Syncler-Tutorial

Release 0.1

Georg Bergholz

14.03.2024

Inhaltsverzeichnis

In diesem Tutorial wird das Syncler Projekt dokumentiert.

Syncler ist eine Plattform zur Realisierung von verschiedenen Synchronisations- und Businessaufgaben.

Für einen Überblick können Sie die Projektseite https://www.syncler.de besuchen.

Bemerkung: Die Dokumentation wird gerade entwickelt.

Verbindungen, Prozesse und Datendienste sind Elemente im Syncler für die Realisierung Ihrer Aufgaben. Daneben gibt es noch allgemeine Elemente für die Koordination, Überwachung und Ausführung.

KAPITEL 1

Verbindungen

Eine Verbindung stellt die Anbindung eines externen Systems oder eine Funktion bereit. Sie speichert Zugangsdaten und Einstellungen und realisiert eine einheitliche Schnittstelle für andere Elemente im Syncler. Neben klassischen System wie CRM oder ERP kann sich hinter einer Verbindung auch der Eingang für Daten, z.B. Push-Nachrichten oder Dateien und der Ausgang von Daten, z.B. Serienbriefe oder Emails verbergen.

Zu den Standardaufgaben zählen die Autorisierung, das Lesen von Daten, das Schreiben von Daten und das Ermitteln eines Datenschemas. Beim Lesen von Daten können zwei Methoden unterschieden werden, die aber nicht jede Verbindung auch implementieren muss. Daten können auf der Basis eines Schemas oder auf der Basis einer Abfrage gelesen werden. Auch beim Speichern von Daten können zwei Methoden unterschieden werden, die ebenfalls nicht von jeder Verbindung bereitgestellt werden müssen. Daten werden Schema-basiert einzeln oder als Bulk gespeichert.

Außerdem stellen manche Verbindungen zusätzliche Funktionen, wie z.B. eine Preisfindung bereit.

Durch das Speichern einer neuen Verbindung oder das Speichern einer bestehenden Verbindung mit aktivierter Checkbox „Schema aktualisieren“ ruft Syncler das Schema des Systems ab und speichert es. Dieses Datenschema ist die Arbeitsgrundlage für die Konfiguration von Schema-basierten Prozessen und wird für eine bessere Performance zwischengespeichert.

Sollten sich das Datenschema ändern, z.B. durch das Anlegen neuer Felder, müssen Sie diese Änderung über die Aktualisierung des Verbindungsschemas bekannt machen, damit Sie diese verwenden können.

In der Detailbeschreibung zu jeder Verbindung finden Sie Angaben zu den folgenden Grundfunktionen.

Schema ermitteln

Wie stellt die Verbindung ein Datenschema bereit und was ist enthalten.

Lesen von Schema-basierten Daten

Wie kann die Verbindung Quelldaten auf der Basis eines bekannten Schemas für Prozesse ermitteln. Diese Abfragen werden für die Ausführung eines Prozesses mit der Verbindung als Quelle benötigt. Auch innerhalb von Transformation kann das Lesen von Daten angewendet werden.

Lesen von Abfrage-basierten Daten

Wie kann die Verbindung Quelldaten auf der Basis einer Abfrage für Prozess ermitteln.

Schreiben von Daten

Wie werden einzelne Datensätze geschrieben.

Schreiben von Bulk-Daten

Wie können Sammlungen von Daten geschrieben werden.

Die folgenden Verbindungen stehen im Syncler zur Verfügung.

1.1 Sage Bäurer b7

Bei Sage Bäurer b7 handelt es sich um eine ERP-Software für Produktion und Handel im Mittelstand.

Diese Verbindung unterstützt die bidirektionale Synchronisation von Kunden, Lieferanten, Personen und Anschriften. Außerdem könnten Teile, Auswahllisten und Metadaten abgerufen, sowie SQL Select-Anweisungen ausgeführt werden.

Die Anbindung erfolgt mittels Azure Service Bus. Damit dies genutzt werden kann, muss das ERP-System entsprechend konfiguriert werden.

Die Besonderheit bei diesem Verfahren ist die entkoppelte Kommunikation. Daten können nicht direkt abgefragt, sondern müssen per Nachricht angefordert werden. Als Resultat wird die Antwort als Nachricht abgestellt und muss abgerufen werden. Aus diesem Grund sind eigene Prozesse für die Synchronisation erforderlich.

Es stehen spezifische Prozesse für Sage CRM und Zoho CRM inklusive Vorlagen zur Verfügung. Weitere Systeme können teilweise per Universalprozesse angebunden werden, wobei nicht alle Parameter durch die Verbindung ausgewertet werden können, wie z.B. Filterangaben. Auch kann mit einem Universalprozess kein begonnener Initialsync fortgesetzt werden. Dieser muss immer komplett neugestartet werden. Das Schreiben mit Universalprozessen ist nicht möglich, da die Verbindung kein Resultat liefern kann und dadurch die Aktion nicht als erfolgreich interpretiert wird.

1.1.1 Funktionen

Schema ermitteln

Die Ermittlung des Schemas erfolgt mit einer Anfragenachricht. Die Antwort enthält die verfügbaren Objekte und deren Felddefinition. Speziell das Kunden- und Lieferantenobjekt werden noch weiter unterteilt für Personen und Anschriften aufbereitet. Außerdem können vorhandene Auswahllisten separat zum Schema (awl) abgefragt werden. Diese können dann z.B. in Transformationen für die Wertumwandlung genutzt werden.

Lesen von Schema-basierten Daten

Daten können initial auf einem Schema basierend angefordert werden. Per Prozess-Parameter wird dieser Vorgang eingeleitet. Sollte dies mehrmals erfolgen, wird vor jedem Anfordern die Warteschlange geleert. Dies kann einige Zeit in Anspruch nehmen. Eine weiterer Prozessparameter ermöglich auch die Wiederaufnahme der initialen Übertragung, ohne die Warteschlange zu leeren. Änderungen werden durch das ERP in einem eigenen Topic abgestellt und von dort abgefragt.

Datensätze können auch einzeln angefordert werden. Je Anforderung von Daten erfolgt mit einer entkoppelten Nachricht. Die Anforderung wird als Nachricht versendet. Danach wird eine definierte Zeit auf die Antwort gewartet. Sollte das ERP die Nachrichten nicht verarbeiten, kann es zum Timeout kommen.

Jedes Schema stellt einen UseCase dar. Mehrfach gelesene Nachrichten im Service Bus führen zu einer Dead-Letter-Situation. Damit dies vermieden wird, werden alle verfügbaren Daten beim Lesen von Nachrichten verarbeitet, unabhängig vom jeweiligen Prozess, der das Lesen ausgelöst hat. Gelesene Nachrichten werden deshalb in den Änderungsspeicher übernommen, falls es einen definierten Prozess für dieses Objekt gibt. Die Prozesse arbeiten von da an über den Änderungsspeicher, um z.B. eine individuelle Fehlerbehandlung zu ermöglichen.

Bei OUT-Nachrichten zu Änderungen kann dem Element s_aktion (Ausprägungen: NONE, INS, UPD, DEL) die Art der Änderung entnommen werden.

Lesen von Abfrage-basierten Daten

Der UseCase SQLQuery ermöglich das Ausführen von SELECT-Anweisungen. Diese werden vom ERP geprüft und ausgeführt.

Schreiben von Daten

Für das Schreiben von Daten werden ebenfalls Nachrichten erzeugt und abgestellt. Da zu diesem Zeitpunkt nicht sicher ist, ob und wann die Aktion ausgeführt wird, wird sie unter Vorbehalt als erfolgreich gewertet. Am Ende einer Prozessausführung werden Responses des ERPs aus der Warteschlange abgerufen. Die Antworten werden mit den aktuellen Datensätzen abgeglichen. Sollte die Antwort zu einer früheren Nachricht gehören, wird das als zusätzliches Resultat in der Prozessausführung behandelt. Speziell bei der Neuanlage wird über die CorrelationID der Nachricht eine Beziehung zum Quelldatensatz hergestellt, da die Kommunikation mit dem ERP nicht direkt erfolgt. Das Löschen wird nicht unterstützt.

Neue Datensätze erhalten eine Datenabbildung mit einer -1 als Ziel-ID. Dies ist eine Abbruchbedingung für eine erneute Übertragung, um die Anlage von Dubletten zu verhindern.

Schreiben von Bulk-Daten

Diese Funktion wird nicht unterstützt.

1.1.2 Einstellungen

Folgende Einstellungen müssen bereitgestellt werden.

Service Bus Endpunkt
Die URL des Nachrichtendienstes wird mit der Kundennummer vervollständigt.

```
sb://sb-#####-b7-crm.servicebus.windows.net/
```

Kundenkennzeichen
Die Eingabe folgt diesem Schema und beginnt mit der Kundennummer.

```
#####.xxxxx
```

Mandant
Damit wird an der Verbindung gekennzeichnet, welcher ERP-Mandant angesprochen werden soll. Für eine Mehrmandanten-Integration wird je Mandant eine Verbindung eingerichtet.

SharedAccessKey Name
Dieser Wert gehört zu den Zugangsdaten.

SharedAccessKey Lesen
Dieser Wert gehört zu den Zugangsdaten.

SharedAccessKey Schreiben
Dieser Wert gehört zu den Zugangsdaten.

SharedAccessKey Initialisierung
Dieser Wert gehört zu den Zugangsdaten.

Timeout Verbindung
Mit dieser Angabe kann das Standardtimeout für die Azure Verbindung angepasst werden. Die Angabe erfolgt in Millisekunden und das Minimum und Standardwert sind 100000. Eine Anpassung des Standardwertes ist nur dann erforderlich, wenn Abfragen mit einem Timeout-Fehler abgebrochen werden.

Timeout Nachrichten
Mit dieser Angabe kann das Standardtimeout für die Azure-Abfragen angepasst werden. Die Angabe erfolgt in Millisekunden und das Minimum und Standardwert sind 5000.

1.1.3 UseCases

Für die Kommunikation sind verschiedene UseCases definiert. Neben den verfügbaren Objekten wird damit auch die Art und das System festgelegt.

```
UC.SageCRM.Schema.REQ
UC.SageCRM.AllCustomers.REQ
UC.Sageb7.AllCustomers.RESP
UC.SageCRM.Selection.REQ
UC.SageCRM.Customer.REQ
UC.Sageb7.Customer.RESP
UC.SageCRM.SqlQuery.REQ
```

Folgende Objekte werden unterstützt.

- Customer - kunde
- Supplier - lieferant
- Part - teil
- ProductFamily - produktgruppe
- Project - projekt
- SalesOpp - vc
- Invoice - rechnung
- Order - auftrag
- Offer - angebot
- Selection - awl

Das Objekt Rechnung wird wegen seiner zusätzlichen Vorgangsunterteilung in mehrere Objekt je Vorgang durch die Verbindung aufgetrennt.

Speziell zu Belegen gibt es noch weitere Funktionalitäten. Siehe *Sage Bäurer b7 Belege*

1.2 Microsoft Dynamics Business Central

Microsoft Dynamics Business Central ist ein Warenwirtschaftssystem für kleine und mittlere Unternehmen.

Diese Verbindung unterstützt die bidirektionale Synchronisation von Entitäten der REST API, welche im ERP-System konfiguriert werden können.

Die Synchronisation mit CRM-Systemen verwendet in der Regel die Kontakte als Basis. Diese gehören nicht in allen Installationen zu den Standard-Endpunkten und müssen über eine API Page bereitgestellt werden. Für eine Kombination mit Kundendaten werden die Kontakt-Geschäftsbeziehungen ebenfalls als Endpunkt benötigt. Hier sind die Objektname „Contact“ und „ContactBusinessRelation“ durch die spezifischen Prozesse vorgeschrieben.

Abfragen sind nur bei lokalen Installationen möglich und über SQL-Zugriffe implementiert. Wenn Datendienste oder Abfrage-Prozesse zum Einsatz kommen sollen, müssen diese Zugangsdaten angegeben werden. Dies kann auch mittels der Syncler-SQL-Bridge abgesichert werden. (*Die SQL- und Dateibrücke*)

Es stehen spezifische Prozesse für die Synchronisation mit Sage CRM und CAS inklusive Vorlagen zur Verfügung. Dazu zählt auch die Anlage und Verknüpfung von Debitoren oder Kreditoren zu vorhandenen Kontakten. Weitere Systeme und Entitäten können über Universalprozesse angebunden werden.

1.2.1 Funktionen

Schema ermitteln

Das Schema der Verbindung wird über die REST API ermittelt. Hierfür wird der $metadata Aufruf ausgeführt und ausgewertet. Alle bereitgestellten Entitäten stehen für alle Operationen zur Verfügung. Geschachtelte Daten stehen zu „salesquote“, „salesorder“, „purchaseinvoice“ und „salesinvoice“ zur Verfügung. Die Änderungsinformation zur Entität wird über den Feldnamen identifiziert.

Lesen von Schema-basierten Daten

Das Lesen von Datensätzen erfolgt über die REST API und wird durch die angegebene Firma eingeschränkt. Zusätzlich können über den Prozess noch weitere Filter im OData-Syntax definiert werden. Spezifische Prozesse unterscheiden automatisch Firmen- und Personenkontakte. In Universalprozessen muss dies per Filter realisiert werden. Type eq ‚Company‘

Lesen von Abfrage-basierten Daten

Das Lesen von Abfrage-basierten Daten wird per SQL-Verbindung realisiert.

Schreiben von Daten

Das Schreiben von Datensätzen erfolgt über die REST API und wird durch die angegebene Firma eingeschränkt. Das Löschen von Datensätzen wird unterstützt. Das Speichern von Positionen in geschachtelten Daten erfolgt mit separaten Aufrufen.

Schreiben von Bulk-Daten

Diese Funktion wird nicht unterstützt.

1.2.2 Einstellungen

Folgende Einstellungen müssen bereitgestellt werden.

API Basis-URL

Die Basis-URL für die REST API. Die Version muss hier angegeben werden. Sollten weitere Endpunkte angelegt werden, müssen diese der selben API Version zugeordnet werden.

```
http://<Server>:<WebServicePort>/<ServerInstance>/api/v1.0
```

Benutzername

Dieser Wert gehört zu den Zugangsdaten.

Passwort

Dieser Wert gehört zu den Zugangsdaten.

Firma

Dies ist der Name des ERP-Mandanten.

Kundennummer statt Kunden-ID als Schlüssel verwenden

Bei Upgrades kann statt der ID-Variante noch die No-Variante für die Identifikation von Datensätzen im Einsatz sein. Damit in diesem Fall eine Kundenverarbeitung möglich ist, muss dieser Parameter auf Ja gestellt werden.

Artikelnummer statt Artikel-ID als Schlüssel verwenden

Bei Upgrades kann statt der ID-Variante noch die No-Variante für die Identifikation von Datensätzen im Einsatz sein. Damit in diesem Fall eine Artikelverarbeitung möglich ist, muss dieser Parameter auf Ja gestellt werden.

Artikelkategorie Code statt Artikelkategorie-ID als Schlüssel verwenden

Bei Upgrades kann statt der ID-Variante noch die No-Variante für die Identifikation von Datensätzen im Einsatz sein. Damit in diesem Fall eine Artikelgruppenverarbeitung möglich ist, muss dieser Parameter auf Ja gestellt werden.

Belegnummer statt Beleg-ID als Schlüssel verwenden

Bei Upgrades kann statt der ID-Variante noch die No-Variante für die Identifikation von Datensätzen im Einsatz sein. Damit in diesem Fall eine Belegverarbeitung möglich ist, muss dieser Parameter auf Ja gestellt werden.

1.3 CAS genesisWorld und SmartWe

Diese beiden CRM-Systeme können durch ihre fast identische REST API mit einer Verbindung angesteuert werden. Die geringfügigen Unterschiede werden durch die Verbindung behandelt, weshalb per Parameter festgelegt werden muss, um welches System es sich handelt.

Diese Verbindung unterstützt die bidirektionale Synchronisation von Datentypen der REST API und kann Emails und Dokumente archivieren. Außerdem besteht die Möglichkeit Bulk-Importe auszuführen

Schema-basierte und Abfrage-basierte Datenzugriffe werden über die REST API implementiert. Für Abfragen wird der spezifische SQL-Syntax von CAS verwendet, der zusätzliche Funktionen für Dossier, Verknüpfungen und Filter bereitstellt.

Es stehen spezifische Prozesse für die Synchronisation mit Sage b7, Microsoft Dynamics Business Central, Sage 100, Sage X3 und Mailing-Anbietern inklusive Vorlagen zur Verfügung. Weitere Systeme und Entitäten können über Universalprozesse angebunden werden.

1.3.1 Funktionen

Schema ermitteln

Das Schema der Verbindung wird über die REST API ermittelt. Basis sind die verfügbaren Datenobjekte. Zusätzlich wird das Schema noch für die direkte Abfrage von Benutzern und Verknüpfungen erweitert.

Berechtigungen und Verknüpfungen können zu jedem Datentyp festgelegt werden.

Das Schema für DOCUMENT und EMAILSTORE wird um das Feld FileContent erweitert. Base64 Daten, die hier zugewiesen werden, werden in CAS abgelegt bzw. archiviert.

Das Schema von APPOINTMENT, TODO und GWPHONECALL wird um die Listen LinkToAddress, LinkToGWOpportunity bzw. Participants erweitert. Speziell Participants ist dabei eine geschachtelte Liste. Diese Daten stehen auch beim Lesen zur Verfügung.

Das Schema für GWOPPORTUNITY und BSVOUCHER wird um eine geschachtelte Positionsliste erweitert, die in der Sortierung die Gruppen berücksichtigt.

Das Schema von GWDISTRIBUTION wird um eine geschachtelete Liste für ADDRESS-Verweise erweitert.

Lesen von Schema-basierten Daten

Mit Ausnahme von Einzeldatensätzen werden alle Abfragen für Schema-Objekte per query ausgeführt und ergänzt. Für genesisWorld wird das Statement automatisch um TEAMFILTER(Objektname; CASLOGGEDINUSER, CASPUBLICRECORDS, CASEXTERNALACCESS) erweitert. Bei der Verwendung von Abfrage-Prozessen muss dies manuell erfolgen, da das Ergebnis ansonsten eine Vervielfälltigung der Daten sein kann.

Lesen von Abfrage-basierten Daten

Auch für diese Art des Lesens wird die query Funktion der REST API genutzt. Hierbei wird allerdings kein bekanntes Schema-Objekt gefüllt, sondern das Resultat dynamisch generiert. Eine Anreichung des Ergebnisses durch Berechtigungen, Verknüpfungen oder Listen ist deshalb nicht möglich.

Schreiben von Daten

Das Schreiben von Daten erfolgt über die REST API, wobei die Verbindung selbstständig eine Etag ermittelt. Auch das Löschen von Datensätzen wird unterstützt. Direkte Verknüpfungen können ebenfalls geschrieben oder gelöscht werden. Das Schreiben von Benutzern ist nicht möglich. Berechtigungen, Verknüpfungen und Listen an Schema-Objekten können geschrieben werden. Das Löschen von Berechtigungen oder Verknüpfungen an Schema-Objekten wird nicht unterstützt.

Schreiben von Bulk-Daten

Da diese Funktion nicht über die REST API möglich ist, sondern ausschließlich in der SOAP Schnittstelle zur Verfügung steht, kommt hier ein besonderes Verfahren zum Einsatz. Es steht eine CAS SOAP Bridge zur Verfügung, die per Veröffentlichungspaket in einen Windows Internet- Informationsdienst eingebunden werden kann. Die Verbindungsdaten zu dieser API werden in den Einstellungen festgelegt. Der Bulk-Import wird in definierte Pakete unterteilt und an die API übergeben, welche diese dann an CAS übermittelt. Dabei können individuelle Verknüpfungen geschrieben und Fehler behandelt werden. Lediglich Berechtigungen können nur gesammelt vergeben werden, wobei einfach die Berechtigungen des ersten Datensatzes für alle übernommen werden. Damit das Anlegen von Datensätzen auch generierte IDs ermitteln und für Datenabbildungen verwenden kann, muss ein zusätzliches GUID-Feld (Textfeld) angelegt und angegeben werden. Hier wird automatisch eine generierte GUID für die Wiedererkennung des Datensätzes eingetragen. Die CAS Zugangsdaten werden dynamisch verschlüsselt an die API übergeben.

1.3.2 Einstellungen

URL REST API

Die Url der REST API. Für SmartWe ist dies https://api.smartwe.de/SmartWe/rest/v6.0

Benutzername

Dieser Wert gehört zu den Zugangsdaten.

Passwort

Dieser Wert gehört zu den Zugangsdaten.

Datenbank

Der Datenbank- bzw. Mandantenname

SmartWe

Ist das Zielsystem SmartWe.

Bridge Url

Optional die Url zur CAS SOAP Bridge.

CAS SOAP Url

Optional die SOAP Url. Für SmartWe ist dies https://api.smartwe.de/SmartWe/services

Bridge Timeout

Optional das Timeout für Aufrufe der Bridge Url. Sollte es bei der Bulk-Verarbeitung zu Timeout-Meldungen kommen, kann hier ein abweichender Wert festgelegt werden. Die Eingabe erfolgt in Millisekunden.

Maximale Anzahl von Datensätzen je Aufruf

Bulk-Import erfolgen in Paketen. Je nach Leistungsfähigkeit und Datenmenge können sich unterschiedliche Beschränkungen ergeben. Werte zwischen 100 und 200 sind in der Praxis üblich. Das Timeout hat auch Einfluss auf diesen Wert.

Feldname Massen-Neuanlage

Optional der Feldname für die Wiedererkennung von angelegten Datensätzen für alle Datentypen. Ohne diesen Wert können neue IDs aus dem Bulk-Import nicht ermittelt und bereitgestellt werden. Das Feld muss als Textfeld manuell angelegt werden.

Abfrage von Berechtigungen deaktivieren

Hiermit kann das Lesen von Berechtigungen deaktiviert werden. Das Schreiben steht dennoch zur Verfügung und überschreibt. Da das Lesen der Berechtigungen zusätzliche Abfragen erfordert, kann hiermit die Lesegeschwindigkeit erhöht werden.

Prüfung von Pflichtfeldern deaktivieren

Hiermit werden keine Pflichtfelder im Schema definiert und im späteren Verlauf geprüft. Dies ist nur in Ausnahmen erforderlich.

1.3.3 Besonderheiten

Das Schema der verfügbaren Datenobjekte wird automatisch um die Objekte Permission, LinkTo und LinkFrom erweitert. Diese unsortierten Listen bieten die Möglichkeit Berechtigungen und Verknüpfungen zu setzen oder Berechtigungen auszulesen. Da es sich hier um 1:n verknüpfte Informationen handelt, die unsortiert bereitgestellt werden, müssen beim Lesen alle Gruppen berücksichtigt werden. Beim Aktualisieren von Berechtigungen wird über die RIGHTOWNERGGUID nach vorhandenen Zuordnungen gesucht. Die Verknüpfungen werden durch einen separaten API Aufruf festgelegt.

Das Zuweisen einer leeren Guid für Link-Ziele überspringt die Anlage.

Wenn Datensätze ohne Berechtigungen angelegt werden, stehen diese in der Anwendung den Benutzern nicht zur Verfügung. Die Berechtigung ACCESSRIGHT = 32768 und RIGHTOWNER = 00000000000000000000000000000000 ermöglicht den Vollzugriff für alle Benutzer und wird in den Vorlage als Standardwert verwendet. Weitere Permission Level sind Lesen = 64, Bearbeiten = 2048 und Löschen = 8192.

Für die Archivierung von Emails kann das Objekt „EMAILSTORE“ verwendet werden. Dieses verfügt über ein Feld „FileContent“. Wenn Sie dort eine Email im EML-Format als Base64 Zeichenkette zuweisen, wird die Email im CAS archiviert. Diese Daten lassen sich mit der Email Verbindung bereitstellen. Die restlichen Felder können für zusätzliche Daten, wie z.B. Verknüpfungen verwendet werden. Die Archivierung wird nur mit neuen Datensätzen ausgeführt. Sollte es sich um einen bekannten Datensatz handelt, wird die Archivierung übersprungen.

1.4 Microsoft Graph API

Die Microsoft Graph API ist eine REST-basierte Schnittstelle für den Zugriff auf Microsoft 365. Diese Verbindung realisiert die Anmeldung an der API und stellt ein ausgewähltes Datenschema bereit. Ein Fokus der Verbindung ist der Zugriff und die Synchronisation von Terminen, Aufgaben und Kontakten, wofür spezielle Parameter und Prozesse vorhanden sind.

Für die Einrichtung der Anmeldung ist eine registrierte Anwendung bei Microsoft erforderlich.

Registrieren Sie die Anwendung über das Registrierungsportal https://go.microsoft.com/fwlink/?linkid=2083908. Melden Sie sich mit Ihrem Microsoft Konto an und rufen Sie die Funktion „Neue Registrierung“ auf. Vergeben Sie einen Anwendungsnamen und klicken Sie auf Registrieren.

Danach wird Ihnen die Übersicht zur Registrierung angezeigt. Für die Verbindung benötigen Sie die Anwendungs-ID (Client) und die Verzeichnis-ID (Mandant). Übernehmen Sie die Angaben in die gleichnamigen Felder.

Für die Authentifikation stehen verschiedene Mechanismen zur Verfügung.

Client-Secret

Dies ist eine Anwendungsanmeldung auf der Basis eines erzeugten Geheimnissen. Unter „Zertifikate & Geheimnisse" müssen Sie einen geheimen Clientschlüssel generieren. Wählen Sie dazu die Funktion „Neuer geheimer Clientschlüssel". Stellen Sie die gewünschte Gültigkeit ein und klicken Sie auf Hinzufügen. Den angezeigte Geheimschlüssel müssen Sie direkt in das Feld „Client Geheimschlüssel" übernehmen. Der Schlüssel ist nachträglich nicht mehr einsehbar. Sollte der Schlüssel nicht mehr zur Verfügung stehen oder abgelaufen sein, müssen Sie einen neuen Geheimschlüssel hinzufügen. Bei dieser Methode werden nur Anwendungsberechtigung der API unterstützt. Für den Scope-Wert muss folgendes eingestellt werden: https://graph.microsoft.com/.default

Administrator Refresh-Token

Dies ist eine benutzerbezogene Anmeldung, bei der die Freigabe direkt in der Verbindungskonfiguration erfolgt. Wechseln Sie auf die Registerkarte „Microsoft 365 OAuth 2.0" und starten Sie den Freigabeprozess. Als nächstes müssen Sie eine Anmeldung durchführen und den Zugriff durch die registrierte Anwendung freigeben. Bei diesem Verfahren werden nur delegierte API Berechtigungen unterstützt. Der Scope-Wert ist entscheidend für die angeforderten Berechtigungen. Diese müssen dennoch der Anwendung hinzugefügt werden. Der Scope offline_access wird automatisch hinzugefügt. Für den Zugriff auf Kalendereinträge wird der Scope Calendars.ReadWrite benötigt. Mehrere Scope-Werte werden durch ein Leerzeichen getrennt.

Ein Refresh-Token muss bei Änderung des Benutzerpassworts aktualisiert werden.

Benutzer Refresh-Token

Auch dieses Verfahren ist eine benutzerbezogene Anmeldung, benötigt aber die Unterstützung durch ein externes Portal. Für Sage CRM steht so ein Portal zur Verfügung und kann die Zustimmung und Refresh-Token von mehreren Benutzern einholen. Das externe Portal kann hierfür auf die API-Endpunkte GraphRefreshToken und DeleteGraphRefreshToken zugreifen. Für die Anforderung des Refresh-Token wird der Autorisierungscode an die API übergeben, die dann den Refresh-Token abruft. Dadurch wird der Refresh-Token nur intern verarbeitet und steht nie außerhalb des Synclers zur Verfügung. Der Scope für die Anforderung des Autorisierungscodes muss identisch mit dem Scope-Wert in der Verbindung sein, damit ein gültiger Access-Token angefordert werden kann. Die vorhandenen Refresh-Token können in verschlüsselter Form im Bereich „Parameter" eingesehen und gelöscht werden. Außerdem kann das externe Portal das Löschen der Freigabe auslösen. Bei diesem Verfahren werden nur delegierte API Berechtigungen unterstützt.

Ein Refresh-Token muss bei Änderung des Benutzerpassworts aktualisiert werden.

Konfigurieren Sie als nächstes die API-Berechtigungen. Für die Synchronisation von Termines des Unternehmen brauchen Sie die Anwendungsberechtigungen „Calendars.ReadWrite". Für eine benutzerbezogene Authentifikation werden delegierte Berechtigungen benötigt. Für die Verwendung des Benutzerfilter wird die Berechtigung „User.Read.All" benötigt. Sie können die Benutzer mit deren Emailadresse aber auch im Feld Benutzerliste auflisten. Die Filterfunktion verhält sich dynamisch, wohingegen die Benutzerliste statisch ist und ggf. für neue Benutzer angepasst werden muss.

Wenn die Benutzerauswahl durch eine Gruppe gesteuert werden soll, wird die Berechtigung GroupMember.Read.All benötigt.

Für die Synchronisation von Kontakten wird die Anwendungsberechtigung „Contacts.ReadWrite" benötigt.

Die Anwendungsberechtigungen müssen durch den Administrator freigegeben werden. Klicken Sie auf „Berechtigung hinzufügen", „Microsoft Graph" und „Anwendungsberechtigung". Mittels des Suchfeldes können Sie die Berechtigungen leicht auswählen.

Die Graph API unterstützt für den Zugriff auf Outlook-Aufgaben keine Anwendungsberechtigung. Damit mit der Verbindung auch Aufgaben synchronisiert werden können, müssen Sie die delegierte Berechtigung „Tasks.ReadWrite" hinzufügen. Im Anschluss müssen die Benutzer den Zugriff bestätigen, wofür je nach Ziel-System eine eigene Oberfläche oder API-Endpunkte (GraphRefreshToken, DeleteGraphRefreshToken) in der Syncler API zur Verfügung stehen.

1.4.1 Funktionen

Schema ermitteln

Das Datenschema wird aus den Metadaten der Graph API (https://graph.microsoft.com/v1.0/\protect\T1\textdollarmetadata) ermittelt. Dabei werden nicht alle Objekte zur Verfügung gestellt, sondern nur eine aktuell geprüfte Teilmenge. ID-Felder und Felder mit Änderungsinformationen werden gekennzechnet, damit diese in Prozessen genutzt werden können. Teilweise wird das Schema auch erweitert, damit z.B. die Information „isDeleted“ im Prozess ausgewertet oder bei Aufgaben und Kontakten über „user“ der zugeordnete Benutzer festgelegt werden kann.

Lesen von Schema-basierten Daten

Mit dem Universalplugin können Daten gelesen werden. Dabei ist auch eine Abfrage per Änderungsdatum möglich. Daneben stehen spezielle Prozesse zur Verfügung, die eine umfassende Synchronisation ermöglichen, und neben Vorkommen einer Serie auch das Löschen von Ereignissen, Aufgaben oder Kontakten verarbeiten.

Die Abfrage von Ereignissen, Aufgaben und Kontakten wird über alle Benutzer einzeln ausgeführt. Die betreffenden Benutzer werden über eine Verbindungseinstellung festgelegt. Das genaue Verfahren wird in den Besonderheiten beschrieben. Für das Lesen von Aufgaben wird nur die Standardliste des jeweiligen Benutzers verwendet.

Lesen von Abfrage-basierten Daten

Diese Funktion wird nicht unterstützt.

Schreiben von Daten

Das Schreiben eines Ereignisses erfolgt über den Benutzer, der für das Lesen genutzt wurde. Bei der Neuanlage wird dieser Benutzer aus dem Feld „Organisator“ oder dem Feld „user“ entnommen. Ungültige Werte führen dabei zu einer Fehlermeldung. Die Anlage einer Ereignis-Serie ruft im Anschluss die Vorkommen mit dem Startzeitpunkt und ggf. dem Ende der Serie ab und speichert Informationen dazu im Änderungsspeicher. Sollte die Serie kein Endzeitpunkt haben, wird das aktuelle Zeitfenster verwendet. Dies ermöglich spezialisierten Prozessen die Vorkommen mit den Quelldaten abzugleichen.

Schreiben von Bulk-Daten

Diese Funktion wird nicht unterstützt.

1.4.2 Einstellungen

Folgende Einstellungen können bereitgestellt werden.

Für die Synchronisation von Kalendereinträgen wird die Delta-Funktion verwendet. Diese setzt ein definiertes Zeitfenster voraus. Ausgenommen von diesem Zeitfenster sind die Vorkommen einer Serie. Diese werden immer vollständig synchronisiert.

Anzahl vergangener Monate

Diese Ganzzahl definiert den Anfang des Zeitfensters für Synchronisationsprozesse von Ereignissen. Die Anzahl an Monaten wird bei kompletten Abfragen dem aktuellem Datum abgezogen. Je größer der Wert ist, umso größer kann das Datenvolumen sein.

Anzahl zukünftiger Monate

Diese Ganzzahl definiert das Ende des Zeitfensters für Synchronisationsprozesse von Ereignissen. Die Anzahl an Monaten wird bei kompletten Abfragen dem aktuellem Datum hinzugefügt. Je größer der Wert ist, umso größer kann das Datenvolumen sein.

Bevorzugte Outlook Zeitzone

Datumswerte werden durch die Graph API mit einem Wert und einer separat angegebenen Zeitzone geliefert und erwartet. Ohne die Angabe einer bevorzugten Zeitzone werden Datumswerte im Format „yyyy-MM-ddTHH:mm:ss“ als UTC zurückgeliefert und müssen z.B. per Transformation in die notwendige Zielzeitzone umgerechnet werden. Wenn Sie Daten speichern, sollten Sie die Zeitzone des Benutzers verwenden, wenn die Zeitzonen-Funktion für Ereignisse nicht aktiviert werden soll.

Authentifizierungsmethode

Diese Auswahl definiert das Anmeldeverfahren, welches oben erläutert wurde. Eine Ausnahme stellen Aufgaben dar. Diese werden immer über eine Benutzerzustimmung synchronisiert.

Verzeichnis-ID (Mandant)

Diese ID definiert das Unternehmen und kann bei der Registrierung der Anwendung ausgelesen werden,

Anwendungs-ID (Client)

Die Anwendungs-ID der registrierten Anwendung.

Client-Geheimschlüssel

Ein geheimer Schlüssel der registrierten Anwendung.

Redirect URI

Für das Authentifizierungsverfahren mittels Administrator Refresh-Token muss eine Weiterleitungs-URI konfiguriert werden. Bei der Freigabe über den Syncler Administrator kann eine beliebige URI angegeben werden. Für die Freigabe über das Web-Frontend ist folgender Wert notwendig. Die Weiterleitungs-URI muss bei der registrierten Anwendung festgelegt werden.

Ist Public Client

Diese Einstellung ist für die Benutzerzustimmung relevant, da der Autorisierungsprozess sich unterscheidet. Sollten Sie die Anwendung als public client konfiguriert haben, darf kein Geheimschlüssel für die Benutzerzustimmung verwendet werden.

Scopes

Diese Berechtigungen werden bei einer Benutzerzustimmung angefordert. Sollten diese geändert werden, muss die Benutzerzustimmung wiederholt werden. Für das Client-Secret-Verfahren ist der Wert https://graph.microsoft.com/.default anzugeben.

Damit Kalender, Aufgaben und Kontakte synchronisiert werden können, müssen die betroffenen Benutzer definiert werden. Hierfür stehen verschiedene Varianten zur Verfügung, die ggf. zusätzliche API-Berechtigungen erfordern.

Methode für Auswahl von Benutzern

Diese Auswahl legt das Verfahren fest, worüber der Benutzerkreis definiert wird. Es stehen folgende Optionen zur Verfügung.

- Benutzerfilter nutzt einen definierten Filter für die Abfrage von Benutzern. Das Verfahren ist dynamisch und benötigt die Berechtigung User.ReadBasic.All
- Benutzerliste ist ein statisches Verfahren, wobei die Emailadressen mit Semikolon getrennt aufgelistet werden.
- Gruppenmitglieder ruft die Mitglieder einer Gruppe ab. Das Verfahren ist dynamisch und benötigt die Berechtigung User.ReadBasic.All und GroupMember.Read.All
- Benutzerzustimmung arbeitet auf der Basis der erteilten Zustimmungen, die über die Syncler API gesammelt wurden. Dieses Verfahren ist ebenfalls dynamisch.

Filter für Benutzer

Dieser Wert in OData-Notation definiert die Menge an Benutzer, für die Ereigniss, Aufgaben und Kontakte abgefragt und ggf. synchronisiert werden. Das genaue Vorgehen ist in den Bensonderheiten beschrieben. Folgender Wert wählt

alle Mitglieder des Unternehmens aus: userType eq ‚Member' Falls Benutzer aus dieser Menge keine Mailbox-Rechte haben, kann es zu Fehlermeldungen kommen.

Benutzerliste

Wenn eine Abfrage von Benutzer nicht möglich oder gewünscht ist, können hier auch gezielt Benutzer mit Semikolon getrennt aufgelistet werden. Dabei wird die Emailadresse der Benutzer erwartet.

Gruppenname für Benutzerauswahl

Durch die Angabe eines Gruppennamens können die Benutzer aus den Mitgliedern der Gruppe entnommen werden. Für diese Funktion wird eine Berechtigung für GroupMember und mindestens User.ReadBasic.All benötigt. Mit jeder Prozessausführung werden die Benutzer aus der Gruppe gelesen und deren Kalender synchronisiert.

Vorhandene Benutzerzustimmungen (schreibgeschützt)

Diese Liste wird bei der Anzeige der Verbindungseinstellungen generiert und zeigt die vorhandenen Benutzerzustimmungen. Das Feld ist schreibgeschützt und kann nicht für die Pflege der Zustimmungen verwendet werden.

1.4.3 Besonderheiten

Für den delegierten Zugriff auf Aufgaben müssen noch weitere Einstellungen vorgenommen werden, damit die Benutzer dem Zugriff zustimmen können, für den Fall, dass ein Geheimschlüssel verwendet wird. Dazu zählt die Definition einer Umleitungs-URI, welche den erzeugten Autorisierungscode verarbeiten kann.

Für Sage CRM ist diese URI möglich: http(s)://…/crm/CustomPages/MicrosoftConsent.asp Damit die Benutzer die Autorisierung vornehmen können, müssen die öffentlichen Clientflows aktiviert werden.

Beim Lesen von Ereignissen gibt es Folgendes zu beachten. Ereignisse, Aufgaben und Kontakte werden über eine Liste von Benutzern ermittelt. Da die ID eines Ereignisses abhängig vom Benutzer ist, über den die Abfrage ausgeführt wurde, wird die Antwort wie folgt verarbeitet. Wenn der aktuelle Benutzer gleich dem Organisator des Ereignisses ist, wird der Datensatz übernommen und die ID z.B. in Datenabbildungen verwendet. Wenn der aktuelle Benutzer nicht gleich dem Organisator ist und der Organisator aber durch die Benutzerauswahl erfasst wird, wird der Datensatz verworfen, damit keine Dubletten angelegt werden. Wenn der Organisator nicht durch die Benutzerauswahl erfasst wird, wird der Termin übernommen und alle Teilnehmer entfernt, die in der Benutzerauswahl enthalten sind und nicht dem aktuellen Benutzer entsprechen. Dieses Verfahren stellt sicher, dass auch geteilte Ereignisse identifiziert werden können und keine Datensätze doppelt verarbeitet werden.

Die Abfragen von Ereignissen mit Universalplugins ermöglicht nicht das Abfragen von Vorkommen einer Serie oder gelöschten Datensätzen.

Für die Synchronisation von Ereignissen können Prozesse die Delta-Funktion der CalendarView verwenden. Diese Funktion benötigt für Ereignisse ein definiertes Zeitfenster (siehe Einstellungen) und einen gespeicherten Änderungstoken. Prozesse, die dies nutzen, speichern das letzte Abfragedatum der kompletten Abfrage. Ab da wird über den Änderungstoken abgefragt, der intern in den Parametern gespeichert und bei erfolgreicher Verarbeitung auch aktualisiert wird. Da ein definierte Zeitfenster keine kontinuierliche Synchronisation ermöglichen würde, wird mit jedem neuen Tag eine komplette Abfrage ausgelöst und das Zeitfenster um einen Tag vorgeschoben. Termine außerhalb des Zeitfensters werden dann nicht mehr von der Synchronisation erfasst.

Für Aufgaben und Kontakte wird ebenfalls eine Delta-Funktion durch spezialisierte Prozesse unterstützt, jedoch ist hier keine Angabe von Zeitfenstern erforderlich.

Wenn die Delta-Funktion einen Termin/Ereignis als gelöscht ausgibt, wird dieser gezielt abgefragt. Sollte er noch existieren, wurde er aus dem aktuellen Zeitfenster verschoben und wird mit verarbeitet. Ohne Resultat wird der Termin als „gelöscht" verarbeitet. Dabei wird das Feld „isDeleted" mit „true" zurückgeliefert.

Eine Datenabfrage aus der Transformation „Daten abfragen" kann in Feldnotation einen einzelnen Datensatz abfragen. Dies kann auch mit einer Benutzereinschränkung ergänzt werden, da sonst alle Benutzer geprüft werden müssen.

Beispiel:

```
id|:|ABC|;|
oder
id|:|ABC|;|user|:|benutzer@domain.de|;|
```

1.4.4 Synchronisationsprozesse

Für eine vollständige Synchronisation sind spezialisierte Prozesse erforderlich. Hier finden Sie eine detaillierte Beschreibung.

Synchronisation zwischen Microsoft 365 und Zoho CRM

Synchronisation zwischen Microsoft 365 und Sage CRM

1.5 Maileon

Maileon ist eine E-Mail-Marketing Lösung für professionelle Newsletter und Marketing Automation. Diese Verbindung realisiert die Anmeldung und Kommunikation mit der API und stellt ein Datenschema mit zusätzlichen Funktionen bereit. Für die Integration mit anderen Verbindungen können die Universal Prozesse verwendet werden.

1.5.1 Funktionen

Schema ermitteln

Das Datenschema wird fest von der Verbindung erstellt. Benutzerdefinierte Kontaktfelder werden dynamisch beim Erstellen des Schemas eingelesen. Die Funktion zum Anlegen von Kontaktfeldern erweitert das interne Datenschema automatisch. Das Datenschema ist eine Kombination aus lesbaren Feldern, angereicherten Feldern, übergebbaren Parametern, z.B. beim Speichern von Daten und zusätzlichen Parametern, die Funktionen direkt ansteuern können. Unter Datenobjekte steht eine Detailbeschreibung zur Verfügung.

Lesen von Schema-basierten Daten

Abhängig vom Datentyp können Listen, einzelne Datensätze oder Änderungen abgerufen werden. Unter Datenobjekte steht eine Detailbeschreibung zur Verfügung.

Lesen von Abfrage-basierten Daten

Diese Funktion wird nicht unterstützt.

Schreiben von Daten

Mittels dem Schreiben von Daten werden Datensätze angelegt, geändert oder auch Funktionen aufgerufen. Unter Datenobjekte steht eine Detailbeschreibung zur Verfügung.

Schreiben von Bulk-Daten

Diese Funktion wird nicht unterstützt.

1.5.2 Einstellungen

Folgende Einstellungen müssen bereitgestellt werden.

API Key

Der API Key kann über die Benutzeroberfläche von Maileon ermittelt werden. Melden Sie sich dazu bei Maileon an und wechseln Sie in den Bereich Einstellungen. Unter Konto / API-Keys können vorhandene API Keys ausgelesen oder neue angelegt werden.

1.5.3 Datenobjekte

Folgende Datenobjekte stehen zur Verfügung.

Das einzelne Abrufen von Datensätzen kann, falls unterstützt, mit den Standard-Prozessfeldern und folgender Notation erfolgen.

```
id|:|1234|;|
```

Kontakt / contact

Mit diesem Objekt können Kontakte in Maileon ausgelesen, angelegt, aktualisiert oder gelöscht werden. Das Löschen eines Kontaktes erfordert die Einrichtung eines speziellen Löschprozesses. Der Bereich custom_fields enthält die benutzerdefinierten Felder, welche auch mit der Verbindung und dem Objekt customfield angelegt werden können.

Die folgenden Felder werden als Parameter für das Anlegen oder Aktualisieren verwendet.

- **src** Insert/Update: Eine Zeichenfolge, die die Quelle des Kontakts beschreiben soll.
- **doimailing** Insert/Update: Dieser Parameter wird ignoriert, wenn triggerdoi nicht angegeben oder falsch ist. Referenziert das zu verwendende doi-Mailing.
- **subscription_page** Insert: Falls diese Methode von einer Abonnementseite aufgerufen wurde, bietet diese Zeichenfolge die Möglichkeit, sie für die Verwendung in Berichten zu verfolgen.
- **doi** Insert: Gibt an, ob für den erstellten Kontakt ein Double-Opt-In-Prozess gestartet werden soll. Beachten Sie, dass der für diese Anfrage zurückgegebene Statuscode nicht bedeutet, dass der doi-Prozess erfolgreich war. Unterstützte Werte sind true und false.
- **doiplus** Insert: Dieser Parameter wird ignoriert, wenn doi nicht angegeben oder falsch ist. Falls der DOI-Prozess erfolgreich ist, darf Maileon Öffnungen und Klicks des Kontakts nachverfolgen.
- **triggerdoi** Update: Wenn bereitgestellt und wahr (unterstützte Werte sind wahr und falsch) und wenn die Berechtigung entweder 1, 2, 3 oder 6 ist, wird die Berechtigung direkt gesetzt und es wird keine DOI-Mail gesendet, da sie für diese Berechtigungen nicht erforderlich ist. Wenn die Berechtigung auf 4 (doi) oder 5 (doi+) gesetzt ist, wird ein doi-Prozess für den Kontakt ausgelöst, anstatt die Berechtigung sofort auf 4 oder 5 zu setzen (nochmals: die Berechtigung wird nicht geändert).

Kontakte können als einzelne Datensätze oder Liste abgerufen werden. Das Abrufen von geänderten Kontakten per Prozess wird ebenfalls unterstützt. Checksummen werden beim Lesen nicht verwendet.

Das Filtern oder Suchen von Kontakten kann in den Standard-Prozessfeldern mit folgender Notation erfolgen. Damit können gezielt Inhalte von Kontaktfiltern abgerufen oder einzelne Kontakte per E-Mail oder externer ID gesucht werden.

```
Suche nach Kontakten:
    emails|:|gesuchte@email.de|;|
    externalid|:|gesuchte_externe_id|;|
```

```
Abrufen eines Kontaktfilters:
    contactfilterid|:|kontaktfilter_id|;|
```

Benutzerdefiniertes Feld / customfield

Dieses Objekt kann nur für die Anlage oder das Umbenennen eines benutzerdefinierten Feldes genutzt werden. Eine Abfrage über die Lesen-Funktion der Verbindung ist nicht vorgesehen und wird immer ein leeres Ergebnis liefern.

Folgende Felder stehen zur Verfügung.

- **name** Der Name des Feldes.
- **oldname** Für den Fall, dass ein vorhandenes Feld umbenannt werden soll, wird hier der aktuelle Name erwartet.
- **type** Definiert den Datentyp des Feldes. (string / integer / float / date / boolean)

Beim Schreiben dieses Objektes über die Verbindung wird automatisch auch das Schema des Kontakt-Objektes aktualisiert. Eine manuelle Aktualisierung ist nicht erforderlich.

Kontaktfilter / contactfilter

Dieses Objekt kann ausgelesen und aktualisiert werden. Die direkte Anlage wird nicht unterstützt. Allerdings kann ein Kontaktfilter über die Verteilerliste automatisiert angelegt werden. Details dazu finden Sie beim Objekt Verteilerliste. Das Löschen eines Kontaktfilters erfordert die Einrichtung eines speziellen Löschprozesses.

Bei der Aktualisierung kann ausschließlich der Name des Kontaktfilters verändert und ein Refresh der Kontakte ausgelöst werden.

-**refresh_contacts** Wird diesem Feld der Wert true beim Schreiben übergeben, wird die Aktualisierung der Kontakte des Kontaktfilters ausgelöst. Die Verwendung ist nur aller 10 Minuten möglich.

Kontaktfilter können einzeln abgerufen werden. Wird das Filtern oder Suchen mit den Standard-Prozessfeldern ausgelöst, wird der Wert direkt mit dem Names des Kontaktfilters verglichen und alle Übereinstimmungen werden zurückgegeben.

Verteilerlisten / targetgroup

Dieses Objekt kann ausgelesen und aktualisiert werden. Das Löschen einer Verteilerliste erfordert die Einrichtung eines speziellen Löschprozesses. Für die Anlage einer Verteilerliste stehen zwei Verfahren zur Verfügung.

- **contact_filter_name** Kann bei Anlage zur Suche eines Kontaktfilters verwendet werden.
- **contact_filter_id** Kann bei Anlage zur Auswahl eines Kontaktfilters verwendet werden.
- **create_contact_filter** Erzeugt ggf. benutzerdefiniertes Feld und gleichnamigen Kontaktfilter, der der neu angelegten Verteilerliste zugeordnet wird.
- **contact_filter_fieldname** Definiert den Feldnamen des Kontaktfilters. Das Feld wird ggf. als Wahrheitsfeld für Kontakte angelegt. Sollte das Feld bereits vorhanden sein und einen anderen Typ haben, resultiert daraus eine Fehlermeldung.

