Project verwaltet werden.



Die Konfiguration der Projektmanagement-Funktionalität einschließlich Projektstereotypen, Release-Status, Assistenten usw. sollte vor der Konfiguration der Integration von Alfabet-Projekten in Microsoft Project-Projekte abgeschlossen werden. Informationen hierzu finden Sie im Abschnitt *Konfigurieren der Projektmanagement-Funktionalität* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Folgendes ist erforderlich, um MS Project-Projekte in Alfabet zu importieren, importierte Alfabet-Projekte mit ihren aktualisierten Projekten in MS Project zu synchronisieren oder Alfabet-Projekte nach MS Project zu exportieren:

- Geben Sie Datenverbindungen und Importmuster im XML-Objekt **MicrosoftProjectPlanConfig** an.
- Stellen Sie sicher, dass ein Assistent erzeugt wurde und der Klasseneinstellung für die Klasse „Projekt“ zugeordnet ist. Dies ist erforderlich, um Projekte über MS Project zu importieren. Dem ersten Assistentenschritt wird ein zusätzlicher Schritt vorangestellt, mit dem der Import ausgelöst werden kann, um das neue Projekt in Alfabet zu erstellen.
- Stellen Sie sicher, dass Release-Status-Definitionen für die Klasse `MicrosoftProjectPlan_Mappings` im XML-Objekt **ReleaseStatusDefs** erzeugt wurden.
- Stellen Sie sicher, dass für die relevanten Anwenderprofile eine Ansicht verfügbar ist, die die Funktionalität zum Importieren von MS Project-Projekten, Synchronisieren von Alfabet-Projekten mit aktualisierten MS Project-Projekten und Exportieren von Alfabet-Projekten nach Microsoft Server bietet. Weitere Informationen zum Konfigurieren von Anwenderprofilen finden Sie im Abschnitt *Konfigurieren von Anwenderprofilen für die Anwender-Community* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*. Dies umfasst die folgenden Ansichten:
 - Funktionalitäten **Projekte erfassen** (`PRJ_CaptureProjects` und `PRJ_CaptureProjects_Ex`)
 - Ansichtsseiten **Projekte** für Projektgruppen und Buckets (`PRJG_Projects` und `BKT_Projects`)
 - Objektprofil **Projekt** (`PRJ_ObjectView`) oder ein Projektstereotyp-Objektprofil
- Geben Sie die MS Project-Methodik an, die für den Import/Export/die Synchronisierung von MS Project-Projekten über den Knoten **Microsoft Project-Methodik** in der Funktionalität **Konfiguration von Integrationslösungen** in der Benutzeroberfläche verwendet werden soll. Informationen hierzu finden Sie im Abschnitt *Definieren von Microsoft-Project-Methodiken für die Interoperabilität mit Microsoft Project* im Referenzhandbuch *Konfigurieren von Bewertungen und Referenzdaten in Alfabet*.



Stellen Sie sicher, dass die Dateierweiterungen `.mpp` und `.mpt` nicht in der im XML-Objekt **FileExtensionLists** definierten schwarzen Liste aufgeführt sind. Wenn eine Whitelist im XML-Objekt **FileExtensionLists** angegeben ist, müssen die Dateierweiterungen `.mpp` und `.mpt` als zulässige Dateierweiterungen aufgeführt sein. Informationen hierzu finden Sie im Abschnitt *Festlegen der zulässigen Dateinamenerweiterungen für das Hochladen/Herunterladen von Dateien* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Folgende Informationen sind verfügbar:

- [Konfigurieren der Verbindungen für die Interoperabilität mit Microsoft Project](#)
- [Angaben eines Assistenten für die Interoperabilität mit Microsoft Project](#)
- [Angaben eines Release-Status für die Interoperabilität mit Microsoft Project](#)

Konfigurieren der Verbindungen für die Interoperabilität mit Microsoft Project

Geben Sie Datenverbindungen und Importmuster im XML-Objekt **MicrosoftProjectPlanConfig** an. Diese sind erforderlich, um die semantischen Verbindungen für die Integration von Microsoft Project zu definieren.

