



User Guide Master Data Management

Version 1.2.1
December 2019

**RedPoint Global Inc. | 36 Washington Street, Suite 120
Wellesley Hills, MA 02481
+1 781 725 0250 | www.redpointglobal.com**

Copyright © 2019 RedPoint Global Inc.

Contents

Part I	RedPoint MDM overview	8
1	How it works.....	10
2	Getting help.....	11
	Troubleshooting	12
Part II	Installation and configuration	12
1	System requirements.....	12
2	Installation prerequisites.....	14
3	Install on Windows.....	14
4	Install on Linux.....	18
5	Helpful Docker commands.....	19
6	MDM scripting tool.....	20
7	Configuring services.....	21
	Port assignments	21
	Services access	21
	Configuring SSL for MDM services	22
Part III	The Web App	25
1	Log on to RedPoint MDM.....	26
2	Log off RedPoint MDM.....	27
3	Console overview.....	27
	Navigation menu	28
Part IV	The data hub	29
1	Select/change the current database.....	30
2	Search for a table.....	30
	From the tables page	30
	From the data page	31
3	Query records in a table.....	32
	How to construct a query	33
	Constructing a query in the Console	34
	Example: constructing a query with two terms	35
4	View a record's history.....	36
5	Change metadata (the data model).....	39
	Create a database	39
	Delete a database	39
	Create a table	40
	Delete a table	40
	Modify a table schema	40

Part V	Workflow	41
1	The workflow Console section.....	42
2	Enter a new record.....	43
	Business user workflow	44
	Steward workflow	45
3	Edit an existing record.....	46
	Business user workflow	47
	Steward workflow	48
4	Review automated matches.....	49
	Steward workflow	50
5	Review manual matches.....	52
	Steward workflow	52
Part VI	RedPoint Data Management's MDM tools	56
1	MDM tool connection settings.....	56
2	MDM tool execution options.....	57
3	Trigger input and output.....	59
4	Configure shared settings.....	59
5	MDM Input tool.....	59
	MDM Input tool parameters	60
	Configure the MDM Input tool	61
6	MDM Output tool.....	62
	MDM Output tool parameters	62
	Configure the MDM Output tool	65
7	MDM Update tool.....	66
	MDM Update tool parameters	66
	Configure the MDM Update tool	68
8	MDM Delete tool.....	68
	MDM Delete tool parameters	68
	Configure the MDM Delete tool	70
9	MDM Key Query tool.....	70
	MDM Key Query tool parameters	70
	Configure the MDM Key Query tool	71
10	MDM Dynamic Query tool.....	72
	MDM Dynamic Query tool parameters	72
	Configure the MDM Dynamic Query tool	74
11	MDM Query View tool.....	74
	MDM Query View tool parameters	75
	Configure the MDM Query View tool	76
12	MDM Set Match Groups tool.....	76
	MDM Set Match Groups tool parameters	77
	Configure the MDM Set Match Groups tool	78
13	MDM Generate Match Group IDs tool.....	79
	MDM Generate Match Group IDs tool parameters	79
	Configure the MDM Generate Match Group IDs tool	79

14	MDM Match Group Query tool	80
	MDM Match Group Query tool parameters	80
	Configure the MDM Match Group Query tool	81
15	MDM Match Suppression Query tool.....	82
	MDM Match Suppression Query tool parameters	82
	Configure the Match Suppression Query tool	83
16	MDM Start Match Review Workflow tool.....	84
	MDM Start Match Review Workflow tool parameters	84
	Configure the MDM Start Match Review Workflow tool	85
17	MDM Diff Report tool.....	86
	MDM Diff Report tool parameters	86
	Configure the MDM Diff Report tool	87
Part VII	Reference	87
1	Databases and data models.....	87
	Table options	88
	Field options	89
	Field data types	90
	Versioned and time-series tables	91
	Versioned tables.....	91
	Time series tables.....	92
2	Nested tables and linkage fields.....	92
	Nested tables	92
	Add a nested field to a table.....	93
	Linkage fields	94
	Add a linkage field to a table.....	95
3	Historical data model.....	95
	Changesets	95
	Example: metadata changeset (create a table).....	96
	Example: data changeset (add a record to a table).....	96
	Conflict resolution.....	97
	Viewing changesets.....	97
	Change history	99
	Data view	99
	View current data.....	100
	View historical data.....	100
	Changeset data.....	101
	Open a new changeset with the data view link.....	102
	Switch back to an open changeset.....	104
	Versioned metadata	104
4	Integrating RedPoint Data Management and MDM.....	105
	Settings and credentials	105
	Select databases and tables	106
	Query records	106
	Query records matching input data	106
	Insert or replace records	106
	Query views	107
	Extract historical data	107
	Shared changesets	107
	Detect and process duplicates during load into MDM	107

Advanced matching	108
Support for different match types.....	108
Match group IDs and match table primary keys.....	108
Match-segmentation key fields.....	108
Match information document.....	109
Handling match suppression.....	109
Automatic matching.....	109
Batch matching.....	110
Real-time matching.....	110
Match review is conducted via workflow.....	110
5 Views and subscriptions.....	111
Defining a view	112
Types of projections.....	112
View examples.....	113
Simple table view.....	113
Aggregating data from several tables to a view of transactions by customer.....	115
Subscribing to a view	118
Create a subscription.....	118
Querying MDM view data with a DM connector	120
6 Advanced workflow.....	120
Standardized workflows	121
Workflow roadmap	121
7 GDPR and privacy support.....	122
Data visibility	122
Masking erases sensitive data fields	122
Erasing (purging) permanently deletes records	122
Masking and purging create hash entries	122
Choose masking or purging based on customer needs	123
Masking and purging	123
Identifying PII in table schemas.....	124
Hashing.....	124
Record history for masking and purging.....	125
Configure a table for masking and purging.....	125
Mask or purge a record.....	126
Verify a record was masked or purged.....	127
8 Security.....	129
Users and groups	129
Special users and groups.....	130
User operations.....	131
Browse users.....	131
Create a user.....	132
Edit a user.....	133
Delete a user.....	135
Group operations.....	135
Browse groups	136
Create a group.....	136
Edit a group.....	137
Delete a group.....	139
Group roles	139
Permissions and entities	141
Permissions inheritance.....	142

How permissions are interpreted for each entity.....	146
System-only permissions.....	149
Synchronization and versions.....	149
Permissions and groups.....	149
Permission entries.....	149
Most-permissive logic.....	150
Change entity permissions.....	150
Adding and deleting permission entry groups.....	151
9 Matching sub-document.....	153
The sub-document	154
10 Aggregation language.....	155
Expression syntax	156
Special identifiers	156
Backquoted strings	157
Expression types	157
Scalar versus aggregating usage.....	158
Aggregators	159
\$Sum with optional type.....	159
\$Min and \$Max.....	159
\$Min and \$Max with type.....	160
\$Count.....	161
\$Average.....	161
\$First	161
\$First with type.....	162
\$Nth	163
\$Nth with type.....	163
Selectors	164
\$All	165
\$Filter	165
\$AgeDays.....	165
\$Sort	166
\$Top	166
\$Unique.....	166
\$MostCommon.....	166
Temporary variables	168
Computed fields	168
Java library support	168
Support functions for expressions	169
\$Compare.....	170
\$Equals.....	170
\$StrCompare.....	170
\$StrEquals.....	171
\$DaysDiff.....	171
\$ToBigDecimal.....	171
\$.ToBoolean.....	171
\$ToDate.....	172
\$ToDateTime.....	172
\$.ToDouble.....	172
\$ToFloat.....	172
\$ToInteger.....	173
\$ToLong.....	173
\$ToString.....	173

\$ToTime.....	174
Aggregation expression notes	174
Rules and conventions.....	174
Java language considerations.....	175
MDM type to Java type mappings.....	175
Null values and Date/DateTime fields.....	176
Selecting multiple fields as a group.....	176
Multiple aggregators in an expression.....	177
Making decisions.....	177
Complex ranking.....	178
11 User configurable properties.....	178
MDM core services	178
MDM core services options.....	178
MDM core services properties file.....	179
Sample file.....	179
File format.....	180
Properties for unit testing.....	184
MDM UI configuration file	185
Properties list.....	185
Connection to Cosmos DB.....	187
Usage and internals.....	188
Index	189

¹ RedPoint MDM overview

RedPoint Master Data Management (MDM) is a data stewardship system for managing the *Golden Record*: a single, accurate, and always up-to-date view of your most critical data. The MDM system includes a variety of capabilities for controlling your data:

Data model management

Database administrators (*DBAs*) can create and manage data models, including:

- Tables (including versioned, reference, and time-series types)
- Schemas
- Linkages
- Indexes
- Personally Identifiable Information (PII) designations
- Masking rules

Data models may evolve over time so long as changes do not invalidate existing data.

You can find more information on the MDM data models [here](#)^[29].

Data versioning

All create, read, update, and delete (CRUD) changes to versioned tables are retained in history, and time-series tables are append-only. This approach ensures that information is never lost (unless purged for privacy compliance reasons), and historical search and comparison is always available. Queries may reference current data or historical data at any point in time, facilitating historical comparison and trending reports. Users of MDM can drill down into individual record history.

Changesets

All data and metadata changes are made via a *changeset* (explicit commit step). This ensures that all changes occur in a coordinated and linear fashion.

Conflict resolution

If simultaneous changes to the same data are attempted, the conflict resolution system will merge compatible changes or flag incompatible changes for review.

Matching

The MDM system supports a rich matching model:

- Users can create RedPoint Data Management projects to perform fuzzy matching on MDM records, either during load or after load.
- Multiple match levels (for example, individual/household) are supported.

- Automatic matches can be reviewed by data stewards using the workflow system.
- If a data steward overrides automatic match logic, these decisions are retained to prevent auto-matching from making the same mistake twice.

Workflow

MDM supports a cooperative workflow model in which users can coordinate and review data changes. For example:

- Data changes are submitted by business users but reviewed by data stewards before commit.
- Borderline automatic match results are reviewed by data stewards and accepted or rejected.
- A manual match override can be suggested by one user and reviewed by another.

ETL connectors

RedPoint Data Management is a powerful extract, transform, and load (ETL) tool that connects directly to the MDM system for the purpose of loading, querying, extracting, and matching. These connectors work in web services to support real-time transactions.

Privacy compliance

MDM supports compliance with privacy rules such as the EU's General Data Protection Regulation (GDPR) by allowing personally identifiable information (PII) to be masked or purged from the system. Such actions reach back into history and remove all traces of PII, so successful erasure can be verified. In addition, masking or purging data prevents re-entry of the same data.

Computed views and subscriptions

MDM supports a computed-view model that combines table data, joins, and aggregation rules. This is ideal for:

- Golden record creation
- Marketing and statistical variables
- Lifetime-value calculations

A view may be queried directly, and such queries may reference historical or current data. A view may also be *subscribed* to, in which case changes to underlying table data trigger real-time updates of the view results, and incremental delivery of the changes. In addition, view aggregation rules may contain date-sensitive filters (for example, transactions in the last 30 days), the results of which are automatically updated nightly.

Web service APIs

For integration with other systems, all of MDM's features are accessible via its web service APIs.

Scripting

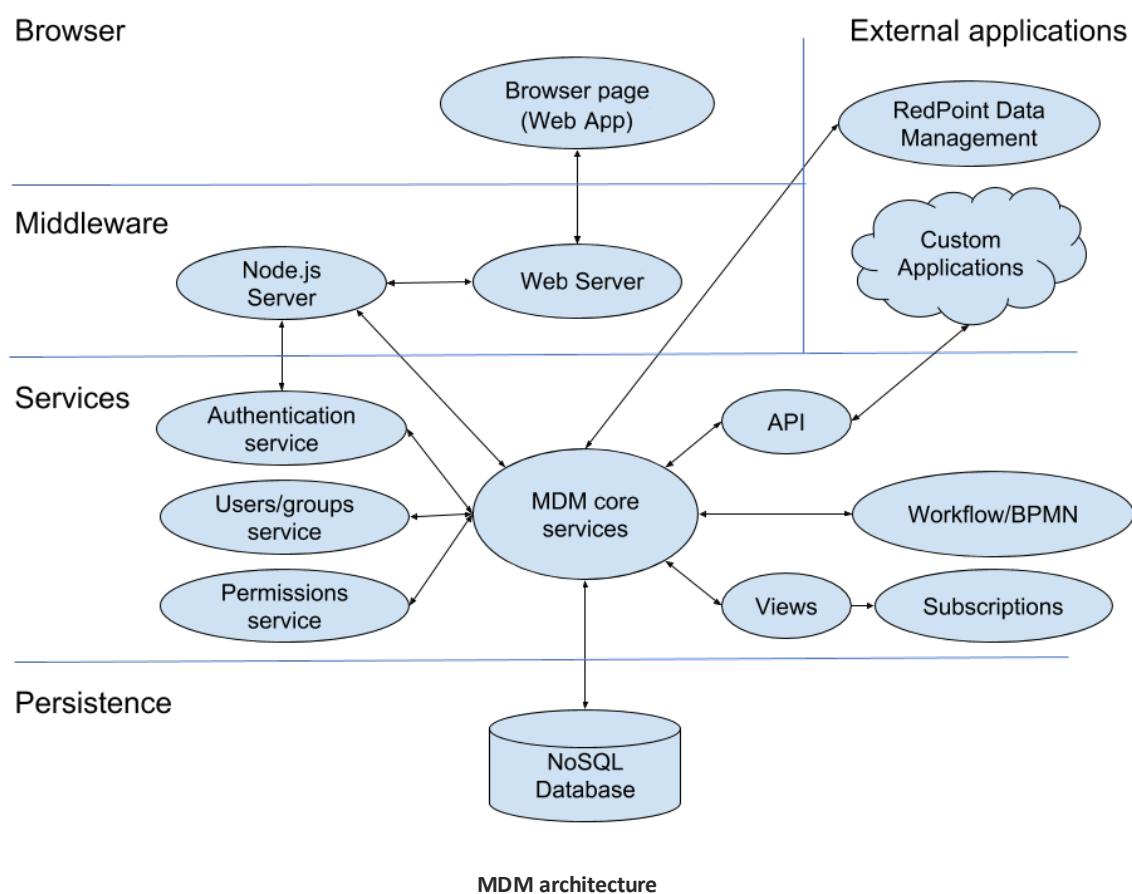
A scripting tool is provided to support typical provisioning steps:

- Creation of data models
- Import of reference data
- Setting up users, groups, and permissions

1.1

How it works

MDM consists of these components:



Browser

The MDM Web App is the JavaScript-based client application for MDM.

External applications

RedPoint Data Management ingests, cleanses, transforms, and integrates data. A RedPoint Data Management ETL process automatically identifies potential duplicate records and compares them in both "loose" and "tight" match passes. If these produce different results, the process initiates a MDM review workflow so that a steward can choose the preferred match.

Middleware

- The web server handles requests from the Web App by sending requests to and accepting data from the Node.js server.
- The Node.js server runs server-side JavaScript to produce dynamic web page content before the page is sent to the web app. It also sends requests to and accepts data from the Services layer.

Services

MDM core services are the heart of MDM. They accept and process requests from the Web App, query the computational subservices, and access a NoSQL database.

- The **Authentication** subservice authenticates MDM users. A user who successfully logs into MDM is issued a security token, which must be submitted along with any request. This service also authenticates a user's security token at every system request.
- The **Users/groups** subservice keeps track of MDM users and the access groups to which they belong.
- The **Permissions** subservice keeps track of user permissions.
- **Workflows** are constructed from Business Process Model and Notation (BPMN) models. This module contains the logic for processing workflow steps.
- A **View** is a directed graph of data transformations (or *projections*) that supports complex queries including joins and aggregations. You can also create **Subscriptions** to a view, which notifies you when the data in a view changes (that is, when records are added, deleted or modified). When views and subscriptions change, these services push the changes out to the subscribers via the chosen method (currently only web-service-call delivery is supported).
- All services are directly accessible via the MDM API.

Persistence

MDM is backed by a NoSQL document database.

1.2

Getting help

We want to make your experience with MDM as pleasant and productive as possible. Please email or call us when you need help. See [Troubleshooting](#)^[12] for a list of information we'll need in order to help you.

Contact us at:

RedPoint Global Inc.
36 Washington Street, Suite 120
Wellesley Hills, MA 02481

Phone: +1 781 725 0250

Fax: +1 781 235 3739
Email: support@redpointglobal.com

1.2.1 Troubleshooting

When you [contact us about a problem](#), please include the following information:

- Product serial number and version number (view this by choosing **About** from the **Help** menu).
- Your name, company name, telephone number, and email address.
- Operating system and version number.
- Exact wording of any error messages that appear.
- Exactly what you were doing when the problem or error occurred.
- How you tried to resolve the problem.

We may also ask you for:

- A sample of the input data.
- Your system configuration.

2 Installation and configuration

An MDM installation has the following major components:

- MongoDB Server
- MDM Core Services
- MDM Web App

The MDM services, both Core and App, run as Docker containers. **Docker must be installed on the MDM server.**

2.1 System requirements

MongoDB instance

For data < 1 million records total:

- 4 GB memory.
- Any storage medium.

For data 1-10 million records total:

- SSD storage.
- 4-16 GB memory.

For data > 10 million records total, or very wide records:

- Must size MongoDB memory to keep working index set in memory. This will depend on the number of records and the number and size of indexed fields.
- Consult RedPoint Global Inc. support or services.