Verteilerlisten können einzeln abgerufen werden. Wird das Filtern oder Suchen mit den Standard-Prozessfeldern ausgelöst, wird der Wert direkt mit dem Names der Verteilerlisten verglichen und alle Übereinstimmungen werden zurückgegeben.

Beim Erzeugen einer Verteilerliste mit dem Setzen der Felder contact_filter_name oder contact_filter_id muss der Kontaktfilter bereits vorhanden sein, da es sonst zu einer Fehlermeldung kommt. Werden für die Anlage die Felder create_contact_filter und contact_filter_fieldname verwendet, wird der gleichnamige Kontaktfilter automatisch angelegt.

Der Name einer Verteilerliste muss eindeutig sein. Identische Namen führen bei Anlage oder Aktualisierung zu einer Fehlermeldung.

Berichte / reports

Dies ist ein Oberbegriff für folgende Typen, die identisch verwendet werden können.

- open
- unique_open
- bounce
- unique_bounce
- click
- unique_click
- block
- unsubscription
- subscriber
- recipient

Abhängig vom Typ des Berichts steht der betreffende Kontakt als untergeordnetes Objekt im Schema zur Verfügung. Außerdem werden die Einträge zu den Berichten mit dem Namen des Mailings **mailing_name** angereichert.

Bei Berichten können keine einzelnen Datensätze, sondern nur Listen abgerufen werden. Das Schreiben und Löschen wird nicht unterstützt. Sollte es bei der Verarbeitung im Prozess zu Fehlern oder Wiederholungen kommen, werden diese über den Änderungsspeicher realisiert.

Für die Änderungsabfrage wird automatisch der Parameter **from_date** verwendet. Alle Eingaben in Filter- und Suchfeldern eines Prozesses werden der verwendeten Url angehängt. Damit können weitere Filter, wie z.B. mailing_id, to_date oder standard_field, realisiert werden.

1.6 Salesforce

1.6.1 Funktionen

Schema ermitteln
Laden von Quelldaten
Laden von Schema-basierten Daten
Laden von Abfrage-basierten Daten
Schreiben von Daten
Schreiben von Bulk-Daten

1.6.2 Einstellungen

Webservice Benutzername
: Der Benutzername der für die Synchronisation verwendet werden soll. Es ist darauf zu achten, dass der Benutzer über die entsprechenden Berechtigungen verfügt, die für die Synchronisation erforderlich sind.

Webservice Passwort
: Passwort des o.g. Benutzers

Sicherheitstoken
: Sicherheitstoken des o.g. Benutzers. Das Sicherheitstoken kann über die Benutzereinstellungen in Salesforce mit der Funktion „Mein Sicherheitstoken zurücksetzen“ generiert bzw. geändert werden. Das Sicherheitstoken ändert sich automatisch bei jeder Passwortänderung. Es ist empfehlenswert,

den Benutzer, der für die Synchronisation verwendet wird, so einzustellen, dass das Passwort nicht abläuft (z.B. über einen Berechtigungssatz)

Login Server Url
: Als Standard wird hier die URL für Salesforce Produktivumgebungen verwendet, z.B. https://login.salesforce.com/services/Soap/u/46.0. Soll eine Sandbox angesprochen werden, ist die URL auf https://test.salesforce.com/services/Soap/u/46.0 zu ändern. Ebenso können hier custom domains eingetragen werden.

Dublettenwarnungen ignorieren
: Erkennt Salesforce eine Dublette bei der Anlage, dann wird die Anlage mit einer Warnung abgebrochen und der Datensatz nicht synchronisiert. In den meisten Synchronisierungsszenarien ist dies jedoch nicht wünschenswert und die Warnungen können mit der entsprechenden Einstellung ignoriert werden. Die Dublettenregel in Salesforce muss auf „warnen“ und nicht auf „blockieren“ gestellt sein.

1.7 Tabellenkalkulation

Diese Verbindung ermöglich die Erstellung und Befüllung von Tabellen auf der Basis einer Vorlage. Das Verfahren ist vergleichbar mit einem Seriendruck.

Das Resultat kann als Tabellenkalkulation oder PDF-Druck weiterverwendet werden.

Neben einzelnen Feldern können auch geschachtelte Listen, HTML und Bilder verarbeitet werden.

Für die Funktion stehen spezifische Prozesse zur Verfügung, die die Konfiguration und den Ablauf unterstützen. Außerdem ist eine direkte Ausführung über die Syncler API möglich.

1.7.1 Funktionen

Schema ermitteln

Die Verbindung erstellt nur ein Schema-Objekt „SpreadsheetFile“ für die Konfiguration und das Resultat. Damit ist auch eine individuelle Konfiguration je Datensatz möglich. Außerdem enthält das Schema generische Nutzfelder, die bei der Ausführung zugewiesen und in der Nachfolger-Bearbeitung ausgewertet werden können. Die eigentlichen Daten werden dynamisch durch die Prozesse bereitgestellt.

Lesen von Schema-basierten Daten

Die Verbindung kann ausschließlich Datensätze aus dem Änderungsspeicher lesen. Eine erfolgreiche Verarbeitung mit Nachfolgerprozess speichert das Resulat im Änderungsspeicher zwischen. Der Nachfolger liest und verabeitet die Datensätze. Bei erfolgreicher Verarbeitung und ohne weiteren Nachfolger werden die Daten aus dem Änderungsspeicher entfernt.

Lesen von Abfrage-basierten Daten

Diese Funktion wird nicht unterstützt.

Schreiben von Daten

Das Schreiben führt die eigentliche Funktion aus. Die Vorlage wird dabei gelesen und mit dem bereitgestellten Daten gefüllt. Das Resultat wird je nach Konfiguration gespeichert oder ausschließlich zurückgegeben.

Schreiben von Bulk-Daten

Diese Funktion wird nicht unterstützt.

1.7.2 Einstellungen

Folgende Einstellungen können bereitgestellt werden.

FTP Quellverzeichnis

Dies definiert das Quellverzeichnis für die Vorlagen, welche per FTP gelesen werden sollen.

FTP Zielverzeichnis

Dies definiert das Zielverzeichnis für die Resultat, welche per FTP gespeichert werden sollen.

FTP Verzeichnis für Schriftarten

Für den PDF-Druck kann hier ein optionales Verzeichnis für Schriftarten angegeben werden. Es stehen bereits einige Standardschriftarten zur Verfügung bzw. werden On-premises die installierten Schriftarten verwendet. Falls der PDF-Druck nicht mit der in der Vorlage konfigurierten Schriftart erfolgt, muss diese über dieses Verzeichnis bereitgestellt werden.

Schriftarten löschen

Nach erfolgreicher Ausführung kann hiermit das Löschen der lokalen Schriftartkopie veranlasst werden.

FTP Server

Serveradresse des FTPs.

FTP Port

Serverport des FTPs.

FTP Benutzername

Benutzername für den FTP-Zugriff.

FTP Passwort

Passwort für den FTP-Zugriff.

SFTP verwenden

Soll SFTP verwendet werden.

Erweitertes Protokoll

Optional können zusätzliche Protokolleinträge erzeugt werden, damit im Fehlerfall eine genauere Analyse möglich ist.

1.7.3 Besonderheiten

Als Eingabeformat stehen die Dateitypen XLS, XLSX, CSV, HTML und ODS zur Verfügung. Auf diese Vorlagen kann per lokalem PFad (nur On-premises), Base64-Daten, Vorlagendatenbank, URL oder FTP zugegriffen werden.

Als Ausgabeformat stehen die Dateitypen XLS, XLSX, CSV, HTML, Image, ODS, XPS und PDF zur Verfügung. Das Resultat kann lokal (nur On-premises) und auf einem FTP abgelegt werden. Außerdem kann das Resultat als Base64 direkt zurückgegeben oder im Änderungsspeicher abgelegt werden.

Nicht alle Funktionen stehen mit jedem Eingabe- oder Ausgabeformat zur Verfügung.

Seriendruck-Felder werden mit der #..#-Notation in der Vorlage definiert und in allen Tabellenblättern verarbeitet.

Wenn der Feldname den Präfix „Html:“ hat, wird der Inhalt als HTML interpretiert und die Formatierungen übernommen. Damit ist eine Daten-bezogene Formatierung des Resultats möglich.

Wenn der Feldname den Präfix „Picture:“ hat, wird der Inhalt als Anweisung zum Lesen einer Grafikdatei verwendet. Die Skalierung des Bildes im Dokument wird dabei durch die Dimension der verwendeten Zelle festgelegt. Die

Anweisung kann in Json angegeben werden. Wird kein gültiges Json vorgefunden, wird der Inhalt direkt als Url zur Datei interpretiert. In Json werden die Eigenschaften „FileMethod“ und „File“ erwartet. FileMethod kann dabei die Ausprägungen PATH, BASE64, URL und DATABASE (Vorlagendatenbank) haben. File enthält die Daten zu dieser Methode. Die Formate JPG, BMP, GIF, PNG, TIFF und WMF werden unterstützt.

Die Verarbeitung von geschachtelten Listen erfordert die Feldern #RangeStart:..# und #RangeEnd:..# in der Vorlage. Alle Zeilen zwischen diesen Positionen werden kopiert und für jeden Listeneintrag ausgefüllt. Das Kopieren erfolgt dabei auf der Basis von kompletten Zeilen, wobei auch Formatierungen erhalten bleiben. Die Start- und Endpositonsfelder werden geleert.

Sollte für die geschachtelte Liste eine Transformation definiert sein, bei der neue Felder erzeugt werden, werden diese in das Listenelement kopiert, damit sie zur Verfügung stehen. Dies unterscheidet sich zur regulären Verarbeitung von geschachtelten Listen.

Der PDF-Druck verwendet das in der Excel-Vorlage definiert Seitenlayout für das Resultat. Außerdem erfolgt der Druck über alle Tabellenblätter.

1.8 Universal-SDK-Verbindung

Diese Verbindung stellt eine Entwicklungsplattform für die Anbindung von individuellen Systemen bereit. Für die grundlegenden Funktionen einer Verbindungen können C# Skripte entwickelt werden, die dann zur Ausführungszeit angewendet werden. Dazu gehört das Erstellen eines Datenschemas, das Lesen von Daten, das Schreiben von Daten und das Ausführen einer Abfrage.

Die Verbindung kann mit den Universalprozessen verwendet werden, es stehen aber auch SDK-Prozesse zur Verfügung, um auch an dieser Stelle zusätzlich Code einbringen zu können.

In einer On-premises Installation unterliegt die Ausführung von Skripten keiner Einschränkung. In der Cloud wird die Speichernutzung und die Ausführungsdauer beschränkt. Sollte das Skript die Limits überschreiten, wird es mit einer Fehlermeldung abgebrochen.

Die verfügbaren Bibliotheken sind identisch zur Transformation „C# Code Script“. Siehe *Der SDK-Helper*

Die Verbindung unterstützt einen generischen code-based OAuth-Grant-Flow. Der erzeugte Code steht in Skripten als Parameter „OAuthCode“ zur Verfügung. Nach der Ausführung des Schema-Skriptes wird er wieder in der Verbindung gespeichert. Damit kann der Code für das Abrufen eines Refresh-Token verwendet und nachträglich entfernt werden.

1.8.1 Funktionen

Schema ermitteln

Das Schema wird von dem JSON Schema Skript generiert. Die Vorlage für das Skript enthält eine Methode Execute, die beim Abrufen des Schemas ausgeführt wird, und das Datenschema in JSON zurückgeben muss. Das Skript muss ein JSON-Array mit JSON-Schemaobjekten zurückliefern (https://json-schema.org/). Für die Primärschlüssel fügen Sie bitte ein String-Array „uniqueidentifier“ mit den Feldnamen hinzu (vgl. required). Der Feldname für eine Aktualisierungsinformation wird mit der Eigenschaft „updated“ definiert.

```
[
        {
                "title": "address",
                "properties": {
                        "street": {
                                "title": "Street",
                                "type": "string",
                                "maxLength": 60,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
                "description": "Hint for user"
            },
            "housenumber": {
                "type": "integer",
                "title": "Housenumber"
            },
            "addressid": {
                "type": "integer",
                "title": "Record ID",
                "readOnly": true
            },
            "updateddate": {
                "title": "Updated at",
                "type": "string",
                "format": "date-time",
                "description": "Last updated date"
            },
            "country": {
                "title": "Land",
                "type": "string",
                "enum": [
                    "DE",
                    "AT",
                    "CH"
                ]
            },
            "geo": {
                "title": "Coordinates",
                "type": "object",
                "properties": {
                    "lat": {
                        "title": "Latitude",
                        "type": "number"
                    },
                    "lng": {
                        "title": "Longitude",
                        "type": "number"
                    }
                }
            },
            "floors": {
                "title": "Floors",
                "type": "array",
                "ischildlist": false,
                "items": {
                    "title": "Floors",
                    "type": "object",
                    "properties": {
                        "appartments": {
                            "title": "Appartments",
                            "type": "number"
                        },
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
                                                "renter": {
                                                        "title": "Renter",
                                                        "type": "number"
                                                }
                                        }
                                }
                        },
                        "colors": {
                                "title": "Colors",
                                "type": "array"
                        }
                },
                "required": [
                        "street"
                ],
                "updated": "updateddate",
                "uniqueidentifier": [
                        "addressid"
                ]
        }
]
```

Lesen von Schema-basierten Daten

Das Lesen von Schema-basierten Daten wird von dem JSON-Daten lesen Skript implementiert. Über die Helper-Bibliothek stehen alle Parameter der aktuellen Anfrage zur Verfügung. Das aktuelle Schema ist in Helper.TargetObject zu finden. Wenn die Seitenweise-Verarbeitung aktiviert ist, wird dieses Skript so oft ausgeführt, bis es keine Daten mehr liefert. Die maximal Anzahl von Aufrufen ist auf 50 beschränkt. Die aktuelle Seitennummer steht im Helper zur Verfügung. Die Methode Execute muss ein Array von Objekten gemäß Datenschema in JSON zurückliefern.

Lesen von Abfrage-basierten Daten

Das Lesen von Schema-basierten Daten wird von dem JSON Abfrage Skript implementiert. Das Verfahren ist weitestgehend identisch zum Lesen von Schema-basierten Daten. Das Skript wird sowohl für die Erzeugung des Abfrage-Schema, als auch für die Abfrage selbst verwendet. Eine Unterscheidung ist mittels Helper.GetParam<bool>(„GetQuerySchema“) möglich.

Schreiben von Daten

Das Schreiben von Daten wird von dem JSON-Daten schreiben Skript implementiert. Das aktuell zu schreibende Objekt wird in Helper.SetObject bereitgestellt. Der Rückgabewert der Methode sollte das Objekt mit ggf. generierte ID oder aktuellem Änderungsdatum sein. Diese beiden Informationen werden dann vom Prozess ausgewertet und z.B. für Datenabbildungen verwendet.

1.8.2 Einstellungen

Die Verbindung verfügt nur über wenig Parameter, da alles von den Skripten getragen wird. Damit Zugangsdaten aber nicht sichtbar in den Skripten enthalten sein müssen, gibt es folgende Werte, die dann in den Helper-Parametern zur Verfügung stehen.

Authentifizierung URL

Platzhalter für eine URL. Parameter ‚Url‘

Benutzername

Platzhalter für einen Benutzernamen. Parameter ‚Username'

Passwort

Platzhalter für ein Passwort, welches besonders geschützt wird. Das externe Auslesen oder Kopieren des Wertes wird nicht unterstützt. Parameter ‚Password'

Authorization Url

Platzhalter für eine URL. Parameter ‚OAuthUrl' Diese Url wird für die OAuth-Authorization verwendet.

Account ID

Platzhalter für eine Account ID. Parameter ‚OAuthAccountId'

Client ID

Platzhalter für eine Client ID. Parameter ‚OAuthClientId'

Secret

Platzhalter für ein Secret. Parameter ‚OAuthSecret'

Redirect Url

Platzhalter für eine Redirect Url. Parameter ‚OAuthRedirectUrl' Diese Url wird für die OAuth-Authorization verwendet und automatisch an die Authorization Url in kodierter Form angehängt.

API URL

Damit zur Authentifizierung eine abweichende URL gepflegt werden kann. Parameter ‚ApiUrl'

API Key

Parameter ‚ApiKey'

1.8.3 Beispiele

/api/example_salesviewer

1.9 CleverReach

CleverReach ist ein Email-Marketing-Tool für den Versand und die Auswertung von Email-Kampagnen. Die CleverReach-Verbindung nutzt die REST API von CleverReach für die Kommunikation.

Für die Einrichtung des Zugangs müssen Sie sich im CleverReach-Portal anmelden. Wechseln Sie dort in den Bereich

- Mein Account
- Extras
- REST API

Hier müssen Sie eine neue OAuth-App erstellen. Vergeben Sie einen individuellen Namen und wählen Sie die REST API Version 3 aus. Abhängig vom Einsatz der Verbindung müssen Sie noch die erforderlichen Scopes aktivieren. Empfänger und Report benötigen Sie für die Empfängerübertragung und das Abrufen von Resultaten. Nachdem Sie die App gespeichert haben, rufen Sie erneut den Bearbeitungsdialog mit dem 3-Punkte-Icon auf.

Jetzt stehen Ihnen zusätzliche Registerkarten zur Verfügung, die die benötigten Informationen enthalten. Wechseln Sie zu „Oauth2 App Daten". Übernehmen Sie die Client ID und das Client Secret in die entsprechenden Felder der Verbindung. Um einen Refresh Token zu generieren, müssen Sie die Schaltfläche „Prozess jetzt testen" anklicken. Es öffnet sich ein neues Fenster, wo Sie sich erneut mit Ihren CleverReach-Zugangsdaten anmelden müssen. Nach

der erfolgreichen Anmeldung erhalten Sie eine Tabelle, aus der Sie den Refresh Token in das Feld der Verbindung übernehmen müssen. Mit dem Speichern der Verbindung haben Sie die Einrichtung abgeschlossen.

Der Refresh Token ist 30 Tage gültig und wird automatisch erneuert, wenn die Verbindung verwendet wird. Sollten Sie die Verbindung mehr als 30 Tage nicht benutzen, muss ein neuer Refresh Token mit der Schaltfläche „Prozess jetzt testen“ abgerufen werden.

1.9.1 Funktionen

Schema ermitteln

Das Schema wird von der Verbindung fest erstellt. Das Empfänger-Schema wird mit benutzerdefinierten globalen Attributen in einer eigenen Objektgruppe ergänzt. Sobald eine Gruppen-ID in den Verbindungseinstellungen hinterlegt wird, werden auch Gruppenattribute in einer eigenen Objektgruppe ergänzt. Für die Arbeit mit Tags werden zwei Verfahren unterstützt. Wenn „Tags als Freitext“ abgeschaltet ist, werden die vorhandenen Tags ermittelt und als einzelne Felder in einer Objektgruppe dem Schema hinzugefügt. Sollten neue Tags hinzukommen, muss das Schema der Verbindung aktualisiert werden. Wenn „Tags als Freitext“ eingeschaltet ist, steht im Empfängerschema das Feld „tags“ zur Verfügung, welches alle Tags in getrennter Form enthält. Nur bei diesem Verfahren stehen auch automatisch neue Tags zur Verfügung. Für die Verarbeitung muss dann allerdings z.B. auf Transformationen zurückgegriffen werden. Das Empfängerschema enthält neben der ID auch noch die Gruppen-ID als Identifikator, da nur mit beiden Werten auch Gruppenattribute zur Verfügung stehen.

Lesen von Schema-basierten Daten

Das Lesen von Daten erfolgt über die REST API. Abhängig vom Schemaobjekt stehen verschiedene Parameter nicht zur Verfügung.

Gruppen können einzeln oder als Liste und auch mittels Änderungserkennung gelesen werden. Ein Filter kann für Gruppen nicht eingesetzt werden.

Mailings können einzeln oder als Liste und auch mittels Änderungserkennung gelesen werden. Mittels Feldnotation können noch folgende Kriterien zur Einschränkung angegeben werden: limit, channel_id, state, start und end.

Empfänger können einzeln oder als Liste abgerufen werden. Die Abfrage benötigt eine Gruppe, die über die Verbindung oder spezialisierte Prozesse definiert werden kann. Sollte keine Gruppe angegeben werden, werden zuerst alle Gruppen und dann alle Empfänger dieser Gruppen abgerufen. Empfänger haben keine Änderungsinformation, wodurch eine Änderungsübertragung ausgehend von CleverReach so nicht möglich ist. Die Aktualisierung eines Empfängers ändert jedoch das Änderungsdatum der Gruppe. Durch die Verbindung wird das Registrierungsdatum des Empfängers als Änderungsdatum herangezogen. Dadurch können keine geänderten Empfänger abgerufen, aber kontinuierlich neue Empfänger übertragen werden.

Resultate oder Links können nur mit den spezialisierten Prozessen oder mit SDK-Prozessen abgerufen werden, da hier immer ein übergordneter Datensatz angegeben werden muss. Dies wird mit den Parameter „GETDATA_RELATED“ und „GETDATA_RELATED_ID“ gesteuert. „GETDATA_RELATED“ kann dabei die Werte groups, mailings oder reports annehmen.

Lesen von Abfrage-basierten Daten

Diese Funktion wird nicht unterstützt.

Schreiben von Daten

Das Schreiben von Daten erfolgt über die REST API. Empfänger können einzeln gespeichert werden. Einige spezialisierte Prozesse unterstützen ein UPSERT-Verfahren zum Abgleich einer Liste von Empfängern.

Schreiben von Bulk-Daten

Diese Funktion wird nicht unterstützt.

1.9.2 Einstellungen

Client ID

Die ID der OAuth App.

Client Secret

Der Geheimschlüssel der OAuth App.

Refresh Token

Der aktuelle Refresh Token. Bei fortlaufender Benutzung wird dieser Wert automatisch aktualisiert.

Gruppen-ID

Damit gruppenspezifische Attribute angesprochen werden können, muss hier die Gruppen-ID hinterlegt werden. Außerdem kann diese Gruppen-ID als Standardgruppe verwendet werden.

Tags als Freitext

Tags werden im Empfänger-Schema als kommata- und leerzeichengetrennter Freitext in einem Feld gepflegt statt als einzelne Spalten in einem Unterobjekt.

1.10 ActiveCampaign

ActiveCampaign ist eine Cloud-Softwareplattform für die Automatisierung von E-Marketing-Kampagnen. Diese Verbindung dient in erster Linie dem Austausch von Stammdaten und unterstützt dabei die benutzerdefinierten Felder.

Für die Einrichtung müssen Sie die API URL und den API Key über das Web-Portal ermitteln. Beides finden Sie unter „Settings“ / „Developer“. Achten Sie darauf, dass der API Key mit ausreichenden Rechten abgerufen wird. Ansonsten besteht die Möglichkeit, dass Daten nicht gelesen oder geschrieben werden können.

1.10.1 Funktionen

Schema ermitteln

Das Schema wird fest vorgegeben und um die benutzerdefinierten Felder für Account und Contact erweitert.

Lesen von Schema-basierten Daten

Das Lesen von Daten erfolgt über die REST API. Eine Abfrage von Änderungen mittels eines Grenzwertes ist möglich. Bei der Abfrage von Accounts kann dieser Grenzwert aber nicht der Abfrage übergeben werden, sondern wird nachträglich als Filter eingesetzt. Bei sehr vielen Accounts kann so ein hoher Datenverkehr entstehen. In diesem Fall sollte das Übertragungsintervall größer gewählt werden. Das abhängige Lesen von Contacts zu einem Account wird unterstützt und kann für Nachfolgeprozesse verwendet werden.

Der Abfragefilter kann verschieden eingesetzt werden. Eine Angabe in Feldnotation beginnend mit „id|:|“ führt zu einer Einzelabfrage des Datensatzes. Eine Angabe in Feldnotation mit den Werten „Related“ und „RelatedId“ kann zum Abfragen von Contacts zu einem Account genutzt werden. Ansonsten wird der Abfragefilter als URL-Ergänzung verwendet und kann z.B. Suchen realisieren.

Für Contact stehen folgende Optionen zur Verfügung. Siehe https://developers.activecampaign.com/reference/list-all-contacts

- email=info@mydomain.com
- email_like=mydomain.com
- search=MyLastName

Für Accounts steht der Parameter „search=MyCompany“ für eine Suche zur Verfügung.

Lesen von Abfrage-basierten Daten

Diese Funktion wird nicht unterstützt.

Schreiben von Daten

Das Schreiben von Daten erfolgt über die REST API. Es können Accounts und Contacts angelegt, aktualisiert oder gelöscht werden. Die doppelte Verwendung einer Emailadresse löst beim Speichern eine Fehlermeldung aus.

Schreiben von Bulk-Daten

Diese Funktion wird nicht unterstützt.

1.10.2 Einstellungen

API URL

Die URL der API. Die Version 3 wird automatisch ergänzt. Der Wert kann dem Web-Portal entnommen werden.

API Key

Der API Key. Der Wert kann dem Web-Portal entnommen werden. Die Berechtigungen sind abhängig vom Benutzer, der den API Key erstellt hat.

1.10.3 Besonderheiten

Für die Kombination mit CAS stehen Vorlagen für eine bidirektionale Synchronisation der Stammdaten zur Verfügung. Für die Realisierung der Account-Contact-Beziehung werden Transformationen für das Abfragen von vorhandenen Datenabbildungen eingesetzt. Diese müssen noch manuell auf den jeweiligen Prozess eingestellt werden, damit diese Funktion zur Verfügung steht.

1.11 Support-Datenbank

Diese Verbindung ermöglich den Zugriff auf die Support-Datenbank. Hierbei handelt es sich um ein zusätzliches Feature, das aktiviert werden muss.

Siehe *Support-Datenbank*

1.11.1 Funktionen

Schema ermitteln

Das Schema wird automatisch über die vorhandenen Tabellen in der Support-Datenbank erstellt. Zusätzlich kann ein geschachtelteter Datentyp definiert werden.

Lesen von Schema-basierten Daten

Das Lesen von Schema-basierten Daten in Universalprozessen wird unterstützt. Für das gezielte Lesen einzelner Datensätze muss die Tabelle über eine ID-Spalte verfügen.

Lesen von Abfrage-basierten Daten

Das Lesen von Abfrage-basierten Daten wird per SQL-Verbindung realisiert.

Schreiben von Daten

Das Schreiben von einzelnen Datensätzen erfolgt basierend auf Schema-Objekten. Daneben können auch SQL-Anweisungen mit speziellen Prozessen ausgeführt werden.

Schreiben von Bulk-Daten

Das Schreiben von Bulk-Importen wird unterstützt.

1.11.2 Einstellungen

Für die Verwendung dieser Verbindung sind keine Angaben erforderlich. Für geschachteltete Datentypen können folgende Einstellungen verwendet werden.

Select-Anweisung für Hauptobjekt

Diese Abfrage ist die Basis für das Schemaobjekt „Mainobject“.

Select-Anweisung für geschachteltete Objekte

Diese Abfrage definiert die Listenobjekte im „Mainobject“. Mittels Platzhalter in Rautenschreibweise werden Daten aus dem Hauptobjekt für die Abfrage eingefügt.

Optionale ID-Felder des Hauptobjektes

Sollte es nicht möglich sein, die ID-Felder des Hauptobjektes zu ermitteln, können diese ggf. kommagetrennt angegeben werden. Dies ist für die Verwendung in Prozessen erforderlich.

1.12 Microsoft Dynamics 365 Customer Engagement

Microsoft Dynamics 365 Customer Engagement ist Enterprise CRM-System mit verschiedenen Modulen.

Diese Verbindung unterstützt die bidirektionale Synchronisation ausgewählter Entitäten mittels Web-API. Außerdem können beliebig lesende Abfragen über die Web-API ausgeführt werden. Die Verbindung realisiert die Anmeldung an der Web-API mit OAuth 2.0.

Für die Einrichtung der Anmeldung ist eine registrierte Anwendung bei Microsoft erforderlich.

Registrieren Sie die Anwendung über das Registrierungsportal https://go.microsoft.com/fwlink/?linkid=2083908. Melden Sie sich mit Ihrem Microsoft Konto an und rufen Sie die Funktion „Neue Registrierung“ auf. Vergeben Sie einen Anwendungsnamen und klicken Sie auf Registrieren.

Danach wird Ihnen die Übersicht zur Registrierung angezeigt. Für die Verbindung benötigen Sie die Anwendungs-ID (Client) und die Verzeichnis-ID (Mandant). Übernehmen Sie die Angaben in die gleichnamigen Felder.

Konfigurieren Sie als nächstes die API-Berechtigungen. Fügen Sie die Berechtigung „Dynamics CRM - user_impersonation“ zur App hinzu.

Sollten Sie die Client-Anmeldeinformationen nutzen wollen, müssen Sie einen Geheimschlüssel anlegen und übernehmen. In diesem Fall muss „Client-Anmeldeinformationen verwenden“ auf „Ja“ gestellt werden. Eine Umleitungs-URI und die OAuth 2.0 Anmeldung entfallen hier.

Ohne Client-Anmeldeinformationen legen Sie nun noch eine Umleitungs-URI fest. Für On-premises Installation oder die Konfiguration mit dem Syncler Administrator können Sie eine beliebige URL angeben.

Zum Beispiel: https://localhost/myapp

Tragen Sie die URL in das Feld „Redirect URI“ ein.

Als nächstes müssen Sie den Web-API-Endpunkt ermitteln. Melden Sie sich dazu in Dynamics 365 an und rufen Sie die Einstellungen auf. Wechseln Sie zu Anpassungen und rufen Sie die Entwicklerressourcen auf. Unter Instanz-Web-API

finden Sie die spezifische URL für Ihre Instanz. Kopieren Sie den Wert in das Feld „Web-API-Endpunkt“ und das Feld „Scopes“. Im Feld „Scopes“ ergänzen Sie am Ende der URL noch „/.default“.

Ohne Client-Anmeldeinformationen können Sie jetzt die OAuth 2.0 Anmeldung durchführen. Rufen Sie die Seite an der Verbindung dafür auf und starten Sie den Anmeldeprozess. Als nächstes müssen Sie die Microsoft Anmeldung ausführen und der App den Zugriff gestatten. Mit dem Speichern der Verbindung wird ein Langzeit-Token ermittelt und gespeichert. Dieser wird bei Verwendung der Verbindung für die Anmeldung genutzt und aktualisiert.

1.12.1 Funktionen

Schema ermitteln

Das Schema der Verbindung wird über die Web-API ermittelt. Dafür wird der Endpunkt „EntityDefinitions“ verwendet. Die Änderungsinformation zur Entität wird über den Feldnamen „modifiedon“ identifiziert.

Lesen von Schema-basierten Daten

Das Lesen von Datensätzen erfolgt über die Web-API. Zusätzlich können über den Prozess noch weitere Filter im OData-Syntax definiert werden. Die Zugriff erfolgen in der Regel mit Universalprozessen. Die Besonderheiten von Lookup-Feldern werden durch die Verbindung behandelt. Eine Beschreibung dazu finden Sie in den Besonderheiten.

Lesen von Abfrage-basierten Daten

Das Lesen von Abfragen erfolgt ebenfalls über die Web-API. Dabei wird automatisch der Web-API-Endpunkt verwendet. Im Prozess definieren Sie lediglich den Teil der URL nach dem Endpunkt. Dabei verwenden Sie kein führenden Schrägstrich.

Schreiben von Daten

Das Schreiben von Datensätzen erfolgt über die Web-API. Dabei werden Anforderungen an die Übergabe von Lookup-Felder durch die Verbindung behandelt. Eine Beschreibung dazu finden Sie in den Besonderheiten. Das Löschen von Datensätzen wird unterstützt.

1.12.2 Einstellungen

Folgende Einstellungen können bereitgestellt werden.

Verzeichnis-ID (Mandant)

Diese ID definiert das Unternehmen und kann bei der Registrierung der Anwendung ausgelesen werden,

Anwendungs-ID (Client)

Die Anwendungs-ID der registrierten Anwendung.

Client-Geheimschlüssel

Ein geheimer Schlüssel der registrierten Anwendung. Dieser wird für das Client-Anmeldeverfahren benötigt.

Redirect URI

Für die Authentifizierung mit OAuth 2.0 ohne Client-Anmeldeinformationen muss eine Weiterleitungs-URI konfiguriert werden. Bei der Freigabe über den Syncler Administrator kann eine beliebige URI angegeben werden. Für die Freigabe über das Web-Frontend ist folgender Wert notwendig. Die Weiterleitungs-URI muss bei der registrierten Anwendung festgelegt werden.

Scopes

Diese Berechtigungen werden bei einer Benutzerzustimmung angefordert. Kopieren Sie den Wert aus dem Feld „Web-API-Endpunkt“ und ergänzen Sie am Ende der URL noch „/.default“. Der offline_access wird automatisch ergänzt.

Web-API-Endpunkt

Diese URL ist die Basis für alle lesenden und schreibenden Zugriffe auf die Web-API. Sie ist spezifisch für Ihre Instanz und kann den Entwicklerressourcen entnommen werden.

Client-Anmeldeinformationen verwenden

Mit „Ja“ erfolgt die Anmeldung mit Client-ID und dem Geheimschlüssel. Dieses Verfahren setzt einen Anwendungsbenutzer voraus. Diesen können Sie im PowerApps Admin Center anlegen. Er muss der registrierten App zugeordnet sein und über ausreichende Berechtigungen für die Synchronisation verfügen.

Auswahllisten abrufen und speichern

Die Aktualisierung des Schemas kann ebenfalls alle Optionssätze (Auswahlen) abrufen und intern speichern. Ein Optionssatz ist eine Liste von Wert-Text-Kombinationen, die für ein Auswahlfeld verwendet werden. Gespeicherte Auswahllisten können in der Transformation „Auswahllisten abbilden“ verwendet werden.

Siehe *Auswahllisten abbilden*

Optionswerte werden in Dynamics 365 CE numerisch gespeichert und sind damit nicht lesbar. Mit den Auswahllisten und der Transformation können Sie eine lesbare Variante erzeugen. Das Abfragen der Auswahllisten benötigt zusätzliche Zeit bei der Aktualisierung.

1.12.3 Besonderheiten

Lookup-Felder wie Suche oder Kunde unterscheiden sich in den Lese- und Schreibanforderungen zu anderen Feldern. Die Anforderungen werden durch die Verbindung weitestgehend kompensiert, damit in der Prozesskonfiguration einheitlich gearbeitet werden kann.

Werte werden beim Lesen durch die Web-API in einem Feld „_entityid_value“ bereitgestellt. Die Verbindung übernimmt diesen Wert in das eigentlich Feld. Dies ist nur beim Schema-basierten Lesen möglich. Im Abfrage-basierten Lesen werden alle Daten des Resultats direkt übergeben. In diesem Fall steht der Wert nicht unter der eigentlichen Feldbezeichnung zur Verfügung. Wenn nach Werten in diesen Feldern gefiltert werden soll, muss die Variante „_entityid_value“ verwendet werden.

Wenn ein Feld ein oder mehrere Lookup-Ziele hat, werden diese in der Feldbeschreibung im Schema dargestellt. Einfache Ziele werden durch die Verbindung automatisch behandelt, solange es sich um eine bekannte Zielentität handelt. Bei Zielen, die nicht in den verfügbaren Schemaobjekte vorhanden sind, muss die Wertübergabe anders erfolgen.

Beispiel Währung:

Die Entität „transactioncurrency“ ist bekannt und einem Feld „transactioncurrencyid“ kann direkt der ID-Wert zugewiesen werden.

Beispiel mit unbekannten Ziel:

Damit die Wertübergabe an die Web-API möglich ist, muss der Wert in Feldnotation angegeben werden. Für die Bezeichnung muss der CollectionName der Entität verwendet werden.

```
accounts|:|83883308-7ad5-ea11-a813-000d3a33f3b4|;|
```

Die Verbindung erzeugt in beiden Fällen daraus die benötigte Darstellung.

```
Feld@odata.bind : CollectionName(83883308-7ad5-ea11-a813-000d3a33f3b4)
```

Wenn mehrere Ziele möglich sind, z.B. beim Feldtyp Kunde, muss die Wertübergabe generell in Feldnotation erfolgen. Die Bezeichnung definiert dabei das gewünschte Ziel und der CollectionName wird über das Schemaobjekt ermittelt.

Beispiel Kunde für Kontakte:

```
account|:|83883308-7ad5-ea11-a813-000d3a33f3b4|;|
```

Die Verbindung erzeugt in beiden Fällen daraus die benötigte Darstellung.

```
Feld_account@odata.bind : accounts(83883308-7ad5-ea11-a813-000d3a33f3b4)
```

1.13 Evalanche

Evalanche ist eine Marketing Automation Plattform und kann u.a. für den Versand und die Auswertung von Newslettern genutzt werden. Mit dieser Verbindung können Sie die Empfänger Ihrer Kampagnen mit jedem beliebigen System synchronisieren. Sie erhalten Zugriff auf verschiedene Auswertungen und können diese in andere Systeme übertragen.

Die Verbindung nutzt die SOAP-Schnittstelle von Evalanche für die Kommunikation. Für die Einrichtung benötigen Sie nur die Domain, Benutzername und Passwort.

Eine eingerichtete Verbindung ist mit einem Mandanten und einem Pool verbunden.

1.13.1 Funktionen

Schema ermitteln

Das Schema der Verbindung wird über die SOAP-API ermittelt. Es stehen Mailing, MailingDetails, MailingStatistics, Article, Profile, Recipient, Bounce, Unsubscription, GrantedPermission, RevokePermission, Impression (Openings) und Clicks zur Verfügung.

Lesen von Schema-basierten Daten

Das Lesen von Datensätzen erfolgt über die SOAP-API. Die Zugriff erfolgen in der Regel mit Universalprozessen. Die Verbindung bereitet Daten auf, damit diese möglichst einfach verwendet werden können. Zum Beispiel werden Clicks mit der Url des Links angereichert. Profile (Empfänger) können beliebig angelegt, aktualisiert oder gelöscht werden. Auch können Berechtigungen gesetzt oder entfernt werden. Profile verfügen über ein Änderungsdatum, was eine Änderungs-basierte Synchronisation ermöglicht.

Lesen von Abfrage-basierten Daten

Diese Funktion wird nicht unterstützt.

Schreiben von Daten

Das Schreiben von Datensätzen erfolgt über die SOAP-API. Für das Schreiben oder Löschen können die Entitäten Profile und RevokePermission genutzt werden.

1.13.2 Einstellungen

Folgende Einstellungen können bereitgestellt werden.

Login Domäne

Die Domäne Ihrer Anmeldung.

Benutzername

Der Benutzername Ihrer Anmeldung.

Passwort

Das Passwort Ihrer Anmeldung.

Mandant ID

Die ID des zuverwendenden Mandanten. Wenn nicht eingetragen wird, wird der Wert ggf. aus dem Namen ermittelt oder der erste verfügbare Mandant ausgewählt.

Mandant Name

Der Name des zuverwendenden Mandanten. Wenn nicht eingetragen wird, wird der Wert ggf. aus der ID ermittelt oder der erste verfügbare Mandant ausgewählt.

Pool ID

Die ID des zuverwendenden Pools. Wenn nicht eingetragen wird, wird der Wert ggf. aus dem Namen ermittelt oder der erste verfügbare Pool des Mandanten ausgewählt.

Pool Name

Der Name des zuverwendenden Pools. Wenn nicht eingetragen wird, wird der Wert ggf. aus der ID ermittelt oder der erste verfügbare Pool des Mandanten ausgewählt.

1.14 SalesViewer

SalesViewer ist ein Anbieter zum Überwachen Ihrer Webseite. Anonyme Website-Besucher können damit in Firmendaten aufgelöst werden. Zusätzlich stehen Informationen zum Besuch und den Interessen zur Verfügung.

Die SalesViewer-Verbindung bietet Schemaobjekte für die Sitzung, den Besuch und die Firma an. Diese können mit Universalprozessen in jedes beliebige System übertragen werden.

Die Daten können dabei kontinuierlich abgerufen und verarbeitet werden.

Für die Einrichtung der Verbindung benötigen Sie lediglich den API Key. Dazu melden Sie sich im SalesViewer Portal an und rufen den Menüpunkt „Projekte und Trackingcode“ auf. Hier können Sie den API Key einsehen.

Beim Anlegen der SalesViewer-Verbindung übertragen Sie nur diesen API Key in das Feld „API Key“ und speichern die Verbindung.

1.14.1 Funktionen

Schema ermitteln

Die Verbindung stellt automatisch die Objekte company, session und visit zur Verfügung. Die Objekte enthalten eine Datum für die kontinuierliche Verarbeitung und einen Schlüssel für die Beziehungen zwischen den Objekten. Dabei sind Besuche (visit) einer Sitzung (session) und Sitzungen einer Firma (company) zugeordnet.

Lesen von Schema-basierten Daten

Das Lesen kann vollständig oder kontinuierlich erfolgen. Das gezielte Abrufen eines Datensatzes wird nicht unterstützt. Bei einer Verarbeitung aller Objekte sollte deshalb die Reihenfolge der Beziehungen beachtet und dies ggf. durch einen Ablauf gesteuert werden.

Lesen von Abfrage-basierten Daten

Diese Funktion wird nicht unterstützt.

Schreiben von Daten

Diese Funktion wird nicht unterstützt.

1.14.2 Einstellungen

Nur die Einstellung „API Key“ muss festgelegt werden. Alle anderen Einstellungen können ignoriert werden.

1.15 Kuehne + Nagel

Kuehne + Nagel bietet als Logistikunternehmen eine Schnittstelle zum Abrufen des Frachtstatus. Mit dieser Verbindung können aktuell Container-Informationen abgerufen und verarbeitet werden.

Die Kuehne und Nagel Verbindung bietet ein Schemaobjekt für die Conatiner-Information und unterstützt auch Abfragen zu diesen Daten.

Für die Einrichtung werden die Anwendungszugangsdaten benötigt. Melden Sie sich im Portal an und rufen Sie den Menüpunkt „My Applications“ auf. Hier finden Sie die vorhandenen Anwendungen. Übertragen Sie die Zugangsdaten (user, password, client-id, client-secret) in die gleichnamigen Felder der Verbindung.

1.15.1 Funktionen

Schema ermitteln

Die Verbindung stellt automatisch die Objekte containerInfo zur Verfügung.

Lesen von Schema-basierten Daten

Das Lesen wird nur mit einem Filter unterstützt. Dieser muss aus zwei Informationen bestehen, die durch ein Semikolon getrennt sind. Der erste Wert legt den „referenceType“ fest. Der zweite Wert muss den Wert enthalten. Als Referenz-Typen stehen folgende zur Verfügung:

- ALL_CUSTOMER_REFERENCES
- SPECIFIC_CUSTOMER_REFERENCE
- BILL_OF_LADING
- DELIVERY_LOCATION
- TRACKING_NUMBER
- CONTAINER_NUMBER

Eine Abfrage eines Container wird wie folgt formuliert.

CONTAINER_NUMBER;ABCD12345

Eine Abfrage aller Kundenreferenzen würde wie folgt formuliert werden.

ALL_CUSTOMER_REFERENCES;%

Lesen von Abfrage-basierten Daten

Diese Funktion arbeitet wie das schema-basierte Lesen. Die Abfrage wird auch mit zwei Werten formuliert.

Damit kann die Abfrage flexibel in Transformationen genutzt werden, um vorhandene Datensätze mit Conatiner-Informationen anzureichern.

Schreiben von Daten

Diese Funktion wird nicht unterstützt.

1.15.2 Einstellungen

Für den Zugriff muss der Benutzername, das Passwort, die Client-ID und der Geheimschlüssel festgelegt werden.

1.16 Databyte

Databyte ist eine Informationsplattform für detaillierte Firmendaten. Mit dieser Verbindung stehen verschiedene Funktionen der Plattform zur Verfügung, die im Folgenden beschrieben werden.

Die Verbindung erzeugt Auswahllisten und Schemaobjekte für die Bedienung.

1.16.1 Schemaobjekte

companylist

Dieses Objekt wird für Firmenlisten verwendet. Damit können vorhandene Firmenlisten abgerufen, angelegt, umbenannt oder gelöscht werden.

companylist_copy

Mit diesem Objekt kann eine vorhandene Firmenliste kopiert werden. Dazu wird der Wert „fls“ mit der ID der Quelle geschrieben und die Antwort enthält die ID der Kopie im Feld „newfls“.

companylist_data

Mit diesem Objekt wird auf den Inhalt einer Firmenliste zugegriffen. Für das Lesen muss die ID einer Firmenliste als Filter übergeben werden. Als Antwort wird eine Liste der Firmen-IDs im Feld „data“ zurückgegeben. Der Inhalt einer Firmenliste kann für die Überwachung von Änderungen verwendet werden.

Dieses Objekt kann auch zum hinzufügen von Firmen zu einer Liste genutzt werden. Dafür wird das Objekt zum Schreiben verwendet. Mit dem Feld „ssid“ können Sie sich auf eine vorherige Suchanfrage beziehen. Mit dem Feld „umfang“ steuern Sie, welche Firmen hinzugefügt werden sollen. Der Wert „2“ erwartet einen Bereich mit „von“ - „bis“ aus dem Suchergebnis. Der Wert „3“ erwartet einen Bereich mit „bereich“ für jeden x-ten Datensatz. Der Wert „4“ arbeitet unabhängig von einer Suche und fügt alle Firmen aus dem Feld „firmen“ der Liste hinzu. Alle anderen Werte fügen das gesamte Suchergebnis der Liste hinzu.

Ein Lösch-Operation leert die Liste.

companylist_truncate

Mit diesem Objekt kann eine vorhandene Firmenliste geleert werden. Dazu wird der Wert „fls“ mit der ID der Liste geschrieben.

datamatch

Mit diesem Objekt können vorhandene Projekte abgerufen oder neue Projekte angelegt werden. Außerdem können vorhandene Projekte gelöscht werden. Wird das Filtern oder Suchen mit den Standard-Prozessfeldern ausgelöst, wird der Wert direkt mit dem Names des Projektes verglichen und alle Übereinstimmungen werden zurückgegeben.

datamatch_data

Mit diesem Objekt werden Vergleichsdatensätze zu einem Projekt hinzugefügt. Dieses Objekt kann mit Prozessen für die Massenverarbeitung eingesetzt werden. Damit können bis zu 500 Datensätze mit einem Aufruf übertragen werden.

Wird dieses Objekt gelesen, ist eine Projekt-ID als Filter erforderlich. Sollte das Projekt noch nicht gestartet sein, wird das Lesen mit einer Fehlermeldung abgebrochen. Sollte das Projekt bereits gestartet, aber noch nicht abgeschlossen sein, wird der Projektstatus alle 10 Sekunden geprüft, bis das Projekt abgeschlossen und das Ergebnis abgerufen werden kann. Dies kann manuell abgebrochen werden.

Ein vorhandenes Resultat kann nur einmalig abgerufen werden. Danach wird das Projekt automatisch entfernt. Damit die Daten möglichst flexibel zur Verfügung stehen, wird das Resultat in einzelnen Datensätzen im Änderungsspeicher abgelegt. Dies ermöglicht ein wiederholtes Lesen durch den Prozess. Jeder Datensatz enthält dabei das Match-Ergebnis und ggf. eine Ziel-ID für die weitere Verarbeitung. Die Datensätze werden dem aktuellem Prozess zugeordnet und zurückgegeben.

Das kunde_id sollte mit Ihrer externen ID gefüllt werden, damit eine Ergebniszuordnung sichergestellt ist.

datamatch_start

Dieses Objekt startet die Projektverarbeitung. Durch das Speichern der Projekt-ID beginnt die Verarbeitung.

datamatch_abort

Dieses Objekt stoppt die Projektverarbeitung. Durch das Speichern der Projekt-ID wird die Verarbeitung abgebrochen.

searchrequest

Mit diesem Schema wird eine Suchanfrage formuliert und übertragen. Die Suche wird durch das Schreiben des Objektes ausgelöst. Das Ergebnis wird im Änderungsspeicher abgelegt und kann über die generierte ID „ssid“ wieder abgerufen werden. Hierfür wird das Schmea mit einem Filter gelesen.

exportrequest

Dieses Objekt fordert durch das Schreiben einen Export an. Dieser Export wird vom Kontingent der Download-Einheiten abgezogen. Ähnlich dem Hinzufügen von Daten zu einer Firmenliste, stehen verschiedene Optionen für die Auswahl der Exportdaten zur Verfügung.

Der Parameter „export“ steuert die Informationstiefe des Exports. Abhängig von dem übergebenen Wert werden die exportierten Daten unter einem anderen Zielschema im Änderungsspeicher abgelegt. Gültige Werte sind std, erw und pro.

Mit dem Feld „ssid“ können Sie sich auf eine vorherige Suchanfrage beziehen. Mit dem Feld „umfang“ steuern Sie, welche Firmen exportiert werden sollen. Der Wert „2“ erwartet einen Bereich mit „von“ - „bis“ aus dem Suchergebnis. Der Wert „3“ erwartet einen Bereich mit „bereich“ für jeden x-ten Datensatz. Der Wert „4“ arbeitet unabhängig von einer Suche und fügt alle Firmen aus dem Feld „firmen“ der Liste hinzu. Alle anderen Werte fügen das gesamte Suchergebnis der Liste hinzu.