Mit dem XML-Objekt **MicrosoftProjectPlanConfig** kann ein Lösungsentwickler Datenverbindungen und die Datenzuordnung angeben, die für den Import und Export von Alfabet -Projekten von/nach Microsoft Project erforderlich sind. Darüber hinaus erfordern die Optionen zum Importieren von Übereinstimmungsregeln für den Import von Microsoft Project -Projekten/-Projektaufgaben in Alfabet, dass die Personalanforderungen festgelegt werden können. Beachten Sie Folgendes:

- Sie können angeben, dass eine Verbindung zu einer oder mehreren Microsoft Project Server-Instanzen erstellt wird, um Daten von der Serverinstanz zu importieren. Alternativ können Daten über eine oder mehrere MPP-Dateien in Alfabet importiert werden. Beachten Sie, dass die Verbindung zu Microsoft Project Server nur den Import von Microsoft Project-Projekten und -Projektaufgaben in Alfabet unterstützt. Der Export von Alfabet nach Microsoft Project Server wird nicht unterstützt.
- Sie müssen die Zuordnung zwischen Alfabet -Eigenschaftstypen und Microsoft Project-Projekttypen angeben, um Daten aus einer Microsoft Project Server-Instanz oder einer MPP-Datei zu importieren/exportieren.
- Sie müssen die Übereinstimmungsregeln für den Import festlegen, die beschreiben, wie die in Microsoft Project erfassten Ressourcendaten verwendet werden sollen, um den Personalbedarf, die Organisation und die Person zu finden, auf die eine im Kontext des Imports einer Microsoft Project-Projektaufgabe erstellte Personalanforderung verweist. Sie müssen mindestens eine Eigenschaft oder ein übereinstimmendes Muster für jede Klasse definieren. Sie können die relevanten Fähigkeiten, die Organisation und die Person in der Alfabet-Datenbank basierend auf den Ressourcendaten, die mit der Projektaufgabe verknüpft sind, die in Alfabet importiert wurde, auf folgende Arten suchen:
 - Geben Sie Objektklasseneigenschaften für die Klassen `Skill` (obligatorisch), `Person` (optional) und `OrgaUnit` (optional) an, die für die Suche nach Fähigkeiten, Personen und Organisationen in der Alfabet-Datenbank verwendet werden sollen.



Die Eigenschaft `Name` ist für die Klasse `Skill` definiert. Eine Projektaufgabe wird als Personalanforderung in Alfabet importiert. Die Projektaufgabe hat eine Ressource für „Java-Programmierung“. Beim Importieren sucht das System nach einem Personalbedarf mit dem Namen „Java-Programmierung“. Wenn ein vorhandener Personalbedarf gefunden wurde, verweist die neue Personalanforderung auf den Personalbedarf „Java-Programmierung“.

- Geben Sie Übereinstimmungsmuster an, die aus mehreren Eigenschaften und einem Trennzeichen für die Klassen `Skill` (obligatorisch), `Person` (optional) und `OrgaUnit` (optional) bestehen, die für die Suche nach Fähigkeiten, Personen und Organisationen in der Alfabet-Datenbank verwendet werden sollen.



Ein übereinstimmendes Muster, bestehend aus Unterstrich-Trennzeichen (`_`) und den Eigenschaften `Name` und `ID`, die für die Klasse `Skill` definiert sind. Eine Projektaufgabe wird als Personalanforderung in Alfabet importiert. Die Projektaufgabe hat eine Ressource für „Java-Programmierung_123“. Beim Importieren sucht das System nach einem Personalbedarf mit dem Namen „Java-Programmierung“ und der ID 123. Wenn ein vorhandener Personalbedarf mit einem übereinstimmenden Namen und einer übereinstimmenden ID gefunden wird, verweist die neue Personalanforderung auf den Personalbedarf „Java-Programmierung“. Wenn die Ressource jedoch „Java-Programmierung/123“ genannt wurde, wird kein Personalbedarf gefunden, da als Trennzeichen im Ressourcennamen ein Schrägstrich statt einem Unterstrich verwendet wurde.