MDM Core Services

For data < 1 million records total:

- 4 GB memory assigned to JVM*.
- 10 GB available temporary disk space**.

For data 1-10 million records total:

- 8 GB memory assigned to JVM.
- 10 GB available temporary disk space.

For data > 10 million records total:

- 8-16 GB assigned to JVM is usually sufficient.
- More memory may be needed depending on the level of activity and the complexity of computed views. Consult RedPoint Global Inc. support or services.
- 20 GB available temporary disk space.

*In other words, start Java with (for example) `-Xmx8gb`. The system or Docker image memory should be 1 GB larger.

**The user running the MDM services has a default temp folder. This is what will need the space. On Linux this is generally `/tmp` and on Windows this is generally `C:/users/username/temp`.

MDM Web App

2 GB memory.

Docker

- Windows—Windows 10 or Windows Server 2019.
- Linux—no specific operating system requirements

2.2

Installation prerequisites

Because the MDM architecture relies on several different services, installing MDM requires some planning.

Obtain installation files and scripts

Contact RedPoint Global Inc. support to obtain the following files necessary to install MDM:

```
mongod.cfg  
start_mdm_containers.bat (for Windows installations)  
start_mdm_containers.sh (for Linux installations)  
rpmdm-java.properties  
log4jsconfig.json  
mdm-connectors-*-.shaded.jar
```

Copy these files to a folder on your target installation machine.

Install the RedPoint Data Management client and server

The RedPoint Data Management client and server must already be installed. See the *Installation and configuration* section of the *RedPoint Data Management User Guide* for installation instructions.

2.3

Install on Windows

Set the MongoDB password and Docker internal IP address

The files `start_mdm_containers.bat` and `rpmdm-java.properties` need to be modified to set the proper MongoDB password. This password will be used later in the installation, so it is easiest to modify the files now.

1. Edit `start_mdm_containers.bat`

For the `MDM_MONG_URI` value, change `yourmongopassword` to your chosen password:

```
MDM_MONGO_URI="mongodb://admin:yourmongopassword@host.docker.internal:27017"
```

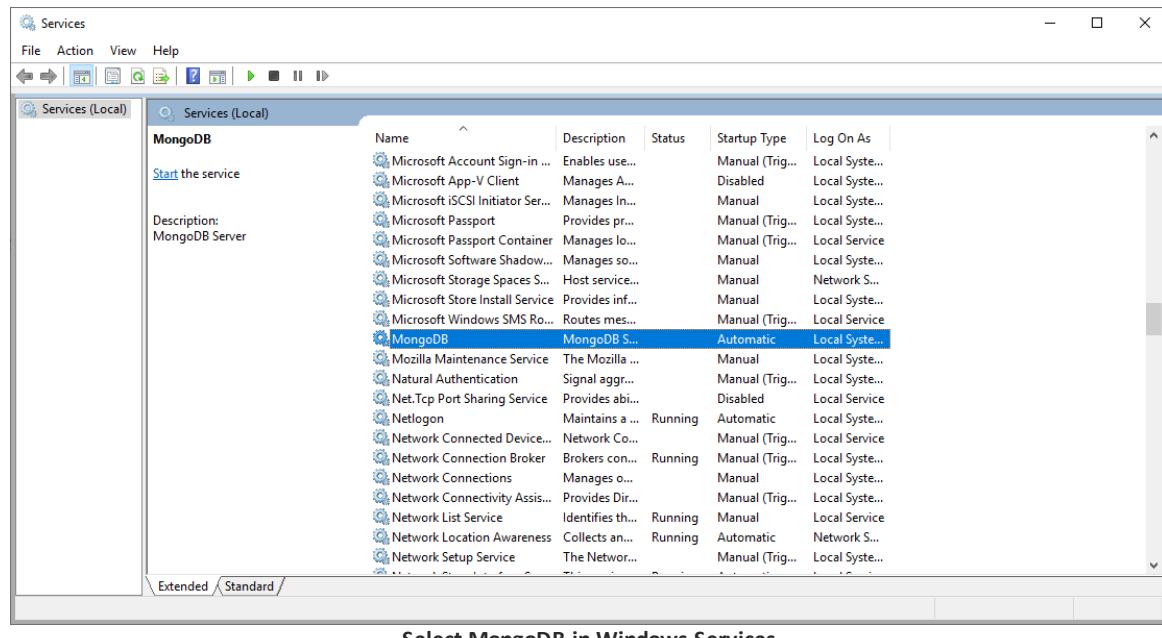
2. Edit `rpmdm-java.properties`:

For the `rpmdm.mongodb.uri` value, change `yourmongopassword` to your chosen password:

```
rpmdm.mongodb.uri=mongodb://admin:yourmongopassword@host.docker.internal:27017
```

Install MongoDB

1. Download the MongoDB server from <https://www.mongodb.com/download-center/community> (release 3.6, Windows MSI installer, 64-bit).
2. Execute the Windows installer file.
3. Follow the instructions in the setup wizard.
4. Accept the terms of the license agreement.
5. Choose the **Complete** setup type and run the installer.
6. Create a folder for the MongoDB database (for example, `c:\mongodb\db`).
7. Create a folder for logs (for example, `c:\mongodb\log`).
8. The `mongod.cfg` file is one of the RedPoint-provided files you previously copied to a directory on your installation machine.
 - a. From this local directory, move `mongod.cfg` to your MongoDB install directory (for example, `C:\Program Files\MongoDB\Server\3.6\bin`).
 - b. Edit `mongod.cfg` and replace `f:\\\\mongodb` with your MongoDB installation path. You will need to change this path in two places:
 - For the `dbPath` value, specify the correct database path.
 - In the `systemLog` section, specify the correct MongoDB log path.
 - c. Note that `mongod.cfg` is in the YAML format, and spaces are significant. For example, the lines after `storage` and `systemLog` must be indented two spaces (and only two spaces). The provided `mongod.cfg` file is properly formatted as-is.
9. Install MongoDB as a Windows service with the following command:
`mongod --config "mongodb-install-dir\\mongod.cfg" --install`
10. Open the Windows Services app (on the Search bar, enter `services` and then select the Services app).
11. Confirm that the MongoDB service exists in Windows Services and start it (select the **MongoDB** service, right-click, and select **Start** on the context menu).



Select MongoDB in Windows Services

- From the command line, enter:

```
mongo
```

- When the > prompt is displayed, enter on the command line:

```
use admin
db.createUser( { user: "admin", pwd: "yourmongopassword", roles: [ "root" ] } );
```

- Close the command line window (enter `exit`).

- Stop the **MongoDB** service (select the **MongoDB** service, right-click, and select **Stop** on the context menu).

- Edit the `mongod.cfg` file. Uncomment the following two lines:

```
#security:
# authorization: enabled
```

and save the file.

- Restart the **MongoDB** service.

Note

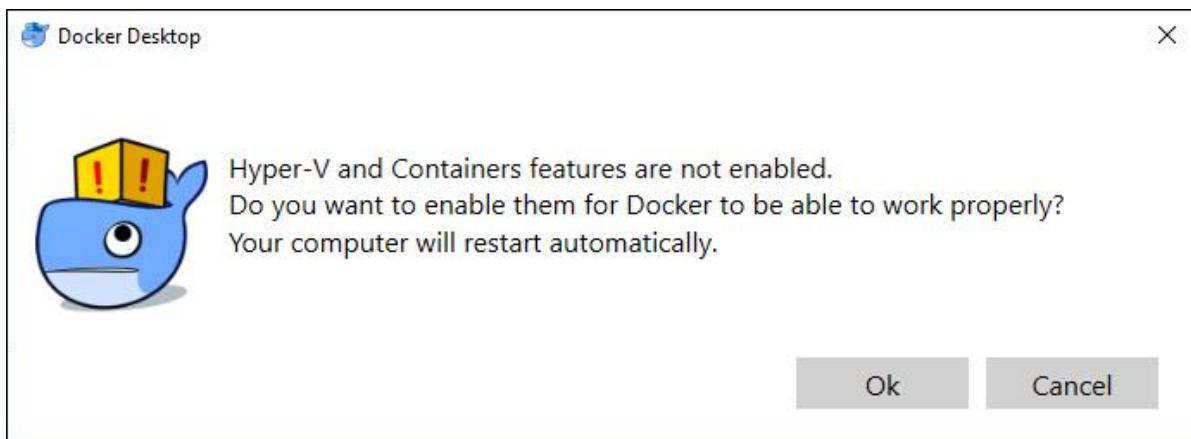
If commands are not found when executing MongoDB commands from the command line, ensure that the MongoDB install path (for example, `C:\Program Files\MongoDB\Server\3.6\bin`) has been added to the PATH environment variable. If it has not, add it and then restart your command line.

Install Docker

1. If you have not already created a Docker account, go to <https://www.docker.com/products/docker-desktop> and do so.
 - a. Give your Docker account ID to your RedPoint Global Inc. support representative so that we can give you access to the MDM Docker images later in the installation.
 - b. Save your Docker account ID and password—you will need that information later in the installation.
2. Download the latest Docker Desktop release from <https://www.docker.com/products/docker-desktop>.
3. Run the Docker Desktop installer. After installation, your machine will restart.
4. Open the **Docker Desktop for Windows** menu by right-clicking the Docker icon in the Notifications area (or System tray).
5. Select **Settings...** to open the **Settings** dialog.
6. On the right-hand menu, click the **Advanced** tab.
7. Move the **Memory** slider to at least 8192 MB (8 GB in binary).

Note

When starting Docker for the first time, the following message is displayed:



Message displayed when starting Docker for the first time

You must click **OK** for Docker to work properly.

Run shell script to start MDM inside local Docker containers

1. Make sure you are logged in to the Docker account you created when downloading Docker Desktop. This will allow the following batch file to access the MDM Docker image.

2. In your shell of choice, execute the RedPoint-provided batch file:

```
start_mdm_containers.bat
```

MDM should now be executing in local Docker containers.

2.4

Install on Linux

Set the MongoDB password and Docker internal IP address

The files `start_mdm_containers.sh` and `rpmdm-java.properties` need to be modified to set the proper MongoDB password. This password will be used later in the installation, so it is easiest to modify the files now.

1. Edit `start_mdm_containers.sh`:

- For the `MDM_MONG_URI` value, change `yourmongopassword` to your chosen password:
`MDM_MONGO_URI="mongodb://admin:yourmongopassword@host.docker.internal:27017"`
- For the `LOCAL_IP` value, change `LOCAL IP OF MACHINE` to the IP address of the MDM server.

2. Edit `rpmdm-java.properties`:

- For the `rpmdm.mongodb.uri` value, change `yourmongopassword` to your chosen password:
`rpmdm.mongodb.uri=mongodb://admin:yourmongopassword@host.docker.internal:27017`

Install MongoDB

1. [Download the MongoDB Community Server](#), selecting the proper Linux operating system.
2. Install MongoDB, following the [installation instructions for Linux](#).
3. If you installed via the package manager, the data directory `/var/lib/mongodb` and the log directory `/var/log/mongodb` were created during the installation. To modify the location of these folders, you must edit the MongoDB configuration file `/etc/mongd.conf`:

```
# Where and how to store data.  
storage:  
  dbPath: /var/lib/mongodb/db  
  
# where to write logging data.  
systemLog:  
  path: /var/log/mongodb/mongodb.log
```

4. Confirm that the `mongod` service has been started: `sudo service mongod status`
5. From the command line, enter: `mongo`
6. To create the Admin user required by MDM, enter from the MongoDB command line:

- ```
use admin
db.createUser({ user: "admin", pwd: "yourmongopassword", roles: ["root"] });
```
7. Type `exit` to exit the MongoDB shell.
  8. Stop the **mongod** service.
  9. Edit the configuration file `/etc/mongod.conf`, uncommenting the following two lines:

```
#security:
#authorization: enabled
```
  10. Save the configuration file and restart the **mongod** service.

### Install Docker

1. Install Docker Community Edition for [Centos](#) or [Ubuntu](#).
2. Once the installation is complete, [create a Docker hub account](#).
3. Give your Docker account ID to your RedPoint Global Inc. support representative so that we can give you access to the MDM Docker images later in the installation.
4. Save your Docker account ID and password—you will need that information later in the installation.

### Run shell script to start MDM inside local Docker containers

1. Log into the Docker account you created after installing Docker, and ensure that RedPoint Global Inc. support team has added your Docker ID to the RedPoint Global Inc. Docker repository.
2. Execute the following script to start the MDM containers:

```
start_mdm_containers.sh
```

MDM should now be executing in local Docker containers.

2.5

## Helpful Docker commands

The following Docker commands are useful for troubleshooting:

| Command                                      | Meaning                       |
|----------------------------------------------|-------------------------------|
| <code>docker ps [OPTIONS]</code>             | List containers               |
| <code>docker logs [OPTIONS] CONTAINER</code> | Fetch the logs of a container |
| <code>docker stop [OPTIONS] CONTAINER</code> | Stop a container              |

2.6

## MDM scripting tool

The MDM scripting tool makes it easy to set up a demo database. The tool also supports the automation of initial setup processes and ongoing interaction with the MDM database. The tool uses a JavaScript interface to interact with MDM. The MDM scripting tool is installed from the Node.js repository.

### Install Node.js

Node.js is needed by **mdmscript**.

1. [Download the latest Node.js release](#), selecting the latest long term support (LTS) version for your operating system.
2. Follow instructions for installing Node.js for your operating system.

### Get started with a demo database

Once Node.js is installed, you can access the **mdmscript** tool which is used to create a database with all required users, groups, and roles, populated with demo data.

1. In your shell of choice, execute the following command to download the **mdmscript** command-line executable:

```
npm install rp-mdm-script -g
```

This command installs **mdmscript** to your system from the node.js repository, making it available from the command line.

2. In your shell, execute the following commands:

```
mdmscript --scriptfile=make_system.js (installs default users, groups, and roles)
```

```
mdmscript --scriptfile=make_mydb.js (creates a demo database)
```

3. Access MDM via the web app at the URL <http://hostname:9090/mdm>.

### Install the MDM tools in RedPoint Data Management

MDM uses a set of RedPoint Data Management tools to perform "back end" ETL and matching functions. These tools require a license and are intended for use with RedPoint Master Data Management.

For a more general explanation of RedPoint Data Management tools, refer to the *RedPoint Data Management User Guide*.

1. Copy the RedPoint-provided "shaded" version of the MDM tools JAR file into the **\java\_plugins** folder in the RedPoint Data Management installation folder. The filename is of the form **mdm-connectors-\*-.shaded.jar**.

2. In the RedPoint Data Management client, click the **Palette** menu button at the top of the tool palette  and select **Reset Palette**.
3. The MDM tools will appear in the **Master Data** tool group.
4. Click the **Repository** tab and go to **Settings > Tools**.
5. Click the **MDM** tab and enter your server and authentication credentials.
6. Close and re-open any projects that were active before installing the MDM tools JAR file.

2.7

## Configuring services

The following sections explain how to configure MDM's multiple services.

2.7.1

### Port assignments

MDM services use the following default port assignments:

| MDM service            | Port number |
|------------------------|-------------|
| Web application        | 9090        |
| Authentication service | 9901        |
| User/group service     | 9903        |
| Permissions service    | 9908        |
| MDM core services      | 9902        |

However, these port numbers are configurable. For information on changing these port numbers, refer to [MDM UI configuration file](#)<sup>185</sup> and [MDM core services properties file](#)<sup>179</sup>.

MDM uses a user-configurable MongoDB connection string to attach to the instance, cluster, or replica set. Consult MongoDB documentation and your system administrator for specifics on your particular database configuration. The default port for a simple MongoDB instance is 27017.

2.7.2

### Services access

In general:

- The user's web browser needs access to only the MDM web application port.
- RedPoint Data Management services need access to all MDM services, and MDM services need access to each other (for example, the web service calling the user authentication or permissions services).
- The MDM web app and services must all have access to MongoDB, and we recommend that the MDM web app and services be deployed on the same virtual machine. Contact RedPoint Global Inc. for assistance in determining the best deployment on your infrastructure.

### 2.7.3 Configuring SSL for MDM services

By default, MDM services are configured to run without Secure Sockets Layer (SSL) so you can validate that the MDM processes are working. After validating that MDM processes are working correctly, you should set them to use HTTPS. HTTPS can be enabled independently for each service.

#### To configure the node services

You will need to modify the following source files:

- `mdm-ui/mdm-ui/Middleware/config/https.js`
- `redpoint_default.js`
- `mdm-ui/Middleware/config/permission.js`

Note that you cannot modify the user configuration file `c:\Program Files\RedPointMDM\redpoint_default.js` because of a TypeScript bug that prevents using files outside of the root directory of the project.

