



TIBCO® Data Virtualization

Marketo Adapter Guide

Version 8.7.0 | October 2023

Contents

Contents	2
Marketo Adapter	5
Getting Started	5
Basic Tab	6
Logging	7
Changelog	9
Advanced Features	14
User Defined Views	15
SSL Configuration	18
Firewall and Proxy	18
Query Processing	19
Logging	19
SQL Compliance	22
SELECT Statements	24
SELECT INTO Statements	26
INSERT Statements	26
UPDATE Statements	27
UPSERT Statements	28
DELETE Statements	29
EXECUTE Statements	30
PIVOT and UNPIVOT	30
SOAP Data Model	31
Tables	32
Leads	32
Programs	44
Views	46
Activities	49

Activities_AddToList	51
Activities_AddToSFDCCampaign	53
Activities_ChangeDataValue	55
Activities_ChangeOwner	57
Activities_ChangeRevenueStage	59
Activities_ChangeScore	61
Activities_ChangeStatusInProgression	63
Activities_ChangeStatusInSFDCCampaign	65
Activities_ClickEmail	67
Activities_ClickLink	70
Activities_ConvertLead	72
Activities_CreateTask	74
Activities_EmailBouncedSoft	77
Activities_EmailDelivered	79
Activities_InterestingMoments	81
Activities_MergeLeads	83
Activities_NewLeads	85
Activities_OpenEmail	87
Activities_RemoveFromFlow	89
Activities_RemoveFromList	91
Activities_RemoveFromSFDCCampaign	93
Activities_SendAlert	95
Activities_SendEmail	97
Activities_SFDCActivity	99
Activities_SFDCMergeLeads	102
Activities_VisitWebpage	103
Campaigns	106
Channels	107
Opportunities	108
OpportunityPersonRoles	110
Tags	112
REST Data Model	113

Tables	113
Views	182
Stored Procedures	217
Connection String Options	261
REST	265
SOAP	266
Schema	268
OAuth	269
SSL	274
Firewall	276
Proxy	279
Logging	286
Miscellaneous	286
TIBCO Product Documentation and Support Services	294
How to Access TIBCO Documentation	294
How to Contact TIBCO Support	295
Release Version Support	295
How to Join TIBCO Community	296
Legal and Third-Party Notices	297

Marketo Adapter

Marketo Version Support

The adapter enables SQL92 access to the entities available through version 1 of the REST API and version 2.6 of the SOAP API.

SQL Compliance

The [SQL Compliance](#) section shows the SQL syntax supported by the adapter and points out any limitations.

Getting Started

Connecting to Marketo

[Basic Tab](#) shows how to authenticate to Marketo and configure any necessary connection properties. Additional adapter capabilities can be configured using the available [Connection](#) properties on the Advanced tab. The Advanced Settings section shows how to set up more advanced configurations and troubleshoot connection errors.

Deploying the Marketo Adapter

To deploy the adapter, you can execute the `server_util` utility via the command line by

1. Unzip the `tdv.marketo.zip` file to the location of your choice.
2. Open a command prompt window.
3. Navigate to the `<TDV_install_dir>/bin`
4. Enter the `server_util` command with the `-deploy` option:

```
server_util -server <hostname> [-port <port>] -user <user> -  
password <password> -deploy -package <TDV_install_  
dir>/adapters/tdv.marketo/tdv.marketo.jar
```

Note: When deploying a build of an existing adapter, you will need to undeploy the existing adapter using the `server_util` command with the `-undeploy` option.

```
server_util -server <hostname> [-port <port>] -user <user> -password  
<password> -undeploy -version 1 -name Marketo
```

Basic Tab

Both the REST and SOAP APIs are supported and can be chosen using the [Schema](#) property.

Connecting to the Marketo REST API

Before you can connect to the Marketo REST API, you will need to create a custom service.

Creating a Custom Service

To create a custom service, follow the procedure below:

1. Navigate to the admin area of your Marketo application.
2. Click **Users & Roles** in the Security section.
3. Select the **Roles** tab and click **New Role** to create a new Role.
4. Enter a Role Name and select the permissions for the Role. The Access API permissions are specific to the REST API.
5. Now that an API Role is created, select the **Users** tab and click Invite New User.
6. Enter the new user information and select the role that was just created with API access. The API Only option can be selected to denote the user as an API Only user.
7. Now that a new user has been created, a new service will need to be created. Click the **LaunchPoint** option (Admin -> Integration -> LaunchPoint).
8. Click **New Service**.
9. Select the Custom service type and enter a display name and description.
10. Select the user you created.

Obtaining OAuth Credentials

To obtain the OAuthClientId and OAuthClientSecret, navigate to the **LaunchPoint** option on the Admin area.

Click the **View Details** link for the desired service. A window containing the authentication credentials is displayed.

Authenticating to the REST API

Once you have created a custom service and obtained your OAuth credentials, set the following to connect to data:

- Schema: Set this to **REST**.
- OAuthClientId: The OAuth Client ID associated with your custom service.
- OAuthClientSecret: The OAuth Client Secret associated with your custom service.
- RESTEndpoint: The URL of the REST Web service endpoint. This can be found on your Marketo Admin area on the Integration -> Web Services option in the REST API section.

The Identity Endpoint will not be needed.

Connecting to the Marketo SOAP API

To connect to the SOAP API, you will need to provide valid Marketo credentials. Specify the following to connect to data:

- Schema: Set this to **SOAP**.
- UserId: The client access ID is found within your Marketo admin SOAP API panel under Integration.
- EncryptionKey: The Marketo SOAP API Encryption Key. This key is generated on the Admin page of the Marketo website.
- SOAPEndpoint: The URL of the SOAP Web service endpoint, provided by Marketo on the Admin page of the Marketo website.

Logging

The adapter uses TDV Server's logging (log4j) to generate log files. The settings within the TDV Server's logging (log4j) configuration file are used by the adapter to determine the type of messages to log. The following categories can be specified:

- Error: Only error messages are logged.
- Info: Both Error and Info messages are logged.
- Debug: Error, Info, and Debug messages are logged.

The Other property of the adapter can be used to set Verbosity to specify the amount of detail to be included in the log file, that is:

```
Verbosity=4;
```

You can use Verbosity to specify the amount of detail to include in the log within a category. The following verbosity levels are mapped to the log4j categories:

- 0 = Error
- 1-2 = Info
- 3-5 = Debug

For example, if the log4j category is set to DEBUG, the Verbosity option can be set to 3 for the minimum amount of debug information or 5 for the maximum amount of debug information.

Note that the log4j settings override the Verbosity level specified. The adapter never logs at a Verbosity level greater than what is configured in the log4j properties. In addition, if Verbosity is set to a level less than the log4j category configured, Verbosity defaults to the minimum value for that particular category. For example, if Verbosity is set to a value less than 3 and the Debug category is specified, the Verbosity defaults to 3.

The following list is an explanation of the Verbosity levels and the information that they log.

- 1 - Will log the query, the number of rows returned by it, the start of execution and the time taken, and any errors.
- 2 - Will log everything included in Verbosity 1 and HTTP headers.
- 3 - Will additionally log the body of the HTTP requests.
- 4 - Will additionally log transport-level communication with the data source. This includes SSL negotiation.

- 5 - Will additionally log communication with the data source and additional details that may be helpful in troubleshooting problems. This includes interface commands.

Configure Logging for the Marketo Adapter

By default, logging is turned on without debugging. If debugging information is desired, uncomment the following line in the TDV Server's log4j.properties file (default location of this file is: C:\Program Files\TIBCO\TDV Server <version>\conf\server):

```
log4j.logger.com.cdata=DEBUG
```

The TDV Server must be restarted after changing the log4j.properties file, which can be accomplished by running the composite.bat script located at: C:\Program Files\TIBCO\TDV Server <version>\bin. Note that reauthenticating to the TDV Studio is required after restarting the server.

Here is an example of the calls:

```
.\composite.bat monitor restart
```

All logs for the adapter are written to the "cs_server_dsrc.log" file as specified in the log4j properties.

Note: The "log4j.logger.com.cdata=DEBUG" option is not required if the **Debug Output Enabled** option is set to true within the TDV Studio. To set this option, navigate to **Administrator > Configuration**. Select **Server > Configuration > Debugging** and set the Debug Output Enabled option to **True**.

Changelog

General Changes

Date	Build Number	Change Type	Description
12/14/2022	8383	General	Changed

			<ul style="list-style-type: none"> Added the Default column to the sys_procedureparameters table.
11/09/2022	8348	Marketo	Changed <ul style="list-style-type: none"> The FileSize column datatype has been updated from integer to long for ActivityBulkExports table.
10/27/2022	8335	Marketo	Changed <ul style="list-style-type: none"> The ActivityId column datatype has been updated from integer to string for Activities as well as custom tables as per API.
10/14/2022	8322	Marketo	Added <ul style="list-style-type: none"> Added Content as an input parameter to support inputstream in CreateFile, ImportCustomObjectsFromFile, ImportLeadsFromFile, ImportProgramMembersFromFile stored procedures. Added the FileStream parameter to support outputstream in GetExportJobFile, GetImportCustomObjectFailures, GetImportCustomObjectWarnings, GetImportLeadsFailures, GetImportLeadsWarnings, GetImportProgramMembersFailures, GetImportProgramMembersWarnings stored procedures. Added the FileData output parameter and Encoding input parameter to print the response in GetExportJobFile, GetImportCustomObjectFailures, GetImportCustomObjectWarnings, GetImportLeadsFailures, GetImportLeadsWarnings, GetImportProgramMembersFailures,

GetImportProgramMembersWarnings stored procedures.			
09/30/2022	8308	General	Changed <ul style="list-style-type: none"> Added the IsPath column to the sys_procedureparameters table.
08/17/2022	8264	General	Changed <ul style="list-style-type: none"> We now support handling the keyword "COLLATE" as standard function name as well.
08/01/2022	8245	Marketo	Added <ul style="list-style-type: none"> Added Activities view to the Rest schema.
06/07/2022	8194	Marketo	Added <ul style="list-style-type: none"> Added StaticLists and ListStaticMemberShip table.
10/29/2021	7972	Marketo	Added <ul style="list-style-type: none"> Added the EarliestUpdatedAt and LatestUpdatedAt pseudo-columns to Emails table.
09/02/2021	7915	General	Added <ul style="list-style-type: none"> Added support for the STRING_SPLIT table-valued function in the CROSS APPLY clause.
08/07/2021	7889	General	Changed <ul style="list-style-type: none"> Added the KeySeq column to the sys_foreignkeys table.
08/06/2021	7888	General	Changed <ul style="list-style-type: none"> Added the new sys_primarykeys system table.

07/23/2021	7874	General	Changed <ul style="list-style-type: none"> Updated the Literal Function Names for relative date/datetime functions. Previously relative date/datetime functions resolved to a different value when used in the projection vs the predicate. I.e: SELECT LAST_MONTH() AS lm, Col FROM Table WHERE Col > LAST_MONTH(). Formerly the two LAST_MONTH() methods would resolve to different datetimes. Now they will match. As a replacement for the previous behavior, the relative date/datetime functions in the criteria may have an 'L' appended to them. I.e: WHERE col > L_LAST_MONTH(). This will continue to resolve to the same values that previously were calculated in the criteria. Note that the "L_" prefix will only work in the predicate - it not available for the projection.
07/08/2021	7859	General	Added <ul style="list-style-type: none"> Added the TCP Logging Module for the logging information happening on the TCP wire protocol. The transport bytes that are incoming and ongoing will be logged at verbosity=5.
04/23/2021	7785	General	Added <ul style="list-style-type: none"> Added support for handling client side formulas during insert / update. For example: UPDATE Table SET Col1 = Concat (Col1, " - ", Col2) WHERE Col2 LIKE 'A%'
04/23/2021	7783	General	Changed <ul style="list-style-type: none"> Updated how display sizes are determined for varchar primary key and foreign key

			columns so they will match the reported length of the column.
04/16/2021	7776	General	<p>Added</p> <ul style="list-style-type: none"> Non-conditional updates between two columns is now available to all drivers. For example: UPDATE Table SET Col1=Col2 <p>Changed</p> <ul style="list-style-type: none"> Reduced the length to 255 for varchar primary key and foreign key columns. Updated implicit and metadata caching to improve performance and support for multiple connections. Old metadata caches are not compatible - you would need to generate new metadata caches if you are currently using CacheMetadata. Updated index naming convention to avoid duplicates Updated and standardized Getting Started connection help. Added the Advanced Features section to the help of all drivers. Categorized connection property listings in the help for all editions.
04/15 /2021	7775	General	<p>Changed</p> <ul style="list-style-type: none"> Kerberos authentication is updated to use TCP by default, but will fall back to UDP if a TCP connection cannot be established
06/30/2021	7766	Marketo	<p>Added</p> <ul style="list-style-type: none"> The View ProgramMembers is converted from a view to a table. Insert, update and delete operation are supported in case of UseBulkApi=false.

Added

- Added following stored procedures :
AssociateLead, CreateFile, MergeLeads, AddLandingPageContentSection, CreateEmailTemplate, DeleteLandingPageContentSection, ScheduleCampaign, UpdateEmailContent, UpdateEmailFullContent, UpdateEmailTemplateContent, UpdateFile, UpdateLandingPageContentSection, UpdateLandingPageTemplateContent and UpdateLeadProgramStatus.
 - Added following tables and views :
WeeklyUsageStatistics, WeeklyErrorStatistics, ThankYouList, Snippets, SnippetContent, Segments, Segmentations, LeadPrograms, LeadLists, LeadChangesFields, LeadChanges, LeadChangesAttributes, LandingPageTemplates, LandingPageTemplateContent, LandingPages, LandingPageContentSection, Forms, EmailTemplates, EmailCCFields, DailyUsageStatistics, DailyErrorStatistics, ActivityTypes and ActivityTypeAttributes.
-

Advanced Features

This section details a selection of advanced features of the Marketo adapter.

User Defined Views

The adapter allows you to define virtual tables, called *user defined views*, whose contents are decided by a pre-configured query. These views are useful when you cannot directly control queries being issued to the drivers. See [User Defined Views](#) for an overview of creating and configuring custom views.

SSL Configuration

Use [SSL Configuration](#) to adjust how adapter handles TLS/SSL certificate negotiations. You can choose from various certificate formats; see the [SSLServerCert](#) property under "Connection String Options" for more information.

Firewall and Proxy

Configure the adapter for compliance with [Firewall and Proxy](#), including Windows proxies and HTTP proxies. You can also set up tunnel connections.

Query Processing

The adapter offloads as much of the SELECT statement processing as possible to Marketo and then processes the rest of the query in memory (client-side).

See [Query Processing](#) for more information.

Logging

See [Logging](#) for an overview of configuration settings that can be used to refine CData logging. For basic logging, you only need to set two connection properties, but there are numerous features that support more refined logging, where you can select subsets of information to be logged using the [LogModules](#) connection property.

User Defined Views

The Marketo Adapter allows you to define a virtual table whose contents are decided by a pre-configured query. These are called *User Defined Views*, which are useful in situations where you cannot directly control the query being issued to the driver, e.g. when using the driver from a tool. The User Defined Views can be used to define predicates that are always applied. If you specify additional predicates in the query to the view, they are combined with the query already defined as part of the view.

There are two ways to create user defined views:

- Create a JSON-formatted configuration file defining the views you want.
- DDL statements.

Defining Views Using a Configuration File

User Defined Views are defined in a JSON-formatted configuration file called *UserDefinedViews.json*. The adapter automatically detects the views specified in this file.

You can also have multiple view definitions and control them using the UserDefinedViews connection property. When you use this property, only the specified views are seen by the adapter.

This User Defined View configuration file is formatted as follows:

- Each root element defines the name of a view.
- Each root element contains a child element, called **query**, which contains the custom SQL query for the view.

For example:

```
{
  "MyView": {
    "query": "SELECT * FROM Leads WHERE MyColumn = 'value'"
  },
  "MyView2": {
    "query": "SELECT * FROM MyTable WHERE Id IN (1,2,3)"
  }
}
```

Use the UserDefinedViews connection property to specify the location of your JSON configuration file. For example:

```
"UserDefinedViews",
"C:\\Users\\yourusername\\Desktop\\tmp\\UserDefinedViews.json"
```

Defining Views Using DDL Statements

The adapter is also capable of creating and altering the schema via DDL Statements such as CREATE LOCAL VIEW, ALTER LOCAL VIEW, and DROP LOCAL VIEW.

Create a View

To create a new view using DDL statements, provide the view name and query as follows:

```
CREATE LOCAL VIEW [MyViewName] AS SELECT * FROM Customers LIMIT 20;
```


If no JSON file exists, the above code creates one. The view is then created in the JSON configuration file and is now discoverable. The JSON file location is specified by the UserDefinedViews connection property.

Alter a View

To alter an existing view, provide the name of an existing view alongside the new query you would like to use instead:

```
ALTER LOCAL VIEW [MyViewName] AS SELECT * FROM Customers WHERE  
TimeModified > '3/1/2020';
```

The view is then updated in the JSON configuration file.

Drop a View

To drop an existing view, provide the name of an existing schema alongside the new query you would like to use instead.

```
DROP LOCAL VIEW [MyViewName]
```

This removes the view from the JSON configuration file. It can no longer be queried.

Schema for User Defined Views

User Defined Views are exposed in the **UserViews** schema by default. This is done to avoid the view's name clashing with an actual entity in the data model. You can change the name of the schema used for UserViews by setting the UserViewsSchemaName property.

Working with User Defined Views

For example, a SQL statement with a User Defined View called *UserViews.RCustomers* only lists customers in Raleigh:

```
SELECT * FROM Customers WHERE City = 'Raleigh';
```

An example of a query to the driver:

```
SELECT * FROM UserViews.RCustomers WHERE Status = 'Active';
```

Resulting in the effective query to the source:

```
SELECT * FROM Customers WHERE City = 'Raleigh' AND Status = 'Active';
```

That is a very simple example of a query to a User Defined View that is effectively a combination of the view query and the view definition. It is possible to compose these queries in much more complex patterns. All SQL operations are allowed in both queries and are combined when appropriate.

SSL Configuration

Customizing the SSL Configuration

By default, the adapter attempts to negotiate SSL/TLS by checking the server's certificate against the system's trusted certificate store.

To specify another certificate, see the [SSLServerCert](#) property for the available formats to do so.

Firewall and Proxy

Connecting Through a Firewall or Proxy

HTTP Proxies

To connect through the Windows system proxy, you do not need to set any additional connection properties. To connect to other proxies, set [ProxyAutoDetect](#) to false.

In addition, to authenticate to an HTTP proxy, set [ProxyAuthScheme](#), [ProxyUser](#), and [ProxyPassword](#), in addition to [ProxyServer](#) and [ProxyPort](#).

Other Proxies

Set the following properties:

- To use a proxy-based firewall, set [FirewallType](#), [FirewallServer](#), and [FirewallPort](#).
- To tunnel the connection, set [FirewallType](#) to TUNNEL.

- To authenticate, specify FirewallUser and FirewallPassword.
- To authenticate to a SOCKS proxy, additionally set FirewallType to SOCKS5.

Query Processing

Query Processing

CData has a client-side SQL engine built into the adapter library. This enables support for the full capabilities that SQL-92 offers, including filters, aggregations, functions, etc.

For sources that do not support SQL-92, the adapter offloads as much of SQL statement processing as possible to Marketo and then processes the rest of the query in memory (client-side). This results in optimal performance.

For data sources with limited query capabilities, the adapter handles transformations of the SQL query to make it simpler for the adapter. The goal is to make smart decisions based on the query capabilities of the data source to push down as much of the computation as possible. The Marketo Query Evaluation component examines SQL queries and returns information indicating what parts of the query the adapter is not capable of executing natively.

The Marketo Query Slicer component is used in more specific cases to separate a single query into multiple independent queries. The client-side Query Engine makes decisions about simplifying queries, breaking queries into multiple queries, and pushing down or computing aggregations on the client-side while minimizing the size of the result set.

There's a significant trade-off in evaluating queries, even partially, client-side. There are always queries that are impossible to execute efficiently in this model, and some can be particularly expensive to compute in this manner. CData always pushes down as much of the query as is feasible for the data source to generate the most efficient query possible and provide the most flexible query capabilities.

More Information

For a full discussion of how CData handles query processing, see [CData Architecture: Query Execution](#).

Logging

Capturing adapter logging can be very helpful when diagnosing error messages or other unexpected behavior.

Basic Logging

You will simply need to set two connection properties to begin capturing adapter logging.

- Logfile: A filepath which designates the name and location of the log file.
- Verbosity: This is a numerical value (1-5) that determines the amount of detail in the log. See the page in the Connection Properties section for an explanation of the five levels.
- MaxLogFileSize: When the limit is hit, a new log is created in the same folder with the date and time appended to the end. The default limit is 100 MB. Values lower than 100 kB will use 100 kB as the value instead.
- MaxLogFileCount: A string specifying the maximum file count of log files. When the limit is hit, a new log is created in the same folder with the date and time appended to the end and the oldest log file will be deleted. Minimum supported value is 2. A value of 0 or a negative value indicates no limit on the count.

Once this property is set, the adapter will populate the log file as it carries out various tasks, such as when authentication is performed or queries are executed. If the specified file doesn't already exist, it will be created.

Log Verbosity

The verbosity level determines the amount of detail that the adapter reports to the Logfile. Verbosity levels from 1 to 5 are supported. These are described in the following list:

- | | |
|---|---|
| 1 | Setting <u>Verbosity</u> to 1 will log the query, the number of rows returned by it, the start of execution and the time taken, and any errors. |
| 2 | Setting <u>Verbosity</u> to 2 will log everything included in <u>Verbosity</u> 1 and additional information about the request. |
| 3 | Setting <u>Verbosity</u> to 3 will additionally log HTTP headers, as well as the body of the request and the response. |
| 4 | Setting <u>Verbosity</u> to 4 will additionally log transport-level communication with the data |

source. This includes SSL negotiation.

- 5 Setting Verbosity to 5 will additionally log communication with the data source and additional details that may be helpful in troubleshooting problems. This includes interface commands.
-

The Verbosity should not be set to greater than 1 for normal operation. Substantial amounts of data can be logged at higher verbosity levels, which can delay execution times.

To refine the logged content further by showing/hiding specific categories of information, see LogModules.

Sensitive Data

Verbosity levels 3 and higher may capture information that you do not want shared outside of your organization. The following lists information of concern for each level:

- Verbosity 3: The full body of the request and the response, which includes all the data returned by the adapter
- Verbosity 4: SSL certificates
- Verbosity 5: Any extra transfer data not included at Verbosity 3, such as non human-readable binary transfer data

Best Practices for Data Security

Although we mask sensitive values, such as passwords, in the connection string and any request in the log, it is always best practice to review the logs for any sensitive information before sharing outside your organization.

Java Logging

When Java logging is enabled in Logfile, the Verbosity will instead map to the following logging levels.

- 0: Level.WARNING
- 1: Level.INFO
- 2: Level.CONFIG

- 3: Level.FINE
- 4: Level.FINER
- 5: Level.FINEST

Advanced Logging

You may want to refine the exact information that is recorded to the log file. This can be accomplished using the LogModules property.

This property allows you to filter the logging using a semicolon-separated list of logging modules.

All modules are four characters long. **Please note that modules containing three letters have a required trailing blank space.** The available modules are:

- **EXEC:** Query Execution. Includes execution messages for original SQL queries, parsed SQL queries, and normalized SQL queries. Query and page success/failure messages appear here as well.
- **INFO:** General Information. Includes the connection string, driver version (build number), and initial connection messages.
- **HTTP:** HTTP Protocol messages. Includes HTTP requests/responses (including POST messages), as well as Kerberos related messages.
- **SSL :** SSL certificate messages.
- **OAUT:** OAuth related failure/success messages.
- **SQL :** Includes SQL transactions, SQL bulk transfer messages, and SQL result set messages.
- **META:** Metadata cache and schema messages.
- **TCP :** Incoming and Ongoing raw bytes on TCP transport layer messages.

An example value for this property would be.

```
LogModules=INFO;EXEC;SSL ;SQL ;META;
```

Note that these modules refine the information as it is pulled after taking the Verbosity into account.

SQL Compliance

The Marketo Adapter supports several operations on data, including querying, deleting, modifying, and inserting.

SELECT Statements

See [SELECT Statements](#) for a syntax reference and examples.

See [REST Data Model](#) for information on the capabilities of the Marketo API.

INSERT Statements

See [INSERT Statements](#) for a syntax reference and examples, as well as retrieving the new records' Ids.

UPDATE Statements

The primary key Id is required to update a record. See [UPDATE Statements](#) for a syntax reference and examples.

UPSERT Statements

An UPSERT updates a record if it exists and inserts the record if it does not. See [UPSERT Statements](#) for a syntax reference and examples.

DELETE Statements

The primary key Id is required to delete a record. See [DELETE Statements](#) for a syntax reference and examples.

GETDELETED Statements

GETDELETED statements return the Ids of deleted records. See [GETDELETED Statements](#) for a syntax reference and examples.

EXECUTE Statements

Use EXECUTE or EXEC statements to execute stored procedures. See [EXECUTE Statements](#) for a syntax reference and examples.

Names and Quoting

- Table and column names are considered identifier names; as such, they are restricted to the following characters: [A-Z, a-z, 0-9, _:@].
- To use a table or column name with characters not listed above, the name must be quoted using double quotes ("name") in any SQL statement.
- Strings must be quoted using single quotes (e.g., 'John Doe').

SELECT Statements

A SELECT statement can consist of the following basic clauses.

- SELECT
- INTO
- FROM
- JOIN
- WHERE
- GROUP BY
- HAVING
- UNION
- ORDER BY
- LIMIT

SELECT Syntax

The following syntax diagram outlines the syntax supported by the Marketo adapter:

```
SELECT {
  [ TOP <numeric_literal> ]
  {
    *
    | {
      <expression> [ [ AS ] <column_reference> ]
      | { <table_name> | <correlation_name> } .*
    } [ , ... ]
  }
}
```



```

}
[ INTO csv:// [ filename= ] <file_path> [ ;delimiter=tab ] ]
{
  FROM <table_reference> [ [ AS ] <identifier> ]
}
[ WHERE <search_condition> ]
[
  LIMIT <expression>
]
} | SCOPE_IDENTITY()
<expression> ::=
  | <column_reference>
  | @ <parameter>
  | ?
  | COUNT( * | { [ DISTINCT ] <expression> } )
  | { AVG | MAX | MIN | SUM | COUNT } ( <expression> )
  | NULLIF ( <expression> , <expression> )
  | COALESCE ( <expression> , ... )
  | CASE <expression>
      WHEN { <expression> | <search_condition> } THEN { <expression> |
NULL } [ ... ]
      [ ELSE { <expression> | NULL } ]
      END
  | <literal>
  | <sql_function>
<search_condition> ::=
  {
    <expression> { = | > | < | >= | <= | <> | != | LIKE | AND | OR |
IN } [ <expression> ]
    } [ { AND | OR } ... ]

```

Examples

1. Return all columns:

```
SELECT * FROM Leads
```

2. Rename a column:

```
SELECT "Email" AS MY_Email FROM Leads
```

3. Cast a column's data as a different data type:

```
SELECT CAST(AnnualRevenue AS VARCHAR) AS Str_AnnualRevenue FROM
Leads
```

4. Search data:

```
SELECT * FROM Leads WHERE Email = 'sample@email.com'
```

5. The Marketo APIs support the following operators in the WHERE clause: =, >, <, >=, <=, <>, !=, LIKE, AND, OR, IN.

```
SELECT * FROM Leads WHERE Email = 'sample@email.com';
```

SELECT INTO Statements

You can use the SELECT INTO statement to export formatted data to a file.