Weitere Parameter sind:

- person: Anzahl Personen die pro Datensatz exportiert werden sollen (z.B. 0,1)
- zusatz: z.B. passive und schwebende Firmen mit exportieren

exportstd

Ein angeforderter Export mit dem Typ „std“ wird unter diesem Schema im Änderungsspeicher abgelegt. Das Lesen des Typs greift auf den Änderungsspeicher zu und liest einzelne oder alle Datensätze. Der Prozess wird dabei nicht eingeschränkt.

exporterw

Ein angeforderter Export mit dem Typ „erw“ wird unter diesem Schema im Änderungsspeicher abgelegt. Das Lesen des Typs greift auf den Änderungsspeicher zu und liest einzelne oder alle Datensätze. Der Prozess wird dabei nicht eingeschränkt.

exportpro

Ein angeforderter Export mit dem Typ „pro“ wird unter diesem Schema im Änderungsspeicher abgelegt. Das Lesen des Typs greift auf den Änderungsspeicher zu und liest einzelne oder alle Datensätze. Der Prozess wird dabei nicht eingeschränkt.

monitoringrequest

Zum Anfordern oder Abbestellen einer Überwachung wird dieses Schema schreibend verwendet. Überwachungen stehen im Zusammenhang zu Firmenlisten. Beim Einlesen des Verbindungsschemas kann eine Firmenliste angegeben werden oder es wird eine bestimmt. Zu dieser Liste werden die verfügbaren Monitor-Eigenschaften ermittelt und als Auswahlliste abgespeichert. Die Werte dieser Auswahlliste werden als Liste mit dem Parameter „merkmale“ übergeben. Zusätzlich muss eine Firmenliste, ein Intervall und das Aktiv-Kennzeichen gesetzt werden.

postbox

Alle Überwachungen senden ggf. eine Nachricht in die Postbox, sobald eine Änderung zu einem Merkmal erkannt wurde. Jede Nachricht kann eine Liste von Merkmalen und Firmen enthalten. Diese Daten werden geschachtelt bereitgestellt. Außerdem steht eine Zusammenstellung aller betroffenen Firmen-IDs bereit.

Wenn eine Nachricht mit dem Parameter „confirm“ gespeichert wird, gilt sie als bestätigt und wird nicht erneut gelesen.

1.16.2 Die Suche nach Firmen

Für die Suche nach Firmen wird das Schemaobjekt „searchrequest“ verwendet. Die Suche ist hier als Auftrag und nicht als Lesen von Daten umgesetzt. Dies hat den Vorteil, dass komplexe Suchanfragen einfach durch Feldzuordnungen formuliert werden können. Nutzen Sie das Schemaobjekt „searchrequest“ zum Schreiben z.B. mittels eines Universalprozesses.

Das Objekt verfügt über verschiedene Bereich, wie firma, anschrift oder branche. Jeder Bereich ist mehrfach vorhanden. Eine Zuordnung in verschiedenen Instanzen wird als Oder-Suche interpretiert. Zuordnungen innerhalb eines Bereiches sind Und-verknüpft. Verschiedene Bereiche sind ebenfalls Und-verknüpft.

Ordnen Sie „firma.1.firma“ den gewünschten Suchnamen zu. Ordnen Sie „anschrift.1.ort“ und „anschrift.2.ort“ jeweils einen Stadtnamen zu. Das Resultat ist eine Suche des Firmennamen in beiden Orten (Oder-Beziehung).

Das Ergebnis der Suche wird im Änderungsspeicher abgelegt und kann von dort durch einen Folgeprozess oder Ablaufschritt verarbeitet werden. Es wird ein geschachteltes Objekt gespeichert, mit den gekürzten Firmendaten als Liste. Mit der „ssid“ kann das Suchergebnis für Firmenlisten oder Exporte verwendet werden. Außerdem werden alle IDs der gefundenen Firmen als Feld bereitgestellt.

1.16.3 Hinzufügen von Firmen zu einer Liste

Um Firmen einer Firmenliste hinzuzufügen wird das Schemaobjekt „companylist_data“ schreibend verwendet. Die Quelle der Daten kann eine vorangegangene Suche oder eine vorhandene Auflistung von Firmen-IDs sein.

Mit dem Feld „ssid“ können Sie sich auf eine vorherige Suchanfrage beziehen. Mit dem Feld „umfang“ steuern Sie, welche Firmen hinzugefügt werden sollen. Der Wert „2“ erwartet einen Bereich mit „von“ - „bis“ aus dem Suchergebnis. Der Wert „3“ erwartet einen Bereich mit „bereich“ für jeden x-ten Datensatz. Der Wert „4“ arbeitet unabhängig von einer Suche und fügt alle Firmen aus dem Feld „firmen“ der Liste hinzu. Alle anderen Werte fügen das gesamte Suchergebnis der Liste hinzu.

Kombinieren Sie zwei Prozesse innerhalb eines Ablaufes. Der erste Prozess führt eine Suche aus. (siehe „Die Suche nach Firmen“) Es kann sich um einen Universalprozess mit individueller Datenquelle oder um einen reinen Schreibprozess handelt, wo die Suchkriterien im Prozess definiert werden.

Der zweite Prozess ist ein Universalprozess mit „searchrequest“ als Quelle und „companylist_data“ als Ziel. Mittels Parameter und Feldzuordnungen können Sie die gewünschte Datenübernahme definieren.

Der Ablauf führt den ersten Prozess manuell aus. Der zweite Prozess wird mit den Zieldaten des Vorgängers ausgeführt. Diese entsprechen dem Ergebnis der Suchanfrage. Nach Abschluss des Ablaufs ist die Suche der Firmenliste hinzugefügt worden.

Zur Auswahl der Firmenliste ist die ID einer Liste erforderlich. Diese kann fest per Transformation oder durch andere Vorgängerprozesse bereitgestellt werden.

1.16.4 Der Export von Firmen und Übertragung in ein externes System

Ähnlich wie dem Hinzufügen von Daten zu einer Firmenliste muss dem Export eine Auswahl von Daten vorangehen. Hier besteht ebenfalls die Möglichkeit eine Suche auszuführen oder ein Suchergebnis zu verwenden. Es kann aber auch direkt mit einer ID-Liste gearbeitet werden, die z.B. durch ein externes System vorgehalten wird. Ebenso kann die ID-Liste einer Überwachungsnachricht zu Änderungen an Merkmalen entnommen werden.

In diesem Beispiel soll eine Suchanfrage genutzt werden.

Kombinieren Sie drei Prozesse innerhalb eines Ablaufes. Der erste Prozess führt eine Suche aus. (siehe „Die Suche nach Firmen“) Es kann sich um einen Universalprozess mit individueller Datenquelle oder um einen reinen Schreibprozess handelt, wo die Suchkriterien im Prozess definiert werden.

Der zweite Prozess ist ein Universalprozess mit „searchrequest“ als Quelle und „exportrequest“ als Ziel. Mittels Parameter und Feldzuordnungen können Sie die gewünschte Datenübernahme definieren.

Mit dem Feld „ssid“ können Sie sich auf eine vorherige Suchanfrage beziehen. Mit dem Feld „umfang“ steuern Sie, welche Firmen hinzugefügt werden sollen. Der Wert „2“ erwartet einen Bereich mit „von“ - „bis“ aus dem Suchergebnis. Der Wert „3“ erwartet einen Bereich mit „bereich“ für jeden x-ten Datensatz. Der Wert „4“ arbeitet unabhängig von einer Suche und fügt alle Firmen aus dem Feld „firmen“ der Liste hinzu. Alle anderen Werte fügen das gesamte Suchergebnis der Liste hinzu.

Zusätzlich muss noch die Informationstiefe mit „export“ festgelegt werden. Dies entscheidet darüber, mit welchem Quellschema im dritten Schritt gearbeitet wird.

Das Resultat des zweiten Prozesses sind exportierte Datensätze im Änderungsspeicher. Der dritte Prozess auf der Basis eines Universalprozesses hat als Quelle das festgelegte Exportschema und als Ziel ein beliebiges externes System.

1.16.5 Eine Data-Match mit vorhandenen Daten

Neben der Suche nach Firmen mittels Kriterien kann auch ein Abgleich mit Ihren Daten erfolgen. Diese werden in der Firmendatenbank gesucht und als Reultat wird eine oder mehrere potetnielle Treffer zurückgeliefert. Mit den so generierten ID-Abbildungen können Firmenliste gefüllt oder Export zur Anreicherung und Aktualisierung Ihres Datenbestandes durchgeführt werden.

Wir unterteilen das Beispiel in zwei Bereiche. Der erste hier beschriebene Bereich beschreibt die Durchführung des Data-Matchings bis zum Erhalt der IDs. Der zweite Bereichsind Folgeaktivitäten, die bereits im vorangegangenen beschrieben wurden.

Für das Data-Matching kombinieren wir 4 Prozesse in einem Ablauf. Die Anzahl ist hier höher, da kleiner Zwischenschritte notwendig sind.

Der erste Prozess kann ein Universalprozess oder Schreibprozess sein. Seine Aufgabe ist die Anlage eines neuen Projektes. Projekte sind der Rahmen des Data-Matchs und werden nach Abschluss automatisch wieder entfernt. Als Ziel wird das Schema „datamatch“ verwendet. Es muss ein Name und ein Typ für die Art des Vergleichs angegeben werden. Durch das Schreiben wird eine ID generiert, die in den folgenden Schritt benötigt wird. Deshalb muss dieser Wert im Ablaufschritt in die Quelldaten übernommen werden.

Der zweite Prozess überträgt die Stammdaten aus Ihrem System. Als Ziel wird das Schema „datamatch_data“ verwendet. Dieses wird auch in Bulk-Prozessen unterstützt, was die Übertragung deutlich beschleunigt. Sie müssen in diesem Prozess mindestens den Firmennamen und die Anschrift übermitteln. Die „kunde_id“ Spalte können Sie für Ihre System-ID verwenden, damit Sie die Resultate später zuordnen können. Auch die Projekt-ID muss hier zugeordnet werden. Mittels Parameter und Abfrage-Platzhalter kann diese im Ablauf übernommen werden.

Nachdem alle Daten in das Projekt übertragen wurden, muss dieses gestartet werden, damit der Abgleich durchgeführt wird. Für diese Aktion steht das Schema „datamatch_start“ zur Verfügung. Durch eine Schreibaktion mit der Projekt-ID wird der Vorgang gestartet. Abhängig von der Anzahl der Daten und ggf. Projekten benötigt der mehrere Minuten für die Durchführung.

Der vierte Prozess verwendet das Schema „datamatch_data“ zum Lesen und muss hierfür die Projekt-ID als Filter übergeben. Das Lesen der Daten wartet auf den Abschluss des Projektes. Dieser wird alle 10 Sekunden geprüft, bis das Projekt abgeschlossen wurde. Nach Abschluss werden die Daten mit dem Match-Ergebnis abgerufen und in den Änderungsspeicher mit einem Prozessbezug gespeichert. Dieser Abruf ist nur einmal möglich. Ein erneutes Lesen wird nur die Daten aus dem Änderungsspeicher liefern. Der vierte Prozess kann nun die generierten IDs zurück in Ihr System schreiben und Sie können diese für Folgeaktivitäten nutzen.

Nur der erste Prozess wird im Ablauf manuell ausgeführt. Der Prozess für die Datenübertragung benötigt die Projekt-ID. Bei einer manuellen Ausführung muss diese extern bereitstehen. Sie können den Prozess aber auch mit den Vorgängerdaten als Filter nutzen, um die ID im Ablauf zu übergeben. Der Projektstart kann direkt die Daten des ersten Prozesses wiederverwenden. Der letzte Prozess verwendet die Daten des ersten als Filter, damit das Projekt gezielt abgerufen werden kann.

1.16.6 Das Überwachen von Merkmalsänderungen

Die Basis dieser Überwachung ist eine gefüllte Firmenliste. Der Weg dorthin ist oben bereits beschrieben. Als nächstes müssen die Merkmale einmalig abonniert werden. Verwenden Sie dafür die Angaben aus der Auswahlliste „Monitoring“ und das Schema „monitoringrequest“. Sobald die gewünschten Merkmale aktiviert wurden, kann ein Prozess für das Abrufen der Nachrichten eingerichtet werden. Das kleinste Intervall der Überwachung ist ein Tag.

Verwenden Sie als Quellschema „postbox“. Da die Nachricht bereits eine vollständige ID-Liste enthält, kann als Zielschema bereits ein „exportrequest“ eingestellt werden. Damit eine Nachricht nur einmal verarbeitet wird, nutzen Sie die Rückschreiben-Funktion und übergeben ein „true“ für das Feld „confirm“. Das Schreiben des „exportrequest“ füllt wieder den Änderungsspeicher, welcher dann mit einem weiteren Prozess abgearbeitet wird.

1.17 HubSpot

HubSpot ist ein CRM-System für die Bereiche Marketing, Vertrieb, Kundenservice und Content-Management. Für die Verbindung wird die HubSpot REST API eingesetzt. Um diese verwenden zu können, müssen folgende Einrichtungsschritte ausgeführt werden.

Für den Zugriff auf HubSpot können Sie mit einer privaten oder öffentlichen App arbeiten.

Für die Nutzung der privaten App führen Sie bitte folgende Schritte aus. Rufen Sie mit administrativen Rechten das Einstellungsmenü in Ihrem Account auf. Im Bereich Integrationen müssen Sie eine private App erstellen. Klicken Sie dazu unter „Private Apps“ auf „Private App erstellen“. Vergeben Sie einen Namen in den grundlegenden Informationen. Unter „Bereiche“ müssen Sie die Berechtigungen für die App konfigurieren. Nur freigegebene Bereiche und Funktionen können genutzt werden.

1.17.1 Funktionen

Schema ermitteln

Das Schema der Verbindung wird über die Web-API ermittelt. Dafür werden die Properties der einzelnen Objekttypen abgerufen.

Lesen von Schema-basierten Daten

Das Lesen von Datensätzen erfolgt über die Web-API. Einzelne Datensätze werden direkt abgerufen. Listen von Datensätzen werden über die Search-Funktion abgerufen. Im Where-Parameter kann hierfür das Json für die filterGroups übergeben werden.

```
{
    "filterGroups":[
        {
            "filters":[
                {
                    "propertyName": "firstname",
                    "operator": "EQ",
                    "value": "Alice"
                }
            ]
        }
    ]
}
```

Wenn der Filter kein gültiges Json enthält, wird der übergebene Wert als query-Parameter für die Suche verwendet. Dieser wird in verschiedenen Standardfeldern des Objektes gesucht.

Die Search-Funktion kann nur maximal 10.000 Datensätze zurückliefern. Sollten Sie mehr Datensätze abrufen wollen, lassen Sie den Where-Parameter im Prozess leer und verwenden Sie stattdessen den zweiten Filter des Prozesses.

Lesen von Abfrage-basierten Daten

Diese Funktion wird nicht unterstützt.

1.18 KlickTipp

KlickTipp ist ein Tool für Newsletter und Marketing Automation. Die Datenstruktur ist geprägt von Tagging, um Attribute, Aktivitäten und Historien zu Kontakten zu erfassen. Für die Anmeldung muss an der Verbindung der Benutzername und das Passwort eingegeben werden.

1.18.1 Schemaobjekte

Folgende Schemaobjekte stehen zur Verfügung.

tag

Dieses Objekt ist die Basis für das Tagging und die Abfrage von anderen Objekten. Tags können nur einzeln per ID oder komplett abgerufen werden. Die ID kann für AdHoc-Ausführungen direkt oder als Filter in Feldnotation übergeben werden. Als Filter kann in Feldnotation auch eine Namenssuche übergeben werden. Tags können erzeugt und aktualisiert werden.

subscriber

Subscriber sind die Kontakte oder Empfänger, die per Newsletter angeschrieben werden. Das Schema enthält sowohl Standard- also auch benutzerdefinierte Felder. Zugeordnete Tags sind als Array-Feld lesend verfügbar. Subscriber können einzeln per ID oder komplett abgerufen werden. Außerdem ist eine Abfrage mit einem Filter in Feldnotation möglich. Hierfür stehen die Felder id und email zur Verfügung. Eine Abfrage über Tags muss mit dem Schemaobjekt taggedsubscriber erfolgen. Die Array-Felder in den Daten enthalten entweder eine ID als Wert oder eine ID als Name und ein Datum als Wert. Das Datum liegt als Unix-Timestamp vor. Subscriber können angelegt und aktualisiert werden. Über die Objektgruppe addTag können direkt mehrere Tags zugeordnet werden. Dafür muss auch das Feld email in dieser Gruppe zugeordnet werden.

tagsubscriber

TagSubscriber dient zum Markieren von vorhandenen Kontakten. Das Schema wird für schreibende Zugriffe verwendet. Es muss die Emailadresse und ein Array von TagIds übergeben werden.

taggedsubscriber

Dieses Schema wird für das Lesen verwendet. Es muss eine ID oder ein Filter übergeben werden, der als TagID interpretiert wird. Das Ergebnis ist eine Liste von Datensätzen mit SubscriberID, Datum und TagID. Diese Subscriber sind mit dem übergebenen Tag verknüpft. Das Datum ist der Zeitpunkt der Verknüpfung.

CSV (Draft) ===

1.19 E-Mail (Draft)

Diese Verbindung kann Postfächer verarbeiten oder E-Mails versenden. Beim Lesen von E-Mails steht eine EML-Version der Nachricht als Base64 im Feld „FileContent“ zur Verfügung.

1.20 Sage 100 (Draft)

Bezüglich der Zubehörartikelunterstützung:

Besitzt ein Artikel automatische Zubehörartikel werden diese entsprechend hinzugefügt. Gehören zu einem Artikel optionale Zubehörartikel, muss dessen Hinzufügen mittels des Positionrequestproperties ‚ForceOptionalZubehoerartikel‘ erzwungen werden. Diese erzwungenen optionalen Zubehörartikelpositionen werden mit Menge = 0 dem Beleg hinzugefügt und müssen im weiteren Verlauf manuell im UI geändert werden.

Im Request Objekt steht jetzt das neue Property „VkAngebotsstatus“ vom Typ int zur Verfügung.

Folgende Werte könnt ihr dort setzen:

None = 0, Neuanlage = 1, Abgelehnt = 3, Angenommen = 4, Erfuellt = 5

Sendet ihr keinen Wert für VkAngebotsstatus, also NULL, wird automatisch wie bisher 1 gesetzt. Im UI steht dann nicht Neuanlage , sondern Neu.

1.21 Sage CRM (Draft)

Beschreibung

1.21.1 Funktionen

Schema ermitteln
Laden von Quelldaten
Laden von Schema-basierten Daten
Laden von Abfrage-basierten Daten
Schreiben von Daten
Schreiben von Bulk-Daten
Diese Funktion wird nicht unterstützt.

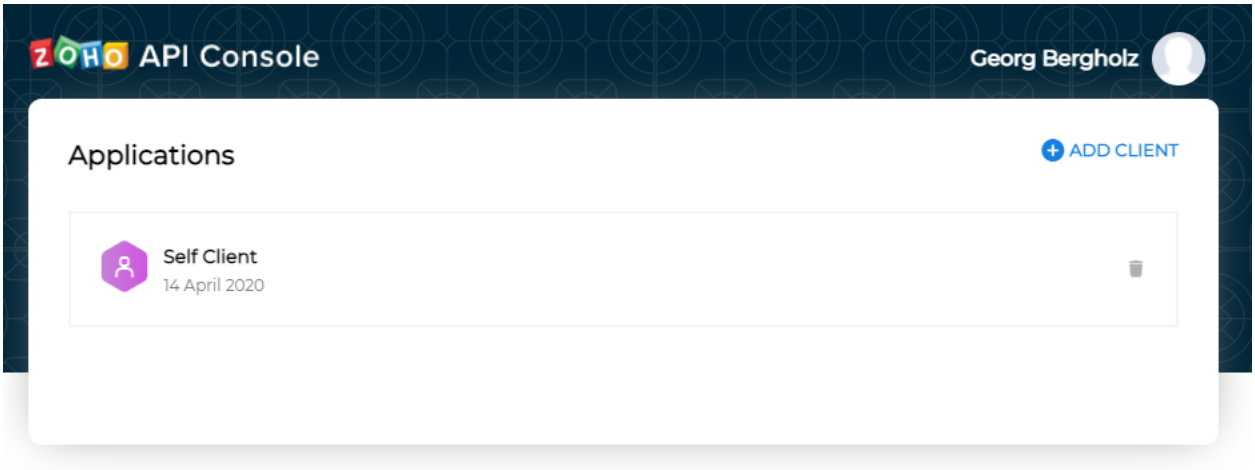
1.21.2 Einstellungen

1.21.3 Besonderheiten

Die Kommunikation erfolgt über den SOAP Webservice des CRMs. Dieser bringt einige Besonderheiten mit sich.

> Auswahlfelder werden nicht als Code / DB-Wert zurückgegeben, sondern als Übersetzung in der Standardsprache des CRMs. Dies gilt nicht für den Feldtyp „Intelligente Auswahl“. Dieser liefert den Code / DB-Wert des Feldes zurück.
>
> Felder werden ohne den DB-Präfix verwendet. Dadurch stehen Felder mit einer Zahl nach dem Präfix im Namen nicht zur Verfügung, da dies zu einem unerlaubten Namen im XML führen würde.
>
> Felder mit dem Kürzel „**sp_**“ im Namen führen zu einer kompletten Ablehnung der Entität. Ursache ist wahrscheinlich eine vermutete SQL-Injektion.
>
> Wird einem Kommunikationsdatensatz ein CRM-Benutzer als Organisator zugewiesen, wird sichergestellt, dass dieser auch als Benutzer zugewiesen ist.

1.22 Die Zoho CRM Verbindung (Draft)



Beschreibung

1.22.1 Funktionen

Schema ermitteln
Laden von Quelldaten
Laden von Schema-basierten Daten
Laden von Abfrage-basierten Daten
Schreiben von Daten
Schreiben von Bulk-Daten
: Diese Funktion wird nicht unterstützt.

1.22.2 Einstellungen

1.22.3 Besonderheiten

Wenn Daten per Transformation geladen werden, wird der Filter als Search Criteria übergeben. Eine Ausnahme stellen Photos dar. Hier wird eine Feldnotation aus Modulname und Modul-ID erwartet.

RelationModule|:|**Accounts**|;|RelationId|:|**23456789**|;|

Das Photo steht dann als Base64 zur Verfügung und kann z.B. in Serienbriefen genutzt werden.

1.23 Emarsys (Draft)

Emarsys ist ein Newsletter- und Online-Marketing-Anbieter. Die Verbindung stellt nicht alle Funktionalitäten bereit.

1.23.1 Funktionen

Schema ermitteln
Das Schema für einen Kontakt wird über das Abrufen der Felder erzeugt. Das Schema für External Events ist fest vorgegeben.

Laden von Quelldaten
Das Laden von Datensätzen als Quelle für die Ausführung eines Prozesses wird nicht unterstützt.

Laden von Schema-basierten Daten
Es können Kontakte über den ID-Wert (Emailadresse) oder einer Suchbedingung in Feldnotation abgerufen werden. Damit ist eine Aktualisierung von Kontakten und eine eingeschränkte Übereinstimmungssuche nach Kontakten möglich.

Laden von Abfrage-basierten Daten
Diese Funktion wird nicht unterstützt.

Schreiben von Daten
Es können Kontakte angelegt oder aktualisiert werden. Das Triggern eines External Events ist mittels Schreibaktion möglich.

1.23.2 Einstellungen

Für den Syncler müssen folgende Einstellungen bereitgestellt werden.

Benutzername

-

Passwort

-

Optional können diese Einstellungen bereitgestellt werden.

1.23.3 Besonderheiten

Es können an der Verbindung External Event IDs definiert werden, die nach der Anlage oder Aktualisierung eines Kontaktes getriggert werden.

1.24 Sage WinCarat (Draft)

1.25 Die SQL- und Dateibrücke

Hierbei handelt es sich um eine zusätzliche API, die separat installiert werden muss. Die API ermöglicht das Ausführung von SQL Abfragen und das Herunterladen von Dateien, ohne direkte SQL Verbindung oder Dateisystemzugriff.

Mittels einer Installationsroutine wird die API als Windowsdienst eingerichtet. Zusätzlich steht eine Administrationsanwendung zur Verfügung, um die Zugangsdaten, die Url und den Port der API einzustellen. Eine Verwendung über einen Reverse Proxy ist ebenfalls möglich. Bei der Url und dem Port handelt es sich um lokale Daten, die vom Dienst abgehört werden. Diese können sich den öffentlichen Angaben im Syncler unterscheiden, abhängig von der Infrastruktur.

Es findet keine weitere Datenspeicherung von Zugangsdaten auf diesem System statt.

Im Syncler an den betreffenden Verbindungen werden die Zugangsdaten zur API eingerichtet. Ab da finden alle direkten SQL- oder Dateizugriffe über die API statt.

Für den Zugriff auf die API wird ein Anmeldeverfahren und eine dynamische Verschlüsselung angewendet. Alle SQL Zugangsdaten werden beim API Zugriff dynamisch verschlüsselt und an die API gesendet.

Je nach Art des Einsatzes stellt die API dann die SQL-Verbindung her und führt die Abfrage aus oder liest eine Datei ein. Das Resultat wird an die aufrufende Verbindung übergeben.

1.26 Das OAuth 2.0 Verfahren (Draft)

Für den SMTP-Versand in On-premises Installation, die Email-Verbindung und die Graph-Verbindung wird das OAuth 2.0 Anmeldeverfahren für Microsoft 365 unterstützt.

Die folgenden Schritte beschreiben die Vorbereitung und Einrichtung des Verfahrens.

1.27 Besonderheiten

Sie können bestehende Verbindungen auch über das Aktionen-Menü kopieren. Dabei werden aber aus Sicherheitsgründen keine Passwörter kopiert. Sie müssen die Verbindung bearbeiten und ggf. alle Passwörter erneut eingeben.

KAPITEL 2

Prozesse

Ein Prozess realisiert eine Aufgabe, z.B. das Synchronisieren von Daten oder das Auslösen einer Benachrichtigung. Dabei haben Prozesse je nach Einsatzgebiet unterschiedliche Parameter und Konfigurationsmöglichkeiten. Diese Typen lassen sich grundlegend unterscheiden.

2.1 Einstellungen für Prozesse (Draft)

Jeder Prozess verfügt über verschiedene Einstellungen. Spezialisierte Prozesse können zusätzliche Einstellungen haben oder auch Einstellungen ausblenden. Im Folgenden sind allgemeine Prozesseinstellungen beschrieben.

2.1.1 Allgemein

Name

Der Name dient der Identifikation eines Prozesses.

Beschreibung

Die Beschreibung ist ein Zusatz zum Namen, den nur der Administrator angezeigt bekommt.

Kategorie

Mit dieser freien Eingabe eines Kategorienames kann die Anzeige im Syncler-Administrator strukturiert werden.

Laufende Nummer des Mandanten

Diese Nummer struktuiert die synchronisierten Daten in Mandanten. In Systemen wie Sage CRM, CAS oder Salesforce können die Sync-Statusfelder zu den Stammdaten nummeriert angegeben werden. Dies steuert, welche Prozesse auf den Datensatz zugreifen dürfen und damit auch, welches Ziel der Datensatz hat. Diese Nummer wird aber auch für Datenabbildungen verwendet. Ein Prozess sucht nur nach vorhandenen Datenabbildung für seine laufende Nummer und auch nur diese Datenabbildungen werden ggf. erzeugt oder aktualisiert. Durch die Verwendung von unterschiedlichen Nummern können Datensätze dadurch auch gezielt doppelt angelegt werden.

Siehe *Die laufende Nummer des Mandanten (Draft)*

2.1.2 Zurückschreiben

In einigen Prozess ist die Möglichkeit, Daten im Quellsystem zu aktualisieren, implementiert. In Prozessen mit reproduzierbaren Daten wird für die Aktualisierung der Quelldatensatz verwendet. In Prozessen mit nicht reproduzierbaren Daten, sogenannten Abfrage-Prozessen, kann das Schema und die ID-Felder oder ein auswertbarer Filter für die Aktualisierung angegeben werden.

Das Zurückschreiben unterscheidet 3 Fälle, bei erfolgreicher Verarbeitung, bei fehlerhafter Verarbeitung und bei Überspringen des Datensatzes.

2.2 Schema-basierte Prozesse (Draft)

2.3 Abfrage-basierte Prozesse (Draft)

Abfrage-Prozesse basieren nicht auf einem global bekannten Datenschema, welche durch Verbindungen bereitsgestellt werden. Ein Abfrage-Prozess erzeugt ein lokales Datenschema, das aus einer spezifischen Abfrage generiert wird.

2.3.1 Parameter

Prozess
: Dieser Parameter kann einen Synchronisationsprozess mit Datenabbildungen auswählen. Sobald ein Prozess festgelegt wurde, wird die Abfrage nicht global einmalig, sonder je Datenabbildung ausgeführt. Dabei können die Platzhalter SourceId und TargetId verwendet werden, um die Abfrage für die Datenabbildung anzupassen.

2.3.2 Abfrage und Ziel

2.3.3 Transformation

2.3.4 Änderungsspeicher

Da der Prozess eine Abfrage für die Gesamtheit der Daten definiert, ist er nicht in der Lage, gezielt einen einzelnen Datensatz nachzuladen. Sollte dies aber in Folge eines Fehlers oder einer notwendigen Wiederholung erforderlich werden, wird der Änderungsspeicher eingesetzt. Der Prozess legt dort für diesen Fall den Datensatz ab, damit er in einer späteren Verarbeitung wieder zur Verfügung steht.

Für dieses Verhalten steht ein Parameter zur Verfügung, der das Resultat beeinflusst. Wenn „Immer den Änderungsspeicher“ aktiviert wird, wird die Kopie direkt nach dem Lesen abgespeichert und steht später unverändert zur Verfügung. Das führt allerdings auch zu einem größeren Verbrauch an Datenbankspeicher und kann die Performance beeinflussen. Sollte der Parameter nicht aktiv sein, kann der Prozess nur auf eine ggf. transformierte Version des Datensatzes zugreifen und diese wird dann im Änderungsspeicher abgelegt. Bei einer späteren Ausführung ist der Prozess nicht mehr in der Lage, dies zu unterscheiden, weshalb die Transformation dieses Datensatzes erneut ausgeführt wird. Das muss bei der Planung der Transformation berücksichtigt werden. Ggf. sollten Felder vor der irreversiblen Veränderung kopiert werden. Es ist auch nicht empfohlen, die Transformation für diese Datensätze zu überspringen, da hier u.a. auch externe Systeme abgefragt werden, was zu einem anderen Resultat führen kann.

2.3.5 Datenabbildung

Falls das Ziel keine ID beim Speichern zurückliefern kann, wird ein Mapping ohne Ziel-ID angelegt. Sollte eine Übereinstimmungsregel definiert sein, wird dieser bei einer erneuten Übertragung wieder ausgeführt. Sollte keine Übereinstimmungsregel definiert sein, wird der Datensatz übersprungen.

Datenabbildungen werden angelegt, sobald eine Quell-ID festgelegt wurde.

2.4 Abfrage-Prozesse mit Massenverarbeitung (Draft)

2.5 Synchronisationsprozesse für geschachtelte Daten (Draft)

2.5.1 Universalprozess für geschachtelte Daten

Aktualisierung von Positionen

> Damit Positionen gezielt aktualisiert werden können, muss eine Übereinstimmungsregel definiert werden. Alle Zielpositionen, zu denen keine Quellposition ermittelt werden kann, werden gelöscht. Alle Quellpositionen, zu denen keine Zielposition ermittelt werden kann, werden neu angelegt. Alle Positionen mit einem Suchergebnis werden aktualisiert.

2.6 Ablauf-basierte Prozesse

Der Universal Ablauf Prozess bietet die Möglichkeit verschiedene Prozesse zu kombinieren und in einer definierten Reihenfolge ausführen zu lassen. Auch die gezielte Ausführung von Datendiensten und der Versand des Resultats können damit eingebunden werden.

Die Verkettung von Prozessen ist auch über Nachfolgeprozesse möglich, wobei die Kombinierbarkeit allerdings eingeschränkt ist. Für die Ausführung von zwei Prozessen ist der Nachfolgeprozesse meist aber einfacher zu konfigurieren.

Der Ablauf führt selbst keine Datenabfragen oder Transformationen durch. Dies wird durch separat definierte Prozesse umgesetzt, welche durch den Ablauf gesteuert werden.

Folgende Standard-Aufgaben werden durch den Ablauf übernommen.

- Bereitstellung der Zeitsteuerung per Warteschlange
- gesammelte Fehlerbehandlung der ausgeführten Prozesse
- Aktivierung einer initialen Ausführung aller zugeordneten Prozesse
- Unterstützung einer Adhoc-Verfügbarkeit

Der Kern des Ablaufs sind die Ablaufschritte. Diese definieren die spezifische Aufgabe und werden in der angegebenen Reihenfolge ausgeführt. Es stehen verschiedene Typen von Schritten zur Verfügung.

- Das Ausführen eines Prozesses
- Das Ausführen eines Datendienstes

Trotz das der Ablauf einen eigenen Warteschlangeneintrag hat, werden denoch manuelle Warteschlangeneinträge für die ausgeführten Prozesse angelegt. Dies stellt sicher, dass die Ausführung konsistent protokolliert werden kann und Standard-Mechanismen des Prozesses, wie die Fehlerbehandlung, möglich sind und individuell konfiguriert werden können.

Der Warteschlangeneintrag des Ablaufs bündelt dabei die einzelnen Ausführung mit Verweisen und einer summierten Statistik.

2.6.1 Adhoc-Ausführung

Am Ablauf kann über die Einstellungen „Quell-Verbindung“ und „Quell-Objekt“ der Ausgang eine Adhoc-Ausführung definiert werden. Der Ablauf ist dabei der Empfänger dieser Anforderung, aber nicht der ausführende Prozess. Bei der Abarbeitung der einzelnen Schritte werden Prozesse, die manuell ausgeführt werden sollen und eine identische Quell-Verbindung sowie Quell-Objekt haben, Adhoc ausgeführt. Alle weiteren Prozesse, die ebenfalls der Quell-Verbindung und dem Quell-Objekt zugeordnet sind, werden mit identischem Verhalten ausgeführt. Alle weiteren Prozesse werden gemäß der Konfiguration ausgeführt.

Damit ist es möglich eine Kette von Aufgaben oder mehrere Adhoc-Übertragungen ausführen zu lassen.

2.6.2 Ausführen eines Prozesses

Der auszuführende Prozess muss bereits vorhanden sein, damit er in einem Ablauf ausgewählt werden kann. Außer Abläufen selbst stehen alle vorhandenen Prozesse zur Verfügung.

Folgende Parameter können in diesem Schritt definiert werden.

Name

Dieser Name wird den einzelnen Meldungen des ausgeführten Prozesses vorangestellt.

Prozess ausführen

Hier wird der Prozess für die Ausführung festgelegt.

Daten aus Prozess

Die Kombination von Prozessen ermöglicht die Weiterverwendung von Quell- und Zieldaten der bereits ausgeführten Prozesse im Ablauf. Dafür stehen folgende Optionen zur Auswahl.

„Quelldaten unverändert übernehmen“ wählt die Quelldaten direkt nach dem Lesen und vor der Transformation aus. Sollte die Transformation bei einzelnen Datensätzen zu Fehlern führen, werden diese zurückgestellt. Auch mittels Zweitfilter ausgeschlossene Datensätze werden nicht weiterverarbeitet. Für Abfrage-basierte Prozesse setzt dies die Definition von ID-Feldern voraus.

„Quelldaten transformiert übernehmen“ wählt die Quelldaten nach der Transformation aus. Fehler und Zweitfilter schließen hier ebenfalls Datensätze wieder aus. Da das Zwischenspeichern der transformierten Daten nicht automatisch das Quell-Schema des folgenden Prozesses verändert, müssen dort ggf. Parameter mit leeren Werten angelegt werden, damit Nicht-Standard-Felder zur Verfügung stehen. Solange der Parameter keinen Wert hat, wird dem Datensatz auch nicht zugewiesen und die vorhandenen Daten können weiterverwendet werden. Zu beachten ist außerdem, dass bei geschachtelten Daten nur die Transformationen der Hauptobjektes übernommen werden können. Transformationen oder Filter an Positionen werden nicht zwischengespeichert.

„Keine Daten übernehmen“ überspringt diese Funktion. Da die Daten Prozess-bezogen zwischengespeichert werden, führt eine wiederholte Ausführen des selben Prozesses zum Überschreiben des vorangegangenen Ergebnisses. Mit dieser Option kann dies verhindert werden.

Zieldaten werden bei den ersten beiden Optionen ebenfalls zwischengespeichert. Es ist allerdings von den beteiligten Systemen abhängig, welche Daten hier zur Verfügung stehen. Die Verarbeitung kann z.B. entkoppelt erfolgen oder nur Basisinformationen zurückliefern.

Daten aus Vorgänger

Hier wird ein Prozess aus den vorangegangenen Schritten ausgewählt, dessen Quell- oder Zieldaten im aktuellen Prozess verwendet werden sollen. Im ersten Schritt des Ablauf steht deshalb nichts zur Verfügung. Sollte die Ausführungsart „Prozess manuell ausführen“ eingestellt sein, wird dieser Wert nicht benötigt.

Ausführungsart

Dies steuert die Art der Prozessausführung.

„Prozess manuell ausführen“ entspricht dabei einer normalen Prozessausführung. Diese kann noch mit weiteren Parametern beeinflusst werden. Der Adhoc-Start des Ablaufs würde dies mit Adhoc übersteuern.

„Prozess direkt mit Quelldaten aus Vorgänger ausführen“ startet den Prozess mit den zwischengespeicherten Daten. Der Prozess führt keine eigene Datenabfrage aus, mit Ausnahme von ggf. konfigurierten Fehlerwiederholungen oder Wiederholungen. Dieser werden eigenständig gelesen, da sie in den aktuellen Quelldaten nicht mehr vorhanden sind. Diese Option steht nur zur Verfügung, wenn beide Prozesse identische Quell-Verbindungen und Quell-Objekte haben.

„Prozess direkt mit Zieldaten aus Vorgänger ausführen“ startet den Prozess mit den zwischengespeicherten Daten. Der Prozess führt keine eigene Datenabfrage aus, mit Ausnahme von ggf. konfigurierten Fehlerwiederholungen oder Wiederholungen. Dieser werden eigenständig gelesen, da sie in den aktuellen Zieldaten nicht mehr vorhanden sind. Diese Option steht nur zur Verfügung, wenn beide Prozesse in Quell- und Zielverbindung, sowie Quell- und Zielobjekt aufeinander abgestimmt sind. Außerdem bleibt zu berücksichtigen, dass auch nicht jeder Prozess über vollständige Zieldaten verfügt.

„Prozess für jeden Quelldatensatz aus Vorgänger und einem ausgewerteten Filter ausführen“ führt für jeden Quelldatensatz eine modifizierte Datenabfrage aus und die gesammelten Daten werden verarbeitet. Diese Funktion setzt die Definition eines Filters mit Platzhaltern in ##-Schreibweise voraus. Auch können nicht alle Prozesse einen Filter auswerten. Dies ist meistens der Fall, wenn im Prozess auch kein Quellfilter definiert werden kann. Bei dieser Option müssen Verbindungen und Objekte nicht aufeinander abgestimmt sein.

„Prozess für jeden Zieldatensatz aus Vorgänger und einem ausgewerteten Filter ausführen“ führt für jeden Zieldatensatz eine modifizierte Datenabfrage aus und die gesammelten Daten werden verarbeitet. Diese Funktion setzt die Definition eines Filters mit Platzhaltern in ##-Schreibweise voraus. Auch können nicht alle Prozesse einen Filter auswerten. Dies ist meistens der Fall, wenn im Prozess auch kein Quellfilter definiert werden kann. Bei dieser Option müssen Verbindungen und Objekte nicht aufeinander abgestimmt sein.

Filter für Ausführung

Dieser Wert übersteuert den Filter des Prozesses in der aktuellen Ausführung, wenn „Prozess manuell ausführen“ eingestellt ist. Nicht alle Prozesse können einen Filter auswerten. Dies ist meistens der Fall, wenn im Prozess auch kein Quellfilter definiert werden kann. Außerdem definiert dies den Filter mit Platzhaltern für die Abfrage zu zwischengespeicherten Daten. Der Syntax muss den Anforderungen der jeweiligen Quell-Verbindung genügen. Sollte es sich um einen Abfrage-Prozess handelt, wird ggf. in der Abfrage der Platzhalter #FlowFilter# mit diesem Filter ersetzt. Bei der Verarbeitung von zwischengespeicherten Daten wird die Abfrage für jeden Datensatz wiederholt.

Bedingung für Ausführung

Diese logische Bedingung in SQL-Notation wird je zwischengespeicherten Datensatz ausgeführt. Dabei werden ##-Platzhalter mit den aktuellen Daten ersetzt. Falls die Bedingung nicht erfüllt wird, wird der Datensatz übersprungen.

Das Lesen der Quelldaten soll nicht mit dem Änderungsgrenzwert des Prozesses eingeschränkt werden

Dies entspricht der Ausführungsoption „Immer alle Datensätze abfragen“ eines Prozess und übersteuert diese Einstellung.

Ablauf beenden, wenn der aktuelle Prozess keine Daten liefert

Wenn der aktuelle Prozess keine Daten erhält, wird der Ablauf hier gestoppt.

Fehlerbehandlung

Generell kann eine Datensatz-bezogene Fehlerbehandlung am Prozess oder am Ablauf eingestellt werden. Mit dieser Einstellung kann der Ablauf gestoppt werden, falls der Prozess nicht erfolgreich beendet wird. Dies kann auch der Fall

sein, falls die Vebrindungsprüfung bereits fehlschlägt.

Die Optionen sind „Ablauf abbrechen“ und „Ablauf fortsetzen“.

Felder kopieren

Mittels Feldnotation können hiermit Daten aus dem Zielobjekt in das zwischengespeicherte Quellobjekt übertragen werden. Diese stehen dann in den folgenden Prozessen zur Verfügung und können z.B. mittels leerem Parameter nutzbar gemacht werden.

2.6.3 Ausführen eines Datendienstes

Datendienste sind ein Werkzeug für die Generierung eines tabellenorientierten Berichts. Diese könnten mit dem ERP-Fokus verwendet, zeitgesteuert gespeichert oder zeitgesteuert versendet werden. Dieser Ablaufschritt ermöglicht das Versenden des Berichts als Email-Anhang. Zusätzlich besteht die Möglichkeit den Emailinhalt festzulegen und zu personalisieren. Die Fehlerbehandlung dieses Typs erfolgt über den Änderungsspeicher, auf den im Warteschlangendatensatz des Ablaufs verwiesen wird. Je nach Einstellung werden dies Daten ausgelesen und für den Datendienst verwendet.

Folgende Parameter können in diesem Schritt definiert werden.

Name

Dieser Name wird den einzelnen Meldungen des ausgeführten Datendienstes vorangestellt. Außerdem wird er als Email-Betreff verwendet, wenn keiner definiert wurde.

Datendienst ausführen

Hier wird der Datendienst für die Ausführung festgelegt. Dieser wird nicht zum Speichern von Daten ausgeführt, sondern ausschließlich für den Versand eines Bericht, unabhängig von den Einstellungen des Datendienstes.

Daten aus Vorgänger

Hier wird ein Prozess aus den vorangegangenen Schritten ausgewählt, dessen Quell- oder Zieldaten im aktuellen Datendienst verwendet werden sollen. Sollte die Ausführungsart „Datendienst manuell ausführen“ eingestellt sein, wird dieser Wert nicht benötigt. Dann wird der Datendienst einmalig ausgeführt und versendet.

Ausführungsart

Dies steuert die Art der Datendienstausführung.

„Datendienst manuell ausführen“ entspricht dabei einer normalen Ausführung für den Versand eines Berichts. Die Abfrage des Datendienstes wird einmalig ausgeführt und das Resultat wird versendet.

„Datendienst für jeden Quelldatensatz aus Vorgänger und einer ausgewerteten Abfrage ausführen“ führt für jeden Quelldatensatz eine modifizierte Datenabfrage aus und die gesammelten Daten werden als Anhang versendet. Diese Funktion setzt die Definition eines Filters mit Platzhaltern in ##-Schreibweise voraus. Der Filter wird über den Platzhalter #FlowFilter# in die Datendienstabfrage eingefügt.

„Datendienst für jeden Zieldatensatz aus Vorgänger und einer ausgewerteten Abfrage ausführen“ führt für jeden Zieldatensatz eine modifizierte Datenabfrage aus und die gesammelten Daten werden als Anhang versendet. Diese Funktion setzt die Definition eines Filters mit Platzhaltern in ##-Schreibweise voraus. Der Filter wird über den Platzhalter #FlowFilter# in die Datendienstabfrage eingefügt.

Bedingung für Ausführung

Diese logische Bedingung in SQL-Notation wird je zwischengespeicherten Datensatz ausgeführt. Dabei werden ##-Platzhalter mit den aktuellen Daten ersetzt. Falls die Bedingung nicht erfüllt wird, wird der Datensatz übersprungen.

Filter für Ausführung

Dieser Wert wird für jeden Datensatz ausgewertet und über den Platzhalter #FlowFilter# in die Abfrage des Datendienstes eingefügt.

Ablauf beenden, wenn der aktuelle Datendienst keine Daten liefert

Wenn die Abfrage des aktuellen Datendienstes keine Daten erhält, wird der Ablauf hier gestoppt.

Empfängeradresse

Empfänger des generierten Berichts. Mehrfachnennungen werden mit Semikolon getrennt. #-Platzhalter werden für jeden Datensatz ersetzt. Dies gilt nicht für die manuelle Ausführung.

CC Adresse

Kopie-Empfänger des generierten Berichts. Mehrfachnennungen werden mit Semikolon getrennt. #-Platzhalter werden für jeden Datensatz ersetzt. Dies gilt nicht für die manuelle Ausführung.

BCC Adresse

Blindkopie-Empfänger des generierten Berichts. Mehrfachnennungen werden mit Semikolon getrennt. #-Platzhalter werden für jeden Datensatz ersetzt. Dies gilt nicht für die manuelle Ausführung.

Betreff

Betreff der generierten Email. #-Platzhalter werden für jeden Datensatz ausgewertet.

Vorlage für Nachrichten

Hier kann eine Html- oder Text-Vorlage aus den Serienbriefvorlagen ausgewählt werden. Je nach Format der Vorlage wird dann eine Html- oder Text-Email versendet. Bevor Sie diesen Wert auswählen können, muss die Vorlage über den Serienbriefbereich gespeichert werden.

Email Verbindung

Der Versand von Datendiensten kann nicht über den Syncler-Smtp erfolgen. Es muss eine Email Verbindung mit Smtp-Angaben eingerichtet werden. In der Email Verbindung wird auch der Absender für den Versand definiert. Mit diesem Wert wählen Sie die gewünschte Verbindung aus.

Fehlerbehandlung

Mit dieser Einstellung kann der Ablauf gestoppt werden, falls der Datendienst nicht erfolgreich ausgeführt und versendet wird. Die Optionen sind „Ablauf abbrechen“ und „Ablauf fortsetzen“.

2.6.4 Versand von E-Mails

Der Versand von Emails kann bereits mit Prozessen realisiert werden. Dieser Schritt vereinfacht das und bietet die Möglichkeit der einfachen Kombination. Der Versand setzt immer das Vorhandensein von zwischengespeicherten Daten voraus. Per Bedingung können einzelne Datensätze ausgeschlossen werden. Für die restlichen Datensätze wird der Versand, ähnlich den Datendiensten, durchgeführt. Die Fehlerbehandlung dieses Typs erfolgt über den Änderungsspeicher, auf den im Warteschlangendatensatz des Ablaufs verwiesen wird. Je nach Einstellung werden dies Daten ausgelesen und für den Datendienst verwendet.

Folgende Parameter können in diesem Schritt definiert werden.

Name

Dieser Name wird den einzelnen Meldungen des Schrittes vorangestellt. Außerdem wird er als Email-Betreff verwendet, wenn keiner definiert wurde.

Daten aus Vorgänger

Hier wird ein Prozess aus den vorangegangenen Schritten ausgewählt, dessen Quell- oder Zieldaten im aktuellen Versand verwendet werden sollen.

Ausführungsart

Dies steuert die Art des Versands.

„Versand mit Quelldaten aus Vorgänger ausführen“ startet den Versand mit den zwischengespeicherten Daten.

„Versand mit Zieldaten aus Vorgänger ausführen“ startet den Versand mit den zwischengespeicherten Daten.

Bedingung für Ausführung

Diese logische Bedingung in SQL-Notation wird je zwischengespeicherten Datensatz ausgeführt. Dabei werden ##-Platzhalter mit den aktuellen Daten ersetzt. Falls die Bedingung nicht erfüllt wird, wird der Datensatz übersprungen.