#### 1. Modify `mdm-ui/redpoint_default.js`

You must replace the `mdm`, `authentication`, `user`, and `rpmdm` blocks to include `host`, `port`, and `protocol` values:

```
"mdm": {
 "host": "www.mydomain.net",
 "port": 9090,
 "protocol": "https"
},
"authentication": {
 "host": "www.mydomain.net",
 "port": 9901,
 "protocol": "https",
 ...
 // system and group configuration here
},
"user": {
 "host": "www.mydomain.net",
 "port": 9903,
 "protocol": "https"
},
"rpmdm": {
 "host": "www.mydomain.net",
 "port": 9902,
 "protocol": "https"
},
```

## 2. Modify mdm-ui/Middleware/config/https.js

Specify the key and certificate files, as well as the password used to encrypt those files (if any). The path can be absolute (recommended) or relative.

```
module.exports = {
 // SSL certificate information
 'secret': 'yoursecretkeyhere',
 "key_file": "../ssl-test-certs/mycertificate.key",
 "crt_file": "../ssl-test-certs/mycertificate.crt"
}
```

## 3. Modify mdm-ui/Middleware/config/permission.js

Update the `host`, `port`, and `protocol` keys:

```
module.exports = {
 base_url: '/mdm/permission/',
 "host": "svc.mydomain.net",
 "port": 9908,
 "protocol": "https"
};
```

To configure the core services

### 1. Override node service URLs

In the `redpointmdm.properties` files, override each of the node service URLs and specify values for:

- core service host
- port
- protocol
- certificate

### 2. Choose REST service URLs

In order for the Core service to successfully call the three other REST services, you must fully specify the other REST services' base URIs:

- protocol—`http` (the default value) or `https`.
- host—The machine name associated with the service (the default value is `rp-mdm-ui`, for example `http://rp-mdm-ui:9901/mdm`).
- port—Unless the service is running on the default port (80/443).

- path—Always /mdm

### 3. Note core service settings

- The default value of host is 0.0.0.0
- If secure is true, the core service uses a JKS certificate file.

### 4. Set core service SSL configuration

```
Calling other services
rpmdm.authentication.url=https://svc.mymdmhost.net:9901/mdm
rpmdm.usergroup.url=https://svc.mymdmhost.net:9903/mdm
rpmdm.permission.url=https://perm.mymdmhost.net:9908/mdm

#My service
rpmdm.service.host=svc.mymdmhost.net
rpmdm.service.port=9902
rpmdm.service.secure=true
rpmdm.service.ssl_jks_file=rphelios.net.jks
rpmdm.service.ssl_jks_password=2shortAPwd
```

### 5. Create a JKS certificate file

The Core service currently only supports using the Java default certificate format, JKS. You will need access to the command-line applications `openssl` (a Linux tool) and `keytool` (part of the Java JDK).

Starting from .cert and .key files

`certfile.crt`, `certkey.key`, and `pfxfile.pfx` are variables that represent the actual names of your files:

```
openssl pkcs12 -export -in certfile.crt -inkey certkey.key -certfile certfile.crt -out pfxfile.pfx
```

You will need to choose (and remember) a password for the newly-created `pfx` file.

Starting from a PFX file

You will need to know the password used to encrypt the `PFX` file in order to do anything with it:

```
keytool -v -list -storetype pkcs12 -keystore pfxfile.pfx
```

Enter the `PFX` file password. Use this command to find the alias (the line starting with `Alias name:`):

```
keytool -importkeystore -srckeystore pfxfile.pfx -srcstoretype pkcs12 -srcalias "alias" \
 -deststoretype jks -deststorepass "a-password-you-choose" -destalias alias-you-choose
```

We recommend choosing the same password for both the `JKS` and `PFX` files, and using the alias "mdmcert".

## 6. Obtain CERT and KEY files

If you have a **PFX** file, you will need **.crt** and **.key** files to run the Node services securely.

You will also need a machine with **openssl** installed (typically, this is a Mac or a Linux box). You might be able to install it via third-party Linux tools on a Windows machine.

For the **.crt** file

a) Run **openssl**. The prompt **OpenSSL>** is displayed.

b) At the prompt, enter:

```
pkcs12 -in pfxfile.pfx -out certificate-tmp.txt -nodes
```

c) Enter the password.

d) To exit **openssl**, enter:

```
exit
```

e) In an editor, open the file **certificate-tmp.txt**.

f) Copy the file contents from **-----BEGIN CERTIFICATE-----** to **-----END CERTIFICATE-----**.

g) Save as the filename **certificate.crt**.

For the **.key** file

a) In the **certificate-tmp.txt** file, copy the section from **-----BEGIN PRIVATE KEY-----** to **-----END PRIVATE KEY-----**.

b) Save as the filename **certificate.key**.

c) Delete the file **certificate-tmp.txt**.

d) Remove the **.txt** extension from the filename if your editor appends it.

This document may be useful:

<https://helpcenter.gsx.com/hc/en-us/articles/115015887447-Extracting-Certificate-crt-and-PrivateKey-key-from-a-Certificate-pfx-File>

3

## The Web App

The Web App (or MDM client) is the stewardship interface for MDM. While all MDM functionality is available through the MDM API, the MDM app is the primary means of:

- Viewing records at any point in the database history
- Adding, modifying, and deleting records (for data stewards and DBAs—for business users, the primary mechanism will be a workflow)
- Creating, modifying, and deleting databases and tables

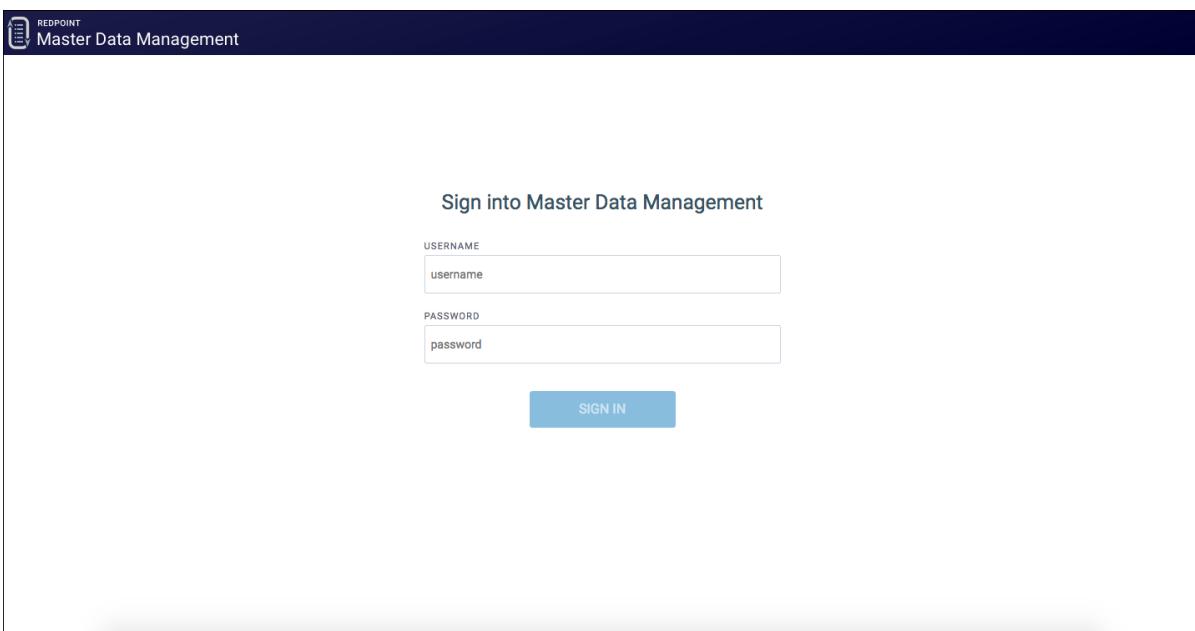
- Reviewing records
- Masking and erasing records
- Confirming that specific records have been erased

3.1

## Log on to RedPoint MDM

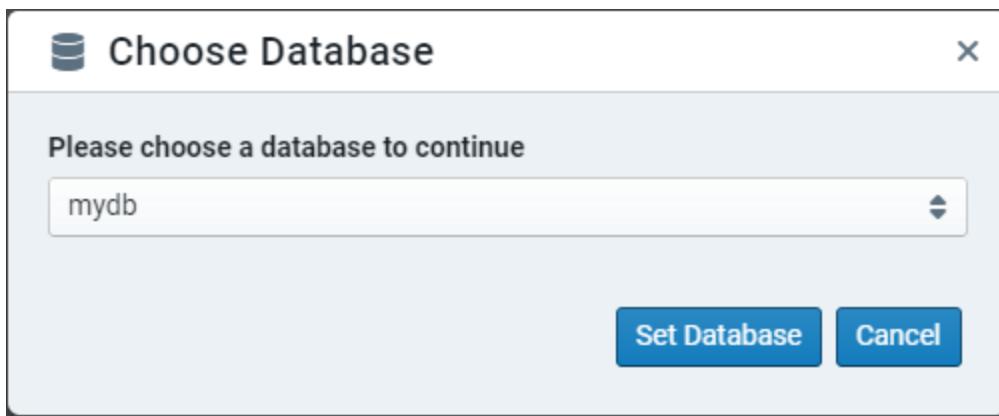
### To log on to the MDM Web App

1. In a browser, enter the URL for your MDM Web App. (By default, the URL is *server/mdm:9090*, where *server* will vary depending on the machine/network.)



MDM logon page

2. Enter your **Username** and **Password**, and click **Sign In**.  
Note that during the initial configuration of MDM, you will log on as **system**.
3. When you log on to MDM for the first time, the **Choose Database** dialog is displayed:

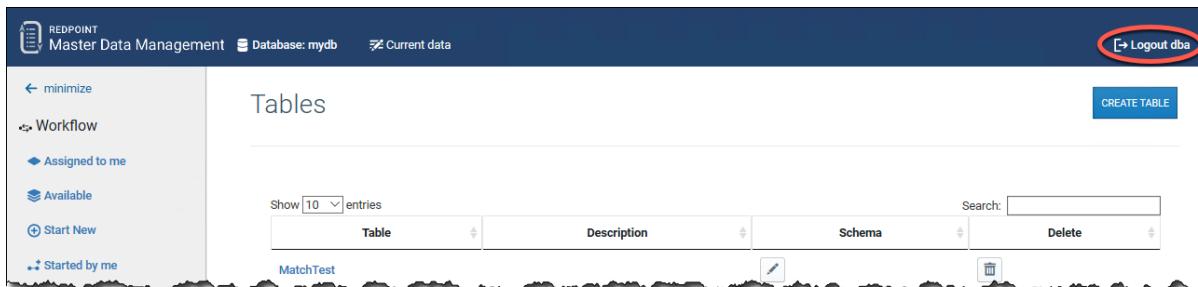


Choose database dialog

### 3.2 Log off RedPoint MDM

To log off the MDM Web App

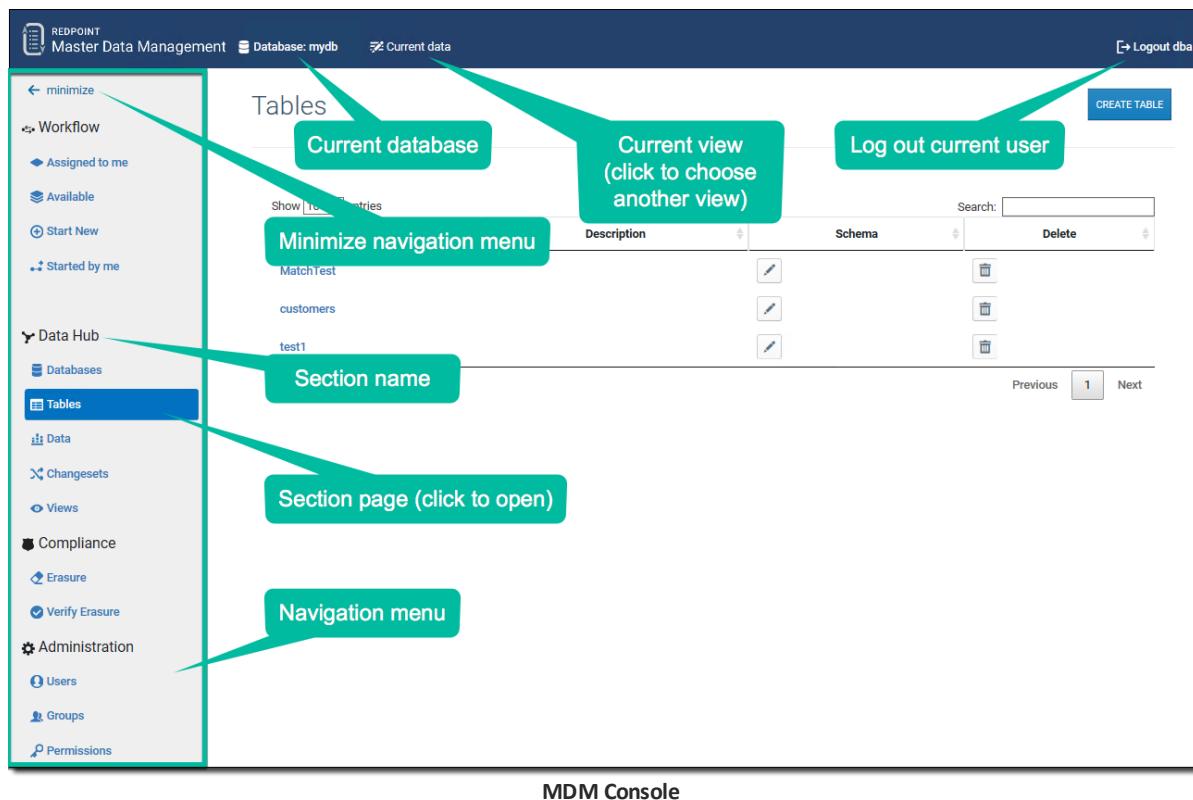
- On the upper right corner of any Web App page, click **Logout**.



MDM Console logout link

### 3.3 Console overview

This screen shot describes the components of the MDM Web App:



### 3.3.1 Navigation menu

The main content areas of the MDM Web App are:

#### Workflow

The concept of **workflow** is central to the MDM process. A workflow is a process for performing a series of tasks on the master data, often requiring actions by multiple users. Workflows are bound to databases. The available actions in a given workflow are determined by the permissions of the group of which the user is a member. For example, when a user who is a member of the **Bizuser** group clicks **Start New** in the **Workflow** section, she may choose to create either a **New Customer** or an **Edit Customer** workflow item (or *workitem*). When that workitem is submitted, it becomes available for review by a user who is a member of the **Steward** group. There are different kinds of workflows such as creating a record, updating a record, reviewing proposed data changes, and reviewing potential record matches.

You can find more information on the MDM workflow [here](#).

#### Data Hub

The Data Hub section allows interactive access to databases and tables. It allows you to:

- Select or change the current database
- Search for and query tables
- View a record's history

- Directly update data (records)
- Change metadata, including:
  - Creating or deleting a database
  - Creating, deleting, or modifying tables

You can find more information on the Data Hub section [here](#)<sup>[29]</sup>.

## Compliance

This section contains Masking and Erasure functions as well as a Verify Erasure function that lets you confirm whether a record was masked or purged. These features are only available for tables that have been configured with one or more PII columns and masking hashes.

You can find more information on the Compliance section under [GDPR and privacy support](#)<sup>[122]</sup>.

## Administration

This section lets you view and configure:

- Who can access MDM
- What users can access in the Web App
- What each user can do with each MDM entity

You can find more information on the Administration under [Security](#)<sup>[129]</sup>.

4

# The data hub

The **Data Hub** section allows interactive access to databases and tables. It allows you to:

- Select/change the current database
- Search for and query tables
- View a record's history
- Directly update data (records)
- Change metadata, including:
  - Creating or deleting a database
  - Creating, deleting, or modifying tables

#### 4.1 Select/change the current database

To select or change the current database

- In the **Data Hub** section of the MDM Web App, click **Databases**. All the databases for the current MDM instance are displayed. **Note:** In many organizations, users will work within a single database.

The screenshot shows the 'Databases' page in the RedPoint Master Data Management web application. The top navigation bar includes 'REDPOINT Master Data Management', 'Database: mydb', 'Current data', 'Logout dba', and a 'CREATE DATABASE' button. On the left, a sidebar lists 'minimize', 'Workflow', 'Assigned to me', 'Available', 'Start New', and 'Started by me'. The main area is titled 'Databases' and shows a table with columns: Database, Description, Create User, and Created. Two databases are listed: 'mydb' (selected) and 'HealthCare'. A red box highlights the 'mydb' row, labeled 'Currently-selected database'. Another red box highlights the 'Search' input field, labeled 'Search for a database by name'. A third red box highlights the 'Show 10 entries' dropdown, labeled 'Number of databases shown per page'. A fourth red box highlights the 'Previous' and 'Next' navigation buttons at the bottom right, labeled 'Move between database pages'. A fifth red box highlights the 'Delete' icons for each database entry. Below the table, a red box highlights the entire page area with the label 'Selects or changes the current database'.

If there are more databases than can be listed on the page:

- You can use the **Previous** and **Next** buttons to move between database pages.
- You can control the number of databases listed per page by changing the **Show N entries** option.

You can also search for a database directly by clicking inside the **Search** box and entering the name of the database that you want to find.

- Click on the database name.

#### 4.2

## Search for a table

Before you search for a table, make sure to [select the correct database](#) [30].

#### 4.2.1

### From the tables page

To search for a table from the **Tables** page

- In the **Data Hub** section of the MDM Web App, click **Tables**. The tables for the selected database are displayed.

The screenshot shows the 'Tables' page of the RedPoint Master Data Management application. On the left, there's a sidebar with navigation links like 'minimize', 'Workflow', 'Assigned to me', 'Available', 'Start New', 'Started by me', 'Data Hub', 'Databases', and 'Tables'. The 'Tables' link is highlighted. The main area has a header 'Tables' and a search bar 'Search for a table by name'. Below that is a dropdown 'Show 10 entries' and a table with columns 'Table', 'Description', 'Schema', and 'Delete'. The table contains three rows: 'MatchTest', 'customers', and 'test1', each with edit and delete icons. At the bottom, it says 'Showing 1 to 3 of 3 entries' and has 'Previous', '1', and 'Next' buttons.

