

## clearMDM – API Guide v0.1 (product version 1.20.x)

Date: October 2016

Work in Progress Draft

*Product support can be contacted directly (support@clearmdm.com) for specific questions not addressed by content of this document or for additional information or feedback.*

### Table of Contents

|   |           |
|---|-----------|
| <b>clearMDM – API Guide v0.1 (product version 1.20.x)</b> ..... | <b>1</b>  |
| <b>Product Introduction</b> .....                               | <b>2</b>  |
| <b>Key Concepts</b> .....                                       | <b>3</b>  |
| <b>API Design Principles</b> .....                              | <b>4</b>  |
| REST API .....  | 4         |
| Custom Operations .....   | 4         |
| <b>Key Use Cases</b> .....                                      | <b>5</b>  |
| MDM Job Invocation .....  | 5         |
| Distributed MDM - Point-of-entry Protection .....               | 5         |
| Distributed MDM - Transactional .....                           | 5         |
| <b>Technical Specification</b> .....                            | <b>6</b>  |
| Authentication .....  | 6         |
| Status Codes .....  | 6         |
| API Operations – Low-Level MDM Resources .....                  | 6         |
| API Operations – High-Level MDM Processes .....                 | 15        |
| <b>Appendix A – Flow Designer – Example Flow Action</b> .....   | <b>16</b> |

## Product Introduction

clearMDM is a native Salesforce application (installed from the AppExchange®) that delivers a master data management solution for the Salesforce® platform. There are no off-platform components to the solution, customer data is not moved off-platform for processing by external services and there is no mobile application.

The clearMDM product supports the MDM operations outlined below in their logical sequence.

(ABT) **Normalisation.** Standardisation of field values using list Custom Setting lookups (Country codes, Root First Names etc.) or predefined transformation formats (email domain, URL, Title case etc.). Normalisation generates Blocking Keys (or match keys) used as the first pass in record matching.

(ABTU) **Matching.** Cross-object matching of records using key, deterministic and probabilistic (fuzzy) matching rules.

(B) **Internal Matching.** Single object large data volume matching operation.

(ABU) **Merge.** Custom merge functionality compatible with Standard and Custom objects and also portal enabled accounts, person accounts and contacts. There is no dependency on the standard platform merge capability.

(AB) **Synchronisation.** Master Records are updated directly to reflect field modifications on their Source Record. Synchronisation adds efficiency as the 2-step match and merge cycle is avoided; existing Source Record to Master Record relationships are utilised.

(AB) **Conversion.** Master Record creation on a one-to-one basis for unmatched records or external data where matching is not required. Once paired Master Records updated directly to reflect field modifications on their Source Records.

(B) **Re-parenting.** Child Records related to a Source Record can be re-parented to the related Master Record via lookup relationship.

(B) **Custom Rollups.** Numeric fields on Child Records can be aggregated to the Master Record (via lookup relationship).

Prefixes indicate entry points:

B = Batch (Scheduled Job or Custom Action – REST API/Process Builder)

T = ApexTrigger

U = User Interface/Manual

A = REST API

## Key Concepts

**Target Object.** A compatible object that is defined as the target for Normalisation operations and where Master Records will be created or updated.

**Data Source.** A compatible object that provides data to MDM operations. Partition data sources enable a single object to support multiple data sources each with distinct configurations.

A broad range of Standard Objects (including Person Accounts) are supported as both Target Objects and Data Sources. Custom Objects are also supported.

**Master Record.** The Master Record is the output of the Conversion and Merge MDM operations and is optionally related to the underlying Source Records via relationship field or simply through the concatenation of record identifiers into a field on the Master Record.

**Source Record.** Source Records are the input to MDM operations and may be related to a Master Record. Source Records can be retained or deleted.

# CLEARMDM

## API Design Principles

### REST API

The clearMDM API is a lightweight Web Service interface to the MDM operations implemented within the clearMDM managed package. The API is implemented as a custom set of resources that extend the standard Force.com REST API.

The documentation for the Force.com REST API can be found at the location below.