Data Export with an SQL Query

The following query exports data into a file formatted in comma-separated values (CSV):

```
boolean ret = stat.execute("SELECT Id, Email INTO 'csv://c:/Leads.txt'
FROM 'Leads' WHERE Email = 'sample@email.com'");
System.out.println(stat.getUpdateCount()+" rows affected");
```

You can specify other file formats in the URI. The following example exports tab-separated values:

```
Statement stat = conn.createStatement();
boolean ret = stat.execute("SELECT * INTO 'Leads' IN
'csv://filename=c:/Leads.csv;delimiter=tab' FROM 'Leads' WHERE Email =
'sample@email.com'");
System.out.println(stat.getUpdateCount()+" rows affected");
```

INSERT Statements

To create new records, use INSERT statements.

INSERT Syntax

The INSERT statement specifies the columns to be inserted and the new column values. You can specify the column values in a comma-separated list in the VALUES clause, as shown in the following example:

```
INSERT INTO <table_name>
( <column_reference> [ , ... ] )
VALUES
( { <expression> | NULL } [ , ... ] )

<expression> ::=
| @ <parameter>
| ?
| <literal>
```

You can use the executeUpdate method of the Statement and PreparedStatement classes to execute data manipulation commands and retrieve the rows affected. To retrieve the Id of the last inserted record use getGeneratedKeys. Additionally, set the **RETURN_GENERATED_KEYS** flag of the Statement class when you call prepareStatement.

```
String cmd = "INSERT INTO Leads (Email) VALUES (?)";
PreparedStatement pstmt = connection.prepareStatement
(cmd,Statement.RETURN_GENERATED_KEYS);
pstmt.setString(1, "John");
int count = pstmt.executeUpdate();
System.out.println(count+" rows were affected");
ResultSet rs = pstmt.getGeneratedKeys();
while(rs.next()){
    System.out.println(rs.getString("Id"));
}
connection.close();
```

UPDATE Statements

To modify existing records, use UPDATE statements.

Update Syntax

The UPDATE statement takes as input a comma-separated list of columns and new column values as name-value pairs in the SET clause, as shown in the following example:

```

UPDATE <table_name> SET { <column_reference> = <expression> } [ , ... ]
WHERE { Id = <expression> } [ { AND | OR } ... ]
<expression> ::=
    | @ <parameter>
    | ?
    | <literal>

```

You can use the `executeUpdate` method of the `Statement` or `PreparedStatement` classes to execute data manipulation commands and retrieve the rows affected, as shown in the following example:

```

String cmd = "UPDATE Leads SET Email='John' WHERE Id = ?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "c2ef66a5-a545-413b-9312-79a53caadb4");
int count = pstmt.executeUpdate();
System.out.println(count + " rows were affected");
connection.close();

```

UPSERT Statements

An UPSERT statement updates an existing record or creates a new record if an existing record is not identified.

UPSERT Syntax

The UPSERT syntax is the same as for insert. Marketo uses the input provided in the `VALUES` clause to determine whether the record already exists. If the record does not exist, all columns required to insert the record must be specified. See [REST Data Model](#) for any table-specific information.

```

UPSERT INTO <table_name>
( <column_reference> [ , ... ] )
VALUES
( { <expression> | NULL } [ , ... ] )

<expression> ::=
    | @ <parameter>
    | ?
    | <literal>

```

You can use the `executeUpdate` method of the `Statement` and `PreparedStatement` classes to issue data manipulation commands and retrieve the rows affected, as shown in the following example: To retrieve the `Id` of the last inserted record, use `getGeneratedKeys`.

Additionally, set the RETURN_GENERATED_KEYS flag of the Statement class when you call prepareStatement.

```
String cmd = "UPSERT INTO Leads (Email) VALUES (?)";
PreparedStatement pstmt = connection.prepareStatement(
    cmd, Statement.RETURN_GENERATED_KEYS);
pstmt.setString(1, "John");
int count = pstmt.executeUpdate();
System.out.println(count+" rows were affected");
ResultSet rs = pstmt.getGeneratedKeys();
while(rs.next()){
    System.out.println(rs.getString("Id"));
}
connection.close();
```

DELETE Statements

To delete information from a table, use DELETE statements.

DELETE Syntax

The DELETE statement requires the table name in the FROM clause and the row's primary key in the WHERE clause, as shown in the following example:

```
<delete_statement> ::= DELETE FROM <table_name> WHERE { Id =
<expression> } [ { AND | OR } ... ]
<expression> ::=
    | @ <parameter>
    | ?
    | <literal>
```

You can use the executeUpdate method of the Statement or PreparedStatement classes to execute data manipulation commands and retrieve the number of affected rows, as shown in the following example:

```
Connection connection = DriverManager.getConnection(
    ("jdbc:marketo:Schema=REST;RESTEndpoint=https://MyMarketoUrl/rest;OAuthC
lientId=MyOAuthClientId;OAuthClientSecret=MyOAuthClientSecret;"));
String cmd = "DELETE FROM Leads WHERE Id = ?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "c2ef66a5-a545-413b-9312-79a53caadbc4");
int count=pstmt.executeUpdate();
connection.close();
```

EXECUTE Statements

To execute stored procedures, you can use EXECUTE or EXEC statements.

EXEC and EXECUTE assign stored procedure inputs, referenced by name, to values or parameter names.

Stored Procedure Syntax

To execute a stored procedure as an SQL statement, use the following syntax:

```
{ EXECUTE | EXEC } <stored_proc_name>
{
  [ @ ] <input_name> = <expression>
} [ , ... ]
<expression> ::=
  | @ <parameter>
  | ?
  | <literal>
```

Example Statements

Reference stored procedure inputs by name:

```
EXECUTE my_proc @second = 2, @first = 1, @third = 3;
```

Execute a parameterized stored procedure statement:

```
EXECUTE my_proc second = @p1, first = @p2, third = @p3;
```

PIVOT and UNPIVOT

PIVOT and **UNPIVOT** can be used to change a table-valued expression into another table.

PIVOT

PIVOT rotates a table-value expression by turning unique values from one column into multiple columns in the output. PIVOT can run aggregations where required on any column value.

PIVOT Syntax

```
"SELECT 'AverageCost' AS Cost_Sorted_By_Production_Days, [0], [1], [2],
[3], [4]
FROM
(
SELECT DaysToManufacture, StandardCost
FROM Production.Product
) AS SourceTable
PIVOT
(
AVG(StandardCost)
FOR DaysToManufacture IN ([0], [1], [2], [3], [4])
) AS PivotTable;"
```

UNPIVOT

UNPIVOT carries out nearly the opposite to PIVOT by rotating columns of a table-valued expressions into column values.

UNPIVOT Syntax

```
"SELECT VendorID, Employee, Orders
FROM
(SELECT VendorID, Emp1, Emp2, Emp3, Emp4, Emp5
FROM pvt) p
UNPIVOT
(Orders FOR Employee IN
(Emp1, Emp2, Emp3, Emp4, Emp5)
)AS unpvt;"
```

For further information on PIVOT and UNPIVOT, see [FROM clause plus JOIN, APPLY, PIVOT \(Transact-SQL\)](#)

SOAP Data Model

The Marketo Adapter models Marketo entities in relational Tables, Views, and Stored Procedures. API limitations and requirements are documented in the following sections; you can use the SupportEnhancedSQL feature, set by default, to circumvent most of these limitations.

Note: Bulk operations are not supported for the SOAP data model.

Tables

[Tables](#) describes the available tables.

Views

[Views](#) are tables that cannot be modified. Typically, read-only data are shown as views.

Tables

The adapter models the data in Marketo into a list of tables that can be queried using standard SQL statements.

Generally, querying Marketo tables is the same as querying a table in a relational database. Sometimes there are special cases, for example, including a certain column in the WHERE clause might be required to get data for certain columns in the table. This is typically needed for situations where a separate request must be made for each row to get certain columns. These types of situations are clearly documented at the top of the table page linked below.

Marketo Adapter Tables

Name	Description
Leads	Create, update, delete, and query Leads for a Marketo organization.
Programs	Query and update Programs for a Marketo organization.

Leads

Create, update, delete, and query Leads for a Marketo organization.

Table Specific Information

Select

All columns must be specified using the '=' operator. The Id and Email columns allow multiple values to be specified by using the OR logical operator. The OldestUpdatedAt column acts as a "since" criteria, and a range can be specified by adding the LatestUpdatedAt column, which specifies the "until" criteria.

Response time from the server can be improved by identifying only the rows and columns you want to retrieve.

```
SELECT Id, FirstName, LastName FROM Leads WHERE Email='test@lead1.com'  
OR Email='test@lead4.com'
```

Insert

To create a new Lead record, specify the information about the Lead to be entered into the database.

The following example demonstrates how to insert a new Lead:

```
INSERT INTO Leads (Email, FirstName, LastName) VALUES ('john@mark.com',  
'John', 'Mark')
```

Update

Any field that is not read-only can be updated.

```
UPDATE Leads SET MobilePhone='111-222-3333' WHERE Id='1'
```

Delete

Deleting a Lead is not directly supported by the Marketo API. However, a Lead can be deleted by creating a campaign, to be run via an API call, that triggers the deletion of a Lead as part of the campaign's flow. Details for how to create a campaign like this can be found here: <http://developers.marketo.com/blog/delete-a-lead-with-the-marketo-api/>

Once the campaign is created within Marketo, a delete request can be made by specifying the campaign with either the CampaignId or CampaignName column, along with the Id or Email column of the Leads to be deleted.

```
DELETE FROM Leads WHERE Id='1000195' AND CampaignId='1027'
```

Columns

Name	Type	ReadOnly	Description
Id [KEY]	<i>Integer</i>	True	The unique, Marketo-assigned identifier of the account.
Email	<i>String</i>	False	The lead's email address.
Salutation	<i>String</i>	False	The lead's salutation.
FirstName	<i>String</i>	False	The lead's first name.
MiddleName	<i>String</i>	False	The lead's middle name.
LastName	<i>String</i>	False	The lead's last name.
DateofBirth	<i>Date</i>	False	The lead's date of birth.
Title	<i>String</i>	False	The lead's job title.
Address	<i>String</i>	False	The lead's street address.

City	<i>String</i>	False	The lead's city.
State	<i>String</i>	False	The lead's state.
PostalCode	<i>String</i>	False	The lead's ZIP/postal code.
Country	<i>String</i>	False	The lead's country.
Website	<i>String</i>	False	The lead's website.
Phone	<i>String</i>	False	The lead's phone number.
MobilePhone	<i>String</i>	False	The lead's mobile phone number.
Fax	<i>String</i>	False	The lead's fax number.
Company	<i>String</i>	False	The name of the lead's company.
MainPhone	<i>String</i>	False	The phone number of the lead's company.
SICCode	<i>String</i>	False	The SIC (Standard Industrial Classification) code of the lead's company.
Site	<i>String</i>	False	The site of the lead's company.

BillingStreet	<i>String</i>	False	The billing street address of the lead's company.
BillingCity	<i>String</i>	False	The billing city of the lead's company.
BillingState	<i>String</i>	False	The billing state of the lead's company.
BillingPostalCode	<i>String</i>	False	The billing ZIP/postal code of the lead's company.
BillingCountry	<i>String</i>	False	The billing country of the lead's company.
NumberOfEmployees	<i>Integer</i>	False	The number of employees at the lead's company.
Industry	<i>String</i>	False	The industry of the lead's company.
AnnualRevenue	<i>Decimal</i>	False	The annual revenue generated at the lead's company.
LeadScore	<i>Integer</i>	False	The lead's score.
Rating	<i>String</i>	False	The lead's rating.
Unsubscribed	<i>Boolean</i>	False	Determines whether the lead is unsubscribed.
UnsubscribedReason	<i>String</i>	False	The reason why the

			lead has unsubscribed.
AnonymousIP	<i>String</i>	False	The IP address of the lead if it is anonymous.
Department	<i>String</i>	False	The lead's department.
DoNotCall	<i>Boolean</i>	False	Identifies whether the lead is on the 'Do Not Call' list.
DoNotCallReason	<i>String</i>	False	The reason why the lead is on the 'Do Not Call' list.
InferredCompany	<i>String</i>	False	The lead's inferred company.
InferredCountry	<i>String</i>	False	The lead's inferred country.
LeadRole	<i>String</i>	False	The lead's role.
LeadSource	<i>String</i>	False	The lead's source.
LeadStatus	<i>String</i>	False	The lead's current status.
ForeignSysPersonId	<i>String</i>	False	The lead's Id from a foreign system.
ForeignSysType	<i>String</i>	False	The foreign system type that the ForeignSysPersonId value came from.

MarketoSocialFacebookDisplayName	<i>String</i>	False	The lead's Facebook display name.
MarketoSocialFacebookId	<i>String</i>	False	The lead's Facebook Id.
MarketoSocialFacebookPhotoURL	<i>String</i>	False	The lead's Facebook photo URL.
MarketoSocialFacebookProfileURL	<i>String</i>	False	The lead's Facebook profile URL.
MarketoSocialFacebookReach	<i>Integer</i>	False	The lead's Facebook reach.
MarketoSocialFacebookReferredEnrollments	<i>Integer</i>	False	The lead's Facebook referred enrollments.
MarketoSocialFacebookReferredVisits	<i>Integer</i>	False	The lead's Facebook referred visits.
MarketoSocialGender	<i>String</i>	False	The lead's social gender.
MarketoSocialLastReferredEnrollment	<i>Datetime</i>	False	The lead's last social referred enrollment.
MarketoSocialLastReferredVisit	<i>Datetime</i>	False	The lead's last social referred visit.
MarketoSocialLinkedInDisplayName	<i>String</i>	False	The lead's LinkedIn display name.
MarketoSocialLinkedInId	<i>String</i>	False	The lead's LinkedIn Id.
MarketoSocialLinkedInPhotoURL	<i>String</i>	False	The lead's LinkedIn photo URL.

MarketoSocialLinkedInProfileURL	<i>String</i>	False	The lead's LinkedIn profile URL.
MarketoSocialLinkedInReach	<i>Integer</i>	False	The lead's LinkedIn reach.
MarketoSocialLinkedInReferredEnrollments	<i>Integer</i>	False	The lead's LinkedIn referred enrollments.
MarketoSocialLinkedInReferredVisits	<i>Integer</i>	False	The lead's LinkedIn referred visits.
MarketoSocialSyndicationId	<i>String</i>	False	The lead's social syndication Id.
MarketoSocialTotalReferredEnrollments	<i>Integer</i>	True	The lead's total social referred enrollments.
MarketoSocialTotalReferredVisits	<i>Integer</i>	True	The lead's total social referred visits.
MarketoSocialTwitterDisplayName	<i>String</i>	False	The lead's Twitter display name.
MarketoSocialTwitterId	<i>String</i>	False	The lead's Twitter Id.
MarketoSocialTwitterPhotoURL	<i>String</i>	False	The lead's Twitter photo URL.
MarketoSocialTwitterProfileURL	<i>String</i>	False	The lead's Twitter profile URL.
MarketoSocialTwitterReach	<i>Integer</i>	False	The lead's Twitter reach.
MarketoSocialTwitterReferredEnrollments	<i>Integer</i>	False	The lead's Twitter referred

			enrollments.
MarketoSocialTwitterReferredVisits	<i>Integer</i>	False	The lead's Twitter referred visits.
MarketoSocialYahooDisplayName	<i>String</i>	False	The lead's Yahoo display name.
MarketoSocialYahooPhotoURL	<i>String</i>	False	The lead's Yahoo photo URL.
MarketoSocialYahooProfileURL	<i>String</i>	False	The lead's Yahoo profile URL.
MarketoSocialYahooReach	<i>Integer</i>	False	The lead's Yahoo reach.
MarketoSocialYahooReferredEnrollments	<i>Integer</i>	False	The lead's Yahoo referred enrollments.
MarketoSocialYahooReferredVisits	<i>Integer</i>	False	The lead's Yahoo referred visits.
MicrosoftAddress2	<i>String</i>	False	The lead's Microsoft street line 2 address.
MicrosoftAddress3	<i>String</i>	False	The lead's Microsoft street line 3 address.
MicrosoftBillingAddress2	<i>String</i>	False	The lead's Microsoft billing street line 2 address.
MicrosoftBillingAddress3	<i>String</i>	False	The lead's Microsoft billing street line 3 address.

MicrosoftDoNotEmail	<i>Boolean</i>	False	Identifies whether the lead is on the Microsoft 'Do Not Email' list.
MicrosoftDoNotFax	<i>Boolean</i>	False	Identifies whether the lead is on the Microsoft 'Do Not Fax' list.
MicrosoftDoNotSendMarketingMaterial	<i>Boolean</i>	False	Identifies whether the lead is on the Microsoft 'Do Not Send Marketing Material' list.
MicrosoftHomePhone	<i>String</i>	False	The lead's Microsoft home phone.
MicrosoftPreferredMethodofContact	<i>String</i>	False	The lead's Microsoft preferred method of contact.
MicrosoftTopic	<i>String</i>	False	The lead's Microsoft topic.
SAP_CRM_ABCClassification	<i>String</i>	False	The lead's SAP ABC classification.
SAP_CRM_AccountID	<i>String</i>	False	The lead's SAP account Id.
SAP_CRM_Account_CreatedOn	<i>Datetime</i>	False	The lead's SAP account creation date.
SAP_CRM_Account_ExternalID	<i>String</i>	False	The lead's SAP account external Id.
SAP_CRM_Account_Fax	<i>String</i>	False	The lead's SAP

			account fax number.
SAP_CRM_Account_HouseNumber	<i>String</i>	False	The lead's SAP account house number.
SAP_CRM_Account_IsDeleted	<i>Boolean</i>	False	Identifies if the lead's SAP account is deleted.
SAP_CRM_Account_Status	<i>String</i>	False	The lead's SAP account status.
SAP_CRM_BestReachedBy	<i>String</i>	False	The lead's SAP best method to be reached.
SAP_CRM_ContactID	<i>String</i>	False	The lead's SAP contact Id.
SAP_CRM_Contact_Name	<i>String</i>	False	The lead's SAP contact name.
SAP_CRM_DepartmentFromBusinessCard	<i>String</i>	False	The lead's SAP department retrieved from business card.
SAP_CRM_EmployeeID	<i>String</i>	False	The lead's SAP employee Id.
SAP_CRM_ExternalSystem	<i>String</i>	False	The lead's SAP external system.
SAP_CRM_Function	<i>String</i>	False	The lead's SAP function.
SAP_CRM_Lead_HouseNumber	<i>String</i>	False	The lead's SAP lead house number.

SAP_CRM_Name	<i>String</i>	False	The lead's SAP name.
SAP_CRM_Person_Type	<i>String</i>	False	The lead's SAP person type.
SAP_CRM_PrimaryContact	<i>String</i>	False	The lead's SAP primary contact.
SAP_CRM_Qualification	<i>String</i>	False	The lead's SAP qualification.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestUpdatedAt	<i>Datetime</i>	Used when performing a GET. Returns all leads updated since the specified time.
LatestUpdatedAt	<i>Datetime</i>	Used when performing a GET. Returns all leads updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.
CampaignSource	<i>String</i>	Used when performing a DELETE and is used to specify the source of the campaign. Valid values are MKTOWS or SALES. If not specified, MKTOWS will be used by default.
CampaignId	<i>Integer</i>	Used when performing a DELETE and is used to specify the Id of the campaign that contains the 'Delete Lead'

		trigger. Note CampaignId and Campaign name cannot both be specified.
CampaignName	String	Used when performing a DELETE and is used to specify the name of the campaign that contains the 'Delete Lead' trigger. Note CampaignId and Campaign name cannot both be specified.

Programs

Query and update Programs for a Marketo organization.

Table Specific Information

Select

Marketo allows the following columns to be used in the WHERE clause of a SELECT query: Id, Name, CreatedAt, UpdatedAt, CRMId, TagType, TagValue, WorkspaceName, WorkspaceId, and IncludeArchive. The Id, Name, and CRMId columns allow multiple values to be specified by using the OR logical operator. The CreatedAt and UpdateAt filters can be specified twice to create a date range.

```
SELECT * FROM Programs WHERE CreatedAt > '08/01/2014' AND CreatedAt <= '08/31/2014'
```

Update

Any field that is not read-only can be updated.

Note that the Tag*** and Cost*** columns only take a single value. Therefore, they do not take a comma-separated list when performing an update.

```
UPDATE Programs SET TagType='Program Owner', TagValue='Admin', CostMonth='11/2014', CostAmount='30' WHERE Id='1002'
```

Columns

Name	Type	ReadOnly	Description
Id [KEY]	<i>Integer</i>	True	The unique, Marketo-assigned identifier of the program.
Name	<i>String</i>	True	The name of the program.
Description	<i>String</i>	True	The description of the program.
WorkspaceId	<i>String</i>	True	The Id of the workspace where the program is located.
WorkspaceName	<i>String</i>	True	The name of the workspace where the program is located.
TreePath	<i>String</i>	True	The folder structure tree path describing the location of the program.
IsArchived	<i>String</i>	True	Specifies whether the program is archived.
TagType#	<i>String</i>	False	A comma-separated list of tag types associated with the program. Each TagType has a value associated with it which is returned via the TagValue column.
TagValue#	<i>String</i>	False	A comma-separated list of tag values. Each value corresponds to the type listed within the TagTypes column.
CostMonth#	<i>String</i>	False	A comma-separated list of period cost months for the program.
CostAmount#	<i>String</i>	False	A comma-separated list of period cost amounts for the program.
CostId#	<i>String</i>	False	A comma-separated list of period cost Ids

			for the program.
CostNote#	<i>String</i>	False	A comma-separated list of period cost notes for the program.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
CreatedAt	<i>Datetime</i>	Used when performing a GET. Filters programs by the date they were created.
UpdatedAt	<i>Datetime</i>	Used when performing a GET. Filters programs by the date they were updated.
CRMid	<i>String</i>	Used when performing a GET. The CRM Id associated with the program. (This value could refer to the Id of the Salesforce campaign connected to the program.)
IncludeArchive	<i>Boolean</i>	Used when performing a GET. When set to 'True', will return archived programs.

Views

Views are composed of columns and pseudo columns. Views are similar to tables in the way that data is represented; however, views do not support updates. Entities that are represented as views are typically read-only entities. Often, a stored procedure is available to update the data if such functionality is applicable to the data source.

Queries can be executed against a view as if it were a normal table, and the data that comes back is similar in that regard.

Dynamic views, such as queries exposed as views, and views for looking up specific combinations of project_team work items are supported.

Marketo Adapter Views

Name	Description
Activities	Query Activities for a Marketo organization.
Activities_AddToList	Query Add To List Activities for a Marketo organization.
Activities_AddToSFDCCampaign	Query Add To SFDC Campaign Activities for a Marketo organization.
Activities_ChangeDataValue	Query Change Data Value Activities for a Marketo organization.
Activities_ChangeOwner	Query Change Owner Activities for a Marketo organization.
Activities_ChangeRevenueStage	Query Change Revenue Stage Activities for a Marketo organization.
Activities_ChangeScore	Query Change Score Activities for a Marketo organization.
Activities_ChangeStatusInProgression	Query Change Status In Progression Activities for a Marketo organization.
Activities_ChangeStatusInSFDCCampaign	Query Change Status In SFDC Campaign Activities for a Marketo organization.
Activities_ClickEmail	Query Click Email Activities for a Marketo organization.
Activities_ClickLink	Query Click Link Activities for a Marketo organization.

Activities_ConvertLead	Query Convert Lead Activities for a Marketo organization.
Activities_CreateTask	Query Create Task Activities for a Marketo organization.
Activities_EmailBouncedSoft	Query Email Bounced Soft Activities for a Marketo organization.
Activities_EmailDelivered	Query Email Delivered Activities for a Marketo organization.
Activities_InterestingMoments	Query Interesting Moments Activities for a Marketo organization.
Activities_MergeLeads	Query Merge Leads Activities for a Marketo organization.
Activities_NewLeads	Query New Lead Activities for a Marketo organization.
Activities_OpenEmail	Query Open Email Activities for a Marketo organization.
Activities_RemoveFromFlow	Query Remove from Flow Activities for a Marketo organization.
Activities_RemoveFromList	Query Remove from List Activities for a Marketo organization.
Activities_RemoveFromSFDCCampaign	Query Remove from SFDC Campaign Activities for a Marketo organization.
Activities_SendAlert	Query Send Alert Activities for a Marketo organization.
Activities_SendEmail	Query Send Email Activities for a Marketo organization.
Activities_SFDCActivity	Query SFDC Activity Activities for a Marketo organization.

	organization.
Activities_SFDCMergeLeads	Query SFDC Merge Leads Activities for a Marketo organization.
Activities_VisitWebpage	Query Visit Webpage Activities for a Marketo organization.
Campaigns	Query Campaigns for a Marketo organization.
Channels	Query Channels for a Marketo organization.
Opportunities	Query Opportunities for a Marketo organization.
OpportunityPersonRoles	Query Opportunity Person Roles for a Marketo organization.
Tags	Query Tags for a Marketo organization.

Activities

Query Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, ActivityCreatedAt. ActivityNameFilter or ExcludeActivities can be specified as an additional column in the WHERE clause. A list of available Activity Types can be found here: <http://developers.marketo.com/activity-types/>. The LeadId, Email, ActivityNameFilter, and ExcludeActivities columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator. Otherwise, an exception will be thrown.

```
SELECT * FROM Activities WHERE LeadId='1' OR LeadId='2'
```

Columns

Name	Type	Description
ActivityId [KEY]	Integer	The unique Id of the activity.
LeadId	Integer	The unique Id of the lead associated with the activity.
ActivityDateTime	Datetime	The date and time the activity was performed.
ActivityType	String	The type of activity.
MktgAssetName	String	The name of the marketing asset tied to the activity.
Campaign	String	The campaign the activity is associated with.
ActivityAttributes#	String	A comma-separated list of name-value pairs (name=value).
Email	String	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.
ActivityNameFilter	String	Used when performing a GET. The name of the specific activities to retrieve.
ExcludeActivities	String	Used when performing a GET. The name of the specific activities to exclude.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_AddToList

Query Add To List Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_AddToList WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
ListId	<i>Integer</i>	The Id of the list that the lead was added to.
ListName	<i>String</i>	The name of the list that the lead was added to.
Source	<i>String</i>	The source by which the lead was added to the list.
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_AddToSFDCCampaign

Query Add To SFDC Campaign Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_AddToSFDCCampaign WHERE Email='test@server.com'
OR Email='testlead@server.com'
```

Columns

Name	Type	Description
------	------	-------------

ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
CampaignId	<i>Integer</i>	The Id of the SFDC campaign that the lead was added to.
Campaign	<i>String</i>	The campaign the activity is associated with.
Status	<i>String</i>	The status of the activity.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.

StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_ChangeDataValue

Query Change Data Value Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_ChangeDataValue WHERE Email='test@server.com'
OR Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.

Source	<i>String</i>	The source by which the data value was changed (i.e., Web service API, SFDC, etc).
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.
AttributeName	<i>String</i>	The name of the attribute whose data value was changed.
NewValue	<i>String</i>	The new and current data value.
OldValue	<i>String</i>	The old and previous data value.
Reason	<i>String</i>	The reason why the data value was changed.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description

OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_ChangeOwner

Query Change Owner Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_ChangeOwner WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId	<i>Integer</i>	The unique Id of the activity.

[KEY]		
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
Owner	<i>String</i>	The name of the new and current owner of the lead.
OwnerId	<i>String</i>	The Id of the new and current owner of the lead.
OldOwner	<i>String</i>	The name of the previous owner of the lead.
OldOwnerId	<i>String</i>	The Id of the previous owner of the lead.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.

ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_ChangeRevenueStage

Query Change Revenue Stage Activities for a Marketo organization.

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
ModelId	<i>String</i>	The id of the revenue model.
ModelVersion	<i>String</i>	The version of the revenue model.
NewStageId	<i>String</i>	The new Id of the stage that the lead is currently in.
NewStage	<i>String</i>	The new name of the stage that the lead is currently in.

OldStageId	<i>String</i>	The Id of the stage that the lead was previously in.
OldStage	<i>String</i>	The name of the stage that the lead was previously in.
Reason	<i>String</i>	The reason why the data value was changed.
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_ChangeScore

Query Change Score Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_ChangeScore WHERE Email='test@server.com' OR  
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	Integer	The unique Id of the activity.
LeadId	Integer	The unique Id of the lead associated with the activity.
ScoreName	String	The name of the score whose data value was changed.
ChangeValue	String	The net change in the score value. For example, +5 or -5.
NewValue	String	The new and current score value.

OldValue	<i>String</i>	The old and previous score value.
Urgency	<i>String</i>	How much the score has changed recently. A higher urgency represents a score that has increased a lot lately, showing customer interest. This value is derived from score value (NewValue).
Reason	<i>String</i>	The reason why the data value was changed.
Priority	<i>String</i>	The priority rank used to decide which lead should be contacted first. Priority has two components: Urgency and Relative Score. The higher the priority, the higher likelihood that the lead will respond positively to a contact.
RelativeScore	<i>String</i>	Contains a measure of how a lead's score compares to other lead scores. This value is derived from score value (NewValue).
RelativeUrgency	<i>String</i>	Contains a measure of how a lead's urgency compares to other lead urgencies.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_ChangeStatusInProgression

Query Change Status In Progression Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_ChangeStatusInProgression WHERE
Email='test@server.com' OR Email='testlead@server.com'
```

Columns

Name	Type	Description
------	------	-------------

ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
NewStatus	<i>String</i>	The new and current status.
NewStatusId	<i>String</i>	The Id of the NewStatus value.
OldStatus	<i>String</i>	The old and previous status.
OldStatusId	<i>String</i>	The Id of the OldStatus value.
Program	<i>String</i>	The name of the program where the status change occurred.
ProgramId	<i>Integer</i>	The Id of the Program.
Success	<i>Boolean</i>	Identifies whether the status represents a success.
AcquiredBy	<i>String</i>	Identifies whether the lead was acquired by this activity.
Reason	<i>String</i>	The reason why the status was changed.
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.

ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_ChangeStatusInSFDCCampaign

Query Change Status In SFDC Campaign Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_ChangeStatusInSFDCCampaign WHERE
Email='test@server.com' OR Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
CampaignId	<i>Integer</i>	The Id of the SFDC campaign that the lead's status was changed.
Campaign	<i>String</i>	The campaign the activity is associated with.
NewStatus	<i>String</i>	The new and current status.
OldStatus	<i>String</i>	The old and previous status.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.

AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_ClickEmail

Query Click Email Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_ClickEmail WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
Campaign	<i>String</i>	The campaign the activity is associated with.
Link	<i>String</i>	The URL of the link that was clicked.
MailingId	<i>String</i>	The Id of the email that the link was contained in.
VariationId	<i>String</i>	The Id of the email variation.
UserAgent	<i>String</i>	The Web browser user agent information obtained when the lead clicked the email link.

IsMobileDevice	<i>String</i>	Identifies whether the device used to click the link was a mobile device.
Platform	<i>String</i>	The operating system platform used when the link was opened.
Device	<i>String</i>	The type of device used when the link was opened.
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.

StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_ClickLink

Query Click Link Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_ClickLink WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.

Campaign	<i>String</i>	The campaign the activity is associated with.
WebpageId	<i>String</i>	The Id of the Web page that contained the link that was clicked.
LinkId	<i>String</i>	The Id of the link that was clicked.
QueryParameters	<i>String</i>	The query parameters contained within the link.
ClientIPAddress	<i>String</i>	The IP address of the client that clicked the link.
MessageId	<i>String</i>	The Id of the message where the link was clicked.
UserAgent	<i>String</i>	The Web browser user agent information obtained when the lead clicked the link.
ReferrerURL	<i>String</i>	The URL of the referrer used to identify where the link click originated from.
CreatedAt	<i>String</i>	The date and time the link click activity was created.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_ConvertLead

Query Convert Lead Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_ConvertLead WHERE Email='test@server.com' OR  
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
Campaign	<i>String</i>	The campaign the activity is associated with.
AssignTo	<i>String</i>	The owner that the lead was assigned to.
SFDCAccountId	<i>String</i>	The Id of the lead's SFDC account.
SFDCType	<i>String</i>	The lead's SFDC type.
SFDCOpportunityId	<i>String</i>	The Id of the lead's SFDC opportunity.
SFDCLeadId	<i>String</i>	The lead's SFDC lead Id.
SFDCContactId	<i>String</i>	The lead's SFDC contact Id.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_CreateTask

Query Create Task Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_CreateTask WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	Integer	The unique Id of the activity.
LeadId	Integer	The unique Id of the lead associated with the activity.
Campaign	String	The campaign the activity is associated with.
Subject	String	The subject of the task.
Operator	String	The task operator.
DataType	String	The task data type.
DueIn	String	The number of days until the task is due.
Owner	String	The owner that the task was assigned to.
Comments	String	The comments for the task.
Priority	String	The priority of the task.
Status	String	The current status of the task.
Notify	String	Identifies whether a notification should be sent.

RemindIn	<i>String</i>	The number of days that a reminder should be sent.
SFDCTaskId	<i>String</i>	The SFDC Id of the task.
OwnerPersonId	<i>String</i>	The Id of the Owner of the task.
DueDate	<i>String</i>	The date the task is due.
ReminderTimestamp	<i>String</i>	The timestamp identifying when the reminder will be sent.
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
-------------	-------------	--------------------

OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_EmailBouncedSoft

Query Email Bounced Soft Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_EmailBouncedSoft WHERE Email='test@server.com'
OR Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId	<i>Integer</i>	The unique Id of the activity.

[KEY]		
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
MailingId	<i>String</i>	The Id of the email message that was sent.
Campaign	<i>String</i>	The campaign the activity is associated with.
Email	<i>String</i>	The email address of the intended recipient.
Details	<i>String</i>	The details about why the email bounced.
VariantId	<i>String</i>	The Id of the email variant.
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_EmailDelivered

Query Email Delivered Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_EmailDelivered WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
------	------	-------------

ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
Campaign	<i>String</i>	The campaign the activity is associated with.
MailingId	<i>String</i>	The Id of the email message that was sent.
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.

ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_InterestingMoments

Query Interesting Moments Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_InterestingMoments WHERE
Email='test@server.com' OR Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.

LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
Campaign	<i>String</i>	The campaign the activity is associated with.
Type	<i>String</i>	The type of moment that occurred such as Web, Email, Milestone, etc.
Description	<i>String</i>	The description about the moment.
Operator	<i>String</i>	The interesting moment operator.
DataType	<i>String</i>	The interesting moment datatype.
Source	<i>String</i>	The source by which the interesting moment occurred.
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.
Date	<i>String</i>	The date the interesting moment occurred.
TriggeringActivityLogID	<i>String</i>	The Id of the activity that triggered the interesting moment.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities

using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_MergeLeads

Query Merge Leads Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by

the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_MergeLeads WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
LeadName	<i>String</i>	The name of the lead that was merged.
MasterUpdated	<i>Boolean</i>	Identifies whether the master lead was updated.
Mergelds	<i>String</i>	The lead Ids that were merged.
MergeFields	<i>String</i>	The fields that were merged.
MergedInSales	<i>String</i>	Identifies whether the lead was merged in sales.
MergeSource	<i>String</i>	The source by which the merge was performed.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.

Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.
-------	---------------	--

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_NewLeads

Query New Lead Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by

the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_NewLeads WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
MktgAssetName	<i>String</i>	The name of the lead.
SourceType	<i>String</i>	The source type that created the new lead, such as Salesforce.com.
LeadSource	<i>String</i>	The source where the new lead was created.
SourceInfo	<i>String</i>	Information about the source of the new lead creation.
SFDCType	<i>String</i>	The SFDC type for this lead.
CreatedDate	<i>String</i>	The date the new lead was created.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.

Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.
-------	---------------	--

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_OpenEmail

Query Open Email Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by

the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_OpenEmail WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
Campaign	<i>String</i>	The campaign the email open activity is associated with.
MailingId	<i>String</i>	The Id of the email that was opened.
VariationId	<i>String</i>	The Id of the email variation.
UserAgent	<i>String</i>	The Web browser user agent information obtained when the lead clicked the email link.
IsMobileDevice	<i>String</i>	Identifies whether the device used to click the link was a mobile device.
Platform	<i>String</i>	The operating system platform used when the link was opened.
Device	<i>String</i>	The type of device used when the link was opened.

StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_RemoveFromFlow

Query Remove from Flow Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_RemoveFromFlow WHERE Email='test@server.com' OR  
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
Campaign	<i>String</i>	The campaign the activity is associated with.
RemoveFromCampaigns	<i>String</i>	The campaign Id that the lead was removed from.
DataType	<i>String</i>	The campaign flow datatype.
StepId	<i>String</i>	The Id of the current step in the flow.

ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_RemoveFromList

Query Remove from List Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_RemoveFromList WHERE Email='test@server.com' OR  
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	Integer	The unique Id of the activity.
LeadId	Integer	The unique Id of the lead associated with the activity.
Campaign	String	The campaign the activity is associated with.
ListId	Integer	The Id of the list the lead was removed from.
StepId	String	The Id of the current step in the flow.
ChoiceNumber	String	The choice number of the current step that triggered the

		activity.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_RemoveFromSFDCCampaign

Query Remove from SFDC Campaign Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_RemoveFromSFDCCampaign WHERE
Email='test@server.com' OR Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
CampaignId	<i>Integer</i>	The Id of the campaign that the lead was removed from.
Campaign	<i>String</i>	The campaign the activity is associated with.
Status	<i>String</i>	The current status of the SFDC campaign.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.

Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.
-------	---------------	--

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_SendAlert

Query Send Alert Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by

the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_SendAlert WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
CampaignRunId	<i>String</i>	The Id of the campaign that the alert was run for.
Campaign	<i>String</i>	The campaign the email send activity is associated with.
MailingId	<i>String</i>	The Id of the email alert sent.
SendToOwner	<i>String</i>	Identifies which owner received the alert. For example Lead, Account, or None.
SendToList	<i>String</i>	Additional recipients of the alert.
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.

ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_SendEmail

Query Send Email Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_SendEmail WHERE Email='test@server.com' OR  
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
CampaignRunId	<i>String</i>	The Id of the campaign the email was sent for.
Campaign	<i>String</i>	The campaign the activity is associated with.
MailingId	<i>String</i>	The Id of the email sent.
StepId	<i>String</i>	The Id of the current step in the flow.
ChoiceNumber	<i>String</i>	The choice number of the current step that triggered the activity.

ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_SFDCActivity

Query SFDC Activity Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_SFDCActivity WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
Subject	<i>String</i>	The subject of the SFDC activity.
Description	<i>String</i>	The description of the SFDC activity.
OwnerId	<i>String</i>	The Id of the SFDC activity owner.
ActivityOwner	<i>String</i>	The owner of the SFDC activity.
Status	<i>String</i>	The current status of the SFDC activity.

Priority	<i>String</i>	The priority of the activity.
IsTask	<i>String</i>	Identifies whether the activity is a task.
DueDate	<i>String</i>	The date the SFDC activity is due.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_SFDCMergeLeads

Query SFDC Merge Leads Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_SFDCMergeLeads WHERE Email='test@server.com' OR  
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	Integer	The unique Id of the activity.
LeadId	Integer	The unique Id of the lead associated with the activity.
LeadName	String	The name of the SFDC lead that was merged.
Merged	String	The fields that were merged for the SFDC lead.
WinningValues	String	The winning values of the merged SFDC lead.

ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.
StaticListId	<i>Integer</i> <i>other:filterable=</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Activities_VisitWebpage

Query Visit Webpage Activities for a Marketo organization.

Table Specific Information

Select

Marketo allows one (and only one) of the following columns to be used in the WHERE clause of a SELECT query: LeadId, Email, StaticListId, StaticListName, OldestCreatedAt, or ActivityCreatedAt. The LeadId and Email columns allow multiple values to be specified by the OR logical operator. All columns must be specified using the '=' operator; otherwise, an exception will be thrown.

```
SELECT * FROM Activities_VisitWebpage WHERE Email='test@server.com' OR
Email='testlead@server.com'
```

Columns

Name	Type	Description
ActivityId [KEY]	<i>Integer</i>	The unique Id of the activity.
LeadId	<i>Integer</i>	The unique Id of the lead associated with the activity.
Campaign	<i>String</i>	The campaign the activity is associated with.
WebpageId	<i>String</i>	The Id of the Web page that was visited.
WebpageURL	<i>String</i>	The URL of the Web page that was visited.
LinkId	<i>String</i>	The Id of the link that was clicked.
QueryParameters	<i>String</i>	The query parameters contained within the URL.

ClientIPAddress	<i>String</i>	The IP address of the client that clicked the link.
MessageId	<i>String</i>	The Id of the message where the link was clicked.
UserAgent	<i>String</i>	The Web browser user agent information obtained when the Web page was visited.
ReferrerURL	<i>String</i>	The URL of the referrer used to identify where the link click originated from.
CreatedAt	<i>String</i>	The date and time the Web page activity was created.
ActivityDateTime	<i>Datetime</i>	The date and time the activity was performed.
AssetName	<i>String</i>	The marketing asset name associated with the activity.
Email	<i>String</i>	Used when performing a GET. Filters activities using the email address of the lead associated with the activity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
OldestCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated since the specified time.
ActivityCreatedAt	<i>Datetime</i>	Used when performing a GET. Returns all activities updated up until the specified time.

StaticListId	<i>Integer</i>	Used when performing a GET. The Id of the static list to retrieve.
StaticListName	<i>String</i>	Used when performing a GET. The name of the static list to retrieve.

Campaigns

Query Campaigns for a Marketo organization.

Table Specific Information

Select

Marketo allows the Source and Name columns to be used in the WHERE clause of a SELECT query. Use the Source column to identify the campaigns that were created within Marketo or by Sales. The Name column allows the use of the '=' or 'LIKE' operator. The '=' operator denotes an exact match; the 'LIKE' operator denotes a partial match of the name.

```
SELECT * FROM Campaigns WHERE Name LIKE 'test'
```

Columns

Name	Type	Description
Id [KEY]	<i>Integer</i>	The unique Id of the campaign.
Name	<i>String</i>	The name of the campaign.
Description	<i>String</i>	The description of the campaign.

Source	<i>String</i>	The source of the campaign. Valid values are MKTOWS and SALES.
--------	---------------	--

Channels

Query Channels for a Marketo organization.

Table Specific Information

Select

Marketo allows the TagValue column to be used in the WHERE clause of a SELECT query. The TagValue column allows multiple values to be specified by using the OR logical operator. Note that only the '=' operator is supported.

```
SELECT * FROM Channels WHERE TagValue='Email Blast' OR TagValue='Blog'
OR TagValue='Webinar'
```

Columns

Name	Type	Description
TagValue [KEY]	<i>String</i>	The tag value of the channel. For example, Webinar, Blog, or Tradeshow.
Status [KEY]	<i>String</i>	The progression status of the channel.
Step	<i>Integer</i>	The step number of each progression status used to sequence the order of the channel.
Success	<i>Boolean</i>	Specifies whether the progression status results in a success.

CheckInStatus	<i>String</i>	The check-in status for a roadshow event. Valid values are None, Registered, Attended.
WebinarBehavior	<i>String</i>	The behavior performed during a webinar event.

Opportunities

Query Opportunities for a Marketo organization.

Table Specific Information

Select

Marketo allows the following columns to be used in the WHERE clause of a SELECT query: Id, Name, CreatedAt, UpdatedAt, Type, and Stage. The CreatedAt and UpdatedAt filters can be specified twice to create a date range.

```
SELECT * FROM Opportunities WHERE Stage='Commercial'
```

Columns

Name	Type	Description
Id [KEY]	<i>Integer</i>	The unique, Marketo-assigned identifier of the opportunity.
Name	<i>String</i>	The name of the opportunity.
Description	<i>String</i>	The description of the opportunity.

CompanyId	<i>String</i>	The Id of the company assigned to the opportunity.
Quantity	<i>Double</i>	Number of items included in this opportunity. Used in quantity-based forecasting.
Amount	<i>Decimal</i>	The estimated total sale amount from the opportunity.
ExpectedRevenue	<i>Decimal</i>	The expected revenue to result from the opportunity.
Stage	<i>String</i>	The stage that the opportunity process is currently in.
LeadSource	<i>String</i>	The source of the opportunity, such as Advertisement or Trade Show.
ExternalCreatedDate	<i>Datetime</i>	The local created date of the opportunity.
LastActivityDate	<i>Date</i>	The date of the last activity performed with this opportunity.
IsWon	<i>Boolean</i>	Determines whether the opportunity was won.
NextStep	<i>String</i>	A description of the next task in closing the opportunity.
IsClosed	<i>Boolean</i>	Determines whether the opportunity is closed.
CloseDate	<i>Date</i>	The date when the opportunity is expected to close.
Fiscal	<i>String</i>	If fiscal years are not enabled, the name of the fiscal quarter or period in which the opportunity CloseDate falls. Value should be in YYYY Q format, for example, '2006 1' for first quarter of 2006.

FiscalQuarter	<i>Integer</i>	The fiscal quarter the opportunity took place. Valid values are 1, 2, 3, or 4.
FiscalYear	<i>Integer</i>	The fiscal year the opportunity took place.
ForecastCategoryName	<i>String</i>	The name of the forecast category.
Probability	<i>Integer</i>	The percentage of estimated confidence in closing the opportunity.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
CreatedAt	<i>Datetime</i>	Used when performing a GET. Filters opportunities by the date they were created.
UpdatedAt	<i>Datetime</i>	Used when performing a GET. Filters opportunities by the date they were updated.
Type	<i>String</i>	Used when performing a GET. Filters opportunities by the opportunity type.

OpportunityPersonRoles

Query Opportunity Person Roles for a Marketo organization.

Table Specific Information

Select

Marketo allows the following columns to be used in the WHERE clause of a SELECT query: Id, CreatedAt, UpdatedAt, OpportunityId, and Role. The CreatedAt and UpdatedAt filters can be specified twice to create a date range.

```
SELECT * FROM OpportunityPersonRoles WHERE Role='Business User'
```

Columns

Name	Type	Description
Id [KEY]	<i>Integer</i>	The unique, Marketo-assigned identifier of the opportunity person role.
OpportunityId	<i>Integer</i>	The Id of the opportunity associated with the person specified via PersonId.
PersonId	<i>String</i>	The Id of the person associated with the opportunity specified via OpportunityId.
Role	<i>String</i>	The role of the contact person in regards to the opportunity.
IsPrimary	<i>Boolean</i>	Determines whether the person, specified via PersonId, is the primary contact for this opportunity.
ExternalCreatedDate	<i>Datetime</i>	The local created data of the OpportunityPersonRole association.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
CreatedAt	<i>Datetime</i>	Used when performing a GET. Filters opportunities by the date they were created.
UpdatedAt	<i>Datetime</i>	Used when performing a GET. Filters opportunities by the date they were updated.

Tags

Query Tags for a Marketo organization.

Table Specific Information

Select

Marketo allows both the Type and Value columns to be used in the WHERE clause of a SELECT query. The Value column allows multiple values to be specified by using the OR logical operator. Note that only the '=' operator is supported.

```
SELECT * FROM Tags WHERE Type='TestTag' AND Value='Value1' OR
Value='Value2'
```

Columns

Name	Type	Description
Type [KEY]	<i>String</i>	The tag type.

Value [KEY]	<i>String</i>	The tag value.
----------------	---------------	----------------

REST Data Model

The Marketo Adapter models Marketo entities in relational Tables, Views, and Stored Procedures. API limitations and requirements are documented in the following sections; you can use the [SupportEnhancedSQL](#) feature, set by default, to circumvent most of these limitations.

Note: Bulk operations are supported for all tables in the REST data model.

Tables

[Tables](#) describes the available tables. The Leads, CustomObjects, and Custom Activity tables are dynamic tables. The data model illustrates a sample of what your Marketo data model might look like. The actual data model will be obtained dynamically based on your user credentials and Marketo account.

Views

[Views](#) are tables that cannot be modified. Typically, read-only data are shown as views.

Stored Procedures

[Stored Procedures](#) are function-like interfaces to Marketo. They can be used to search, update, and modify information in Marketo.

Tables

The adapter models the data in Marketo into a list of tables that can be queried using standard SQL statements.

Generally, querying Marketo tables is the same as querying a table in a relational database. Sometimes there are special cases, for example, including a certain column in the WHERE clause might be required to get data for certain columns in the table. This is typically

needed for situations where a separate request must be made for each row to get certain columns. These types of situations are clearly documented at the top of the table page linked below.

Marketo Adapter Tables

Name	Description
CustomActivities	Query Custom Activities for a Marketo organization.
CustomObjects	Create, update, delete, and query custom objects for a Marketo organization.
Emails	Query Emails for a Marketo organization.
EmailTemplates	Query, update and delete EmailTemplates for a Marketo organization.
Folders	Create, update, delete, and query Folders for a Marketo organization.
Forms	Create, update, delete and query Forms for a Marketo organization.
LandingPages	Create, update, delete and query Landing Pages for a Marketo organization.
LandingPageTemplates	Create, update, delete and query LandingPageTemplates for a Marketo organization.
Leads	Create, update, delete, and query Leads for a Marketo organization.
ListStaticMemberShip	Create, delete and query query static list members for a Marketo organization.
NamedAccounts	Query Named Accounts for a Marketo organization.
Opportunities	Query Opportunities for a Marketo organization.

OpportunityRoles	Query Opportunity Roles for a Marketo organization.
ProgramMembers	Query Program members for a program.
Programs	Query Programs for a Marketo organization.
SalesPersons	Query Sales Persons for a Marketo organization.
SmartCampaigns	Create, update, delete, and query SmartCampaigns for a Marketo organization.
SmartLists	Query and delete SmartLists for a Marketo organization.
Snippets	Create, update, delete and query Snippets for a Marketo organization.
StaticLists	Create, update, delete and query Static Lists for a Marketo organization.
Tokens	Create, update, delete, and query Tokens for a Marketo organization.

CustomActivities

Query Custom Activities for a Marketo organization.

Table Specific Information

Each custom activity contained within your Marketo organization will be returned as it's own table. Each table name will be prefixed with 'Activity_' followed by the name of your custom activity.

Select

Custom activities can be retrieved by performing a SELECT query on the custom activity table.

```
SELECT * FROM Activity_MyCustomActivity
```

Insert

Custom activities can be added by performing an INSERT. To create a new custom activity record, specify the information about the custom activity to be entered into the database.

The following properties are required when creating a custom activity record: ActivityDate, LeadId, and PrimaryAttributeValue (note this column name changes based on the name you designated for the primary field of the activity).

The following example demonstrates how to insert a new custom activity called MyCustomActivity:

```
INSERT INTO Activity_MyCustomActivity (ActivityDate, PrimaryFieldValue,
LeadId, MyInt, Email) VALUES ('11/08/2016', '123', '1', 12345,
'insert@test.com')
```

Update

UPDATEs are not supported on custom activities.

Delete

DELETEs are not supported on custom activities.

Columns

Name	Type	ReadOnly	Filterable	Description
ActivityId [KEY]	Integer	True		The unique Id of the activity.
LeadId	Integer	True	True	The unique Id of the lead associated with the activity.
ActivityDate	Datetime	True	True	The date and time the lead was added

				to the list. Can be used as a filter to specify the starting date and time to retrieve all activities on or after the specified date.
PrimaryAttribute	<i>String</i>	True	True	The primary field Id
PrimaryAttributeValue	<i>String</i>	True		The primary field value.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
ListId	<i>Integer</i>	A List Id used to retrieve activities for all leads contained within the specified list.

CustomObjects

Create, update, delete, and query custom objects for a Marketo organization.

Table Specific Information

Each custom object contained within your Marketo organization will be returned as it's own table. Each table name will be prefixed with 'CustomObject_' followed by the name of your custom object.

Select

The Marketo REST API requires that a filter be specified to retrieve custom objects. The filter must contain at least one column that is the Key, a dedupeField, or a column that is searchable. Only the '=' operator is supported on these filter types. Other filters can be specified in addition to this but at least one of the previously mentioned filters must be specified. All filterable columns allow multiple values to be specified by using the IN operator or the OR logical operator.

Response time from the server can be improved by identifying only the rows and columns you want to retrieve.

```
SELECT Make, Model, Year, Color FROM CustomObject_MyCustomBikeObject
WHERE VIN='12345'
```

Insert

To create a new custom object record, specify the information about the custom object to be entered into the database.

The following example demonstrates how to insert a new custom object called MyCustomBikeObject:

```
INSERT INTO MyCustomBikeObject (VIN, Make, Model, Color) VALUES
('99999', 'Yamaha', 'FZ-09', 'Blue')
```

Update

Any field that is not read-only can be updated.

```
UPDATE MyCustomBikeObject SET Year=2016 WHERE VIN='99999'
```

Delete

Delete is used to remove custom objects from Marketo. To perform a delete, the table key or a dedupeField column must be specified.

```
DELETE FROM MyCustomBikeObject WHERE VIN='99999'
```

Columns

Name	Type	ReadOnly	Filterable	Description
MarketoGUID [KEY]	<i>String</i>	True		The unique, Marketo-assigned identifier of the custom object.
CreatedAt	<i>Datetime</i>	False		The datetime the custom object was created.
UpdatedAt	<i>Datetime</i>	False		The datetime the custom object was updated.

Emails

Query Emails for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of emails from the target instance, filterable by name.

```
SELECT * FROM Emails
SELECT * FROM Emails WHERE Name = 'CRUD Test'
```

Retrieve the email record for the given target Id.

```
SELECT * FROM Emails WHERE Id = 1192
```

Insert

To create a new Email, specify at least the Name, Template, FolderId and FolderType column.

```
INSERT INTO Emails (Name, Template, FolderId, FolderType) Values ('My Email', '1078', 2307, 'Folder')
```

Update

Any field that is not read-only can be updated.

```
UPDATE Emails SET Description='Testing Update' WHERE Id = 1192
```

Delete

To delete a Email you must specify the ID field.

```
DELETE FROM Emails WHERE Id=1192
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	The unique, Marketo-assigned identifier of the email.
Name	<i>String</i>	False	True	The name of the email.
Description	<i>String</i>	False		The description of the email.
Subject	<i>String</i>	False		The email subject.

FromName	<i>String</i>	False	The from name.
FromEmail	<i>String</i>	False	The from email address.
ReplyEmail	<i>String</i>	False	The reply email address.
FolderId	<i>Integer</i>	False	The Id of the folder where the email is located
FolderType	<i>String</i>	False	The type of the folder where the email is located.
FolderName	<i>String</i>	False	The name folder where the email is located.
Operational	<i>Boolean</i>	False	Identifies whether the email is operational.
TextOnly	<i>Boolean</i>	False	Identifies whether the email is text only.
PublishToMSI	<i>Boolean</i>	False	Identifies whether the email is published.
WebView	<i>Boolean</i>	False	Identifies whether the email is web view.
Status	<i>String</i>	False	The status of the email.
Version	<i>Integer</i>	False	The version of the email.
AutoCopyToText	<i>Boolean</i>	False	Identifies whether the email is auto copied to text.

Template	<i>Integer</i>	False	The template associated with the email.
Workspace	<i>String</i>	False	The name of the workspace where the email is located.
CreatedAt	<i>Datetime</i>	True	The date and time the email was created.
UpdatedAt	<i>Datetime</i>	True	The date and time the email was last updated.
PreHeader	<i>String</i>	False	The preheader text for the email.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
EarliestUpdatedAt	<i>Datetime</i>	Exclude emails prior to this date. Must be valid ISO-8601 string.
LatestUpdatedAt	<i>Datetime</i>	Exclude emails after this date. Must be valid ISO-8601 string.

EmailTemplates

Query, update and delete EmailTemplates for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of email templates from the target instance, filterable by name and status.