Searches for a table from the Tables page

If there are more tables than can be listed on the page:

- You can use the **Previous** and **Next** buttons to move between table pages.
- You can control the number of tables listed per page by changing the **Show N entries** option.

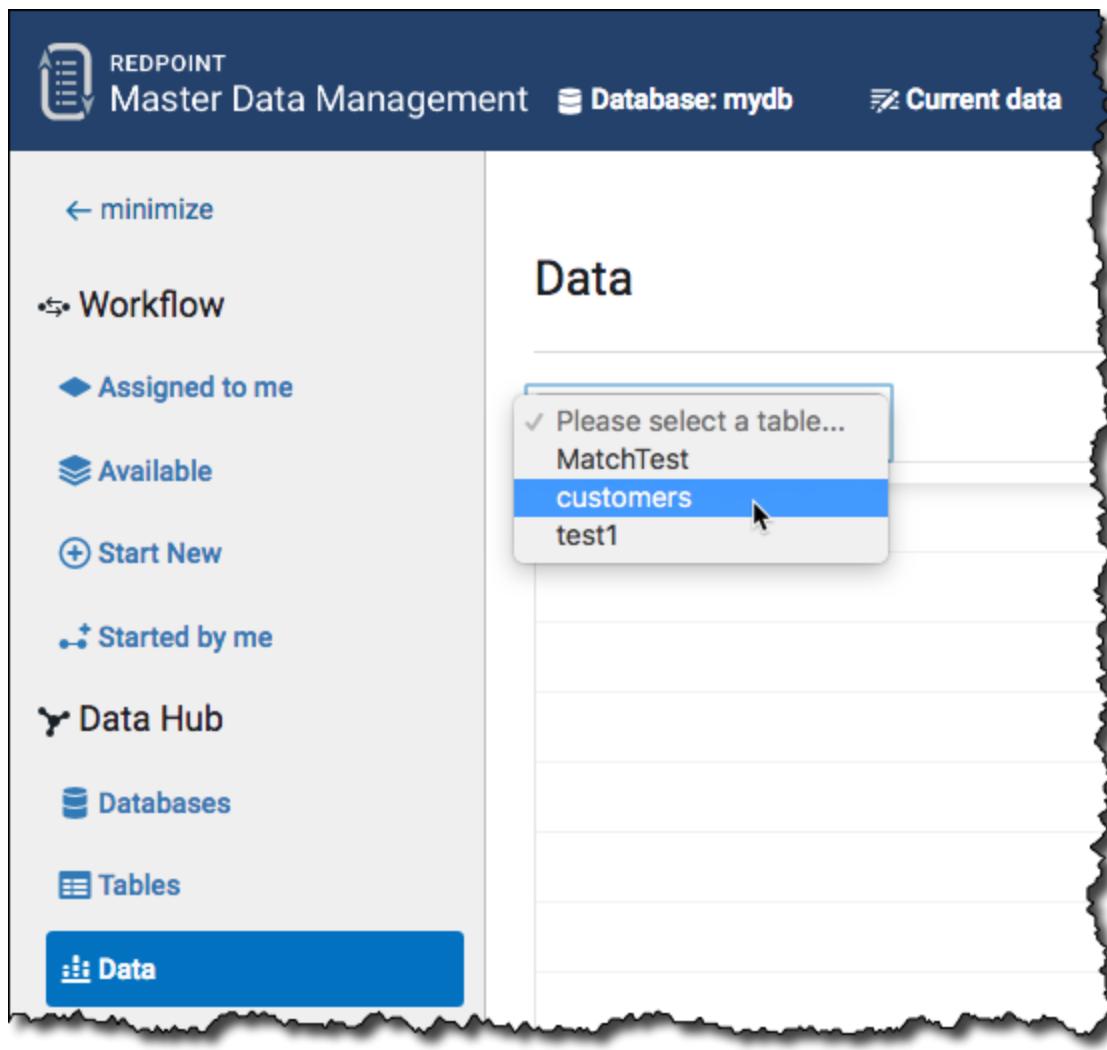
You can also search for a table directly by clicking inside the **Search** box and entering the name of the table that you want to find.

2. Click on the table name.

#### 4.2.2 From the data page

##### To search for a table from the Data page

1. In the **Data Hub** section of the MDM Web App, click **Data**.
2. On the **Data** page, click **Please select a table** and select a table.



4.3

## Query records in a table

Before you query table records, make sure to [select the correct database](#) <sup>[30]</sup> and [table](#) <sup>[30]</sup>.

### To query records in a table

1. In the **Data Hub** section of the MDM Web App, click **Data**.
2. In the **Record Query** section, [construct a query](#) <sup>[33]</sup> and click the **Search** button . The records that match the search query are displayed.

The screenshot shows the 'Data' section of the MDM Web App. At the top, a dropdown menu is set to 'customers'. Below it, a 'Record Query' section contains a search bar with '+', 'Search', and '-' buttons, and a table header with columns 'Query field', 'Operation', 'Values', and 'Delete'. A row is being edited with 'CITY' selected, an '=' operator, and the value 'AGAWAM'. Below this is a table with columns: CONTACT, COMPANY, ADDRESS, CITY, STATE, ZIP, PHONE, ID, emailaddresses, and History. Four records are listed, all matching the 'CITY: AGAWAM' query. Each record includes a 'View History' link.

| CONTACT        | COMPANY                          | ADDRESS         | CITY   | STATE | ZIP   | PHONE      | ID  | emailaddresses | History                      |
|----------------|----------------------------------|-----------------|--------|-------|-------|------------|-----|----------------|------------------------------|
| DOUGLAS DREYE  | DREYER PLUMBING                  | 53 RAMAH CIR N  | AGAWAM | MA    | 01001 | 4137892260 | 100 | 0 records      | <a href="#">View History</a> |
| RICHARD ZIEMBA | RICHIE'S AIR CON                 | 81 INDUSTRIAL L | AGAWAM | MA    | 01001 | 4137891244 | 101 | 0 records      | <a href="#">View History</a> |
| LEN ELIE       | LEN'S HEATING & 37 SAINT JACQUES | AGAWAM          | MA     | MA    | 01001 | 4137893920 | 102 | 0 records      | <a href="#">View History</a> |
| TOM KENNEDY    | UNITED RENTALS                   | 28 RAMAH CIR N  | AGAWAM | MA    | 01001 | 4137463348 | 1   | 0 records      | <a href="#">View History</a> |

Queries records in a table

#### 4.3.1 How to construct a query

Various MDM Web App tasks and workflows require that you define a query in order to locate records.

A query is a list of query terms. A query term consists of three elements:

- The *field* that will be queried.
- The query *operation* to perform.
- The *values* to be tested with the query operation.

Records returned by the query must match all query terms (that is, logical *and* behavior terms).

MDM supports the following query operations:

| Operation                         | Meaning                                               |
|-----------------------------------|-------------------------------------------------------|
| <code>field = value</code>        | The set of records where field is equal to value.     |
| <code>field &lt;&gt; value</code> | The set of records where field is not equal to value. |
| <code>field &gt; value</code>     | The set of records where field is greater than value. |
| <code>field &lt; value</code>     | The set of records where field is less than value.    |

| Operation                                  | Meaning                                                                                     |
|--------------------------------------------|---------------------------------------------------------------------------------------------|
| <code>field &gt;= value</code>             | The set of records where field is greater than or equal to value.                           |
| <code>field &lt;= value</code>             | The set of records where field is less than or equal to value.                              |
| <code>field inrange value1,value2</code>   | The set of records where field is within the inclusive range of two comma-separated values. |
| <code>field startswith value</code>        | The set of records where field starts with value.                                           |
| <code>field endswith value</code>          | The set of records where field ends with value.                                             |
| <code>field contains value</code>          | The set of records where field contains value.                                              |
| <code>field oneof value1,value2,...</code> | The set of records where field is one of a comma-separated list of values.                  |
| <code>field regex value</code>             | The set of records where field matches a standard regular expression value.                 |

**Note:** Queries are case-sensitive. The query "NAME contains John" will not return records containing the value "JOHN".

#### 4.3.2 Constructing a query in the Console

##### To construct a query in the MDM Web App

1. To display masked records in the query result, select **Include masked records** and define a query on a field that is not configured as PII.
  - Masked records are displayed in the data grid with asterisks (\*) instead of data in PII fields.
  - See [GDPR and privacy support](#)<sup>122</sup> for more information on record masking.
2. For each term choose a **Query field** and **Operation**, and enter a **Value**.
  - You can add a query term by clicking the **Add query term** button .
  - To delete a query term, click the term's **Delete** button (x).
  - To clear the entire query and start over, click the **Clear query** button .
3. To execute the query, click the **Search** button .

**Record Query**

**Include masked records**

**+        -**

| Query field                                                                                         | Operation                                                                           | Values               | Delete                                                                              |
|-----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|----------------------|-------------------------------------------------------------------------------------|
| Choose a field...  | =  | <input type="text"/> |  |

Constructs a query in the MDM Console

#### 4.3.3

#### Example: constructing a query with two terms

For a **customers** table, let's construct a query in which we search for customers who live in Dallas and work for Intel.

In the MDM Web App **Data Hub > Data** section, we need to define two query terms—one to select customers who live in Dallas, and one to select customers who work for Intel.

For the first query term, we select **CITY** as the query field, and set that equal to the value **Dallas**. The first query term looks like this:

**Record Query**

**+        -**

| Query field                                                                                     | Operation                                                                             | Values                                                                                              | Delete |
|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|--------|
| <b>CITY</b>  | =  | <b>Dallas</b>  |        |

Constructs a query with one term

Now we click the **Add query term** button . For the second query term, we select **COMPANY** as the query field, and set that equal to the value **Intel**. The second query term looks like this:

**Record Query**

| Query field | Operation | Values | Delete |
|-------------|-----------|--------|--------|
| CITY        | =         | Dallas | X      |
| COMPANY     | =         | Intel  | X      |

Constructs a query with two terms

Now we click the **Search** button  to execute the query. The matching records are displayed (in our case, one record matched the query):

**Record Query**

| Query field | Operation | Values | Delete |
|-------------|-----------|--------|--------|
| CITY        | =         | Dallas | X      |
| COMPANY     | =         | Intel  | X      |

| CONTACT     | COMPANY | ADDRESS         | CITY   | STATE | ZIP   | PHONE        | ID    | emailaddresses | History                                                                                            |
|-------------|---------|-----------------|--------|-------|-------|--------------|-------|----------------|----------------------------------------------------------------------------------------------------|
| Jane Carter | Intel   | 999 Barsoom Lan | Dallas | TX    | 99999 | 555-555-5555 | 90005 | 0 records      |  View History |

Executes a search with a two-term query

### Notes

For a query, MDM returns a maximum of 1000 records. If you reach this limit, you should tighten your query to return fewer records.

In record queries, only fields with a **display** field value of **true** are shown.

#### 4.4

### View a record's history

On the MDM Web App's **Data Hub > Data** page, you can view the history for a given record.

To view a record's history

1. In the **Data Hub** section of the MDM Web App, click **Data**.

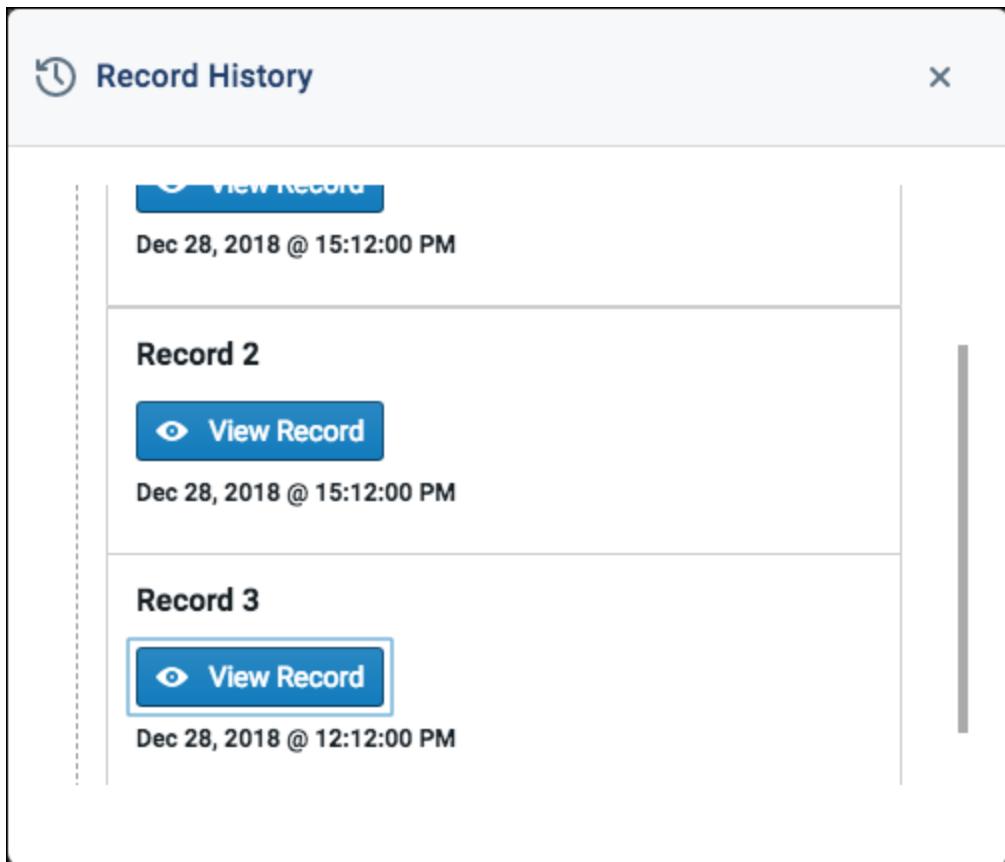
2. Select a table.

3. For the record of interest (you may have to [define a record query](#) <sup>[33]</sup>), in the **History** column click the **View History** link.

| CONTACT     | COMPANY | ADDRESS          | CITY   | STATE | ZIP   | PHONE        | ID    | emailaddresses | History                      |
|-------------|---------|------------------|--------|-------|-------|--------------|-------|----------------|------------------------------|
| Jane Carter | Intel   | 999 Barsoom Lane | Dallas | TX    | 88888 | 555-555-5555 | 90005 | 1 records      | <a href="#">View History</a> |

Record View History link

4. The **Record History** dialog displays each version of the record, along with a timestamp.



Record History dialog

5. To view the contents of a particular version of the record, click that version's **View Record** button.

The screenshot shows a list of records in a software application:

- Record 1**:
  - View Record** button
  - Timestamp: Dec 28, 2018 @ 15:12:00 PM
- Record 2**:
  - Close Record** button
  - JSON data:

```
{
 "CONTACT": "Jane Carter",
 "COMPANY": "Intel",
 "ADDRESS": "999 Barsoom Lane",
 "CITY": "Dallas",
 "STATE": "TX",
 "ZIP": "88888",
 "PHONE": "555-555-5555",
 "ID": 90005,
 "rpmdm_version": {
 "lo": 24,
 "hi": 24,
 "del": false,
 "prev": 23
 }
}
```
  - Timestamp: Dec 28, 2018 @ 15:12:00 PM
- Record 3**:
  - View Record** button
  - Timestamp: Dec 28, 2018 @ 12:12:00 PM

Views a specific record version

4.5

## Change metadata (the data model)

The **Data Hub** section allows you to change database and table metadata.

### Note

Only users in the **system** or **dbas** groups have authority to do this.

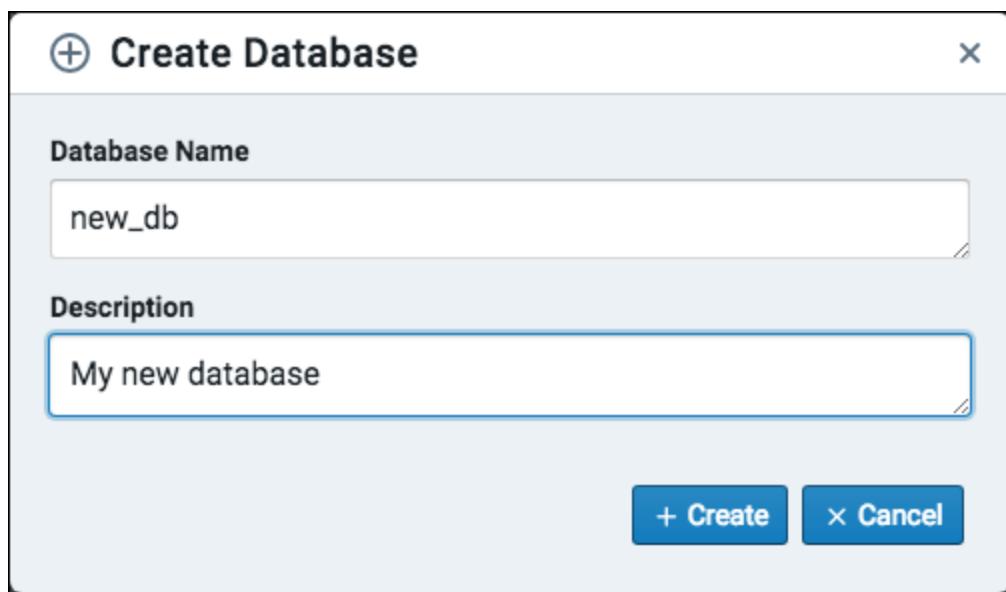
4.5.1

### Create a database

#### To create a database

1. In the **Data Hub** section of the MDM Web App, click **Databases**.
2. On the **Databases** page, click **Create Database**.
3. Enter a **Database Name** and **Description**.