[https://developer.salesforce.com/docs/atlas.en-us.api\\_rest.meta/api\\_rest/intro\\_what\\_is\\_rest\\_api.htm](https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/intro_what_is_rest_api.htm)

### Custom Operations

The clearMDM product design philosophy is to deliver a robust set of independent MDM operations that can be assembled in a flexible manner to deliver the required data management processes. Within the Salesforce application MDM operations can be orchestrated via Batch Job scheduling, manual invocation or through standard process automation features such as Process Builder. The clearMDM API is consistent with this design approach. MDM operations (Matching, Merge etc.) are exposed as independent API resources that can be invoked directly where external orchestration is required.

In technical terms the API resources are implemented as Apex Actions and can be invoked wherever Apex Actions are supported on the Salesforce platform. The Force.com REST API exposes Apex Actions as callable resources via the actions endpoint (example path below).

```
/v38.0/actions/custom/apex/clearmdm__RecordMatchingAction
```

Force.com Flow (or Visual Workflow) can be implemented to orchestrate MDM operations into higher-level MDM processes that blend single or multiple MDM operations with custom business logic. Flows are exposed as Custom Flow Actions via the Force.com REST API. As such, high-level MDM processes implemented as flows can be exposed as distinct API callable resources – example path below.

```
/v38.0/actions/custom/flow/clearmdm__MatchAndMergeSourceRecord
```

The clearMDM package includes a complete set of low-level API resources (Apex Actions) and a single example high-level MDM process (Flow Action). [Appendix A](#) provides a screenshot of how the example flow appears within the Flow Design Environment.

## Key Use Cases

The table below provides summary of the key uses that the clearMDM API is intended to address. Note, this is not exhaustive – and as outlined in the preceding section, custom API resources are straightforward to assemble using Force.com Flow and the low-level Apex Actions that expose MDM functionality.

### MDM Job Invocation

Jobs that represent bulk record MDM operations can be invoked via the API. This allows data integration tools (ETL, Middleware etc.) to invoke MDM jobs as a post-processing step. This capability increases the efficiency of the end-to-end data processing cycle and removes dependencies on scheduled tasks.

### Distributed MDM - Point-of-entry Protection

Where clearMDM is implemented in a MDM hub architecture, the API allows external (spoke) systems to efficiently check the existence of new records in the master data set within point-of-entry functions. Validation of this type can prevent the creation of duplicate records at source.

### Distributed MDM - Transactional

The clearMDM API has been designed to support message-based or transactional data integration flows where new/updated source records are presented in near-real-time via an ESB (Enterprise Service Bus) or equivalent technology. A message-oriented architecture typically requires that end-to-end record processing occurs on receipt of the message not via subsequent batch-style data processing tasks. For example, it should be possible to return the master record id for a new source record within a single API interaction.

# CLEARMDM

## Technical Specification

The following sub-sections outline the key technical aspects of the clearMDM API. As the API is implemented as an extension to the Force.com REST API in many places references are provided to the appropriate standard API documentation.

Primary Force.com REST API Resources:

[https://developer.salesforce.com/page/REST\\_API](https://developer.salesforce.com/page/REST_API)

[https://developer.salesforce.com/docs/atlas.en-us.api\\_rest.meta/api\\_rest/](https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/)

For clarity, the clearMDM is a RESTful web service interface that exposes REST resources for the purposes of manipulating MDM operations for bulk record or transactional record processing. The API supports both XML and JSON format messages; JSON is the default.

### Authentication

Authentication to the Force.com REST API is via the standard OAuth 2.0 protocol.

A complete description of the authentication model can be found at the link below.

[https://developer.salesforce.com/docs/atlas.en-us.api\\_rest.meta/api\\_rest/](https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/)

### Status Codes

The primary indicator of the success or failure of an API request is indicated by the HTTP Status Code, the list of applicable values is found at the link below.

[https://developer.salesforce.com/docs/atlas.en-us.api\\_rest.meta/api\\_rest/errorcodes.htm](https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/errorcodes.htm)

In failure cases it is likely that the response body will contain additional information in relation to the cause of the error.

