Marketo®

Database/Data Warehouse Integration Architecture

Contents

Introduction

This document provides detailed information regarding implementation of:

- Extraction architecture between Marketo and an external Business Intelligence system (BI)
- Synchronization architecture between Marketo and an external Database/Data Warehouse system (DB)

Entities are described, and the specifics of maintaining synchronization of new and updated records.

Overview

Business Intelligence Integration

This use case answers the question, "How do I get Marketo data into my enterprise BI solution for custom reporting?" BI solutions permit flexible reporting on Marketo marketing data combined with other data sources such as from customer service, operations or finance. Combined data is often analyzed and presented in the form of dashboards that are used to track business metrics.

Marketo recommends implementation of BI integration applications using a one-way polling model, retrieving new records from Marketo, and pushing those records as mapped in the application configuration to the BI system. The application should query for new records periodically and then perform any necessary translation and filtration before pushing the updates to BI data store.

Database Integration

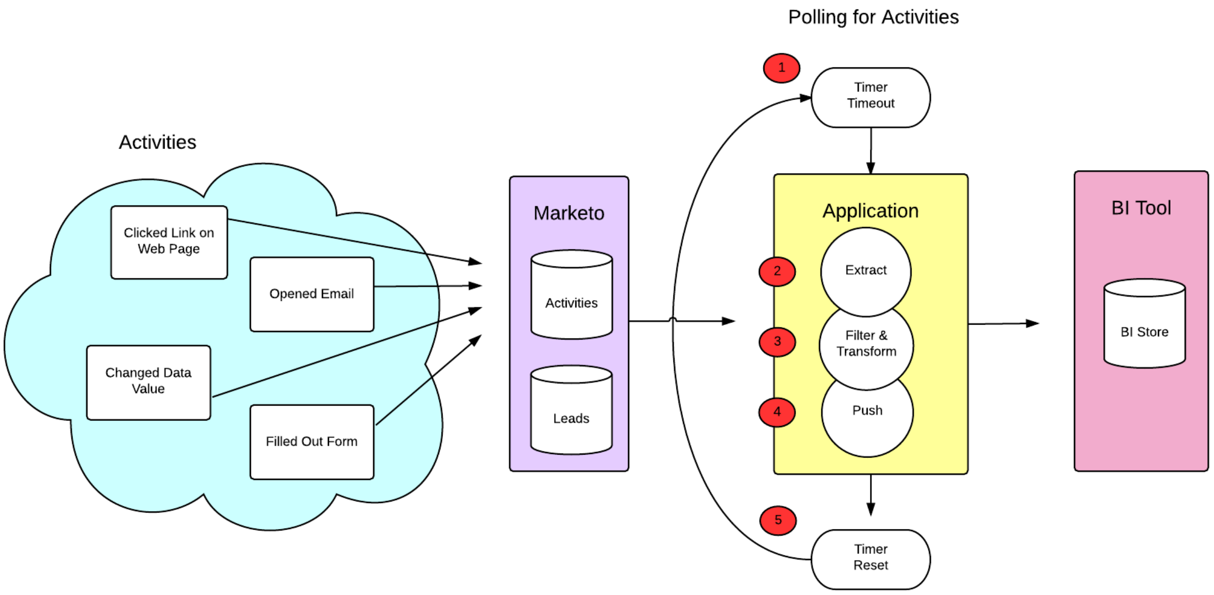
This use case answers the question, "How do I synchronize Marketo with an external Database/Data Warehouse?" A common data management pattern is to maintain a "system of record" (SOR), which serves as the authoritative data source for a given element or piece of information. The SOR is typically a repository where data objects are maintained. Keeping lead/contact data in synch between marketing campaigns and back end system processes ensures consistency. Modeling Marketo custom objects using data objects such as sales orders or order fulfillment can help improve your marketing campaigns by including enriched data in customer communications.

Marketo recommends implementation of DB synchronization applications using a two-way polling model, retrieving changes and new records from either system, and pushing those changes as mapped in the application configuration to the other system. The application should query for changes of selected record types periodically and then perform any necessary translation and filtration before pushing the updates to DB or Marketo respectively.

Business Intelligence Integration

Extraction Process

The simplest and most efficient way to extract data is to implement a polling process that retrieves new activity records from Marketo and pushes data to the BI data store. The cycle is then repeated after a predetermined period each time. This could be implemented as a daily scheduled job for example.