**Database Name** must be unique and may not contain spaces. MDM is case-insensitive regarding database names, so "db1" is considered the same as "DB1".



Create Database dialog

4.5.2

### Delete a database

WARNING: Database deletion is not a versioned operation and **cannot** be undone!

#### To delete a database

1. In the **Data Hub** section of the MDM Web App, click **Databases**.

2. On the **Databases** page, locate the database you want to delete and click its **Delete** icon .
3. Click **OK** to confirm.

#### 4.5.3 Create a table

##### To create a table

1. In the **Data Hub** section of the MDM Web App, click **Tables**.
2. On the **Databases** page, click **Create Table**.
3. On the **Create Table** page, enter a **Name**. Table names must be unique, are case-insensitive, and may not contain spaces.
4. Configure [table options](#) .
5. You can add fields to the table now, or add them later by [editing the table schema](#)  on the **Edit Table** page.
6. Click **Create Table**.

#### 4.5.4 Delete a table

##### To delete a table

1. In the **Data Hub** section of the MDM Web App, click **Tables**.
2. On the **Tables** page, locate the table you want to delete and click its **Delete** icon .
3. Click **OK** to confirm.

##### Note

Deleted tables are not truly deleted—they still exist in historical views. In fact, you can delete a table and then re-create a table with the same name, and they will both exist in the database, but at different points in time.

#### 4.5.5 Modify a table schema

After table creation, you can:

- Modify a subset of the table attributes
- Add fields
- Change the field order
- Delete fields

## To modify a table schema

1. In the **Data Hub** section of the MDM Web App, click **Tables**.
2. On the **Tables** page, locate the desired table and click its **Schema** button .
3. You cannot modify most of the table attributes after table creation, but you can modify the **Enable RedPoint Data Management Matching Support** option and the table **Description**.
4. You cannot modify a field's **Name**, **Type**, or **Details** settings after a field has been defined, but you can change the **PII**, **Index**, and **Display** settings.
5. To add a **field** , click the **Add** button  and configure the new field's **options** .
6. To remove a field, click the field's **Delete** button .
7. Click **Update Table** when finished.

### Note

Because table metadata is versioned, data model changes can have potentially unexpected effects:

- New fields must be nullable, because when historic records are queried the missing fields will be filled with null values.
- Deleted fields will still exist in historical data. Perform a historic query to see them.
- Deleting and re-creating a field of the same name actually results in two fields—the historic field with historic data, and the new field with new data. Which one you see depends on the timestamp of your query. Historic data is not automatically migrated to the new records. To do that, you must use an ETL process.

5

## Workflow

A *workflow* is a process for performing a task on a specific database and table. It:

- coordinates tasks between multiple users and the MDM system.
- handles decisions and error conditions.
- allows for entering of notes and feedback.
- gives low-powered users a controlled path to data change.

A workflow typically consists of a series of tasks, often requiring actions by multiple users. An instance of a workflow is called a *workitem*.

For example, the workflow for adding a new customer record might look something like this:

1. User "bizuser" starts a **New Customer** workflow, creating a new workitem.

2. "Bizuser" enters customer data on the workitem's **New Record** page and submits it.
3. The workitem appears in the data steward's (user "steward") **Available** queue.
4. "Steward" clicks **Claim** to open the workitem and review the proposed customer record. Noticing a missing **PHONE** field, "steward" adds a comment "Correct missing phone" and rejects the record.
5. The workitem appears in the "bizuser" **Workitems Assigned to Me** queue.
6. "Bizuser" clicks the workitem's **Continue** button. On the workitem's **Edit Record** page, "bizuser" reads the steward's comment, enters the customer's phone number, and submits the workitem.
7. The workitem appears in the "steward" **Workitems Assigned to me** queue.
8. "Steward" clicks the workitem's **Continue** button to open the workitem and review the edited customer record. Confirming that the missing phone number was added, "steward" approves the record and it is added to the target table.

## 5.1

### The workflow Console section

The **Workflow** section of the MDM Web App is where you:

- open workitems that are assigned to you.
- claim available workitems.
- start new workflows.
- view the workitems you've started.

#### Assigned to me

| Workitem Id | Workflow          | Start Timestamp              | Task                  | Assigned To | Last User | Table     | Continue |
|-------------|-------------------|------------------------------|-----------------------|-------------|-----------|-----------|----------|
| 47          | Auto match review | 2018-11-13T18:41:41.624+0000 | OverrideMatchProposed | steward     | steward   | MatchTest |          |
| 53          | Auto match review | 2018-11-13T18:41:41.811+0000 | OverrideMatchProposed | steward     | steward   | MatchTest |          |
| 59          | Auto match review | 2018-11-13T18:41:42.006+0000 | OverrideMatchProposed | steward     | steward   | MatchTest |          |

Displays all of the workitems assigned to you (that is, workitems that require your interaction)

## Available

The screenshot shows the RedPoint Master Data Management interface with the title bar "REDPONT Master Data Management Database: mydb Current data" and a user "Logout steward". On the left, a sidebar menu includes "minimize", "Workflow", "Assigned to me" (4), "Available" (40), "Start New", "Started by me", and "Data Hub". The main area is titled "Select a Workitem to Claim" and displays a table of workitems:

| Workitem Id | Workflow      | Start Timestamp              | Task         | Assigned To | Last User | Table     | Claim |
|-------------|---------------|------------------------------|--------------|-------------|-----------|-----------|-------|
| 35          | New Customer  | 2018-11-13T15:22:04.566+0000 | ReviewRecord |             | bizuser   | customers |       |
| 36          | New Customer  | 2018-11-13T15:25:22.072+0000 | ReviewRecord |             | bizuser   | customers |       |
| 38          | Edit Customer | 2018-11-13T17:11:50.250+0000 | ReviewRecord |             | bizuser   | customers |       |

Displays all the workitems that are available for you to claim. This includes all the workitems assigned to you, as well as all the workitems you have the authorization to claim

## Start new

The screenshot shows the RedPoint Master Data Management interface with the title bar "REDPONT Master Data Management Database: mydb Current data" and a user "Logout bizuser". On the left, a sidebar menu includes "minimize", "Workflow", "Assigned to me" (2), "Available", "Start New", and "Started by me". The main area is titled "Select a Workflow to Start" and displays a table of workflows:

| Description   | Start |
|---------------|-------|
| New Customer  |       |
| Edit Customer |       |

Allows you to start a new workflow. Note that different access levels allow the creation of different kinds of workflows

## Started by me

The screenshot shows the RedPoint Master Data Management interface with the title bar "REDPONT Master Data Management Database: mydb Current data" and a user "Logout bizuser". On the left, a sidebar menu includes "minimize", "Workflow", "Assigned to me" (2), "Available", "Start New", and "Started by me". The main area is titled "Workitems Started by Me" and displays a table of workitems:

| Workitem Id | Workflow      | Start Timestamp              | Task         | Assigned To | Last User | Table     |
|-------------|---------------|------------------------------|--------------|-------------|-----------|-----------|
| 35          | New Customer  | 2018-11-13T15:22:04.566+0000 | ReviewRecord |             | bizuser   | customers |
| 36          | New Customer  | 2018-11-13T15:25:22.072+0000 | ReviewRecord |             | bizuser   | customers |
| 37          | Edit Customer | 2018-11-13T17:11:45.841+0000 | FindRecord   | bizuser     | bizuser   | customers |

Displays all of the workitems you have started

5.2

## Enter a new record

The workflow for entering a new record (creating a new customer record, for example) is as follows:

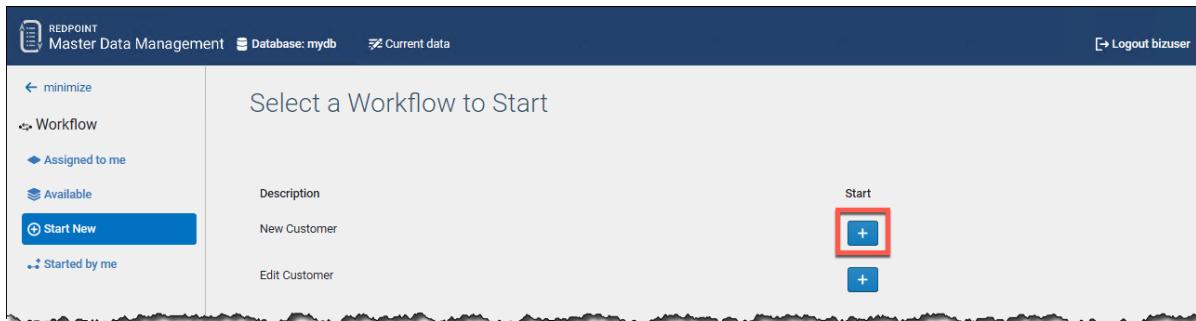
- Business user starts workflow
- Business user enters data and submits

- Steward reviews data and accepts or rejects

#### 5.2.1 Business user workflow

The business user workflow is as follows:

1. In the **Workflow** section of the MDM Web App, click **Start New**.
2. In the **New Customer** row, click the **Start** button .



3. On the **New Record** page, enter the customer's data.

New Record

**CONTACT**  
Wile E. Coyote

**COMPANY**  
Acme Enterprises

**ADDRESS**  
1515 Roadrunner Way

**CITY**  
Monument Valley

**STATE**  
UT

**Allows you to enter customer data for a new record in the Business User workflow**

4. Optionally, click in the **Comments** box and enter a comment.
5. Click **Submit** to place this workitem in the steward's queue for review. You can also **Save** this workitem and edit it later, or **Discard** it.

#### 5.2.2 Steward workflow

The steward workflow is as follows:

1. In the **Workflow** section of the MDM Web App, click **Available**.
2. On the **Select a Workitem to Claim** page, find the workitem containing the record to review and click its **Claim** button . In our case, the workflow is "New Customer", the task is "ReviewRecord", and "bizuser" was the last user (the user who created the record).

|    |              |                              |              |  |         |           |                                                                                       |
|----|--------------|------------------------------|--------------|--|---------|-----------|---------------------------------------------------------------------------------------|
| 86 | New Customer | 2019-01-08T18:44:45.714+0000 | ReviewRecord |  | bizuser | customers |  |
|----|--------------|------------------------------|--------------|--|---------|-----------|---------------------------------------------------------------------------------------|

**Allows you to claim a new record workitem in the Steward workflow**

3. On the **Data Change Review** page, review the proposed record.

## Data Change Review

• Proposed New Record

**DISCARD** **REJECT** **APPROVE**

| Proposed                              |
|---------------------------------------|
| <b>CONTACT</b><br>Wile E. Coyote      |
| <b>COMPANY</b><br>Acme Enterprises    |
| <b>ADDRESS</b><br>1515 Roadrunner Way |
| <b>CITY</b><br>Monument Valley        |

Allows you to review the proposed new record in the Steward workflow

4. At this point, you can:
  - Leave the record as-is, or make any changes necessary, and click **Approve**.
  - Write a **Comment** to the user who created the record and click **Reject**. The record will be sent back to that user's queue, and the user can make the requested changes.
  - Click **Discard**, which deletes the proposed record.

### 5.3

## Edit an existing record

The workflow for editing an existing record (changing a customer record, for example) is as follows:

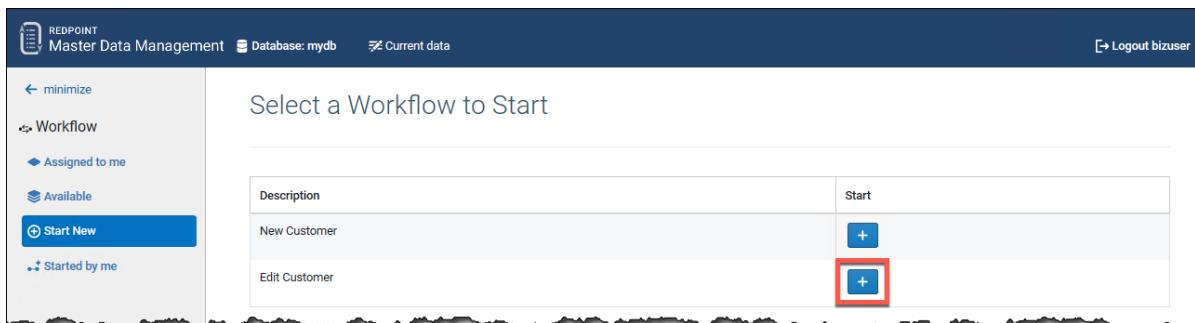
- Business user starts workflow.
- Business user finds record.
- Business user edits data and submits changes.

- Steward reviews changes and accepts or rejects.

### 5.3.1 Business user workflow

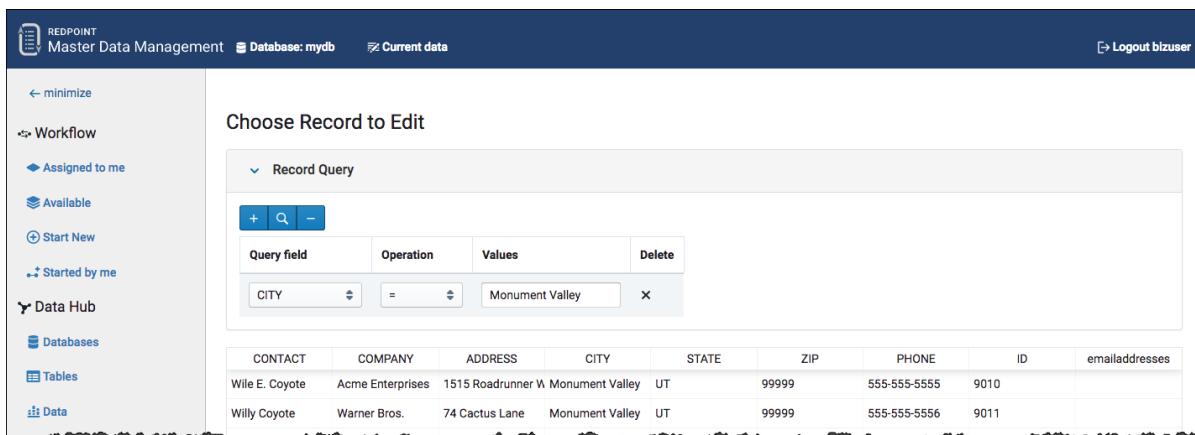
The business user workflow is as follows:

1. In the **Workflow** section of the MDM Web App, click **Start New**.
2. In the **Edit Customer** row, click the **Start** button .



Allows you to start the Business User workflow and edit an existing record

3. On the **Choose Record to Edit** page, [construct a query](#)  and click the **Search** button .



| CONTACT        | COMPANY          | ADDRESS            | CITY            | STATE | ZIP   | PHONE        | ID   | emailaddresses |
|----------------|------------------|--------------------|-----------------|-------|-------|--------------|------|----------------|
| Wile E. Coyote | Acme Enterprises | 1515 Roadrunner W. | Monument Valley | UT    | 99999 | 555-555-5555 | 9010 |                |
| Willy Coyote   | Warner Bros.     | 74 Cactus Lane     | Monument Valley | UT    | 99999 | 555-555-5556 | 9011 |                |

Allows you to choose an existing record to edit in the Business User workflow

4. Double-click on the record you want to edit.
5. On the **Edit Record** page, edit the customer's data.

**Edit Record**

**SAVE**   **DISCARD**   **SUBMIT**

|                |                            |
|----------------|----------------------------|
| <b>CONTACT</b> | AL GARCIA                  |
| <b>COMPANY</b> | Acme Gadgets International |
| <b>ADDRESS</b> | 1615 SUFFIELD ST           |
| <b>CITY</b>    | AGAWAM                     |
| <b>STATE</b>   | MA                         |

Allows you to edit customer data in an existing record in the Business User workflow

6. Optionally, click in the **Comments** box and enter a comment.
7. Click **Submit** to place this workitem in the steward's queue for review.  
You can also **Save** this workitem and edit it later, or **Discard** it.

### 5.3.2 Steward workflow

The steward workflow is as follows:

1. In the **Workflow** section of the MDM Web App, click **Available**.
2. On the **Select a Workitem to Claim** page, find the workitem containing the record to review and click its **Claim** button . In this case the workflow is "Edit Customer", the task is "ReviewRecord", and "bizuser" was the last user (the user who created the record).

|    |               |                              |              |  |         |           |                                                                                       |
|----|---------------|------------------------------|--------------|--|---------|-----------|---------------------------------------------------------------------------------------|
| 93 | Edit Customer | 2019-01-09T01:20:47.958+0000 | ReviewRecord |  | bizuser | customers |  |
|----|---------------|------------------------------|--------------|--|---------|-----------|---------------------------------------------------------------------------------------|

Allows you to select an existing record edit workitem to claim in the Steward workflow

3. On the **Data Change Review** page, review the proposed record changes.

**Data Change Review**

• 1 Proposed Change

| Original                             | Proposed                  |
|--------------------------------------|---------------------------|
| • CONTACT<br>DOUGLAS DREYER          | DOUG DREYER               |
| COMPANY<br>DREYER PLUMBING & HEATING | DREYER PLUMBING & HEATING |
| ADDRESS<br>53 RAMAH CIR N            | 53 RAMAH CIR N            |
| CITY<br>AGAWAM                       | AGAWAM                    |
| STATE<br>MA                          | MA                        |

Allows you to review the proposed existing record changes in the Steward workflow

4. At this point, you can:
  - Leave the record as-is, or make any changes necessary, and click **Approve**.
  - Write a **Comment** to the user who created the record and click **Reject**. The record will be sent back to that user's queue, and the user can make the requested changes.
  - Click **Discard**, which deletes the proposed record.