Empfängeradresse

Empfänger der generierten Email. Mehrfachnennungen werden mit Semikolon getrennt. #-Platzhalter werden für jeden Datensatz ersetzt.

CC Adresse

Kopie-Empfänger der generierten Email. Mehrfachnennungen werden mit Semikolon getrennt. #-Platzhalter werden für jeden Datensatz ersetzt.

BCC Adresse

Blindkopie-Empfänger der generierten Email. Mehrfachnennungen werden mit Semikolon getrennt. #-Platzhalter werden für jeden Datensatz ersetzt.

Betreff

Betreff der generierten Email. #-Platzhalter werden für jeden Datensatz ausgewertet.

Vorlage für Nachrichten

Hier kann eine Html- oder Text-Vorlage aus den Serienbriefvorlagen ausgewählt werden. Je nach Format der Vorlage wird dann eine Html- oder Text-Email versendet. Bevor Sie diesen Wert auswählen können, muss die Vorlage über den Serienbriefbereich gespeichert werden.

Email Verbindung

Der Versand von Emails kann nicht über den Syncler-Smtp erfolgen. Es muss eine Email Verbindung mit Smtp-Angaben eingerichtet werden. In der Email Verbindung wird auch der Absender für den Versand definiert. Mit diesem Wert wählen Sie die gewünschte Verbindung aus.

Fehlerbehandlung

Mit dieser Einstellung kann der Ablauf gestoppt werden, falls der Versand nicht erfolgreich ausgeführt und versendet wird. Die Optionen sind „Ablauf abbrechen“ und „Ablauf fortsetzen“.

2.6.5 Das Lesen von Quelldaten

Die meisten Prozess sind für die Verarbeitung von Quelldaten und das Schreiben von Zieldaten ausgerichtet. In einem Ablauf kann es aber auch die Aufgabe des reinen Lesens von Daten geben, da in den folgenden Schritten auf diesen Pool zugegriffen wird oder das Schreiben der Quelldaten nicht im ersten Schritt erfolgen soll. Für diesen Zweck kann ein Prozess definiert werden, der keine Feldzuordnungen enthält. Oder es wird der Prozess „Universal Ablauf - Daten lesen“ dafür eingerichtet. Dieser Prozess verfügt über Parameter für das Lesen von Daten und kann diese auch transformieren. Eine weitere Verarbeitung wird nicht ausgeführt.

Das könnte Sie interessieren. *Beispiele für den Einsatz von Abläufen*

2.7 Die Transformation

Prozesse, die der Datenübertragung dienen, verfügen über eine Transformation. Die Transformation definiert eine Liste von einzelnen Aktionen, die die Quelldaten verändern oder erweitern. Das Ziel der Transformation ist die Anpassung des Inhalt an die Anforderung des Ziels. Die Transformation kann auch funktional genutzt werden. Zum Beispiel kann die Transformation durch Abfragen Daten generieren, die für Filter oder Suchen genutzt werden können.

Die Transformation setzt sich aus verschiedenen Aktionen zusammen, die in einer festgelegten Reihenfolge ausgeführt werden.

Folgende Aktionstypen stehen zur Verfügung.

2.7.1 Json in Spalten

Diese Transformation erzeugt einzeln verwendbare Spalten auf der Basis von Json. Im Bereich Json-Vorlage wird ein Json-Objekt eingegeben, das als Vorlage für die Daten zur Laufzeit dient. Die Json-Vorlage kann mit der Schaltfläche „Spalten ermitteln“ ausgewertet werden, wobei auch geschachtelte Strukturen berücksichtigen Das Resultat ist eine Liste von Spalten, die unterhalb angezeigt wird.

Wenn keine Angaben zu den Json-Daten erfolgen, wird hiermit nur eine Reihe neuer Spalten ohne Wert erzeugt. Das ist z.B. innerhalb eines Ablaufs hilfreich, um die Daten eines lesenden Prozesses verfügbar zu machen.

Sobald unter Json-Daten eine Eingabe erfolgt ist, wird dieser Bereich als Json interpretiert und die Daten in die Spalten übernommen. Dabei können auch Platzhalter zu anderen Spalten verwendet werden, die Json als Zeichenfolge enthalten.

Einstellungen

Folgende Parameter werden konfiguriert.

Name

Ist eine administrative Bezeichnung.

Json-Vorlage

Definiert die Struktur der Json-Daten, aus denen die neuen Spalten erzeugt werden.

Json-Daten

Beinhaltet Laufzeitdaten, mit denen die neuen Spalten gefüllt werden. Es können Platzhalter mit anderen Feldern genutzt werden. Die Struktur muss der Vorlage entsprechen.

2.7.2 Webhook aufrufen

Diese Transformation ermöglich den Aufruf eines Webhooks. Der Einsatz kann für lesende oder schreibende Aktionen genutzt werden.

Basis ist eine Url, die für jeden Datensatz aufgerufen wird. Zusätzlich kann der Aufruf mit einem Header erweitert werden. Die Methode legt die HTTP-Methode des Aufrufes fest. Unter „Daten senden“ wird ein Json definiert, welches als Nachricht versendet wird. Diese Vorlage kann mit Platzhaltern aus dem aktuellen Datensatz gefüllt werden. Das Json sollte mit einfachen Hochkommas definiert werden, da diese automatisch in den Platzhaltern escaped werden. Bei der Verwendung von doppelten Anführungszeichen und doppelten Anführungszeichen im Datensatz kann es zu einem fehlerhaften Json kommen.

Die Ausführung kann Datensatz-spezifisch mit der Ausführungsbedingung übersprungen werden. Die logische Bedingung in SQL-Notation kann Platzhalter enthalten und wird für jedenDatensatz ausgewertet. Damit kann z.B. gesteuert werden, ob ein neuer Datensatz erzeugt werden muss.

Die Antwort des Webhooks wird in ein neues Feld gespeichert, dessen Name unter „Zielfeld für Antwort“ festgelegt wird. Sollte die Antwort ein Json-Objekt sein, kann der Inhalt mit der Transformation „Json in Spalten“ (*Json in Spalten*) ausgewertet werden.

Mit der Checkbox für das Zwischenspeichern kann festgelegt werden, ob Aufrufe mit identischem Inhalt wiederverwendet werden können. Das kann bei lesenden Webhooks mit identischem Resulat die Ausführungszeit verkürzen.

Einstellungen

Folgende Parameter werden konfiguriert.

Name

Ist eine administrative Bezeichnung.

Url

Url des Webhooks. Enthaltene Platzhalter werden ersetzt.

Optionaler Header

Name des zusätzlichen Headers.

Wert für Header

Wert des zusätzlichen Headers.

Methode

HTTP-Methode des Aufrufs.

Daten senden

Vorlage für den Versand der Json-Nachricht. Inhalte des Datensatzes werden mit Platzhaltern übernommen.

Ausführungsbedingung

Logische Bedingung in SQL-Notation. Inhalte des Datensatzes werden mit Platzhaltern übernommen. Ein negatives Resultat überspringt den aktuellen Datensatz.

Zielfeld für Antwort

Feldname für die Ablage der Antwort des Aufrufs. Das Feld wird ggf. neu hinzugefügt.

Zwischenspeichern gleicher Anfragen

Bei identischem Nachrichteninhalt werden Aufrufe nicht erneut ausgeführt, sondern die zwischengespeicherte Antwort verwendet.

2.7.3 Auswahllisten abbilden

Diese Transformation bildet Werte auf Texte oder Texte auf Werte ab. Die Basis dafür sind intern gespeicherte Auswahllisten, die durch bestimmte Verbindungen bereitgestellt werden. Es stehen auch Prozesse zur Verfügung, um eine Auswahlliste auf der Basis einer SQL-Abfrage zu erzeugen. Das Ziel der Transformation ist eine dynamische Abbildung von verschlüsselten Werten auf lesbare Texte und umgekehrt.

Für die Konfiguration muss eine Verbindung festgelegt werden. Als nächstes stehen alle Auswahllisten in dem gleichnamigen Auswahlfeld zur Verfügung. Die Richtung der Umwandlung muss angegeben werden. Sie können einen Wert in einen Text oder umgekehrt umwandeln.

Mit der Feldliste legen Sie fest, welche Quellfelder manipuliert werden sollen.

Außerdem enthält die Transformation eine Testfunktion, mit der die Ausführung kontrolliert und auch die Zielzuordnung überprüft werden kann.

/processes/converting/dryrun

Es gibt auch noch weitere Prozesse, die nicht zu diesen Typen passen.

2.8 Wartungsprozess

Der Wartungsprozess dient dazu, den Inhalt der Syncler Datenbank von alten Daten zu bereinigen. Es werden Warteschlangeneinträge, Protokolle, Sicherungen und Änderungsdatensätze abhängig vom Alter entfernt.

Bei der Wartung von Sicherungsdatensätzen gibt es eine Besonderheit. Der erste und letzte Eintrag zu einem Datensatz wird nicht automatisch gelöscht. Dies soll den initialen Zustand und die letzte Veränderung vorhalten. Vor allem der letzte Eintrag soll die Untersuchung von Fehlerquellen ermöglichen, was nicht abhängig vom Alter ist.

Wenn kein Wartungsprozess eingerichtet wird, führt Syncler eine tägliche Zwangswartung durch. Dabei wird das Standardalter von 14 Tagen angewendet.

Wenn ein abweichendes Alter gewünscht ist, muss ein Wartungsprozess eingerichtet werden. Falls dieser nicht in die Warteschlange eingereiht wird, wird er ebenfalls täglich automatisch ausgeführt.

Das Erhöhen des maximalen Alters führt zu einem größeren Bedarf an Datenbank-Speicherplatz. Da dieser limitiert ist, kann dies zur Unterbrechung der Synchronisation führen. In den Einstellungen kann die aktuelle und maximale Datenbankgröße eingesehen werden.

2.9 Microsoft Teams Benachrichtigung (Draft)

Für eine Synchronisation speichert der Prozess alle Einstellungen, Transformationen und Feldzuordnungen für eine Richtung zwischen zwei beteiligten Verbindungen und Objekten. Das Konzept sieht auch nicht genau einen Prozess für eine Übertragungsrichtung vor, sondern unterstützt eine beliebige Anzahl von Prozessen. Jeder Prozess wird dabei separat konfiguriert und ausgeführt und kann mit den anderen Prozessen indirekt über die Datenabbildungen interagieren, um z.B. ein Konfliktverhalten zu definieren. Dabei können diese Prozesse den Datenbestand separieren oder die Synchronisation ergänzen. Im Fall des Ergänzens werden zwei verknüpfte Datensätze mit mehreren Prozessen synchronisiert. Syncler übernimmt dabei die Orchestrierung der Prozesse.

Beispiel:

Es gibt Prozessvorlage für die Umsatz-Aktualisierung. In den meisten Systemen ist eine Umsatzänderung keine Änderung der Stammdaten und löst dadurch auch keine änderungsgetriebene Übertragung aus. Damit aktuelle Umsatzzahlen angeboten werden können, überträgt dieser Prozess ausschließlich die Umsatzzahlen für alle Datensätze und ergänzt damit die Änderungssynchronisation. Da das Schreiben der Umsatzzahlen eine Änderung im Zielsystem darstellt, aktualisieren Prozess die Synchronisationsinformationen von Datenabbildungen aus parallelen und komplementären Prozessen. Diese Interaktion führt dazu, dass diese Prozesse ggf. keine (Rück-) Übertragung ausführen oder es nicht zu falschen Konflikterkennungen kommt.

In folgenden Situationen ist eine Verteilung auf mehrere Prozesse sinnvoll.

Die bidirektionale Synchronisation ist asymmetrisch definiert

Wenn die Feldzuordnungen zwischen den Übertragungsrichtungen abweichen, kann ein Konflikt zu inkonsistenten Daten führen. Konflikte entstehen aus einer zeitgleichen Änderungen in verschiedenen Systemen und enden meist mit der Auswahl einer bestimmten Änderung und dem Verwerfen der anderen Änderung. Damit nur einseitig synchronisierte Felder durch einen Konflikt nicht verworfen werden, kann die Übertragung dieser Felder in einen weiteren Prozess mit einer eigenen Konfliktbehandlung ausgelagert werden.

Dies kann auch mit einem Prozess und dem Speichern des Datensatzes in der Datenabbildung zusammen mit dem Konfliktverhalten „Nur Quelländerungen übertragen“ realisiert werden. Dadurch werden aber auch mehr Daten in der Syncler Datenbank gespeichert, damit eine Änderungserkennung auf Feldebene möglich ist.

Sie benötigen in bestimmten Situationen unterschiedliche Filter

Wenn z.B. der Sage CRM Account-Manager mit dem Vertreter in Sage 100 synchronisiert werden soll, ist ein Kontokorrent erforderlich, da der Vertreter dort gespeichert wird. Wenn aber keine Kontenanlage durch die Integration gewünscht ist, benötigen Sie einen weiteren Prozess, der nur für Firmen mit bestehenden Kontokorrent dieses Feld zuordnet und überträgt. Hier bietet sich ein Filter auf eine vorhandene Kontonummer an.

Sie benötigen in bestimmten Situationen unterschiedliche Parameter

Jeder Prozess definiert eine Vielzahl von Parametern, die auch Datenobjekt-spezifisch sein können. Diese Parameter können während der Synchronisation nicht beeinflusst werden.

2.10 Universalprozess oder spezifischer Prozess

Es gibt spezifische Prozesse unter den Plugins, die die System- und Objekttypen genau festlegen. Diese Konstellation kann i.d.R. auch über den Universalprozess konfiguriert werden. Die spezifischen Prozesse führen aber auch zusätzliche Prüfungen aus oder speichern weitere Daten, damit die Konsistenz im Zielsystem sichergestellt ist. Sie sollten immer den spezifischen Prozess bevorzugt einrichten, damit es nicht zu inkonsistenten Zuständen kommen kann. Eine genaue Beschreibung, was der spezifische Prozess automatisch vorsieht, kann der jeweiligen Dokumentation entnommen werden.

Siehe *Integrationsszenarien*

Hier finden Sie eine Beschreibung zu den Universalprozessen.

2.10.1 Der Universalprozess (Draft)

2.10.2 Der Universalprozess für geschachtelte Daten (Draft)

2.10.3 Der Universal Lösch-Prozess (Draft)

KAPITEL 3

Datendienste (Draft)

Ein Datendienst ist die Definition eines Berichtes und wird dazu verwendet, Informationen aus einem entfernten System aufzubereiten. Datendienste können manuell durch eine Benutzer aufgerufen oder automatisch mit einer Warteschlange ausgeführt werden. Dabei können die Informationen im Kontext einer Synchronisierung stehen, oder auch komplett unabhängig von der Synchronisation sein. Die Ausführung eines Datendienstes kann das Resultat speichern oder per Email versenden.

KAPITEL 4

Allgemeine Elemente

Diese allgemeinen Elemente sind Bestandteile im Syncler für die Koordination, Überwachung und Ausführung.

4.1 Konfiguration

Über den Menüpunkt Einstellungen / Konfiguration erreichen Sie die Konfiguration Ihres Syncler-Accounts. Im Syncler Administrator ist das im Kontextmenü des obersten Navigationseintrages als „Eigenschaften“ enthalten.

Dort können Sie auf Einstellungen und Parameter zugreifen, die im Folgenden erläutert werden.

Administrator E-Mailadressen

Es können mehrere Emailadressen mit Semikolon getrennt hinterlegt werden. Sollten keine abweichenden Adressen hier oder am Prozess definiert sein, werden alle Email an diese Adressen versendet. Automatische Emails umfassen:

- Emails zu Fehlern bei Prozess- oder Datendienstausführung
- Tägliche Statusemail bei aktiver Prozessausführung
- Täglicher Warnungsbericht bei aktiver Prozessausführung. Diese wird nur versendet, wenn Fehler oder Warnungen in den vergangenen 24 Stunden protokolliert wurden.

Mandant ID

Die Mandant ID wird für den Zugriff auf die API für die Authentifizierung benötigt. Dort wird sie als Client ID bei der Anforderung eines Tokens angegeben.

Siehe *Syncler Web API und SDK*

Mandant Schlüssel

Der Mandanten-Schlüssel wird für den Zugriff auf die API für die Authentifizierung benötigt. Dort wird sie als Client Secret bei der Anforderung eines Tokens angegeben.

Siehe *Syncler Web API und SDK*

Abweichende E-Mailadressen für Fehlermeldungen

Alle Emails zu Fehlermeldung werden an diese Emailadressen versendet. Diese Emails werden beim Auftreten des Fehlers versendet.

Abweichende E-Mailadressen für Warnungen

Alle Emails zu Warnung werden an diese Emailadressen versendet. Es wird eine tägliche Zusammenfassung zu Warnungen versendet.

Abweichende E-Mailadressen für Statusberichte

Die tägliche Status-Email wird an diese Emailadressen versendet.

Absendername

Sie können einen individuellen Absendernamen für Emails von Syncler definieren.

Protokollierungsstufe für Prozessereignisse

Sie können eine Protokollstufe für die Prozessausführung angeben. Abhängig davon werden weniger oder mehr Protokolle zur Ausführung generiert und gespeichert. Davon ist auch die Nachrichtenausgabe am Prozess betroffen. Debug-Meldungen werden generell nur in der Nachrichtenausgabe dargestellt und nicht in den Protokollen gespeichert. Details zu den Protokollstufen finden Sie im Bereich „Protokolle“.

Protokollierungsstufe für Datendienstereignisse

Sie können eine Protokollstufe für die Datendienstausführung angeben. Abhängig davon werden weniger oder mehr Protokolle zur Ausführung generiert und gespeichert. Davon ist auch die Nachrichtenausgabe am Datendienst betroffen. Debug-Meldungen werden generell nur in der Nachrichtenausgabe dargestellt und nicht in den Protokollen gespeichert. Details zu den Protokollstufen finden Sie im Bereich „Protokolle“.

Sicherungsspeicher verwenden

Syncler kann vor jeder Änderung eines Zieldatensatzes den Zustand im Zielsystem sichern und auch wiederherstellen. Diese Backups werden in Ihrer Datenbank gespeichert und durch die Wartung werden auch ältere Einträge entfernt. Die Backup sind auch einsehbar, um ggf. Änderungen nachzuvollziehen. Diese Daten erhöhen den Speicherverbrauch der Datenbank.

Datensatzkopie in Protokollen speichern

Syncler kann zu jedem Protokolleintrag eine Kopie des Quelldatensatzes inkl. Transformationen speichern, sofern die Ausgabe einen Datensatz-Bezug hat. Diese Kopie kann über die Protoklle eingesehen werden, um ggf. Änderungen nachzuvollziehen. Diese Daten erhöhen den Speicherverbrauch der Datenbank.

Datensatzkopie in Datenabbildung speichern

Syncler kann zu jeder Datenabbildung eine Kopie des Quelldatensatzes inkl. Transformationen speichern. Diese Kopie wird automatisch bei jeder Synchronisation aktualisiert und stellt den letzten bekannten Quell-Zustand dar. Diese Daten können im Falle eines Konfliktes genutzt werden, um ausschließlich Änderungen in der Quelle und nicht Differenzen zwischen Quelle und Ziel zu übertragen. Dies ist eine Option der Konfliktbehandlung.

Datenbankgröße

Das ist Ihre aktuelle Datenbankgröße. Die Synchronisations- und eine ggf. vorhandene Hilfsdatenbanken werden summiert.

Siehe *Wartungsprozess*

Kontingent

Das ist Ihr aktuelles Transaktionskontingent. Als Transaktion wird das Lesen eine Datensatzes und das Schreiben eines Datensatzes gewertet. Sollten technologisch bedingt mehrere Lesevorgänge zu einem Datensatz erforderlich sein, wird dennoch nur eine Transaktion gezählt. Sollte das Schreiben nicht ausgeführt werden müssen, da keine Änderungen festgestellt wurden, wird keine Transaktion gezählt.

Seriendruck Kontingent

Das ist Ihr aktuelles Kontingent für die Erstellung von Serienbriefen. Hier ist auch die direkte Erstellung per API-Aufruf eingeschlossen. Es wird jede Ausführung gezählt, unabhängig von der Anzahl der Datensätze.

Maximale Datenbankgröße in MB

Sobald die maximale Datenbankgröße erreicht wird, kann kein Prozess ausgeführt werden. Es wird beim Versuch einen Prozess zu starten eine Fehlermeldung generiert und ggf. versendet.

Siehe *Wartungsprozess*

Sprache

Diese Sprache wird für die Darstellung der Oberfläche und für die Ausgabe von Nachrichten und Protokollen genutzt.

4.2 Warteschlangen (Draft)

Die Ausführung von Prozessen oder Datendiensten wird mit einem Aktionseintrag in einer Warteschlange gestartet. Es gibt jeweils eine Warteschlange für Prozesse und für Datendienste. Deren Abarbeitung kann separat über die Einstellungen gestartet oder gestoppt werden.

Warteschlangen werden kontinuierlich auf auzuführende Aktionen geprüft. Nach der Ausführung bilden die Einträge ein Protokoll der jeweiligen Ausführung und speichern die gesammelten Statusinformation. Für die Fehlerbehandlung werden historische Einträge auch genutzt, um fehlgeschlagene Verarbeitungen zu wiederholen.

Folgende Typen von Aktionen werden bei Prozessen unterschieden.

Manuell

Startet den Prozess einmalig mit den gleichen Parametern wie für eine zeitgesteuerte Ausführung. Nachfolgerprozesse werden gestartet, aber eine Fehlerbehandlung über einen weiteren Aktionsdatensatz ist nicht vorgesehen.

Einzelübertragung

Startet die gezielte Verarbeitung eines Datensatzes durch den Prozess. Eine Fehlerwiederholung wird abhängig von der Konfiguration mit weiteren Aktionsdatensätzen realisiert.

Geplant

Ist die zeitgesteuerte Ausführung eines Prozesses. Die Anlage des nächsten geplanten Aktionsdatensatzes wird vom Prozess durchgeführt. Dieser berücksichtigt dabei die Einstellung zur Fehlerbehandlung.

Nachfolger

Bilden eine Prozesskette, wo der 1. Prozess die Zielmenge des folgenden Prozesses einschränkt. Dabei wird auf den Aktionsdatensatz des Vorgängers zugegriffen.

Wiederholung

Stellt eine Fehlerbehandlung dar, bei der aufgetretene Fehler die Ausführung sofort wiederholen soll. Die zeitliche Differenz beträgt dabei immer drei Minuten.

Wiederherstellen

Ruft die Rollback-Funktion eines Prozesses zu einen Sicherungsdatensatz auf. Diese Funktion steht nicht für alle Prozesse zur Verfügung.

Ablauf

Dieser Typ wird durch einen Ablauf erzeugt, wenn der betreffende Prozess nicht selbst Daten abfragt, sondern diese aus dem Ablauf übernimmt.

Folgende Typen von Aktionen werden bei Datendiensten unterschieden.

Manuell

Startet den Datendienst einmalig mit den gleichen Parametern wie für eine zeitgesteuert Ausführung.

Einzelübertragung

Startet die Verarbeitung einer Datenabbildung durch den Datendienst.

Geplant

Ist die zeitgesteuerte Ausführung eines Datendienstes. Die Anlage des nächsten geplanten Aktionsdatensatzes wird vom Datendienst durchgeführt. Dieser berücksichtigt dabei die Einstellung zur Fehlerbehandlung.

Wiederholung

Stellt eine Fehlerbehandlung dar, bei der aufgetretene Fehler die Ausführung sofort wiederholen soll. Die zeitliche Differenz beträgt dabei immer drei Minuten.

4.3 Protokolle (Draft)

Verbindungen, Prozesse und Datendienste erzeugen Nachrichten, die in den Protokollen gespeichert werden. Jede Nachricht verfügt über eine Einstufung und kann noch zusätzliche Daten enthalten.

Folgende Stufen werden unterschieden.

Meldung

Sind nicht kategorisierbare Nachrichten des Synclers.

Fehler

Sind Ausnahmen, unter denen die aktuelle Aufgabe nicht beendet werden konnte.

Warnung

Kennzeichnen anormale Situationen, in denen die aktuelle Aufgabe dennoch beendet werden kann. Manche dieser Situationen benötigen Vorgaben für die gewünschte Reaktion. Zum Beispiel: Konfliktmanagement

Nachricht

Wird bei Informationen für den Administrator eingesetzt.

Rückmeldung

Sind Ausgaben zum aktuellem Stand der Aufgabenbearbeitung.

Debugging

Maximiert die Protokollierung bis hin zur kompletten Aufzeichnung von Inhalten der Kommunikation mit Endsystemen. Diese Stufe wird nicht gespeichert, sondern nur im Monitoring ausgegeben.

4.4 Datenabbildungen (Draft)

Für eine kontinuierliche Synchronisation von Datensätzen werden Datenabbildungen je Datensatz und Richtung angelegt. Die Abbildungen speichern ID-Werte, Synchronisationsinformationen und können auch Kopien des Quelldatensatzes für eine genaue Änderungsanalyse enthalten. Des weiteren werden Datenabbildungen für das Konfliktmanagement verwendet. Aus diesem Grund wird auch für jeden Prozess eine eigene Datenabbildung erzeugt, da die Prozesse keine direkte Beziehung untereinander haben. Die letzte Funktion ist die Verwaltung von Löschkennzeichen, die ein unerwünschtes erneutes Anlegen nach einer einseitigen Löschung vermeiden.

Der grundlegende Ablauf bei der Arbeit mit Datenabbildungen ist folgender. Ein Prozess wird ausgeführt und wählt seine Quelldaten aus, die im Anschluss noch transformiert werden. Danach beginnt die Bearbeitung der einzelnen Datensätze. Zu jedem Quelldatensatz wird eine vorhandene Datenabbildung für diesen Prozess gesucht. Wenn dabei

nichts gefunden wurde, wird die Suche auf parallele und komplementäre Prozesse ausgeweitet. Diese Beziehung wird durch die verwendeten Verbindungen und die laufende Mandantennummer gebildet. Sollte auch dies erfolglos sein, wird ggf. eine Übereinstimmungssuche ausgeführt. Wenn selbst diese kein Resultat liefert, geht der Syncler von einer Neuanlage aus und erzeugt ggf. eine neue Datenabbildung.

Sollte eine bestehende Abbildung für diesen Prozess gefunden werden, wird das Löschkennzeichen geprüft. Wenn dieses gesetzt ist, wird der Datensatz nicht weiter verarbeitet. Anderenfalls werden die Aktualisierungsinformationen, falls vorhanden, ausgewertet. Dabei wird zuerst die Quelle gegen die Aktualisierungsinformation der Quelle aus der Datenabbildung verglichen.

Falls diese identisch sind, wird davon ausgegangen, dass der Datensatz synchron ist und die Bearbeitung wird abgebrochen. Dieser Mechanismus verhindert ein „Ping-Pong“ der Prozesse, da eine Aktualisierung durch Prozess A als Auslöser für den komplementären Prozess B gilt. Als nächstes wird die Datenabbildung mit dem Ziel verglichen. Sollte aus einem der Vergleiche ein Konflikt entstehen, wird die Konfliktlösung angewendet. Sollte die Synchronisation durchgeführt werden, wird die eigene Datenabbildung aktualisiert. Abhängig von einer ggf. angewendeten Konfliktlösung werden auch parallele und komplementäre Datenabbildungen aktualisiert. Dies geschieht aber nur einseitig, damit diese Prozesse den Datensatz nicht als synchron behandeln und keinen Konflikt feststellen.

4.5 Änderungsspeicher (Draft)

Der Änderungsspeicher ist ein Zwischenspeicher für Datensätze, die noch verarbeitet oder weiterverarbeitet werden müssen. Abhängig von Prozess und Verbindung kommt der Änderungsspeicher auch in der Fehlerbehandlung oder verzögerten Übertragung zum Einsatz.

4.6 Datensatzsperren (Draft)

Datensatzsperren werden bei der Verarbeitung erzeugt und dienen der Konfliktbehandlung. Außerdem stellen sie die Aktualität von Daten sicher. Sobald ein Datensatz verarbeitet wird, wird der Quelldatensatz gesperrt. Sollte diese Sperre bereits existieren und keine oder neuere Änderungsinformtion enthalten, wird die Verarbeitung zurückgestellt. Wenn es sich um eine Aktualisierung handelt, wird auch das Ziel gesperrt. Sollte diese Sperre bereits existieren und keine oder neuere Änderungsinformtion enthalten, wird die Verarbeitung zurückgestellt. Nach Abschluss der Verarbeitung wird die Sperre mit den aktuellen Änderungsinformationen aktualisiert. Dies ermöglicht es anderen Prozessen zu prüfen, ob ihre bereits gelesenen Daten noch aktuell sind. Datensatzsperren bleiben über die Ausführung des Prozesses hinaus noch 60 Minuten gespeichert, damit die überkreuzte Ausführungen verschiedener Prozesse diese zur Verfügung hat.

4.7 Serienbrief Vorlagen (Draft)

Für die Seriendruckfunktion können Vorlagen in der Datenbank verwaltet werden. Durch einen Prozess oder die API kann bei Ausführung eines Seriendrucks diese Vorlage verwendet werden.

4.8 Eingehende Webhooks

In diesem Bereich können eingehende Webhooks konfiguriert werden, die dann über die API aufgerufen werden können. Dies bietet einen einfachen Weg für die Bereitstellung von Funktionen und erleichtert die Integration von externen Systemen.

Mit Webhooks können Daten gelesen und geschrieben werden. Außerdem kann eine Adhoc-Ausführung eines Prozesses gestartet werden. Die Konfiguration des Webhooks ermöglicht die Definition eines Quell- und Zielschema in JSON und ist dadurch optimal an die Bedürfnisse und Vorgaben externer Systeme anpassbar.

Webhooks können mit einem GET oder POST Aufruf angesprochen werden. Bei einem GET Aufruf werden die URL-Parameter für die Quelldaten der Ausführung verwendet. Der POST Aufruf erwartet entweder ein JSON Objekt im Nachrichten-Body oder Formulardaten. Aus Formulardaten wird ein Json-Objekt für die interne Verarbeitung erzeugt. Aus den Feldnamen werden dabei die Namen der Eigenschaften erzeugt.

Die komplexe Webhook ID, die für die URL verwendet wird, wird automatisch generiert und kann den Eigenschaften entnommen werden. Um die Sicherheit zu verbessern, kann zusätzlich ein Geheimschlüssel manuell angegeben werden. Dieser Geheimschlüssel muss als Header „syncler_webhooksecret“ im Aufruf übergeben werden. Wenn das Feld leer bleibt, wird diese Prüfung übersprungen.

Außerdem kann eine Ausführungsbedingung in SQL-Notation definiert werden, die einen Wahrheitswert generiert und ggf. die Aktion nicht ausführt. In der Bedingung können Quellfelder in Rauten eingeschlossen als Platzhalter verwendet werden.

4.8.1 Einstellungen

Folgende Parameter werden unter „Allgemein“ konfiguriert.

Name

Ist eine administrative Bezeichnung.

Ist aktiv

Nur aktive Webhooks können aufgerufen werden. Inaktive Webhooks liefern einen 410-Fehler beim Aufruf.

Webhook

Das ist die generierte Webhook ID. Das Feld ist schreibgeschützt und bei der Neuanlage noch leer.

Webhook Url

Mit dieser URL kann der Webhook aufgerufen werden. Die URL der API muss noch als Präfix ergänzt werden.

Geheimnis

Dieser Geheimschlüssel muss als Header „syncler_webhooksecret“ im Aufruf übergeben werden. Wenn das Feld leer bleibt, wird diese Prüfung übersprungen.

Ausführungsbedingung

Die in SQL formulierte Wahrheitsbedingung wird mit Quelldaten über Raute-Platzhalter gefüllt und geprüft. Ein negatives Ergebnis überspringt die Ausführung.

Aktion ausführen

Dieser Wert definiert die Funktion, die beim Aufruf ausgeführt werden soll. Eine Detailbeschreibung finden Sie unter „Aktionen“.

Prozess für Aktion

Mit diesem Wert wählen Sie den vorhandenen Prozess aus, der von der Aktion verwendet wird. Für das Starten kann jeder beliebige Prozess verwendet werden. Lesen und Schreiben mit einem Prozess erfordern die Verwendung eines Webhook Prozesses.

Verbindung für Aktion

Mit diesem Wert wählen Sie die vorhandene Verbindung aus, die für Lesen oder Schreiben verwendet werden soll. Zusätzlich muss ein Schema von dieser Verbindung festgelegt werden.

Schema-Objekt für Verbindung

Dieses Schema-Objekt stammt von der ausgewählten Verbindung und wird für das Lesen oder Schreiben mit einer Verbindung genutzt.

Weiterleitungs-Url nach dem Ausführen

Speziell für das Schreiben von Daten kann dieser Parameter genutzt werden, um einer Weiterleitungs-Url zu definieren, die dann nach der Ausführung als Weiterleitungsaufforderung zurückgemeldet wird. Damit ist es möglich, den Webhook als Empfänger eines Formulars zu definieren, welches nach dem Speichern eine LandingPage anzeigen soll.

Daten im Änderungsspeicher für den Prozess ablegen

Hier kann ein Prozess definiert werden, für den ein Datensatz im Änderungsspeicher abgestellt werden soll. Speziell hierfür gibt es den Prozesstyp „Universal Ablauf - Daten aus Änderungsspeicher lesen“. Dieser Prozesstyp dient zum Einlesen der Daten aus dem Änderungsspeicher für eine anschließende Verarbeitung in einem Ablauf. Sobald hier ein Prozess ausgewählt wird, wird mit den übergebenen Daten ein Änderungsdatensatz gespeichert. Das Objekt erhält dabei den Names des Quellobjektes aus dem ausgewählten Prozess. Sobald dieser Parameter definiert wird, erhält der Adhoc-Prozessstart nur die Guid des Änderungsdatensatzes und nicht mehr die Daten selbst.

Suche nach Quelldatensatz für Prozess und Verbindung

Diese Where-Clause wird mit Raute-Platzhaltern gefüllt und für das Lesen von Daten über einen Prozess oder eine Verbindung verwendet. Die Formulierung muss für das Zielsystem passend erfolgen (SQL oder OData usw.).

Sortierung

Sofern die Verbindung, die das Lesen direkt oder indirekt über den Prozess ausführt, das Sortieren des Ergebnisses ermöglicht, wird dieser optionales Wert dafür verwendet. Auch hier werden Raute-Platzhalter mit den Quelldaten gefüllt.

Nur den ersten Datensatz zurückgeben

Abhängig vom aufrufenden System kann die Antwort mit einem Array von Daten nicht gewünscht oder verwendbar sein. Für diesen Fall kann mit diesem Parameter nur der erste Datensatz als JSON Objekt zurückgegeben werden. Ansonsten erfolgt die Antwort immer als Array, sobald ein Resultat gefunden wurde.

Abfrage-Filter für Lesen einer Abfrage

Dieses Filterfragment ersetzt den Plathalter FlowFilter in der Abfrage des Prozesses. Dabei werden Platzhalter im Fragment mit den Daten des Aufrufs ersetzt.

Suche nach Zieldatensatz für Verbindung

Wenn das Schreiben mit einer Verbindung einen Datensatz aktualisieren soll, muss eine Suchanfrage formuliert werden. Diese Where-Clause wird mit Raute-Platzhaltern gefüllt und für das Suchen des Zieldatensatzes über eine Verbindung verwendet. Die Formulierung muss für das Zielsystem passend erfolgen (SQL oder OData usw.).

Verhalten bei keiner Übereinstimmung für Schreiben mit Verbindung

Sollte keine Suche formuliert sein oder diese kein Ergebnis geliefert haben, steuert dieser Parameter das weitere Vorgehen. Der aktuelle Aufruf kann übersprungen werden oder es kann ein neuer Datensatz angelegt werden.

Verhalten bei exakter Übereinstimmung für Schreiben mit Verbindung

Sollte eine Suche formuliert sein und diese exakte einen Zieldatensatz gefunden haben, steuert dieser Parameter das weitere Vorgehen. Der aktuelle Aufruf kann übersprungen werden oder der gefundene Datensatz kann aktualisiert werden.

Neben den allgemeinen Einstellungen stehen auch noch weitere Registerkarten zur Verfügung, die je nach Aktion konfiguriert werden müssen.

Für das Lesen von Daten mit einer Verbindung wird bereits im Webhook das Antwortschema definiert und die Felder aus dem Schemaobjekt den Feldern des Zielschema zugeordnet. Unter „Json-Schema Zieldaten“ definieren Sie ein JSON Objekt, dessen Eigenschaften als Zielfelder für die Zuordnung verwendet werden. Es können auch geschachtelte Strukturen definiert werden. Die Verwendung von Arrays ist durch die Feldzuordnung nicht möglich. Unter „Zuordnung Lesen“ verbinden Sie die Quellfelder der Verbindung mit den definierten Zielfeldern für die Antwort. Wenn das Lesen mit einem Prozess erfolgt, müssen die Registerkarten nicht konfiguriert werden.

Für das Schreiben von Daten mit einer Verbindung wird bereits im Webhook das Quellschema definiert und die Felder aus dem Schemaobjekt den Feldern des Quellschema zugeordnet. Unter „Json-Schema Quelldaten“ definieren Sie ein JSON Objekt, dessen Eigenschaften als Quellfelder für die Zuordnung verwendet werden. Es können auch geschachtelte Strukturen definiert werden. Die Verwendung von Arrays ist durch die Feldzuordnung nicht möglich. Unter „Zuordnung Schreiben“ verbinden Sie die definierten Quellfelder mit den Zielfeldern der Verbindung. Wenn das Schreiben mit einem Prozess erfolgt, müssen die Registerkarten nicht konfiguriert werden. Zusätzlich können auch Zieldaten und eine Abbildung dafür definiert werden. Dies ermöglich eine direkte Antwort mit dem gespeicherten Datensatz, um z.B. eine generierte ID zuerhalten.

4.8.2 Aktionen

Diese Aktionen können mit einem Webhook ausgelöst werden.

Starte Prozess

Hier wird ein Adhoc-Warteschlangendatensatz für den ausgewählten Prozess angelegt. Die Quelldaten werden dabei direkt als Ausführungsparameter dem Warteschlangendatensatz hinzugefügt. Die Ausführung erfolgt mit dem Daemon, dessen Prozessverarbeitung dafür aktiv sein muss.

Lese Daten mit Prozess

Hierfür muss ein Webhook Prozess zum Lesen von Daten konfiguriert werden. Die definierte Suche und Sortierung wird mit den Quelldaten gefüllt und über den Prozess direkt ausgeführt. Das Ergebnis wird durch den Prozess transformiert und zugeordnet. Die Antwort erfolgt mit einem JSON Array und dem am Prozess definierten Zielschema. Per Parameter kann die Antwort auf das erste Objekt des Ergebnisses beschränkt werden. In diesem Fall erfolgt die Antwort als JSON Objekt.

Lese Abfrage mit Prozess

Hierfür muss ein Webhook Prozess zum Lesen einer Abfrage konfiguriert werden. Der Abfrage-Filter wir mit den Quelldaten erzeugt und der Platzhalter in der Abfrage damit ersetzt. Dann wird die resultierende Abfrage direkt ausgegeben. Das Ergebnis wird durch den Prozess transformiert und zugeordnet. Die Antwort erfolgt mit einem JSON Array und dem am Prozess definierten Zielschema. Per Parameter kann die Antwort auf das erste Objekt des Ergebnisses beschränkt werden. In diesem Fall erfolgt die Antwort als JSON Objekt.

Speichere Daten mit Prozess

Hierfür muss ein Webhook Prozess zum Schreiben von Daten konfiguriert werden. Im Prozess kann eine Transformation und Feldzuordnung konfiguriert werden. Das Quellschema und eine Suche im Zielsystem wird ebenfalls im Prozess konfigiriert.

Lese Daten mit Verbindung

Hier werden direkt Daten aus einer Verbindung angefordert. Dafür muss hier die Suche, das Zielschema und die Feldzuordnung definiert werden. Eine Transformation des Ergebnisses ist nicht möglich.

Speichere Daten mit Verbindung

Hier werden direkt Daten mit einer Verbindung geschrieben. Das Zielschema wird aus den Schemaobjekten der Verbindung ausgewählt. Mit dem in JSON definierten Quellschema können die Felder direkt zugeordnet werden. Für die Suche eines vorhandenen Datensatzes stehen Parameter zur Verfügung. Eine Transformation der Quelldaten ist nicht möglich.

4.8.3 Der Webhook Prozess für das Lesen von Daten

Für das Lesen von Daten mit einem Prozess muss dieser spezielle Prozesstyp verwendet werden. Im Prozess wird die Quellverbindung und das Quellobjekt definiert. Außerdem stehen Registerkarten für die Definition des Zielschemas in JSON und die Feldzuordnungen zur Verfügung. Das gelesene Ergebnis aus der Quellverbindung durchläuft die Transformation und den zweiten Filter, bevor es an den Webhook übergeben wird.

4.8.4 Der Webhook Prozess für das Schreiben von Daten

Für das Schreiben von Daten mit einem Prozess muss dieser spezielle Prozesstyp verwendet werden. Im Prozess wird die Zielverbindung und das Zielobjekt definiert. Außerdem stehen Registerkarten für die Definition des Quellschemas in JSON und die Feldzuordnungen zur Verfügung. Eine Vorbedingung für die Neuanlage und die Übereinstimmungssuche für eine Aktualisierung wird im Prozess definiert. Die übergebenen Quelldaten durchlaufen die Transformation und den zweiten Filter, bevor das Zielobjekt geschrieben wird.

4.8.5 Webhook Prozess zum Lesen einer Abfrage

Für das Lesen einer Abfrage mit einem Prozess muss dieser spezielle Prozesstyp verwendet werden. Im Prozess wird die Quellverbindung und die Abfrage definiert. Dabei kann der Platzhalter FlowFilter mit einem durch den Webhook erzeugten Filter ersetzt werden. Außerdem stehen Registerkarten für die Definition des Zielschemas in JSON und die Feldzuordnungen zur Verfügung. Das gelesene Ergebnis aus der Quellverbindung durchläuft die Transformation und den zweiten Filter, bevor es an den Webhook übergeben wird.

4.8.6 Starten eines Ablaufs

Ein Universal-Ablauf kann auch für die Aktion „Starte Prozess“ genutzt werden, allerdings muss dabei eine bestimmte Konfiguration eingehalten werden. Abläufe führen keine direkte Datensatzverarbeitung aus, weshalb eine Übergabe an den Ablauf keine Ausführung erzeugt. Hierfür muss ein Lese-Prozess für das Lesen aus dem Änderungsspeicher eingerichtet werden. Für diesen Prozesstyp kann eine Quellverbindung und ein Quellobjekt definiert werden. Dies soll ermöglichen, dass die Daten innerhalb des Ablaufs auch an andere Prozesse übergeben werden können. Das Json-Schema für die Quelldaten am Prozess kann definiert werden, damit eine Transformation und ein zweiter Filter eingerichtet werden können. Die Übernahme in den Änderungsspeicher durch den Webhook erfordert aber auch ein Json-Schema für die Quelldaten. Dieser Lese-Prozess wird nun in den Ablauf eingefügt und liefert die Daten für die weitere Verarbeitung. Nach der erfolgreichen Verarbeitung wird der Datensatz im Änderungsspeicher gelöscht.

4.8.7 Formulare

Ein Webhook kann auch als Empfänger für ein Web-Formular genutzt werden. Dabei erkennt die API selbstständig am Content-Type, ob ein Json-Objekt in der Nachricht übergeben wurde, oder ob es sich um Formulardaten handelt. Aus dem Formulardaten wird ein Json-Objekt erzeugt, wobei die Feldnamen für die Eigenschaften verwendet werden. Da Formular vorrangig durch Benutzer verwendet werden, ist eine LandingPage nach dem Empfang der Daten notwendig. Die Url für diese Weiterleitung wird am Webhook definiert. Nachdem die Verarbeitung abgeschlossen wurde, wird an diese Url weitergeleitet. So kann der Webhook Daten eines Formlars annehmen, damit diverse Prozesse starten oder Daten direkt schreiben und im Anschluss eine Folgeseite aufrufen.

4.9 Support-Datenbank

Die Support-Datenbank ist eine zusätzliche Funktion zum Speichern und Aufbereiten von Daten. Es wird eine SQL-Datenbank bereitgestellt, die die Ausführung von SQL-Anweisungen zulässt und in der beliebig Tabellen angelegt werden können. Auch können Daten aus Excel- oder CSV-Dateien direkt in eine neue automatisch erstellte Tabelle importiert werden.

Die Support-Datenbank steht zur Verfügung, sobald das Feature für den Mandanten aktiviert wurde. Alle Berechtigungen bei der Benutzung sind auf diese Datenbank beschränkt. Zugriff auf die Synchronisationsdatenbank werden nicht gewährt.

Sobald das Feature aktiviert wurde, wird der verbrauchte Datenbankspeicher aus beiden Datenbanken summiert. Unter Umständen ist das Limit der Datenbankgröße dann anzupassen.

In der Admin-Oberfläche steht eine Übersicht zu den vorhandenen Tabellen zur Verfügung und die Inhalte der Tabellen können direkt bearbeitet, eingefügt oder gelöscht werden. Auch können ganze Tabellen angelegt oder direkt gelöscht werden.

Beim Anlegen einer Tabelle muss ein Name festgelegt werden. Danach können die gewünschten Spalten in einer Liste angegeben werden. Dabei kann auch eine Spalte als Auto-ID-Spalte definiert werden. Dies ermöglich einen optimalen Einsatz in Prozessen.

Es steht eine Import-Funktion für Excel- und CSV-Dateien zur Verfügung. Diese Import-Funktion erstellt automatisch eine neue Tabelle und legt Spalten und Inhalte an. Die erste Zeile in der Datei wird für die Spaltenbezeichnung verwendet. Über die zweite Zeile in der Datei wird der Datentyp für die erzeugte Spalte bestimmt. Besonders muss hier auf Ganz- und Kommazahlen geachtet werden. Eine Ganzzahl wird eine entsprechende Spalte in der Tabelle anlegen. Sollte in weiteren Zeilen dann Kommazahlen folgen, werden diese nicht importiert. Es wird automatisch eine ID-Spalte der Import-Tabelle hinzugefügt, damit diese optimal eingesetzt werden kann. Mit zusätzlichen Parametern kann erreicht werden, dass eine vorhandene Tabelle gleichen Names geleert oder ersetzt wird.

Eine nachträgliche Bearbeitung der Tabellenstruktur wird nicht unterstützt.

Es wird auch eine Export-Funktion bereitgestellt, mit der eine komplette Tabelle als Excel-Datei exportiert werden kann.

4.9.1 Die Support-Datenbank-Verbindung

Für den Einsatz der Support-Datenbank in Prozessen steht eine eigene Verbindung zur Verfügung. Für diese Verbindung kann optional ein geschachtelteter Datentyp definiert werden. Weitere Konfiguration sind nicht erforderlich. Die Schema-Objekte werden über die vorhandenen Tabellen erstellt.

Diese Verbindung kann wie eine SQL-Verbindung in Prozessen zum Lesen, Schreiben oder für Anweisungen verwendet werden.

KAPITEL 5

Integrationsszenarien

Für bestimmte Integrationsszenarien stehen spezialisierte Prozesse und Vorlagen zur Verfügung. Außerdem können noch Sonderfälle darin berücksichtigt sein. Hier finden Sie verschiedene Szenarien im Detail beschrieben.

5.1 Synchronisation zwischen Microsoft 365 und Zoho CRM

Die Microsoft Graph Verbindung stellt das Objekt „Event“ für den Zugriff auf Kalendereinträge zur Verfügung. Dabei handelt sich um ein geschachteltes Objekt, das die Teilnehmer des Termins als Positionsliste anbietet. Dieses Objekt kann z.B. in Universalprozessen für die Abfrage von Terminen bzw. auch geänderten Terminen genutzt werden. Ausgenommen von dieser Art des Zugriffs sind gelöschte Daten und Instanzen einer Serie. Für eine vollständige Synchronisation müssen aber auch diese Informationen ausgewertet werden. Deshalb stehen für die Synchronisation von Terminen eigene Prozesse und Vorlagen zur Verfügung.

5.1.1 Microsoft Graph Ereignis nach Zoho CRM Meeting

Dieser Prozess bildet die Richtung Microsoft 365 nach Zoho CRM ab. Für dieses Szenario steht eine Vorlage zur Verfügung, die neben Feldzuordnung auch Teilnehmer aus Benutzern, Kontakten oder Interessenten verbindet und deren Antwort überträgt. Wenn der Teilnehmer in diesen Bereichen nicht gefunden wird, kann er als Emailadresse übernommen werden. Für diesen Fall setzt Zoho eine gültige Emailadresse voraus, ansonsten wird das Speichern mit einer Fehlermeldung abgebrochen. Dies wird bereits in der Vorlage behandelt. Die Vorlage bildet den Organisator auf den Host des Meetings ab. Sollte dies nicht möglich sein, wird der Verbindungsadministrator automatisch von Zoho zugeordnet. In der Vorlage werden diese Termine aber über den zweiten Filter übersprungen. Da Zoho keine privaten Termine unterstützt, werden diese in der Vorlage bereits ausgefiltert. Die Daten werden von der Graph Verbindung mittels Delta-Funktion gelesen. Siehe *Microsoft Graph API*

Dabei werden alle Benutzer berücksichtigt, die durch den Benutzerfilter oder -liste in der Verbindung ausgewählt wurden. Delta-Datum und Delta-Token werden Prozess-bezogen intern gespeichert.