Recommended Extraction Order

1. Extract Activities from Marketo
2. Extract Lead detail for each Activity from Marketo (optional)
3. Update BI data store

Activities are read-only after creation. They are read from Marketo via the **Get Lead Activities** endpoint, in groups of up to 10 activity types. The endpoint accepts an earliest creation date via a paging token which is retrieved via the **Get Paging Token** endpoint. The set of results should be paged through until the moreResult parameter in the response is returned as false, and the corresponding activities written to BI data store. Begin the next extraction cycle by adjusting the earliest creation date and creating a new pagingToken. Move the date forward in time by the desired amount to establish a new earliest creation date. The amount of time selected becomes the polling interval.

Since Marketo maintains only per-second resolution for datetimes, it is possible that the same activity record could be returned in two different extract cycles. To avoid data duplication, the application should use the activity id which is a unique identifier. Duplicate activity records can be safely ignored. If the activity record does not provide all of the lead-related data required to meet your business need, you may perform an additional lookup of the related lead record. To lookup the related lead record, you pass the leadId field from the activity record to **Get Multiple Leads by FilterType**. This endpoint allows you to send a batch of leadIds in a single request. This helps minimize the overall number of API calls required for the lookup. You will then combine activity record data with lead field data. The logic used to combine the two data sets should reflect an inner join operation, using leadId as primary key and as foreign key.

Application Configuration

The application should support a set of configuration options to control program behavior. These options could be stored in a properties file for example.

Fields

Not all fields are necessary to be mapped between Marketo and an associated BI system. An option to enable or disable extraction of specific fields from Lead and Activity entity types is recommended. Only the leadId field is mandatory, while all others should be optional. Reducing the number of extracted fields will improve performance in all cases.

Activity Types

Admin users should be able to select which activity types are and are not extracted to limit API call usage and improve performance.

Polling Interval

The time between extraction batch jobs needs to strike a balance between low data latency, and high utilization of API calls.

Your polling interval should be based on how many API calls a typical extraction cycle will take. The number of new activity records that a client expects to be created will be the greatest influence on the cumulative number of API calls which will be used in a given day. For accounts provisioned or renewed after March 2016, the default number of API calls per day is 50,000. Additional API calls may be purchased in groups of 10,000/day.

Traditional BI presents historical data for manual analysis, so latency is not a concern. In this case a 24 hour interval is appropriate. Real-time BI relies on event driven processing, so latency is a concern. In this case a 5 minute interval is appropriate.

Entities

The following are the Marketo entities that apply for BI integration use case: Leads, Activities.

Leads

Primary Key: id

The integer id of a Marketo lead record and the primary key. This is system managed by Marketo, and may only be assigned by Marketo.

In Marketo, leads represent any person-record which represents a sales or marketing target. All Smart Campaigns (commonly referred to as a “workflow” in non-Marketo systems), filter, trigger, and operate on lead records, based on their characteristics and actions.

Model

Leads are highly extensible in Marketo and may include a large number of custom fields. When extracting data from any particular subscription, the **Describe Lead** endpoint of the REST API should be used as the exclusive source of truth to determine field availability in a particular subscription.

Note that the model for a lead is also potentially dynamic, as fields may be added or hidden by end-users at any time. The application must be resilient to such changes, and not break when they occur.

Relationships

Leads are related to numerous accessible object types in Marketo. For this use case, we are only concerned with Activities.

Object	Relationship Type	Link Type	Link Field
Activities	1:N	Activity Field	leadId

Accessibility

In order to read leads, an API user must have Read-Only Lead permission. They can be read through the following endpoints:

- **Get Multiple Leads by Filter Type**
- **Get Multiple Leads by List Id**
- **Get Multiple Leads by Program Id**

Lead Metadata is read through the **Describe Lead** endpoint.

Activities

Primary Key: id

Activities have a unique integer id that is set upon creation by Marketo.

Activities are records of activities associated to lead records in Marketo. They may record activity of many different types, as indicated by their activityTypeId. Activities are read-only in Marketo. Certain activity types are pruned after 90 days in the Marketo system.

Pruned Activities

- Data Value Change
- Add to List
- Remove from List
- Visit Web Page
- Click Link
- Change Score

The types of activities available in a given subscription vary depending on many factors, including the type of subscription. The available types and their metadata should always be determined by calling **Get Activity Types** endpoint from the target subscription.

Model

Activities have a semi-strict schema. The following fields are defined, but not necessarily used for all activity types:

Name	Datatype	Description
id	Integer	Unique id for activity.
leadId	Integer	Id of the linked lead. Maps to id on lead records.
activityTypeId	Integer	Id of the type of activity. Corresponds to a result of Get Activity Types.
activityDate	Datetime	Date that the activity occurred.
primaryAttributeValue	String	Value of the primary attribute.
primaryAttributeValueId	Integer	Id of the primary attribute.
attributes	Array	Array of name/value pairs representing the secondary attributes of the activity.