```
SELECT * FROM EmailTemplates
SELECT * FROM EmailTemplates WHERE Name = 'CRUD Test'
```

Retrieve the email record for the given target Id.

```
SELECT * FROM EmailTemplates WHERE Id = 1192
```

Update

Any field that is not read-only can be updated.

```
UPDATE EmailTemplates SET Name = 'Update Test' Description = 'Testing
Update' WHERE Id = 1192
```

Delete

To delete a Email you must specify the ID field.

```
DELETE FROM EmailTemplates WHERE Id = 1192
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	Integer	True	True	Id of the asset.

Name	<i>String</i>	False	True	Name of the asset.
CreatedAt	<i>Datetime</i>	True		Datetime the asset was created.
Description	<i>String</i>	False		Description of the asset.
FolderId	<i>Integer</i>	True		The Id of the folder
FolderType	<i>String</i>	True		The Type of folder The allowed values are <i>Folder</i> , <i>Program</i> .
FolderName	<i>String</i>	True		The Name of folder
Status	<i>String</i>	True	True	Status filter for draft or approved versions
UpdatedAt	<i>Datetime</i>	True		Datetime the asset was most recently updated
Url	<i>String</i>	True		Url of the asset in the Marketo UI
Version	<i>Integer</i>	True		The Template version type The allowed values are <i>1</i> , <i>2</i> .
Workspace	<i>String</i>	True		Name of the workspace
Content	<i>String</i>	True		HTML content for template. Multipart file.

Folders

Create, update, delete, and query Folders for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve all folders within two levels of the folder hierarchy.

```
SELECT * FROM Folders
```

Retrieve all folders under a specific root folder.

```
SELECT * FROM Folders WHERE RootFolderId=38 AND MaxDepth=5
```

Insert

To create a new Folder, specify at least the Name, ParentId and ParentType column.

```
INSERT INTO Folders(Name,ParentId,ParentType) Values('New_Folder_Marketo_CData_Driver', 38, 'Folder')
```

Update

Any field that is not read-only can be updated.

```
Update Folders Set Description='Updated Folder', IsArchive=true WHERE Id=1996
```

Delete

To delete a folder you must specify the ID field. Deletions can be made against single folders if they are empty, meaning that they contain no assets or subfolders. If a folder is of type Program, or has the isSystem field set to true, it cannot be deleted.

```
DELETE FROM Folders WHERE Id=1996
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	The unique, Marketo-assigned identifier of the folder.
Name	<i>String</i>	False	True	The name of the folder.
Description	<i>String</i>	False		The description of the folder.
Type	<i>String</i>	True		The type of the folder.
CreatedAt	<i>Datetime</i>	True		The date and time the folder was created.
UpdatedAt	<i>Datetime</i>	True		The date and time the folder was last updated.
ParentId	<i>Integer</i>	False		The Id of the parent folder.
ParentType	<i>String</i>	False		The type of the parent folder.
Path	<i>String</i>	True		The path of a folder shows its hierarchy in the folder tree, similar to a Unix-style path.
WorkSpace	<i>String</i>	True	True	The name of the smart campaign workspace.
URL	<i>String</i>	True		The explicit URL of the asset in the designated instance.

IsSystem	<i>Boolean</i>	True	Whether or not the folder is a system folder.
IsArchive	<i>Boolean</i>	False	Whether or not the folder is archived.
AccessZoneId	<i>Integer</i>	True	The access zone id

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
RootFolderId	<i>Integer</i>	The parent folder ID under which the query will be performed.
RootFolderType	<i>String</i>	<p>The parent folder type under which the query will be performed.</p> <p>The allowed values are <i>Folder</i>, <i>Program</i>.</p> <p>The default value is <i>Folder</i>.</p>
MaxDepth	<i>Integer</i>	<p>Maximum folder depth to traverse.</p> <p>The default value is 2.</p>

Forms

Create, update, delete and query Forms for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of accessible forms from the target instance.

```
SELECT * FROM Forms
```

Retrieve the form for the given Id.

```
SELECT * FROM Forms WHERE Id = '1214'
```

Insert

To create a new form, specify at least the Name, FolderId and FolderType column.

```
INSERT INTO Forms (Name, Description, FolderId, FolderType) Values ('My Snippet','Test Snippet insert', 1089, 'Program')
```

Update

Any field that is not read-only can be updated.

```
UPDATE Forms SET Description='Testing Update', Name='Test Update' WHERE Id = '1214'
```

Delete

To delete a Snippet you must specify the ID field.

```
DELETE FROM Forms WHERE Id = '1214'
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	Id of the asset.
Name	<i>String</i>	False	True	Name of the asset.
ButtonLabel	<i>String</i>	False		Label text of the button.
ButtonLocation	<i>Integer</i>	False		Location in pixels of the button relative to the left of the form.
CreatedAt	<i>Datetime</i>	True		Datetime the asset was created.
Description	<i>String</i>	False		Description of the asset.
FolderId	<i>Integer</i>	False	True	Id of the folder.
FolderType	<i>String</i>	False	True	Type of folder.
FontFamily	<i>String</i>	False		font-family property for the form.
FontSize	<i>String</i>	False		font-size property of the form.
KnownVisitorTemplate	<i>String</i>	False		Template of the known visitor behavior for the form.

KnownVisitorType	<i>String</i>	False		Type of the known visitor behavior for the form.
LabelPosition	<i>String</i>	False		Default positioning of labels.
Language	<i>String</i>	False		Language of the form.
Locale	<i>String</i>	False		Locale of the form.
ProgressiveProfiling	<i>Boolean</i>	False		Whether progressive profiling is enabled for the form.
Status	<i>String</i>	False	True	Status filter for draft or approved versions. The allowed values are <i>approved</i> , <i>draft</i> .
Theme	<i>String</i>	False		CSS theme for the form to use.
UpdatedAt	<i>Datetime</i>	True		Datetime the asset was most recently updated.
Url	<i>String</i>	False		Url of the asset in the Marketo UI.
WaitingLabel	<i>String</i>	False		Waiting text of the button.

LandingPages

Create, update, delete and query Landing Pages for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of accessible landing pages from the target instance, filterable by Status and folder.

```
SELECT * FROM LandingPages
SELECT * FROM LandingPages WHERE Status = 'draft'
SELECT * FROM LandingPages WHERE FolderId = 1184 AND FolderType =
'Program'
```

Retrieve the landing page record for the given name or a target Id.

```
SELECT * FROM LandingPages WHERE Id = 1234
SELECT * FROM LandingPages WHERE Name = 'Agenda'
```

Insert

To create a new LandingPage, specify at least the Name, FolderId, FolderType and Template column.

```
INSERT INTO LandingPages
(CustomHeadHTML,Description,FacebookOgTags,FolderId,FolderType,Keywords,
MobileEnabled,Name,FormPrefill,Robots,Template,Title,URL,Workspace)
VALUES ('<!DOCTYPE html>\n<html>\n<body>\n<h1>My First
Heading</h1>\n<p>My first paragraph.</p>\n</body></html>', 'Testing
Insert operation', '', 1184, 'Program', '', false, 'Test Insert', false, 'index,
nofollow', 1, 'Insert Operation', 'http://na-ab23.marketo.com/lp/119-IEY-
862/LPtest_08.html', 'CRH')
```

Update

Any field that is not read-only can be updated.

```
UPDATE LandingPages SET CustomHeadHTML = '<!DOCTYPE
html>\n<html>\n<body>\n<h1>My First Heading</h1>\n<p>My first
paragraph.</p>\n</body></html>', Description='Testing Update',
```

```
FacebookOgTags='', Keywords='', MobileEnabled=false, Name='Test Update',
Robots='index, nofollow', Title='Update Operation', URL='http://na-
ab23.marketo.com/lp/119-IEY-862/LPtest_08.html' WHERE Id = 1103
```

Delete

To delete a LandingPage you must specify the ID field.

```
DELETE FROM LandingPages WHERE Id = 1996
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	Id of the asset.
Name	<i>String</i>	False	True	Name of the asset.
ComputedUrl	<i>String</i>	True		Computed Url of the asset.
CreatedAt	<i>Datetime</i>	True		Datetime the asset was created.
CustomHeadHTML	<i>String</i>	False		Any custom HTML to embed in the tag of the page.
Description	<i>String</i>	False		Description of the asset.
FacebookOgTags	<i>String</i>	False		Any OpenGraph meta tags to apply to the page.

FolderId	<i>Integer</i>	False	True	Id of the folder.
FolderType	<i>String</i>	False	True	Type of folder. The allowed values are <i>Folder, Program</i> .
FolderName	<i>String</i>	False	False	Name of folder.
FormPrefill	<i>Boolean</i>	False		Boolean to toggle whether forms embedded in the page will prefill. Default false. The default value is <i>false</i> .
Keywords	<i>String</i>	False		Keywords
MobileEnabled	<i>Boolean</i>	False		Whether the page has mobile viewing enabled. Free-form pages only. Default false. The default value is <i>false</i> .
Robots	<i>String</i>	False		Robots directives to apply to the pages meta tags
Status	<i>String</i>	True	True	Status filter for draft or approved versions. The allowed values are <i>approved, draft</i> .
Template	<i>Integer</i>	False		Id of the template used.

Title	<i>String</i>	False	Title element of the landing page.
UpdatedAt	<i>Datetime</i>	True	Datetime the asset was most recently updated.
URL	<i>String</i>	False	Url of the asset in the Marketo UI. You have to send the URL path of the page while creating or updating.
Workspace	<i>String</i>	False	Name of the workspace.

LandingPageTemplates

Create, update, delete and query LandingPageTemplates for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of accessible landing page templates from the target instance, filterable by Name, Status and folder.

```
SELECT * FROM LandingPageTemplates
SELECT * FROM LandingPageTemplates WHERE Status='draft'
SELECT * FROM LandingPageTemplates WHERE FolderId=1184 AND
FolderType='Program'
```

Retrieve the landing page template record for the given name or a target Id.

```
SELECT * FROM LandingPageTemplates WHERE Id=1234
SELECT * FROM LandingPageTemplates WHERE Name='Agenda'
```

Insert

To create a new LandingPageTemplate, specify at least the Name, FolderId, FolderType and TemplateType column.

```
INSERT INTO LandingPageTemplates (Description, EnableMunchkin, FolderId,
FolderType, Name, TemplateType) VALUES ('Testing Insert', true, 19,
'Folder', 'Test Insert 1', 'guided')
```

Update

Any field that is not read-only can be updated.

```
UPDATE LandingPageTemplates SET Description='Testing Update',
EnableMunchkin=false, Name='Test Update' WHERE Id = 1312
```

Delete

To delete a LandingPageTemplate you must specify the ID field.

```
DELETE FROM LandingPageTemplates WHERE Id = 1312
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	Id of the asset.
Name	<i>String</i>	False	True	Name of the asset.
CreatedAt	<i>Datetime</i>	True		Datetime the asset was created.
Description	<i>String</i>	False		Description of the asset.

EnableMunchkin	<i>Boolean</i>	False		Whether to enable munchkin on the derived pages. Defaults to true. The default value is <i>true</i> .
FolderId	<i>Integer</i>	False	True	Id of the folder.
FolderType	<i>String</i>	False	True	Type of folder. The allowed values are <i>Folder, Program</i> .
FolderName	<i>String</i>	False		Name of folder.
Status	<i>String</i>	True	True	Status filter for draft or approved versions. The allowed values are <i>draft, approved</i> .
TemplateType	<i>String</i>	False		Type of template to create 'guided' or 'freeForm' The allowed values are <i>guided, freeForm</i> . The default value is <i>freeForm</i> .
UpdatedAt	<i>Datetime</i>	True		Datetime the asset was most recently updated.
Url	<i>String</i>	True		Url of the asset in the Marketo UI.
Workspace	<i>String</i>	True		Name of the workspace.

Leads

Create, update, delete, and query Leads for a Marketo organization.

Table Specific Information

Select

All columns must be specified using the '=' operator. All filterable columns allow multiple values to be specified by using the IN operator or the OR logical operator.

Response time from the server can be improved by identifying only the rows and columns you want to retrieve.

```
SELECT Id, FirstName, LastName FROM Leads WHERE Id IN (1, 2, 5, 10)
```

If a filter is not specified, the Activities_NewLead table will be queried to retrieve a list of Lead Ids. Once the Lead Ids have been compiled, the Leads table will be queried using the compiled list of Lead Ids. The 'CreatedAt' column can be used as a filter to specify the created datetime range of Leads to retrieve. When the '>' or '>=' operator is specified, the datetime value will be submitted in the Activities_NewLead table request. Note that this requires additional API calls and will at least double the amount of API calls made as one Activities_NewLead request must be made for each Leads request made. The maximum batch size per request for the REST API is 300, so to get a rough estimate of the number of API calls required, the following formula can be used: $(\text{Total Number of Leads} / 300) * 2$

The 'UpdatedAt' column can also be used as a filter to specify the updated datetime range of Leads to retrieve. When specified, the Activities_LeadChanges table will be queried to retrieve a list of Lead Ids. Once the Lead Ids have been compiled, the Leads table will be queried using the compiled list of Lead Ids. When the '>' or '>=' operator is specified, the datetime value will be submitted in the Activities_LeadChanges table request. Note that this requires additional API calls and will at least double the amount of API calls made as one Activities_LeadChanges request must be made for each Leads request made. In most cases, the total API calls will more than double as individual values for a Lead are returned as a record). Thus to get a full list of Lead Ids, multiple Activities_LeadChanges requests may need to be made. The maximum batch size per request for the REST API is 300, so to get a minimum estimate of the number of API calls required, the following formula can be used: $(\text{Total Number of Leads} / 300) * 2$

Alternatively, the SOAP API can be used to retrieve a list of Lead Ids by setting "UseSOAPForLeadIds=True" in the 'Other' property. When specified the SOAP API will be used (provided the SOAP connection details are specified) to compile a list of Lead Ids which will then be used as a filter for the REST API. Note that the SOAP API is significantly slower than the REST API, although this hybrid approach is faster than using just the SOAP API by itself. The maximum batch size per request for the SOAP API is 1000, so to get a minimum estimate of the number of API calls required, the following formula can be used: $(\text{Total Number of Leads} / 1000) + (\text{Total Number of Leads} / 300)$

To bypass utilizing multiple API calls and for the best performance, create a static list of Leads within Marketo and then specify the ListId to retrieve them.

Insert

To create a new Lead record, specify the information about the Lead to be entered into the database.

The following example demonstrates how to insert a new Lead:

```
INSERT INTO Leads (Email, FirstName, LastName) VALUES ('john@mark.com',
'John', 'Mark')
```

Update

Any field that is not read-only can be updated. Updates can be performed using an external key which is any 'Filterable' column, such as Email.

The following example demonstrates how to update using the 'Id' column as the lookup field.

```
UPDATE Leads SET MobilePhone='111-222-3333' WHERE Id = 1
```

The following example demonstrates how to update using the 'Email' column as the lookup field.

```
UPDATE Leads SET MobilePhone='111-222-3333' WHERE Email =
'john@mark.com'
```

Additionally you can use a custom field as the lookup field. In such a case, you will need to set the 'LookupField' column as well to the name of the custom field you are using. The

following example demonstrates how to update use a custom field called 'MyCustomField' as the lookup field.

```
UPDATE Leads SET MobilePhone='111-222-3333' WHERE LookupField =
'MyCustomField' AND MyCustomField = 'my_value'
```

Delete

Delete is used to remove leads from Marketo. To perform a delete, the lead Id field is required.

```
DELETE FROM Leads WHERE Id = 1
```

GetDeleted

GetDeleted is used to get deleted leads from Marketo. To get deleted leads, you need to specify the UpdatedAt filter. If not specified, it will fetch data for last 14 days.

```
GetDeleted FROM Leads
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	The unique, Marketo-assigned identifier of the account.
Email	<i>String</i>	False	True	The lead's email address.
Salutation	<i>String</i>	False		The lead's salutation.
FirstName	<i>String</i>	False		The lead's first

			name.
MiddleName	<i>String</i>	False	The lead's middle name.
LastName	<i>String</i>	False	The lead's last name.
DateOfBirth	<i>Date</i>	False	The lead's date of birth.
Title	<i>String</i>	False	The lead's job title.
Address	<i>String</i>	False	The lead's street address.
City	<i>String</i>	False	The lead's city.
State	<i>String</i>	False	The lead's state.
PostalCode	<i>String</i>	False	The lead's ZIP/postal code.
Country	<i>String</i>	False	The lead's country.
Website	<i>String</i>	False	The lead's website.
Phone	<i>String</i>	False	The lead's phone number.
MobilePhone	<i>String</i>	False	The lead's mobile phone number.
Fax	<i>String</i>	False	The lead's fax number.

Company	<i>String</i>	False	The name of the lead's company.
MainPhone	<i>String</i>	False	The phone number of the lead's company.
SICCode	<i>String</i>	False	The SIC (Standard Industrial Classification) code of the lead's company.
Site	<i>String</i>	False	The site of the lead's company.
BillingStreet	<i>String</i>	False	The billing street address of the lead's company.
BillingCity	<i>String</i>	False	The billing city of the lead's company.
BillingState	<i>String</i>	False	The billing state of the lead's company.
BillingPostalCode	<i>String</i>	False	The billing ZIP/postal code of the lead's company.
BillingCountry	<i>String</i>	False	The billing country of the lead's company.
NumberOfEmployees	<i>Integer</i>	False	The number of employees at the lead's company.

Industry	<i>String</i>	False	The industry of the lead's company.
Department	<i>String</i>	False	The lead's department.
AnnualRevenue	<i>Double</i>	False	The annual revenue generated at the lead's company.
AnonymousIP	<i>String</i>	False	The IP address of the lead if it is anonymous.
Unsubscribed	<i>Boolean</i>	False	Determines whether the lead is unsubscribed.
UnsubscribedReason	<i>String</i>	False	The reason why the lead has unsubscribed.
EmailInvalid	<i>Boolean</i>	False	Identifies whether the lead's email address is invalid.
EmailInvalidCause	<i>String</i>	False	The reason why the lead's email address is invalid.
DoNotCall	<i>Boolean</i>	False	Identifies whether the lead is on the 'Do Not Call' list.
DoNotCallReason	<i>String</i>	False	The reason why the lead is on the 'Do Not Call' list.
PersonType	<i>String</i>	False	The type of person

			the current record is, such as a contact.
IsAnonymous	<i>Boolean</i>	True	Identifies whether the lead is anonymous or not.
IsLead	<i>Boolean</i>	False	Identifies whether the person is a lead or not.
LeadRole	<i>String</i>	False	The lead's role.
LeadSource	<i>String</i>	False	The lead's source.
LeadStatus	<i>String</i>	False	The lead's current status.
LeadScore	<i>Integer</i>	False	The lead's score.
Rating	<i>String</i>	False	The lead's rating.
Urgency	<i>Double</i>	True	The lead's urgency.
Priority	<i>Integer</i>	True	The lead's priority.
RelativeScore	<i>Integer</i>	True	The lead's relative score.
OriginalSourceType	<i>String</i>	True	The original source type where the lead originated from.

OriginalSourceInfo	<i>String</i>	True		Information about the original source of the lead.
RegistrationSourceType	<i>String</i>	False		The original source type where the lead originated from.
RegistrationSourceInfo	<i>String</i>	False		Information about the original source of the lead.
CreatedAt	<i>Datetime</i>	True	True	The date the lead was created.
UpdatedAt	<i>Datetime</i>	True	True	The date the lead was last updated.
Cookies	<i>String</i>	False	True	The cookies associated with the lead.
AcquisitionProgramId	<i>String</i>	False		The Id of the program in which the lead was acquired.
Gender	<i>String</i>	False		The lead's social gender.
TotalReferredVisits	<i>Integer</i>	True		The lead's total social referred visits.
TotalReferredEnrollments	<i>Integer</i>	True		The lead's total social referred enrollments.
LastReferredEnrollment	<i>Datetime</i>	False		The lead's last

				social referred enrollment.
LastReferredVisit	<i>Datetime</i>	False		The lead's last social referred visit.
SyndicationId	<i>String</i>	False		The lead's social syndication Id.
FacebookDisplayName	<i>String</i>	False		The lead's Facebook display name.
FacebookId	<i>String</i>	False	True	The lead's Facebook Id.
FacebookPhotoURL	<i>String</i>	False		The lead's Facebook photo URL.
FacebookProfileURL	<i>String</i>	False		The lead's Facebook profile URL.
FacebookReach	<i>Integer</i>	False		The lead's Facebook reach.
FacebookReferredEnrollments	<i>Integer</i>	False		The lead's Facebook referred enrollments.
FacebookReferredVisits	<i>Integer</i>	False		The lead's Facebook referred visits.
LinkedInDisplayName	<i>String</i>	False	True	The lead's LinkedIn display name.

LinkedInId	<i>String</i>	False	The lead's LinkedIn Id.
LinkedInPhotoURL	<i>String</i>	False	The lead's LinkedIn photo URL.
LinkedInProfileURL	<i>String</i>	False	The lead's LinkedIn profile URL.
LinkedInReach	<i>Integer</i>	False	The lead's LinkedIn reach.
LinkedInReferredEnrollments	<i>Integer</i>	False	The lead's LinkedIn referred enrollments.
LinkedInReferredVisits	<i>Integer</i>	False	The lead's LinkedIn referred visits.
TwitterDisplayName	<i>String</i>	False	The lead's Twitter display name.
TwitterId	<i>String</i>	False	The lead's Twitter Id.
TwitterPhotoURL	<i>String</i>	False	The lead's Twitter photo URL.
TwitterProfileURL	<i>String</i>	False	The lead's Twitter profile URL.
TwitterReach	<i>Integer</i>	False	The lead's Twitter reach.
TwitterReferredEnrollments	<i>Integer</i>	False	The lead's Twitter referred enrollments.

TwitterReferredVisits	<i>Integer</i>	False		The lead's Twitter referred visits.
ListId	<i>Integer</i>	True	True	A List Id used to retrieve all leads contained within the specified list. This is a filter only field made available to enhance filtering capabilities.
ProgramId	<i>Integer</i>	True	True	A Program Id used to retrieve all leads associated with the specified program. This is a filter only field made available to enhance filtering capabilities.
LookupField	<i>String</i>	True	True	Used to specify the field used to find duplicate leads. Only used when performing an INSERT or UPDATE. Available values are: id (default), cookie, email, twitterId, facebookId, linkedInId, sfdcAccountId, sfdcContactId, sfdcLeadId, sfdcLeadOwnerId, and custom fields.

This is a filter only field made available to enhance filtering capabilities.

ListStaticMemberShip

Create, delete and query query static list members for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of Static MemberShip from the target instance, filterable by Id and ListId.

```
SELECT * FROM ListStaticMemberShip WHERE ListId = 1014
SELECT * FROM ListStaticMemberShip WHERE ListId = 1014 AND id = 1016
```

Insert

To create a new Static MemberShip, specify at least the ListId and Id column.

```
INSERT INTO ListStaticMemberShip (ListId,Id) VALUES (1014,1014)
```

Delete

To delete a ListStaticMemberShip you must specify the ID and ListId field.

```
DELETE FROM ListStaticMemberShip WHERE Listid = 1014 AND Id = 1014
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	Integer	True	True	Id of static MemberShip.
FirstName	String	True		FirtName of the member
LastName	String	True		LastName of the member
Email	String	True		Email
CreatedAt	Datetime	True		The date and time the membership was created.
UpdatedAt	Datetime	True		The date and time the membership was last updated.
ListId	Integer	True	True	ListId.

NamedAccounts

Query Named Accounts for a Marketo organization.

Table Specific Information

Select

A filter must be specified when retrieving named accounts. Valid filters are any searchable columns which include MarketoGUID, Name, Industry, State, City, etc.

```
SELECT * FROM NamedAccounts WHERE Name='MyAccount '
```

Insert

To create a new NamedAccount record, specify the information about the named account to be entered into the database.

The following example demonstrates how to insert a new NamedAccount:

```
INSERT INTO NamedAccounts (Name, City, Country, Industry) VALUES
('MyAccount', 'MyCity', 'USA', 'Tech')
```

Update

Any field that is not read-only can be updated.

```
UPDATE NamedAccounts SET NumberOfEmployees=100, State='NC',
AnnualRevenue='10000000.00' WHERE Name='MyAccount'
```

Delete

Delete is used to remove named accounts from Marketo. To perform a delete, either the MarketoGUID field or the Name field is required.

```
DELETE FROM NamedAccounts WHERE Name=='MyAccount'
```

Columns

Name	Type	ReadOnly	Filterable	Description
MarketoGUID [KEY]	<i>String</i>	True	True	The unique, Marketo-assigned identifier of the named account.
Name	<i>String</i>	False	True	The name of the account.
AccountOwnerId	<i>Integer</i>	False	True	The owner id of the account.

AnnualRevenue	<i>Double</i>	False	True	The annual revenue for the account.
City	<i>String</i>	False	True	The city for the account.
Country	<i>String</i>	False	True	The country for the account.
DomainName	<i>String</i>	False	True	The name of the domain for the account.
Industry	<i>String</i>	False	True	The industry for the account.
LogoURL	<i>String</i>	False	True	The URL to the logo for the account.
MembershipCount	<i>Integer</i>	True	True	The number of members for the account.
NumberOfEmployees	<i>Integer</i>	False	True	The number of employees for the account.
OpptyAmount	<i>Double</i>	True	True	The total amount of opportunities for the account.
OpptyCount	<i>Integer</i>	True	True	The total number of opportunities for the account.
SICCode	<i>String</i>	False	True	The SIC Code for the account.
State	<i>String</i>	False	True	The state for the account.

CreatedAt	<i>Datetime</i>	True	The date and time the named account was created.
UpdatedAt	<i>Datetime</i>	True	The date and time the named account was last updated.

Opportunities

Query Opportunities for a Marketo organization.

Table Specific Information

Note: This table is only available for Marketo subscriptions which do not have a native CRM sync enabled. If sync is enabled, an error will be returned when attempting to query the table stating that the API is disabled.

Select

A filter must be specified when retrieving opportunities. Valid filters are any searchable columns which include MarketoGUID, ExternalOpportunityId, ExternalCompanyId, or ExternalSalesPersonId.

```
SELECT * FROM Opportunities WHERE ExternalOpportunityId='CDATA1'
```

Insert

To create a new Opportunity record, specify the information about the Opportunity to be entered into the database.

The following example demonstrates how to insert a new Opportunity:

```
INSERT INTO Opportunities (ExternalOpportunityId, Description,
ExternalCompanyId, Name) VALUES ('CDATA1', 'CData Software Inc
Opportunity', 'CDATA', 'CData')
```


Update

Any field that is not read-only can be updated.

```
UPDATE Opportunities SET IsWon=true, FiscalYear=2016, Amount='1000.00'
WHERE ExternalOpportunityId='Opportunity1'
```

Delete

Delete is used to remove opportunities from Marketo. To perform a delete, either the Id field or the ExternalOpportunityId field is required.

```
DELETE FROM Opportunities WHERE ExternalOpportunityId='Opportunity1'
```

Columns

Name	Type	ReadOnly	Filterable	Description
MarketoGUID [KEY]	String	True	True	The unique, Marketo-assigned identifier of the opportunity.
ExternalOpportunityId	String	False	True	The external Id of the opportunity.
Amount	Double	False		The amount of the opportunity.
CloseDate	Datetime	False		The date and time the opportunity was closed.
Description	String	False		The description of the opportunity.
ExpectedRevenue	Double	False		The expected

				revenue of the opportunity.
ExternalCompanyId	<i>String</i>	False	True	The external company Id of the opportunity.
ExternalCreatedDate	<i>Datetime</i>	False		The external date and time the opportunity was created.
ExternalSalesPersonId	<i>String</i>	False	True	The external sales person Id of the opportunity.
Fiscal	<i>String</i>	False		The fiscal of the opportunity.
FiscalQuarter	<i>String</i>	False		The fiscal quarter of the opportunity.
FiscalYear	<i>String</i>	False		The fiscal year of the opportunity.
ForecastCategoryName	<i>String</i>	False		The forecast category name of the opportunity.
IsClosed	<i>Boolean</i>	False		Specifies whether the opportunity is closed.
IsWon	<i>Boolean</i>	False		Specifies whether the opportunity was won.
LastActivityDate	<i>Datetime</i>	False		The date and time the last activity occurred on the

			opportunity.
LeadSource	<i>String</i>	False	The lead source of the opportunity.
Name	<i>String</i>	False	The name of the opportunity.
NextStep	<i>String</i>	False	The next step of the opportunity.
Probability	<i>Integer</i>	False	The probability of the opportunity.
Quantity	<i>Double</i>	False	The quantity of the opportunity.
Stage	<i>Double</i>	False	The stage of the opportunity.
Type	<i>Double</i>	False	The type of the opportunity.
CreatedAt	<i>Datetime</i>	True	The date and time the opportunity was created.
UpdatedAt	<i>Datetime</i>	True	The date and time the opportunity was last updated.