Beachten Sie Folgendes bezüglich der Implementierung von Übereinstimmungsregeln für den Import:

- Ein oder mehrere Eigenschaften und Übereinstimmungsmuster können vom Anwender ausgewählt werden, der die Methode im Editor **Microsoft Project-Methodik** in der Funktionalität **Konfiguration von Integrationslösungen** definiert:
- Kriterien wie `Contains`, `Starts With` usw. können für die ausgewählten Eigenschaften und Übereinstimmungsmuster vom Anwender definiert werden, der die Methode im Editor **Microsoft Project-Methodik** in der Funktionalität **Konfiguration von Integrationslösungen** definiert:
- Die erste gefundene relevante Personalanforderung, Organisation oder Person, die den Kriterien in Alfabet-Datenbank entspricht, wird verwendet.
- Jede Personalanforderung, die in Alfabet importiert wird, erfordert einen Verweis auf einen Personalbedarf. Dies ist obligatorisch. Der Verweis auf eine Person oder Organisation ist für eine Personalanforderung optional. Der Anwender, der das Microsoft Project-Projekt importiert, kann einen Standardpersonalbedarf auswählen, wenn durch die Übereinstimmungsregeln für den Import kein Personalbedarf gefunden wird.



Beachten Sie, dass Sie bei Änderungen am XML-Objekt **MicrosoftProjectPlanConfig**, nachdem die Microsoft Project-Methodiken in der Ansicht **Microsoft Project-Methodik** in der Funktionalität **Konfiguration der Integrationslösungen** erzeugt wurden, die Import- und Exportzuordnungsregeln der Microsoft Project-Methodiken überprüfen und bei Bedarf Anpassungen vornehmen sollten. Änderungen am XML-Objekt **MicrosoftProjectPlanConfig** werden nicht automatisch in vorhandenen Microsoft Project-Methodiken aktualisiert. Weitere Informationen über das Konfigurieren von Microsoft Project-Methodiken finden Sie unter *Definieren von Microsoft-*

Project-Methodiken für die Interoperabilität mit Microsoft Project im Referenzhandbuch Konfigurieren von Bewertungen und Referenzdaten in Alfabet.

So definieren Sie das XML-Objekt **MicrosoftProjectPlanConfig**:

- 1) Wechseln Sie in Alfabet Expand zur Registerkarte **Präsentation**, und erweitern Sie die Explorer-Knoten **XML-Objekte > IntegrationSolutions**.
- 2) Klicken Sie mit der rechten Maustaste auf **MicrosoftProjectPlanConfig**, und wählen Sie **XML bearbeiten....** Das XML-Objekt **MicrosoftProjectPlanConfig** wird geöffnet.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist über das Attribut **XML-Vorlage** im Attributraster des XML-Objekts **MicrosoftProjectPlanConfig** eine Vorlage verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. In diesem Fall bearbeiten Sie die nachfolgend beschriebenen XML-Elemente. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf.

- 3) **Wenn Projekte von Microsoft Project Server importiert werden sollen** (optional): Fügen Sie dem Root-XML-Element `MicrosoftProjectPlanConfig` ein XML-Element `DataConnections` hinzu. Fügen Sie für jede Microsoft® Project Server-Instanz ein untergeordnetes XML-Element `DataConnections` hinzu und geben die folgenden XML-Attribute an:
 - **Name:** Der Name der Verbindung zum Microsoft® Project Server. Für die Verbindung muss ein eindeutiger Name festgelegt werden. Der Name wird im Feld **Microsoft Project-Serververbindung** des Projektassistenten angezeigt, der geöffnet wird, wenn der Import über MS Project Server ausgelöst wird.
 - **IsActive:** Geben Sie „true“ ein, um die Interoperabilität mit Microsoft® Project Server zu aktivieren. Ein Microsoft® Project Server kann jederzeit aktiv sein. Wenn mehrere Verbindungen auf `IsActive=true` eingestellt sind, wird die erste aktive Verbindung verwendet.
 - **ServiceURL:** Geben Sie die URL der Microsoft® Project Server-Instanz an.
 - **ServiceUserName:** Geben Sie den Anwendernamen für den Zugriff auf die Microsoft® Project Server-Instanz an.
 - **ServiceUserDomain:** Definieren Sie bei Bedarf den Domännennamen, der als Teil des Anwendernamens zur Authentifizierung am Proxyserver verwendet werden soll.



Hinweis: Die Servervariablen lesen den Wert des XML-Attributs zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation aus, wenn eine Verbindung zur Integrationslösung hergestellt wurde. Informationen zur Konfiguration der Servervariablen im Server-Alias finden Sie unter [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).

- **ServicePassword:** Geben Sie das Passwort für den Zugriff auf die Microsoft® Project Server-Instanz an.
- **DBDataSource:** Geben Sie die Datenbankdatenquelle der Microsoft® Project Server-Instanz an, auf die zugegriffen werden soll.
- **DBInitialCatalog:** Geben Sie den Datenbankkatalognamen der Microsoft® Project Server-Instanz ein, auf den zugegriffen werden soll.