Each activity type has a primary attribute that corresponds to a value of some kind. The primary attribute may be related to any type of asset or object in Marketo. For example, the primaryAttributeValue of the Visits Web Page type corresponds to the name or URL of the web page that was visited. The Marketo name will be presented if the page was a landing page and the page URL if it is not a Marketo page. Secondary attributes consist of an array of name/value pairs, naming each of the fields for an activity type and the corresponding value. Continuing with the example, some of the secondary attributes would be Client IP Address, Query Parameters, Referrer URL, and User Agent.

Relationships

Activities in Marketo are always related to lead records through the leadId field. Some activity types may have a relationship to other Marketo assets through their primaryAttributeValue.

Accessibility

In order to read activities, an API user must have the Read-Only Activity permission.

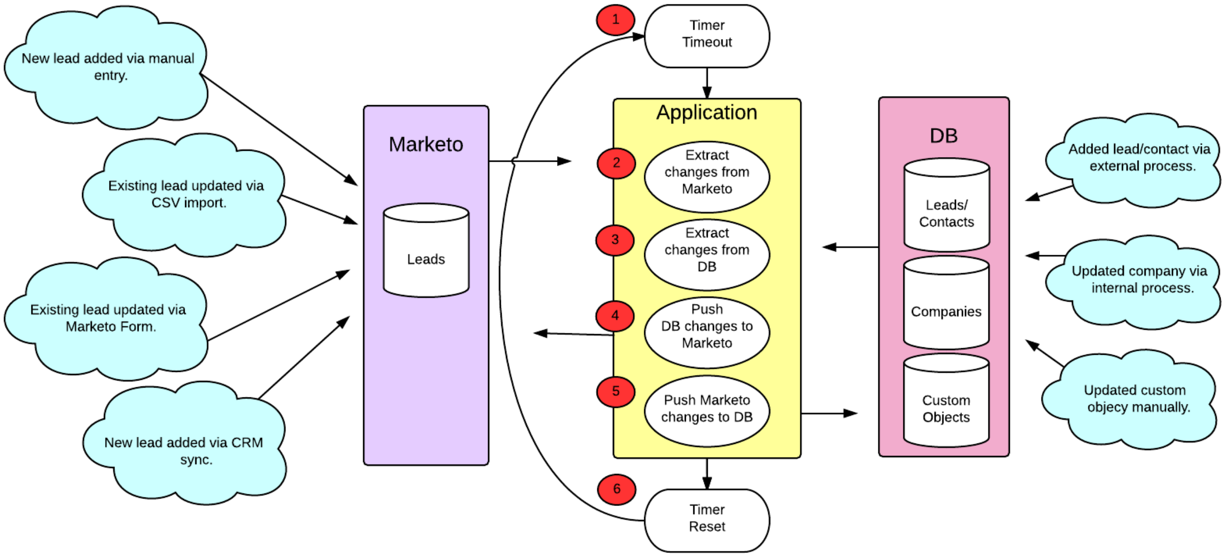
Activities can be read through the **Get Lead Activities**, **Get Lead Changes**, and **Get Deleted Leads** endpoints.

Database Integration

Synchronization Process

The simplest and most efficient way to maintain continuous synchronization is to implement a polling process that retrieves changes to lead records in Marketo and pushes them to DB and then from Lead/Contact, Custom Object, or Company records in DB and pushes them to Marketo. The cycle is then repeated after a predetermined period each time. This could be implemented as a scheduled job for example.

Polling for Lead/User Changes


New lead added via manual entry.
Existing lead updated via CSV import.
Existing lead updated via Marketo Form.
New lead added via CRM sync.
Marketo
Leads
1
Timer Timeout
Application
2
Extract changes from Marketo
3
Extract changes from DB
4
Push DB changes to Marketo
5
Push Marketo changes to DB
6
Timer Reset
DB
Leads/ Contacts
Companies
Custom Objects
Added lead/contact via external process.
Updated company via internal process.
Updated custom objecy manually.

To retrieve changes from Marketo, a high watermark must be maintained for changes to lead/company fields which have occurred since the most recently retrieved change. These are datetime values. When retrieving changes in a subsequent synchronization cycle, the exact datetime of the most recently created record should be used. Since Marketo maintains only per-second resolution for datetimes, it is possible that the same lead change record could be returned in two different extract cycles. To avoid data duplication, the application should use the activity id which uniquely identifies the activity record. Duplicate activity records can be safely ignored.