Serien werden mittels Seriendefinition in Zoho angelegt. Eine nachträgliche Änderung der Serie lässt Zoho nicht zu. Sollte eine Serie angelegt oder aktualisiert werden, werden alle Serientermine im Zoho gelöscht und die Datensätze neu angelegt. Im Anschluss findet ein Abgleich der Serienvorkommen statt, damit ggf. Lücken und Ausnahmen übernommen werden. Unbegrenzte Serien werden nicht unterstützt und übersprungen.

Eine Serie mit einer definierten Anzahl wird auf das Maximum von Zoho von 99 begrenzt, da die Serie sonst nicht angelegt werden kann.

Die Delta-Funktion synchronisiert nur ein definiertes Zeitfenster. Durch den Abgleich der Serientermine können aber auch Termine bzw. Vorkommen außerhalb des Zeitfensters synchronisiert werden.

Bei einer bidirektionalen Synchronisation ist zu beachten, dass anhand der Zoho CRM Daten nicht unterschieden werden kann, ob es sich um ein Vorkommen oder eine Ausnahme handelt. Deshalb werden alle Einträge aktualisiert, wodurch diese in Microsoft 365 automatisch zu Ausnahmen werden.

Besonderheiten

Abgebrochene Termine werden nicht als neue Termine übertragen. Der Prozess verfügt nur über einen sekundären Filter, da die Delta-Funktion nicht mit einem Filter kombiniert werden kann.

Ganztägige Termine haben als Endzeit in Microsoft 365 0 Uhr des Folgetages, wohingegen Zoho 23:59 Uhr voraussetzt.

Zoho unterstützt keine formatierten Inhalte für die Details. Deshalb wird der Html-Inhalt per Transformation in reinen Text konvertiert. Sollte durch eine Änderung eine Übertragung in der entgegengesetzten Richtung ausgelöst werden, wird der Html-Inhalt mit der Textdarstellung überschrieben.

5.1.2 Zoho CRM Meeting nach Microsoft Graph Ereignis

Dieser Prozess bildet die Richtung Zoho CRM nach Microsoft 365 ab und arbeitet auf der Basis des Änderungsdatums von Meetings. Da bei dieser Art der Abfrage keine gelöschten Datensätze erfasst werden, ist hierfür ein weiterer Prozess erforderlich, für den ebenfalls eine Vorlage bereitgestellt wird.

Der Owner des Termins wird in der Vorlage als Organisator zugeordnet. Deshalb ist es erforderlich, dass alle im Prozess berücksichtigten Benutzer auch Mitglied des Unternehmens sind, da sonst die Neuanlage nicht möglich ist. Bei Aktualisierungen wird der Graph Benutzer verwendet, über den der Termin abgerufen wurde.

Serien in Zoho CRM haben keinen direkt nutzbaren Master, sondern sind über $u_id verbunden. Der Serienmaster in Microsoft 365 erhält eine Datenabbildung zu dieser $u_id, damit bekannt ist, dass eine Serie bereits angelegt wurde. Dies wird auch in umgekehrter Richtung vorgenommen. Sollte diese Datenabbildung nicht vorliegen, wird eine neue Serie mit dem Muster aus Zoho angelegt. Nach der Anlage einer Serie werden die Vorkommen abgeglichen, damit Lücken und Ausnahmen übertragen werden.

Da Serien in Zoho nicht geändert werden können, entfällt das Löschen und Neuanlagen einer kompletten Serie.

Besonderheiten

Problematisch ist die Verarbeitung von ganztägigen Terminen. Diese werden von Zoho fälschlicherweise mit 0 Uhr UTC zurückgeliefert. Der Prozess korrigiert deshalb einen positiven Zeitzonen-Offset, damit der korrekte Tag zugeordnet wird. Negative Zeitzonen-Offsets müssen per Transformation ausgeglichen werden, was nicht Teil der Vorlage ist.

5.1.3 Zoho CRM gelöschte Meetings nach Microsoft Graph Ereignis

Diese Vorlage basiert auf einem Standardprozess, der gelöschte Datensätze in Zoho ermittelt und für Aktualisierung oder Löschen nutzen kann. Dabei werden die Ereignisse in Mircosoft 365 über vorhandene Datenabbildungen zwischen den eingestellten Verbindungen ermittelt. Diese werden gelöscht und auch die Datenabbildungen werden entfernt.

5.2 Synchronisation zwischen Microsoft 365 und Sage CRM

Die Microsoft Graph Verbindung stellt verschiedene Objekte für den Zugriff auf Microsoft 365 zur Verfügung. Dazu gehören z.B. Kalendereinträge (Event), Aufgaben (Todotask) und Kontakte (Contact).

Bei den Kalendereinträgen handelt sich um ein geschachteltes Objekt, das die Teilnehmer des Termins als Positionsliste anbietet. Dieses Objekt kann z.B. in Universalprozessen für die Abfrage von Terminen bzw. auch geänderten Terminen genutzt werden. Ausgenommen von dieser Art des Zugriffs sind gelöschte Daten und Instanzen einer Serie. Für eine vollständige Synchronisation müssen aber auch diese Informationen ausgewertet werden. Deshalb stehen für die Synchronisation von Terminen spezialisierte Prozesse und Vorlagen zur Verfügung.

Es stehen ebenfalls Prozesse und Vorlagen für die Objekte „todoTask“ und „Contact“ zur Verfügung. Diese ermöglichen die Synchronisation von Aufgaben und Kontakten. Die Synchronisation von Aufgaben beschränkt sich auf die Standardliste des Benutzers.

Für eine optimale Unterstützung sollte in Sage CRM in den Systemparametern die Verwendung einer Exchange-Integration auf „Ja“ gestellt werden. Dadurch wird der Organisator-Mechanismus für Termine und die Einschränkung der Serien-Schemata in der Oberfläche aktiviert. Eine weitere Einrichtung ist im CRM nicht erforderlich.

5.2.1 Microsoft Graph Ereignis nach Sage CRM Kommunikation

Dieser Prozess bildet die Richtung Microsoft 365 nach Sage CRM ab. Für dieses Szenario steht eine Vorlage zur Verfügung, die neben Feldzuordnung auch Teilnehmer aus Benutzern oder externe Teilnehmer aus Personen verbindet und deren Antwort überträgt. Die Vorlage bildet den Organisator auf den Organisator des Termins ab und ordnet dabei auch vorhandene Benutzer zu. Bei externen Organisatoren wird der Schreibschutz am Termin aktiviert (isstub), da keine bidirektionale Synchronisation möglich ist. Die Daten werden von der Graph Verbindung mittels Delta-Funktion gelesen. Siehe *Microsoft Graph API*

Dabei werden alle Benutzer berücksichtigt, die durch die Benutzerauswahl in der Verbindung festgelegt wurden. Delta-Datum und Delta-Token werden Prozess-bezogen gespeichert.

Serien werden mit Recurrence und RecuMaster übertragen. Der Serien-Master wird dabei mit dem nicht sichtbaren Recu-Master verbunden. Bei Neuanlage oder Aktualisierung einer Serie werden alle Vorkommen abgeglichen, damit ggf. Lücken und Ausnahmen übernommen werden. Eine Änderung der Serie löscht alle Vorkommen im CRM und erzeugt eine neue Serie. Eine Seriendefinition ohne Enddatum wird übersprungen, da dies vom CRM nicht unterstützt wird. Bei Serien mit einer definierten Anzahl wird das kleinste und größte Datum aus den Vorkommen für die Seriendefinition verwendet.

Ganztägige Termine haben als Endzeit in Microsoft 365 0 Uhr des Folgetages, wohingegen Sage CRM 23:59 Uhr voraussetzt. Eine Angabe von 0 Uhr wird von Sage CRM wie ein weiterer Tag interpretiert. Aus diesem Grund wird die Endzeit von ganztägigen Terminen, die auf 0 Uhr enden um 1 Minute zurückdatiert.

Besonderheiten

Abgebrochene Termine werden nicht als neue Termine übertragen. Der Prozess verfügt nur über einen sekundären Filter, da die Delta-Funktion nicht mit einem Filter kombiniert werden kann. Die Vorlage filtert private Termine aus. Sollte dies nicht gewünscht sein, muss der Filter angepasst werden. In diesem Fall sollte das Privat-Kennzeichen in Sage CRM gesetzt werden. Sage CRM unterstützt keine formatierten Inhalte für die Details. Deshalb wird der Html-Inhalt per Transformation in reinen Text konvertiert. Sollte durch eine Änderung eine Übertragung in der entgegengesetzten Richtung ausgelöst werden, wird der Html-Inhalt mit der Textdarstellung überschrieben.

5.2.2 Microsoft Graph Aufgabe nach Sage CRM Kommunikation

Für Aufgaben werden in Microsoft 365 keine Anwendungsberechtigungen unterstützt. Aus diesem Grund können Aufgaben nur für Benutzer synchronisiert werden, wenn diese dem Zugriff mit ihren Zugangsdaten zustimmen oder ein Administrator Refresh-Token in der Verbindung genutzt wird. Für die Zustimmung steht in Sage CRM unter Mein CRM eine Registerkarte zur Verfügung, worüber die Zustimmung erteilt oder entfernt werden kann.

Für die Synchronisation wird auch hier eine Delta-Abfrage verwendet, die neben Änderungen auch Lösungen erkennt und überträgt. Das Feld „isDeleted“ ist für diesen Fall vorhanden. Outlook-Aufgaben haben Datumsangaben ohne Zeitwerte. Wenn eine Aufgabe bidirektional übertragen wurde, wären dadurch die Zeitangaben in den Datumswerten auf 0 Uhr zurückgesetzt. Damit dies nicht passiert, wird bei Aktualisierungen die Uhrzeit der Sage CRM Aufgabe beibehalten und ggf. nur der Datumsanteil angepasst. Dieser Mechanismus wird auf Anfangs-, End- und Benachrichtigungszeit automatisch angewendet.

5.2.3 Sage CRM Kommunikation nach Microsoft Graph Ereignis

Dieser Prozess und die vorhandene Vorlage realisieren die Synchronisation von Sage CRM Terminen nach Microsoft 365. Der Prozess wendet keine hartkodierten Filter auf die Quelldaten an, sondern die Vorlage enthält die korrekten Angaben für eine optimale Unterstützung. Wenn Sie den Standardfilter der Vorlage reduzieren, riskieren Sie eine fehlerhafte Synchronisation. Serien werden nur über einen Serien-Master (RecuMaster) synchronisiert. Deshalb darf dieser spezielle Kommunikationsdatensatz nicht ausgeschlossen werden, wenn Serien übertragen werden sollen.

Da über die Schnittstellen von Sage CRM keine gelöschten Datensätze abgerufen werden können, steht für die Übertragung von Löschungen eigene Prozesse zur Verfügung, die konfigurierte SQL-Zugangsdaten in der Sage CRM Verbindung voraussetzen.

Sollte eine Serie in Sage CRM geändert werden, wird die Serie in Microsoft 365 gelöscht und neu angelegt. Sobald eine Serie angelegt wurde, werden die Vorkommen abgeglichen, damit ggf. Lücken und Ausnahmen übernommen werden.

Da ganztägige Termine in Sage CRM auf 23:59 Uhr enden und Microsft 365 Mitternacht voraussetzt, wird das Enddatum durch den Prozess um eine Minute erhöht.

Damit der Organisator nicht als Teilnehmer dem Termin hinzugefügt wird, wird dieser Kommunikation-Link-Eintrag per Vorlagenfilter übersprungen. Dafür steht extra das Feld Organisator in den Kommunikation-Link-Einträgen zur Verfügung.

5.2.4 Sage CRM Kommunikation nach Microsoft Graph Aufgabe

Dieser Prozess und die vorhandene Vorlage realisieren die Synchronisation von Sage CRM Aufgaben nach Microsoft 365. Auch hier werden keine hartkodierten Filter angewendet, sondern sind per Vorlage voreingestellt. Die Benutzerzuordnung erfolgt über das Feld „user“. Dafür wird die Emailadresse des Benutzers der Aufgabe abgefragt und zugeordnet. Dieser Benutzer muss eine Zustimmung für den Zugriff erteilt haben oder an der Verbindung als Zugangsdaten hinterlegt sein. Ansonsten wird das Speichern mit einer Fehlermeldung abgebrochen. Sollte nur ein Teil der Benutzer synchronisiert werden, empfiehlt sich ein Prozessfilter auf diese Benutzer. Der Prozess unterstützt auch das Verschieben von Aufgaben zwischen Benutzern. Sollte in Sage CRM der Benutzer an der Aufgabe geändert werden, wird die Aufgabe im ersten Postfach gelöscht und im zweiten ggf. neu angelegt. Dabei werden auch die vorhandenen Datenabbildungen entfernt und ggf. neu erzeugt. Damit ist sichergestellt, dass die Aufgabe nicht in Sage CRM gelöscht wird, wenn die Delta-Abfrage im ersten Postfach die Löschung ermittelt.

5.2.5 Sage CRM Kontakte nach Microsoft Graph Kontakt

Mit diesem Prozess und der vorhandenen Vorlage können Kontakte aus Sage CRM in die Outlook Kontakte synchronisiert werden. Dabei ist die Basis-Datenquelle die Kontaktzuordnung zwischen Benutzer und Person. Die Details werden dann per Transformation nachgeladen und reichern den Kontakt an. Die CRM Verbindung unterstützt die Änderungsabfrage zu Kontakten mit einer zusätzlichen Prüfung des Aktualisierungsdatum von Person, Adresse und Firma aus der Sicht vSearchListPerson. Damit die Übertragung dann nicht an der Synchron-Prüfung stoppt, werden diese Aktualisierungsdaten im Quellobjekt gesucht und das Maximum wird verwendet. Damit kann dieser Prozess den Kontakt aktualisieren, wenn sich eine der vier Datenobjekte ändert.

5.2.6 Sage CRM gelöschte Kommunikation nach Microsoft Graph Ereignis

Da über die Schnittstellen von Sage CRM nicht gezielt auf gelöschte Daten geprüft werden kann, sind diese Prozesse erforderlich. Sie führen eine Abfrage über die eingestellte SQL Verbindung aus und übertragen die Löschung und bereinigen die Datenabbildungen.

5.2.7 Sage CRM gelöschte Kontakte nach Microsoft Graph Kontakt

Da über die Schnittstellen von Sage CRM nicht gezielt auf gelöschte Daten geprüft werden kann, sind diese Prozesse erforderlich. Sie führen eine Abfrage über die eingestellte SQL Verbindung aus und übertragen die Löschung und bereinigen die Datenabbildungen.

5.2.8 Sage CRM gelöschte Kommunikation nach Microsoft Graph Aufgabe

Da über die Schnittstellen von Sage CRM nicht gezielt auf gelöschte Daten geprüft werden kann, sind diese Prozesse erforderlich. Sie führen eine Abfrage über die eingestellte SQL Verbindung aus und übertragen die Löschung und bereinigen die Datenabbildungen.

5.3 Sage Bäurer b7 Belege

Für die Objekte Angebot, Auftrag und Rechnung können zusätzliche Selektionskriterien definiert werden, um die Daten zu filtern. Die Angabe wird im Filterfeld des jeweiligen Prozesses in Feld-Notation eingegeben.

Beispiel: datvon|:|**2022-04-01T00:00:00+0100**|;|

5.3.1 Angebote

Folgende Kriterien sind für Angebote verfügbar.

kontoid
: eindeutige ID eines Kunden / Interessenten

konto
: Kontonummer eines Kunden / Interessenten, die in der Regel nur mit Angabe einer Satzart eindeutig ist

satzart
: Kontoart („1“ = Kunde, „4“ = Interessent)

offer_id
: eindeutige ID eines Angebots; entspricht dem DB-Feld „aufnr“ unter Berücksichtigung der für diesen UseCase fixen Vorgangsart für Angebot

datvon
: Die Angabe eines Zeitpunktes ermöglicht die Einschränkung auf „neuere“ Datenobjekte, d.h. Angebotsdatum ist gleich oder größer. Beispiel: „2019-01-01T00:00:00+0100“

nur_offen
: Mögliche Werte sind true oder false. Bei true werden nur offene Angebote abgestellt.

5.3.2 Aufträge

Folgende Kriterien sind für Aufträge verfügbar.

kontoid
: eindeutige ID eines Kunden

konto
: Kontonummer eines Kunden, die in der Regel nur mit Angabe einer Satzart eindeutig ist

satzart
: Kontoart („1“ = Kunde, „4“ = Interessent nicht möglich)

order_id
: eindeutige ID eines Auftrags; entspricht dem DB-Feld aufnr unter Berücksichtigung der für diesen UseCase fixen Vorgangsart für Auftrag

datvon
: Die Angabe eines Zeitpunktes ermöglicht die Einschränkung auf „neuere“ Datenobjekte, d.h. Auftragsdatum ist gleich oder größer. Beispiel: „2019-01-01T00:00:00+0100“

nur_offen
: Mögliche Werte sind true oder false. Bei true werden nur offene Aufträge abgestellt.

5.3.3 Rechnungen

Folgende Kriterien sind für Rechnungen verfügbar.

kontoid
: eindeutige ID eines Kunden

konto
: Kontonummer eines Kunden, die in der Regel nur mit Angabe einer Satzart eindeutig ist

satzart
: Kontoart („1“ = Kunde [„4“ = Interessent nicht möglich]

order_id
: eindeutige ID eines Auftrags; entspricht dem DB-Feld aufnr unter Berücksichtigung der für diesen UseCase fixen Vorgangsart für Auftrag

rg_nr
: Rechnungsnummer

rg_dat_von
: Die Angabe eines Zeitpunktes ermöglicht die Einschränkung auf „neuere“ Datenobjekte, d.h. Rechnungsdatum ist gleich oder größer. Beispiel: „2019-01-01T00:00:00+0100“

5.4 CSV Prozesse

Für CSV-Prozesse stehen Prozess-Parameter zur Konfiguration der Datei zur Verfügung.

Separator
Definiert das Trennzeichen der Felder.

Datei Kopfdaten
Dieser mehrzeilige Text wird als Präfix den CSV-Daten vorangestellt. Er kann z.B. für Steuerinformation für das endgültige Zielsystem genutzt werden.

Präfix Dateiname
Dieser Präfix wird dem Dateinamen vorangestellt. Automatisch wird der Dateiname mit der aktuellen Zeit (yyyy-MM-dd_HHmmss) und der Erweiterung .csv ergänzt.

Export-Feldliste
Die CSV-Erzeugung basiert in der Regel auf einer Abfrage. Falls nicht alle Felder des Resultats exportiert werden sollen, kann hier die gewünschte Feldliste definiert werden. Dieser Parameter kann außerdem die Reihenfolge der Spalten festlegen.

Zeichenkodierung
Diese Auswahl definiert die Zeichenkodierung der Resultat-Datei.

Datei ohne Kopfzeile
Dieser Parameter kann das Hinzufügen einer Kopfzeile mit Spaltennamen deaktivieren.

5.5 Microsoft Dynamics 365 CE - Sage 100

Für dieses Integrationsszenario stehen verschiedene Vorlage zur Verfügung, die den Universalprozess verwenden. Für die Synchronisation der Verkaufspreise für Artikel wird ein spezieller Prozess verwendet, der die Besonderheit der Preislistenhierarchie auswertet und verarbeitet.

5.5.1 Sage 100 Adresse nach Dynamics Firma

Diese Vorlage basiert auf dem Universalprozess und überträgt Adressen als Firmen. Es sind bestimmte Standardfelder in der Transformation und Zuordnung enthalten. Der Prozess filtert auf die Kategorie 0, was den Standard-Adressen entspricht. Auch Kontokorrent-Felder werden teilweise zugeordnet, sofern vorhanden. Zum Beispiel wird das Währungskennzeichen mit einer Abfrage auf die ID der Währung abgebildet.

5.5.2 Sage 100 Ansprechpartner nach Dynamics Kontakt

Diese Vorlage basiert auf dem Universalprozess und überträgt Ansprechpartner als Kontakte. Es sind bestimmte Standardfelder in der Transformation und Zuordnung enthalten.

Die Firmenzuordnung wird den Datenabbildungen des Adresse-Firmen-Prozesses entnommen. Deshalb muss bei der ersten Einrichtung diese Transformation bearbeitet und der korrekte Prozess eingestellt werden.

Für die Kundenzuordnung wird die Firma als Ziel verwendet. Außerdem werden Ansprechpartner wiederholt, sollte die Firma noch nicht übertragen worden sein.

5.5.3 Sage 100 Preisliste nach Dynamics Preisliste

Diese Vorlage basiert auf dem Universalprozess und überträgt Preislisten als Preisebene. Es sind bestimmte Standardfelder in der Transformation und Zuordnung enthalten. Der Prozess filtert auf den Typ 0, was Standardpreislisten entspricht. Diese Prozess wird für die Preisübertragung vorausgesetzt.

5.5.4 Sage 100 Artikel nach Dynamics Produkt

Diese Vorlage basiert auf dem Universalprozess und überträgt Artikel als Produkte. Es sind bestimmte Standardfelder in der Transformation und Zuordnung enthalten. Die Einheitengruppe und die Einheit werden ohne konkrete Einschränkung abgefragt, womit die Standarddatensätze zugeordnet werden. Diese Transformationen müssen im Filter angepasst werden, wenn andere Zuordnungen gewünscht sind. Diese Prozess wird für die Preisübertragung vorausgesetzt.

5.5.5 Sage 100 VK-Preis nach Dynamics Preis

Diese Vorlage basiert auf einem speziellen Prozess und überträgt Verkaufspreise als Preise. Es sind bestimmte Standardfelder in der Transformation und Zuordnung enthalten. Die Einheitengruppe und die Einheit werden ohne konkrete Einschränkung abgefragt, womit die Standarddatensätze zugeordnet werden. Diese Transformationen müssen im Filter angepasst werden, wenn andere Zuordnungen gewünscht sind. Der spezielle Prozess bildet die Hierarchie der Preislisten auf einzelne Preise ab. Dazu können die Preislisten noch mit einem eigenen Filter „Preislisten Filter“ eingeschränkt werden. Die Prozesse für Preislisten und Artikel sind die Voraussetzung für die Ausführung. Der Prozess sucht selbständig passende Datenabbildungen, um Ziel-IDs zu ermitteln.

5.5.6 Dynamics Firma nach Sage 100 Adresse

Diese Vorlage basiert auf dem Universalprozess und überträgt Firmen als Adressen. Es sind bestimmte Standardfelder in der Transformation und Zuordnung enthalten. Die Vorlage beinhalten noch keinen Quellfilter. Sollen nicht alle Firmen übertragen werden, muss ein Filter angegeben werden.

Siehe dazu auch *Übertragungen und Prozessfilter*

Der Matchcode wird aus Name und Stadt gebildet. Wenn dieses Format nicht gewünscht ist, kann die entsprechende Transformation angepasst werden.

5.5.7 Dynamics Kontakt nach Sage 100 Ansprechpartner

Diese Vorlage basiert auf dem Universalprozess und überträgt Kontakte als Ansprechpartner. Es sind bestimmte Standardfelder in der Transformation und Zuordnung enthalten.

Die Adresszuordnung wird den Datenabbildungen des Firmen-Adresse-Prozesses entnommen. Deshalb muss bei der ersten Einrichtung diese Transformation bearbeitet und der korrekte Prozess eingestellt werden.

Das Feld „Ansprechpartner“ wird per Transformation aus Anrede Vorname Nachname gebildet. Wenn dieses Format nicht gewünscht ist, kann die entsprechende Transformation angepasst werden.

Kontakte werden wiederholt, sollte die Adresse noch nicht übertragen worden sein.

5.6 CAS - Dynamics Business Central (Draft)

5.6.1 BC Kunden nach CAS Firma

Die Übertragung wird nur ausgeführt, wenn es bereits eine Datenabbildung zwischen dem Kontakt des Kunden und einer Firma gibt. Sollte keine direkte Datenabbildung existieren, wird eine neue Datenabbildung für den aktuellen Kunden und der Firma des Kontaktes angelegt. Deshalb steht keine Übereinstimmungsregel zur Verfügung.

5.6.2 CAS Firma nach BC Kunden

Bei der Neuanlage eines Kunden wird von der API automatisch ein neuer Kontakt angelegt. Da dies nicht steuerbar ist, kann der ggf. bereits vorhandene Kontakt der Firma nicht direkt zugeordnet werden. Aus diesem Grund wird der neue Kontakt direkt wieder gelöscht und eine Beziehung zwischen dem neuen Kunden und dem vorhandenen Kontakt hergestellt.

Die Übertragung wird nur ausgeführt, wenn es bereits eine Datenabbildung zu einem Kontakt gibt.

In Business Central ist nur ein Kunde je Kontakt zugelassen. Deshalb steht keine Übereinstimmungsregel zur Verfügung. Allerdings kann mit dem Verhalten für keine Übereinstimmung die Neuanlage gesteuert werden. Wenn Datensatz angelegt werden soll, können nur Änderungen übertragen werden. Sollte sich keine Datenabbildung finden oder ableiten lassen, wird nach Geschäftsbeziehungen gesucht. Eine gefundene Geschäftsbeziehung wird direkt für die Zuordnung verwendet. Ansonsten wird von einer Neuanlage ausgegangen.

Die Kundennummer wird nach erfolgreicher Übertragung in das Feld SIS_ACCOUNT# zurückgeschrieben und steht damit unmittelbar zur Verfügung.

5.6.3 CAS Ansprechpartner nach Dynamics BC Kontakt

5.7 SAP IDoc und Salesforce (Draft)

5.7.1 Saleforce Verkaufschance nach SAP Orders

Für dieses Szenario steht ein eigener Prozess zur Verfügung, der die Verkaufschancen-Positionen auf die Angebotspositionen abbilden kann. Hierbei steht eine spezielle Funktion zur Verfügung, die die Positionseinteilung des SAP Angebots erzeugt. Positionen können per Parameter über den Produkt-Code gruppiert werden. Die erste Produktzeile füllt dabei die Belegposition und alle Produktzeilen füllen die Einteilung der Belegposition. Dabei ist es außerdem möglich vorher Summen zu bilden, damit diese der Belegposition zugeordnet werden können. Dazu definiert man eine Liste von Feldern kommagetrennt in einem Parameter. Die Feldbezeichung folgt dabei der Pfadangabe mit Punkt getrennt. Die ermittelte Summe wird wieder der ersten Zeile zugeordnet, die für das Füllen der Belegposition verwendet wird.

5.8 Synchronisation zwischen Inxmail und CAS (Draft)

Übertragung von Listen mit identischem Namen nicht möglich: Ressource bereits vorhanden.

Anlage von Empfängern mit identischer Emailadresse überschreibt vorhandenen ohne Vorwarnung. Syncler gibt Fehlermeldung und Verweis aus. Weitere Korrektur erfolgt nicht.

> Die Basis im CAS wird dafür über vier Datensatztypen realisiert: - Adressen, welche die Empfänger innerhalb der Empfängerlisten umfasst - Verteilter, welcher im Inxmail die Empfängerliste darstellt - Newsletter, welche Mailings darstellen und an Empfängerlisten gesendet werden - Response, welche das Feedback der Empfänger aus diesem Mailing darstellt Innerhalb verschiedener Prozesse wird dann in beide Richtungen ein Informationsaustausch eingerichtet. Gängige Beispiele: - Verteiler können bidirektional synchronisiert werden, d.h. Bestände können jeweils aus beiden Systemen übernommen werden - Adressen, vorwiegend Ansprechpartner, welche in einem Verteiler enthalten sind, können per Übertragung des Verteilers in die Synchronisation aufgenommen und dauerhaft aktuell gehalten werden - Newsletter werden in Abhängigkeit zu den Verteilern und Adressen regelmäßig auch in CAS synchronisiert - Responses, welche daraus entstehen, werden ebenfalls aus Inxmail an des jeweilige Mailing im CAS übertragen - Anhand der im CAS dadurch vorhandenen Daten, können wiederum neue Verteiler mit Empfängern im CAS erstellt und dann an Inxmail übergeben werden. Auch Auswertungen und Darstellungen sind möglich: - Es können Kennzahlen über Empfänger, Abmeldungen oder weitere Zahlen am Mailing dargestellt werden - Es kann die komplette Sendungsstatistik aus Inxmail im CAS dargestellt werden - Eigene Kennzahlen, Reports und Diagramme können nach Wunsch auf Basis der Daten angezeigt und angelegt werden

5.9 Weitere Prozesse (Draft)

Neben gezielten Integrationsszenarien gibt es auch Prozesse, die universell eingesetzt werden können. Hier steht eine Liste und Beschreibung zur Verfügung.

5.9.1 CAS Abfrage

Dieser Prozess gehört zur Kategorie der Abfrage-Prozesse und führt eine SQL-Abfrage über die REST API von CAS aus. Das Ergebnis kann mit einer beliebigen Zielverbindung kombiniert werden.

5.9.2 CAS Abfrage Massenverarbeitung

Dieser Prozess gehört zur Kategorie der Abfrage-Prozesse und führt eine SQL-Abfrage über die REST API von CAS aus. Das Ergebnis kann mit jeder Zielverbindung, die eine Massenverarbeitung unterstützt, kombiniert werden.

5.9.3 CAS Abfrage nach CSV

Dieser Prozess gehört zur Kategorie der Abfrage-Prozesse und führt eine SQL-Abfrage über die REST API von CAS aus. Das Ergebnis kann mit einer CSV-Zielverbindung kombiniert werden. Es stehen CSV-Optionen für die Konfiguration der Datei zur Verfügung.

Siehe *CSV Prozesse*

5.9.4 CAS Abfrage nach Email Nachricht

Dieser Prozess gehört zur Kategorie der Abfrage-Prozesse und führt eine SQL-Abfrage über die REST API von CAS aus. Das Ergebnis wird je Datensatz mit einer Email versendet. Die Konfiguration der Email erfolgt über die Feldzuordnungen.

KAPITEL 6

Anleitungen und Lösungen

Für häufige Anwendungsfälle finden Sie hier Anleitungen, die die Einrichtung erleichtern sollen.

6.1 Verknüpfungen zwischen Objekttypen im Universalprozess

Wenn verschiedene Objekttypen zwischen System synchronisiert werden, müssen zwischen diesen unter Umständen auch Verknüpfungen übertragen werden. Das kann zum Beispiel die Beziehung zwischen Firma und Kontakt oder zwischen Kontakt und Verkaufschance sein. Die spezialisierten Prozesse leisten dies in der Regel automatisch und eine zusätzliche Konfiguration ist nicht erforderlich. Bei der Übertragung von unbekannten Objekttypen oder Konstellation mit dem Universalprozess muss dies aber in die Konfiguration aufgenommen werden.

Grundlegend können dabei zwei Ansätze unterschieden werden.

6.1.1 Verknüpfungen auf der Basis von Datenabbildungen

Wenn Objekttypen mit der Unterstützung von Datenabbildungen synchronisiert werden, steht bereits eine Abbildungsvorschrift von Quell-ID zu Ziel-ID zur Verfügung. Bei bekannten Objekttypen mit einem Schema ist das auch die Regel. Bei Prozessen für nicht reproduzierbaren Daten, sogenannte Abfrageprozesse, muss dies erst gezielt eingestellt werden.

Sobald eine Datenabbildung existiert, kann durch eine Transformation die ID für das Zielsystem ermittelt und als neues Feld bereitgestellt werden. Dieses Feld wird dann dem Zielfeld zugeordnet.

Für die Einrichtung gehen Sie wie folgt vor.

Wechseln Sie in den Bereich Transformation und fügen Sie dort eine neue Transformation „Datenabbildung abfragen“ hinzu. Jeder Prozess legt eine eigenen Datenabbildung an. Bei bidirektionalen Synchronisationen gibt es mindestens zwei Prozesse mit jeweils einer Datenabbildung. Dabei sind die ID-Werte vertauscht, da diese nur anonym als Quell- und Ziel-ID behandelt werden.

Wählen Sie im Feld Prozess einen Prozess aus, der den zu verknüpfenden Objekttyp überträgt. Nun können Sie festlegen, ob Sie die Quell- oder Ziel-ID des Prozesses für die Suche verwenden möchten. Im Suchfeld tragen Sie nun einen Platzhalter für ein Quellfeld ein. Dies sollte die ID des zu verknüpfenden Datensatzes aus dem Quellsystem

enthalten. Sie können nun noch einen Präfix für die abgefragten Daten definieren, damit diese besser lesbar sind. Es werden automatisch drei Felder den Quelldaten hinzugefügt. SourceId und TargetId enthalten die Quell- und Ziel-ID der Datenabbildung. TargetIsDeleted ist ein Wahrheitswert, der durch Synchronisation gesetzt wird, falls das Ziel nicht mehr auffindbar ist. Je nachdem welche Richtung der ausgewählte Prozess zwischen den Systemen beschreibt, enthält SourceId oder TargetId den Wert, den Sie dem Zielfeld zuordnen müssen.

Datenabbildungen enthalten nicht ausschließlich Feldwerte. Bei System mit mehreren ID-Feldern je Datensatz ist der Wert die Feldnotation der ID-Felder. Diese Transformation führt immer einen vollständigen Vergleich durch. In diesem Fall muss der Suchwert der Feldnotation entsprechen und direkt mit mehreren Platzhalten konstruiert werden.

Der Weg über Datenabbildungen sollte bevorzugt werden, sofern er angewendet werden kann. Es werden keine zusätzlichen Abfragen in den beteiligten Systemen ausgeführt und die Suche in der eigenen Datenbank ist auch performanter. Außerdem erhalten Sie ein eindeutiges Ergebnis.

6.1.2 Verknüpfungen auf der Basis von Suchen

Wenn keine Datenabbildungen zur Verfügung stehen, kann die passende Ziel-ID auch auf der Basis eine Suche im Zielsystem ermittelt werden. Dabei muss das Zielsystem aber auch entsprechende Suchen oder Abfragen unterstützen.

Es stehen zwei Transformationen für die Abfrage von Daten zur Verfügung. Die Transformation „Daten abfragen“ unterstützt eine Suche im aktuellem Quell- oder Zielsystem auf der Basis von Schemaobjekten. Die Transformation „Abfrage ausführen“ führt ein Abfrage-Statement mit einer beliebigen Verbindung aus. Diese Funktion ist vergleichbar mit einem Abfrageprozess, was vom jeweiligen System unterstützt werden muss.

Im Folgenden wird die Umsetzung mit der Transformation „Daten abfragen“ beschrieben. Diese Variante ist einfacher und wird von den meisten Verbindungen unterstützt. Sie müssen im Bereich „Verbindung“ festlegen, ob die Suche im Quell- oder Zielsystem erfolgen soll. In unserem Fall wollen wir eine ID für das Zielsystem ermitteln und suchen deshalb die Daten im Zielsystem. Als nächstes wird unter „Tabelle“ das Schema ausgewählt, mit dem gesucht werden soll. Im Bereich „Filter“ formulieren Sie eine für das System korrekte Suchbedingung. Das kann ein SQL- oder OData-Syntax sein. Manche System wie zum Beispiel Zoho verwenden auch einen eigenen Syntax (siehe SearchRecord). Da die Funktion nur einen Datensatz zurückliefert und die Suche unter Umständen mehrere Treffer generiert, kann mit „Sortieren nach“ noch eine Reihenfolge festgelegt werden, sofern das System dies unterstützt.

Nun wählen Sie die Ergebnisspalten aus, die mit dem festgelegten Präfix in die Quelldaten aufgenommen werden. Für eine Verknüpfung sollte hier das ID-Feld ausgewählt werden.

Diese Transformation ist hauptsächlich für die Anreicherung von Daten gedacht und für dieses Szenario unnötig aufwändig. Zusätzliche Abfragen erhöhen die Ausführungsdauer und produzieren ggf. auch API-Verbräuche. Außerdem besteht auch die Gefahr, dass die verwendete Suche keine eindeutigen Daten liefert. Dies kann noch optimiert werden, indem die Quell-ID des verwendeten Objekttyps ebenfalls im Zielsystem als Referenz gespeichert wird.

6.1.3 Abhängigkeiten in der Ausführung

Die beschriebenen Verfahren ermöglichen die Übertragung von Beziehungen sobald die entsprechenden Daten zur Verfügung stehen. Da Prozesse aber meistens zeitgesteuert und unabhängig ausgeführt werden, ist dies nicht automatisch gewährleistet.

Falls die Firma später als die Kontakte übertragen wird, kann diese nicht verknüpft werden und eine nachträgliche Verknüpfung ist nicht oder nur erschwert möglich. Deshalb sollten Sie für diesen Fall die Funktion der „Vorbedingung für Neuanlage“ einsetzen. Diese steht für die meisten Universal- und Abfrageprozesse zur Verfügung.

In diesem Feld wird eine logische Bedingung im SQL-Syntax formuliert, die Platzhalter für Quelldaten beinhalten kann und für jeden Datensatz einzeln ausgewertet wird. Diese Bedingung prüft das abgerufene Feld auf einen vorhandenen Wert.

```
'#GDM_TargetId#' <> ''
```

Sollte die Transformation keine Daten ermitteln können, ist das Feld leer und die Bedingung nicht erfüllt. In diesem Fall wird der Datensatz zurückgestellt und in der nächsten Ausführung wiederholt. Das funktioniert nur im Zusammenhang mit einer zeitgesteuerten Ausführung. Die Anzahl an Wiederholungen wird mit dem Parameter „Maximale Anzahl an fortlaufenden Wiederholungen“ eingestellt. Nach Erreichen dieses Limits wird der Datensatz nicht weiter wiederholt, da davon ausgegangen wird, dass die Bedingung nicht erfüllbar ist. Wählen Sie die Anzahl so, dass der betreffende Prozess den Datensatz für die Verknüpfung auch sicher übertragen kann. Durch unterschiedliche Laufzeiten oder Intervalle in der Prozessausführung ist eine einmalige Wiederholung schnell überschritten.

6.2 Formulierung von Bedingungen und Formeln

An verschiedenen Stellen in Prozessen und Transformationen können Bedingungen oder Formeln definiert werden. Der Syntaxt dafür ist an SQL angelehnt, weißt aber ein paar Besonderheiten auf.

Im Folgenden wird der Syntax und seine Funktion erläutert.

Eine formulierte Bedingung muss einen Wahrheitswert zurückliefern. Eine Formel kann aber auch andere Werte, wie Zahlen oder Zeichenketten generieren.

Folgende Vergleichsoperatoren können verwendet werden.

<

>

<=

>=

<>

=

IN (Auflistung)

LIKE

Folgende arithmetische Operatoren können verwendet werden.

+ (Addition)

- (Subtraktion)

* (Multiplikation)

/ (Division)

% (Modulo)

Für die Verbindung von Zeichenketten wird + verwendet.

Als Platzhalterzeichen in LIKE-Vergleichen wird * verwendet.

'#ItemName#' LIKE '*product*'

'#ItemName#' LIKE '*product'

'#ItemName#' LIKE 'product*'

Folgende Funktionen stehen zur Verfügung.

LEN('#ItemName#')

IIF('#ItemName#' <> '', 'nicht leer', 'leer')

TRIM('#ItemName#')

SUBSTRING('#ItemName#', start, length)

6.3 Email-Archivierung mit CAS

Die Verarbeitung von eingehenden Email gehört zu einer häufigen Aufgabe. In diesem Beispiel wird die Archivierung einer eingehenden Email in CAS beschrieben.

Ausgangspunkt für die Lösung sind eine Email-Verbindung für das Lesen von Nachrichten und eine CAS-Verbindung für das Speichern der Nachricht.

Für die Archivierung stehen zwei Wege zur Auswahl. Sie können eine Email als Email in CAS archivieren und verarbeiten. Außerdem können Sie ausschließlich die Anhänge einer Email einzeln archivieren und verarbeiten.

6.3.1 Archivierung als Email

Für die Archivierung als Email kommt der Prozess „Universal Sync Prozess für Email“ zum Einsatz. Dieser Prozess hat als mögliche Quelle die Email-Verbindung. Als Ziel können Sie jede vorhandene Verbindung und Objekt auswählen. In unserem Fall wählen wir die CAS-Verbindung aus und als Zielobjekt „EMAILSTORE“.

Die Archivierung der Email funktioniert auf der Basis von EML-Daten. Die Email-Verbindung stellt die EML-Daten im Feld „EML (FileContent)“ bereit. Unser Ziel verfügt ebenfalls über ein Feld „EML (FileContent)“. Stellen Sie eine Verbindung zwischen den Feldern her und die Nachricht wird komplett archiviert.

Zusätzlich können Sie jetzt noch Verknüpfungen zu anderen CAS-Daten herstellen. Nutzen Sie die Transformation, um z.B. Nummern aus dem Betreff per regulärem Ausdruck zu extrahieren. Damit können Sie Daten abfragen, um z.B. die GGUID einer Verkaufschance zu ermitteln. Erzeugen Sie noch einen Parameter für den OBJECTTYPE und weisen Sie beides den „LinkFrom“-Feldern zu.

Es stehen noch weitere Email-Optionen zur Verfügung, um eine verarbeitete Email zu löschen oder zu verschieben. Generell arbeitet der Prozess aber mit einem Änderungsdatum, womit nur neue Emails verarbeitet werden können. Außerdem erzeugt der Prozess Datenabbildungen, wodurch eine bereits abgelegte Email nicht erneut archiviert wird. Es können aber nachträglich noch Verknüpfungen zu archivierten Emails aktualisiert werden.

An der Beispielvorlage „Email-Archivierung in CAS“ können Sie Vorgehensweise nachvollziehen.

6.3.2 Archivierung als Dokument

Für die Archivierung als Dokument kommt der Prozess „Email Anhänge nach CAS Dokument“ zum Einsatz. Dieser Prozess verarbeitet ausschließlich die Anhänge eine Email. Sie können aber die Nachricht nutzen, um Felder und Verknüpfungen des Dokuments festzulegen. Diese Angaben werden für jeden Anhang wiederverwendet. Außerdem stehen Parameter zur Verfügung, um Mindestgröße und Dateityp zu kontrollieren.

6.4 Beispiele für den Einsatz von Abläufen

Hier ein paar Beispiele, wie mit Abläufen komplexere Anforderungen umgesetzt werden können.

6.4.1 Übertragung eines Beleges

Bevor ein Beleg übertragen werden kann, muss meistens auch ein Stammdatensatz bereits angelegt sein, welchem der Beleg zugeordnet wird. Übertragungsprozesse für den Beleg können so etwas prüfen und dann eine Fehlermeldung generieren. Jetzt muss der Benutzer eigenverantwortlich diese Meldung interpretieren, die Stammdaten übertragen und kann erst dann den Beleg erfolgreich übertragen.

Diese einzelnen Schritte sind nicht nur umständlich, sondern führen auch zu Fehlern, falls der Benutzer diesem nicht nachkommt.

Für dieses Szenario bietet sich ein Ablauf an, der sicherstellt, dass die Voraussetzungen gegeben sind. Mit folgenden Schritten kann dieser Ablauf gestaltet werden.

Beginnen wir mit den einzelnen Prozessen. Es wird ein oder mehrere Prozesse für die Stammdatenübertragung benötigt. Diese können direkt mit Vorlagen oder auch manuell konfiguriert werden. Natürlich können auch die Prozesses einer eingerichteten Synchronisation verwendet werden. Ein weiterer Prozess realisiert die Belegübertragung. Auch dieser kann aus einer Vorlage oder der Synchronisation stammen.

Und ein weiterer Prozess übernimmt ausschließlich das Lesen der Quelldaten des Beleges. Dieser ist üblicherweise nicht Teil einer Synchronisation und muss neu angelegt werden. Dafür kann z.B. der Prozess „Universal Ablauf - Geschachtelte Daten lesen“ verwendet werden. Stellen Sie die Verbindung und das Objekt ein. Weitere Transformationen sind an dieser Stelle nicht unbedingt erforderlich. Wichtig ist, dass Sie mit den generierten Daten in der Lage sind einen Filter für die Stammdaten zu formulieren.

Jetzt kann der Ablauf angelegt werden. Als Verbindung und Objekt stellen wir die Quelldaten ein, damit der Prozess auch für Adhoc-Übertragungen genutzt werden kann. Der erste Schritt führt den Prozess „Lesen der Quelldaten“ als manuellen Prozess aus. Eine Adhoc-Anforderung wird hiermit dann ebenfalls die Daten abrufen. Der zweite Schritt soll die Stammdaten übertragen. Wählen Sie den betroffenden Prozess aus. Als Ausführungsart wählen Sie „Prozess für jeden Quelldatensatz aus Vorgänger und einem ausgewerteten Filter ausführen“ und stellen als Vorgänger den Prozess aus Schritt 1 ein. Der Filter muss jetzt so formuliert werden, dass exakt die benötigten Daten erfasst werden. Hier ein Beispiel zu Sage CRM:

```
comp_companyid IN (SELECT oppo_primarycompanyid FROM Opportunity WHERE oppo_
↪opportunityid = #quot_opportunityid#)
```

An dieser Stelle kann ein weiterer Prozess erforderlich sein. Bestimmt Prozess für die Stammdaten-Synchronisation prüfen automatisch ein Status-Kennzeichen, um nicht alle Daten in das ERP-System zu übertragen. Für Sage CRM gilt dies zum Beispiel. Um dies zu umgehen brauchen wir einen weiteren Lese-Prozess „Universal Ablauf - Daten lesen“, der hier mit dem Filter ausgeführt wird. An diesem Prozess kann dieses Verhalten deaktiviert werden. Diesem folgt dann erst der eigentliche Übertragungsprozess, der direkt die Daten des Vorgängers verwenden kann, wodurch die Prüfung des Status-Kennzeichen nicht mehr stattfindet.

Der letzte Schritt führt dann die Belegübertragung aus und kann direkt die Quelldaten aus Schritt 1 verwenden.

6.4.2 Verarbeitung von Emails

In diesem Szenario wird ein Postfach kontinuierlich überwacht. Eingehende Emails sollen ein Ticket im Zielsystem anlegen und die Email dazu abspeichern. Außerdem sollen Emails auch bestehenden Tickets zugeordnet werden, sofern eine Übereinstimmung mittels einer Referenz gefunden wird. Die Umsetzung kann abhängig vom Zielsystem verschieden komplex ausfallen. Manche Systeme können Email gesamt abspeichern und in anderen Systemen muss das Speichern der Anhänge separat erfolgen. Dieses Beispiel beleuchtet den komplexeren Fall am Beispiel von Sage CRM. Zu den einzelnen Prozessen stehen auch Vorlagen zur Verfügung, die in einem Ablauf kombiniert werden können.

Für die Umsetzung werden zuerst die zwei Verbindungen zu Email-Postfächern und Sage CRM benötigt. Folgende Prozesse müssen eingerichtet werden.

1. **Emails abrufen und aufbereiten**
   Dieser Prozess ist vom Typ „Universal Ablauf - Geschachtelte Daten lesen“ und verwendet die Email-Verbindung und das Objekt MimeMessage. Mit Transformationen wird die Person und Firma im CRM ermittelt. Ausschließlich dafür wird diesem Prozess das Ziel Sage CRM zugeordnet. Es ist wichtig, diesen Typ zu verwenden, damit die Anhangliste durch die Transformation nicht entfernt wird.

2. **Email als Ticket anlegen oder zuordnen**
   Dieser Prozess ist vom Typ „Universal Sync Prozess für Email“ und verwendet die Email-Verbindung. Als Ziel wird Sage CRM und cases eingestellt. Der Prozess verändert nichts am Postfach, damit ggf. eine Fehlerwiederholung möglich ist. Per Transformation wird die Referenz aufbereitet und in der Übereinstimmungsregel vorhandene Tickets damit gesucht. Falls bereits per Email ein Ticket angelegt wurde, kann eine Datenabbildung vorhanden sein. Das besondere an diesem Prozess ist, dass das nicht zum Überspringen führt, sondern die Verarbeitung fortgesetzt werden kann. Neue Emails können also das Ticket auch aktualisieren, aber zur identischen Email wird kein neues Ticket angelegt, falls dort der Referenz-Ansatz nicht möglich ist.

3. **Email als Kommunikation ablegen**
   Dieser Prozess ist vom Typ „Universal Sync Prozess für Email“ und verwendet die Email-Verbindung. Als Ziel wird Sage CRM und communication eingestellt. Auch dieser Prozess verändert nichts am Postfach, damit ggf. eine Fehlerwiederholung möglich ist. Der Prozess verwendet die transformierten Daten von Prozess 1 und kann so direkt Firma und Person im Unterobjekt comm_link zuordnen. Als Besonderheit wird hier die Transformation „Datenabbildung abfragen“ verwendet. Zusammen mit der Uid der Email kann so die Ticket-ID ermittelt und der Kommunikation zugeordnet werden.