- **DBSchema:** Geben Sie den Namen des Datenbankschemas der Microsoft® Project Server-Instanz ein, auf das zugegriffen werden soll.
 - **DBUserName:** Geben Sie den Datenbank-Benutzernamen für den Zugriff auf die Microsoft® Project Server-Instanz ein.
 - **DBPassword:** Geben Sie das Datenbank-Benutzerkennwort für den Zugriff auf die Microsoft® Project Server-Instanz ein.
- 4) Fügen Sie dem Root-XML-Element `MicrosoftProjectPlanConfig` ein XML-Element `DataMappings` hinzu. Fügen Sie das XML-Attribut `UnknownMPPTypesAsString` hinzu, und geben Sie „true“ an, wenn Datentypen, die nicht in einem XML-Element `DataTypes` angegeben sind, aus Microsoft® Project mit dem Datentyp `String` nach Alfabet importiert werden sollen. Geben Sie „false“ an, wenn unbekannte Datentypen nicht nach Alfabet importiert werden sollen.
- 5) Erstellen Sie ein XML-Element `DataTypes` für jeden Microsoft Project-Datentyp, der beim Importieren/Exportieren von Daten einem Alfabet -Datentyp zugeordnet werden soll. Spezifizieren Sie für jedes XML-Element `DataTypes` die folgenden XML-Attribute:
- **MPPTypes:** Geben Sie den Microsoft Project-Datentyp ein, der dem Alfabet -Datentyp zugeordnet werden soll, der im folgenden XML-Attribut angegeben ist: `ADIFTypes`
 - **ADIFTypes:** Geben Sie den Alfabet -Datentyp ein, der dem im XML-Attribut `MPPTypes` angegebenen Microsoft Project-Datentyp zugeordnet werden soll.



Beachten Sie, dass keine ADIF-Konfiguration erforderlich ist, um Daten aus/nach Microsoft Project zu importieren oder zu exportieren.

- 6) Fügen Sie dem Root-XML-Element `MicrosoftProjectPlanConfig` ein XML-Element `ResourceMatchOptions` hinzu, um die Übereinstimmungsregeln für den Import zu definieren, die möglicherweise verwendet werden können, um die Ressourcen von Microsoft® Project mit Personalbedarfen, Organisationen oder Personen in Alfabet abzugleichen, wenn eine Personalanforderung auf der Grundlage einer Projektaufgabe in Microsoft Project in Alfabet erzeugt wird. Sie müssen mindestens eine Eigenschaft oder ein übereinstimmendes Muster für jede Klasse definieren. Die angegebenen Eigenschaften und Übereinstimmungsmuster stehen im Bereich **Übereinstimmungsregeln importieren** des Editors **Microsoft Project-Methodik** auf der Alfabet -Benutzeroberfläche zur Auswahl und können nach Bedarf einer Methodik zugewiesen werden. Definieren Sie für das XML-Element `ResourceMatchOptions` die folgenden untergeordneten XML-Elemente:
- **Class:** Erzeugen Sie für jede der folgenden Klassen ein XML-Element `Class: Skill`, `Person` und `OrgaUnit`. Definieren Sie für jedes XML-Element `Class` die folgenden XML-Attribute:
 - **Name:** Geben Sie den technischen Namen der Alfabet -Klasse ein (z. B.: `Skill`, `Person` und `OrgaUnit`). Der technische Name muss richtig geschrieben sein. Das XML-Attribut `Name` darf nicht länger als 64 Zeichen sein.
 - **Properties:** Definieren Sie eine durch Kommas getrennte Liste der Eigenschaften der relevanten Alfabet -Klasse, die möglicherweise Ressourcen in Microsoft® Project zugeordnet werden können.
 - **MatchPatterns:** Erstellen Sie ein oder mehrere XML-Elemente `Pattern`, um die Muster anzugeben, die für die Übereinstimmung mit Eigenschaften verwendet werden sollen, die nicht im XML-Attribut `Properties` angegeben sind. Geben Sie für jedes XML-Element `Pattern` Folgendes an: Nehmen Sie mehrere Eigenschaften in Alfabet und verketteten Sie sie mit Trennzeichen, um sie mit Zeichenfolgen in MS abzugleichen.