### API Operations – Low-Level MDM Resources

The full list of Apex Actions and Flow Actions can be queried as below.

```
GET https://<instance>.salesforce.com/services/data/<version>/actions/custom/apex
GET https://<instance>.salesforce.com/services/data/<version>/actions/custom/flow
```

The details of a specific resource can be queried as below – the response includes input and output parameter definitions.

```
GET
https://<instance>.salesforce.com/services/data/<version>/actions/custom/apex/clearmdm__Batch
JobRunAction
```

# CLEARMDM

## 1. BatchJobRunAction

Purpose: Invoke the specified MDM Job (bulk-record) for immediate processing.

Path : /services/data/v38.0/actions/custom/apex/clearmdm\_\_BatchJobRunAction

| Input        | Data Type | Required? | Example           | Description                                     |
|--------------|-----------|-----------|-------------------|---|
| targetObject | Text      | Yes       | Account           | Account, PersonAccount etc.                     |
| batchJobType | Text      | Yes       | Matching          | Normalisation, Matching, Merge etc.             |
| dataSources  | Text      | No        | "Legacy Accounts" | Comma separated list of Data Source names.      |
| processCount | Integer   | No        | 5                 | 1 to 10 as supported by the specific operation. |
| Output       | Data Type |           | Example           | Description                                     |
| output       | Boolean   |           | True              | Job submission success flag.                    |

Example Request :

```
{
  "inputs": [
    {
      "batchJobType": "Normalisation",
      "processCount": 1,
      "targetObject": "Account"
    },
    {
      "batchJobType": "Normalisation",
      "processCount": 1,
      "targetObject": "Contact"
    }
  ]
}
```

As the JSON above shows, multiple jobs can be invoked from a single API call as the request takes a JSON array as the input.

Example Response :

```
[
  {
    "actionName": "clearmdm__BatchJobRunAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "output": true
    }
  },
  {
    "actionName": "clearmdm__BatchJobRunAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "output": true
    }
  }
]
```

The highlighted section of the example response shows that both job requests were successfully validated and submitted.

## 2. RecordNormalisationAction

Purpose: Normalise one or more Source Records.

Path : /services/data/v38.0/actions/custom/apex/clearmdm\_\_RecordNormalisationAction

| Input    | Data Type | Required? | Example         | Description                |
|----------|-----------|-----------|-----------------|----------------------------|
| recordId | Text      | Yes       | 0010Y0000091mqI | Salesforce Record Id       |
|          |           |           |                 |                            |
| Output   | Data Type |           | Example         | Description                |
| output   | Boolean   |           | True            | Operation completion flag. |
|          |           |           |                 |                            |

Pre-requisites : Source Records must exist within a Salesforce object that is correctly configured (Target Object plus Data Source). External records must be inserted or upserted via a separate API call before calling required MDM resources.

Example Request :

```
{
  "inputs": [
    {
      "recordId": "0010X0000051mqk"
    },
    {
      "recordId": "0010Y0000051mqI"
    }
  ]
}
```

As the JSON above shows, multiple records can be processed from a single API call as the request takes a JSON array as the input.

Example Response :

```
[
  {
    "actionName": "clearmdm__RecordNormalisationAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "output": true
    }
  },
  {
    "actionName": "clearmdm__RecordNormalisationAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "output": true
    }
  }
]
```

The highlighted section of the example response shows that both operation requests were successfully validated and completed.



### 3. RecordMatchingAction

**Purpose:** Perform matching for a given set of records and return matching results.

**Path :** /services/data/v38.0/actions/custom/apex/clearmdm\_\_RecordMatchingAction

| Input    | Data Type | Required? | Example  | Description                         |
|----------|-----------|-----------|--|-------------------------------------|
| recordId | Text      | Yes       | 0010Y0000091mql  | Salesforce Record Id                |
|          |           |           |  |                                     |
| Output   | Data Type |           | Example  | Description                         |
| recordId | Text      |           | 0010Y0000091mql  | Input record Id.                    |
| matches  | Text      |           | 1 Existing Record Matches [Record Name:Acme Inc, Data Source:Account, Score:70.00, Match Type:Fuzzy] | Id,Name pairs for matching records. |
|          |           |           |  |                                     |

**Pre-requisites :** Source Records must exist within a Salesforce object that is correctly configured (Target Object plus Data Source). External records must be inserted or upserted via a separate API call before calling required MDM resources.