4. **Email Anhänge als Dokumente ablegen**
   Als letzter Prozess wird das Speichern von Anlagen mit dem Typ „Email Anhänge nach Sage CRM Dokument“ eingerichtet. Wie im Prozess 3 wird mit der Transformation „Datenabbildung abfragen“ die Ticket-ID ermittelt. Dies wird auch für die Kommunikation durchgeführt, womit die Datei mit Ticket und Kommunikation, sowie Firma und Person verknüpft werden kann. Als letzter Prozess in unserem Ablauf kann dieser nun das Postfach verändern und die Email löschen oder verschieben. In diesem Prozess muss der Dateiinhalt (FileContent) und Dateiname nicht zugeordnet werden. Dies wird automatisch durch den Prozess durchgeführt.

Nun kommen wir zum Ablauf. Dieser nimmt die 4 Prozesse in dieser Reihenfolge auf. Prozess 1 speichert transformierte Quelldaten zwischen. Die Prozesse 2 bis 4 verwenden die transformierten Quelldaten von Prozess 1 und müssen selbst keine Daten zwischenspeichern.

6.4.3 Verarbeitung von CSV-Dateien

In diesem Beispiel gehen wir von einer CSV-Datei aus, die Daten für unterschiedliche Zielobjekte enthält. Die CSV-Datei könnte z.B. aus einer Lead-Quelle oder Messe-Erfassung stammen. Die Daten sollen eine Firma, einen Ansprechpartner und eine Verkaufschance erzeugen. Diese Aufgabe kann auch mit Nachfolgeprozessen realisiert werden, wodurch sich aber einige Nachteile ergeben. Sollte z.B. ein Prozess in der Kette nicht erfolgreich ausgeführt werden, wird der Nachfolgeprozess nicht gestartet, um einen Datenverlust vorzubeugen. Dies würde also die komplette Kette stoppen und die Verarbeitung von erfolgreichen Datensätzen kann nicht fortgeführt werden. Außerdem liest jeder Prozess die Daten aus der Datei oder dem Änderungsspeicher erneut ein und je nachdem muss auch der richtige Prozess für die Archivierung bzw. das Löschen der Quelldatei ausgewählt werden.

Um diese Schritte zu vereinfachen, definieren wir in diesem Beispiel einen Ablauf. Dieser kann dann z.B. eine gesamthafte Fehlerbehandlung durchführen und verarbeitet erfolgreiche Datensätze bis zum Schluss.

Es wird ein Prozess „Universal Ablauf - Daten aus CSV lesen“ für das Lesen der CSV angelegt. Als nächstes werden einzelne Prozesse für CSV nach Firma, nach Person und nach Verkaufschance angelegt. Im Prozess für die Person definieren wir einen leeren Parameter „CompanyID“. Im Prozess für die Verkaufschance definieren wir die leeren

Parameter „CompanyID“ und „PersonID“. Jetzt legen wir den Ablauf an und fügen alle Prozesse der Reihenfolge nach ein. Die Prozesse für Firma, Person und Verkaufschance verwendet die Quelldaten des CSV-Lesen Prozesses. Für den Prozess Firma wird eine Kopierregel für Felder hinzugefügt: CompanyID|:**|#Ziel-ID#|**;| Für den Prozess Person wird eine Kopierregel für Felder hinzugefügt: PersonID|:**|#Ziel-ID#|**;|

Durch die Kopierregeln stehen die ggf. erzeugten oder gefundenen IDs aus den einzelnen Prozessen im Folgeschritt zur Verfügung. Die Person kann damit der Firma und die Verkaufschance der Firma und Person zugeordnet werden.

6.5 Zoho CRM - Kontakt-Rollen

Die Zoho CRM-Verbindung stellt das Schemaobjekt „Contact_Roles“ zur Verfügung. Hierbei handelt es sich nicht um ein Modul-Objekt, sondern um ein Funktionsobjekt zur Verwaltung von Kontakt-Rollen eines Abschlusses.

Jeder Abschluss kann eine beliebige Liste von Kontakt-Rollen besitzen. Dies sind 1:n Verknüpfungen zu Kontakten, die zusätzlich noch mit einer Rolle charakterisiert werden können.

Die Kontakt-Rolle besitzt selbst keine ID, weshalb das Schemaobjekt eine zusammengesetzte ID aus Abschluss-ID (deal_id) und Kontakt-ID (contact_id) bereitstellt. Auch eine Änderungsinformation steht hier nicht zur Verfügung.

Bei einer Abfrage von Kontakt-Rollen antwortet die Zoho CRM-API mit dem Kontaktdatensatz und fügt die Rolle als zusätzliches Feld hinzu. Damit diese Daten auch genutzt werden können, beinhaltet das Schemaobjekt alle Kontaktfelder schreibgeschützt und zusätzlich noch die ID-Felder für Abschluss und Kontakt, sowie den Rollennamen zum Speichern oder Löschen einer Kontakt-Rolle.

Die Antwort liefert die ID der Rolle und nicht den Namen, wohingegen das Anlegen oder Bearbeiten nur den Namen der Rolle akzeptiert. Deshalb stellt das Schemaobjekt beide Felder für ID (contact_role_id) und Name (contact_role_name) bereit und reichert die Daten auch entsprechend an.

Die Zoho CRM-Verbindung unterstützt das Lesen von Kontakt-Rollen (Liste) eines Abschlusses oder das einzelne Lesen einer Kontakt-Rolle zwischen einem Abschluss und einem Kontakt. Für den Aufruf erwartet die Verbindung die Parameter in Feldnotation. Dies kann als Prozessfilter (z.B. auch kombiniert in Abläufen), als Adhoc-Ausführung oder in einer Übereinstimmungsregel erfolgen.

> deal_id|:**|1234|**;| liest alle Kontaktrollen des Abschlusses mit der ID 1234
>
> deal_id|:1234|;|contact_id|:**|5678|**;| liest die Kontaktrolle zwischen dem Abschluss mit der ID 1234 und dem Kontakt mit der ID 5678

Für das Anlegen oder Aktualisieren einer Kontaktrolle müssen die ID des Abschlusses (deal_id) und die ID des Kontaktes (contact_id) zugeordnet werden. Im Fall einer Aktualsierung kann dies bereits durch eine Datenabbildung oder eine Übereinstimmungssuche erfolgt sein. Wenn die Übereinstimmungssuche allerdings kein Ergebnis liefert, müssen die ID-Werte als Felder zugeordnet sein. Die Angabe der Rolle ist optional und muss mit dem Rollennamen (contact_role_name) erfolgen.

Das Löschen von Kontakt-Rollen kann mit dem Universal-Löschprozess oder einem Abfrageprozess erfolgen. Diese arbeiten auf der Basis einer Datenabbildung oder einer Übereinstimmungssuche und erhalten darüber die ID-Werte für das gezielte Löschen der Kontakt-Rolle.

6.6 Übertragungen und Prozessfilter

Wenn Felder zwischen System übertragen werden, die gleichzeitig den Filter für die Übertragung beeinflussen, muss die Konfiguration geplant werden. Wenn zum Beispiel nur aktive Firmen aus dem CRM in das ERP übertragen werden sollen, aber der Status „inaktiv“ synchronisiert werden muss, blockieren sich diese Anforderung.

Der Prozessfilter wird eine Firma, die auf „inaktiv“ geändert wird, komplett ausschließen und damit kann dieser Status nicht übertragen werden. Auf der anderen Seite sollen aber inaktive Firmen auch nicht als neue Daten im ERP angelegt werden.

Zur Auflösung dieser Situation sind zwei Varianten möglich.

6.6.1 Variante 1

Der Status wird nicht als Filter verwendet und damit wird das Inaktiv-Kennzeichen in jedem Fall übertragen. Sollten die System aber über unterschiedliche Datenbestände verfügen, werden dadurch auch bereits inaktive Firmen als neue Datensätze übertragen. Wenn dies nicht gewünscht ist, sollte Variante 2 verwendet werden.

6.6.2 Variante 2

Bei dieser Umsetzung bleibt der Prozessfilter bestehen und der Prozess überträgt nur aktive Firmen. Damit jetzt aber auch das Inaktiv-Kennzeichen übertragen werden kann, wird ein zusätzlicher Prozess eingerichtet. Dieser filtert auf inaktive Firmen und darf keine neuen Datensätze anlegen. Dies kann mit dem Parameter „Verhalten bei keiner Übereinstimmung“ erreicht werden. Der Prozess muss auch nur das Statusfeld zuordnen.

Beide Prozesse arbeiten kontinuierlich und auf der Basis von Änderungen. Unbekannte bzw. neue inaktive Firmen werden nicht in das ERP übernommen. Für bekannte Firmen kann der Status „inaktiv“ übertragen werden. Eine weitere Übertragung findet ab da nicht mehr statt.

6.7 Sage 100 Belegübertragung (Draft)

Die Belegübertragung ist eine Funktion der Sage 100 Verbindung zum Anlegen von EK- oder VK-Belegen. Diese Funktion kann nur neue Belege erzeugen.

6.7.1 Einschränkungen

6.7.2 Funktionsfelder

6.7.3 Positionstyp

6.8 Seriendruck (Draft)

Die Zahlenformatierung scheint sich nach der Sprache des jeweiligen Wortes(!) zu richten. Da Word im Standard die Sprache automatisch erkennt, waren manche Merge-Fields als Englisch markiert, manche als Deutsch. Deswegen hatte ich heute wieder ständig andere Währungformate, obwohl alle gleich mit „#,##0.00 €“ formatiert waren. Jetzt habe ich mal alle Wörter auf „Deutsch“ gestellt und damit reicht der numerische Schalter # „C“, um eine Zahl im korrekten Format mit € Zeichen auszugeben (vorausgesetzt, die Sprache steht auf „Deutsch (Deutschland)“. Wenn die Word-Automatik wieder „schlau“ ist, stellt es die Sprache auf English (US) um, und dann erfolgt die Ausgabe mit $ Zeichen.

Die automatische Spracherkennung sollte also deaktiviert sein, dann braucht es nur das „C“ für die Formatierung. Ich hoffe, das Deaktivieren der Spracherkennung ist eine Dokumenteigenschaft und keine Programmeinstellung.

Umgang mit Bildern und Picture:

Umgang mit Ranges und geschachtelten Ranges

Parameter zur Steuerung der Ausgabe

6.9 Nachfolgeprozesse (Draft)

Mit Nachfolgeprozessen können Sie einzelne Prozesse in eine Abfolge bringen, um z.B. verschiedene Ziele anzusprechen.

Der ausgewählte Nachfolgeprozess wird mit einem Warteschlangeneintrag vom Typ „Nachfolger“ gestartet und erhält zusätzlich einen Verwis auf den aktuellen Warteschlangeneintrag. Im aktuellen Warteschlangeneintrag wird vermerkt, welche Datensätze erfolgreich oder mit einer Warnung verarbeitet wurden. Nur diese Datensätze werden vom Nachfolger verarbeitet.

Ein Schema-basierter Prozess kann nur Daten an einen Schema-basierten Prozess übergeben. In den meisten Fällen muss dabei auch das Quellobjekt identisch sein.

Ein Abfrage-basierter Prozess kann nur Daten an einen Abfrage-basierten Prozess übergeben. Dabei kann festgelegt werden, ob die Daten selbst oder die ID-Werte per Datenabbildung übergeben werden. Ohne ID-Felder in den Quelldaten wird immer der Änderungsspeicher verwendet. Wenn ID-Felder angegeben wurden…

Beim Änderungsspeicher, ohne die Datenabbildung, muss im Nachfolger das ID-Feld angegeben, sonst hält er das für eine ChangedGuid und findet nichts

6.10 Gelöschte Daten aus Zoho CRM (Draft)

6.11 Die laufende Nummer des Mandanten (Draft)

Diese Nummer an Prozessen struktuiert die synchronisierten Daten in Mandanten. In Systemen wie Sage CRM, CAS oder Salesforce können die Sync-Statusfelder zu den Stammdaten nummeriert angegeben werden und beziehen sich auf diese Nummer. Dies steuert, welche Prozesse auf den Datensatz zugreifen dürfen und damit auch, welches Ziel der Datensatz hat. Diese Nummer wird aber auch für Datenabbildungen verwendet. Ein Prozess sucht nur nach vorhandenen Datenabbildung für seine laufende Nummer und auch nur diese Datenabbildungen werden ggf. erzeugt oder aktualisiert. Durch die Verwendung von unterschiedlichen Nummern können Datensätze dadurch auch gezielt doppelt angelegt werden.

Folgende Beispiele sollen die Verwendung der Nummer veranschaulichen.

KAPITEL 7

On-Premises Version

Die On-Premises Version kann lokal installiert werden. Dies wird mit einer Windows-Installationsroutine unterstützt. Die Anwendung kann aber auch in einer Linux-Umgebung eingesetzt werden. Hier erfolgt die Bereitstellung der Anwendung und der Web-API manuell.

Damit die installierte Version verwendet werden kann, wird eine Lizenz benötigt. Diese kann bei Sellmore erworben werden.

Durch die Windows-Installationsroutine werden die Hauptanwendung (Daemon), die Web-API (Service) und der Desktop-Administrator in einem Windows-Betriebssystem installiert. Die Anwendungen setzen .Net-Core SDK 3.1 und .Net-Framework 4.7.2 voraus.

Durch die Installationsroutine wird kein Datenbanksystem installiert. Sie können einen SQL-Server oder eine MariaDB für den Syncler verwenden. Beides muss eigenständig installiert und konfiguriert werden. Bei der ersten Einrichtung des Synclers mit dem Desktop-Administrator konfigurieren Sie die Zugangsdaten und den Typ des Datenbanksystems. Die Datenbank wird dann automatisch mit dem angegebenen Namen erzeugt, falls diese nicht bereits existsiert. Alles weitere erfolgt automatisch.

Die Installationsroutine registriert zwei Windows-Dienste.

Der Windows-Dienst „Syncler Windows Daemon“ ist die Hauptanwendung. Diese Anwendung ist für die Ausführung von Prozessen, Datendiensten und Projekten zuständig. Sie kommuniziert ausschließlich über die Datenbank und besitzt keine eigenen Schnittstellen.

Der Windows-Dienst „Syncler Windows Service“ stellt die Web-API des Synclers mit einem Kestrel-Webserver bereit. In der Grundeinstellung ist diese über http://localhost:3030/SisService aufrufbar. Der Listen-Port der Web-API kann über die Syncler-Konfiguration geändert werden. Eine Änderung des Ports erfordert den Neustart des Windows-Dienstes.

Die Web-API ist die Schnittstelle für alle Zugriffe auf den Syncler und wird auch vom Desktop-Administrator genutzt. Sie stellt Endpunkte für die Konfiguration, Push-Nachrichten, IDoc-Nachrichten, ERP-Fokus-Zugriffe oder Webhooks bereit. Die Schnittstellen-Dokumentation kann über http://localhost:3030/SisService/swagger/index.html aufgerufen werden. In bestimmten Endpunkten wird eine „TenantId“ benötigt. Diese lautet bei lokalen Installationen immer „Master“.

Für einen Zugriff auf die Web-API kann direkt der Kestrel-Server verwendet werden. Dazu muss der definierte Port in den beteiligten Firewalls freigegeben werden.

Es besteht aber auch die Möglichkeit, den Zugriff durch einen Reverse-Proxy bereitzustellen. Dies ist in zentralverwalteten Infrastrukturen von Vorteil.

Im IIS können Sie dazu eine Anwendung hinzufügen, die eine URL Rewrite InboundRule definiert. Diese leitet den externen Zugriff dann auf die lokal definierte Adresse inklusive des Listen-Ports weiter.

KAPITEL 8

Syncler Web API und SDK

Die Plattform Syncler bietet eine Web API und eine SDK Funktionalität. Damit können eigene Lösung außerhalb der bereitgestellten Szenarien entwickelt werden.

8.1 Der SDK-Helper

Der Helper ist in Skripten der Kommunikationsweg zur Anwendung und stellt darüberhinaus unterstützende Funktionen zur Verfügung. Parameter aus Verbindung und Prozessen können damit ausgetauscht werden.

Neben dem Helper und einigen .Net Basisfunktionen steht auch noch die Bibliothek Newtonsoft.Json zur Verfügung, die vorallem das Lesen und Erzeugen von Json-Daten stark vereinfacht.

Alle Eigenschaften und Funktion des Helper beziehen sich immer auf den aktuellen Account.

8.1.1 Eigenschaften

Folgende Eigenschaften stehen am Helper zur Verfügung. Einige werden nur intern vom Helper genutzt oder indirekt über Methoden gesetzt.

Helper.IsCancelled (bool)

Die Wert zeigt an, dass das Skript einen Abbruch angefordert hat. Der Wert wird auch mit der Methode Cancel gesetzt. Der Wert wird in den SDK-Prozessen, der SDK-Verbindung und der Transformation ausgewertet.

Helper.ConnectionId (int)

Die ID der aktuellen Verbindung. Dieser Wert kann für API-Aufrufe verwendet werden.

Helper.ProcessId (int)

Die ID des aktuellen Prozesses. Dieser Wert kann für API-Aufrufe verwendet werden. Wird von den Methoden IncreaseComplementaryMappings, IncreaseParallelMappings und InsertLog verwendet.

Helper.ClientNumber (int)

Die Client-Number des aktuellen Prozesses. Dies ist ein Prozessparameter, der für die Datenstrukturierung genutzt werden kann. Die Methoden GetDataMappingComplementaryBySourceId und GetDataMappingParallelByTargetId schränken das Ergebnis mit diesem Wert ein, realisieren das aber mittels Abfrage direkt.

Helper.WhereClause (string)

Dies ist der Prozess-Filter für die Abfrage von Quelldaten. In den SDK-Prozessen wird der Wert zurück an den Prozess gegeben und kann damit die Prozessausführung direkt beeinflussen.

Helper.ForceQueryAll (bool)

Dies ist der Prozessparameter für die Aktivierung des kompletten Lesens ohne Änderungseinschränkung. In den SDK-Prozessen wird der Wert zurück an den Prozess gegeben und kann damit die Prozessausführung direkt beeinflussen.

Helper.ConflictAction (string)

Dies ist der Prozessparameter für die Behandlung von Konflikten. Die möglichen Werte sind USE_SOURCE, USE_TARGET, SKIP, IGNORE und SOURCE_CHANGES.

Helper.ErrorAction (string)

Dies ist der Prozessparameter für die Behandlung von Fehlern. Die möglichen Werte sind REPEAT_ALL, REPEAT_DIFFERENCES, QUERY_NEXT, IGNORE und STOP.

Helper.ClientWhereClause (string)

Dies ist der Prozessparameter für die Suche nach Übereinstimmungen bei unbekannten Datensätzen.

Helper.DublettActionNoMatch (string)

Dies ist der Prozessparameter für die Behandlung von keinem Treffer bei der Übereinstimmungssuche. Die möglichen Werte sind SKIP, SKIP_WITH_WARNING, WRITE, REPEAT und REPEAT_WRITE.

Helper.DublettActionOneMatch (string)

Dies ist der Prozessparameter für die Behandlung von einem Treffer bei der Übereinstimmungssuche. Die möglichen Werte sind SKIP, SKIP_WITH_WARNING, MAPPING_ONLY, MAPPING_AND_SYNCH und WRITE_NEW.

Helper.DublettActionMultipleMatches (string)

Dies ist der Prozessparameter für die Behandlung von mehreren Treffern bei der Übereinstimmungssuche. Die möglichen Werte sind SKIP, SKIP_WITH_WARNING, MAPPING_ONLY_FIRST, MAPPING_AND_SYNCH_FIRST und WRITE_NEW.

Helper.SecondWhereClause (string)

Dies ist der Prozess-Zweitfilter für die Auswahl von transformierten Quelldaten. In den SDK-Prozessen wird der Wert zurück an den Prozess gegeben und kann damit die Prozessausführung direkt beeinflussen.

Helper.SourceObject (string)

Der aktuelle Quellobjekttyp in Prozessskripten.

Helper.TargetObject (string)

Der aktuelle Zielobjekttyp in Prozessskripten. In der SDK-Verbindung definiert der Wert das aktuell angeforderte Objekt.

Helper.SourceType (string)

Der Typ der Quellverbindung in Prozessskripten.

Helper.TargetType (string)

Der Typ der Zielverbindung in Prozessskripten.

Helper.SourceSchemaObject (JObject)

Ein Schemaobjekt zur Beschreibung des aktuellen Quellobjekttyps.

Helper.TargetSchemaObject (JObject)

Ein Schemaobjekt zur Beschreibung des aktuellen Zielobjekttyps. In der SDK-Verbindung definiert der Wert das aktuell angeforderte Objekt.

Helper.Statement (string)

Die Abfrage der aktuellen Abfrageanforderung. Dies wird in JSON Abfrage Skript übergeben.

Helper.Action (JObject)

Dieses Objekt beschreibt die aktuellen Ausführungsparameter.

Helper.Page (int)

Dieser Wert wird beim Lesen mit Paging mit jeder Ausführung erhöht. Der Startwert ist 0.

Helper.SkipLoading (bool)

Dieser Wert wird von der Methode SkipLoad gesetzt. In den SDK-Prozessen führt der Wert true zum Überspringen des Lesens aus der Quelle.

Helper.InnerException (string)

Dieser Wert wird vom Datensatzskript in den SDK-Prozessen verwendet. Er wird als Fehlermeldung in den Prozess gegeben, wenn das Ergebnis Failed lautet. In diesem Fall muss ein Wert zugewiesen sein.

Helper.GetObject (JObject)

In den SDK-Prozessen wird hier der aktuelle Datensatz an das Datensatzskript übergeben.

Helper.GetChildObjects (JArray)

Im SDK-Prozess für geschachtelte Daten enthält dieses Array die Liste der transformierten Positionsdatensätze. Der Objektparameter „DELETE“ führt je nach Verbindung zum Löschen der Position.

Helper.SetObject (JObject)

Dieses Objekt repräsentiert das aktuelle Zielobjekt. In den SDK-Prozessen wird das Zielobjekt für das Datensatzskript übergeben. In der SDK-Verbindung und dem JSON-Daten schreiben Skript ist hier das Zielobjekt enthalten.

Helper.SetObjectChanges (JObject)

Dieses Objekt repräsentiert die Änderungen des aktuellen Zielobjektes und steht in der SDK-Verbindung zur Verfügung. Sobald der Datensatz über einen Primärschlüssel verfügt können so die tatsächlichen Änderungen in den Daten ermittelt werden.

Helper.DeleteSetObject (bool)

Dieses Objekt soll von der Verbindung gelöscht werden.

Helper.Mappings (JArray)

Diese Liste enthält alle Feldzuordnungen, die im aktuellen Prozess definiert sind. In den SDK-Prozessen werden Änderungen an dieser Liste für das Schreiben übernommen. Die Eigenschaften des einzelen Objektes sind SourcePath (string), SourceColumn (SchemaColumn), TargetPath (string) und TargetColumn (SchemaColumn).

8.1.2 Methoden

Folgende Methoden stehen im Helper direkt oder über Eigenschaften zur Verfügung.

Helper.Cancel()

Fordert einen Abbruch der aktuellen Ausführung an. Die Anforderung wird in SDK-Prozessen, der SDK-Verbindung und in Transformationen ausgewertet.

Helper.SkipLoad()

Fordert das Überspringen der Lese-Operation an. Die Anforderung wird in SDK-Prozessen ausgewertet und überspringt das Lesen der Quelldaten.

Helper.GetParam

Parameter:

- string Name

Rückgabewert: string

Liefert einen Parameterwert aus der Parameterliste über den Namen des Parameters. Wenn der Parameter nicht vorhanden ist, wird eine leere Zeichenkette zurückgeliefert.

Helper.GetParam<T>

Parameter:

- string Name
- T DefaultValue = default

Rückgabewert: T

Liefert einen Parameterwert aus der Parameterliste über den Namen des Parameter in einem bestimmten Typ. Wenn der Parameter nicht vorhanden ist, wird der DefaultValue verwendet. Die Typangabe kann bei einem typisierten Defaultvalue auch weggelassen werden. Abhängig vom Typ des Parameters wird der Wert oder Verweis geliefert.

Helper.GetParamOrNull<T>

Parameter:

- string Name

Rückgabewert: T

Liefert einen Parameterwert aus der Parameterliste über den Namen des Parameter in einem bestimmten Typ. Wenn der Parameter nicht vorhanden ist, wird Null verwendet. Diese Notation kann für explizites Nullable verwendet werden. z.B. DateTime?

Helper.SetParam<T>

Parameter:

- string Name
- T Value

Speichert einen Parameter in der Parameterliste. Sollte der Parameter bereits vorhanden sein, wird der Wert aktualisiert. Die Typangabe kann bei typisierten Value auch weggelassen werden.

Helper.FieldStringToColumnObject

Parameter:

- string FieldString

Rückgabewert: JObject

Mit dieser Methode kann eine Zeichenkette, z.B. aus einem Filter, die in Feldnotation übergeben wird, in ein JObject mit ansprechbaren Eigenschaften umgewandelt werden.

Helper.GetDataMappingBySourceId

Parameter:

- int ProcessId
- string SourceId

Rückgabewert: SisDataMapping

Ruft eine Datenabbildung aus der Datenbank des aktuellen Accounts über die Quell-ID ab. Datenabbildungen sind Prozess-bezogen und enthalten eine Quell- und Ziel-Identifikation.

Helper.GetDataMappingByTargetId

Parameter:

- int ProcessId
- string TargetId

Rückgabewert: SisDataMapping

Ruft eine Datenabbildung aus der Datenbank des aktuellen Accounts über die Ziel-ID ab. Datenabbildungen sind Prozess-bezogen und enthalten eine Quell- und Ziel-Identifikation.

Helper.GetDataMappingComplementaryBySourceId

Parameter:

- int ProcessId
- string SourceId
- string TargetObject = null

Rückgabewert: List<SisDataMapping>

Ruft eine Liste von Datenabbildungen entgegengerichteter Prozesse zum aktuellen Prozess aus der Datenbank des aktuellen Accounts über eine Quell-ID ab. Zusätzlich kann das Target-Objekt eingeschränkt werden. Dies kann erforderlich sein, wenn zwei unterschiedliche Objekte das gleiche Zielobjekt haben. Die Arbeit mit komplementären Datenabbildungen ist für das Konfliktmanagement und die Zielerkennung bei bidirektionalen Synchronisationen relevant.

Helper.GetDataMappingParallelByTargetId

Parameter:

- int ProcessId
- string TargetId

Rückgabewert: List<SisDataMapping>

Ruft eine Liste von Datenabbildungen paralleler Prozesse zum aktuellen Prozess aus der Datenbank des aktuellen Accounts über eine Ziel-ID ab. An diesen muss die Änderungsinformation des Ziels angepasst werden, da dort ansonsten ein Konflikt erkannt wird.

Helper.SaveDataMapping

Parameter:

- SisDataMapping Mapping

Rückgabewert: SisDataMapping

Speichert eine neue oder aktualisiert eine bestehende Datenabbildung.

Helper.IncreaseOwnDataMapping

Parameter:

- SisDataMapping DataMapping
- object NewUpdatedInfoA
- object NewUpdatedInfoB
- bool OnlyA = false
- bool OnlyB = false

Diese Methode aktualisiert die Änderungsinformation von Quelle und Ziel für die aktuelle Datenabbildung. Zum Auflösen eines Konflikts können die einzelnen Informationen gezielt per Parameter aktualisiert werden.

Helper.IncreaseComplementaryMappings

Parameter:

- string CurrentSourceId
- string CurrentTargetId
- object NewUpdatedInfoA
- object NewUpdatedInfoB
- bool OnlyA = false
- bool OnlyB = false

Diese Methode sucht und aktualisiert oder legt komplementäre Datenabbildungen an. Bei Neuanlagen oder in Konfliktsituationen kann dies auch nur partiell erfolgen.

Helper.IncreaseParallelMappings

Parameter:

- string CurrentSourceId
- string CurrentTargetId
- object NewUpdatedInfoA
- object NewUpdatedInfoB
- bool OnlyA = false
- bool OnlyB = false

Diese Methode sucht und aktualisiert oder legt parallele Datenabbildungen an. Bei Konfliktsituationen kann dies auch nur partiell erfolgen.

Helper.GetProcessInfoList

Parameter:

- int? SourceConnectionId = null
- int? TargetConnectionId = null

Rückgabewert: List<SisProcessInfo>

Liefert eine Liste von Informationsobjekten zu den Prozessen des aktuellen Accounts. Die Abfrage kann auf bestimmte Verbindungen eingeschränkt werden.

Helper.GetProcessInfoComplementary

Parameter:

- int ProcessId

Rückgabewert: List<SisProcessInfo>

Liefert eine Liste von Informationsobjekten zu den komplementären Prozessen einen Prozesses.

Helper.GetProcessInfoParallel

Parameter:

- int ProcessId

Rückgabewert: List<SisProcessInfo>

Liefert eine Liste von Informationsobjekten zu den parallelen Prozessen einen Prozesses.

Helper.InsertLog

Parameter:

- SisLog Log

Speichert einen Protokolleintrag in der Datenbank des aktuellen Accounts.

Helper.InsertLog

Parameter:

- string Message
- int Level

Speichert einen Protokolleintrag in der Datenbank des aktuellen Accounts. Mit dem Level wird der Typ des Eintrags festgelegt. 0 = Meldung, 1 = Fehler, 2 = Warnung, 3 = Nachricht, 4 = Rückmeldung, 5 = Debug

Helper.InsertMessage

Parameter:

- string Message
- int Level

Speichert eine Nachricht in der Broadcast-Datenbank des aktuellen Accounts. Diese Nachrichten werden in der Benutzeroberfläche angezeigt, aber nicht dauerhaft vorgehalten. Mit dem Level wird der Typ des Eintrags festgelegt. 0 = Meldung, 1 = Fehler, 2 = Warnung, 3 = Nachricht, 4 = Rückmeldung, 5 = Debug

Helper.InvokeUrl

Parameter:

- string Url
- string Method
- JObject Header
- string Data
- bool Base64 (default: false)

Rückgabewert: string

Diese Methode führt einen HTTP-Request aus und liefert die Antwort als Zeichenkette zurück. Zusätzliche Header können als JObject übergeben werden. Einzelne Properties werden als einzelner Header-Parameter übernommen. Mit dem Parameter Base64 werden die abgerufenen Daten binär in Base64 konvertiert und zurückgegeben.

Helper.GetParameterList

Parameter:

- string Name
- string ConnectionId = null
- string ProcessId = null

Rückgabewert: List<SisParam>

Ruft eine Parameterliste aus der internen Datenbank ab. Jeder Parameter hat einen Namen, einen Wert und eine ID. Zusätzlich können Parameter noch einer Verbindung oder einem Prozess zugeordnet werden. Das ist für die Strukturierung und Bereinigung sinnvoll.

Helper.SaveParameter

Parameter:

- SisParam Parameter
- string ConnectionId = null
- string ProcessId = null

Speichert einen Parameter in der internen Datenbank.

Helper.DeleteParameter

Parameter:

- int ParameterId

Löscht einen Parameter aus der internen Datenbank.

Helper.InvokeGetData

Parameter:

- string ConnectionId
- string TargetObject
- List<SisParam> GetParams

Rückgabewert: JArray

Ruft Daten aus einer Verbindung ab. Das Schemaobjekt wird über TargetObject festgelegt. Die Liste der Parameter steuert die Abfrage. Dabei unterstützen nicht alle Verbindungen den gleichen Umfang an Optionen. Übliche Namen von Parametern sind: GETDATA_ID, GETDATA_RELATED_ID, GETDATA_RELATED, GETDATA_WHERE, GETDATA_MODIFIED, GETDATA_ORDER, LAST_SYNC_DATE, LAST_SYNC_VERSION Die Ausführung erfolgt über die Syncler API.

Helper.InvokeSetData

Parameter:

- string ConnectionId
- string TargetObject
- JObject JsonObject

Rückgabewert: JObject

Speichert Daten über eine Verbindung. Das Schemaobjekt wird über TargetObject festgelegt. Zusätzliche Parameter, wie „DELETE“ (bool) können über die Eigenschaft „Params“ am JsonObjekt mitgegeben werden.

Helper.ServiceCall

Parameter:

- string Method
- string Url
- string Data

Rückgabewert: string

Führt einen Aufruf der Syncler API aus. Als Url muss nur der Endpunkt übergeben werden.

Helper.InsertAction

Parameter:

- int ProcessId
- DateTime ExecuteDate
- bool IsAdhoc
- List<SisParam> ActionParams

Speichert einen neuen Warteschlangeneintrag für einen Prozess.

8.1.3 Parameter

SisParam.Name

Parameter Name (string)

SisParam.Value

Parameter Wert (object)

SisParam.ID

Parameter ID (int) Wird beim Speichern in der Datenbank gesetzt.

SisParam.GetValue()

Parameter Wert als String

SisParam.GetValue<T>(DefaultValue)

Parameter Wert als Typ T oder Default

SisParam.GetValueOrNull<T>()

Parameter Wert als Typ T oder Null

SisParam.HasValue()

Parameter hat einen Wert (bool)

8.1.4 Datenabbildung

SisDataMapping.ID

Interne Datenbank ID (int)

SisDataMapping.ProcessId

Zugeordneter Prozess (int)

SisDataMapping.Description

Datensatzbeschreibung (string)

SisDataMapping.SourceRecordId

ID des Quelldatensatzes (string)

SisDataMapping.TargetRecordId

ID des Zieldatensatzes (string)

SisDataMapping.LastSyncDate

Letztes Datum der Synchronisation (DateTime)

SisDataMapping.TargetIsDeleted

Zieldatensatz wurde gelöscht (bool)

SisDataMapping.LastSyncInfo

Parameterliste mit Änderungs- und Zusatzinformationen (List<SisParam>)

8.1.5 Prozessbeschreibung

SisProcessInfo.ID

Prozess ID (int)

SisProcessInfo.Name

Prozessname (string)

SisProcessInfo.DisplayName

Voller Prozessname (string)

SisProcessInfo.SourceObject

Names des Quelldatensatzes (string)

SisProcessInfo.TargetObject

Name des Zieldatensatzes (string)

SisProcessInfo.SourceConnectionId

Quellverbindung ID (int)

SisProcessInfo.TargetConnectionId

Zielverbindung ID (int)

SisProcessInfo.ClientNumber

Mandantennummer (int)

SisProcessInfo.IsScheduled

Zeitsteuerung ist aktiv (bool)

SisProcessInfo.SourceType

Typ der Quellverbindung (string)

SisProcessInfo.TargetType

Typ der Zielverbindung (string)

SisProcessInfo.ProcessType

Typ des Prozesses (string)

8.1.6 Protokoll

SisLog.CreatedDate

Erstelldatum (DateTime)

SisLog.Level

Level der Nachricht 0 (message) - 5 (debug) (int)

SisLog.ProcessId

Zugeordneter Prozess (int)

SisLog.ActionId

Zugeordneter Warteschlangeneintrag (int)

SisLog.RecordType

Names des Datensatzes (string)

SisLog.RecordId

ID des Datensatzes (string)

SisLog.LogMessage

Meldung (string)

8.2 Die Syncler API (Draft)

8.3 Universal-SDK-Prozesse

Die Universal SDK-Verbindung dient der Entwicklung einer Anbindung eines fremden Systems zum Lesen, Schreiben oder Ausführen von Abfragen.

Die Universal SDK-Prozesse sind unabhängig von dieser Verbindung und können mit jeder beliebigen Verbindung verwendet werden.

Es stehen zwei grundlegende Prozess-Plugins zur Verfügung.

8.3.1 Universal Prozess (SDK)

Dies ist ein einfacher Universalprozess für die Übertragung eines Datenobjektes zwischen zwei Systemen. Es stehen die Bereiche Transformation und Zuordnungen zur Verfügung. Außerdem gibt es drei Skripte, die im Anschluss erläutert werden.

Aufbauend auf diesem Prozess gibt es noch eine Variante für Sage CRM, die eine gezielte Nachbearbeitung von Stammdaten für eine konsistente Datenstruktur beinhaltet. Der Einsatz wird bei der Arbeit mit Personen empfohlen.

8.3.2 Universal Sync Prozess für geschachtelte Daten (SDK)

Dies ist ein Universalprozess für die Übertragung von geschachtelten Datentypen zwischen zwei Systemen. Es stehen die Bereiche Transformation und Zuordnungen für den Kopfdatensatz und die Positionsdatensätze zur Verfügung. Außerdem gibt es drei Skripte, die im Anschluss erläutert werden.

8.3.3 Universal Abfrageprozess (SDK)

Bei diesem Prozess-Plugin handelt es sich um einen klassischen Abfrageprozess für die Verwendung mit einer Universal SDK-Verbindung. Der Prozess selbst bietet keine zusätzlichen Skript-Möglichkeiten.

8.3.4 SDK-Skripte

Für die Prozess-Plugins „Universal Prozess (SDK)“ und „Universal Sync Prozess für geschachtelte Daten (SDK)“ können zusätzliche Skripte definierte werden, die an bestimmten Punkten der Verarbeitung ausgeführt werden. In allen Skripten steht der SDK-Helper zur Verfügung, der die Schnittstelle zwischen Prozess, Daten und Skript realisiert.

Das „Prozess ausführen“-Skript wird zu Beginn der Prozessausführung gestartet und läuft noch vor dem Lesen von Daten. Das Skript kann dabei den Filter, das Abrufen aller Datensätze und den zweiten Filter überschreiben. Auch werden alle Parameter aus dem Helper in den Prozess übernommen und stehen dadurch auch in der Quellverbindung beim Lesen von Daten zur Verfügung. Mit SkipLoading kann das eigentliche Lesen auch übersprungen werden.

Das „Datensatz Verarbeitung“-Skript wird für jeden gelesenen Datensatz einzeln ausgeführt. Die Ausführung findet direkt vor dem Füllen und Schreiben des Zielobjektes statt. Transformationen, Konfliktprüfung, Übereinstimmungssuche und die Neuanlagebedingung wurden zu diesem Zeitpunkt bereits ausgeführt. Falls der Datensatz dabei übersprungen wurde, wird das Skript nicht ausgeführt. Über den SDK-Helper stehen das Quellobjekt (GetObject), das Zielobjekt (SetObject) und die Feldzuordnungen (Mappings) zur Verfügung und können verändert werden. Die Skript-Methode kann folgende Werte zurückgeben, die dann die weitere Verarbeitung beeinflussen können.

- cancelled (die gesamte Verarbeitung wird abgebrochen)
- skipped (aktueller Datensatz wird übersprungen)
- failed (InnerException wird ausgewertet und Datensatz wird mit diesem Fehler abgebrochen)
- repeat (der aktuelle Datensatz wird wiederholt)
- success

Für den Universal Sync Prozess für geschachtelte Daten (SDK) bietet das Skript auch Zugriff auf die Positionsliste (GetChildObjects). Da die Positionsliste nach der Skript-Ausführung komplett übernommen wird, können so auch Positionen hinzugefügt oder entfernt werden.

Das „Prozess abschließen“-Skript wird beim Abschluss der Prozessausführung gestartet. Dieses Skript kann die aktuelle Verarbeitung nicht mehr beeinflussen, aber andere Prozesse starten, oder die Prozessparameter für die Änderungsgrenzwerte verändern.

8.4 Die Entwicklung eines Verbindungsplugins in .Net

Jedes Verbindungsplugin wird durch eine Klasse implementiert, die von ConnectionBase ableitet. Die Klasse wird in einer .Net Bibliothek entwickelt, die das SynclerCommon Assembly einbindet. Bereitgestellt wird die DLL der Bibliothek in den connections Verzeichnissen der Installation. Diese werden nur beim Start des Systems eingelesen.

In der Basisklasse werden virtuelle Methoden definiert, die in den Verbindungsplugins individuell implementiert werden. Diese Methoden dienen der Steuerung des Verbindungsplugins. Bei der Planung eines Verbindungsplugins müssen diese Berührungspunkte mit der Syncler Umgebung berücksichtigt werden.

Auch stehen Eigenschaften bereit, die verwendet werden könnnen. Unter anderen werden Objekte für den Zugriff auf die Konfiguration der Instanz, die aktuelle Datenbank und Übersetzungen bereitgestellt.

- SisDatabaseWrapper Database: Datenbank-Instanz
- SisTenantConfig Config: Konfiguration der Instanz
- SisTenantInstance Instance: Instanz mit Übersetzer

Die Benutzeroberfläche für die Konfiguration der Verbindung wird durch Attribute an der Klasse und den Eigenschaften definiert. Die Page-Attribute an der Klasse aktivieren bestimmte Konfigurationsseiten. Mit ‚PageCommon' wird die Konfiguration von Eigenschaften aktiviert. Eigenschaften können mit Attributen beschriftet und kategorisiert werden. Auch Anleitungstexte lassen sich so platzieren. Für alle Texte können Übersetzungscodes aus den erweiterbaren XML-Übersetzungsdateien genutzt werden.

- LocalizedCategory: übersetzte Kategorie
- LocalizedDisplayName: übersetzte Beschriftung
- LocalizedDescription: übersetzte Anleitung

Mit weiteren Attributen kann die Konfiguration noch detailierter definiert werden.

- CommaSeparatedValues: komma-separierte Daten
- JsonFormat: Legt ein Schema für JSON-Daten fest
- KeyValueList: Liste von Name-Wert-Paaren in Feldnotation
- LocalizedEnumDescription: für die Übersetzung von Enumerations
- Multiline: mehrzeiliger Wert
- OnlyMaster: Eigenschaft steht nur On-premises zur Verfügung
- Required: Es muss ein Wert festgelegt werden
- RequiresSchemaRefresh: Eine Änderung erzwingt ein Schema-Refresh

8.4.1 Einsatz der Verbindung

Von der Verbindung wird bei Konfiguration eine verschlüsselte Serialisierung in der Datenbank gespeichert. Sobald die Verbindung verwendet wird, wird für jede Anforderung eine Instanz erzeugt und ausgeführt. In einer parallelen Ausführung existieren dadurch mehrere Instanzen dieser Klasse, die untereinander keine Berührungspunkte haben. Aus diesem Grund sollten gemeinsam genutzte Informationen zur Laufzeit (z.B. Token) nicht in der Verbindung, sondern separat in der Parameter-Tabelle gespeichert werden. Wird eine Verbindung nicht weiter benötigt, wird diese wieder aus dem Speicher entfernt. Geänderte Informationen in der Verbindung gehen dabei verloren bzw. werden zurückgesetzt.

8.4.2 Planen einer Verbindung

Wenn Sie eine Verbindung planen, muss die Funktion mit den Berührungspunkten zum Syncler abgestimmt werden. Daraus ergeben sich dann Vorgehensweisen für die Implementierung.

8.4.3 Nutzung der Verbindung

Folgende Fällen lassen sich bei der Nutzung unterscheiden.

Konfiguration

Die Konfiguration erfolgt über die Benutzeroberfläche des Synclers und wird durch Attribute gesteuert. Durch die Attribute können bereits Pflichtfelder und ähnliches definiert werden, die der Benutzer angeben muss. Beim Speichern der Verbindung und vor der Nutzung durch einen Prozess wird die Validate-Methode aufgerufen. Diese kann Prüfungen durchführen, die die Einsatzfähigkeit gewährleisten. Ggf. aufgetretene Fehler werden als Rückgabe der Methode gemeldet. Eine positive Antwort ist ein null-Wert. Bei einer direkten Verwendung der Verbindung durch die API wird die Validate-Methode nicht aufgerufen.

Schema

Jede Verbindung kann Schemaobjekte bereitstellen, die in u.a. Prozessen verwendet werden. Die Gestaltung von Schemaobjekten ist dabei relativ frei und muss durch die Verbindung verarbeitet werden. So können z.B. Datenobjekte für CRUD-Operationen definiert werden. Auch lassen sich Funktionen mittels Schemaobjekt bereitstellen, die dann durch Prozesse ausgelöst werden. Schemaobjekte können auch beliebig geschachtelt werden, um z.B. logische Business-Objekte bereitzustellen. Schemaobjekte können ID-Werte für das Objekt definieren. Dies ist die Voraussetzung für einen Sync, der Datenabbildungen nutzen soll. Auch können Einzelausführung nur mit ID-Werten ausgelöst werden. Ein Datensatz ohne ID-Werte wird immer als neu interpretiert. Zusätzlich kann ein Schemaobjekt eine Änderungsinformation definieren. Diese wird in Prozessen genutzt, um Grenzwerte für eine änderungsbasierte Synchronisation zu ermitteln. Das Konfliktmanagement basiert ebenfalls auf dieser Information. Wichtig hierbei ist, dass beim Schreiben eines Datensatzes die aktuellen Daten zurückgeliefert werden.

Das Schema wird durch die Methode GetConnectionSchema erzeugt. Diese liefert ein Dictionary mit dem Namen als Key und einer XML-Serialisierung des SisSchemaObject als Value zurück. Die Methode wird automatisch bei der Neuanlage ausgeführt und jedesmal beim Speichern mit aktivierter Schema-Aktualisierung. Nachdem die Methode ausgeführt wurde, wird die Verbindung erneut gespeichert.

Schema-basiertes Lesen

Ein lesender Aufruf an die Verbindung mit einem Schema wird durch die Methode GetData ausgeführt. Dies erhält das aktuelle Schemaobjekt und eine Liste von Parametern. Die Methode muss nun alle Anwendungsfälle umsetzen, die sich aus Parametern und Schema-Objekt ergeben. Sollte die Ausführung durch den Benutzer abgebrochen werden, wird CancellationPending gesetzt. Die Behandlung muss in der Verbindung erfolgen und an geeignete Ausstiegspunkte geprüft werden. In diesem Fall muss mit der Exception SisOperationCanceledException die Verarbeitung verlassen werden.

Folgende Parameter werden verwendet und müssen für eine fehlerfreie und optimierte Nutzung ausgewertet werden.

- SisParam_ProcessObjects: Die gelesenen Daten sollen nicht als Rückgabe, sondern über die Methode ProcessMethod fortlaufend geliefert werden.
- SisParam_GetDataById: Ein einzelner Datensatz soll gelesen werden. Der Wert kann in Feldnotation übergeben werden.
- SisParam_NoMessage: Es sollen keine Nachrichten bei der Verarbeitung erzeugt werden. Ansonsten können Nachrichten über die Methode MessageMethod ausgegeben werden.
- SisParam_GetDataByWhere: Stellt den Filter für eine Listenabfrage durch einen Prozess bereit.
- SisParam_GetDataLimit: Die Rückgabe soll auf einen Anzahl von Datensätzen beschränkt werden. Dies wird durch Testläufe in der Konfiguration von Prozessen genutzt.

Abfrage-basiertes Lesen

Abfrage-Prozesse und Reports nutzen diese Art des Lesens. Eine Implementierung ist optional für diese Anwendungsfälle. Die Methode GetQuerySchema wird aufgerufen, um das Schema der aktuellen Abfrage zu ermitteln und in der Konfiguration zu verwenden. Die Methode GetQueryData wird bei der Anforderung von Daten aufgerufen. Beide Methoden erhalten die definierte Abfrage als Parameter. Die Anforderung von Daten erhält zusätzlich eine Datenabbildung für die Definition des Kontexts.

Reports im Syncler-Focus basieren auf SQL und werden entsprechend der Benutzerangaben (Sortierung, Filter, Paging) aufbereitet übergeben. Zusätzlich wird ein Count-Statement übergeben, welches die Gesamtanazhl an Datensätzen ermittelt. Da dieser Anwendungsfall durch eine Benutzeroberfläche gesteuert wird, muss das Antwortverhalten optimiert werden.

Abfrage-Prozesse stellen keine Anforderungen an das Format. Platzhalter für Änderungsinformationen werden ggf. durch den Prozess ersetzt.

Folgende Platzhalter können genutzt werden.

- #Mandant#: frei verwendbarer globaler Wert
- #UserService#: aktuelles Benutzerkennzeichen aus Syncler-Focus
- #SourceId#: Kontext-ID aus dem aktuellen System. Wird per Datenabbildung bereitgestellt und muss ersetzt werden.
- #TargetId#: Kontext-ID aus dem anfragenden System. Wird per Datenabbildung bereitgestellt und muss ersetzt werden.
- #OpportunityId#: Kontext-ID aus dem anfragenden System. Wird per Datenabbildung bereitgestellt und muss ersetzt werden.
- #LastVersion#: Wird ggf. durch den Prozess ersetzt.
- #LastDatetime#: Wird ggf. durch den Prozess ersetzt.
- #FlowFilter#: Wird ggf. durch den Ablauf ersetzt.

Schema-basiertes Schreiben

Für das Schema-basierte Schreiben wird die Methode SetData aufgerufen. Diese Methode erhält das aktuelle Objekt und das aktuelle Schemaobjekt als Parameter. Als Antwort wird der aktuelle Datensatz nach der Änderung erwartet. Diese muss ggf. erzeugte ID-Werte und Änderungsinfromationen enthalten. Dies wird für Datenabbildungen und Konfliktmanagement in einem Sync-Szenario verwendet. Eine Lösch-Anforderung wird durch den Parameter „DELETE“ am aktuellen Objekt übergeben. Sollten Fehler auftreten nachdem der Datensatz bereits erzeugt wurde, können diese über den out-Parameter InnerException gemeldet werden. In diesem Fall wird eine Datenabbildung erzeugt, womit keine erneute Neuanlage ausgeführt wird. Die Operation wird dennach als fehlerhaft in der Fehlerbehandlung ausgewertet. Z.B. kann damit der Datensatz wiederholt werden.