#### 5.4

## Review automated matches

If a RedPoint Data Management ETL process has automatically identified potential duplicate records, you can review these match groups and edit, accept, or reject them. For example, you could use this workflow to review individual match level for new customers.

The RedPoint Data Management ETL process compares both looser and tighter match passes. If these match passes produce different results, the process initiates a MDM review workflow so that a person can accept the looser or tighter match.

You can set the tight and loose match thresholds to any value you wish in RedPoint Data Management. For example, you could set a threshold value of 100% for accepted match groups (meaning a steward shouldn't need to review since the records 100% match). You could then

lower the proposed match group threshold to something like 74%, creating a looser match that a steward should review.

The workflow for reviewing matches identified by a RedPoint Data Management ETL process is as follows:

- Steward accepts workflow from the **Available** queue.
- Steward accepts loose or tight match, or customizes match groups.
- Steward commits changes.
- MDM stores steward decisions to prevent automatic matching from overturning decisions.

#### 5.4.1 Steward workflow

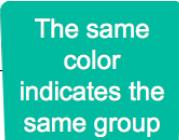
The steward workflow is as follows:

1. In the **Workflow** section of the MDM Web App, click **Available**.
2. On the **Select a Workitem to Claim** page, locate the workitem for the auto match review and click its **Claim** button . In this case the workflow is "Auto match review" and the task is "OverrideMatchProposed".

|    |                   |                              |                       |  |  |           |  |
|----|-------------------|------------------------------|-----------------------|--|--|-----------|--|
| 49 | Auto match review | 2018-11-13T18:41:41.683+0000 | OverrideMatchProposed |  |  | MatchTest |  |
|----|-------------------|------------------------------|-----------------------|--|--|-----------|--|

Allows you to select an auto match review workitem to claim in the Steward workflow

3. On the **Match Review** page, review the proposed matches:
  - a. On the **Sort Groups By** menu, select **Accepted** to display the current (tight) match groups:




| Match Review        |  | Match Review |              |             |         |      |       |            |       |    |  |
|---------------------|--|--------------|--------------|-------------|---------|------|-------|------------|-------|----|--|
|                     |  | Match Review |              |             |         |      |       |            |       |    |  |
|                     |  | Match Review |              |             |         |      |       |            |       |    |  |
| Accepted ▲ Proposed |  | Custom       | CONTACT      | COMPANY     | ADDRESS | CITY | STATE | ZIP        | PHONE | ID |  |
|                     |  | LEIGH DURAN  | GENESIS PHYS | 540 MEADOWS | AGAWAM  | MA   | 01001 | 4137891836 | 570   |    |  |
|                     |  | LEIGH DURLAN | GENESIS PHYS | 540 MEADOWS | AGAWAM  | MA   | 01001 | 4137814000 | 574   |    |  |

Allows you to view the current (tight) match groups in the selected auto match review (Steward workflow)

- b. On the **Sort Groups By** menu, select **Proposed** to display the proposed (loose) match groups:

Match Review

| Accepted | Proposed ▲ | Custom | CONTACT      | COMPANY      | ADDRESS     | CITY   | STATE | ZIP   | PHONE      | ID  |
|----------|------------|--------|--------------|--------------|-------------|--------|-------|-------|------------|-----|
|          |            |        | LEIGH DURAND | GENESIS PHYS | 540 MEADOWS | AGAWAM | MA    | 01001 | 4137891836 | 570 |
|          |            |        | LEIGH DURLAN | GENESIS PHYS | 540 MEADOWS | AGAWAM | MA    | 01001 | 4137814000 | 574 |

Comments - 0 >

Write a comment

Allows you to view the proposed (loose) match groups in the selected auto match review (Steward workflow)

- On the **Sort Groups By** menu, select **Custom** to define your own match groups:

Match Review

| Accepted | Proposed | Custom ▲ | CONTACT      | COMPANY      | ADDRESS     | CITY   | STATE | ZIP   | PHONE      | ID  |
|----------|----------|----------|--------------|--------------|-------------|--------|-------|-------|------------|-----|
|          |          |          | LEIGH DURAND | GENESIS PHYS | 540 MEADOWS | AGAWAM | MA    | 01001 | 4137891836 | 570 |
|          |          |          | LEIGH DURLAN | GENESIS PHYS | 540 MEADOWS | AGAWAM | MA    | 01001 | 4137814000 | 574 |

Comments - 0 >

Write a comment

Allows you to define your own match groups in the selected auto match review (Steward workflow)

- Click **Edit Custom Groups** to create your own match groups. When finished, click **Accept**. Click **Cancel** to return to the initial page.

**Match Review**

Drag Records To Edit Groups

| Group 1      |               |              |        |       |       |            |     |  |
|--------------|---------------|--------------|--------|-------|-------|------------|-----|--|
| CONTACT      | COMPANY       | ADDRESS      | CITY   | STATE | ZIP   | PHONE      | ID  |  |
| LEIGH DURAND | GENESIS PHYSI | 540 MEADOW S | AGAWAM | MA    | 01001 | 4137891836 | 570 |  |

| Group 2      |               |              |        |       |       |             |     |  |
|--------------|---------------|--------------|--------|-------|-------|-------------|-----|--|
| CONTACT      | COMPANY       | ADDRESS      | CITY   | STATE | ZIP   | PHONE       | ID  |  |
| LEIGH DURLAN | GENESIS PHYSI | 540 MEADOW S | AGAWAM | MA    | 01001 | 41378914000 | 574 |  |

Drag record here to create new group

Comments - 0 >

Write a comment

Allows you to create your own match groups in the selected auto match review (Steward workflow)

- Click **Submit** to approve the new match groups. MDM will record match group overrides for future matching. Note that automatic match group review cannot be discarded.

## 5.5

## Review manual matches

You can manage duplicate records in a table by querying the table to identify likely duplicate records and manually grouping them into match groups. For example, this workflow allows you to review and change current customer matches at individual match level.

The workflow for reviewing manual matches is as follows:

- Steward starts a new workflow
- Steward searches for existing records and match groups
- Steward recombines into correct match groups
- Steward commits changes
- MDM system stores steward decisions to prevent automatic matching from overturning decisions

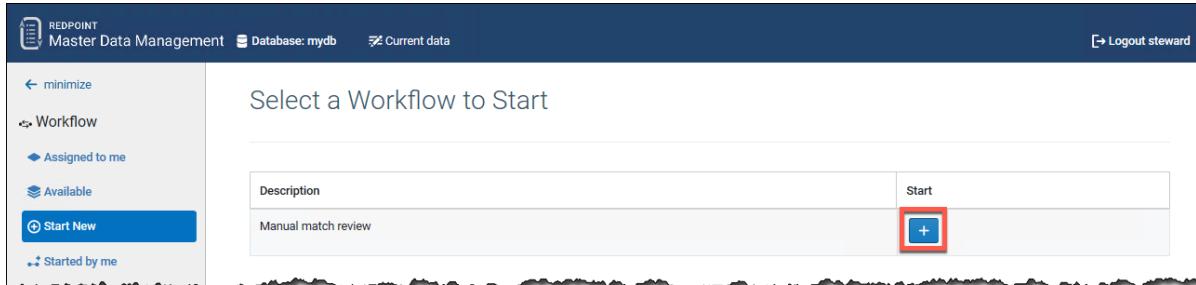
### 5.5.1

### Steward workflow

The steward workflow is as follows:

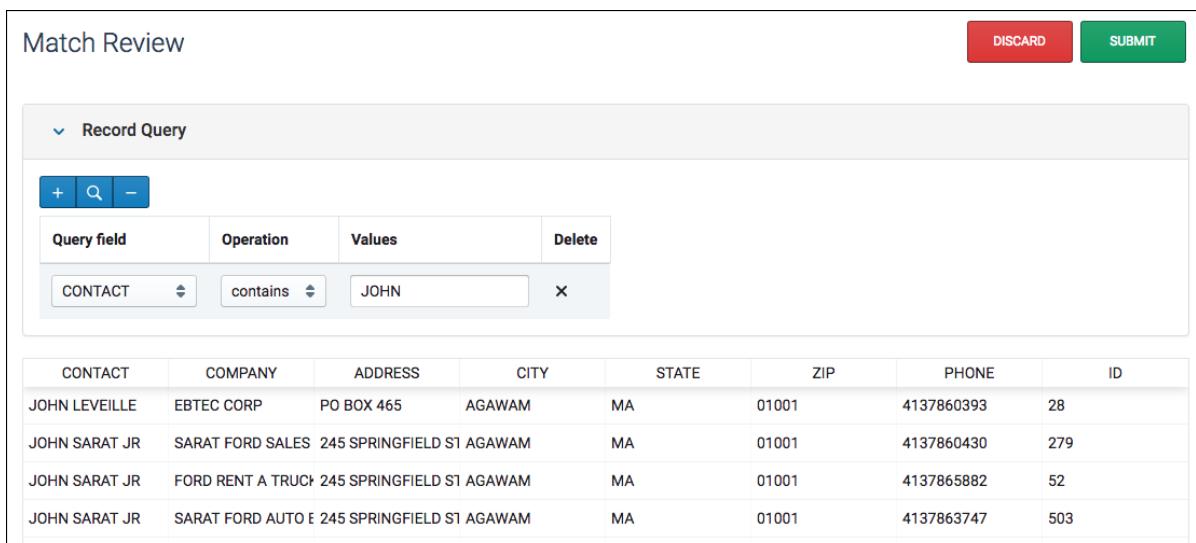
- In the **Workflow** section of the MDM Web App, click **Start New**.

2. In the **Manual match review** row, click the **Start** button .



Allows you to start a new manual match review workflow (Steward workflow)

3. On the **Match Review** page, [construct a query](#)<sup>[33]</sup> to find the records you want to review, and click the **Search** button .



| CONTACT       | COMPANY           | ADDRESS                   | CITY   | STATE | ZIP   | PHONE      | ID  |
|---------------|-------------------|---------------------------|--------|-------|-------|------------|-----|
| JOHN LEVEILLE | EBTEC CORP        | PO BOX 465                | AGAWAM | MA    | 01001 | 4137860393 | 28  |
| JOHN SARAT JR | SARAT FORD SALES  | 245 SPRINGFIELD ST AGAWAM |        | MA    | 01001 | 4137860430 | 279 |
| JOHN SARAT JR | FORD RENT A TRUCK | 245 SPRINGFIELD ST AGAWAM |        | MA    | 01001 | 4137865882 | 52  |
| JOHN SARAT JR | SARAT FORD AUTO E | 245 SPRINGFIELD ST AGAWAM |        | MA    | 01001 | 4137863747 | 503 |

Allows you to construct a query to find the records you want to review (Steward workflow)

4. Double-click on each record you want to review. This brings this record *and all of the records in its current match group* into the selected records list.

## Match Review

**DISCARD** **SUBMIT**

**Record Query**

| Query field | Operation | Values | Delete |
|-------------|-----------|--------|--------|
| CONTACT     | contains  | JOHN   | X      |

| CONTACT       | COMPANY    | ADDRESS    | CITY   | STATE | ZIP   | PHONE      | ID |
|---------------|------------|------------|--------|-------|-------|------------|----|
| JOHN LEVEILLE | EBTEC CORP | PO BOX 465 | AGAWAM | MA    | 01001 | 4137860393 | 28 |

**Selected Records**

| Group ▲       | CONTACT       | COMPANY          | ADDRESS         | CITY   | STATE | ZIP   | PHONE      | ID  |
|---------------|---------------|------------------|-----------------|--------|-------|-------|------------|-----|
| John Sarat Jr | JOHN SARAT JR | FORD RENT A TRU  | 245 SPRINGFIELD | AGAWAM | MA    | 01001 | 4137865882 | 52  |
| John Sarat Jr | JOHN SARAT JR | SARAT FORD SALE  | 245 SPRINGFIELD | AGAWAM | MA    | 01001 | 4137860430 | 279 |
| Jack Sarat    | JACK SARAT    | SARAT FORD & HII | 245 SPRINGFIELD | AGAWAM | MA    | 01001 | 4137895400 | 292 |
| John Sarat Jr | JOHN SARAT JR | SARAT FORD AUTO  | 245 SPRINGFIELD | AGAWAM | MA    | 01001 | 4137863747 | 503 |

**Allows you to select the records you want to review (Steward workflow)**

5. When you have added all the desired records to the **Selected Records** group, click **Continue**.
6. On the **Manual Match** page, drag and drop matching records into the same group. You can create multiple groups of matching records. MDM displays previously-defined match groups (if any exist).

**Manual Match**

Drag Records Between Groups

| Group 1       |                |                |        |       |       |            |     |  |
|---------------|----------------|----------------|--------|-------|-------|------------|-----|--|
| CONTACT       | COMPANY        | ADDRESS        | CITY   | STATE | ZIP   | PHONE      | ID  |  |
| JOHN SARAT JF | FORD RENT A T  | 245 SPRINGFIEL | AGAWAM | MA    | 01001 | 4137865882 | 52  |  |
| JOHN SARAT JF | SARAT FORD SA  | 245 SPRINGFIEL | AGAWAM | MA    | 01001 | 4137860430 | 279 |  |
| JACK SARAT    | SARAT FORD & I | 245 SPRINGFIEL | AGAWAM | MA    | 01001 | 4137895400 | 292 |  |
| JOHN SARAT JF | SARAT FORD AL  | 245 SPRINGFIEL | AGAWAM | MA    | 01001 | 4137863747 | 503 |  |

Drag record here to create new group

Comments - 0

Write a comment

Allows you to move matching records into the same group (Steward workflow)

**Manual Match**

Drag Records Between Groups

| Group 1    |                |                |        |       |       |            |     |  |
|------------|----------------|----------------|--------|-------|-------|------------|-----|--|
| CONTACT    | COMPANY        | ADDRESS        | CITY   | STATE | ZIP   | PHONE      | ID  |  |
| JACK SARAT | SARAT FORD & I | 245 SPRINGFIEL | AGAWAM | MA    | 01001 | 4137895400 | 292 |  |

| Group 2       |               |                |        |       |       |            |     |  |
|---------------|---------------|----------------|--------|-------|-------|------------|-----|--|
| CONTACT       | COMPANY       | ADDRESS        | CITY   | STATE | ZIP   | PHONE      | ID  |  |
| JOHN SARAT JF | FORD RENT A T | 245 SPRINGFIEL | AGAWAM | MA    | 01001 | 4137865882 | 52  |  |
| JOHN SARAT JF | SARAT FORD SA | 245 SPRINGFIEL | AGAWAM | MA    | 01001 | 4137860430 | 279 |  |
| JOHN SARAT JF | SARAT FORD AL | 245 SPRINGFIEL | AGAWAM | MA    | 01001 | 4137863747 | 503 |  |

Drag record here to create new group

Allows you to approve the new manual match groups (Steward workflow)

- Click **Submit** to approve the new match groups. MDM will record match group overrides for future matching.
- You can also click **Back** to return to the **Match Review** page, or **Discard** to abandon the match review workflow entirely.

6

## RedPoint Data Management's MDM tools

MDM uses a set of RedPoint Data Management tools to perform "back end" ETL and matching functions. These tools require a license and are intended for use with RedPoint Master Data Management.

For a more general explanation of RedPoint Data Management tools, refer to the *RedPoint Data Management User Guide*.

### To install MDM tools in RedPoint Data Management

1. Copy the RedPoint-provided "shaded" version of the MDM tools JAR file into the `\java_plugins` folder in the RedPoint Data Management installation folder. The filename is of the form `mdm-connectors-*-.shaded.jar`.
2. In the RedPoint Data Management client, click the **Palette** menu button at the top of the tool palette  and select **Reset Palette**.  
The MDM tools will appear in the **Master Data** tool group.
3. Click the **Repository** tab and go to **Settings > Tools**.
4. Click the **MDM** tab and enter your server and authentication credentials.
5. Close and re-open any projects that were active before installing the MDM tools JAR file.

6.1

### MDM tool connection settings

RedPoint Data Management's MDM tools use [shared settings](#)<sup>[59]</sup>, which allow you to define a single set of configuration properties (typically access credentials) to share across multiple tools in your RedPoint Data Management site. You can override these settings on a per-tool basis by opening the **Connection settings** section on the tool's Properties pane, selecting **Override**, and specifying values for that specific tool.