**Example Request :**

```
{
  "inputs": [
    {
      "recordId": "0010X0000051mqk"
    },
    {
      "recordId": "0010Y0000051mql"
    }
  ]
}
```

As the JSON above shows, multiple records can be processed from a single API call as the request takes a JSON array as the input.

**Example Response :**

```
[
  {
    "actionName": "clearmdm__RecordMatchingAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "matches": "1 Existing Record Matches [Record Name:BALANCE POINT LTD, Data Source:Account, Score:90.00, Match Type:Fuzzy]",
      "recordId": "0010X0000051mqkQAA"
    }
  },
  {
    "actionName": "clearmdm__RecordMatchingAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "matches": "1 Existing Record Matches [Record Name:BALANCE OUT LTD, Data Source:Account, Score:90.00, Match Type:Fuzzy]",
      "recordId": "0010Y0000051mqlQAA"
    }
  }
]
```

# CLEARMDM

The highlighted section of the example response shows that both operation requests were successfully validated and completed. The returned matching results provides the details of any matching records including record name, match score % and match type (Key or Fuzzy).

Note, the Matching API resource creates Matched Record Pair records for identified matches; this approach allows a subsequent Merge API resource call to merge the matches efficiently. A later version of the clearMDM API will introduce two changes to the Matching API resource; firstly the dependency that Source Records must be persisted before matching will be removed, secondly the creation of Matched Record Pair data will become optional and controlled via an additional input parameter.

## 4. RecordMergeAction

**Purpose:** Perform merge for a given set of records that have previously been matched and return the master record Id per source record.

**Path :** /services/data/v38.0/actions/custom/apex/clearmdm\_\_RecordMergeAction

| Input          | Data Type | Required? | Example         | Description               |
|----------------|-----------|-----------|-----------------|---------------------------|
| recordId       | Text      | Yes       | 0010Y0000091mqk | Salesforce Record Id      |
|                |           |           |                 |                           |
| Output         | Data Type |           | Example         | Description               |
| recordId       | Text      |           | 0010X0000091mqk | Input record Id.          |
| masterRecordId | Text      |           | 0010Y0000061mql | Related master record Id. |
|                |           |           |                 |                           |

**Pre-requisites :** Source Records must exist within a Salesforce object that is correctly configured (Target Object plus Data Source). External records must be inserted or upserted via a separate API call before calling required MDM resources.

**Example Request :**

```
{
  "inputs": [
    {
      "recordId": "0010X0000051mqk"
    },
    {
      "recordId": "0010Y0000051mql"
    }
  ]
}
```

As the JSON above shows, multiple records can be processed from a single API call as the request takes a JSON array as the input.

**Example Response :**

```
[
  {
    "actionName": "clearmdm__RecordMergeAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "masterRecordId": "0010X0000051mqqQAA",
      "recordId": "0010Y0000051mqlQAA"
    }
  },
  {
    "actionName": "clearmdm__RecordMergeAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "masterRecordId": "0010Y0000051mqlQAA",
      "recordId": "0010Y0000051mqlQAA"
    }
  }
]
```

# CLEARMDM

The highlighted section of the example response shows that both operation requests were successfully validated and completed. Note, where the source record is the master record for the matched record group the returned master record Id will be the same as the input record id.

## 5. RecordConversionAction

Purpose: Perform conversion for a given set of records and return results. Note, conversion operations relate to External Data Sources only (Target Object & Source Object are distinct).