- **Name:** Geben Sie den Namen der Übereinstimmungsregel ein. Das XML-Attribut `Name` darf nicht länger als 64 Zeichen sein. Beachten Sie, dass bei der Angabe der Microsoft Project-Methodik Fehler auftreten können, wenn die Namen der übereinstimmenden Muster mehr als 64 Zeichen enthalten.
 - **Properties:** Geben Sie die Eigenschaften an, aus denen das Muster bestehen soll, das für die Suche nach den Personalbedarfen (oder Personen oder Organisationen) in der Alfabet-Datenbank verwendet werden soll, die mit den Ressourcen in MS Project übereinstimmen.
 - **Delimiter:** Geben Sie ein oder mehrere Trennzeichen an, die in den Ressourcen verwendet werden. Nur Ressourcen mit den angegebenen Trennzeichen gelten als Ressourcen, für die übereinstimmende Personalbedarfe gefunden werden können.
- 7) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste von Alfabet Expand auf die Schaltfläche **Speichern**  .

Angeben eines Assistenten für die Interoperabilität mit Microsoft Project

Um MPP-Dateien in Alfabet zu importieren, muss ein Assistent konfiguriert und der Klasseneinstellung für die Klasse „Projekt“ zugewiesen werden. Dies ist erforderlich, um Projekte über MS Project zu importieren. Dem ersten Assistentenschritt wird ein zusätzlicher Schritt vorangestellt, mit dem der Import ausgelöst werden kann, um das neue Projekt in Alfabet zu erstellen. Für die Klasseneinstellung muss das Attribut **Standard-Editortyp** auf `Wizard` gesetzt sein, und der entsprechende Assistent muss für das Attribut **Assistent** angegeben werden.

Wenn der Anwender in den entsprechenden Ansichten die Optionen **Projekt aus Microsoft Project-Datei importieren** oder **Projekt vom Microsoft Project-Server importieren** auswählt, wird dem entsprechenden Assistenten automatisch ein zusätzlicher Assistentenschritt vorangestellt, der den Anwender auffordert, die MPP-Datei, die gewünschte MS Project-Methodik und den Standardpersonalbedarf auszuwählen, der für alle Ressourcen in der MPP-Datei verwendet werden soll, die nicht auf der Grundlage der in der MS Project-Methodik verfügbaren Übereinstimmungsregeln zugeordnet werden können. Wenn Sie im vorangestellten Assistentenschritt auf die Schaltfläche **Weiter** klicken, wird das Projekt erzeugt und der Assistent wechselt zum konfigurierten ersten Assistentenschritt, sodass das neue Projekt bearbeitet werden kann. Weitere Informationen über das Konfigurieren der Projektmanagement-Funktionalität sowie über die Spezifikation von Projektstereotypen über das XML-Objekt **ProjectManager** finden Sie unter *Konfigurieren der Projektmanagement-Funktionalität* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Angeben eines Release-Status für die Interoperabilität mit Microsoft Project

Stellen Sie sicher, dass Release-Statusdefinitionen für die Klasse **Microsoft Project-Methodik** definiert wurden. Erzeugen Sie im XML-Objekt **ReleaseStatusDefs** ein XML-Element **ReleaseStatusDef** für die Klasse `MicrosoftProjectPlan_Mappings`. Stellen Sie sicher, dass ein XML-Attribut `ApprovedStatus` definiert ist. Für den Import oder Export von MS Project-Projekten können nur MS-Projektmethodiken ausgewählt werden, für die der genehmigte Status definiert wurde.

Weitere Informationen zum Definieren des XML-Objekts **ReleaseStatusDefs** finden Sie im Abschnitt *Konfigurieren von Release-Status-Definitionen für Objektklassen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Kapitel 14: Konfigurieren der Interoperabilität mit einem Übersetzungsdienst

Mit dem XML-Objekt **AlfaTranslationServicesConfig** können Sie die Interoperabilität mit einem Übersetzungsservice für die Bereitstellung einer automatisierten Übersetzung für benutzerdefinierte Zeichenfolgen in den Vokabularen, die im Kontext der Lösungskonfiguration definiert wurden, sowie für die Objektdateiübersetzung im Kontext der Editoren und Assistenten auf der Alfabet -Benutzeroberfläche angeben. Weitere Informationen zur benötigten Konfiguration finden Sie in den Abschnitten *Übersetzen von benutzerdefinierten Zeichenfolgen über die automatisierte Übersetzungsfunktion* und *Konfigurieren der Übersetzung von Objektdateien* im Kapitel *Lokalisierung und Mehrsprachenunterstützung für die Alfabet -Benutzeroberfläche* des Referenzhandbuchs *Konfigurieren von Alfabet mit Alfabet Expand*.