OpportunityRoles

Query Opportunity Roles for a Marketo organization.

Table Specific Information

Note: This table is only available for Marketo subscriptions which do not have a native CRM sync enabled. If sync is enabled, an error will be returned when attempting to query the

table stating that the API is disabled.

Select

A filter must be specified when retrieving opportunities. Valid filters are any searchable columns which include MarketoGUID and a combination of ExternalOpportunityId and LeadId.

```
SELECT * FROM OpportunityRoles WHERE  
ExternalOpportunityId='Opportunity1' AND LeadId='1'
```

Insert

To create a new Opportunity Role record, specify the information about the Opportunity Role to be entered into the database.

The following example demonstrates how to insert a new Opportunity Role:

```
INSERT INTO OpportunityRoles (ExternalOpportunityId, LeadId, IsPrimary,  
Role) VALUES ('CDATA1', '1', false, 'MyRole')
```

Update

Any field that is not read-only can be updated.

```
UPDATE OpportunityRoles SET IsPrimary=true WHERE MarketoGUID='c674bda8-  
6e94-40cf-a853-98833b85b7cb'
```

Delete

Delete is used to remove Opportunity Roles from Marketo. To perform a delete, the MarketoGUID field is required.

```
DELETE FROM OpportunityRoles WHERE MarketoGUID='c674bda8-6e94-40cf-  
a853-98833b85b7cb'
```

Columns

Name	Type	ReadOnly	Filterable	Description
MarketoGUID [KEY]	<i>String</i>	True	True	The unique, Marketo-assigned identifier of the opportunity role.
ExternalOpportunityId	<i>String</i>	False	True	The external Id of the opportunity.
ExternalCreatedDate	<i>Datetime</i>	False		The external date and time the opportunity role was created.
IsPrimary	<i>Boolean</i>	False		Specifies whether the opportunity role is the primary role on the opportunity.
LeadId	<i>Integer</i>	False	True	The lead Id associated with the opportunity role.
Role	<i>String</i>	False		The role associated with the opportunity.
CreatedAt	<i>Datetime</i>	True		The date and time the opportunity role was created.
UpdatedAt	<i>Datetime</i>	True		The date and time the opportunity role was last updated.

ProgramMembers

Query Program members for a program.

Table Specific Information

Select

Retrieve all Program Members for the specific ProgramId.

Note:

- When **UseBulkAPI=true** and ProgramId is not specified, it will fetch the first ProgramId. We can only use =, IN operator with ProgramID.
- When **UseBulkAPI=false** and ProgramId is not specified, it will fetch records for all the ProgramId. We can specify the =, IN, >, <, <=, >= operator with ProgramID.

```
SELECT * FROM ProgramMembers WHERE ProgramId = 1102
SELECT * FROM ProgramMembers WHERE ProgramId = '1044' AND LeadId IN
('1789','1789','1790','1791','1792')
SELECT * FROM ProgramMembers WHERE ProgramId = '1001' AND LeadId = '4'
SELECT * FROM ProgramMembers WHERE ProgramId IN (1102, 1103, 1104)
SELECT * FROM ProgramMembers WHERE LeadId IN
('1789','1790','1791','1792') AND reachedSuccess IN (false,true)
SELECT * FROM ProgramMembers WHERE reachedSuccess IN (false,true) AND
LeadId = '1789'
```

Insert

To create a new Program Member, specify at least the LeadId, StatusName and ProgramId column. This operation is only supported when UseBulkApi=false.

```
INSERT INTO ProgramMembers (LeadId,StatusName,ProgramId) VALUES
(4, 'member', '1001')
```

Update

Fields WebinarURL and RegistrationCode can be updated. To update any Program Member, specify at least the LeadId, ProgramId column. This operation is only supported when UseBulkApi=false.

```
UPDATE ProgramMembers SET WebinarURL =
'www.testURL.com',RegistrationCode = 'dcff5f12-a7c7-11eb-bcbc-
0242ac130001' WHERE LeadId = '4' AND ProgramId = '1001'
```

Delete

To delete a Program member you must specify the LeadId and ProgramId field. This operation is only supported when UseBulkApi=false.

```
DELETE FROM ProgramMembers WHERE LeadId = '4' AND ProgramId = '1001'
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id	<i>Integer</i>	True		The Id of the Program Member
AcquiredBy	<i>Boolean</i>	False		Indicates this program was responsible for creating the lead record
AttendanceLikelihood	<i>Integer</i>	False		The likelihood of the attendance at the individual level
LeadId [KEY]	<i>Integer</i>	False	True	The Id of the Lead
MemberDate	<i>Datetime</i>	False		The date of the membership
NurtureProgramId	<i>String</i>	True		The Id of the Nurture Program

Program	<i>String</i>	False		The Program with which the member is associated
ProgramId [KEY]	<i>Integer</i>	False	True	The Id of the Program
ProgramTypeId	<i>String</i>	True		The Id of the Program Type
RegistrationCode	<i>String</i>	False		The registration code
RegistrationLikelihood	<i>String</i>	False		The registration likelihood
Status	<i>String</i>	False		The Status of the Program Member
StatusId	<i>String</i>	True		The Id of the Status of the Program Member
StatusReason	<i>String</i>	False		The Reason of the Status of the Program Member
StatusName	<i>String</i>	False	True	The name of the Status of the Program Member
ReachedSuccess	<i>Boolean</i>	False	True	Indicates when a person reaches the status that achieves that goal
CreatedAt	<i>Datetime</i>	False		The created date of the membership

IsExhausted	<i>Boolean</i>	False	Indicates whether the membership is exhausted or not
MemberShipDate	<i>Datetime</i>	False	The date of the membership
NurtureCadence	<i>String</i>	False	The Nurture Program
ReachedSuccessDate	<i>Datetime</i>	False	The date when a person reaches the status that achieves that goal
UpdatedAt	<i>Datetime</i>	False	The updated date of the membership
Success	<i>Boolean</i>	False	Indicates when a person reaches the status that achieves that goal
Track	<i>String</i>	False	The track of the Program Member Activity
TrackName	<i>String</i>	False	The name of the track
WaitlistPriority	<i>Integer</i>	False	Indicates the program member waitlist priority
WebinarURL	<i>String</i>	False	The Webinar URL

Programs

Query Programs for a Marketo organization.

Table Specific Information

Select

Note: Tag and Cost columns are not returned when browsing all Programs (such as performing a `SELECT *` query). These columns are only returned when filtering by a specific Program Id or Name.

Tag and Cost Columns are not returned in this case.

```
SELECT * FROM Programs
```

Tag and Cost Columns are returned in this case.

```
SELECT * FROM Programs WHERE Id='1001'
```

Insert

To create a new Program record, specify the information about the Program to be entered into the database.

The following example demonstrates how to insert a new Program:

```
INSERT INTO Programs (Name, FolderId, FolderType, Type, Description,
Channel, TagTypes, TagValues, CostStartDates, Costs, CostNotes) VALUES
('InsertEvent', '35', 'Folder', 'Default', 'Test Insert Description',
'Email Blast', 'Program Owner', 'Admin', '01/01/2015,02/02/2015',
'100,200', 'January,February')
```

Update

Any field that is not read-only can be updated.

```
UPDATE Programs SET Name='UpdatedProgram', Description='Updated
Description' WHERE Id='1'
```

Delete

Delete is used to remove programs from Marketo. To perform a delete, the program Id field is required.

```
DELETE FROM Programs WHERE Id='1'
```

Note: FolderId and FolderName can be included when **inserting** a new record, but they cannot be **updated** in existing records (read-only once a record exists).

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	The unique, Marketo-assigned identifier of the program.
Name	<i>String</i>	False	True	The name of the program.
Description	<i>String</i>	False		The description of the program.
Type	<i>String</i>	False		The program type.
Channel	<i>String</i>	False		The channel the program is associated with.
Workspace	<i>String</i>	False		The name of the workspace where the program is located.
Url	<i>String</i>	True		The URL reference to the program.
Status	<i>String</i>	False		The status of the program.

FolderType	<i>String</i>	False		The folder type that the program is contained in.
FolderId	<i>Integer</i>	False		The folder id that the program is contained in.
FolderName	<i>String</i>	False		The name of the folder the program is contained in.
TagTypes#	<i>String</i>	False	True	A comma-separated list of tag types associated with the program. Each TagType has a value associated with it which is returned via the TagValue column.
TagValues#	<i>String</i>	False	True	A comma-separated list of tag values. Each value corresponds to the type listed within the TagTypes column.
CostStartDates#	<i>String</i>	False		A comma-separated list of cost start dates. Each value corresponds to the costs and notes listed within the Costs and CostNotes columns.
Costs#	<i>String</i>	False		A comma-separated list of costs (integer values). Each value corresponds to the start dates and notes listed within the CostStartDates and CostNotes columns.
CostNotes#	<i>String</i>	False		A comma-separated list of cost notes. Each value corresponds to the costs

			and start dates listed within the Costs and CostStartDates columns.
CreatedAt	Datetime	True	The date and time the program was created.
UpdatedAt	Datetime	True	The date and time the program was last updated.

SalesPersons

Query Sales Persons for a Marketo organization.

Table Specific Information

Note: This table is only available for Marketo subscriptions which do not have a native CRM sync enabled. If sync is enabled, an error will be returned when attempting to query the table stating that the API is disabled.

Select

A filter must be specified when retrieving companies. Valid filters are any searchable columns which include Id, ExternalSalesPersonId, or Email.

```
SELECT * FROM SalesPersons WHERE ExternalSalesPersonId='sales@cdata.com'
```

Insert

To create a new SalesPerson record, specify the information about the sales person to be entered into the database.

The following example demonstrates how to insert a new Opportunity:

```
INSERT INTO SalesPersons (ExternalSalesPersonId, Email, FirstName,
LastName) VALUES ('sales@cdata.com', 'sales@cdata.com', 'Sales',
'Person')
```

Update

Any field that is not read-only can be updated.

```
UPDATE SalesPersons SET Phone='919-928-5214', Title='Technical Sales',
Email='sales@cdata.com' WHERE ExternalSalesPersonId='sales@cdata.com'
```

Delete

Delete is used to remove a sales person from Marketo. To perform a delete, either the Id field or the ExternalSalesPersonId field is required.

```
DELETE FROM SalesPersons WHERE ExternalSalesPersonId=='sales@cdata.com'
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	The unique, Marketo-assigned identifier of the sales person.
ExternalSalesPersonId	<i>String</i>	False	True	The external Id of the sales person.
Email	<i>String</i>	False	True	The email address of the sales person.
Fax	<i>String</i>	False		The fax number of the sales person.
FirstName	<i>String</i>	False		The first name of the sales person.
LastName	<i>String</i>	False		The last name of the sales person.

MobilePhone	<i>String</i>	False	The mobile phone number of the sales person.
Phone	<i>String</i>	False	The phone number of the sales person.
Title	<i>String</i>	False	The sales person's title.
CreatedAt	<i>Datetime</i>	True	The date and time the sales person was created.
UpdatedAt	<i>Datetime</i>	True	The date and time the sales person was last updated.

SmartCampaigns

Create, update, delete, and query SmartCampaigns for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve all smart campaigns

```
SELECT * FROM SmartCampaigns
```

Retrieve a specific smart campaign

```
SELECT * FROM SmartCampaigns WHERE Id=2046
```

Insert

To create a new smart campaign, specify at least the Name, FolderId and FolderType column.

```
INSERT INTO SmartCampaigns(Name,FolderId,FolderType) VALUES
('NewSmartCampaign','1357','Folder')
```

Update

Only the Name and Description columns can be updated.

```
Update SmartCampaigns Set Name='UpdatedSmartCampaignName',
Description='CData Campaign' WHERE Id=2047
```

Delete

To delete a smart campaign you must specify the ID field.

```
DELETE FROM SmartCampaigns WHERE Id=2047
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	The Id of the smart campaign.
Name	<i>String</i>	False		The name of the smart campaign.
ComputedUrl	<i>String</i>	False		The Computed Url of the Smart Campaign
Description	<i>String</i>	False		The description of

			the smart campaign.
Type	<i>String</i>	True	The type of the the smart campaign. Batch: has at least one filter and no triggers. Trigger: has at least one trigger. Default: has no smart list rules.
SmartListId	<i>Integer</i>	True	The Id of the smart campaign's child smart list.
FlowId	<i>Integer</i>	True	The Id of the smart campaign's child flow.
CreatedAt	<i>Datetime</i>	True	The date and time the smart campaign was created.
UpdatedAt	<i>Datetime</i>	True	The date and time the smart campaign was last updated.
WorkSpace	<i>String</i>	True	The name of the workspace where the folder is located.
Status	<i>String</i>	True	The status of the smart campaign. The allowed values are <i>Inactive</i> , <i>Single Run</i> , <i>Invalid</i> , <i>Recurring Run</i> , <i>Active</i> , <i>Requested</i> , <i>Never Run</i> .

IsSystem	<i>Boolean</i>	True	Whether smart campaign is system managed.
IsActive	<i>Boolean</i>	True	Whether smart campaign is active.
IsRequestable	<i>Boolean</i>	True	Whether smart campaign is requestable (is active and contains 'Campaign is Requested' trigger with Source of 'Web Service API').
IsCommunicationLimitEnabled	<i>Boolean</i>	True	Whether smart campaign communication limit is enabled (i.e. block non-operational emails).
MaxMembers	<i>Integer</i>	True	The smart campaign membership limit.
QualificationRuleType	<i>String</i>	True	<p>The type of qualification rule.</p> <p>The allowed values are <i>once</i>, <i>any</i>, <i>interval</i>.</p>
QualificationRuleInterval	<i>Integer</i>	True	<p>The interval of qualification rule.</p> <p>Only set when qualificationRuleType is 'interval'</p>
QualificationRuleUnit	<i>String</i>	True	The unit of measure

			of qualification rule. Only set when qualificationRuleType is 'interval' = ['hour', 'day', 'week', 'month']
RecurrenceStartAt	<i>Datetime</i>	True	The datetime of the first scheduled campaign to run. Required if setting recurrence. Not required to create a smart campaign that has no recurrence.
RecurrenceEndAt	<i>Datetime</i>	True	The datetime after which no further runs will be automatically scheduled.
RecurrenceIntervalType	<i>String</i>	True	The recurrence interval. Not required to create a smart campaign that has no recurrence = ['Daily', 'Weekly', 'Monthly'].
RecurrenceInterval	<i>Integer</i>	True	The number of interval units between recurrences.
RecurrenceWeekDayOnly	<i>Boolean</i>	True	Only run smart campaign on weekdays. May only be set if intervalType is 'Daily'.

RecurrenceWeekDayMask	<i>String</i>	True	String array of empty or one or more of 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'. May only be set if intervalType is 'Weekly'.
RecurrenceDayOfMonth	<i>Integer</i>	True	The day of the month to recur. Permissible range 1-31. May only be set if intervalType is 'Monthly' and dayOfWeek and weekOfMonth are unset.
RecurrenceDayOfWeek	<i>String</i>	True	The day of the week to recur. May only be set if dayOfMonth is not set, and weekOfMonth is set = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'].
RecurrenceWeekOfMonth	<i>Integer</i>	True	The week of the month to recur. Permissible range 1-4. May only be set if dayOfMonth is not set, and dayOfWeek is set.
FolderId	<i>Integer</i>	False	The Id of the folder.

FolderType	String	False	The type of folder. The allowed values are <i>Folder</i> , <i>Program</i> .
------------	--------	-------	--

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
EarliestUpdatedAt	Datetime	Exclude smart campaigns prior to this date.
LatestUpdatedAt	Datetime	Exclude smart campaigns after this date.
Folder	String	JSON representation of parent folder, with members 'id', and 'type' which may be 'Folder' or 'Program'.

SmartLists

Query and delete SmartLists for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve all smart lists

```
SELECT * FROM SmartLists
```

Retrieve a specific smart list

```
SELECT * FROM SmartLists WHERE Id=1142
```

Delete

To delete a smart list you must specify the ID field.

```
DELETE FROM SmartLists WHERE Id=1142
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	The Id of the smart list.
Name	<i>String</i>	True	True	The name of the smart list.
CreatedAt	<i>Datetime</i>	True		The date and time the smart list was created.
Description	<i>String</i>	True		The description of the Smart list
UpdatedAt	<i>Datetime</i>	True		The date and time the smart list was last updated.
WorkSpace	<i>String</i>	True		The name of the workspace where the smart list is located.

Url	<i>String</i>	True	The url of the smart list.
FolderId	<i>Integer</i>	False	The Id of the folder.
FolderType	<i>String</i>	False	The type of folder. The allowed values are <i>Folder</i> , <i>Program</i> .

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
EarliestUpdatedAt	<i>Datetime</i>	Exclude smart list prior to this date.
LatestUpdatedAt	<i>Datetime</i>	Exclude smart list after this date.
Folder	<i>String</i>	JSON representation of parent folder, with members 'id', and 'type' which may be 'Folder' or 'Program'.
SmartCampaignId	<i>Integer</i>	The Id of the smart campaign
ProgramId	<i>Integer</i>	The Id of the Program

Snippets

Create, update, delete and query Snippets for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of accessible snippets from the target instance, filterable by Status.

```
SELECT * FROM Snippets  
SELECT * FROM Snippets WHERE Status='draft'
```

Retrieve the snippet record for the given target Id.

```
SELECT * FROM Snippets WHERE Id=1234
```

Insert

To create a new Snippet, specify at least the Name, FolderId and FolderType column.

```
INSERT INTO Snippets (Name,Description,FolderId,FolderType) Values ('My  
Snippet','Test Snippet insert',31,'Folder')
```

Update

Any field that is not read-only can be updated.

```
UPDATE Snippets SET Description='Testing Update', IsArchive='No',  
Name='Test Update' WHERE Id=8
```

Delete

To delete a Snippet you must specify the ID field.

```
DELETE FROM Snippets WHERE Id=1934
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	Id of the asset.
Name	<i>String</i>	False		Name of the asset.
CreatedAt	<i>Datetime</i>	True		Datetime the asset was created.
Description	<i>String</i>	False		Description of the asset.
FolderId	<i>Integer</i>	False		Id of the folder.
FolderType	<i>String</i>	False		Type of folder. The allowed values are <i>Folder</i> , <i>Program</i> .
FolderName	<i>String</i>	False		Name of folder.
Status	<i>String</i>	True	True	Status filter for draft or approved versions.
UpdatedAt	<i>Datetime</i>	True		Datetime the asset was most recently updated.
Url	<i>String</i>	True		Url of the asset in the Marketo UI.
Workspace	<i>String</i>	True		Name of the workspace.
IsArchive	<i>String</i>	False		Archival status of the snippet

StaticLists

Create, update, delete and query Static Lists for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of Static List from the target instance, filterable by name.

```
SELECT * FROM StaticLists
SELECT * FROM StaticLists WHERE Name = 'test0319'
```

Retrieve the Static List record for the given target Id.

```
SELECT * FROM StaticLists WHERE Id = 1192
```

Retrieve the Static List record for the given folder.

```
SELECT * FROM StaticLists WHERE Folder = '{id:12,type:Folder}'
```

Insert

To create a new Static List, specify at least the FolderId,FolderType,Name and FolderName column.

```
INSERT INTO StaticLists(FolderId,FolderType,Name ,FolderName) VALUES
(12,'folder','testvs' , 'ManualList')
```

Update

Any field that is not read-only can be updated.

```
UPDATE StaticLists SET Name = 'testupdate' WHERE Id = 1058
```

Delete

To delete a Email you must specify the ID field.

```
DELETE FROM StaticLists WHERE Id = 1058
```

Columns

Name	Type	ReadOnly	Filterable	Description
Id [KEY]	<i>Integer</i>	True	True	Id of the static list.
Name	<i>String</i>	False	True	Name of the static list.
CreatedAt	<i>Datetime</i>	True		Datetime the static list was created.
UpdatedAt	<i>Datetime</i>	True		Datetime the static list was most recently updated.
FolderId	<i>Integer</i>	False		Id of the folder.
FolderType	<i>String</i>	False		Type of folder.
FolderName	<i>String</i>	False		Type of folder.
ComputedUrl	<i>String</i>	False		Computed urls of static list.
Workspace	<i>String</i>	False		Workspace of static list.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
Folder	<i>String</i>	The folder parameter can be used to specify the parent folder under which the query will be performed

Tokens

Create, update, delete, and query Tokens for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve Tokens under a specific ParentResourceType.

```
SELECT * FROM Tokens WHERE ParentResourceId = 1121 AND  
ParentResourceType = 'program'
```

Insert

To create a new query Tokens, specify ParentResourceId, ParentResourceType, Name, Type, and Value fields.

```
INSERT INTO Tokens (ParentResourceId, ParentResourceType, Name, Type,  
Value) VALUES (1111, 'program', 'testname', 'text', 'testvalue')
```

Update

To Update a Token you must specify the ParentResourceId, ParentResourceType, Name, Type and Value fields.

```
UPDATE Tokens SET value = 'testvalue', Name = 'testname', Type = 'text'
WHERE ParentResourceId = 1 AND ParentResourceType = 'folder'
```

Delete

To Delete a Token you must specify the ParentResourceId, ParentResourceType, Name and Type fields.

```
DELETE FROM Tokens WHERE ParentResourceId = 1 AND ParentResourceType =
'program' AND Name = 'testname' AND Type = 'text'
```

Columns

Name	Type	ReadOnly	Filterable	Description
ParentResourceId	<i>Integer</i>	True	True	The Id of the Folder or Program.
ParentResourceType	<i>String</i>	True	True	<p>The type of the token. It could be either Folder or Program.</p> <p>The allowed values are <i>folder</i>, <i>program</i>.</p> <p>The default value is <i>folder</i>.</p>
Name	<i>String</i>	False		The name of the Token.
Type	<i>String</i>	False		The data type of the Token. The supported

			values are date, number, rich text, score, sfmc campaign and text The allowed values are <i>date, number, rich text, score, sfmc campaign, text.</i>
Value	<i>String</i>	False	The value of the Token.
ComputedURL	<i>String</i>	False	The Computed URL of the Token.

Views

Views are composed of columns and pseudo columns. Views are similar to tables in the way that data is represented; however, views do not support updates. Entities that are represented as views are typically read-only entities. Often, a stored procedure is available to update the data if such functionality is applicable to the data source.

Queries can be executed against a view as if it were a normal table, and the data that comes back is similar in that regard.

Dynamic views, such as queries exposed as views, and views for looking up specific combinations of project_team work items are supported.

Marketo Adapter Views

Name	Description
Activities	Returns a list of activities from after a datetime given by the nextPageToken parameter.
ActivityBulkExports	Returns a list of activity export jobs that were created in the past 7 days.

ActivityTypes	Get activity types for a Marketo Organization
ActivityTypesAttributes	Get activity types attributes for a Marketo Organization
Campaigns	Query Campaigns for a Marketo organization.
ChannelProgressionStatuses	Query ProgressionStatuses of Channels for a Marketo organization.
Channels	Query Channels for a Marketo organization.
DailyErrorStatistics	Gets a list of users and the count of each error type they have encountered in the current day
DailyUsageStatistics	Gets a list of users and the number of calls they have consumed in the current day
EmailCCFields	Query Emails CC Fields for a Marketo organization.
Files	Query Files for a Marketo organization.
LandingPageContentSection	Get section of a landing page content for a Marketo organization.
LandingPageTemplateContent	Query the LandingPageTemplateContent for a Marketo organization.
LeadBulkExports	Returns a list of lead export jobs that were created in the past 7 days.
LeadChanges	Returns a list of Data Value Changes and New Lead activities after a given datetime.
LeadChangesAttributes	Returns a list of Data Value Changes and New Lead activities after a given datetime.
LeadChangesFields	Returns a list of Data Value Changes and New Lead activities after a given datetime.
LeadLists	Query static list membership for one lead.

LeadPartitions	Query Lead Partitions for a Marketo organization.
LeadPrograms	Query program membership for one lead.
Lists	Query Lists for a Marketo organization.
PreviewEmail	View a preview of an email.
ProgramMembersBulkExports	Returns a list of program members export jobs that were created in the past 7 days.
Segmentations	Query segmentations for a Marketo organization.
Segments	Query segments for a Marketo organization.
SmartListRuleFilters	Query SmartLists rule filters
SnippetContent	Query the content of the specific snippet for a Marketo Organization
Tags	Query Tags for a Marketo organization.
ThankYouList	Query Thank you list for the forms
WeeklyErrorStatistics	Gets a list of users and the count of each error type they have encountered in the past 7 days
WeeklyUsageStatistics	Gets a list of users and the number of calls they have consumed in a week

Activities

Returns a list of activities from after a datetime given by the nextPageToken parameter.

Table Specific Information

SELECT

The adapter uses the Marketo API to process WHERE clause conditions built with the following columns and operators. The rest of the filter is executed client-side within the adapter.

- **ActivityDate** supports the '<,>,>=,<='
- **ActivityTypeId** supports the '=,IN'.
- **LeadId** supports the '=,IN'.
- **ListId** supports the '='

For example, the following query is processed server-side:

```
Select * from Activities where activitydate > '2022-09-3' and
activitydate < '2022-09-5'
Select * from Activities where activitytypeid = 11
Select * from Activities where activitytypeid IN (11,12) and leadid IN
(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
Select * from Activities where ActivityTypeId='1' and LeadId='123'
```

Columns

Name	Type	Filterable	Description
Id [KEY]	String	True	Unique id of the activity.
ActivityDate	Datetime	True	Datetime of the activity.
ActivityTypeId	Integer	True	Id of the activity type.
LeadId	Integer	True	Id of the lead associated to the activity.

MarketoGUID	<i>String</i>	Unique id of the activity (128 character string).
PrimaryAttributeValue	<i>String</i>	Value of the primary attribute.
PrimaryAttributeValuelid	<i>Integer</i>	Id of the primary attribute field.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
ListId	<i>String</i>	Id of a static list. If set, will only return activities of members of this static list.

ActivityBulkExports

Returns a list of activity export jobs that were created in the past 7 days.

Columns

Name	Type	Filterable	Description
ExportId [KEY]	<i>String</i>		Unique id of the export job.
ErrorMessage	<i>String</i>		The error message in case of failed

		status.
CreatedAt	<i>Datetime</i>	The date when the export request was created.
FileSize	<i>Long</i>	The size of file in bytes. This column will have a value only when status is 'Completed'.
FinishedAt	<i>Datetime</i>	The finish time of export job. This column will have a value only when status is 'Completed' or 'Failed'.
Format	<i>String</i>	The format of the file.
NumberOfRecords	<i>Integer</i>	The number of records in the export file. This column will have a value only when the status is 'Completed'.
QueuedAt	<i>String</i>	The queue time of the export job. This column will have a value only when 'Queued' status is reached.
StartedAt	<i>String</i>	The start time of the export job. This column will have a value only when 'Processing' status is reached.
Status	<i>String</i>	The status of the export.

ActivityTypes

Get activity types for a Marketo Organization

Table Specific Information

Select

Note: To specify all filterable columns, you must use the '=' operator.

To retrieve a list of activity types for the target instance:

```
SELECT * FROM ActivityTypes
```

Columns

Name	Type	Filterable	Description
Id [KEY]	<i>Integer</i>		The unique, Marketo-assigned identifier of the Activity Types
Name	<i>String</i>		The name of the Activity Types
Description	<i>String</i>		The description of the Activity Types
PrimaryAttributeName	<i>String</i>		The name of the primary attribute
PrimaryAttributeDatatype	<i>String</i>		The data type of the primary attribute

ActivityTypesAttributes

Get activity types attributes for a Marketo Organization

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieves a list of activity types attributes for the target instance

```
SELECT * FROM ActivityTypesAttributes
```

Columns

Name	Type	Filterable	Description
ActivityTypeId	<i>Integer</i>		The unique, Marketo-assigned identifier of the Activity Types.
ActivityTypeName	<i>String</i>		The name of the Activity Types.
AttributeName	<i>String</i>		The name of the primary attribute
AttributeDataType	<i>String</i>		The description of the Activity Types.

Campaigns

Query Campaigns for a Marketo organization.

Columns

Name	Type	Filterable	Description
------	------	------------	-------------

Id [KEY]	<i>Integer</i>	True	The unique, Marketo-assigned identifier of the campaign.
Name	<i>String</i>	True	The name of the campaign.
Description	<i>String</i>		The description of the campaign.
Type	<i>String</i>		The campaign type.
ProgramId	<i>Integer</i>		The Id of the program associated with the campaign.
ProgramName	<i>String</i>		The name of the program associated with the campaign.
WorkspaceName	<i>String</i>		The name of the workspace associated with the campaign.
CreatedAt	<i>Datetime</i>		The date and time the campaign was created.
UpdatedAt	<i>Datetime</i>		The date and time the campaign was last updated.
Active	<i>Boolean</i>		Identifies whether the campaign is active.

ChannelProgressionStatuses

Query ProgressionStatuses of Channels for a Marketo organization.

Columns

Name	Type	Filterable	Description
ChannelName	<i>String</i>	True	The name of the channel.
Name	<i>String</i>		Name of the status.
Description	<i>String</i>		Description of the program status.
Hidden	<i>Boolean</i>		Whether the status has been hidden.
Step	<i>Integer</i>		Step number of the status.
Success	<i>Boolean</i>		Whether this status is a success step for program members.

Channels

Query Channels for a Marketo organization.

Columns

Name	Type	Filterable	Description
Id [KEY]	<i>Integer</i>		The unique, Marketo-assigned identifier of the channel.
Name	<i>String</i>	True	The name of the channel.
Description	<i>String</i>		The description of the channel.

ApplicableProgramType	<i>String</i>	The type of program that the channel is used for.
CreatedAt	<i>Datetime</i>	The date and time the channel was created.
UpdatedAt	<i>Datetime</i>	The date and time the channel was last updated.

DailyErrorStatistics

Gets a list of users and the count of each error type they have encountered in the current day

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieves a list of users and the count of each error type they have encountered in the current day

```
SELECT * FROM DailyErrorStatistics
```

Columns

Name	Type	Filterable	Description
Date	<i>Date</i>		The date when the user encountered error
Total	<i>Integer</i>		The total count of the errors

ErrorCode	<i>String</i>	The error code
ErrorCount	<i>Integer</i>	The error count for the particular error code

DailyUsageStatistics

Gets a list of users and the number of calls they have consumed in the current day

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieves a list of users and the number of calls they have consumed in the current day

```
SELECT * FROM DailyUsageStatistics
```

Columns

Name	Type	Filterable	Description
Date	<i>Date</i>		The date when the API Calls made
Total	<i>Integer</i>		The total count of the API Calls
UserId	<i>String</i>		The ID of the user
APICount	<i>Integer</i>		The individual count for the user

EmailCCFields

Query Emails CC Fields for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of email cc fields.