### To define MDM shared tool settings

1. Open the **Tools** folder under **Settings** in the repository.
2. Click the **MDM** tab, and then view the Properties pane.
3. Configure the tool properties for your environment:

|                                  |                                                                                                                                             |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MDM server URL</b>            | URL for the MDM server, typically a MongoDB server URI. A default "local" install of MDM will use <code>http://localhost:12345/mdm</code> . |
| <b>Authentication server URL</b> | URL for the MDM authentication server. A default "local" install of MDM will use                                                            |

|                                    |                                                                                                        |
|------------------------------------|--------------------------------------------------------------------------------------------------------|
|                                    | <code>http://localhost:9901/mdm.</code>                                                                |
| <b>Username</b>                    | User logon name. The <code>system</code> user is recommended because it has privileges to do anything. |
| <b>Password</b>                    | User logon password. The default password for the <code>system</code> user is "system".                |
| <b>Override webservice timeout</b> | If selected, override the standard connection timeout of 60 seconds.                                   |
| <b>Webservice wait seconds</b>     | If <b>Override webservice timeout</b> is selected, the new connection timeout value (in seconds).      |

### Tips

If you change the shared settings, you may need to close and re-open any active projects for them to take effect.

To configure default shared tool settings from an MDM tool's Properties pane, open the **Connection settings** section, and click **Edit default settings**.

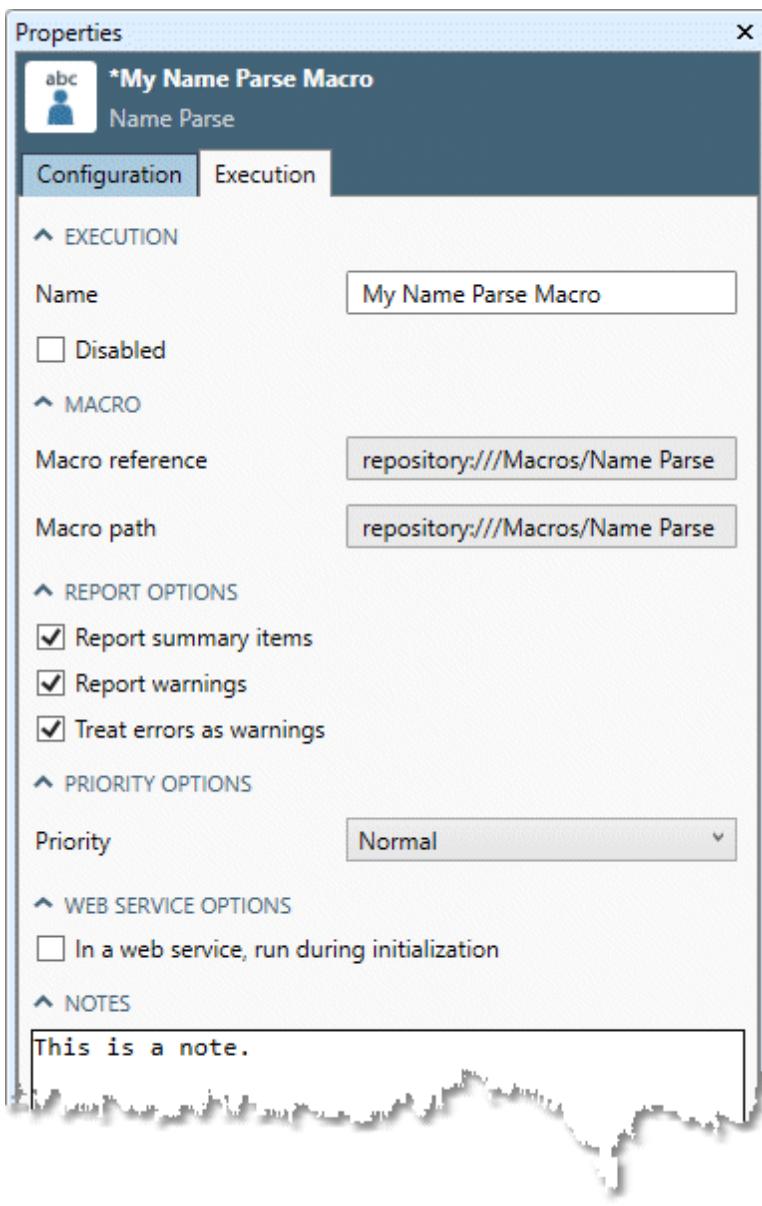
#### To override MDM tool shared settings

1. Select the desired MDM tool, and then click the **Configuration** tab on the Properties pane.
2. Open the **Connection settings** section, click **Override**, and then specify new values for the tool.

6.2

## MDM tool execution options

When you configure RedPoint Data Management's tools and macros, you can click the **Execution** tab on their **Properties** pane and specify the tool name and how to report run-time details on the results of the tool's processing. These details appear in the Message Viewer.



Tool execution options

- All tools and macros have a **Disable** option, which you can access on the Execution tab or on the context menu.
- If the object is a macro, **Macro reference** defines the relative path to the macro, while **Macro path** defines the absolute path to the macro.
- Select **Report summary items** to report the results of the tool's process—for example, the number of records read from an input file.
- Select **Report warnings** to report minor problems as warning messages in the Message Viewer. If this option is *not* selected, problems that do not result in the project being aborted will not be reported.

- Select **Treat errors as warnings** if you want errors to be reported as warnings rather than causing the project to fail. If this option is *not* selected, errors will cause the project to be aborted, and will be reported as error messages in the Message Viewer.

### 6.3 Trigger input and output

Many RedPoint Data Management MongoDB and MDM tools have an **Enable trigger output** option. When selected, this option causes the tool to sprout a T "Trigger" connector. A single record will be sent to this connection after all operations are complete. This can be used to sequence other operations by connecting to downstream tools that support a Trigger input connector. Selecting **Enable trigger input** in these tools causes the tool to sprout the T input connector, which can be used to make the tool wait for previous tools to complete before starting.

### 6.4 Configure shared settings

RedPoint Data Management's MDM tools use shared settings, which allows you to share a single set of default configuration properties (typically access credentials) across multiple tools in your site. You can override these settings on a per-tool basis by checking **Override** on a tool's Properties pane, and specifying values for that specific tool.

#### To define shared tool settings

1. Open the **Tools** folder under **Settings** in the RedPoint Data Management repository.
2. Click the tab for the desired tool name, and then view the Properties pane.
3. Configure the default tool properties for your environment.

#### *Tip*

To configure default shared tool settings from a tool's Properties pane, open the **Connection settings** or **Shared settings** section, and click **Edit default settings**.

### 6.5 MDM Input tool

This tool reads records from the MDM database.

- By default, the tool reads the current database last committed changeset.
- You can specify a changeset or timestamp to read from a database view at a specific point in history.
- You can specify a query filter to reduce the record set.
- Options to read the version, conflict, and matching sub-documents are available. These documents are produced as JSON strings.

### 6.5.1 MDM Input tool parameters

The MDM Input tool has three sets of parameters in addition to the standard [execution options](#)<sup>57</sup>: *Configuration*, *Filters*, and *Options*.

#### Configuration

|                         |                                                                                                                                                                                                                                                                                                                         |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Override</b>         | See <a href="#">MDM tool shared settings</a> <sup>57</sup> .                                                                                                                                                                                                                                                            |
| <b>Database</b>         | The name of the MDM database from which records will be read.                                                                                                                                                                                                                                                           |
| <b>Table</b>            | The name of the table from which records will be read.                                                                                                                                                                                                                                                                  |
| <b>Detailed logging</b> | If selected, sends detailed activity logs to the Message Viewer.                                                                                                                                                                                                                                                        |
| <b>Select data from</b> | Specifies which version of the data to read records from: <ul style="list-style-type: none"> <li><b>Current</b>: the most recently-committed data.</li> <li><b>Timestamp</b>: the latest changeset committed before a specified timestamp.</li> <li><b>Pending changeset</b>: a specified pending changeset.</li> </ul> |
| <b>Timestamp</b>        | If <b>Select data from</b> is <b>Timestamp</b> , specifies read data as the table existed at that moment in time.                                                                                                                                                                                                       |
| <b>Changeset</b>        | If <b>Select data from</b> is <b>Pending changeset</b> , specifies the changeset.                                                                                                                                                                                                                                       |

#### Filters

|                                    |                                                                                                                                                                                                  |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field value filter</b>          | Use the grid to filter the desired input by defining <b>Field - Operation - Values</b> expressions. For example:<br><br>ZIP = "91520"<br><br>The terms in the grid are logically ANDed together. |
| <b>Enable recent commit filter</b> | Limits input to records committed within the defined time period.                                                                                                                                |

|                              |                                                                                                   |
|------------------------------|---------------------------------------------------------------------------------------------------|
| <b>Committed within last</b> | If <b>Enable recent commit filter</b> is selected, use the time picker to select the time period. |
|------------------------------|---------------------------------------------------------------------------------------------------|

## Options

|                             |                                                                                                                                                                                                                                                                                |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Limit records</b>        | If selected, limits the number of records read.                                                                                                                                                                                                                                |
| <b>Record limit</b>         | If <b>Limit records</b> is selected, specifies the maximum number of records to be read.                                                                                                                                                                                       |
| <b>Return match info</b>    | Select to add a JSON document field to the output record containing match group IDs, match scores, and user overrides for all match types.                                                                                                                                     |
| <b>Block size</b>           | Maximum number of records per transaction/request. Default is <b>100</b> . This is a tuning parameter. Generally, you can make this larger for narrow records and smaller for wide records.                                                                                    |
| <b>Enable trigger input</b> | If selected, causes the tool to sprout a T "Trigger" input connector. This can be used to sequence operations by connecting to upstream tools that support a Trigger output connector. For more information, refer to <a href="#">Trigger input and output</a> <sup>59</sup> . |

### 6.5.2 Configure the MDM Input tool

Before configuring an MDM Input tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>56</sup>.

#### To configure the MDM Input tool

1. Select the **MDM Input** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>56</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. On the **Select data from** dropdown, choose the desired view of the data:
  - **Current**: The most recently-committed data.
  - **Timestamp**: Data committed on or before the defined timestamp. If you select this option, use the **Timestamp** data picker to choose a specific timestamp.

- **Pending changeset:** Data from the selected changeset. If you select this option, select a **Changeset** from the dropdown.
6. Optionally, click **Refresh sample data** to view a sample of the input data.
  7. Optionally, click the **Filters** tab and define filters for the input data:
    - Use the **Field value filter** grid to define one or more **Field - Operation - Values** expressions.
    - Click **Enable recent commit filter** to configure a **Committed within last** time value.
  8. Optionally, click the **Options** tab and configure processing options:
    - Click **Limit records** and specify a **Record limit**.
    - Click **Return match info** to return a JSON document containing match information for each record.
    - Change **Block size** from the default value of **100**.
    - Add a T input connector by clicking **Enable trigger input**. For more information, refer to [Trigger input and output](#)<sup>59</sup>.
  9. Set any desired [execution options](#)<sup>57</sup>.

## 6.6

## MDM Output tool

This tool inserts, upserts, or modifies records in the MDM database. It can create a table if one does not yet exist. It cannot create or modify matching information (this is handled by a separate tool).

### 6.6.1

### MDM Output tool parameters

The MDM Output tool has the following sets of parameters in addition to the standard [execution options](#)<sup>57</sup>:

- *Configuration*
- *Options*
- *Create*
- *Mapping*

#### Configuration

|                 |                                                                |
|-----------------|----------------------------------------------------------------|
| <b>Override</b> | See <a href="#">MDM tool shared settings</a> <sup>57</sup> .   |
| <b>Database</b> | The name of the MDM database to which records will be written. |

|                         |                                                                  |
|-------------------------|------------------------------------------------------------------|
| <b>Table</b>            | The name of the table to which records will be written.          |
| <b>Detailed logging</b> | If selected, sends detailed activity logs to the Message Viewer. |

## Options

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Processing mode</b>           | Specifies how RedPoint Data Management will output records to the target <b>Database</b> and <b>Table</b> . <ul style="list-style-type: none"> <li><b>Create:</b> Inserts new records into the table. If a record with the same key field value already exists (or if there is another unique index violation), the operation fails and the project aborts.</li> <li><b>Upsert:</b> Attempts to insert new records into the table. If a record with the same key field value already exists (or if there is another unique index violation), the existing record is replaced by the new record, otherwise a new record is inserted.</li> <li><b>Replace:</b> Replaces each existing record with a new one. If records with the same primary key do not exist, the operation fails and the project aborts.</li> </ul> |
| <b>Block size</b>                | Maximum number of records per transaction/request. Default is <b>100</b> . This is a tuning parameter. Generally, you can make this larger for narrow records and smaller for wide records.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Use shared changeset</b>      | All MDM tools with this option selected cooperate around a single changeset. Pending changes across all such tools are gathered into a single shared changeset. When the last MDM tool in the project with this option selected finishes, the shared changeset is committed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Wait for changeset commit</b> | Select this option to wait for the changeset to commit at the end of the tool's processing. If not selected, the changeset commit will be requested but the tool/project will not wait for it to complete.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

|                                   |                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Commit wait seconds</b>        | If <b>Wait for changeset commit</b> is selected, the time to wait for a changeset commit to complete. If the commit does not complete within this time, the project fails and aborts; however, the commit will continue to process.                                                                                                                                              |
| <b>Threads</b>                    | Degree of parallel processing desired. Values up to <b>3</b> may help speed processing. Default is <b>1</b> .                                                                                                                                                                                                                                                                    |
| <b>Enable trigger output</b>      | If selected, this option causes the tool to sprout a T "Trigger" connector. A single record will be sent to this connection after all operations are complete. This can be used to sequence other operations by connecting to downstream tools that support a Trigger input connector. For more information, refer to <a href="#">Trigger input and output</a> <sup>[59]</sup> . |
| <b>Clear table before loading</b> | If selected, deletes all existing records from the table before writing to it. This is appropriate only for development environments.                                                                                                                                                                                                                                            |

## Create

This tab allows you to specify how tables are created, and to create and delete them interactively.

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Create table from</b> | <ul style="list-style-type: none"> <li>• <b>JSON:</b> Table will be created from JSON schema that you specify below.</li> <li>• <b>Upstream:</b> Table will be created to mimic the schema of the input connector. Not all schema features are available when using this option.</li> </ul>                                                                                                                                                                                                                                                                                      |
| <b>Actions</b>           | <ul style="list-style-type: none"> <li>• <b>Create table:</b> Create a table. Will fail if the table already exists.</li> <li>• <b>Delete table:</b> Delete a table (you will be asked to confirm). Note that when you delete a table, the historic table data is retained and can be queried using a historic view.</li> <li>• <b>Refresh:</b> Refresh the table list.</li> <li>• <b>Make JSON from upstream:</b> Create JSON that mimics the upstream connector schema, that you can then edit to add MDM-specific features such as indexes, sub-tables, and so on.</li> </ul> |

|                                |                                                                                                                                                                              |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Enter JSON schema</b>       | If <b>Create table from</b> is <b>JSON</b> , the JSON schema to be applied when creating the table.                                                                          |
| <b>Primary key field</b>       | If <b>Create table from</b> is <b>Upstream</b> , specify the primary key field for the new table.                                                                            |
| <b>Table supports matching</b> | If <b>Create table from</b> is <b>Upstream</b> , whether the new table supports matching.                                                                                    |
| <b>Choose display fields</b>   | If <b>Create table from</b> is <b>Upstream</b> , the fields to be displayed by default in the MDM web application. These are typically the "common" or "identifying" fields. |

## Mapping

|                           |                                                                                                                                                                                                                                                                                                                       |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Mapping type</b>       | <ul style="list-style-type: none"> <li>• <b>Implicit:</b> Mapping between upstream fields and MDM fields is based on case-insensitive name.</li> <li>• <b>Explicit:</b> Mapping between upstream fields and MDM fields is entered here manually.</li> </ul>                                                           |
| <b>Actions</b>            | <p>If <b>Mapping type</b> is <b>Explicit</b>, specify how to map input fields to MDM fields:</p> <ul style="list-style-type: none"> <li>• <b>Refresh:</b> Read the MDM table schema again.</li> <li>• <b>Default:</b> Match up field names and create entries.</li> <li>• <b>Clear:</b> Clear all mapping.</li> </ul> |
| <b>Field mapping grid</b> | Displays the mapping of <b>Input</b> fields to <b>MDM</b> fields and allows editing of that mapping.                                                                                                                                                                                                                  |

### 6.6.2 Configure the MDM Output tool

Before configuring an MDM Output tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>[56]</sup>.

#### To configure the MDM Output tool

1. Select the **MDM Output** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>[58]</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.

3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. Choose a **Processing mode**:
  - **Create**: Inserts new records into the table.
  - **Upsert**: Attempts to insert new records into the table.
  - **Replace**: Replaces each existing record with a new one.
6. Configure **Use shared changeset**, **Wait for changeset commit**, and **Commit wait seconds** options to specify how the changeset is handled.
7. To enable parallel loading, specify **Threads** as a value greater than **1**.
8. To support sequencing of later tools after the load is complete, select **Enable trigger output**.
9. Optionally, select **Clear table before loading**. This is generally used only for development and is not recommended for production processing.
10. If the target table does not yet exist, click the **Create** tab and [specify how the table should be created](#)<sup>[64]</sup>. See the "JSON: Schemas" section of the MDM API documentation if you require more granular control over table structure.
11. To use an input-to-MDM mapping other than the default **Implicit** name matching, click the **Mapping** tab and configure a detailed field mapping.
12. Set any desired [execution options](#)<sup>[57]</sup>.

**6.7**

## MDM Update tool

This tool updates fields in existing MDM records. If you know you only need to update a single field (for example, recomputing the "closest store ID" in batch), the MDM Update tool is a safer and faster alternative to modifying the entire record.