Folgende Voraussetzungen müssen erfüllt sein, um Übersetzungen über die automatisierte Übersetzungsfunktion abrufen zu können:

- Ihr Unternehmen muss über eine gültige Lizenz für einen der folgenden Übersetzungsservices verfügen:
 - Google Translate®
 - AWS Translate®
 - DeepL®-Übersetzer
 - Microsoft® Azure® Translate Text
- Das XML-Objekt **AlfaTranslationServicesConfig** muss konfiguriert sein, und die Verbindung zum Übersetzungsservice muss aktiviert sein. Weitere Informationen über die Konfiguration des XML-Objekts **AlfaTranslationServicesConfig** finden Sie im Referenzhandbuch *API-Integration mit Drittanbieterkomponenten* im Abschnitt [Konfigurieren der Interoperabilität mit einem Übersetzungsdienst](#).
- Die Voraussetzungen für die Aktivierung der Rest-API müssen erfüllt sein. Informationen hierzu finden Sie im Referenzhandbuch *Alfabet RESTful API*.
- Es muss ein Anwender im Unternehmen angegeben werden, um selbstreflektierende Ereignisse auszuführen, und der Alfabet-Server muss gestartet werden. Informationen hierzu finden Sie im Abschnitt *Festlegen eines Anwenders als selbstreflektierenden Anwender zur Ausführung von Ereignissen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- Der Alfabet-Server muss ausgeführt werden und eine Verbindung mit dem Internet herstellen können.



Das XML-Objekt enthält normalerweise eine Beispielformatierung. Zusätzlich ist über das Attribut **XML-Vorlage** im Attributraster des XML-Objekts **AlfaTranslationServicesConfig** eine Vorlage verfügbar. Sie können die Vorlage in das XML-Objekt kopieren, um die Konfiguration nicht manuell schreiben zu müssen. In diesem Fall bearbeiten Sie die nachfolgend beschriebenen XML-Elemente. Im Folgenden finden Sie eine Beschreibung der Konfiguration von Grund auf.

So bearbeiten Sie das XML-Objekt **AlfaTranslationServicesConfig**:

- 1) Wechseln Sie zur Registerkarte **Präsentation**, und erweitern Sie den Ordner **XML-Objekte** sowie anschließend den Ordner **Integrationslösungen**.
- 2) Klicken Sie mit der rechten Maustaste auf **AlfaTranslationServicesConfig**, und wählen Sie **XML bearbeiten...** aus. Das XML-Objekt **AlfaTranslationServicesConfig** wird geöffnet.

3) Optional können Sie die Integrationsschnittstelle dazu konfigurieren, Anforderungen an einen Übersetzungsservice über einen Proxyserver zu senden. Um einen Proxyserver zu definieren, fügen Sie dem XML-Element **AlfaTranslationServicesConfig** ein untergeordnetes XML-Element `Proxy` hinzu, und definieren Sie die folgenden XML-Attribute für das XML-Element `Proxy`:

- `url`: Definieren Sie die URL des Proxyservers.
- `user`: Geben Sie bei Bedarf den Anwendernamen für den Zugriff auf den Proxyserver ein. Der Domänenname für die Authentifizierung wird separat über das XML-Attribut „Domäne“ definiert und muss nicht als Teil des Anwendernamens angegeben werden.
- `password`: Geben Sie bei Bedarf das Passwort für den Zugriff auf den Proxyserver ein.
- `domain`: Definieren Sie bei Bedarf den Domännennamen, der als Teil des Anwendernamens zur Authentifizierung am Proxyserver verwendet werden soll.



Hinweis: Die Servervariablen lesen den Wert des XML-Attributs zur Laufzeit aus der Server-Alias-Konfiguration der Alfabet-Webapplikation aus, wenn eine Verbindung zur Integrationslösung hergestellt wurde. Informationen zur Konfiguration der Servervariablen im Server-Alias finden Sie unter [Konfigurieren von Servervariablen für Integrations- und Interoperabilitätslösungen](#).