Recommended Synchronization Order

1. Retrieve Leads changes from Marketo*
2. Retrieve Lead/Contact changes from DB*
3. Retrieve Company changes from DB (optional)*
4. Retrieve Custom Object changes from DB (optional)*
5. Resolve conflicts (optional)
6. Update DB with Marketo Data
7. Update Marketo with DB Data

* Includes new, changed, and deleted records

Leads and Companies

Synchronization for Leads and Companies is maintained primarily using the **Get Lead Changes** endpoint which retrieves data value change records which occur after a timestamp given by a paging token which is retrieved via the **Get Paging Token** endpoint. This endpoint will return both New Lead activities, which indicate the creation of a new known lead in Marketo, and data value change activities for a set of fields given in the parameters of the call.

The change activities should be applied in the order of the createdDate, from earliest to latest, given in the activity to the records in DB which correspond to the record given by leadId in the activity. New Lead activities should be added to a queue of new lead records which need to be retrieved by id, “leadId” in the activity, using the **Get Leads by Filter Type** endpoint, with id as the filterType. These may be retrieved up to 300 at a time. It is recommended to wait until there are 300 records to retrieve, and then to make the call to retrieve these records, instead of calling whenever these become available. If, upon reaching the end of the set of changes, there are less than 300 records, than the set should be retrieved.

In addition to changes and new leads, in order to maintain synchronization the **Merge Lead** and **Delete Lead** activity must be retrieved in order account for lead records which are merged together. The merge activity indicates that two records have been merged into a single record. The DB may or may not choose to honor the merge/delete, delete the losing record and retrieve the changes from the winning lead, or it may be ignored and have a “Deleted in Marketo” flag set to indicate that the Marketo Lead ID for that record is no longer valid.

As part of a standard synchronization cycle, changes from the DB should also be retrieved. Ideally only fields which have been updated since the most recent synch cycle should be retrieved, but this may not be possible given the constraints of the system. If this is available, all the changes for a given record should be aggregated into a lead record to be submitted to Marketo. If a changes-only option is not available, then it is viable to retrieve the whole record with all of the Marketo-mapped fields for submission to Marketo.

To push changes for lead records, use the **Create/Update Leads** endpoint. **Create/Update Leads** allows for the input of up to 300 lead records as JSON.

For incremental synching of updates from DB, the lookupField should be specified as the primary key selected from the DB system, and the createOrUpdate mode should be used. This allows sharing of the same queue by net new leads and lead updates which need to be pushed into Marketo.

In the case that a Marketo Lead is linked to a company record via externalCompanyId, commonly conceptualized as a contact, the company type fields that were part of the lead record are no longer writeable through the lead record and are deferred to the linked company record

Companies

It is important to determine which fields are Company-type fields, and which fields are Lead-type fields. This can be done with the **Describe Company** endpoint. All fields listed there are Company-type fields, of which most are mirrored as lead fields for unlinked leads. If a Company-type field is reflected in a Change Data Value operation, then the change should be reflected against the company record in DB if the change was made against a lead which is linked to a company record via externalCompanyId. If not, the change should just be reflected against the lead record.

Custom Objects

Since the DB is the system of record for Custom Objects, synchronization is one-way from DB to Marketo. The definition of a Custom Object should be derived from the type in DB. Custom Object records should be created, updated, or deleted in Marketo whenever a corresponding event occurs in the DB system. This check should be performed by the integration software upon every synchronization cycle to see if changes are required.

Application Configuration

The application should support a set of configuration options to control program behavior. These options could be stored in a properties file for example.

Fields

Not all fields are necessary to be mapped and synchronized between Marketo and an associated DB. An option to enable or disable synchronization of specific fields from the Lead or Custom Object entities is recommended. Only the leadId and DB foreign key fields should be mandatory for synchronization, while all others should be optional. Reducing the number of synchronized fields will improve performance in all cases.

Custom Objects

An option to enable or disable custom object synchronization is recommended.

Polling Interval

Marketo's native synchronization connectors queue a new batch of pushes and pulls 5 minutes after the completion of the previous batch. This covers a great deal of cases and strikes an acceptable compromise between low synchronization latency, and excessive utilization of API calls. For your solution, you should base the synchronization interval based on how many API calls a typical synchronization cycle will take. For accounts provisioned or renewed after March 2016, the default number of API calls per day is 50,000. Additional API calls purchased in groups of 10,000/day.

The number of changes that a client expects to occur for their lead records will be the greatest influence on the cumulative number of API calls which will be used in a given day, and this should influence your design.