Generell sollte eine Verbindung nur tatsächliche Änderungen übertragen. Dafür sind Objekte und Spalten als geändert markiert (isupdated) und können dementsprechend nur partiell übertragen werden.

Prozessausführung

Ein Schema-basierter Prozess führt bei der Ausführung keine direkte Anfrage an die Quelle mit GetData aus. Als Zielverbindung wird GetData für die Abfrage eines einzelnen Datensatzes verwendet und erwartet eine direkte Antwort. Quellanfragen erfolgen über Load-Methoden, die den unterschiedlichen Fälle entsprechen. Die Rückgabe erfolgt über die ProcessMethod. Jede Load-Methode erhält den Prozess als Parameter, worüber Schema und Parameter ausgewertet werden können. Innerhalb der Load-Methoden sollten Sie mit GetData arbeiten.

Folgende Load-Methoden werden unterschieden.

- LoadAllData: Wird für die Anforderung alle Daten genutzt. Diese Variante kann per Parameter am Prozess erzwungen werden. Ohne Änderungsinformation am Prozess wird ebenfalls diese Methode genutzt.

- LoadChangedData: Wird für die Anforderung von geänderten Daten genutzt. Der Grenzwert steht in Parameters als LAST_SYNC_DATE oder LAST_SYN_VERSION zur Verfügung.
- LoadAdhocData: Wird für die Anforderung eines Datensatzes genutzt. Der konkrete Wert ist im Action-Wert in den Params enthalten und hat als Key den SourceObject-Wert. (Process.Action.Params)
- LoadLockedData: Sollten Datensätze während der Verarbeitung durch eine Sperre zurückgestellt werden, werden diese am Ende erneut angefordert. Die Datensätze sind in LockedObjects enthalten und sollten erneut gelesen werden.
- LoadRepeatData: In bestimmten Fällen sollen Datensätze wiederholt werden. Dies kann per Prozessparameters gesteuert werden. Die betroffenen Datensätze werden indirekt über die vorangegangende Ausführung als Action-Parameter bereitgestellt (Process.Action.GetParam<int>(„PREVIOUS_ACTION_ID“) und PreviousAction.RepeatRecordIds / PreviousAction.LockedRecordIds).
- LoadPlannedData: Diese Methode soll Datensätze abrufen, die infolge der Fehlerbehandlung durch die nächste Ausführung verarbeitet werden sollen. (Process.Action.PlannedRecordIds)
- LoadFailedData: Eine direkte Fehlerwiederholung nutzt diese Methode. Die Datensätze werden indirekt bereitgestellt. (Process.Action.GetParam<int>(„PREVIOUS_ACTION_ID“) und PreviousAction.FailedRecordIds)
- LoadSuccessfulData: Nachfolgeprozesse nutzen diese Methode, um gezielt kontextbezogende Daten abzurufen. Die Bereitstellung erfolgt indirekt. (Process.Action.GetParam<int>(„PREVIOUS_ACTION_ID“))

8.4.4 Anmerkungen

Das Design von Verbindung und Schemaobjekten hängt vom gewünschten Einsatz ab. Lesende Anfragen können nur IDs, Grenzwerte und Filter nutzen. Komplexe Datenanforderungen sind damit nur schwer umsetzbar. Schreibende Anfragen erhalten strukturierte Daten, die beliebige eingesetzt werden können. Eine komplexe Datenanforderungen kann z.B. auch zweistufig mit Anfrage- und Antwort-Schema erfolgen. Die schreibende Operation, der Request, startet die Anfrage und kann das Resultat im Änderungsspeicher ablegen. (Database.SaveChangedData(ChangedData)) Das Antwort-Schema liest den Änderungspeicher aus und liefert das Resultat für die weitere Verarbeitung zurück. Durch Abläufe oder Nachfolgeprozesse kann dann die Kombination realisiert werden.

KAPITEL 9

Release Notes

Hier finden Sie Informationen zu erfolgten Veröffentlichungen und enthaltenen Änderungen.

9.1 Release 4.3.5 - 06.01.2022

9.1.1 Änderungen

Die Datendienste für CAS und Sage CRM zu Sage 100 wurden in eine universelle Form überführt. Damit stehen nur noch 3 Datendienstplugins zur Verfügung, die alle bisherigen Anwendungsfälle durch Konfiguration abdecken.

Alle Abfrageplugins wurden vereinheitlicht. Die Unterscheidung findet nur noch auf der Ebene des Quellsystems statt, da dies ausschlaggebend für die Funktionsweise ist. Nur für Sage CRM gibt es noch eine Ausnahme, da hier für Stammdaten zusätzliche Nacharbeiten enthalten sind.

Die Suche nach vorhandenen Datenabbildungen in anderen Prozessen wird über die laufende Nummer des Mandanten eingeschränkt. Damit ist eine gezielte Anlage von Dubletten mit den selben Verbindungen möglich.

Bei der Subscription eines Empfängers zu einer Inxmail-Liste wird automatisch die Tracking-Permission vergeben.

Die REST-API Verbindung unterstützt alle Lade-Methoden, womit sie in den Universalprozessen verwendet werden kann. Dafür wurden zwei weitere Lese-Urls, für alle Datensätze und für Änderungen von Datensätzen eingeführt. Außerdem kann das Lesen seitenweise erfolgen, wofür zwei weitere Parameter eingeführt wurden. Damit die Änderungsabfrage kontinuierlich auf der Datenbasis erfolgen kann, wurde die Schema-Angabe erweitert. Per UpdatedInfo kann ein Feld für die Änderungsinformation definiert werden. Dies muss entweder ein Datums- oder ein Integer-Feld sein.

Im CSV-Export kann ein zusätzlicher Datei-Kopf definiert werden. Die CSV-Export-Prozesse haben ein neues mehrzeiliges Feld „Datei Kopfdaten“. Der Inhalt wird den eigentlichen CSV-Daten vorangestellt und mit einem Zeilenumbruch abgeschlossen.

Die Sage CRM Verbindung kann Angebote und Aufträge mit Positionen speichern. Damit können diese Objekte im Universalprozess für geschachtelte Daten eingesetzt werden.

Ein neuer Prozess für CAS Firma nach Business Central Kunde ermöglich diese Synchronisation inklusive Neuanlage, bei der der Kunde mit dem vorhanden Kontakt verknüpft wird.

Die Transformation führt eine Datensatz-spezifische Fehlerbehandlung aus. Fehler brechen nicht mehr die gesamte Transformation ab, sondern führen die Fehlerbehandlung für den auslösenden Datensatz aus.

Die Serienbrief Verbindung hat einen neuen Parameter im Ausführungsschema. Mit „ParseInputColumnType“ wird der Inhalt nach seinem definierten Datentyp übersetzt. Aus „“ wird dann 0 für ein Zahlenfeld. Aber aus „312.23 Euro“ wird 0, wenn der Typ eine Zahl ist. Der Standardwert ist „deaktiviert“.

Die Salesforce Verbindung definiert ein Standard-Belegobjekt, welches über den Universalprozess synchronisiert werden kann.

Die Sage 100 Verbindung unterstützt abw. Rechnungsempfänger (VK) bzw. Rechnungsersteller (EK). Die zugehörigen Adressdaten „A2“ können ebenso übergeben werden. Für die Lieferanschrift wird der Ansprechpartner unterstützt.

Die CAS Verbindung kann eine neue SOAP-Bridge ansprechen. Diese wird über ein Paket im IIS eingefügt und braucht keine zusätzliche Konfiguration. Über diese Bridge kann ein Bulk-Import in CAS durchgeführt werden. Dazu stehen Prozesse auf der Basis von CAS, Sage b7, CSV und SQL zur Verfügung.

Abfrageplugins haben einen neuen Parameter „Seitenverarbeitung“. Mit diesem Parameter wird eine Seitenverarbeitung für die Quelldaten gestartet. Die Daten werden komplett gelesen und dann in 1000-Schritten transformiert, verarbeitet und gespeichert.

Der neue Prozessparamter „Initialsync für Konfigurationsänderungen“ ermöglicht eine einzelne gezielt Komplettverarbeitung im Fall einer geänderten Konfiguration, z.B. neue Feldabbildungen. Der bisherige Initialsync-Parameter hatte den Fokus auf verpassten Änderungen, wodurch unveränderte Datensätze auf der Basis der Datenabbildung nicht verarbeitet wurden.

Die Parameter wurden umstrukturiert, damit eine logische und ablauforientierte Übersicht entsteht.

Der Hotfolder-Prozess ermöglicht ein Upload für Sage CRM. Dafür wird die SData-Schnittstelle verwendet. Die zugehörige Url muss in der Verbindung eingestellt werden.

Die neue Eigenschaft „Kategorie“ an Prozessen und Datendiensten ermöglicht eine individuelle Strukturierung in der Administration (aktuell nur On-premises).

Ein globaler Fehler hat bisher immer die Fehlerbehandlung mit einer kompletten Wiederholung übersteuert. Dies wird jetzt nur noch ausgeführt, wenn die Fehlerbehandlung nicht auf „Ignorieren“ eingestellt ist.

9.1.2 Korrekturen

Die CSV-Prozesse haben keine Datensätze im Änderungsspeicher für Nachfolgeprozesse angelegt. Damit war keine universelle Prozesskette möglich.

Die interne XML-Verarbeitung verwendet jetzt einen Textreader. Damit werden Zeilenumbrüche CRLF nicht mehr auf LF reduziert. Dies war z.B. in Prozessparametern der Fall.

Die Beleg-Ressource in der Sage 100 Verbindung definiert eine Änderungsinformation im Datenschema. Damit kann dieses Objekt kontinuierlich in einer Änderungssynchronisation verwendet werden.

In Abfrageplugins war die Erzeugung von Datenabbildungen mit Zielen ohne ID-Rückgabe möglich, wodurch eine Datenabbildung mit leerem Ziel erzeugt wurde. Das konnte beim Nachladen des Zieles in einem späteren Lauf zum Fehler führen. Jetzt prüft der Prozess und springt raus bzw. führt die Übereinstimmungsregel erneut aus, damit ein vorher gespeicherter Schlüssel gesucht werden kann. Den leeren Wert kann auch der Komplementärprozess füllen.

Die Adresse-Kontokorrentversion in der Sage 100 Verbindung wurde nicht in der Änderungsinformation beachtet. Wenn die Adresse über das Kontokorrent First aktiviert wurde, wird die maximale Version für die Adresse verwendet. Bisher wurde das nur im Parameter gespeichert, aber nicht in UpdatedInfoColumn. Dadurch konnte ein Universalprozess eine alte Version erhalten.

Das Leeren von Datumsfeldern hat in den Verbindungen Sage b7, CAS, Mailchimp, Sage 100, REST-API, Infor CRM, Odoo und Zoho zu einem Konvertierungsfehler geführt.

9.2 Release 4.3.6 - 15.02.2022

9.2.1 Änderungen

Die Zoho Verbindung prüft das Schema Price und führt selbständig ein PUT statt POST aus. Damit kann das Schema Price auch in Universalprozessen genutzt werden.

Sage 100 Belegübertragung: Das Verhalten zu Ansprechpartnern und Adressfeldern wurde geändert. Der Zusatz für A0, A1 und A2 wird nicht mehr vom Ansprechpartner überschrieben. Im Belegrequestobjekt ist die Property „A1AdressNummer“ hinzugekommen. Hier kann nun optional eine dem Auftraggeber zugehörige abweichende Lieferanschrift mitgegeben werden. Wird nichts übermittelt bleibt die A1AdressNummer auf der Adresse des Auftraggebers stehen. Der Ansprechpartner der abweichenden Lieferanschrift muss immer in Bezug zur A1AdressNummer stehen.

Die Universal SDK-Prozesse wurden im Verhalten den Universalprozessen angeglichen bzgl. Rückschreiben, Bedingungen und Änderungsspeicher.

Es steht eine rudimentäre Verbindung zu EmarSys zur Verfügung. Diese kann Kontakte übertragen oder aktualisieren. Außerdem können Events getriggert werden.

Die REST Verbindung unterstützt das Anmeldeverfahren WSSE UsernameToken.

Die SDK-Verbindung hat generische Parameter für Zugangsdaten, damit diese den Passwortschutz nutzen können und nicht im Skript enthalten sein müssen.

Die REST Verbindung hat einen neuen Parameter „Immer alle Felder senden“. Damit wird im Update-Fall ein vollständiges Objekt versendet, statt ausschließlich ein Delta-Objekt.

Für Sage b7 werden Datensätze mit unbestimmter Datenabbildung wiederholt und nicht übersprungen. Unbestimmte Datenabbildungen liegen vor, wenn auf eine Insert-Nachricht noch keine Antwort zur Verfügung steht.

Für IDoc werden Datensätze mit unbestimmter Datenabbildung wiederholt und nicht übersprungen. Unbestimmte Datenabbildungen liegen vor, wenn auf eine Insert-Nachricht noch keine Antwort zur Verfügung steht.

Mit der Übersetzung „SisDataServiceDropdown“ = Y kann im ERP-Fokus für Sage CRM die Registerkartenleiste durch ein Dropdown-Menü ersetzt werden.

Es steht eine neue Transformation „HTML zu Text“ zur Verfügung.

Die Sage CRM Verbindung unterstützt geschachtelte Daten für das Kommunikationsobjekt. Die Teilnehmer (Comm_Link) werden als geschachtelte Liste behandelt. Zusätzlich wird der Organisator automatisch zu den Teilnehmern.

Die Zoho Verbindung unterstützt das Abfragen von gelöschten Daten. Über das Schema und einen spezifischen Prozess können gelöschte Daten abgefragt werden. Neben der ID beinhaltet die Antwort noch den Anzeigenamen des Datensatzes.

Die Email-Verbindung enthält das Feld FileContent in der MimeMessage. Dieses wird beim Lesen mit einer EML-Version in Base64 der Email gefüllt.

Die CAS Verbindung enthält das Feld FileContent im Objekt EMAILSTORE. Zugewiesene Daten werden als Email in CAS archiviert.

Es steht ein neuer Prozess „Universal Sync Prozess für Email“ zur Verfügung. Dieser Prozess basiert auf dem Universalprozess und verarbeitet eingehende Emails. Abweichend zum Basisprozess wird keine komplementäre Richtung oder Konflikte berücksichtigt. Außerdem können verarbeitete Emails gelöscht oder verschoben werden.

Im Schema der REST Verbindung kann per Wert „NestedList“ definiert werden, welche Unterliste das Alleinstellungsmerkmal bekommt. Diese Eindeutigkeit ist für Belegprozesse relevant.

Die Zoho Verbindung unterstützt das Schemaobjekt Photo. Der Prozess „Hotfolder nach Zoho Foto“ erlaubt den Upload von Fotos aus einem Verzeichnis. Modulname und Modul-ID müssen per Transformation bestimmt werden.

Die Zählung von lesenden Transaktionen wurde von eine physischen auf eine logische Anzahl umgestellt. Bisher haben die Verbindungen jedes Lesen gezählt. Jetzt wird das Lesen über die Prozess-Zusammenfassung gezählt und hat damit einen direkten Bezug zur Ausführungshistorie. Lesen zum Anreichern von Daten in der Verbindung werden nicht mehr extra gezählt.

Das Produkt-Schema in der Sage CRM Verbindung unterstützt einen Standardpreis für Standardpreisliste und Standardmengeneinheit.

Der Refresh-Token der Sage 200 Verbindung wird direkt in der Verbindung ermittelt.

Der neue Feldtyp „Array“ ermöglicht die Verarbeitung von Json-Arrays. Der String wird abhängig von der Verbindung als JArray verarbeitet.

Prozesse zu SQL Hauptobjekt legen keine parallelen Mappings mehr an und prüfen eine Änderungsinformation gegen die eigene Datenabbildung für die Feststellung der Synchronität.

In der SQL-Verbindung können für das Hauptobjekt optional ID-Felder angegeben werden. Durch die Abfrage ermittelte ID-Felder haben weiterhin Priorität.

Es steht eine neue Transformation „Formel ausführen“ zur Verfügung. Mittels Platzhalter kann eine Formel in SQL-Notation definiert werden. Das Resultat wird als Feld übernommen. Möglich sind dadurch mathematische Operationen oder auch Zeichenkettenoperationen.

9.2.2 Korrekturen

Die Erfolgsrückmeldung zu Sage 100 Belegübertragung war unvollständig, wodurch die Statistik in der Warteschlange unvollständig war.

Das Laden aller Datensätze in der Sage 100 Verbindung ignoriert den Parameter LAST_SYNC_VERSION. Dieser wurde standardmäßig aus den Universalprozessen übergeben, was dazu geführt hat, das statt allen Daten nur Änderungen geladen wurden.

Der API Endpunkt StartProcessDirect schreibt das Finish-Datum in den Warteschlangendatensatz.

Beim Speichern von Verbindungen hat die API keine Anreicherung mit Schemaobjekten zurückgeliefert.

Der Universalprozess für geschachtelte Daten hat den Source-Lock nicht aktualisiert. Dadurch konnten Datensatzsperren erhalten bleiben.

In der CRM-Verbindung wurde die Order und Quote ID bei der Aktualisierung von Positionen übergeben. Das hat im CRM eine Fehlermeldung ausgelöst.

Für CAS werden LinkTo und LinkFrom nicht übergeben, falls die Partner ID leer ist. Dies beugt einer Fehlermeldung vor.

Datendienste mit Berechtigungsgruppen wurden im ERP-Fokus nicht als Registerkarten dargestellt, die die Abfrage danach den leeren Fall nicht berücksichtigt hat.

Die Übernahme der transformierten Daten wurde intern korrigiert. Felder mit beliebiger Schachtelungstiefe wurden nicht berücksichtigt.

9.3 Release 4.3.7 - 31.03.2022

9.3.1 Änderungen

Die Datenabbildung beinhaltet bei geschachtelten Datensätzen den reinen Quelldatensatz und nicht mehr ausschließlich den Hauptdatensatz.

Der Universalprozess für geschachtelte Daten markiert alle vorhandenen Zielposition zum Löschen, wenn keine Übereinstimmungsregel für Positionen angegeben wurde.

Die Zoho CRM Verbindung wurde auf die API Version 2.1 umgestellt. Diese Umstellung erfordert eine Aktualisierung des Schemas und ggf. von Belegprozessen. Die Beleg-Positionen wurden mit einem anpassbaren Modul ersetzt, wodurch vorhandene Feldzuordnungen ungültig werden. Die Behandlung von Nachschlag-Feldern und Subforms hat sich auch geändert.

Die Fehlerbehandlung bei Business-Central-Kundenprozessen wurde angepasst. Fehler bei Business-Relations werden separat behandelt, damit eine gültige Datenabbildung erzeugt werden kann.

Die Zoho CRM Verbindung unterstützt ein neues Schema-Objekt users, womit Abfragen nach Benutzern möglich werden. Ggf. muss das Scope des Self-Clients ergänzt werden.

Die Zoho CRM Verbindung legt interne Auswahllisten für Auswahlfelder an. Diese können in der Transformation „Auswahlliste abbilden“ verwendet werden, um Text-Wert oder Text-ID Umwandlungen vorzunehmen.

Die Sage 100 Verbindung unterstützt Zubehörartikel bei der Belegerzeugung. Besitzt ein Artikel automatische Zubehörartikel werden diese entsprechend hinzugefügt. Gehören zu einem Artikel optionale Zubehörartikel, muss dessen Hinzufügen mittels des Positionrequestproperties „ForceOptionalZubehoerartikel“ erzwungen werden. Diese erzwungenen optionalen Zubehörartikelpositionen werden mit Menge = 0 dem Beleg hinzugefügt und müssen im weiteren Verlauf manuell im UI geändert werden.

Die Projektnummer kann EK-Positionen in der Sage 100 Verbindung zugeordnet werden.

In Prozessen mit geschachtelten Daten steht ein sekundärer Filter für Positionen zur Verfügung. Damit können Positionen für die weitere Verarbeitung ausgeschlossen werden.

Die neue Verbindung zur Microsoft Graph-API steht zur Verfügung. Dies beinhaltet auch Prozesse und Vorlagen für die Synchronisation mit Zoho CRM Meetings.

Die Zoho CRM Verbindung wurde für die Behandlung von Serienterminen erweitert.

In der Sage CRM Verbindung kann per Einstellung kann festgelegt werden, ob die per Typ ausgewählten Adressen an Firma und Person auch mehrfach verwendet werden können. Dies kann beim Schreiben allerdings zu einem Konflikt führen und muss genau geprüft werden.

Für die SAP IDoc Verbindung brauchen Multitexte im Schema nur noch ein MaxOccurs von 1 und das Feld TDLINE. Damit stehen Multitexte auch am Produkt zur Verfügung.

Die neue Transformation „Datenabbildung abfragen“ steht zur Verfügung. Es wird ein Prozess ausgewählt und festgelegt, ob die Suche per Source oder Target erfolgt. Der geparste Suchstring wird dann direkt verglichen und als Resultat werden SourceId, TargetId und TargetIsDeleted bereitgestellt. Häufige Abfragen an das Zielsystem nach Datensätzen, die verknüpft werden sollen, können damit reduziert werden.

Die On-premises Version beinhaltet eine aktive Benachrichtigung per Email zu neuen Versionen.

Die CAS Verbindung unterscheidet nach Datum und Datum-Uhrzeit Feldern anhand der percision-Angabe. Datumsfelder werden nicht in nach lokal beim Lesen oder nach UTC beim Schreiben konvertiert. Die Übergabe beim Schreiben erfolgt ohne Uhrzeit.

9.3.2 Korrekturen

Bei der direkten Verwendung der Syncler-API per Client-ID konnte bei bestimmten Funktionen ein Unauthorized access ausgelöst werden, wenn Protokolle geschrieben werden sollten.

Der Standard-Preis für Sage CRM Produkte konnte einen Fehler auslösen, wenn das entsprechende Schema nicht vorhanden war. Dies wurde mit einer zusätzlichen Prüfung korrigiert.

Beim Schreiben von Daten mit der Business-Central-Verbindung wird der eingestellte Mandant per Url übergeben. Die Definition einer Standard-Firma entfällt.

Die API-Aufrufe bei Business-Central können case-sensitiv sein, wodurch die Prozesse zur Kundenanlage den neuen Kontakt nicht behandeln konnten.

In einigen Prozessen wurde vor dem Rückschreiben nicht geprüft, ob die Sync-Felder auch im Schema enthalten sind. Dies konnte zu einem kontinuierlichen „leeren“ Rückschreiben führen, da das neue Änderungsdatum größer als der neue Grenzwert ist.

Das Abfrufen von Daten mit der SQL-Verbindung und der SQL-Bridge wurde korrigiert. Die abgefragten Daten wurden nicht der Verarbeitung in Prozessen übergeben. Das betrifft ausschließlich das Hauptobjekt.

9.4 Release 4.3.8 - 12.05.2022

9.4.1 Änderungen

Die Sage 100 API unterstützt das Objekt VkBelegeStuecklisten.

Die Sage 100 Verbindung speichert die Url des Objektes. Damit sollen Fehlern bei neuen Objekten vorgebeugt werden.

Die Sage 100 Verbindung unterstützt die OfficeLine Version 6.2 nicht länger.

Die Transformation „Daten abfragen“ unterstützt untergeordnete Strukturen im Objekt. Diese werden mit einem Doppelpunkt notiert in das Ergebnis übernommen.

Im Syncler-Administrator wurde das Neu-Menü für Transformationen strukturiert.

Es wurde eine zusätzliche Aktivitätskontrolle für die Warteschlangen-Verarbeitung eingeführt. Diese kann ggf. die Verarbeitung reaktivieren, falls diese in Folge eines Fehlers nicht mehr aktiv sein sollte.

Für Datendienste steht der Parameter „Format für Datums- und Uhrzeitfilter“ zur Verfügung. Damit kann ein individuelles Format für Filterabfragen definiert werden, falls dieses die Datenquelle erfordert.

Die Serienbrief-Verbindung kann Serienbrief-Felder in Word-Vorlagen mit strukturiertem Inhalt aus Html, Rtf oder Word füllen. Das Feld muss dafür den Präfix Html: Rtf: oder Docx: tragen.

Der Universal Sync Prozess für Emails lässt mehrfach zugeordnete Datenabbildungen zu. Da diese Daten nur unidirektional übertragen werden können, kann kein Konflikt entstehen. Damit können verschiedene Emails dem identischen Ziel zugeordnet werden. Z.B. Emails zu einem Ticket. Die Datenabbildungen sind in Folgeprozessen oder zur Dublettenvermeidung vorhanden.

Die Transformation „Parameter erzeugen“ überschreibt keine Quelldaten, falls der Parameter keinen Wert hat. Damit können Parameter für Feldzuordnungen definiert werden, welche auf Felder aus Vorgängerprozessen verweisen. Siehe Abläufe.

Das Universal Ablauf Plugin steht zur Verfügung. Damit können komplexere Abläufe definiert werden. Siehe *Ablaufbasierte Prozesse*

Der Platzhalter „#FlowFilter#“ kann in Abfrageprozessen eingefügt werden. Damit kann die Abfrage aus Abläufen heraus individualisiert werden.

Die Standardwerte im Wartungsprozess wurden auf 14 Tage reduziert.

Für die Abläufe stehen zwei weitere Prozesse zur Verfügung. Mit „Daten lesen“ und „Geschachtelte Daten lesen“ können Daten ohne eine Zielverarbeitung initial für einen Ablauf gelesen werden.

Die Email Verbindung bereitet den HtmlBody als HtmlBodyInlineImages auf, damit ContentId durch Filename ersetzt wird und dieser so z.B. direkt im Sage CRM verwendet werden kann. Die Attachments erhalten das Feld ContentId, damit zwischen Inline-Bild und Anhang unterschieden werden kann.

Der neue Prozess „Email Anhänge nach Sage CRM“ führt die gesammelt Ablage von Anhängen durch. Libr_IsInlineImage, Libr_FileName und Libr_FileSize werden dabei automatisch gesetzt. Weitere Felder können per Feldzuordnung gefüllt werden.

Der neue Feldtyp „Objekt“ ermöglich die Übergabe eines Json-Strings, der dann als Objekt ausgewertet wird.

Die Email Verbindung enthält Felder für einen Standardabsender. Dieser wird bei Emailversand aus Abläufen verwendet.

Es steht eine neue Verbindung zu Maileon zur Verfügung. Siehe *Maileon*

Es wurden die Voraussetzungen für abschließende Fehler ohne Wiederholung geschaffen. Diese produzieren eine Warnung werden aber nicht weiter wiederholt. Das wird bereits in der Maileon-Verbindung genutzt, wo bestimmte Fehler gar nicht aufgelöst werden können.

Die Suche nach Email, Phone und Word in der Zoho CRM Verbindung wurde angepasst. Die Kriteriumsnotation wird weiterhin verwendet, intern aber in eine gezielte Abfrage umgeformt. Die Kriteriumssuche nach diesen Feldern wird von Zoho CRM nicht mehr unterstützt.

Die Zoho CRM Verbindung unterstützt die Nachfolger-Beziehung Campaigns zu Contacts, welche per Nachfolgerprozessen ausgewertet werden kann.

Der Universal Lösch Prozess wurde hinzugefügt. Dieser kann gezielt per Datenabbildung oder Übereinstimmungssuche einen Datensatz im Ziel löschen. Optional können auch die Datenabbildungen entfernt werden.

Der Universal Sync Prozess erzeugt Daten im Änderungsspeicher, falls diese keine ID-Werte haben. Das Lesen muss durch die Verbindung unterstützt werden.

Alle Sync-Prozesse haben einen neuen Parameter für das Abschalten einer Datensatzkopie in den Datenabbildungen. Damit kann Speicherplatz gespart werden, wenn z.B. Prozesse nur unidirektional übertragen. Das Abschalten per Servereinstellungen kann damit nicht umgangen werden. Dies hat Priorität.

Der On-premises-Setup konfiguriert den Dienst für Neustarten im Fehlerfall.

Zur Syncler Bridge kann in den Verbindungen ein Timeout definiert werden.

9.4.2 Korrekturen

Alle Abfrage-Prozess mit Ziel Sage CRM wurden korrigiert. Bei der Übertragung von Personen wurde die Firma nicht korrekt ermittelt, wodurch das Setzen der Primärperson fehlgeschlagen ist.

Die CSV-Verbindung wurde ergänzt, damit fehlende Spalten übersprungen werden.

In der Sage Bäurer b7 Verbindung werden die Felder aufnr und rg_nr im Schema nicht mehr als Auswahl geführt, sondern als String gespeichert. Damit findet das Abrufen von AWL dazu nicht statt, was i.d.R. in ein Timeout läuft.

Datensatzsperren werden für Datensätze ohne IDs nicht mehr angelegt.

Die Zoho CRM Verbindung hat für Activities ein doppeltes Tag-Objekt in das Schema eingefügt. Dies hat in der Verarbeitung zu Fehlern geführt.

In der Sage Bäurer b7 Verbindung wurde die Verarbeitung von Request-Antworten nicht auf den Objekttyp geprüft. Damit konnte eine Verarbeitung durch einen fremden Prozess erfolgen, falls parallel verschiedene Requests aufgeführt wurden.

Die Graph Verbindung konnte ohne ein Schema nicht neu angelegt werden.

Die Sprache des Syncler-Administrators richtet sich nach den Einstellungen und nicht nach der Umgebung.

9.5 Release 4.3.9 - 28.06.2022

9.5.1 Änderungen

Die erneute Eingabe eines Zoho Grant-Token aktualisiert auch aktuell verwendete Access-Token, damit geänderte Scopes sich direkt auswirken und nicht erst nach dem Refresh des Access-Tokens.

Sage 200 wurde in Infoniqa One 200 umbenannt.

Prozessvorlagen für Sage 100 und Microsoft 365 wurden überarbeitet.

Das Datenbanksystem MariaDB wird als Systemdatenbank und in der SQL-Verbindung unterstützt. Für die Migration von Microsoft SQL-Server steht der Prozess „Syncler Datenbank Kopierprozess“ zur Verfügung. Mit diesem Prozess kann die aktuelle Datenbank in eine bereits vorhandene Datenbank kopiert werden. ID-Werte bleiben dabei erhalten und der Prozess kann auch kontinuierlich verwendet werden.

Für Abfragen durch die REST API Verbindung wird die Url auch durch das Objekt geparst. Dadurch sind neben #name#, #idfield# und #id# auch weitere Parameter steuerbar.

Die Verbindungstypen SQL, CSV und Email stehen nicht mehr automatisch zur Verfügung, sondern müssen explizit bestellt werden.

Die aktuelle Schnittstelle zur Dynamics Business Central unterstützt keine Unterstriche in Feldbezeichnungen mehr. Damit einen flexible Unterstützung gewährleistet ist, prüfen die Verbindung und die spezifischen Prozesse das ermittelte Datenschema und passen ihre Abfragen dementsprechend an.

Im Syncler-Administrator Testlauf einer Transformation wird der Offset zu Datumswerten ausgegeben.

Die Meldungen zu übersprungenen Datensätzen wurden für eine bessere Nachvollziehbarkeit angepasst.

Für Bulk-Prozesse wurde eine zusätzliche parallele Verarbeitung eingeführt, um die Gesamtdauer zu optimieren. Die Nachbearbeitung (Resultat, Datenabbildung, Zurückschreiben) des Resulats wird parallel zur Datensatzverarbeitung (Datenabbildung suchen, Ziel ermitteln) ausgeführt.

Die neue Transformation „Dokument konvertieren“ ermöglich die Umwandlung bestimmter Typen. Die Typen HTML, TXT, RTF, XML und DOCX können als Eingabetyp definiert werden. Die Typen HTML, TXT, RTF, XML, DOCX, PDF und XPS können als Ausgabetyp definiert werden. Die binären Typen DOCX, PDF und XPS werden als Base64 erwartet und geliefert. Die Typen HTML, TXT und RTF werden als Zeichenkette erwartet und geliefert.

Es steht ein neuer Verbindungstyp „Tabellenkalkulation“ zur Verfügung. Dieser kann ähnlich wie der Seriendruck eingerichtet und eingesetzt werden. Als Vorlagen können XLSX Dateien verwendet werden. Platzhalter werden mit der #..# Notation definiert. Ranges, Bilder und Html-Inhalte werden wie im Seriendruck definiert. Die Ausgabe kann auch per PDF erfolgen.

Die Serienbrief Verbindung legt das Zielverzeichnis an, falls es noch nicht vorhanden ist.

Die Verbindungsherstellung durch die SQL Verbindung wurde auf eine lokale Verbindung statt einer globalen umgestellt. Damit werden Fehler aus paralleler Verarbeitung vermieden.

9.5.2 Korrekturen

Die Sortierung von Listen im Syncler-Administrator wurde korrigiert.

Das Parsing von Json-Angaben für Picture-Felder im Seriendruck wurde korrigiert.

Wenn eine Abfrage über die REST API Verbindung nur ein Objekt ohne typische Antwortstruktur (data, success, ...) liefert, wird dieses Objekt als Ergebnis verwendet.

In der CAS-Verbindung wird vor dem Speichern der Positionen zu Verkaufschancen oder Belegen geprüft, ob eine gültige Etag vorliegt und ggf. wird diese angefordert. Durch die Konstellation unveränderter Datensatz und geänderte Positionen konnte diese Fehlersituation eintreten.

Die Transformation „Typ konvertieren“ erzeugt Ergebnisfelder vom Typ Zeichenkette. Dies entspricht auch dem tatsächlichem Ergebnis dieser Funktion.

Die EmarSys-Verbindung hat keine ID zu angelegten Kontakten zurückgeliefert, wodurch dieser Datensatz fälschlich als „übersprungen“ interpretiert wurde.

Die Sage b7 Verbindung wurde korrigiert. Leere Werte zu Feldern werden nicht mehr als leere Zeichenkette, sondern als NULL übergeben. Außerdem wurd die Verabreitung der Antwort zu einer neu angelegten Person oder Anschrift korrigiert.

9.6 Release 4.4.0 - 18.08.2022

9.6.1 Änderungen

Die API hat einen Endpunkt für die Ausführung der Tabellenkalkulation erhalten (RunSpreadsheet). Die Verwendung ist identisch zum API Endpunkt für Serienbriefe.

Konfigurationsfehler für Serienbriefe und Tabellenkalkulation in Prozessen lösen keine Fehlerwiederholung aus, da der Fehler nur durch manuelle Anpassung behoben werden kann.

Das Format XLS wurde für die Verbindung Tabellenkalkulation ergänzt.

Die On-Premises Lizenz wurde angepasst. Für den Lizenztyp Basic müssen verfügbare Verbindungstypen definiert werden. Der Lizenztyp Extended ist nicht betroffen.

Der Schritt „Datendienst ausführen“ im Universal Ablauf kann per Checkbox das Resultat als Anhang oder Seriendruck mit einer Emailvorlage ausgeben.

Die CAS Fehlermeldung „Data object not found“ wird nicht als Fehler, sondern als leere Antwort interpretiert. Damit kann das Ziel als gelöscht erkannt werden.

Die Anmeldung an der Microsoft Graph API kann optional auch mit Benutzernamen und Passwort erfolgen.

Die Microsoft Graph API Verbindung unterstützt das Objekt OnlineMeeting.

In der Tabellenkalkulation wird die Zelle bei gefundenem Platzhalter für Textdaten nicht komplett ersetzt, sondern nur der Platzhalter. Damit können auch mehrere Platzhalter in einer Zelle verwendet werden. Für Zahlen- und Datumswerte wird weiterhin der Zellenwert ersetzt, damit die Zellenformatierung eingehalten werden kann.

In der Microsoft Graph API Verbindung können die Benutzer optional auch in einem Feld mit Semikolon getrennt aufgelistet werden, statt eine Abfrage zu verwenden. Als Wert ist die Emailadresse zu verwenden.

In der REST API Verbindung kann neben der automatischen Schemaerzeugung per Beispielabfrage die Feld- und Listen-Eigenschaften des Objekts auch mit der properties-Eigenschaft definiert werden. In diesem Fall darf keine Beispiel-ID angegeben werden. Die Basis der Definition folgt der json-schema.org Konvention, wie sie auch in der SDK-Verbindung verwendet wird.

Das Schema für die Tabellenkalkulation Verbindung wurde um die Gruppe AppendSheets erweitert. Hier kann eine beliebig lange Liste von weiteren Quelldokumenten angegeben werden. Die Tabellenblätter dieser Dateien werden dem Hauptobjekt hinzugefügt, bevor der Merge ausgeführt wird. Die Namen der Tabellenblätter werden geprüft und ggf. angepasst, da Dopplungen nicht zulässig sind.

Die Meldung zum Überspringen eines Datensatzes in der Übereinstimmungsregel wurde von Debug auf Rückmeldung geändert.

Die Prozesse zu Sage b7 und IDoc Verbindungen haben einen neuen Parameter erhalten, worüber Quelldaten aus dem Änderungsspeicher gelöscht werden können, sollten diese durch den zweiten Filter ausgeschlossen werden.

9.6.2 Korrekturen

Bei Serienbriefen und Tabellenkalkulation standen Felder in Listen, die durch eine Transformation erzeugt wurden, nicht in den Daten zur Verfügung, da diese auf oberster Ebene eingefügt wurden. Diese Felder werden jetzt in das Listenobjekt kopiert und überschreiben ggf. bereits vorhandene gleichnamige Felder.

In der Sage 100 Verbindung wurde bei der Abfrage von geänderten abweichenden Lieferanschriften das Feld Kto nicht gefüllt. In Einzel- oder Komplettabfragen stand das Feld bereits zur Verfügung.

Beim Anlegen eines Empfängers in der Kopplung CAS nach Cleverreach wird nicht mehr das Aktualisierungsdatum des Empfängers geprüft. Dieser Wert entspricht dem Registrierungsdatum und hat nur für die entgegengesetzte Richtung eine Relevanz.

In der Kopplung CAS nach Cleverreach lässt eine Warnung im Empfängerprozess den Verteilerprozess nicht mehr stoppen.

In der Microsoft Business Central Verbindung wird bei der Suche nach Feldern mit Änderungsinformationen die Schreibweise nicht mehr unterschieden.

In der Salesforce Verbindung werden Felder vom Typ Numerisch oder Währung automatisch in die Punktschreibweise vor dem Schreiben überführt.

Passwörter werden in der API auch für Prozessparameter gecovert.

Das Füllen von Listen im Email-Seriendruck wurde korrigiert.

Die generische Erzeugung eines Json-Objektes wurde korrigiert. Leere Felder in Unterobjekten werden bei Aktualisierungen nicht erneut übertragen.

Die CAS Verbindung für SmartWe erfordert einen TYPE für Berechtigungen.

Eine leere Quelle in der Transformation „Dokument konvertieren“ hat zu einer Fehlermeldung geführt.

Der REST API Verbindung sucht die ID von neu angelegten Daten zuerst auf der obersten Feldebene und danach erst in Unterstrukturen. Damit sollen falsche Zuordnung durch identische Namen unterbunden werden.

Multitext-Felder in der IDoc Verbindung haben zu einem Fehler geführt, wenn die erste Zeile mit einem Null-Wert übergeben wurde.

Die REST API Verbindung hat untergeordnete Listen mit unbenannten Werten als JValue übernommen, was zu einem Fehler in der Verarbeitung geführt hat.

Die Serienbrief- und Tabellenkalkulationsverbindung ignorieren nicht XML-zulässige Zeichen in den übergebenen Daten. Dies soll die Erzeugung von korrupten Resultaten verhindern.

9.7 Release 4.4.1 - 22.09.2022

9.7.1 Änderungen

Die Erzeugung eines Delta-JSON auf der Basis von geänderten Felder eines Objektes wurde angepasst. Ungeänderte Pflichtfelder und Unterobjekte mit ungeänderten Pflichtfeldern werden nicht übersprungen, sondern in das Resultat übernommen. Abhängig vom Zielsystem verhindert das Fehlermeldungen durch fehlende Pflichtfelder.

Das Schema der Zoho Verbindung wurde u.a. für die Erstellung eines Delta-JSON angepasst. Die Felder „type“ und „participant“ des Objektes „Participants“ wurden als Pflichtfelder definiert. Das Feld „id“ des Objektes „Participants“ wurde als ID-Feld definiert. Das Feld „percentage“ des Objektes „Linetax“ wurde als Pflichtfeld definiert. Der Feldtyp des Feldes „Remind_At“ des Objektes „Events“ wurde auf Datetime geändert, damit eine automatische Formatierung ausgeführt werden kann.

Der neue Prozesstyp „SQL Abfrage nach Auswahllisten“ kann interne Auswahllisten für Transformationen auf der Basis von SQL-Abfragen erzeugen.

Bei der Erzeugung einer Terminserie in Zoho werden die Termine direkt über die $u_id abgefragt. Die erzeugten Einträge im Änderungsspeicher werden um das Anfangsdatum ergänzt, damit die Zuordnung von Terminvorkommen das Ziel nicht erneut in Zoho abfragen muss. Dies hatte zu wiederholten Abfragen bei historischen Serien geführt, die das API-Limit unnötig belasten.

Bei der Erzeugung einer Terminserie in Graph wird das Startdatum und der Benutzer am erzeugten Eintrag im Änderungsspeicher hinterlegt. Damit kann das Vorkommen für die Zuordnung zum Quelldatensatz effizienter ermittelt werden.

Die Verarbeitung von JSON in der CAS Verbindung wurde intern zur Optimierung umgestellt.

Die Graph Verbindung wurde angepasst und erweitert.

- Das automatische Parsing von Datumswerten wurde deaktiviert. Dies hat in der Verarbeitung zu nicht ISO-konformen Datumsangaben und damit API-Fehlern geführt. Datumswerte werden jetzt als Zeichenketten verarbeitet und müssen ggf. manuell konvertiert werden (z.B. für Berechnungen).
- Per Verbindungsparameter kann eine bevorzugte Zeitzone eingestellt werden. Datumswerte werden bei Abfragen dann in dieser Zeitzone zurückgemeldet und müssen nicht extra umgerechnet werden. Ohne eine angegebene Zeitzone werden Datumswerte in UTC bereitgestellt.
- Das Verbindungsschema wurde um die Objekte „TodoTask“ und „Contact“ erweitert.
- Die Schemaobjekte werden um das Feld „isDeleted“ (bool) ergänzt. Bei Delta-Abfragen wird dieses Feld gefüllt, damit gezielt gelöschte Datensätze in Prozessfiltern berücksichtigt werden können.
- Die Schemaobjekte „TodoTask“ und „Contact“ erhalten zusätzlich das Feld „user“, damit der zugeordnete Benutzer per Feldzuordnung festgelegt werden kann.
- Die Abfrage von Vorkommen zu einer erzeugten Terminserie erfolgt nicht mehr über das Zeitfenster, sondern über Start- und Enddatum der Serie. Nur bei Serien ohne Enddatum wird das Ende des Zeitfensters verwendet.
- Bei Request-Fehlern wird die Url in die Fehlermeldung aufgenommen, da diese sonst nur als Debug-Nachricht ausgegeben wird.
- Fehlermeldungen zu ErrorItemNotFound, ResourceNotFound oder MailboxNotEnabledForRESTAPI erzeugen nur eine Warnung. Damit soll der Abbruch mit globalen Fehler unterbunden werden, wenn einzelne Benutzer nicht abgefragt werden können.

Die Zoho-Graph Vorlagen wurden überarbeitet, um den genannten Änderungen an der Graph Verbindung gerecht zu werden. Dies optimiert die Prozesse, da Skript-Transformationen entfallen können.

Die Sage CRM Verbindung wurde angepasst und erweitert.

- Das Schema „comm_link“ wird mit dem Feld „comm_organizer“ angereichert, wenn Kommunikationen abgefragt werden. Damit können diese ggf. für die Synchronisation über die Graph Verbindung ausgefiltert werden.
- Wenn die Objekte „Quotes“, „Orders“, „Communication“ und „ProjectOrders“ gelöscht werden, werden auch die Positionsdatensätze gelöscht.
- Als gelöscht markierte Positionen werden bei der Verarbeitung nicht übersprungen, wenn sie keine Änderung enthalten.
- Die Abfrage nach geänderten UserContacts prüft automatisch auch die Tabellen „Person“, „Address“ und „Company“ über die Sicht „vSearchListPerson“ auf Änderungen.
- Der Upload wurde für die Version 2022 R2 erweitert. Es kann über das Feld „libr_filepath“ ein Zielordner im Library-Verzeichnis definiert werden.
- Die Zeichenfolgen /* und */ in Where-Bedingungen werden ersetzt, damit die Abfragen durch das CRM nicht abgelehnt werden.

Die Syncler API enthält die neuen Endpunkte „GraphRefreshToken“ und „DeleteGraphRefreshToken“. Diese können für die verschlüsselte Speicherung eines Benutzer-bezogenen Refresh-Tokens genutzt werden. Da die Graph API für den Zugriff auf „TodoTask“ nur delegierte und keine Anwendungsberechtigungen unterstützt, können so Benutzerzustimmungen für das Lesen und Schreiben von Aufgaben gespeichert und fortlaufend verwendet werden. Die Implementierung des Zustimmungsprozesses ist abhängig vom Endsystem und steht in Sage CRM als neue Registerkarte in „Mein-CRM“ zur Verfügung. Der Zugriff auf „TodoTask“ ist automatisch gegeben, wenn die Verbindung mit einer Benutzeranmeldung eingerichtet wird. Allerdings beschränkt sich der Zugriff dann auf diesen Benutzer.

Die interne Verarbeitung von Datensatzsperren wurde angepasst, damit auch Zeichenketten als Änderungsdatum interpretiert werden können.

Es stehen Synchronisationsprozesse mit Delta-Funktion für die Kombination Sage CRM und Microsoft Graph API zur Verfügung. Die unterstützen Objekte sind Termine, Aufgaben und Kontakte. Für dieses Szenario wurden auch Vorlagen hinzugefügt.

Die Zoho-Graph Prozesse wurden angepasst. Die Änderung einer Serie erzeugt im Zielsystem eine neue Serie und löscht den Vorgänger. Der Parameter Monat wurde für Zoho-Serien mit dem Typ „Jährlich“ nicht bereitgestellt. Die Prozesse wurden um die Unterstützung von parallelen Datenabbildungen erweitert. Damit können neue Prozesse aus Vorlagen die Datenabbildungen vorhandener Prozesse weiterverwenden.

Die Basis für Abfrageprozesse wurde erweitert. Sobald Datenabbildungen verwendet werden und das Ziel nicht gefunden wird, wird die Abbildung als gelöscht markiert und das Fehler-Rückschreiben ausgeführt, sowie der Datensatz übersprungen.

Die Antworteigenschaft „list“ wird von der REST API Verbindung für das Resultat verwendet.

9.7.2 Korrekturen

Das Löschen von Verbindungen und Prozessen löscht auch zugeordnete Parameter aus der Datenbank.

In der Graph Verbindung hat eine Einzel-Abfrage durch einen Delta-Prozess zu einer fehlerhaften Abfrage geführt.

Im Syncler-Administrator konnte das Löschen von Feldzuordnungen zu Positionsdaten einen Fehler erzeugen.

Die Transformation „Dokument konvertieren“ wurde angepasst. Bei einer Konvertierung in das Ausgabeformat „Text“ wurden vertikale Tabulatoren erzeugt. Dies hat zu internen Verarbeitungsfehlern geführt, da diese Zeichen in XML nicht zulässig sind.

Die CAS Bulk-Übertragung hat Verknüpfungen des ersten Datensatzes für alle Datensätze der aktuellen Übertragung genutzt.

Die Antwortabfrage zur Sage 100 Belegerzeugung hat mit der Version 9.0.x ungültige Header enthalten, die zu einem Fehler geführt haben. Der Beleg wurde dennoch korrekt angelegt.

Das Laden von Wiederholungen, Nachfolgern und Fehlerwiederholungen durch die CAS Verbindung hat eine fehlerhafte Abfrage erzeugt.

9.8 Release 4.4.2 - 11.11.2022

9.8.1 Änderungen

Die Graph-Verbindung:

- Die Filter-Abfragen in Feldnotation kann um „user“ ergänzt werden, damit die Abfrage bei vielen Benutzern beschleunigt wird.
- Benutzer mit leerer Emailadresse werden bei der Verarbeitung übersprungen, da diese zu fehlerhaften Abfragen führen.
- Die Warnungen bei Abfragen wurden in Debug-Meldungen umgewandelt. (ErrorItemNotFound, ResourceNotFound, MailboxNotEnabledForRESTAPI)
- Per Parameter kann der Client als Public-Client definiert werden. Dies ist für das Zustimmungsverfahren relevant.
- Es wurden zusätzliche Fehlermeldungen für Hintergrundabfragen eingefügt, damit Fehlerquellen leichter gefunden werden.
- Die Benutzerauswahl kann durch die Angabe eines Gruppennamens getroffen werden. Benutzer, die zur Gruppe hinzugefügt werden, werden dann automatisch synchronisiert. Diese Funktion setzt zusätzliche Berechtigungen voraus.

Prozess „SQL Abfrage nach Auswahllisten“:

- Es kann eine beliebige Zielverbindung ausgewählt werden. Diese wird dann der Auswahlliste zugeordnet und in deren Prozessen kann die Auswahlliste dann verwendet werden.