4) Für jeden Übersetzungsservice, den Sie im XML-Objekt definieren möchten, erstellen Sie ein untergeordnetes XML-Element `TranslationServiceInfo`.

5) Definieren Sie für jedes XML-Element `TranslationServiceInfo` die folgenden XML-Attribute:

- `Name`: Geben Sie einen Namen für die Verbindung zum Übersetzungsservice ein.
- `IsActive`: Geben Sie „true“ ein, um die Interoperabilität mit den Übersetzungsservice zu aktivieren. Ein Übersetzungsservice kann jederzeit aktiv sein. Wenn mehrere Verbindungen auf `IsActive=true` eingestellt sind, wird die erste aktive Verbindung verwendet.
- `Type`: Geben Sie je nach dem Übersetzungsservice, zu dem Sie eine Verbindung herstellen möchten und für den Sie eine gültige Lizenz besitzen, eine der folgenden Optionen ein:
 - Google
 - AWS
 - DeepL
 - Azure
- `ServiceType`: Sie müssen `AzureCognitive` angeben, wenn das XML-Element `Type` auf `Azure` gesetzt ist. Falls dieses XML-Attribut nicht angegeben ist, kommt es zu einem Fehler, wenn Interoperabilität mit Microsoft® Azure® Translate Text angegeben ist.
- `Timeout`: Definieren Sie das Zeitlimit für HTTP-Anforderungen für die Datenverbindung.
- `AccessKey`: Der Zugriffsschlüssel für die Verbindung mit dem Übersetzungsservice.
- `SecretKey`: Der geheime Schlüssel für die Verbindung mit dem Übersetzungsservice.

6) In the toolbar, click the **Save** button to save the XML definition.

Index

ADIF-Exportschema	
Assistent für ServiceNow-Exportschemata	122
Assistent für ServiceNow-Importschemata	107
ServiceNow-Integration	122
ADIF-Importschema	
Amazon Web Services-Schnittstelle	96
Jira	135
ServiceNow-Integration	107
ALFABET_TECHNOPEDIA_UPDATE	30
Amazon Web Services-Schnittstelle	
ADIF-Importschema	96
Ausführung	93
Konfigurieren der Datenintegration	96
Konfigurieren der Verbindung	93
Proxyserver	95
Übersicht	93
XML-Objekt	93
AmazonWebServicesConfig	93
Anbieterprodukt	
Technopedia	30
Apigee-Integration	
Proxyserver	83
Assistent	
ServiceNow-Integration ADIF-Schema	107, 122
Ausschließen von Daten	
ADIF-Import für ServiceNow	110
Datenintegration	
Jira	126
Datentyp	
Export nach ServiceNow	120
Importieren aus ServiceNow	105
Importschema	
JIRA	126
Technopedia	30
Jira	
ADIF-Importschema	135
Export nach Alfabet	128
Importieren von Alfabet-Daten	141
Konfigurieren des Datenimports	135
JIRAConfig	126

Jira-Integration	126
Komponente	
Technopedia	30
Maximale Ergebnisanzahl	
Datenexport nach ServiceNow	119
Datenimport aus ServiceNow	104
PageSize	
Datenexport nach ServiceNow	119
Datenimport aus ServiceNow	104
Proxyserver	
Amazon Web Services-Schnittstelle	95
Apigee-Integration	83
ServiceNow-Integration	106, 121
ServiceNowImportConfig	101
ServiceNow-Integration	
ADIF-Exportschema	122
ADIF-Importschema	107
Ausführung	100, 112
Ausschließen von Daten aus dem ADIF-Import	110
Datentypdefinition	105, 120
Deaktivieren von ADIF-Eintrag	110
Konfigurationsübersicht	100, 112
Konfigurieren der Datenintegration	107, 122
Konfigurieren einer Datenverbindung	101, 117
Proxyserver	106, 121
Übertragungsparameter	104, 119
XML-Objekt	101, 117
Technopedia	30
Teilweises Deaktivieren	
ADIF-Import für ServiceNow	110
Timeout	
Datenexport nach ServiceNow	119
Datenimport aus ServiceNow	104
XML-Objekt	
AmazonWebServicesConfig	93
JIRAConfig	126, 141
JIRAConfig - Import	128
ServiceNowExportConfig	117
ServiceNowImportConfig	101