Entities

The following are the Marketo entities that apply for DB integration use case: Leads, Companies, Custom Objects.

Leads

Primary Key: id

The integer id of a Marketo lead record and the primary key. This is system managed by Marketo, and may only be assigned by Marketo. Any insert operations attempted by a foreign system which include id will be rejected.

In Marketo leads represent any person-record which represents a sales or marketing target. All Smart Campaigns (commonly referred to as a workflow in non-Marketo systems), filter, trigger, and operate on lead records, based on their characteristics and actions.

Model

Leads are highly extensible in Marketo and may include a large number of Custom Fields. When synchronizing any particular subscription, a set of standard fields should not be relied upon, and the **Describe Lead** function of the REST API should be used as the exclusive source of truth to determine field availability and updateability in a particular subscription.

Note that the model for a lead is also potentially dynamic, as fields may be added or hidden by end-users at any time. The application must be resilient to such changes, and not break when they occur.

Relationships

Leads are related to numerous accessible object types in Marketo. For this use case, we are only concerned with Companies and Custom Objects.

Object	Relationship Type	Link Type	Link Field
Company	N:1	Lead Field	externalCompanyId
Custom Objects	1:N, N:1, or N:N	See Custom Object Section	See Custom Object Section

Accessibility

Leads may be read and written freely in Marketo given the Read-Write Lead Permission is granted to the API user being used. They can be read through the following endpoints:

- **Get Multiple Leads by Filter Type**
- **Get Multiple Leads by List Id**
- **Get Multiple Leads by Program Id**

They can be written through the following endpoints:

- **Create/Update Leads**
- **Import Lead**

Leads may be deleted through the **Delete Lead** endpoint.

Lead Metadata is read through the **Describe Lead** endpoint.

Companies

Primary Key: externalCompanyId, id

externalCompanyId is an arbitrary string field set upon creation by the external system. The primary key of company records in DB should be mapped to externalCompanyId, which is not updatable and must be unique. Id is a unique system-generated integer id.

Company objects represent the organization to which lead records belong. Leads are added to a Company by populating their corresponding externalCompanyId. Leads linked to a company record will directly inherit the values from a company record as though the values existed on the lead's own record. Attributes available on the company record are available for triggering and filtering on lead records from within the application.

Company records may only be created by external systems, and the DB should be treated as the source of truth.

Model

- Companies are fully extensible and may have any number and type of custom fields
- **Describe Company** should be used to obtain the schema of company

Relationships

Object	Relationship Type	Link Type	Link Field
Lead	1:N	Lead Field	externalCompanyId
Opportunity	1:N	Opportunity Field	externalCompanyId

Accessibility

In order to read and write to company records, an API user must have been granted the Read-Write Company permission.

Companies may be read through a single endpoint, **Get Companies by Filter Type**. Companies may only be filtered on a limited number of fields, which are provided in the searchableFields attribute of the **Describe Company** result.

Companies can be written to using the **Create/Update Companies** endpoint.

Companies may be deleted through the **Delete Companies** endpoint.

Custom Objects

Primary Key: marketoGUID, Additional User Defined keys

Custom Objects always have a unique system-generated marketoGUID which is set upon creation. There will be at least one additional key, and possibly more, which are user-defined in the Custom Object definition. Keys may be single fields for types which are linked directly to Leads or Companies, and may have compound keys for types which are linked to Leads and another Custom Object type. Keys can be determined by using **Describe Custom Object** to retrieve the list of dedupeFields.

Marketo allows the definition of Custom Object types by users to extend the Marketo schema. Marketo Custom Objects may be related to Leads or Companies in either a 1:N or N:N configuration through the usage of intermediate custom objects.

Model

Note that the schema for custom objects within a DB is potentially dynamic, as fields may be added or removed at any time. The application must be resilient to such changes, and not break when they occur.

Relationships

Marketo Custom Objects may be related to leads or companies in either a 1:N or N:N configuration through the usage of intermediate custom objects. Relationships can be derived from the relationships parameter of the result of **Describe Custom Object**.

Accessibility

To read and write to Custom Objects, an API user must have the Read-Write Custom Object permission. Custom Objects are manipulated using the following endpoints: **Get Custom Objects**, **Create/Update/Upsert Custom Objects**, **and Delete Custom Objects**. A **List Custom Objects** endpoint is also provided to a give a means of determining what Custom Object types are available in a given subscription.

Links to Related Material

- REST API reference: Marketo Developers Site
- REST API error codes: Marketo Developers Site Error Codes
- Best practices and code samples: Marketo Developers Blog