```
SELECT * FROM EmailsCFields
```

Columns

Name	Type	Filterable	Description
Attributeld	String		The attribute identifier
ObjectName	String		Object Name; Lead or Company
DisplayName	String		The display name
ApiName	String		The API name

Files

Query Files for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieves a list of Files for the target instance

```
SELECT * FROM Files
```

Retrieve the File for the given Id.

```
SELECT * FROM Files WHERE Id = '2012'
```

Columns

Name	Type	Filterable	Description
Id [KEY]	<i>Integer</i>	True	Id of the file.
Name	<i>String</i>	True	Name of the file.
CreatedAt	<i>Datetime</i>		Datetime when the file was created
Description	<i>String</i>		Description of the file
FolderId	<i>Integer</i>	True	Id of the folder
FolderName	<i>String</i>		The Name of the folder
FolderType	<i>String</i>	True	The Type of folder

		The allowed values are <i>Folder</i> , <i>Program</i> .
MimeType	<i>String</i>	MIME type of the file
Size	<i>Integer</i>	Size of the file in bytes
UpdatedAt	<i>Datetime</i>	Datetime when the file was most recently updated
Url	<i>String</i>	Publically accessible URL of the file
FileName	<i>String</i>	The filename for the file to insert.
File	<i>String</i>	Multipart file. Content of the file
InsertOnly	<i>Boolean</i>	Whether the calls hould fail if there is already an existing file with the same name

LandingPageContentSection

Get section of a landing page content for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of landing page content section for a given landing Page Id.

```
SELECT * FROM LandingPageContentSection WHERE LandingPageId = 1193
```

Columns

Name	Type	Filterable	Description
Id [KEY]	<i>String</i>		Id of the content section, may be a string or an int.
Content	<i>String</i>		Content of the section. Expected values vary based on type. Image: An image URL. RichText: HTML Content.
ContentType	<i>String</i>		Content of the section. Expected values vary based on type. Image: An image URL. RichText: HTML Content.
ContentUrl	<i>String</i>		Content of the section. Expected values vary based on type. Image: An image URL. RichText: HTML Content.
FollowupType	<i>String</i>		<p>Follow-up behavior of a form. Only available for form-type content sections. Defaults to form defined behavior.</p> <p>The allowed values are <i>url</i>, <i>lp</i>, <i>formDefined</i>.</p>
FollowupValue	<i>String</i>		Where to follow-up on form submission. When <i>followupType</i> is <i>lp</i> , accepts the integer id of a landing page. For <i>url</i> , it accepts a url string.
FormattingOptionsZIndex	<i>Integer</i>		The zindex of the content

FormattingOptionsLeft	<i>String</i>	The left margin of the content
FormattingOptionsTop	<i>String</i>	The top margin of the content
Index	<i>Integer</i>	Index of the content section. Index orients the elements from lowest to highest.
Type	<i>String</i>	Type of content section. The allowed values are <i>Image</i> , <i>SocialButton</i> , <i>Form</i> , <i>DynamicContent</i> , <i>Rectangle</i> , <i>Snippet</i> , <i>RichText</i> , <i>HTML</i> , <i>Video</i> , <i>Poll</i> , <i>ReferralOffer</i> , <i>Sweepstakes</i> .

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
LandingPageId	<i>Integer</i>	Id of the LandingPage.
Status	<i>String</i>	Status filter for draft or approved versions. The allowed values are <i>draft</i> , <i>approved</i> .

LandingPageTemplateContent

Query the LandingPageTemplateContent for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of landing page template content for a given landing Page template Id.

```
SELECT * FROM LandingPageTemplateContent WHERE LandingPageTemplateId = 1
```

Columns

Name	Type	Filterable	Description
Id	<i>Integer</i>	True	Unique integer id of the template.
Content	<i>String</i>		HTML content of the landing page template.
EnableMunchkin	<i>Boolean</i>		Whether to enable munchkin on the derived pages. Defaults to true.
Status	<i>String</i>	True	Status filter for draft or approved versions The allowed values are <i>approved</i> , <i>draft</i> .
TemplateType	<i>String</i>		Type of template to create. Defaults to freeForm. The allowed values are <i>guided</i> , <i>freeForm</i> .

LeadBulkExports

Returns a list of lead export jobs that were created in the past 7 days.

Columns

Name	Type	Filterable	Description
ExportId [KEY]	String		Unique id of the export job.
ErrorMessage	String		The error message in case of failed status.
CreatedAt	Datetime		The date when the export request was created.
FileSize	Integer		The size of file in bytes. This column will have a value only when status is 'Completed'.
FinishedAt	Datetime		The finish time of export job. This column will have a value only when status is 'Completed' or 'Failed'.
Format	String		The format of the file.
NumberOfRecords	Integer		The number of records in the export file. This will have a value only when the status is 'Completed'.
QueuedAt	String		The queue time of the export job. This column will have a value only when 'Queued' status is reached.
StartedAt	String		The start time of the export job. This column will have a value only when 'Processing' status is reached.
Status	String		The status of the export.

LeadChanges

Returns a list of Data Value Changes and New Lead activities after a given datetime.

Columns

Name	Type	Filterable	Description
Id	<i>Integer</i>		Integer id of the activity
LeadId	<i>Integer</i>		Id of the lead associated to the activity
ActivityDate	<i>Datetime</i>		Datetime of the activity.
ActivityTypeId	<i>Integer</i>		Id of the activity type.
CampaignId	<i>Integer</i>		Id of the Campaign.
MarketoGUID	<i>String</i>		Unique id of the activity (128 character string).

LeadChangesAttributes

Returns a list of Data Value Changes and New Lead activities after a given datetime.

Columns

Name	Type	Filterable	Description
------	------	------------	-------------

Id	<i>Integer</i>	Integer id of the activity
AttributeAPIName	<i>String</i>	API Name of the attribute
AttributeName	<i>String</i>	Name of the attribute
AttributeValue	<i>String</i>	Value of the attribute

LeadChangesFields

Returns a list of Data Value Changes and New Lead activities after a given datetime.

Columns

Name	Type	Filterable	Description
Id	<i>Integer</i>		Integer id of the activity
LeadChangeFieldId	<i>Integer</i>		Unique integer id of the change record
LeadChangeFieldName	<i>String</i>		Name of the field which was changed
LeadChangeFieldNewValue	<i>String</i>		New value after the change
LeadChangeFieldOldValue	<i>String</i>		Old value before the change

LeadLists

Query static list membership for one lead.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieves a list of lists for the specific lead id.

```
SELECT * FROM LeadLists WHERE LeadId = 1021579
```

Columns

Name	Type	Filterable	Description
ListId	<i>Integer</i>		The Id of the Program Member
CreatedAt	<i>Datetime</i>		Indicates this program was responsible for creating the lead record
UpdatedAt	<i>Datetime</i>		The likelihood of the attendance at the individual level

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
LeadId	<i>Integer</i>	

LeadPartitions

Query Lead Partitions for a Marketo organization.

Columns

Name	Type	Filterable	Description
Id [KEY]	<i>Integer</i>		The unique, Marketo-assigned identifier of the lead partition.
Name	<i>String</i>	True	The name of the partition.
Description	<i>String</i>		The description of the partition.

LeadPrograms

Query program membership for one lead.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieves a list of programs for the specific lead id.

```
SELECT * FROM LeadPrograms WHERE LeadId = 1021579
```

Columns

Name	Type	Filterable	Description
Id	<i>Integer</i>		Unique integer id of a program record.
ProgressionStatus	<i>String</i>		Program status of the lead in the parent program.
ProgressionStatusType	<i>String</i>		Program status Type of the lead in the parent program.
IsExhausted	<i>Boolean</i>		Whether the lead is currently exhausted in the stream, if applicable.
AcquiredBy	<i>Boolean</i>		Whether the lead was acquired by the parent program.
ReachedSuccess	<i>Boolean</i>		Whether the lead is in a success-status in the parent program.
MembershipDate	<i>Datetime</i>		Date the lead first became a member of the program.
UpdatedAt	<i>Datetime</i>		Datetime when the program was most recently updated.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
LeadId	<i>Integer</i>	The Marketo lead id

Lists

Query Lists for a Marketo organization.

Columns

Name	Type	Filterable	Description
Id [KEY]	<i>Integer</i>	True	The unique, Marketo-assigned identifier of the list.
Name	<i>String</i>	True	The name of the list.
Description	<i>String</i>		The description of the list.
ProgramName	<i>String</i>		The name of the program associated with the list.
WorkspaceName	<i>String</i>		The name of the workspace associated with the list.
CreatedAt	<i>Datetime</i>		The date and time the list was created.

UpdatedAt	<i>Datetime</i>	The date and time the list was last updated.
-----------	-----------------	--

PreviewEmail

View a preview of an email.

Columns

Name	Type	Filterable	Description
Id [KEY]	<i>Integer</i>	True	The id of the email asset you wish to preview.
Status	<i>String</i>	True	Accepts the values 'draft' or 'approved' which will default to the approved version, if approved, draft if unapproved.
Content	<i>String</i>		The content of the email.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
Type	<i>String</i>	Accepts 'Text' or 'HTML' and defaults to HTML.

LeadId	<i>Integer</i>	Accepts the integer id of a lead. When set, previews the email as though it were received by the designated lead
--------	----------------	--

ProgramMembersBulkExports

Returns a list of program members export jobs that were created in the past 7 days.

Columns

Name	Type	Filterable	Description
ExportId [KEY]	<i>String</i>		Unique id of the export job.
ErrorMessage	<i>String</i>		The error message in case of failed status.
CreatedAt	<i>Datetime</i>		The date when the export request was created.
FileSize	<i>Integer</i>		The size of file in bytes. This column will have a value only when status is 'Completed'.
FinishedAt	<i>Datetime</i>		The finish time of export job. This column will have a value only when status is 'Completed' or 'Failed'.
Format	<i>String</i>		The format of the file.
NumberOfRecords	<i>Integer</i>		The number of records in the export file. This column will have a value only when the status is 'Completed'.
QueuedAt	<i>String</i>		The queue time of the export job. This

		column will have a value only when 'Queued' status is reached.
StartedAt	<i>String</i>	The start time of the export job. This column will have a value only when 'Processing' status is reached.
Status	<i>String</i>	The status of the export.

Segmentations

Query segmentations for a Marketo organization.

Columns

Name	Type	Filterable	Description
Id	<i>Integer</i>		Id of the asset.
Name	<i>String</i>		Name of the asset.
CreatedAt	<i>Datetime</i>		Datetime the asset was created.
Description	<i>String</i>		Description of the asset.
FolderId	<i>Integer</i>		Id of the folder.
FolderType	<i>String</i>		Type of folder. The allowed values are <i>Folder</i> , <i>Program</i> .

Status	<i>String</i>	True	Status filter for draft or approved versions. The allowed values are <i>approved</i> , <i>draft</i> .
UpdatedAt	<i>Datetime</i>		Datetime the asset was most recently updated.
Url	<i>String</i>		Url of the asset in the Marketo UI.
Workspace	<i>String</i>		Name of the workspace.

Segments

Query segments for a Marketo organization.

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieves a list segments for the given segmentation id.

```
SELECT * FROM Segments WHERE SegmentationId = 1012
```

Columns

Name	Type	Filterable	Description
Id	<i>Integer</i>		Id of the asset.
Name	<i>String</i>		Name of the asset.

CreatedAt	<i>Datetime</i>		Datetime the asset was created.
Description	<i>String</i>		Description of the asset.
SegmentationId	<i>Integer</i>	True	Id of the Segmentation.
Status	<i>String</i>	True	Status filter for draft or approved versions. The allowed values are <i>approved</i> , <i>draft</i> .
UpdatedAt	<i>Datetime</i>		Datetime the asset was most recently updated.
Url	<i>String</i>		Url of the asset in the Marketo UI.

SmartListRuleFilters

Query SmartLists rule filters

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator. Atleast one of SmartListId, SmartCampaignId or ProgramId is required to get the Rule Filters.

Retrieve rule filters for Smart Lists by SmartListId

```
SELECT * FROM SmartListRuleFilters WHERE SmartListId = 1143
SELECT * FROM SmartListRuleFilters WHERE SmartListId IN (SELECT Id FROM SmartLists)
```

Retrieve rule filters for Smart Lists by SmartCampaignId

```
SELECT * FROM SmartListRuleFilters WHERE SmartCampaignId = 1682
SELECT * FROM SmartListRuleFilters WHERE SmartCampaignId IN (SELECT Id
FROM SmartCampaigns)
```

Retrieve rule filters for Smart Lists by ProgramId

```
SELECT * FROM SmartListRuleFilters WHERE ProgramId = 1089
SELECT * FROM SmartListRuleFilters WHERE ProgramId IN (SELECT Id FROM
SmartCampaigns)
```

Columns

Name	Type	Filterable	Description
Id [KEY]	Integer		The Id of the smart list rule filter.
Name	String		The name of the smart list rule filter.
Operator	String		The operator used in the filter.
RuleType	String		The type of the rule.
RuleTypeId	Integer		The Id of the rule type.
Conditions	String		The Rule filter conditions.
FilterMatchType	String		The rule filter match type
FilterCustomRuleLogic	String		The rule filter custom logic

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
SmartCampaignId	<i>Integer</i>	The Id of the smart campaign
ProgramId	<i>Integer</i>	The Id of the Program
SmartListId	<i>Integer</i>	The Smart List Id

SnippetContent

Query the content of the specific snippet for a Marketo Organization

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieve a list of snippetcontent for a given snippetId.

```
SELECT * FROM SnippetContent WHERE SnippetId = 3
```

Columns

Name	Type	Filterable	Description
Type	<i>String</i>		Type of the content
Content	<i>String</i>		The content of the snippet

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
SnippetId	<i>Integer</i>	The Id of the Snippet

Tags

Query Tags for a Marketo organization.

Columns

Name	Type	Filterable	Description
TagType [KEY]	<i>String</i>	True	The name/type of the tag.
ApplicableProgramTypes	<i>String</i>		The types of program that the tag is used for.

Required	<i>Boolean</i>	The date and time the channel was created.
AllowableValues	<i>String</i>	The date and time the channel was last updated.

ThankYouList

Query Thank you list for the forms

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieves a list of Thank you pages for the forms.

```
SELECT * FROM ThankYouList
```

Columns

Name	Type	Filterable	Description
FormId [KEY]	<i>Integer</i>	True	Id of the asset.
FollowupType	<i>String</i>	True	Name of the asset.
FollowupValue	<i>String</i>		Label text of the button.
Default	<i>Boolean</i>		Location in pixels of the button relative to the left of the form.

WeeklyErrorStatistics

Gets a list of users and the count of each error type they have encountered in the past 7 days

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieves a list of users and the count of each error type they have encountered in the past 7 days

```
SELECT * FROM WeeklyErrorStatistics
```

Columns

Name	Type	Filterable	Description
Date	<i>Date</i>		The date when the user encountered error
Total	<i>Integer</i>		The total count of the errors
ErrorCode	<i>String</i>		The error code
ErrorCount	<i>Integer</i>		The error count for the particular error code

WeeklyUsageStatistics

Gets a list of users and the number of calls they have consumed in a week

Table Specific Information

Select

Note: All filterable columns must be specified using the '=' operator.

Retrieves a list of users and the number of calls they have consumed in the paast 7 days

```
SELECT * FROM WeeklyUsageStatistics
```

Columns

Name	Type	Filterable	Description
Date	<i>Date</i>		The date when the API Calls made
Total	<i>Integer</i>		The total count of the API Calls
UserId	<i>String</i>		The ID of the user
APICount	<i>Integer</i>		The individual count for the user

Stored Procedures

Stored procedures are function-like interfaces that extend the functionality of the adapter beyond simple SELECT/INSERT/UPDATE/DELETE operations with Marketo.

Stored procedures accept a list of parameters, perform their intended function, and then return, if applicable, any relevant response data from Marketo, along with an indication of whether the procedure succeeded or failed.

Marketo Adapter Stored Procedures

Name	Description
AddLandingPageContentSection	Adds a content section to the target landing page.
AddLeadsToList	Adds leads to a list in Marketo.
AssociateLead	Associates a known Marketo lead record to a munchkin cookie and its associated web activity history
CancelExportJob	Cancels an export job.
CreateEmailTemplate	Creates a new email template.
CreateExportJob	Create export job for search criteria defined via filter parameter. Returns the 'JobId' which is passed as a parameter in subsequent calls to Bulk Export Activities. Use EnqueueExportJob to queue the export job for processing. Use GetExportJobStatus to retrieve status of export job.
CreateFile	Creates a new file from the included payload.
DeleteUpdatePageContentSection	Deletes the section of the landing page content in Marketo.
EnqueueExportJob	Places an export job in queue and starts the job when computing resources become available.
GetExportJobFile	Returns the file content of an export job. The export job must be in 'Completed' state. Use GetExportJobStatus to get the status of an export job.
GetExportJobStatus	Returns the status of an export job. Job status is available for 30 days after Completed or Failed status was reached.

GetImportCustomObjectsFailures	Downloads and saves the import custom objects failures file from Marketo.
GetImportCustomObjectsStatus	Gets the status of an import custom objects from file operation for Marketo.
GetImportCustomObjectsWarnings	Downloads and saves the import custom objects warnings file from Marketo.
GetImportLeadsFailures	Downloads and saves the import leads failures file from Marketo.
GetImportLeadsStatus	Gets the status of an import leads from file operation for Marketo.
GetImportLeadsWarnings	Downloads and saves the import leads warnings file from Marketo.
GetImportProgramMembersFailures	Downloads and saves the import program members failures file from Marketo.
GetImportProgramMembersStatus	Gets the status of an import program members from file operation for Marketo.
GetImportProgramMembersWarnings	Downloads and saves the import program members warnings file from Marketo.
GetOAuthAccessToken	Gets an authentication token from Marketo.
ImportCustomObjectsFromFile	Imports custom objects from a file into Marketo.
ImportLeadsFromFile	Imports leads from a file into Marketo.
ImportProgramMembersFromFile	Imports program members from a file into Marketo.
ListMembersOfList	Determines whether leads are contained within a list in Marketo.
MergeLeads	Merges two or more known lead records into a

	single lead record.
RefreshOAuthAccessToken	Gets an authentication token from Marketo.
RemoveLeadsFromList	Removes leads from a list in Marketo.
ScheduleCampaign	Remotely schedules a batch campaign to run at a given time.
UpdateEmailContent	Updates the content of an email
UpdateEmailFullContent	Updates the content of an email
UpdateEmailTemplateContent	Updates the content of the given email template.
UpdateFile	Replaces the current content of the file with the included payload
UpdateLandingPageContentSection	Add or update the section of the landing page content in Marketo.
UpdateLandingPageTemplateContent	Updates the content for the target landing page template.
UpdateLeadProgramStatus	Changes the program status of a list of leads in a target program. Only existing members of the program may have their status changed with this API.

AddLandingPageContentSection

Adds a content section to the target landing page.

EXECUTE Example:

```
EXECUTE AddLandingPageContentSection LandingPageId = '1003', ContentId = '1' Type = 'HTML'
```

Input

Name	Type	Required	Description
LandingPageId	<i>String</i>	<i>True</i>	Id of the landing page.
ContentId	<i>String</i>	<i>True</i>	Id of the content section. Also the HTML id of the section.
Type	<i>String</i>	<i>True</i>	Type of content section. The allowed values are <i>Image, Form, Rectangle, Snippet, RichText, HTML</i> .
BackgroundColor	<i>String</i>	<i>False</i>	background-color property of the HTML section.
BorderColor	<i>String</i>	<i>False</i>	Border-color property of the HTML section.
BorderStyle	<i>String</i>	<i>False</i>	Border-style property of the HTML section.
BorderWidth	<i>String</i>	<i>False</i>	Border-width property of the HTML section.
Height	<i>String</i>	<i>False</i>	Height property of the HTML section.
HideDesktop	<i>String</i>	<i>False</i>	Hide the section when displayed on a desktop browser.
HideMobile	<i>String</i>	<i>False</i>	Hide the section when displayed on a mobile browser. Default false.
ImageOpenNewWindow	<i>String</i>	<i>False</i>	Image open new window.
Index	<i>String</i>	<i>False</i>	Index of the content section. Determines the order of the section in

			the landing page.
Left	<i>String</i>	<i>False</i>	Left property of the HTML section.
LinkUrl	<i>String</i>	<i>False</i>	URL parameter of a link type section.
Opacity	<i>String</i>	<i>False</i>	Opacity property of the HTML section.
Top	<i>String</i>	<i>False</i>	Top property of the HTML section.
Value	<i>String</i>	<i>False</i>	Type of content section.
Width	<i>String</i>	<i>False</i>	Width property of the HTML section.
ZIndex	<i>String</i>	<i>False</i>	Z-index property of the HTML section.

Result Set Columns

Name	Type	Description
Id	<i>String</i>	Id of the content section.

AddLeadsToList

Adds leads to a list in Marketo.

EXECUTE Example:

```
EXECUTE AddLeadsToList ListId='1001', LeadId='10,13,20'
```

Input

Name	Type	Required	Description
ListId	<i>String</i>	<i>True</i>	The Id of the list to add leads to.
LeadId	<i>String</i>	<i>True</i>	A comma-separated list of lead Ids to be added to the list specified by ListId.

Result Set Columns

Name	Type	Description
LeadId	<i>String</i>	The LeadId that was attempted to be added to the list.
Status	<i>String</i>	The status of the lead being added to the list. Applicable values: added and skipped.
ReasonCode	<i>String</i>	The reason code pertaining to why a lead was skipped (status will return skipped).
ReasonMessage	<i>String</i>	The reason message pertaining to why a lead was skipped (status will return skipped).

AssociateLead

Associates a known Marketo lead record to a munchkin cookie and its associated web activity history

EXECUTE Example:

```
EXECUTE AssociateLead LeadId = 1021579, Cookie = 'id:119-IEY-862%26token:_mch-amazonaws.com-1541561364144-71837'
```

Input

Name	Type	Required	Description
LeadId	<i>Integer</i>	<i>True</i>	Id of the Lead to associate
Cookie	<i>String</i>	<i>True</i>	The cookie value to associate

Result Set Columns

Name	Type	Description
Status	<i>String</i>	The status of the lead being associated

CancelExportJob

Cancels an export job.

Input

Name	Type	Required	Description
JobId	<i>String</i>	<i>True</i>	The id of the export job.

Type	<i>String</i>	<i>True</i>	<p>The object type of the BulkExtract. Available values are : Activities, Leads, ProgramMembers</p> <p>The allowed values are <i>Activities, Leads, ProgramMembers</i>.</p>
------	---------------	-------------	---

Result Set Columns

Name	Type	Description
JobId	<i>String</i>	The Id of the export job.
JobStatus	<i>String</i>	The status of the export process.
Format	<i>String</i>	The format of the export job.
CreatedAt	<i>String</i>	The date when the export job was created.
StartedAt	<i>String</i>	The date when the export job was started.
QueuedAt	<i>String</i>	The date when the export job was queued.

CreateEmailTemplate

Creates a new email template.

EXECUTE Example:

```
EXECUTE CreateEmailTemplate Name = 'Marketo.html', File =
'C:\\Users\\Dell\\Desktop\\Test_Logs\\Marketo.html', Description = 'Test
Create Email Template', FolderId = 45, FolderType = 'Folder'
```

Input

Name	Type	Required	Description
File	<i>String</i>	<i>False</i>	Multipart File. HTML content for template.
FolderId	<i>Integer</i>	<i>False</i>	The Id of the folder
FolderType	<i>String</i>	<i>False</i>	The type of the folder
Name	<i>String</i>	<i>False</i>	The name of the Email Template. Must be unique under the parent folder.
Description	<i>String</i>	<i>False</i>	The description of the email template

Result Set Columns

Name	Type	Description
Id	<i>String</i>	The Id of the created email template

CreateExportJob

Create export job for search criteria defined via filter parameter. Returns the 'JobId' which is passed as a parameter in subsequent calls to Bulk Export Activities. Use `EnqueueExportJob` to queue the export job for processing. Use `GetExportJobStatus` to retrieve status of export job.

Input

Name	Type	Required	Description
Fields	<i>String</i>	<i>False</i>	A comma separated list of Marketo fields.
Type	<i>String</i>	<i>True</i>	<p>The object type of the BulkExtract. Available values are : Activities, Leads, ProgramMembers.</p> <p>The allowed values are <i>Activities, Leads, ProgramMembers</i>.</p>
Format	<i>String</i>	<i>False</i>	<p>Format of file as given in the request. Available values are: CSV, TSV, SSV</p> <p>The allowed values are <i>CSV, TSV, SSV</i>.</p> <p>The default value is <i>CSV</i>.</p>
*ColumnName	<i>String</i>	<i>False</i>	<p>The header column name for the field. For example : If you want to set the header name of two fields called 'FirstName' and 'LastName', you just need to set the value of 'FirstNameColumnName' and 'LastNameColumnName'.</p>
CreatedAtStartAt	<i>String</i>	<i>False</i>	The start date of the Marketo Object created date
CreatedAtEndAt	<i>String</i>	<i>False</i>	The end date of the Marketo Object created date

UpdatedAtStartAt	<i>String</i>	<i>False</i>	The start date of the Marketo Object updated date
UpdatedAtEndAt	<i>String</i>	<i>False</i>	The end date of the Marketo Object updated date
ActivityTypeIds	<i>String</i>	<i>False</i>	List of activity type ids to filter on. Available only when the Type is set to 'Activities'.
StaticListName	<i>String</i>	<i>False</i>	The name of a static list you want to use as a filter. Available only when the Type is set to 'Leads'
StaticListId	<i>String</i>	<i>False</i>	The id of a static list you want to use as a filter. Available only when the Type is set to 'Leads'
SmartListName	<i>String</i>	<i>False</i>	The name of a smart list you want to use as a filter. Available only when the Type is set to 'Leads'
SmartListId	<i>String</i>	<i>False</i>	The id of a smart list you want to use as a filter. Available only when the Type is set to 'Leads'
ProgramId	<i>String</i>	<i>False</i>	The id of the program you want to use as a filter. Available only when the Type is set to 'ProgramMembers'

Result Set Columns

Name	Type	Description
JobId	<i>String</i>	The Id of the export job.

Format	<i>String</i>	The format of the export job.
CreatedAt	<i>String</i>	The date when the export job was created.
JobStatus	<i>String</i>	The status of the export process.

CreateFile

Creates a new file from the included payload.

EXECUTE Example:

```
EXECUTE CreateFile File = 'C:\\Users\\Dell\\Desktop\\Test_
Logs\\Marketo.html', Description = 'Test Create File', Name =
'john@abc.com', FolderId = 1032, FolderType = 'folder'
```

Input

Name	Type	Required	Accepts Input Streams	Description
File	<i>String</i>	<i>True</i>	<i>False</i>	Multipart File. Content of the file.
FolderId	<i>Integer</i>	<i>True</i>	<i>False</i>	The Id of the folder
FolderType	<i>String</i>	<i>True</i>	<i>False</i>	The type of the folder
InsertOnly	<i>Boolean</i>	<i>False</i>	<i>False</i>	Indicates whether the call should fail if there is already an existing file with the same name

Name	<i>String</i>	<i>True</i>	<i>False</i>	The name of the file
Description	<i>String</i>	<i>False</i>	<i>False</i>	The description of the asset
Content	<i>String</i>	<i>False</i>	<i>True</i>	The content as InputStream to be uploading when File is not specified.

Result Set Columns

Name	Type	Description
Id	<i>String</i>	The Id of the created file

DeleteUpdatePageContentSection

Deletes the section of the landing page content in Marketo.

Input

Name	Type	Required	Description
LandingPageId	<i>String</i>	<i>True</i>	Id of the landing page.
ContentId	<i>String</i>	<i>True</i>	Id of the content section. Also the HTML id of the section.

Result Set Columns

Name	Type	Description
Id	String	Id of the landing page.

EnqueueExportJob

Places an export job in queue and starts the job when computing resources become available.

Input

Name	Type	Required	Description
JobId	String	True	The id of the export job.
Type	String	True	The object type of the BulkExtract. Available values are : Activities, Leads, ProgramMembers. The allowed values are <i>Activities</i> , <i>Leads</i> , <i>ProgramMembers</i> .

Result Set Columns

Name	Type	Description
JobId	String	The Id of the export job.
JobStatus	String	The status of the export process.

Format	<i>String</i>	The format of the export job.
CreatedAt	<i>String</i>	The date when the export job was created.
QueuedAt	<i>String</i>	The date when the export job was queued.

GetExportJobFile

Returns the file content of an export job. The export job must be in 'Completed' state. Use GetExportJobStatus to get the status of an export job.

Input

Name	Type	Required	Accepts Output Streams	Description
JobId	<i>String</i>	<i>True</i>	<i>False</i>	The id of the export job.
Type	<i>String</i>	<i>True</i>	<i>False</i>	The type of the BulkExtract. Available values are : Activities, Leads, ProgramMembers The allowed values are <i>Activities, Leads, ProgramMembers</i> .
File	<i>String</i>	<i>False</i>	<i>False</i>	The file path where you want to store the file.
Encoding	<i>String</i>	<i>False</i>	<i>False</i>	The FileData input encoding type. The allowed values are <i>NONE</i> ,

				<i>BASE64.</i>
				The default value is <i>BASE64</i> .
FileStream	<i>String</i>	<i>False</i>	<i>True</i>	An instance of an output stream where file data is written to. Only used if FileLocation is not provided.

Result Set Columns

Name	Type	Description
ResponseData	<i>String</i>	The content of the file. Returns data only if File and FileStream is not specified.

GetExportJobStatus

Returns the status of an export job. Job status is available for 30 days after Completed or Failed status was reached.

Input

Name	Type	Required	Description
JobId	<i>String</i>	<i>True</i>	The id of the export job.
Type	<i>String</i>	<i>True</i>	The object type of the BulkExtract. Available values are : Activities, Leads, ProgramMembers. The allowed values are <i>Activities</i> , <i>Leads</i> , <i>ProgramMembers</i> .

Result Set Columns

Name	Type	Description
JobId	String	The Id of the export job.
Format	String	The format of the export job.
CreatedAt	String	The date when the export job was created.
JobStatus	String	The status of the export process. Applicable values: Created, Queued, Processing, Cancelled, Completed, Failed.
QueuedAt	String	The date when the export job was queued.
StartedAt	String	The date when the export job was started.
FinishedAt	String	The date when the export job was finished.
NumberOfRecords	String	The number of records contained within the generated file.
FileSize	String	The size of the generated file.

GetImportCustomObjectsFailures

Downloads and saves the import custom objects failures file from Marketo.

Input

Name	Type	Required	Accepts Output Streams	Description
BatchId	<i>String</i>	<i>True</i>	<i>False</i>	The batch Id of the import custom objects operation to retrieve the failures file for.
TableName	<i>String</i>	<i>True</i>	<i>False</i>	The name of the custom object table. Ex: CustomObject_test
FilePath	<i>String</i>	<i>False</i>	<i>False</i>	The file and path where the downloaded failures file will be saved, i.e. 'C:\temp\myfile.csv'. Note the file contains CSV (comma separated value) data.
Encoding	<i>String</i>	<i>False</i>	<i>False</i>	The FileData input encoding type. The allowed values are <i>NONE</i> , <i>BASE64</i> . The default value is <i>BASE64</i> .
FileStream	<i>String</i>	<i>False</i>	<i>True</i>	An instance of an output stream where file data is written to. Only used if FilePath is not provided.

Result Set Columns

Name	Type	Description
FileData	<i>String</i>	The file data. This is only populated if the 'FilePath' and 'FileStream' input was not set to write the data to a file. Note the file contains CSV (comma separated value) data.

GetImportCustomObjectsStatus

Gets the status of an import custom objects from file operation for Marketo.

Input

Name	Type	Required	Description
BatchId	<i>String</i>	<i>True</i>	The batch Id returned from an import custom objects from file operation.
TableName	<i>String</i>	<i>True</i>	The name of the custom object table. Ex: CustomObject_test

Result Set Columns

Name	Type	Description
Status	<i>String</i>	The status of the import process.
NumOfObjectsProcessed	<i>String</i>	The number of custom objects processed.
NumOfRowsFailed	<i>String</i>	The number of rows that failed.
NumOfRowsWithWarning	<i>String</i>	The number of rows that contain a warning.
ImportTime	<i>String</i>	The elapsed time of the import process.
Message	<i>String</i>	The message about the import operation.

GetImportCustomObjectsWarnings

Downloads and saves the import custom objects warnings file from Marketo.

Input

Name	Type	Required	Accepts Output Streams	Description
BatchId	<i>String</i>	<i>True</i>	<i>False</i>	The batch Id of the import custom objects operation to retrieve the warnings file for.
TableName	<i>String</i>	<i>True</i>	<i>False</i>	The name of the custom object table. Ex: CustomObject_test
FilePath	<i>String</i>	<i>False</i>	<i>False</i>	The file and path where the downloaded warnings file will be saved, i.e. 'C:\temp\myfile.csv'. Note the file contains CSV (comma separated value) data.
Encoding	<i>String</i>	<i>False</i>	<i>False</i>	The FileData input encoding type. The allowed values are <i>NONE</i> , <i>BASE64</i> . The default value is <i>BASE64</i> .
FileStream	<i>String</i>	<i>False</i>	<i>True</i>	An instance of an output stream where file data is written to. Only used if FilePath is not provided.

Result Set Columns

Name	Type	Description
FileData	<i>String</i>	The file data. This is only populated if the 'FilePath' and 'FileStream' input was not set to write the data to a file. Note the file contains CSV (comma separated value) data.

GetImportLeadsFailures

Downloads and saves the import leads failures file from Marketo.

Input

Name	Type	Required	Accepts Output Streams	Description
BatchId	<i>String</i>	<i>True</i>	<i>False</i>	The batch Id of the import leads operation to retrieve the failures file for.
FilePath	<i>String</i>	<i>False</i>	<i>False</i>	The file and path where the downloaded failures file will be saved, i.e. 'C:\temp\myfile.csv'. Note the file contains CSV (comma separated value) data.
Encoding	<i>String</i>	<i>False</i>	<i>False</i>	The FileData input encoding type. The allowed values are <i>NONE</i> , <i>BASE64</i> . The default value is <i>BASE64</i> .
FileStream	<i>String</i>	<i>False</i>	<i>True</i>	An instance of an output stream where file data is written to. Only used if FilePath is not provided.

Result Set Columns

Name	Type	Description
FileData	<i>String</i>	The file data. This is only populated if the 'FilePath' and 'FileStream' input was not set to write the data to a file. Note the file contains CSV (comma separated value) data.

GetImportLeadsStatus

Gets the status of an import leads from file operation for Marketo.

Input

Name	Type	Required	Description
BatchId	<i>String</i>	<i>True</i>	The batch Id returned from an import leads from file operation.

Result Set Columns

Name	Type	Description
BatchId	<i>String</i>	The Id of the import leads batch.
Status	<i>String</i>	The status of the import process.
NumOfLeadsProcessed	<i>String</i>	The number of leads processed.

NumOfRowsFailed	<i>String</i>	The number of rows that failed.
NumOfRowsWithWarning	<i>String</i>	The number of rows that contain a warning.
Message	<i>String</i>	The message about the import operation.

GetImportLeadsWarnings

Downloads and saves the import leads warnings file from Marketo.

Input

Name	Type	Required	Accepts Output Streams	Description
BatchId	<i>String</i>	<i>True</i>	<i>False</i>	The batch Id of the import leads operation to retrieve the warnings file for.
FilePath	<i>String</i>	<i>False</i>	<i>False</i>	The file and path where the downloaded warnings file will be saved, i.e. 'C:\temp\myfile.csv'. Note the file contains CSV (comma separated value) data.
Encoding	<i>String</i>	<i>False</i>	<i>False</i>	The FileData input encoding type. The allowed values are <i>NONE</i> , <i>BASE64</i> . The default value is <i>BASE64</i> .
FileStream	<i>String</i>	<i>False</i>	<i>True</i>	An instance of an output stream where file data is written to. Only used if FilePath is not provided.

Result Set Columns

Name	Type	Description
FileData	<i>String</i>	The file data. This is only populated if the 'FilePath' and 'FileStream' input was not set to write the data to a file. Note the file contains CSV (comma separated value) data.

GetImportProgramMembersFailures

Downloads and saves the import program members failures file from Marketo.

Input

Name	Type	Required	Accepts Output Streams	Description
BatchId	<i>String</i>	<i>True</i>	<i>False</i>	The batch Id of the import program members operation to retrieve the failures file for.
FilePath	<i>String</i>	<i>False</i>	<i>False</i>	The file and path where the downloaded failures file will be saved, i.e. 'C:\temp\myfile.csv'. Note the file contains CSV (comma separated value) data.
Encoding	<i>String</i>	<i>False</i>	<i>False</i>	The FileData input encoding type. The allowed values are <i>NONE</i> , <i>BASE64</i> . The default value is <i>BASE64</i> .

FileStream	<i>String</i>	<i>False</i>	<i>True</i>	An instance of an output stream where file data is written to. Only used if FilePath is not provided.
------------	---------------	--------------	-------------	---

Result Set Columns

Name	Type	Description
FileData	<i>String</i>	The file data. This is only populated if the 'FilePath' and 'FileStream' input was not set to write the data to a file. Note the file contains CSV (comma separated value) data.

GetImportProgramMembersStatus

Gets the status of an import program members from file operation for Marketo.

Input

Name	Type	Required	Description
BatchId	<i>String</i>	<i>True</i>	The batch Id returned from an import program members from file operation.

Result Set Columns

Name	Type	Description
BatchId	<i>String</i>	The Id of the import leads batch.

Status	<i>String</i>	The status of the import process.
NumOfLeadsProcessed	<i>String</i>	The number of leads processed.
NumOfRowsFailed	<i>String</i>	The number of rows that failed.
NumOfRowsWithWarning	<i>String</i>	The number of rows that contain a warning.
Message	<i>String</i>	The message about the import operation.

GetImportProgramMembersWarnings

Downloads and saves the import program members warnings file from Marketo.

Input

Name	Type	Required	Accepts Output Streams	Description
BatchId	<i>String</i>	<i>True</i>	<i>False</i>	The batch Id of the import program members operation to retrieve the warnings file for.
FilePath	<i>String</i>	<i>False</i>	<i>False</i>	The file and path where the downloaded failures file will be saved, i.e. 'C:\temp\myfile.csv'. Note the file contains CSV (comma separated value) data.
Encoding	<i>String</i>	<i>False</i>	<i>False</i>	The FileData input encoding type. The allowed values are <i>NONE</i> ,

				<i>BASE64.</i>
				The default value is <i>BASE64</i> .
FileStream	<i>String</i>	<i>False</i>	<i>True</i>	An instance of an output stream where file data is written to. Only used if FilePath is not provided.

Result Set Columns

Name	Type	Description
FileData	<i>String</i>	The file data. This is only populated if the 'FilePath' and 'FileStream' input was not set to write the data to a file. Note the file contains CSV (comma separated value) data.

GetOAuthAccessToken

Gets an authentication token from Marketo.

Result Set Columns

Name	Type	Description
OAuthAccessToken	<i>String</i>	The access token used for communication with Marketo.
ExpiresIn	<i>String</i>	The remaining lifetime on the access token.

ImportCustomObjectsFromFile

Imports custom objects from a file into Marketo.

Input

Name	Type	Required	Accepts Input Streams	Description
File	<i>String</i>	<i>False</i>	<i>False</i>	The path to the file containing custom objects to be imported into Marketo, i.e. 'C:\temp\custom_object_import.csv'.
TableName	<i>String</i>	<i>True</i>	<i>False</i>	The name of the custom object table. Ex: CustomObject_test
Format	<i>String</i>	<i>False</i>	<i>False</i>	The format of the file that will be used to import custom objects into Marketo. The allowed values are <i>csv</i> , <i>tsv</i> , <i>ssv</i> .
Content	<i>String</i>	<i>False</i>	<i>True</i>	The content as InputStream to be uploading when File is not specified.

Result Set Columns

Name	Type	Description
BatchId	<i>String</i>	The Id of the import custom objects batch.
Status	<i>String</i>	The status of the import process.

ImportLeadsFromFile

Imports leads from a file into Marketo.

Stored Procedure Specific Information

The 'File' parameter must be a path to a file formatted as specified in the 'Format' parameter.

The first row in the file will be the columns, which are the Marketo API field names and map directly to the column names for the Leads table. The remaining rows will be the Leads data you want to import into Marketo.

The maximum file size is 10 MB and all the processing is performed on the Marketo back-end which creates faster performance and less work for the local machine.

GetImportLeadsFailures, GetImportLeadsStatus, and GetImportLeadsWarnings are related stored procedures that provide information on the import process.

Input

Name	Type	Required	Accepts Input Streams	Description
File	String	False	False	The path to the file containing leads to be imported into Marketo, i.e. 'C:\temp\myleads.csv'.
Format	String	True	False	The format of the file that will be used to import leads into Marketo. Applicable values are: 'csv' (comma separated value), 'tsv' (tab separated value), and 'ssv' (semi colon separated value).
LookupField	String	False	False	Used to specify the field used to

				find duplicate leads. Available values are: id, cookie, email, twitterId, facebookId, linkedInId, sfdcAccountId, sfdcContactId, sfdcLeadId, sfdcLeadOwnerId.
ListId	<i>String</i>	<i>False</i>	<i>False</i>	The Id of the list to import the leads into.
PartitionName	<i>String</i>	<i>False</i>	<i>False</i>	The name of the partition to import the leads into.
Content	<i>String</i>	<i>False</i>	<i>True</i>	The content as InputStream to be uploading when File is not specified.

Result Set Columns

Name	Type	Description
BatchId	<i>String</i>	The Id of the import leads batch.
Status	<i>String</i>	The status of the import process.

ImportProgramMembersFromFile

Imports program members from a file into Marketo.

Stored Procedure Specific Information

The 'File' parameter must be a path to a file formatted as specified in the 'Format' parameter.

The first row in the file will be the columns, which are the Marketo API field names and map directly to the column names for the Leads table. The remaining rows will be the Program Members data you want to import into Marketo.

The maximum file size is 10 MB and all the processing is performed on the Marketo back-end which creates faster performance and less work for the local machine. `GetImportProgramMembersFailures`, `GetImportProgramMembersStatus`, and `GetImportProgramMembersWarnings` are related stored procedures that provide information on the import process.

```
Execute ImportProgramMembersFromFile ProgramId='1089',
ProgramMemberStatus='Member',
File='C:\Dev\tests\v20\ProviderMarketo\importprogrammembers.csv'
```

Input

Name	Type	Required	Accepts Input Streams	Description
File	<i>String</i>	<i>False</i>	<i>False</i>	The path to the file containing program members to be imported into Marketo, i.e. 'C:\temp\importprogrammembers.csv'.
ProgramId	<i>String</i>	<i>True</i>	<i>False</i>	The Id of the program
ProgramMemberStatus	<i>String</i>	<i>True</i>	<i>False</i>	Program member status for members being added.
Format	<i>String</i>	<i>False</i>	<i>False</i>	The format of the file that will be used to import program members into Marketo. The allowed values are <i>csv</i> , <i>tsv</i> , <i>ssv</i> .

Content	<i>String</i>	<i>False</i>	<i>True</i>	The content as InputStream to be uploading when File is not specified.
---------	---------------	--------------	-------------	--

Result Set Columns

Name	Type	Description
BatchId	<i>String</i>	The Id of the import program members batch.
ImportId	<i>String</i>	The Id of the import job.
Status	<i>String</i>	The status of the import process.

ListMembersOfList

Determines whether leads are contained within a list in Marketo.

EXECUTE Example:

```
EXECUTE ListMembersOfList ListId='1001', LeadId='10,13,20'
```

Input

Name	Type	Required	Description
ListId	<i>String</i>	<i>True</i>	The Id of the list to check for leads.

LeadId	<i>String</i>	<i>True</i>	A comma-separated list of lead Ids to be checked if they are in the list specified by ListId.
--------	---------------	-------------	---

Result Set Columns

Name	Type	Description
LeadId	<i>String</i>	The LeadId that was checked if it is in the list.
Status	<i>String</i>	The status of the lead and whether it is in the list. Applicable values: memberof, notmemberof, and skipped.
ReasonCode	<i>String</i>	The reason code pertaining to why a lead was skipped (status will return skipped).
ReasonMessage	<i>String</i>	The reason message pertaining to why a lead was skipped (status will return skipped).

MergeLeads

Merges two or more known lead records into a single lead record.

EXECUTE Example:

```
EXECUTE MergeLeads WinningLeadId = 1021579, LoosingLeadId = 1021580
EXECUTE MergeLeads WinningLeadId = 1021579, LoosingLeadId =
1021632,1021630
```

Input

Name	Type	Required	Description
------	------	----------	-------------

WinningLeadId	<i>String</i>	<i>True</i>	The id of the winning lead record
LoosingLeadId	<i>String</i>	<i>True</i>	A comma separated list of loosing lead ids

Result Set Columns

Name	Type	Description
Status	<i>String</i>	The status of the merge lead

RefreshOAuthAccessToken

Gets an authentication token from Marketo.

Result Set Columns

Name	Type	Description
OAuthAccessToken	<i>String</i>	The access token used for communication with Marketo.
ExpiresIn	<i>String</i>	The remaining lifetime on the access token.

RemoveLeadsFromList

Removes leads from a list in Marketo.

EXECUTE Example:

```
EXECUTE RemoveLeadsFromList ListId='1001', LeadId='10,13,20'
```

Input

Name	Type	Required	Description
ListId	<i>String</i>	<i>True</i>	The Id of the list to remove leads from.
LeadId	<i>String</i>	<i>True</i>	A comma-separated list of lead Ids to be removed from the list specified by ListId.

Result Set Columns

Name	Type	Description
LeadId	<i>String</i>	The LeadId that was attempted to be removed from the list.
Status	<i>String</i>	The status of the lead being removed from the list. Applicable values: removed and skipped.
ReasonCode	<i>String</i>	The reason code pertaining to why a lead was skipped (status will return skipped).
ReasonMessage	<i>String</i>	The reason message pertaining to why a lead was skipped (status will return skipped).

ScheduleCampaign

Remotely schedules a batch campaign to run at a given time.

EXECUTE Example:

```

INSERT into TokenAggregate#Temp (TokenName, TokenValue) VALUES ('test',
'testvalue_cdata')
INSERT into TokenAggregate#Temp (TokenName, TokenValue) VALUES ('test1',
'testvalue1')
EXECUTE ScheduleCampaign TokenAggregate = 'TokenAggregate#Temp',
CampaignId = 1019, CloneToProgramName = 'Program 1'

```

Input

Name	Type	Required	Description
CampaignId	<i>String</i>	<i>True</i>	Id of the batch campaign to schedule.
CloneToProgramName	<i>String</i>	<i>False</i>	Name of the resulting program.
RunAt	<i>String</i>	<i>False</i>	Datetime to run the campaign at. If unset, the campaign will be run five minutes after the call is made.
TokenName	<i>String</i>	<i>False</i>	Name of the token. Should be formatted as '{{my.name}}'
TokenValue	<i>String</i>	<i>False</i>	Value of the token
TokenAggregate	<i>String</i>	<i>False</i>	List of my tokens to replace during the run of the target campaign. The tokens must be available in a parent program or folder to be replaced during the run

Result Set Columns

Name	Type	Description
------	------	-------------

Success	String	Whether the request succeeded ,
---------	--------	---------------------------------

UpdateEmailContent

Updates the content of an email

EXECUTE Example:

```
EXECUTE UpdateEmailContent EmailId = '1187', FromName = 'John',
FromEmail = 'john@abc.com', Subject = 'test', ReplyTo = 'steve@abc.com'
```

Input

Name	Type	Required	Description
EmailId	String	True	The Id of the email.
FromEmail	String	False	From-address of the Email
FromName	String	False	From-name of the Email
ReplyTo	String	False	Reply-To address of the Email
Subject	String	False	Subject Line of the Email

Result Set Columns

Name	Type	Description
Id	<i>String</i>	The Id of the updated email content

UpdateEmailFullContent

Updates the content of an email

EXECUTE Example:

```
EXECUTE UpdateEmailFullContent EmailId = '1187', File =
'C:\\Users\\Dell\\Desktop\\Test_Logs\\Marketo.html'
```

Input

Name	Type	Required	Description
EmailId	<i>String</i>	<i>True</i>	The Id of the email.
File	<i>String</i>	<i>False</i>	From-address of the Email

Result Set Columns

Name	Type	Description
Id	<i>String</i>	The status of the lead being added to the list. Applicable values: added and skipped.

UpdateEmailTemplateContent

Updates the content of the given email template.

EXECUTE Example:

```
EXECUTE UpdateEmailTemplateContent File =
'C:\\Users\\Dell\\Desktop\\Test_Logs\\Marketo.html', EmailTemplateId =
'1'
```

Input

Name	Type	Required	Description
EmailTemplateId	<i>Integer</i>	<i>True</i>	The Id of the email template.
File	<i>String</i>	<i>True</i>	Content for the email template. Multipart file.

Result Set Columns

Name	Type	Description
Id	<i>String</i>	The status of the lead being added to the list. Applicable values: added and skipped.

UpdateFile

Replaces the current content of the file with the included payload

EXECUTE Example:


```
EXECUTE UpdateFile File = 'C:\\Users\\Dell\\Desktop\\Test_
Logs\\Marketo.html', FileId = '43996'
```

Input

Name	Type	Required	Description
File	<i>String</i>	<i>False</i>	Multipart File. Content of the file.
FileId	<i>Integer</i>	<i>False</i>	The Id of the folder

Result Set Columns

Name	Type	Description
Id	<i>String</i>	The Id of the created file

UpdateLandingPageContentSection

Add or update the section of the landing page content in Marketo.

EXECUTE Example:

```
EXECUTE UpdateLandingPageContentSection LandingPageId = '1003',
ContentId = '1009', Type = 'HTML', Value = '<html><body><h3>Test
Update</h3></body></html>'
```

Input

Name	Type	Required	Description
LandingPageId	String	True	Id of the landing page.
ContentId	String	True	Id of the content section. Also the HTML id of the section.
Type	String	True	Type of content section. The allowed values are <i>Image, Form, Rectangle, Snippet, RichText, HTML</i> .
BackgroundColor	String	False	background-color property of the HTML section.
BorderColor	String	False	Border-color property of the HTML section.
BorderStyle	String	False	Border-style property of the HTML section.
BorderWidth	String	False	Border-width property of the HTML section.
Height	String	False	Height property of the HTML section.
HideDesktop	String	False	Hide the section when displayed on a desktop browser.
HideMobile	String	False	Hide the section when displayed on a mobile browser. Default false.
ImageOpenNewWindow	String	False	Image open new window.
Index	String	False	Index of the content section. Determines the order of the section in the landing page.

Left	<i>String</i>	<i>False</i>	Left property of the HTML section.
LinkUrl	<i>String</i>	<i>False</i>	URL parameter of a link type section.
Opacity	<i>String</i>	<i>False</i>	Opacity property of the HTML section.
Top	<i>String</i>	<i>False</i>	Top property of the HTML section.
Value	<i>String</i>	<i>False</i>	Type of content section.
Width	<i>String</i>	<i>False</i>	Width property of the HTML section.
ZIndex	<i>String</i>	<i>False</i>	Z-index property of the HTML section.

Result Set Columns

Name	Type	Description
Id	<i>String</i>	Id of the landing page.

UpdateLandingPageTemplateContent

Updates the content for the target landing page template.

EXECUTE Example:

```
EXECUTE UpdateLandingPageTemplateContent File =
'C:\\Users\\Dell\\Desktop\\Test_Logs\\Marketo.html',
LandingPageTemplateId = '2'
```

Input

Name	Type	Required	Description
LandingPageTemplateId	<i>Integer</i>	<i>True</i>	Id of the landing page template.
File	<i>String</i>	<i>True</i>	Multipart File

Result Set Columns

Name	Type	Description
Id	<i>String</i>	Id of the asset.

UpdateLeadProgramStatus

Changes the program status of a list of leads in a target program. Only existing members of the program may have their status changed with this API.

EXECUTE Example:

```
EXECUTE UpdateLeadProgramStatus ProgramId = '1091', LeadIds =
'1021579,1021686', Status = 'Not in program'
```

Input

Name	Type	Required	Description
ProgramId	String	True	Id of the batch campaign to schedule.
LeadIds	String	True	A comma separated list of lead ids for input.
Status	String	True	Program status of the record.

Result Set Columns

Name	Type	Description
Success	String	Whether the request succeeded

Connection String Options

The connection string properties are the various options that can be used to establish a connection. This section provides a complete list of the options you can configure in the connection string for this provider. Click the links for further details.

For more information on establishing a connection, see [Basic Tab](#).

REST

Property	Description
RESTEndpoint	The Marketo REST API Endpoint.

SOAP

Property	Description
UserId	The Marketo SOAP API User Id.
EncryptionKey	The Marketo SOAP API Encryption Key.
SOAPEndpoint	The Marketo SOAP API Endpoint.

Schema

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
Schema	The type of schema to use.

OAuth

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client Id assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.

OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

SSL

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

Firewall

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

Proxy

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

Logging

Property	Description
LogModules	Core modules to be included in the log file.

Miscellaneous

Property	Description
JobPollingInterval	Specifies the polling interval (in seconds) when checking the status of a bulk API job.

MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Marketo.
Readonly	You can use this property to enforce read-only access to Marketo from the provider.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseBulkAPI	Specifies whether to use the Marketo Bulk API.
UserDefinedViews	A filepath pointing to the JSON configuration file containing your custom views.

REST

This section provides a complete list of the REST properties you can configure in the connection string for this provider.

Property	Description
RESTEndpoint	The Marketo REST API Endpoint.

RESTEndpoint

The Marketo REST API Endpoint.

Data Type

string

Default Value

""

Remarks

The URL of the REST Web service endpoint is provided by Marketo on the Admin page of the Marketo website.

SOAP

This section provides a complete list of the SOAP properties you can configure in the connection string for this provider.

Property	Description
UserId	The Marketo SOAP API User Id.
EncryptionKey	The Marketo SOAP API Encryption Key.
SOAPEndpoint	The Marketo SOAP API Endpoint.

UserId

The Marketo SOAP API User Id.

Data Type

string

Default Value

""

Remarks

The User Id is provided by Marketo and is used to authenticate to the Marketo SOAP Web service.

EncryptionKey

The Marketo SOAP API Encryption Key.

Data Type

string

Default Value

""

Remarks

The EncryptionKey is generated on the Admin page of the Marketo website and is used to authenticate to the Marketo SOAP Web service.

SOAPEndpoint

The Marketo SOAP API Endpoint.

Data Type

string

Default Value

""

Remarks

The URL of the SOAP Web service endpoint is provided by Marketo on the Admin page of the Marketo website.

Schema

This section provides a complete list of the Schema properties you can configure in the connection string for this provider.

Property	Description
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
Schema	The type of schema to use.

Location

A path to the directory that contains the schema files defining tables, views, and stored procedures.

Data Type

string

Default Value

"%APPDATA%\CDATA\Marketo Data Provider\Schema"

Remarks

The path to a directory which contains the schema files for the adapter (.rsd files for tables and views, .rsb files for stored procedures). The folder location can be a relative path from the location of the executable. The [Location](#) property is only needed if you want to customize definitions (for example, change a column name, ignore a column, and so on) or extend the data model with new tables, views, or stored procedures.

Note: Given that this adapter supports multiple schemas, the structure for Marketo custom schema files is as follows:

- Each schema is given a folder corresponding to that schema name.

- These schema folders are contained in a parent folder.
- The **parent folder** should be set as the Location, not an individual schema's folder.

If left unspecified, the default location is "%APPDATA%\CDData\Marketo Data Provider\Schema" with %**APPDATA**% being set to the user's configuration directory:

Platform	%APPDATA%
Windows	The value of the APPDATA environment variable
Mac	~/Library/Application Support
Linux	~/.config

Schema

The type of schema to use.

Possible Values

REST, SOAP

Data Type

string

Default Value

"REST"

Remarks

The schemas available are REST (to use Marketo's REST API) and SOAP (to use Marketo's SOAP API).

OAuth

This section provides a complete list of the OAuth properties you can configure in the connection string for this provider.

Property	Description
InitiateOAuth	Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.
OAuthClientId	The client Id assigned when you register your application with an OAuth authorization server.
OAuthClientSecret	The client secret assigned when you register your application with an OAuth authorization server.
OAuthAccessToken	The access token for connecting using OAuth.
OAuthSettingsLocation	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.

InitiateOAuth

Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.

Possible Values

OFF, GETANDREFRESH, REFRESH

Data Type

string

Default Value

"GETANDREFRESH"

Remarks

The following options are available:

1. **OFF:** Indicates that the OAuth flow will be handled entirely by the user. An OAuthAccessToken will be required to authenticate.
2. **GETANDREFRESH:** Indicates that the entire OAuth Flow will be handled by the adapter. If no token currently exists, it will be obtained by prompting the user via the browser. If a token exists, it will be refreshed when applicable.
3. **REFRESH:** Indicates that the adapter will only handle refreshing the OAuthAccessToken. The user will never be prompted by the adapter to authenticate via the browser. The user must handle obtaining the OAuthAccessToken and OAuthRefreshToken initially.

OAuthClientId

The client Id assigned when you register your application with an OAuth authorization server.

Data Type

string

Default Value

""

Remarks

As part of registering an OAuth application, you will receive the OAuthClientId value, sometimes also called a consumer key, and a client secret, the [OAuthClientSecret](#).

OAuthClientSecret

The client secret assigned when you register your application with an OAuth authorization server.

Data Type

string

Default Value

""

Remarks

As part of registering an OAuth application, you will receive the [OAuthClientId](#), also called a consumer key. You will also receive a client secret, also called a consumer secret. Set the client secret in the [OAuthClientSecret](#) property.

OAuthAccessToken

The access token for connecting using OAuth.

Data Type

string

Default Value

""

Remarks

The [OAuthAccessToken](#) property is used to connect using OAuth. The [OAuthAccessToken](#) is retrieved from the OAuth server as part of the authentication process. It has a server-dependent timeout and can be reused between requests.

The access token is used in place of your user name and password. The access token protects your credentials by keeping them on the server.

OAuthSettingsLocation

The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.

Data Type

string

Default Value

"%APPDATA%\\CData\\Marketo Data Provider\\OAuthSettings.txt"

Remarks

When [InitiateOAuth](#) is set to GETANDREFRESH or REFRESH, the adapter saves OAuth values to avoid requiring the user to manually enter OAuth connection properties and allowing the credentials to be shared across connections or processes.

Alternatively to specifying a file path, memory storage can be used instead. Memory locations are specified by using a value starting with 'memory://' followed by a unique identifier for that set of credentials (ex: memory://user1). The identifier can be anything you choose but should be unique to the user. Unlike with the file based storage, you must manually store the credentials when closing the connection with memory storage to be able to set them in the connection when the process is started again. The OAuth property values can be retrieved with a query to the sys_connection_props system table. If there are multiple connections using the same credentials, the properties should be read from the last connection to be closed.

If left unspecified, the default location is "%APPDATA%\\CData\\Marketo Data Provider\\OAuthSettings.txt" with %**APPDATA**% being set to the user's configuration directory:

Platform	%APPDATA%
Windows	The value of the APPDATA environment variable
Mac	~/Library/Application Support
Linux	~/.config

OAuthExpiresIn

The lifetime in seconds of the OAuth AccessToken.

Data Type

string

Default Value

""

Remarks

Pair with OAuthTokenTimestamp to determine when the AccessToken will expire.

OAuthTokenTimestamp

The Unix epoch timestamp in milliseconds when the current Access Token was created.

Data Type

string

Default Value

""

Remarks

Pair with OAuthExpiresIn to determine when the AccessToken will expire.

SSL

This section provides a complete list of the SSL properties you can configure in the connection string for this provider.

Property	Description
SSLServerCert	The certificate to be accepted from the server when connecting using TLS/SSL.

SSLServerCert

The certificate to be accepted from the server when connecting using TLS/SSL.

Data Type

string

Default Value

""

Remarks

If using a TLS/SSL connection, this property can be used to specify the TLS/SSL certificate to be accepted from the server. Any other certificate that is not trusted by the machine is rejected.

This property can take the following forms:

Description	Example
A full PEM Certificate (example shortened for brevity)	-----BEGIN CERTIFICATE----- MIICHTCCAe4CAQAwDQYJKoZIhvd.....Qw == -----END CERTIFICATE-----
A path to a local file containing the certificate	C:\cert.cer
The public key (example shortened for brevity)	-----BEGIN RSA PUBLIC KEY----- MIGfMA0GCSq.....AQAB -----END RSA PUBLIC KEY-----

The MD5 Thumbprint (hex values can also be either space or colon separated) `34a92226ae0819f2ec14b4a3d904f801cbb150d`

The SHA1 Thumbprint (hex values can also be either space or colon separated) `34a92226ae0819f2ec14b4a3d904f801cbb150d`

If not specified, any certificate trusted by the machine is accepted.

Certificates are validated as trusted by the machine based on the System's trust store. The trust store used is the 'javax.net.ssl.trustStore' value specified for the system. If no value is specified for this property, Java's default trust store is used (for example, JAVA_HOME\lib\security\cacerts).

Use '*' to signify to accept all certificates. Note that this is not recommended due to security concerns.

Firewall

This section provides a complete list of the Firewall properties you can configure in the connection string for this provider.

Property	Description
FirewallType	The protocol used by a proxy-based firewall.
FirewallServer	The name or IP address of a proxy-based firewall.
FirewallPort	The TCP port for a proxy-based firewall.
FirewallUser	The user name to use to authenticate with a proxy-based firewall.
FirewallPassword	A password used to authenticate to a proxy-based firewall.

FirewallType

The protocol used by a proxy-based firewall.

Possible Values

NONE, TUNNEL, SOCKS4, SOCKS5

Data Type

string

Default Value

"NONE"

Remarks

This property specifies the protocol that the adapter will use to tunnel traffic through the [FirewallServer](#) proxy. Note that by default, the adapter connects to the system proxy; to disable this behavior and connect to one of the following proxy types, set [ProxyAutoDetect](#) to false.

Type	Default Port	Description
TUNNEL	80	When this is set, the adapter opens a connection to Marketo and traffic flows back and forth through the proxy.
SOCKS4	1080	When this is set, the adapter sends data through the SOCKS 4 proxy specified by FirewallServer and FirewallPort and passes the FirewallUser value to the proxy, which determines if the connection request should be granted.
SOCKS5	1080	When this is set, the adapter sends data through the SOCKS 5 proxy specified by FirewallServer and FirewallPort . If your proxy requires authentication, set FirewallUser and FirewallPassword to credentials the proxy recognizes.

To connect to HTTP proxies, use [ProxyServer](#) and [ProxyPort](#). To authenticate to HTTP proxies, use [ProxyAuthScheme](#), [ProxyUser](#), and [ProxyPassword](#).

FirewallServer

The name or IP address of a proxy-based firewall.

Data Type

string

Default Value

""

Remarks

This property specifies the IP address, DNS name, or host name of a proxy allowing traversal of a firewall. The protocol is specified by [FirewallType](#): Use [FirewallServer](#) with this property to connect through SOCKS or do tunneling. Use [ProxyServer](#) to connect to an HTTP proxy.

Note that the adapter uses the system proxy by default. To use a different proxy, set [ProxyAutoDetect](#) to false.

FirewallPort

The TCP port for a proxy-based firewall.

Data Type

int

Default Value

0

Remarks

This specifies the TCP port for a proxy allowing traversal of a firewall. Use [FirewallServer](#) to specify the name or IP address. Specify the protocol with [FirewallType](#).

FirewallUser

The user name to use to authenticate with a proxy-based firewall.

Data Type

string

Default Value

""

Remarks

The [FirewallUser](#) and [FirewallPassword](#) properties are used to authenticate against the proxy specified in [FirewallServer](#) and [FirewallPort](#), following the authentication method specified in [FirewallType](#).

FirewallPassword

A password used to authenticate to a proxy-based firewall.

Data Type

string

Default Value

""

Remarks

This property is passed to the proxy specified by [FirewallServer](#) and [FirewallPort](#), following the authentication method specified by [FirewallType](#).

Proxy

This section provides a complete list of the Proxy properties you can configure in the connection string for this provider.

Property	Description
ProxyAutoDetect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
ProxyServer	The hostname or IP address of a proxy to route HTTP traffic through.
ProxyPort	The TCP port the ProxyServer proxy is running on.
ProxyAuthScheme	The authentication type to use to authenticate to the ProxyServer proxy.
ProxyUser	A user name to be used to authenticate to the ProxyServer proxy.
ProxyPassword	A password to be used to authenticate to the ProxyServer proxy.
ProxySSLType	The SSL type to use when connecting to the ProxyServer proxy.
ProxyExceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

ProxyAutoDetect

This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.

Data Type

bool

Default Value

true

Remarks

This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.

NOTE: When this property is set to True, the proxy used is determined as follows:

- A search from the JVM properties (**http.proxy**, **https.proxy**, **socksProxy**, **etc.**) is performed.
- In the case that the JVM properties don't exist, a search from **java.home/lib/net.properties** is performed.
- In the case that java.net.useSystemProxies is set to True, a search from **the SystemProxy** is performed.
- In Windows only, an attempt is made to retrieve these properties from the **Internet Options** in the **registry**.

To connect to an HTTP proxy, see [ProxyServer](#). For other proxies, such as SOCKS or tunneling, see [FirewallType](#).

ProxyServer

The hostname or IP address of a proxy to route HTTP traffic through.

Data Type

string

Default Value

""

Remarks

The hostname or IP address of a proxy to route HTTP traffic through. The adapter can use the HTTP, Windows (NTLM), or Kerberos authentication types to authenticate to an HTTP proxy.

If you need to connect through a SOCKS proxy or tunnel the connection, see [FirewallType](#).

By default, the adapter uses the system proxy. If you need to use another proxy, set [ProxyAutoDetect](#) to false.

ProxyPort

The TCP port the ProxyServer proxy is running on.

Data Type

int

Default Value

80

Remarks

The port the HTTP proxy is running on that you want to redirect HTTP traffic through. Specify the HTTP proxy in [ProxyServer](#). For other proxy types, see [FirewallType](#).

ProxyAuthScheme

The authentication type to use to authenticate to the ProxyServer proxy.

Possible Values

BASIC, DIGEST, NONE, NEGOTIATE, NTLM, PROPRIETARY

Data Type

string

Default Value

"BASIC"

Remarks

This value specifies the authentication type to use to authenticate to the HTTP proxy specified by [ProxyServer](#) and [ProxyPort](#).

Note that the adapter will use the system proxy settings by default, without further configuration needed; if you want to connect to another proxy, you will need to set

[ProxyAutoDetect](#) to false, in addition to [ProxyServer](#) and [ProxyPort](#). To authenticate, set [ProxyAuthScheme](#) and set [ProxyUser](#) and [ProxyPassword](#), if needed.

The authentication type can be one of the following:

- **BASIC:** The adapter performs HTTP BASIC authentication.
- **DIGEST:** The adapter performs HTTP DIGEST authentication.
- **NEGOTIATE:** The adapter retrieves an NTLM or Kerberos token based on the applicable protocol for authentication.
- **PROPRIETARY:** The adapter does not generate an NTLM or Kerberos token. You must supply this token in the Authorization header of the HTTP request.

If you need to use another authentication type, such as SOCKS 5 authentication, see [FirewallType](#).

ProxyUser

A user name to be used to authenticate to the ProxyServer proxy.

Data Type

string

Default Value

""

Remarks

The [ProxyUser](#) and [ProxyPassword](#) options are used to connect and authenticate against the HTTP proxy specified in [ProxyServer](#).

You can select one of the available authentication types in [ProxyAuthScheme](#). If you are using HTTP authentication, set this to the user name of a user recognized by the HTTP proxy. If you are using Windows or Kerberos authentication, set this property to a user name in one of the following formats:

```
user@domain  
domain\user
```

ProxyPassword

A password to be used to authenticate to the ProxyServer proxy.

Data Type

string

Default Value

""

Remarks

This property is used to authenticate to an HTTP proxy server that supports NTLM (Windows), Kerberos, or HTTP authentication. To specify the HTTP proxy, you can set [ProxyServer](#) and [ProxyPort](#). To specify the authentication type, set [ProxyAuthScheme](#).

If you are using HTTP authentication, additionally set [ProxyUser](#) and [ProxyPassword](#) to HTTP proxy.

If you are using NTLM authentication, set [ProxyUser](#) and [ProxyPassword](#) to your Windows password. You may also need these to complete Kerberos authentication.

For SOCKS 5 authentication or tunneling, see [FirewallType](#).

By default, the adapter uses the system proxy. If you want to connect to another proxy, set [ProxyAutoDetect](#) to false.

ProxySSLType

The SSL type to use when connecting to the ProxyServer proxy.

Possible Values

AUTO, ALWAYS, NEVER, TUNNEL

Data Type

string

Default Value

"AUTO"

Remarks

This property determines when to use SSL for the connection to an HTTP proxy specified by [ProxyServer](#). This value can be AUTO, ALWAYS, NEVER, or TUNNEL. The applicable values are the following:

AUTO	Default setting. If the URL is an HTTPS URL, the adapter will use the TUNNEL option. If the URL is an HTTP URL, the component will use the NEVER option.
ALWAYS	The connection is always SSL enabled.
NEVER	The connection is not SSL enabled.
TUNNEL	The connection is through a tunneling proxy. The proxy server opens a connection to the remote host and traffic flows back and forth through the proxy.

ProxyExceptions

A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

Data Type

string

Default Value

""

Remarks

The [ProxyServer](#) is used for all addresses, except for addresses defined in this property. Use semicolons to separate entries.

Note that the adapter uses the system proxy settings by default, without further configuration needed; if you want to explicitly configure proxy exceptions for this connection, you need to set [ProxyAutoDetect](#) = false, and configure [ProxyServer](#) and [ProxyPort](#). To authenticate, set [ProxyAuthScheme](#) and set [ProxyUser](#) and [ProxyPassword](#), if needed.

Logging

This section provides a complete list of the Logging properties you can configure in the connection string for this provider.

Property	Description
LogModules	Core modules to be included in the log file.

LogModules

Core modules to be included in the log file.

Data Type

string

Default Value

""

Remarks

Only the modules specified (separated by ';') will be included in the log file. By default all modules are included.

See the [Logging](#) page for an overview.

Miscellaneous

This section provides a complete list of the Miscellaneous properties you can configure in the connection string for this provider.

Property	Description
JobPollingInterval	Specifies the polling interval (in seconds) when checking the status of a bulk API job.
MaxRows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
Other	These hidden properties are used only in specific use cases.
Pagesize	The maximum number of results to return per page from Marketo.
Readonly	You can use this property to enforce read-only access to Marketo from the provider.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.
UseBulkAPI	Specifies whether to use the Marketo Bulk API.
UserDefinedViews	A filepath pointing to the JSON configuration file containing your custom views.

JobPollingInterval

Specifies the polling interval (in seconds) when checking the status of a bulk API job.

Data Type

int

Default Value

120

Remarks

This property is used to specify the polling interval (in seconds) to identify when a bulk API job has completed. The adapter will wait JobPollingInterval seconds between calls to check a bulk API job status. Once the job is identified as 'Completed', the adapter will download and parse the generated file returning the results of the specified query.

This property can be set to 0 to just create and enqueue a job in which case the Job Id will be returned in the result set. The job status can then be checked using stored procedures.

Note: This property is only applicable when [UseBulkAPI](#) is set to True. See the [UseBulkAPI](#) page for more information about using the Bulk API.

MaxRows

Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.

Data Type

int

Default Value

-1

Remarks

Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.

Other

These hidden properties are used only in specific use cases.

Data Type

string

Default Value

""

Remarks

The properties listed below are available for specific use cases. Normal driver use cases and functionality should not require these properties.

Specify multiple properties in a semicolon-separated list.

Integration and Formatting

DefaultColumnSize	Sets the default length of string fields when the data source does not provide column length in the metadata. The default value is 2000.
ConvertDateTimeToGMT	Determines whether to convert date-time values to GMT, instead of the local time of the machine.
RecordToFile=filename	Records the underlying socket data transfer to the specified file.

Pagesize

The maximum number of results to return per page from Marketo.

Data Type

int

Default Value

1000

Remarks

The Pagesize property affects the maximum number of results to return per page from Marketo. Setting a higher value may result in better performance at the cost of additional

memory allocated per page consumed.

Readonly

You can use this property to enforce read-only access to Marketo from the provider.

Data Type

bool

Default Value

false

Remarks

If this property is set to true, the adapter will allow only SELECT queries. INSERT, UPDATE, DELETE, and stored procedure queries will cause an error to be thrown.

Timeout

The value in seconds until the timeout error is thrown, canceling the operation.

Data Type

string

Default Value

"300"

Remarks

If the Timeout property is set to 0, operations do not time out: They run until they complete successfully or encounter an error condition.

If Timeout expires and the operation is not yet complete, the adapter throws an exception.

UseBulkAPI

Specifies whether to use the Marketo Bulk API.

Data Type

bool

Default Value

false

Remarks

When set to 'True', the Marketo Bulk API will be used to extract or import data, where applicable. The Bulk API is an interface that allows you to retrieve or import large sets of data using delimited (CSV, TSV, or SSV) files. Currently the only tables that support the Bulk API are: Leads (extract and import) and Activities (extract). For any tables that do not support the Bulk API, this property will be ignored.

The Bulk API causes all the data to be retrieved in a single request and requires the data to be accumulated on the server side prior to sending. Therefore requesting a large amount of data using the Bulk API may be advantageous over using the REST API and you may see performance improvements. Additionally the Bulk API requires less API requests to be made (which helps preserve your API calls and staying within the API restrictions enforced by Marketo).

To use the Bulk API to extract records, a job must be created and enqueued. Once enqueued, Marketo will begin processing the job to retrieve the requested data and generate the delimited file. The status of the job can be polled to determine the current status and whether the file is available to be downloaded. Once the status shows that the job is complete and the file is ready, the data can then be downloaded.

When UseBulkAPI is set to True and JobPollingInterval is set to a value greater than 0, the adapter will perform all the previous mentioned steps for you when executing a SELECT query on a Leads or Activities table. This will create and enqueue a job with the specified columns and filters. Note that a filter is required when exporting bulk data. For the Activities tables, an ActivityDate range must be specified. For the Leads table, a CreatedAt or UpdatedAt range may be specified or a Static or Smart list. The adapter will poll the job status to identify when the job has completed, waiting JobPollingInterval seconds in

between calls. Once the job is complete, the adapter will download the delimited file that was created, parse it, and return the results for the specified query.

Note that job status calls count against your API call limit and thus it is suggested to space out your status requests based on the amount of data you are requesting. The job status polling interval is configurable via [JobPollingInterval](#). Marketo will only update the status every 60 seconds and thus it is suggested that your polling interval be larger than 60 seconds. When expecting large datasets, it may be best to increase the polling interval to a value greater than 5 minutes to minimize API calls. It is possible that it may take a while for the job to be processed and thus it may seem like the query is exhibiting a hanging behavior when it is actually just waiting for the job to complete.

In the case that you want to issue your own job status polling requests, you can set [JobPollingInterval](#) to 0. This will just create and enqueue the job for you when you execute a SELECT query on a Leads or Activities table, returning the JobId in the result set.

Once a job has been enqueued, the status of the job can be polled by calling the GetExportJobStatus stored procedure.

The JobStatus value will be 'Complete' signaling that the job has finished processing and is ready to be downloaded. To finish executing your initial SELECT query, add the JobId filter to the WHERE clause of the initial SELECT statement. This query will download the file for the specified JobId and parse the result set.

Logic/Code Example (JobPollingInterval = 0):

```
SELECT JobId, Company, FirstName AS fn, LastName AS ln FROM Leads WHERE
CreatedAt>='10/01/2017' AND CreatedAt<'10/31/2017'
# Retrieve the JobId value from the ResultSet (e.g. c4ebf745-b0e3-4bb8-
bfc9-bd8472a28d35). Only one row is returned and JobId will be the only
relevant value returned.
loop(desired time interval) {
    EXEC GetExportJobStatus @JobId='c4ebf745-b0e3-4bb8-bfc9-bd8472a28d35',
    @Type='Leads'
    if (JobStatus == 'Completed') break;
}
SELECT Company, FirstName AS fn, LastName AS ln FROM Leads WHERE
CreatedAt>='10/01/2016' AND CreatedAt<'10/31/2016' AND JobId='c4ebf745-
b0e3-4bb8-bfc9-bd8472a28d35'
```

Note: this property is only applicable when using the REST API.

UserDefinedViews

A filepath pointing to the JSON configuration file containing your custom views.

Data Type

string

Default Value

""

Remarks

User Defined Views are defined in a JSON-formatted configuration file called *UserDefinedViews.json*. The adapter automatically detects the views specified in this file.

You can also have multiple view definitions and control them using the [UserDefinedViews](#) connection property. When you use this property, only the specified views are seen by the adapter.

This User Defined View configuration file is formatted as follows:

- Each root element defines the name of a view.
- Each root element contains a child element, called **query**, which contains the custom SQL query for the view.

For example:

```
{
  "MyView": {
    "query": "SELECT * FROM Leads WHERE MyColumn = 'value'"
  },
  "MyView2": {
    "query": "SELECT * FROM MyTable WHERE Id IN (1,2,3)"
  }
}
```

Use the [UserDefinedViews](#) connection property to specify the location of your JSON configuration file. For example:

```
"UserDefinedViews",
"C:\\Users\\yourusername\\Desktop\\tmp\\UserDefinedViews.json"
```

TIBCO Product Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join the TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The [TIBCO Product Documentation](#) website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the [TIBCO® Data Virtualization](#) page.

- **Users**
 - TDV Getting Started Guide
 - TDV User Guide
 - TDV Web UI User Guide
 - TDV Client Interfaces Guide
 - TDV Tutorial Guide
 - TDV Northbay Example
- **Administration**
 - TDV Installation and Upgrade Guide
 - TDV Administration Guide
 - TDV Active Cluster Guide
 - TDV Security Features Guide
- **Data Sources**

TDV Adapter Guides

TDV Data Source Toolkit Guide (Formerly Extensibility Guide)

- **References**

TDV Reference Guide

TDV Application Programming Interface Guide

- **Other**

TDV Business Directory Guide

TDV Discovery Guide

- *TIBCO TDV and Business Directory Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on the website.

Release Version Support

TDV 8.5 is designated as a Long Term Support (LTS) version. Some release versions of TIBCO® Data Virtualization products are selected to be long-term support (LTS) versions. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also

https://docs.tibco.com/pub/tdv/general/LTS/tdv_LTS_releases.htm.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, visit [TIBCO Community](#).

Legal and Third-Party Notices

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, TIBCO logo, TIBCO O logo, ActiveSpaces, Enterprise Messaging Service, Spotfire, TERR, S-PLUS, and S+ are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the

readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2002-2023 Cloud Software Group, Inc All Rights Reserved.