Path : /services/data/v38.0/actions/custom/apex/clearmdm\_\_RecordConversionAction

| Input    | Data Type | Required? | Example         | Description                |
|----------|-----------|-----------|-----------------|----------------------------|
| recordId | Text      | Yes       | 0010Y0000091mqI | Salesforce Record Id       |
|          |           |           |                 |                            |
| Output   | Data Type |           | Example         | Description                |
| output   | Boolean   |           | True            | Operation completion flag. |
|          |           |           |                 |                            |

Pre-requisites : Source Records must exist within a Salesforce object that is correctly configured (Target Object plus Data Source). External records must be inserted or upserted via a separate API call before calling required MDM resources.

### Example Request :

```
{
  "inputs": [
    {
      "recordId": "0010X0000051mqk"
    },
    {
      "recordId": "0010Y0000051mqI"
    }
  ]
}
```

As the JSON above shows, multiple records can be processed from a single API call as the request takes a JSON array as the input.

### Example Response :

```
[
  {
    "actionName": "clearmdm__RecordConversionAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "output": true
    }
  },
  {
    "actionName": "clearmdm__RecordConversionAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "output": true
    }
  }
]
```

The highlighted section of the example response shows that both operation requests were successfully validated and completed.

## 6. RecordSynchronisationAction

**Purpose:** Perform synchronisation for a given set of records and return results.

**Path :** /services/data/v38.0/actions/custom/apex/clearmdm\_\_RecordSynchronisationAction

| Input    | Data Type | Required? | Example         | Description                |
|----------|-----------|-----------|-----------------|----------------------------|
| recordId | Text      | Yes       | 0010Y0000091mqI | Salesforce Record Id       |
|          |           |           |                 |                            |
| Output   | Data Type |           | Example         | Description                |
| output   | Boolean   |           | True            | Operation completion flag. |
|          |           |           |                 |                            |

**Pre-requisites :** Source Records must exist within a Salesforce object that is correctly configured (Target Object plus Data Source). External records must be inserted or upserted via a separate API call before calling required MDM resources.

**Example Request :**

```
{
  "inputs": [
    {
      "recordId": "0010X0000051mqk"
    },
    {
      "recordId": "0010Y0000051mqI"
    }
  ]
}
```

As the JSON above shows, multiple records can be processed from a single API call as the request takes a JSON array as the input.

**Example Response :**

```
[
  {
    "actionName": "clearmdm__RecordSynchronisationAction",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "output": true
    }
  },
  {
    "actionName": "clearmdm__RecordSynchronisationAction ",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "output": true
    }
  }
]
```

The highlighted section of the example response shows that both operation requests were successfully validated and completed.

## API Operations – High-Level MDM Processes

### 1. Match and Merge Process

**Purpose:** Perform Match and Merge operations for a given set of records and return the Master Record Id.

**Path :**

/services/data/v38.0/actions/custom/flow/clearmdm\_\_Match\_and\_Merge\_Source\_Record

| Input                 | Data Type | Required? | Example         | Description   |
|-----------------------|-----------|-----------|-----------------|---|
| recordId              | Text      | Yes       | 0010Y0000091mql | Salesforce Record Id                                  |
|                       |           |           |                 |   |
| Output                | Data Type |           | Example         | Description   |
| Flow__InterviewStatus | Text      |           | Created         | One of { Created, Started, Finished, Error, Waiting } |
| MasterRecordId        | Text      |           | 0010Y0000081mqq | Master Record Id                                      |
|                       |           |           |                 |   |

**Pre-requisites :** Source Records must exist within a Salesforce object that is correctly configured (Target Object plus Data Source). External records must be inserted or upserted via a separate API call before calling required MDM resources.

**Example Request :**

```
{
  "inputs": [
    {
      "SourceRecordId": "0010X0000051mqk"
    }
  ]
}
```

As the JSON above shows, a single source record can be processed by each API call.

**Example Response :**

```
[
  {
    "actionName": "clearmdm__Match_and_Merge_Source_Record",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "MasterRecordId": "0010Y0000051mqqQAA",
      "Flow__InterviewStatus": "Finished"
    }
  }
]
```

The highlighted section of the example response shows that the process request was successfully validated and completed.

## Appendix A – Flow Designer – Example Flow Action