Die Abläufe unterstützen Abfrage-Prozesse und es stehen auch Prozesse für das Lesen von Abfragen zur Verfügung. Dafür müssen Abfragen eine Identität für jeden Datensatzes erzeugen. Dies geschieht abgestuft. Sollte eine ID bekannt sein, wird diese verwendet. Sollte ein Änderungsdatensatz zugeordnet sein, wird dessen GUID verwendet. Ansonsten wird eine temporäre GUID erzeugt.

Siehe *Beispiele für den Einsatz von Abläufen*

Die globale Konfiguration für On-premises hat einen neuen Parameter „Erweiterte Fehlermeldung“ erhalten, der einen StackTrace in Fehlermeldungen aktivieren kann.

In Abläufen können in der Prozessausführung mit „Felder kopieren“ über eine Feldnotation mit Platzhaltern (z.B. GUID|:|**#GGUID#**|;|) Daten aus dem Zieldatensatz in den zwischengespeicherten Quelldatensatz übernommen werden. Diese stehen dann in folgenden Schritten zur Verfügung, wenn diese ebenfalls die zwischengespeicherten Quelldaten verwenden.

Die Fehlermeldungen von Zoho CRM werden interpretiert und übersetzt zurückgemeldet.

Die REST-Verbindung:

- Mit einem Parameter kann das Datumsformat für Json-Daten definiert werden, welches für Lesen und Schreiben genutzt wird.
- Das Abfragen von Änderungen ersetzt auch die Platzhalter „LastDatetime“ und „LastVersion“. Bisher wurden nur „LAST_SYNC_DATE“ und „LAST_SYNC_VERSION“ ersetzt.
- Einzelabfragen ohne ID-Platzhalter werden nicht ausgeführt.

Der SMTP-Versand in On-premises Installationen und die Email-Verbindung unterstützen das OAuth 2.0 Verfahren für Microsoft 365. Die Unterscheidung erfolgt durch die Angabe einer Client-ID.

Siehe *Das OAuth 2.0 Verfahren (Draft)*

Die generische Aktualisierung des Quellobjektes:

Dieses Verfahren wurde vereinheitlicht und in verschiedenen Prozessen ergänzt. Es werden zwei Verfahren unterschieden, für reproduzierbare und für nicht reproduzierbare Datensätze. Außerdem können für die Fälle Erfolgreich, Fehlerhaft und Übersprungen individuelle Daten aktualisiert werden. Bei dem Verfahren für nicht reproduzierbare Datensätze wird ein Schema und eine Suchbedingung für die Aktualisierung definiert. Bei folgenden Prozessen wurde die Funktion ergänzt.

- Alle Abfrage-Prozesse mit beschreibbarer Quelle
- Prozesse zur Kombination aus CAS und Sage 100
- Prozesse zur Kombination aus CAS und Business Central
- Prozesse zur Kombination aus Salesforce und Sage 100
- Prozesse zur Kombination aus Salesforce und IDoc
- Prozesse zur Kombination aus Sage CRM und Sage 100
- Prozesse zur Kombination aus Zoho CRM und Sage 100

Die HotFolder-Prozesse können einen Unterordner zum Basisordner aus der HotFolder-Verbindung definieren.

Die Seriendruck-Verbindung hat im Schema 4 neue Parameter für die Ausführungssteuerung erhalten.

- Durch „RemoveEmptyRange“ (Bool) werden Bereiche ohne gefüllte Felder komplett entfernt. Diese Option führt bei Dokumenten ohne Seriendruckfelder zum Leeren des Dokuments.
- Um das zu verhindern kann ein „BaseRangeName“ definiert werden, der dann mittels RangeStart und RangeEnd das Dokument umschließt.
- Mit „RemoveEmptyTableRows“ werden Zeilen mit Feldern ohne Quelldaten komplett entfernt, unabhängig von sonstigen Inhalten. Diese Funktion ist jetzt nicht länger standardmäßig aktiviert.
- Mit „RemoveEmptyTables“ werden Tabellen mit Feldern ohne Quelldaten komplett entfernt, unabhängig von sonstigen Inhalten. Diese Funktion ist jetzt nicht länger standardmäßig aktiviert.

Bei der Verarbeitung von Bildern mit dem Präfix „Picture:“ kann im Json auch die Breite in Bildpunkten angegeben werden. Die Grafik wird dann entsprechend skaliert. Die unterstützen Eigenschaften im Json sind „FileMethod“, „File“ und „Width“.

Im Syncler-Administrator kann eine Adhoc-Ausführung ausgehend vom Änderungsspeicher gestartet werden. In der Transformation „Werte abbilden“ wurde an den Abbildungen eine Aktualisierungsfunktion ergänzt.

Es gibt neue Vorlagen für die Synchronisation zwischen SpiceCRM und Sage 100. Das SpiceCRM wird dabei über die REST-Verbindung angebunden.

9.8.2 Korrekturen

Synchronisation von Microsoft 365 nach Zoho CRM:

- Der Prozess prüft auf ganztägige Termine und passt die Endzeit für Zoho CRM an, da dort 23:59 Uhr erwartet wird.

Die Vorlage wurde ebenfalls angepasst.

- Die Organizer-ID wird mit dem zweiten Filter auf ungleich leer geprüft.
- Das Setzen der Benachrichtigungsfunktion wurde entfernt.
- Die Positionsübereinstimmung sucht nach Email, Id oder Participant.

Syncler-Administrator:

- In der Transformation „Datenabbildung abfragen“ wurde das Ändern des Prozesses nicht als Änderung erkannt.

Die Fehlerbehandlung in Abläufen wurde korrigiert.

- Der ursprüngliche Parameter im Warteschlangendatensatz wurde umbenannt, wodurch nur eine einmalige Wiederholung möglich war.
- Wiederholte Datensätze wurden nicht in die zwischengespeicherten Listen aufgenommen, wodurch diese für Folgeschritte nicht zur Verfügung standen.
- Ein Prozess mit zwischengespeicherten Daten wurde nur ausgeführt, wenn der Zwischenspeicher Einträge enthielt. Die Fehlerbehandlung wurde ohne Einträge nicht ausgeführt.
- Nur manuell ausgeführte Prozesse werden bei erfolgreichem Resultat in der direkten Fehlerwiederholung übersprungen.

Die Nachrichtenausgabe durch eine Verbindung wurde in der Transformation „Abfrage ausführen“ ergänzt.

Die Sage 100-Verbindung:

- Die Antwort zu neu angelegten Kontokorrenten in Kombination mit einer Adresse hat nur die Versionsnummer der Adresse zurückgemeldet. Das hatte Einfluss auf die Änderungsprüfung mittels Datenabbildung.

In Prozessen, die das Quellobjekt verändert haben, wurde die neue Änderungsinformation für die Datenabbildung verwendet. Dadurch bestand die Möglichkeit, dass Änderungen zwischen dem Lesen und dem Aktualisieren nicht erkannt werden.

Die Sage CRM-Verbindung:

- Wenn die SQL-Bridge und die SQL-Zugangsdaten definiert wurden, hat die Validierung nicht die SQL-Bridge verwendet.

Bei schreibenden Datenbankzugriffen werden leere Zeichenketten zu Datums- und numerischen Felder nicht übergeben, da dies zu einem Convert-Fehler führt. Der Parameter verwendet dann den NULL-Wert.

Die Seriendruck-Prozesse haben bei der Grenzwertbehandlung nicht zwischen Datum und Version unterschieden, was zu einem Typ-Fehler führen konnte.

Die Seriendruck-Verbindung:

- Die Behandlung von Bildern wurde korrigiert. Der Präfix „Picture:“ wird nur in der Vorlage verwendet.

Der Prozess für den Emailversand von Serienbriefen wurde korrigiert. Die Quelldaten wurden nicht an die Email-Verbindung übergeben, wodurch Platzhalter in der Nachricht nicht ersetzt wurden.

9.9 Release 4.4.3 - 24.01.2023

9.9.1 Änderungen

Der Prozess „Sage CRM ABC-Analyse“

- Dieser neue Prozess setzt auf dem Addon ABC-Analyse für Sage CRM auf und aktualisiert Firmen gemäß der definierten Kriterien und dem erreichten Scoring.

Die Zoho-Verbindung

- Eine Feldnotation für gefilterte Abfragen, z.B. aus der Transformation, kann direkt für die Einzelabfrage von Datensätzen genutzt werden.
- Die Beschreibung zu Terminen enthält den Betreff und das Datum, damit auch Serienvorkommen einfach unterschieden werden können.

Die CAS-Verbindung

- Für das Lesen und Löschen von Links wird der Parameter „Attribute“ ausgewertet, um die korrekte Verknüpfung auszuwählen.

Die SAP-IDOC-Verbindung

- Für die Generierung der DOCNUM werden auch die Millisekunden verwendet, damit bei einer schnellen Folge von Dokumenten keine Dopplungen entstehen.

Die Email-Verbindung

- Beim Versand einer Email mit einem Datenobjekt wird der geparste Inhalt der Email zurückgeliefert und kann so für Archivierungen genutzt werden.

Der Zweitfilter wurde bei diesen Prozessen aktiviert.

- Sage CRM Marketing-Center Eintrag für Personen nach CleverReach Empfänger
- Sage CRM Marketing-Center Eintrag für Personen nach InxMail Empfänger
- InxMail Bounce nach Sage CRM Marketing-Center Gruppe
- InxMail WebBeaconHit nach Sage CRM Marketing-Center Gruppe
- InxMail Klick nach Sage CRM Marketing-Center Gruppe
- InxMail Abmeldung nach Sage CRM Marketing-Center Gruppe

Bei diesen InxMail-Prozessen wird ein Skip ausgelöst, wenn keine Listen-ID vorhanden ist. Dies kann passieren, wenn die Liste in InxMail gelöscht wird und Resultate dazu abgerufen werden.

- InxMail Bounce nach Sage CRM Marketing-Center Gruppe
- InxMail WebBeaconHit nach Sage CRM Marketing-Center Gruppe
- InxMail Klick nach Sage CRM Marketing-Center Gruppe
- InxMail Abmeldung nach Sage CRM Marketing-Center Gruppe

Die ActiveCampaign-Verbindung

- Diese neue Verbindung ermöglicht die Anbindung des E-Marketing-Anbieters ActiveCampaign.
- Synchronisationen können mit dem Universalprozess eingerichtet werden.
- Für die Kombination mit CAS stehen bereits Vorlagen zur Verfügung.
- Siehe *ActiveCampaign*

Für die Erzeugung eines unbedenklichen Feldnamens für Abfrageergebnisse wird eine neue Methode eingesetzt, die folgende Sonderzeichen ersetzt: . : ? % - + * | ( ) [ ] / “ Diese Methode wird in der REST API-, der Zoho- und der ActiveCampaign-Verbindung eingesetzt.

Die Graph-Verbindung

- Die Konfiguration der Verbindung wurde überarbeitet und kann jetzt durch verschiedene Auswahlen besser gesteuert werden.
- Das Anmeldeverfahren wird per Auswahl gesteuert und kann einen Admin- oder Benutzer-Refresh-Token verwenden.
- Der Benutzerkreis für Synchronisationen wird per Auswahl gesteuert und unterstützt die Benutzer-Refresh-Token.
- Der Scope kann konfiguriert werden, was durch die Unterstützung von Refresh-Token erforderlich wurde.
- Die Beschreibung zu Terminen enthält den Betreff und das Datum, damit auch Serienvorkommen einfach unterschieden werden können.
- Es steht eine schreibgeschützte Liste der vorhandenen Benutzer-Zustimmungen zur Verfügung. Dies soll eine Übersicht zu den synchronisierten Benutzern liefern.
- Das Objekt Event enthält das schreibgeschützte Feld „user“, damit der Benutzerbezug in Transformationen ausgewertet werden kann. Dies ist z.B. bei externen Organisatoren und Zoho Terminen relevant, da der Host immer ein Benutzer sein muss.

Die Synchronisation zwischen Microsoft Graph und Zoho CRM, sowie Sage CRM wurde überarbeitet. Das bisherige Verfahren konnte zu Fehlern bei Serienterminen führen.

- Der Änderungsspeicher wird nicht länger verwendet.
- Eine neue oder geänderte Serie löscht ggf. die vorhandene Serie im Ziel komplett und legt diese neu an. Danach werden alle Instanzen der Serien abgeglichen, um Ausnahmen und Lücken zu übertragen. Der Abgleich erzeugt auch die Datenabbildungen zwischen den Vorkommen.
- Eine Neuanlage von Vorkommen außerhalb der Serienverarbeitung findet nicht statt.

Der Syncler-Administrator

- Es steht eine neue Liste für die internen Parameter zur Verfügung, die z.B. Delta- oder Refresh-Token verwendet werden.
- Die Protokoll-Listen wurden in Allgemein, Prozesse, Datendienste und Webhooks aufgetrennt.

Die Synchronisation zwischen Sage CRM Marketing-Center und CleverReach oder InxMail

- Die betroffenen Prozesse prüfen das Änderungsdatum der Quelle gegen die Datenabbildung und überspringen ggf. den Datensatz.
- Das Zurückschreiben der Referenz-ID erfolgt nur bei unterschiedlichen Werten.
- Das Abfragen von geänderten Gruppeneinträgen wählt Einträge ohne Referenz-ID unabhängig vom Datum aus. Damit soll eine zeitliche Differenz zwischen der Synchronisation der Gruppe und der Gruppeneinträge kompensiert werden.

Die eingehenden Webhooks

- Diese neue Funktion bietet die Möglichkeit eingehende Webhooks zu definieren, um eine einfache Integration in externe Systeme zu ermöglich.
- Webhooks können dabei lesend oder schreibend verwendet werden.
- Die ein- oder ausgehenden Daten können dabei flexibel als JSON-Objekt definiert werden.

- Es stehen auch zwei spezielle Prozesse für die Webhook-Funktion zur Verfügung, damit die Verarbeitung mit einer Transformation kombiniert werden kann.
- Siehe /webhooks

Der Prozess „Sage 100 VK-Preis nach Sage CRM Preis“ wurde komplett überarbeitet und mit einem Initial-Prozess ergänzt. Dies dient vorallem der Leistungssteigerung.

- Die neue Version verzichtet vollständig auf Datenabbildungen, um die Leistung zu steigern. Ein Preis wird eindeutig durch die Mengeneinheit, das Produkt und die Preisliste definiert. Eine Verdichtung wird nicht mehr unterstützt (Übereinstimmungsregeln).
- Die betrachteten Preislisten können direkt im Prozess mit einem neuen Parameter gefiltert werden.
- Der Initial-Prozess fragt immer alle Preise und Datenabbildungen von Artikeln und Preislisten ab. Damit werden Negativabfragen für untergeordnete Preislisten und mehrfache Datenbankabfragen vermieden.

Die CSV-Prozesse

- Neue Felder aus der Transformation werden dem Export hinzugefügt.
- Der Parameter „Export-Feldliste“ kann neben der Feldauswahl auch die Feldreihenfolge steuern.

Die Sage b7-Prozesse

- Durch einen neuen Parameter kann das Auslesen des Änderungsspeichers ohne Berücksichtigung des Datums erfolgen. Dadurch gehen keine Daten verloren.

Die Sage b7-Verbindung

- Durch einen internen Parameter wird verhindert, dass unterschiedliche Prozesse zeitgleich aus dem Service Bus Nachrichten abrufen.

9.9.2 Korrekturen

Die JSON-Konvertierung im SDK-Helper

- Die automatische JSON-Konvertierung für den SDK-Helper und der SDK-Funktionalität hat für das Resultat nur das bekannte Datenschema verwendet. Neue Felder aus der SDK-Verarbeitung wurden nicht berücksichtigt.

Die JSON-Konvertierung

- Die Datentypen Array und Objekt, die als Feld definiert wurden, wurden nicht übernommen.

Der Prozess „Microsoft 365 Aufgaben nach Sage CRM Aufgaben“

- Die Uhrzeit für die Aufgabenbenachrichtigung wurde nicht übernommen.

Der SDK-Helper

- Die Methoden InvokeGetData und InvokeSetData haben eine fehlerhafte URL verwendet.

Die Salesforce-Verbindung

- Bei Aktualisierungen wird die Spalte „id“ generell nicht in die Feldliste übernommen.

Der Prozess „Salesforce Opportunity nach SAP Beleg“

- Das Element „E1EDP05“ in den Positionen wird nicht übernommen, wenn das Feld „KSCHL“ den Wert „ZPR0“ hat. Dies ist Teil der Preisberechnung und verhindert eine Fehlermeldung in der IDOC-Verarbeitung.

Der Prozess „Zoho CRM Meeting nach Microsoft Graph Ereignis“

- Vermeindlich endlose Serien in Zoho liefern als Enddatum eine -1. Das wurde nicht in einen gültigen Wert überführt. Da eine endlose Serie dennoch eine beschränkte Anzahl an Vorkommen hat, wird diese Serie mit dem maximalen Enddatum aller Vorkommen angelegt.

Die CSV-Verbindung

- Zeilenumbrüche werden bei der Schema-Generierung aus den Feldnamen entfernt.

Die Funktion „Zurücksetzen der Sync-Infos“ überspringt Einträge mit Graph-Sync-spezifischen Parametern, da dies ansonsten zu einer Störung der Synchronisation führen würde.

Die ID-Werte in den Datenabbildungen zu Microsoft 365 Objekten werden case-sensitive behandelt.

Der Bulk-Abfrage-Prozess

- Für die Fehlerbehandlung „Fehler ignorieren“ werden keine Änderungsdatensätze mehr angelegt.

Behandlung von Nachfolgeprozessen durch Abfrageprozesse

- In der Release 4.4.2 ist ein Fehler entstanden, wodurch der Änderungsdatensatz nicht dem Nachfolgeprozess zugeordnet wurde. Der erste Prozess hat den Datensatz ein zweites Mal verarbeitet und erst dann an den Nachfolger übergeben. Sollten ID-Felder definiert sein, hat der Nachfolger bereits bei der ersten Ausführung den Datensatz verarbeitet.

9.10 Release 4.4.4 - 22.03.2023

9.10.1 Änderungen

Die Sage b7-Verbindung

- Die Verbindung wurde grundlegend überarbeitet und auf die Bibliothek Azure.Messaging.ServiceBus migriert.

Der Syncler Administrator

- In den Bereichen Datenabbildung, Protokolle, Änderungsspeicher und Sicherungsspeicher steht eine neue Datensatz-Zusammenfassung zur Verfügung, welche Informationen trennt und Felder in Listenform ausgibt.
- Mit einem Doppelklick können Feldinhalte leicht entnommen werden.

Der Prozess „Sage CRM Aufgabe nach Microsoft Aufgabe“.

- Ein geänderter Benutzer im CRM führt zum Löschen der Aufgabe im ursprünglichen Postfach und zum Neuanlegen im aktuellen Postfach. Damit kann eine neu zugewiesene Aufgabe synchronisiert werden.
- Dabei werden auch die Datenabbildungen gelöscht, damit die Aufgabe nicht im CRM gelöscht wird, falls der neue Benutzer nicht synchronisiert wird.

Die Zoho CRM-Verbindung

- Es werden Contact-Roles zwischen Abschlüssen und Kontakten unterstützt. Siehe *Zoho CRM - Kontakt-Rollen*
- Es werden Notes für andere Module unterstützt. Für Notes steht ein eigenes Schema-Objekt zur Verfügung.
- Per Filter können Datensätze aus einer Beziehung abgefragt werden. Die Formulierung erfolgt in Feldnotation mit den Parametern „Related“ und „RelatedId“. Der Filter „Related|:**|Accounts|**;|RelatedId|:**|1234|**;|“ ruft nur die Datensätze zu diesem Account ab.
- Der Fehler „Maximum of 200 records allowed“ zu Angebotspositonen bei der Übertragung eines Angebots wird nicht wiederholt, da eine Korrektur nur durch eine Quelländerung möglich ist.

Der Universal-Löschprozess

- Es wurden Feldzuordnungen, Adhoc-Ausführungen und Filter freigegeben.

Die Universal-Prozesse

- Mit einem zusätzlichen Parameter kann festgelegt werden, ob die IDs in Datenabbildungen Groß- und Kleinschreibung unterscheiden müssen.

Der Prozess „SAP Beleg nach Salesforce Opportunity“

- Mit einer zusätzlichen Suche können Positionen für die Aktualisierung zugeordnet werden.
- Ohne eine Suchangabe erfolgt die Zuordnung weiterhin über die Reihenfolge der Positionen.

Die SQL-Brücke

- Die API der SQL-Brücke hat den neuen Endpunkt „Check“, der die Verfügbarkeit der API prüfen kann.

Die Salesforce-Verbindung

- Das Speichern einer Verkaufschance unterstützt das kombinierte Schreiben von Opportunity und OpportunityLineItems.

Die Graph-Verbindung

- Mit einem Parameter können die Tage bis zur Aktualisierung des Zeitfensters eingestellt werden. Damit müssen nicht jeden Tag alle Termine des neuen Zeitfensters geprüft werden.
- Das Teilnehmer-Schema hat die Felder isExternalOrganizer und isCurrentUser erhalten. Beide werden gefüllt, wenn es sich um einen externen Organisator handelt und der Teilnehmer auch der aktuelle Benutzer der Abfrage ist.
- Der Refresh-Token wird automatisch bei der Anforderung eines Access-Tokens aktualisiert.

Die Serienbrief-Verbindung

- Das Schema für die Serienbrieferzeugung wurde um „InsertDocument“ erweitert. Über einen „InsertMarker“ im Hauptdokument kann ein fremdes Dokument gezielt eingefügt werden. Das Einfügen erfolgt auf Block-Ebene, wodurch Kopf- und Fußzeile nicht übernommen werden.
- Einem Feld mit dem Präfix „Picture:“ kann neben einer Json-Anweisung auch direkt eine Url für das Einbinden einer Grafik übergeben werden.

Die Support-Datenbank

- Das neue Feature „Support-Datenbank“ steht zur Verfügung und kann gebucht werden.
- Damit wird eine neue Datenbank für den Account angelegt, deren Größe in die maximale Datenbankgröße mit eingerechnet wird.
- Innerhalb dieser Datenbank besteht weitestgehend eine SQL-Autonomie.
- Es können beliebig Tabellen amgelegt oder importiert werden.
- Die Importfunktion kann auch vorhandene Tabellen erweitern oder leeren.
- Als Importquelle können Xlsx- oder CSV-Dateien genutzt werden.
- Außerdem steht eine neue Verbindung „Support-Datenbank“ zur Verfügung, über die dieser Bereich in die Prozessausführung integriert werden kann.
- Es können Tabellen über Prozesse gefüllt oder mit beliebigen SELECT-Anweisung für Prozesse ausgelesen werden.
- Jede vorhandene Tabelle erzeugt dabei ein eigenes Schema für diese Verbindung.
- Es können auch SQL-Anweisungen innerhalb der Datenbank mit einem Prozess ausgeführt werden.
- Im Syncler Administrator steht eine Tabellenbearbeitung für die Inhalte der Tabelle zur Verfügung.
- Siehe *Support-Datenbank*

CSV-Prozesse

- Die Parameter LastDateTime- und LastVersion-Spalten wurden aktiviert, damit diese Informationen in der Verarbeitung genutzt werden können, um eine synchrone Situation zu erkennen.

Webhooks

- Webhooks unterstützen die Verwendung eines Abfrageprozesses für das Lesen von Daten. Dafür steht ein neuer Filterparameter zur Verfügung, der in die Abfrage einefügt werden kann.
- Webhooks können Formulardaten empfangen und verarbeiten. Die Unterscheidung erfolgt über den Content-Type der Anfrage.
- Zusätzlich kann eine LandingPage für die Weiterleitung definiert werden. Dadurch kann ein Webhook als Universal-Empfänger für Web-Formulare verwendet werden.
- Webhooks können die empfangenen Daten im Änderungsspeicher für einen Prozess ablegen.
- Mit einem speziellen Leseprozess können die Daten aus dem Änderungsspeicher in einem Ablauf verarbeitet werden.
- Wenn ein Webhook Daten mit einer Verbindung speichert, kann das Lese-Schema für die Rückgabe von Daten genutzt werden. Damit können z.B. erzeugte IDs weiterverarbeitet werden.

MailChimp-Prozesse für Sage CRM Marketing-Center

- Für das Sage CRM Addon „Marketing-Center“ stehen Prozesse für die Verarbeitung zur Verfügung.

Die MailChimp-Verbindung

- Die Verbindung unterstützt Clicks, Opens und Email-Activities.

Die Sage WinCarat-Verbindung

- Für Syncler steht die Verbindung zu Sage WinCarat zur Verfügung.
- Für die Synchronisation können die Universal-Prozesse genutzt werden.
- Die Interessenten-Konvertierung kann mit separaten Prozessen und einer eigenen laufenden Mandantennummer umgesetzt werden. Voraussetzung dafür ist ein gleichbleibendes Kriterium, z.B. der Matchcode.

Der Testlauf in Prozessen

- Der Testlauf innerhalb der Transformation wurde grundlegend überarbeitet und erweitert.
- Es können Quelldaten per Filter oder aus Abfragen ermittelt werden.
- Außerdem können Zieldaten über ID, Filter oder Datenabbildung gelesen und die Feldzuordnungen getestet werden.
- Änderungen werden dabei farblich hervorgehoben.
- Siehe /processes/converting/dryrun

Die Transformation „Webhook aufrufen“

- Diese neue Transformation kann einen beliebigen Webhook aufrufen, um Daten zu lesen oder zu schreiben.
- Siehe *Webhook aufrufen*

Die Transformation „Json in Spalten“

- Diese neue Transformation kann Json-Daten aus einem Feld in einzelne Spalten überführen.
- Siehe *Json in Spalten*

9.10.2 Korrekturen

Die Sage b7-Verbindung

- Die Verarbeitung von geteilten Nachrichten prüft die ID des Datensatzes eines jeden Teils für die Zusammenführung. Bisher wurde nur die Reihenfolge verwendet, was bei wiederholtem Lesen von Nachrichten zu Fehlern führen konnte.
- Das Löschen von Nachrichten erfolgt bereits bei der Verarbeitung, damit das Lock-Timeout nicht überschritten wird. Dies war in der Cloud-Umgebung durch die eingeschränkte Parallelität möglich und hat zum wiederholten Lesen von Nachrichten geführt.
- Bei geteilten Nachrichten erfolgt das Löschen erst bei der Verarbeitung des letzten Teils.

Die Sage CRM-Verbindung

- Die Prüfung des Organisators einer Kommunikation und die Zwangsverknüpfung als Teilnehmer wurde auch bei Aufgaben angewendet. Dadurch könnten zusätzliche Kommunikationslinks für Aufgaben entstehen.

Die Graph-Verbindung

- Das Änderungsdatum des Serienmasters wurde bei der Delta-Verarbeitung noch an die Vorkommen übergeben. Das führte durch die neue Serienverarbeitung zu unnötigen Änderungserkennungen und Konfliktwarnungen.
- Der Vergleich zwischen Organisator und aktuellem Benutzer war case-sensitiv. Das konnte beim Schreiben von Terminen zu einem Fehler führen, falls sich die Schreibweise zwischen den Systemen unterschieden hat.
- Beim Schreiben einer Aufgabe wurde das Fehlen des Refresh-Tokens als Fehler interpretiert. Jetzt wird eine Warnung und Überspringen ausgelöst.

Die InxMail-Verbindung

- Das Abrufen eines gelöschten Datensatzes führt zu einen 404-Fehler, der als Fehler behandelt und nicht als gelöschtes Ziel interpretiert wurde.

Die Vorlagen für Prozesse ausgehend von Microsoft Graph

- Die Vorlagen haben eine Transformation „Dokument konvertieren“ für die Umwandlung von HTML zu Text verwendet. Falls der Termin oder die Aufgabe aber bereits reinen Text enthalten hat, wurden dadurch Zeilenumbrüche entfernt.
- Jetzt wird die Transformation „Html zu Text“ verwendet, bei der Zeilenumbrüche erhalten bleiben.

Die CAS-Verbindung

- Die bisherige Abfrage nach Verknüpfungen hat deaktivierte Adressen nicht berücksichtigt.

Interne Abfragen zu Datenabbildungen mit externen IDs verwenden einen Unicode-Präfix, da sonst Datenabbildung unter Umständen nicht gefunden werden.

Das Zurückschreiben in Prozessen wurde angepasst, damit nur tatsächliche Änderungen erkannt und übertragen werden.

Der Prozess „Microsoft Graph Ereignis nach Zoho CRM Meeting“

- Wenn die Funktion „Alle zukünftigen Vorkommen aktualisieren“ im Zoho genutzt wird, ändert sich die u_id bei diesen Datensätzen. Das ist vergleichbar mit einem Splitting der Serie.
- Außerdem ist das Serienschema der neuen Serie mit der alten vermischt und damit ungültig.
- Bei der Neuanlage einer Serie wurden so nicht alle Vorkommen entfernt und es kam zu Dubletten.
- Das neue Verfahren löscht die Zielvorkommen über die Datenabbildungen.

Der Prozess „Zoho CRM Meeting nach Microsoft Graph Ereignis“

- Wenn die Funktion „Alle zukünftigen Vorkommen aktualisieren“ im Zoho genutzt wird, ändert sich die u_id bei diesen Datensätzen. Das ist vergleichbar mit einem Splitting der Serie.
- Da das Serienschema der einzelnen Teile aber fehlerhaft ist, werden nicht alle Vorkommen korrekt angelegt.
- Die Ermittlung des Anfangs und Ende der Serie erfolgt deshalb über das Maximum und Minimum aus den Vorkommen.

In den Bulk und CSV-Prozessen kam es zu einem Konvertierungsfehler bei der Verwendung von Datensatzversionsnummern.

Das OAuth2-Anmeldeverfahren fordert zum Login statt zur Account-Auswahl auf. Damit kann ein Login auch gewechselt werden.

Das Oauth2-Anmeldverfahren im Syncler Administrator gibt ggf. eine Fehlerbeschreibung aus.

9.11 Release 4.5.0 - 13.07.2023

9.11.1 Änderungen

Die Microsoft Dynamics 365 Customer Engagement Verbindung

- Für Syncler steht die Verbindung zu Microsoft Dynamics 365 Customer Engagement zur Verfügung.
- Für die Synchronisation können die Universal-Prozesse genutzt werden.
- Mit Abfrageprozessen können beliebige Anfragen an die API gestellt werden.
- Es werden diese Schemaobjekte unterstützt: „account“, „contact“, „lead“, „opportunity“, „opportunityproduct“, „product“, „businessunit“, „transactioncurrency“, „externalparty“, „territory“, „service“, „systemuser“, „team“, „equipment“, „pricelevel“, „productpricelevel“, „uomschedule“, „uom“
- Es stehen Vorlagen für die Kombination mit der Sage 100 bereit.
- Die Anmeldung kann per OAuth 2.0 oder Client-ID erfolgen.

Der Syncler Administrator

- Transformationsschritte können mit Klick statt Doppelklick geöffnet werden.
- Es wurde eine Import-Funktion für Prozesse ergänzt.
- Tabellen der Support-Datenbank können per Menü geleert werden.
- Es können Spalten in der Support-Datenbank an Tabellen ergänzt werden.
- Ein Abfrage-Dialog kann für die Support-Datenbank geöffnet werden, um beliebige Abfragen mit Ergebnisausgabe auszuführen.
- Einzelne Spalten einer Tabelle in der Support-Datenbank können als Index definiert werden, um Abfragen zu optimieren.
- An einer Verbindung können Auswahllisten eingesehen, bearbeiten und angelegt werden. Diese können in Transformation oder dem DQM eingesetzt werden.
- Im Testlauf wird nicht ausschließlich das Objektschema für Quell- und transformierte Daten verwendet, sondern zusätzlich vorhandene Daten werden ergänzt. Dies kann z.B. in untergeordneten Listen der Fall sein.
- An Verbindungen steht die Funktion „Abfrage ausführen“ zur Verfügung. Der folgende Abfrage-Dialog kann beliebige Abfragen ausführen und das Ergebnis darstellen.
- Der Neustart eines Warteschlangeneintrags leert die Liste der fehlgeschlagenen Datensätze, was sonst das neue Resultat beeinträchtigt hat.

Der IDoc-Salesforce-Beleg Prozesse

- Es kann ein Skript für die Datensatzverarbeitung hinterlegt werden. Damit kann unabhängig von der Transformation die Verarbeitung beeinflusst werden.
- Wenn sich bei der Aktualisierung einer Position die PricebookEntryId ändert, wird diese Zielposition zum Löschen markiert. Die Quelldaten werden dann als neue Position übertragen.

Schema und Prozesse für geschachtelte Datensätze

- Die Positionsdatensätze enthalten jetzt immer eine Kopie des Hauptdatensatzes und eine einzelne Position.

Damit stehen auch Daten des Hauptdatensatzes in Transformationen oder Bedingungen zur Verfügung.

Prozess Sage CRM Kommunikation nach Microsoft Graph Ereignis

- Mit einem neuen Parameter kann das Verschieben eines Termins zwischen Postfächern / Organisatoren aktiviert werden.

Die Basis dafür ist ein Feldmapping auf den Organisator. Quell- und Zielwert werden bei der Aktualisierung verglichen und ggf. wird Ziel und Mapping gelöscht. Der Termin wird dann neu angelegt.

Die SQL-Verbindung

- Das Speichern von UniqueIdentifier durch Guid-Spalten aus Prozessen wurde ergänzt.

Datenabbildungen

- Fehler beim Speichern erzeugen einen abschließenden Fehler, der nicht automatisch wiederholt wird.

Die Saleforce-Verbindung wurde auf Version 57 aktualisiert.

Bei der Verarbeitung von Ablaufdatensätzen werden zusätzliche Debug-Meldungen erzeugt.

Die Auswahllisten können um deutsche und englische Übersetzungen ergänzt werden. Diese können dann im generischen ERP-Fokus zur Übersetzung verwendet werden.

Die Transformation „Daten abfragen“ verwendet einen neuen Limit-Parameter, der automatisch die Resultatsmenge begrenzt. Dies optimiert die Transformation bei umfangreichen Abfrageresultaten.

Der Universal-SDK-Prozess

- Verschiedene Ergänzungen aus dem Universalprozess wurden in diesen Prozess übernommen, damit der Funktionsumfang identisch ist.

Die Salesviewer-Verbindung

- Mit dieser neuen Verbindung können Salesviewer Leads und Besuchsdaten mit einem Universalprozess in jedes beliebige System übertragen werden.

Die Docu-Sign-Verbindung

- Mit dieser neuen Verbindung kann ein Remote-Signing-Prozess implementiert werden.
- Der Versand von Unterschriftsanfragen und das Speichern von unterschriebenen Dokumenten kann damit realisiert werden.

Universalprozess nach Seriendruck

- Mit diesem neuen Prozess können auch schema-basierte Daten für Seriendrucke genutzt werden. Dies war bisher nur mit Abfragen möglich.

Die Evalanche-Verbindung

- Mit dieser neuen Verbindung kann der dei Evalanche Marketing Software angebunden werden.
- Die Verbindung unterstützt Mandanten und Pools.

- Es stehen Mailing, MailingDetails, MailingStatistics, Article, Profile, Recipient, Bounce, Unsubscription, GrantedPermission, RevokePermission, Impression (Openings) und Clicks zur Verfügung.
- Profile können beliebig angelegt, aktualisiert oder gelöscht werden. Auch können Berechtigungen gesetzt oder entfernt werden.
- Die Reultate werden für einen einfachen Einsatz in Prozessen aufbereitet und erweitert.

Ablauf-Prozesse

- Die Übernahme der Zieldaten erfolgt automatisch, sobald der aktuelle Prozess als Vorgänge ausgewählt wurde und dessen Zieldaten verarbeitet werden sollen.
- Bisher war das an die Übernahme der Quelldaten gekoppelt, was missverständlich war und unnötig Ressourcen beansprucht hat.

Data Quality Management

- Dieses neue Feature kann in zwei unterschiedlichen Editionen gebucht werden.
- Mit dem Data Quality Manager haben sie die Möglichkeit einen Zielzustand für ihre Daten zu definieren und die Erreichung zu analysieren.
- Daten die dem Zielzustand nicht entsprechen können manuell oder automatisch korrigiert werden.
- Eine Übertragung in ein beliebiges Zielsystem schließt den Prozess ab.
- Neben Feldeigenschaften können auch Beziehung zwischen Datenobjekten behandelt werden.
- Die Suche und Verarbeitung nach Dubletten rundet den Funktionsumfang ab.
- Die Nutzung kann manuell oder auch vollautomatisch konfiguriert werden.

9.11.2 Korrekturen

Abfrage nach Seriendruck Prozesse

- Das Zielobjekt wurde nicht für Abläufe bereitgestellt, wodurch keine Kombination möglich war.

Die Sage b7 Verbindung

- Die Responseverarbeitung wurde korrigiert. Dies betrifft u.a. die Antwort auf neu angelegt Datensätze. Fehler wurden als Kommunikationsfehler statt Systemfehler behandelt.
- Die Übergabe von leeren Integer- und Auswahlfeldern erfolgt als 0 und nicht als Null.

Die Json zu Datenobjekt Konvertierung

- Bei der Behandlung von primitiven Arrays werden Objektgruppen mit Value-Feldern gebildet. Bei der Konvertierung zu Json wurde daraus keine primitiven Arrays erzeugt. Dadurch gelangt eine Json-Notation in das Datenobjekt. Dies passiert in SDK-Prozessen, da dort die Objekte in den Helper als JObject gegeben werden und nach Skriptausführung wieder in ein Datenobjekt konvertiert werden.
- Die Konvertierung hat keinen Verknüpfungstyp für Unterobjekte definiert, wenn der Wertvorrat des Schemas erschöpft war. Dadurch wurden diese Daten in der Transformation verworfen. Jetzt wird ein fortlaufender Index verwendet.

Abfrage-Prozesse

- Wenn in Abfrage-Prozessen mit Datenabbildungen gearbeitet wird, hat die Übereinstimmungssuche keine bereits zugeordneten Suchergebisse ausgeschlossen. Dadurch konnten mehrere Quelldatensätze auf einen Zieldatensatz abgebildet werden.

Die Microsoft-Graph-Verbindung

- Der Filter für Abfragen wird nicht mehr auf Sonderzeichen geparst. Dadurch war keine Abfrage mit Sonderzeichen möglich.

Der Sage CRM ERP-Fokus

- Das Filtern von Datumswerten wurde korrigiert. Durch einen erneuten Seitenaufruf ist der vorangegangene Wert ungültig geworden.
- Der Excel-Export wurde angepasst und HTML aus den Aggregatszeilen entfernt.

Die REST-API-Verbindung

- Datumswerte werden explizit als lokal geparst.
- Das Parsen von Datumswerten kann mittels Parameter abgeschaltet werden.

Die Datensatzverarbeitung prüft auf vorhandene Primärschlüssel, bevor doppelte Datensätze ausgeschlossen werden. Dadurch werden alle Daten ohne Primärschlüssel auch verarbeitet.

CAS und Zoho Prozesse

- Die SIS_MESSAGE-Felder wurden beim Leeren mit einem Leerzeichen beschrieben. Dies ist nur bei Sage CRM eine notwendige Vorgehensweise.

CAS Adresse nach Cleverreach oder InxMai

- Die CAS-Aktualisierungsinformation wurde beim Zurückschreiben nicht korrigiert, wodruch eine erneute Verarbeitung ausgelöst wurde.

Die SQL-Bridge

- Beim Schreiben wurde versucht die Transaktion im entfernten System zu zählen, was einen Verbindungsfehler auslöst.
- Einzeln gelesene Datensätze hatten durch die Json-Serialisierung bereits als geändert markierte Felder.

Die Sage 100 Belegübertragung

- Die bisherige Fehlerbehandlung hat keine Wiederholung nach der Übergabe an die Sage 100 zugelassen.
- Nach dieser Anpassung ist eine Fehlerwiederholung abhängig von der Antwort zur Anfrage möglich.

Prozesse mit Unterprozessen

- Bei verschiedenen Prozessen, die Unterprozesse nutzen, wurde die Nachrichtenübermittlung nicht angebunden.
- Betroffen sind CAS-Inxmail, CAS-Cleverreach, CRM-Cleverreach, CRM-Inxmail, CRM-Mailchimp

Die Inxmail-Verbindung

- Die Suche nach Emailadressen wurde korrigiert.
- Die Übertragung von geschachtelten Daten wurde korrigiert. Doppelte Listenanmeldungen werden automatisch übersprungen.

9.12 Release 4.5.1 - 03.11.2023

9.12.1 Änderungen

WinCarat Verbindung

- Die Änderungsinformation wurde auf die Spalte „ROWDATUM“ umgestellt. Die Versionsnummern werden nicht mehr verwendet.

Microsoft Graph nach Sage CRM und Zoho CRM Prozesse

- Der Reminder für vergangene Termine wird anhand des Startdatums abgeschaltet.

Data Quality Management

- Für die Dublettensuche kann je Spalte eine Ersetzungsliste angegeben werden. Mit dieser Pickliste wird der Vergleichswert beidseitig für den Vergleich angepasst. Damit können Bindeworte oder Unternehmensformen normalisiert werden.
- Transformationen können in der Aufbereitung angewendet werden.

SDK Helper

- Die InvokeUrl-Methode hat einen zusätzlichen optionalen Parameter für den ContentType bekommen. Der Standard ist application/json.

Syncler Administrator

- Neuer PageType OAuthGrantFlow für eine generische OAuth Authorization eingefügt. Wird aktuell in der SDK Verbindung und Vorlage SDK Verbindung angeboten.
- OAuth-Flow und OAuth-Flow 365 mit WebView2 umgesetzt, damit aktuelles Javascript unterstützt wird. Ggf. muß WebView2 auf dem Rechner nachinstalliert werden.

Syncler API

- Nach dem Schema-Refresh einer Verbindung wird diese nochmal gespeichert. Damit ist es möglich, Verbindungseigenschaften in z.B. Schema-Skripten zu beeinflussen.
- Für den Endpunkt RunConnectionSetData wurde ein RequestSizeLimit von 100MB festgelegt. Der Standard sind 30MB.

Syncler Datenbank

- Bei 95% der DB Nutzung wird für den Tenant eine Warnung täglich versendet.

Transformation

- In der Transformation „Daten abfragen“ kann eine beliebige Verbindung ausgewählt werden.
- Neue Transformation „Geschlecht ermitteln“. Mittels Vornamen und Land kann das Geschlecht einer Person ermittelt werden. Zu den Resultaten Männlich, Weiblich und Unisex können Ausgabewerte definiert werden.

Graph Verbindung

- Neues Schema UserRefreshToken ermöglicht die Überwachung von gesammelten Refresh-Token. Per Prozess kann damit auf abgelaufene Refresh-Token reagiert werden.
- Die Notation Related|:**|group|**;|RelatedId|:**|1234|**;| als Filter wird für Abfrageparameter ausgewertet. Damit können z.B. Mitglieder einer Gruppe abgerufen werden.

Universal Schreibprozess

- Damit können Json-Daten ohne Quelle direkt geschrieben werden.

Databyte Verbindung

- Die neue Verbindung zu Databyte steht zur Verfügung.

Sage 100 - Salesforce Belegübertragungsprozess

- Parameter für Positionssortierung wurde hinzugefügt.

Neuer Prozess

- Universal Prozess für geschachtelte Daten nach Seriendruck.

9.12.2 Korrekturen

Prozessausführung durch Ablauf

- Bei der Variante „Prozess für jeden Datensatz und angepassten Filter ausführen“ wurde bei den einzelnen Ausführung nicht auf deren Abschluss gewartet.
- Dadurch konnten einzelne Ausführungen parallel laufen.

CAS Verbindung

- Der Upload von Dokumenten wurde korrigiert. Ein Aufruf per PUT liefert keinen Location-Header, der für die Ergebnisverarbeitung erforderlich ist.

Abfrageprozesse

- Die exakte Übereinstimmungssuche hat mit der Einstellungen „Nur Datenabbildung“ den Datensatz übersprungen, wodurch dieser nicht für Nachfolgeprozesse bereitgestellt wurde. Der Datensatz wird jetzt als Erfolgreich markiert.

Speicheroptimierung

- Erzeugte Nachrichten wurden nach der Datensatzverarbeitung bis zum Abschluss des Prozesses im Speicher gehalten. Dieser werden jetzt nach Weiterleitung aus dem Speicher entfernt.

Webhooks

- Webhooks haben aus Prozessen keine typisierten Daten geliefert, falls diese transformiert wurden. Die Json-Antwort wird jetzt nach dem Datentyp geparst.

Support-Datenbank

- Bei der manuellen Bearbeitung wurde das Leeren eines Feldes nicht als Änderung erkannt.

Die Universal- und Universal-SDK-Prozesse entfernen Änderungsspeicherdatensätze nur noch anhand der Guid und nicht der Prozess-Zuordnung.

9.13 Release 5.0.1 -

Mit dieser Version wird eine neue Generation des Synclers veröffentlicht. Die Anwendung wurde für die .Net-Version 8.0 überarbeitet und verabschiedet sich damit komplett von .Net-Core. Außerdem wurden einige strukturelle Veränderungen vorgenommen, die die Entwicklung betreffen und den Weg für ein neues Frontend vorbereiten. Eine on-premises Version sollte nicht aktualisiert, sondern ausgetauscht werden.

9.13.1 Änderungen

CAS-Verbindung

- Die Verbindung wurde für die Webservice-Version 7.0 angepasst.
- Das Schemaobjekt Tag wurde ergänzt.
- Abhängig von der Version wird zwischen dossier und link unterschieden.
- Mit dem neuen Schema „recyclebin“ können Datensätze aus dem Papierkorb abgerufen werden. Der Filter mit in Feldnotation im Wert „deleted“ den Objekttyp enthalten.
- Mit dem neuen Prozess „CAS gelöschte Daten“ können gezielt gelöschte Datensätze verarbeitet werden.

KAPITEL 10

Editionen (Draft)

Syncler kann über die Cloud Plattform oder mit einer On-premises Installation genutzt werden.

In der Cloud-Umgebung gibt es einige Einschränkungen, die On-premises nicht vorliegen.

In der Cloud werden Datensätze in einem Prozess seriell verarbeitet. Eine lokale Installation kann das Lesen und Verarbeiten parallel ausführen.

Die Größe der Cloud-Datenbank ist beschränkt und richtet sich nach Ihrem Abo.

Die Anzahl der Transaktionen in der Cloud ist beschränkt und richtet sich nach Ihrem Abo.

SDK-Skripte und Transformationsskripte werden in der Cloud isoliert ausgeführt. Dadurch stehen die Delegaten für Verbindungs- und Prozesszugriffe nicht zur Verfügung. Siehe „SDK Referenz“. Außerdem ist die Leistung für die Skriptausführung beschränkt.

Die Syncler API kann nur im eingeschränkten Modus verwendet werden.

KAPITEL 11

Fragen und Antworten (Draft)

Was ist beim Kopieren von Verbindungen zu beachten?

> Das Kopieren schließt gesicherte Daten, wie Passwörter, nicht mit ein. Diese müssen erneut eingegeben werden.

Was ist bei der Gesamtanzahl von zu verarbeitenden Datensätzen zu beachten?

> Die Anzahl wird bei bekannten ID-Werten über eindeutige ID-Werte ermittelt. Falls die Quelle aber Dubletten liefert, die durch den Prozess nicht eliminiert werden können, stimmt die Gesamtanzahl und die Anzahl der verarbeiteten Datensätze nicht mehr überein.

Die Ausführung eines C#-Skripts liefert mir trotz Umwandlung keine UTC-Zeit zurück.

> Der Resultat der Skript-Ausführung durchläuft einen Umwandlungsprozess. Dabei geht die Zeitzonen-Information aus reinen DateTime-Werten verloren und das Resultat wird in die lokale Zeit konvertiert. Um den UTC-Wert zu erhalten können Sie folgendes verwenden.

```
InputValues["Datum"] = new DateTime(Datum.ToUniversalTime().Ticks, DateTimeKind.Local);
```

Die Ausführung eines C#-Skriptes liefert mir die Fehlermeldung „Connection refused“.

> Für die Ausführung von C#-Skripten sind die Ressourcen für Speichernutzung und Ausführungsdauer beschränkt. Sollten die Grenzwerte durch Ihr Skript überschritten werden, führt das zum Abbruch der Ausführung. In diesem Fall sollte das Skript bzgl. Speichernutzung und Laufzeit kontrolliert und ggf. angepasst werden. Einfache Berechnungen können z.B. auch durch Transformationen wie „Formel ausführen“ umgesetzt werden. Sollte dies keine Abhilfe schaffen, wenden Sie sich an unseren Support.

KAPITEL 12

Glossar (Draft)

Änderungsspeicher
Datenabbildungen
Datendienst
Datensatzsperren
Feldnotation
: Eingabeform für die Definition von Schlüssel und Wert in einem Textfeld. Wird häufig in Verbindung mit Datenfeldern angewendet. Kommt auch bei manchen Verbindungen zum Einsatz. Anzahl von Paaren ist nicht beschränkt. Schlüssel und Wert werden durch **|:|** getrennt und mit **|;|** abgeschlossen.

```
Name1|:|Wert1|;|Name2|:|Wert2|;|
```

Protokolle
Prozess
Verbindung
: Repräsentation eines externen Systems oder Funktion. Siehe *Verbindungen*

Warteschlangen