**6.7.1**

### MDM Update tool parameters

The MDM Update tool has two sets of parameters in addition to the standard [execution](#)<sup>[57]</sup> options: *Configuration* and *Mapping*.

#### Configuration

|                 |                                                                |
|-----------------|----------------------------------------------------------------|
| <b>Override</b> | See <a href="#">MDM tool shared settings</a> <sup>[57]</sup> . |
| <b>Database</b> | The name of the MDM database to which records will be written. |
| <b>Table</b>    | The name of the table to which records will be written.        |

|                                  |                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Detailed logging</b>          | If selected, sends detailed activity logs to the Message Viewer.                                                                                                                                                                                                                                                                                                                 |
| <b>Block size</b>                | Maximum number of records per transaction/request. Default is <b>100</b> . This is a tuning parameter. Generally, you can make this larger for narrow records and smaller for wide records.                                                                                                                                                                                      |
| <b>Use shared changeset</b>      | All MDM tools with this option selected cooperate around a single changeset. Pending changes across all such tools are gathered into a single shared changeset. When the last MDM tool in the project with this option selected finishes, the shared changeset is committed.                                                                                                     |
| <b>Wait for changeset commit</b> | Select this option to wait for the changeset to commit at the end of the tool's processing. If not selected, the changeset commit will be requested but the tool/project will not wait for it to complete.                                                                                                                                                                       |
| <b>Commit wait seconds</b>       | If <b>Wait for changeset commit</b> is selected, the time to wait for a changeset commit to complete. If the commit does not complete within this time, the project fails and aborts; however, the commit will continue to process.                                                                                                                                              |
| <b>Threads</b>                   | Degree of parallel processing desired. Values up to <b>3</b> may help speed processing. Default is <b>1</b> .                                                                                                                                                                                                                                                                    |
| <b>Enable trigger output</b>     | If selected, this option causes the tool to sprout a T "Trigger" connector. A single record will be sent to this connection after all operations are complete. This can be used to sequence other operations by connecting to downstream tools that support a Trigger input connector. For more information, refer to <a href="#">Trigger input and output</a> <sup>[59]</sup> . |

## Mapping

|                              |                                                                                                                                                                                                                                       |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field mapping Actions</b> | <p>Specify how to map input fields to MDM fields:</p> <ul style="list-style-type: none"> <li>• <b>Refresh:</b> Read the MDM table schema again.</li> <li>• <b>Map all fields:</b> Match up field names and create entries.</li> </ul> |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                           |                                                                                                      |
|---------------------------|------------------------------------------------------------------------------------------------------|
|                           | <ul style="list-style-type: none"> <li>• <b>Clear mapping:</b> Clear all mapping.</li> </ul>         |
| <b>Field mapping grid</b> | Displays the mapping of <b>Input</b> fields to <b>MDM</b> fields and allows editing of that mapping. |

### 6.7.2 Configure the MDM Update tool

Before configuring an MDM Update tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>[56]</sup>.

#### To configure the MDM Update tool

1. Select the **MDM Update** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>[56]</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. Optionally, click the **Options** tab and configure processing options:
  - Change **Block size** from the default value of **100**.
  - Configure **Use shared changeset**, **Wait for changeset commit**, and **Commit wait seconds** options to specify how the changeset is handled.
  - To enable parallel updating, specify **Threads** as a value greater than **1**.
  - To support sequencing of later tools after the update is complete, select **Enable trigger output**. For more information, refer to [Trigger input and output](#)<sup>[59]</sup>.
6. To use an input-to-MDM mapping other than the default name matching, click the **Mapping** tab and configure a detailed field mapping.
7. Set any desired [execution options](#)<sup>[57]</sup>.

### 6.8

## MDM Delete tool

This tool deletes records from the MDM database that match a series of primary key values. The tool does not currently support finding (and then deleting) records by anything other than primary key.

### 6.8.1

### MDM Delete tool parameters

The MDM Delete tool has two sets of parameters in addition to the standard [execution options](#)<sup>[57]</sup>: *Configuration* and *Options*.

## Configuration

|                           |                                                                   |
|---------------------------|-------------------------------------------------------------------|
| <b>Override</b>           | See <a href="#">MDM tool shared settings</a> [57].                |
| <b>Database</b>           | The name of the MDM database from which records will be deleted.  |
| <b>Table</b>              | The name of the table from which records will be deleted.         |
| <b>Detailed logging</b>   | If selected, sends detailed activity logs to the Message Viewer.  |
| <b>Select input field</b> | Input field containing the primary key values you want to delete. |

## Options

|                                  |                                                                                                                                                                                                                                                                              |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Block size</b>                | Maximum number of records per transaction/request. Default is 100. This is a tuning parameter. Generally, you can make this larger for narrow records and smaller for wide records.                                                                                          |
| <b>Use shared changeset</b>      | All MDM tools with this option selected cooperate around a single changeset. Pending changes across all such tools are gathered into a single shared changeset. When the last MDM tool in the project with this option selected finishes, the shared changeset is committed. |
| <b>Wait for changeset commit</b> | Select this option to wait for the changeset to commit at the end of the tool's processing. If not selected, the changeset commit will be requested but the tool/project will not wait for it to complete.                                                                   |
| <b>Commit wait seconds</b>       | If <b>Wait for changeset commit</b> is selected, the time to wait for a changeset commit to complete. If the commit does not complete within this time, the project fails and aborts; however, the commit will continue to process.                                          |
| <b>Threads</b>                   | Degree of parallel processing desired. Values up to 3 may help speed processing. Default is 1.                                                                                                                                                                               |

|                              |                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Enable trigger output</b> | If selected, this option causes the tool to sprout a T "Trigger" connector. A single record will be sent to this connection after all operations are complete. This can be used to sequence other operations by connecting to downstream tools that support a Trigger input connector. See <a href="#">Trigger input and output</a> <sup>[59]</sup> . |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### 6.8.2 Configure the MDM Delete tool

Before configuring an MDM Delete tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>[56]</sup>.

##### To configure the MDM Delete tool

1. Select the **MDM Delete** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>[56]</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. In the **Select input field**, select the primary key values you want to delete.
6. Optionally, click the **Options** tab and configure processing options:
  - Change **Block size** from the default value of **100**.
  - Configure **Use shared changeset**, **Wait for changeset commit**, and **Commit wait seconds** options to specify how the changeset is handled.
  - To enable parallel deletion, specify **Threads** as a value greater than **1**.
  - To support sequencing of later tools after deletion is complete, select **Enable trigger output**.
7. Set any desired [execution options](#)<sup>[57]</sup>.

#### 6.9 MDM Key Query tool

This tool reads records from the MDM database that match a series of primary key values. It does not currently support query by anything other than primary key.

##### 6.9.1 MDM Key Query tool parameters

The MDM Key Query tool has two sets of parameters in addition to the standard execution options: *Configuration* and *Options*.

## Configuration

|                           |                                                                                                                                                                                                                                                                                                                   |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Override</b>           | See <a href="#">MDM tool shared settings</a> <sup>57</sup> .                                                                                                                                                                                                                                                      |
| <b>Database</b>           | The name of the MDM database to query.                                                                                                                                                                                                                                                                            |
| <b>Table</b>              | The name of the table to query.                                                                                                                                                                                                                                                                                   |
| <b>Detailed logging</b>   | If selected, sends detailed activity logs to the Message Viewer.                                                                                                                                                                                                                                                  |
| <b>Select data from</b>   | Specifies which version of the data to query: <ul style="list-style-type: none"> <li>• <b>Current</b>: the most recently-committed data.</li> <li>• <b>Timestamp</b>: the latest changeset committed before a specified timestamp.</li> <li>• <b>Pending changeset</b>: a specified pending changeset.</li> </ul> |
| <b>Timestamp</b>          | If <b>Select view date by</b> is <b>Timestamp</b> , specifies the timestamp.                                                                                                                                                                                                                                      |
| <b>Changeset</b>          | If <b>Select data from</b> is <b>Pending changeset</b> , specifies the changeset.                                                                                                                                                                                                                                 |
| <b>Select input field</b> | Input field containing the primary key values you want to query.                                                                                                                                                                                                                                                  |

## Options

|                          |                                                                                                                                                                                             |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Return match info</b> | Select to add a JSON document field to the output record containing match group IDs, match scores, and user overrides for all match types.                                                  |
| <b>Block size</b>        | Maximum number of records per transaction/request. Default is <b>100</b> . This is a tuning parameter. Generally, you can make this larger for narrow records and smaller for wide records. |

### 6.9.2 Configure the MDM Key Query tool

Before configuring an MDM Key Query tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>56</sup>.

### To configure the MDM Key Query tool

1. Select the **MDM Key Query** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>56</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. On the **Select data from** dropdown, choose the desired view of the data:
  - **Current**: The most recently-committed data.
  - **Timestamp**: Data committed on or before the defined timestamp. If you select this option, use the **Timestamp** data picker to choose a specific timestamp.
  - **Pending changeset**: Data from the selected changeset. If you select this option, select a **Changeset** from the dropdown.
6. In the **Select input field**, select the primary key values you want to query.
7. Optionally, click **Refresh sample data** to view a sample of the input data.
8. Optionally, click the **Options** tab and configure processing options:
  - Click **Return match info** to return a JSON document containing match information for each record.
  - Change **Block size** from the default value of **100**.
9. Set any desired [execution options](#)<sup>57</sup>.

6.10

## MDM Dynamic Query tool

This tool reads records from the MDM database with fields that match a series of values. For example, this tool is useful for querying "match candidates" by segmentation key. The fields should be indexed in the schema or performance will be poor.

6.10.1

### MDM Dynamic Query tool parameters

The MDM Dynamic Query tool has two sets of parameters in addition to the standard execution options: *Configuration* and *Options*.

#### Configuration

|                 |                                                              |
|-----------------|--------------------------------------------------------------|
| <b>Override</b> | See <a href="#">MDM tool shared settings</a> <sup>57</sup> . |
| <b>Database</b> | The name of the MDM database to query.                       |

|                                |                                                                                                                                                                                                                                                                                                                   |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Table</b>                   | The name of the table to query.                                                                                                                                                                                                                                                                                   |
| <b>Detailed logging</b>        | If selected, sends detailed activity logs to the Message Viewer.                                                                                                                                                                                                                                                  |
| <b>Select data from</b>        | Specifies which version of the data to query: <ul style="list-style-type: none"> <li>• <b>Current</b>: the most recently-committed data.</li> <li>• <b>Timestamp</b>: the latest changeset committed before a specified timestamp.</li> <li>• <b>Pending changeset</b>: a specified pending changeset.</li> </ul> |
| <b>Timestamp</b>               | If <b>Select view date by</b> is <b>Timestamp</b> , specifies the timestamp.                                                                                                                                                                                                                                      |
| <b>Changeset</b>               | If <b>Select data from</b> is <b>Pending changeset</b> , specifies the changeset.                                                                                                                                                                                                                                 |
| <b>MDM field</b>               | MDM field you want to query.                                                                                                                                                                                                                                                                                      |
| <b>Input field</b>             | Input field containing the primary key values you want to query in the MDM field.                                                                                                                                                                                                                                 |
| <b>Additional filter terms</b> | Use the grid to filter the desired input by defining <b>Field - Operation - Values</b> expressions. For example:<br><br>ZIP = "91520"<br><br>The terms in the grid are logically ANDed together.                                                                                                                  |

## Options

|                          |                                                                                                                                                                                             |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Return match info</b> | Select to return a JSON document containing match information for each record.                                                                                                              |
| <b>Block size</b>        | Maximum number of records per transaction/request. Default is <b>100</b> . This is a tuning parameter. Generally, you can make this larger for narrow records and smaller for wide records. |

#### 6.10.2 Configure the MDM Dynamic Query tool

Before configuring an MDM Dynamic Query tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>56</sup>.

##### To configure the MDM Dynamic Query tool

1. Select the **MDM Dynamic Query** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>56</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. On the **Select data from** dropdown, choose the desired view of the data:
  - **Current**: The most recently-committed data.
  - **Timestamp**: Data committed on or before the defined timestamp. If you select this option, use the **Timestamp** data picker to choose a specific timestamp.
  - **Pending changeset**: Data from the selected changeset. If you select this option, select a **Changeset** from the dropdown.
6. In the **MDM field**, select the MDM field you want to query.
7. In the **Input field**, select the primary key values you want to query in the MDM field.
8. In the **Additional filter terms** section, define filters for the input data. Use the grid to define one or more **Field - Operation - Values** expressions.
9. Optionally, click **Refresh sample data** to view a sample of the input data.
10. Optionally, click the **Options** tab and configure processing options:
  - Click **Return match info** to return a JSON document containing match information for each record.
  - Change **Block size** from the default value of **100**.
11. Set any desired [execution options](#)<sup>57</sup>.

#### 6.11 MDM Query View tool

This tool queries results of a view, reading all records in batch. It can read the current data or query a historical view of the data.

### 6.11.1 MDM Query View tool parameters

The MDM Query View tool has two sets of parameters in addition to the standard execution options: *Configuration* and *Options*.

#### Configuration

|                         |                                                                                                                                                                                                                                                                                                                   |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Override</b>         | See <a href="#">MDM tool shared settings</a> <sup>[57]</sup> .                                                                                                                                                                                                                                                    |
| <b>Database</b>         | The name of the MDM database to query.                                                                                                                                                                                                                                                                            |
| <b>View</b>             | The name of the view to query.                                                                                                                                                                                                                                                                                    |
| <b>Detailed logging</b> | If selected, sends detailed activity logs to the Message Viewer.                                                                                                                                                                                                                                                  |
| <b>Select data from</b> | Specifies which version of the data to query: <ul style="list-style-type: none"> <li>• <b>Current</b>: the most recently-committed data.</li> <li>• <b>Timestamp</b>: the latest changeset committed before a specified timestamp.</li> <li>• <b>Pending changeset</b>: a specified pending changeset.</li> </ul> |
| <b>Timestamp</b>        | If <b>Select data from</b> is <b>Timestamp</b> , specifies the timestamp.                                                                                                                                                                                                                                         |
| <b>Changeset</b>        | If <b>Select data from</b> is <b>Pending changeset</b> , specifies the changeset.                                                                                                                                                                                                                                 |

#### Options

|                             |                                                                                                                                                                                             |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Limit records</b>        | If selected, limits the number of records read.                                                                                                                                             |
| <b>Record limit</b>         | If <b>Limit records</b> is selected, specifies the number of records to be read.                                                                                                            |
| <b>Block size</b>           | Maximum number of records per transaction/request. Default is <b>100</b> . This is a tuning parameter. Generally, you can make this larger for narrow records and smaller for wide records. |
| <b>Enable trigger input</b> | If selected, causes the tool to sprout a T "Trigger" input connector. This can be used to sequence operations by connected to upstream tools that                                           |

|  |                                                                                                  |
|--|--------------------------------------------------------------------------------------------------|
|  | support a Trigger output connector. See <a href="#">Trigger input and output</a> <sup>59</sup> . |
|--|--------------------------------------------------------------------------------------------------|

### 6.11.2 Configure the MDM Query View tool

Before configuring an MDM Query View tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>56</sup>.

#### To configure the MDM Query View tool

1. Select the **MDM Query View** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>56</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **View**.
5. On the **Select data from** dropdown, choose the desired view of the data:
  - **Current**: The most recently-committed data.
  - **Timestamp**: Data committed on or before the defined timestamp. If you select this option, use the **Timestamp** data picker to choose a specific timestamp.
  - **Pending changeset**: Data from the selected changeset. If you select this option, select a **Changeset** from the dropdown.
6. Optionally, click **Refresh sample data** to view a sample of the input data.
7. Optionally, click the **Options** tab and configure processing options:
  - Click **Limit records** and specify a **Record limit**.
  - Change **Block size** from the default value of **100**.
  - Add a T input connector by clicking **Enable trigger input**. For more information, refer to [Trigger input and output](#)<sup>59</sup>.
8. Set any desired [execution options](#)<sup>57</sup>.

### 6.12 MDM Set Match Groups tool

This tool sets match group IDs on existing records. It accepts a sequence of primary key values and group IDs.

It is assumed that the RedPoint Data Management projects are well-behaved:

- Honor the suppression information and not match records that have overrides or are in-match-review. Failure to do so will result in an inconsistent database.
- Use the [MDM Generate Match Group IDs tool](#)<sup>[79]</sup> to obtain the "next match group" IDs. RedPoint Data Management projects should not make up their own IDs using their own rules. Even if the matching macros produce match group IDs, they must be re-mapped onto MDM match group IDs.

#### 6.12.1 MDM Set Match Groups tool parameters

The MDM Set Match Groups tool has a single set of parameters in addition to the standard execution options: *Configuration*.

##### Configuration

|                                  |                                                                                                                                                                                                                                                                              |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Override</b>                  | See <a href="#">MDM tool shared settings</a> <sup>[57]</sup> .                                                                                                                                                                                                               |
| <b>Database</b>                  | The name of the MDM database to query.                                                                                                                                                                                                                                       |
| <b>Table</b>                     | The name of the table to query.                                                                                                                                                                                                                                              |
| <b>Detailed logging</b>          | If selected, sends detailed activity logs to the Message Viewer.                                                                                                                                                                                                             |
| <b>Match type</b>                | The match type for this workflow (for example, <b>individual</b> or <b>household</b> ).                                                                                                                                                                                      |
| <b>Primary key input field</b>   | The primary key field for the input records, which must be an <b>integer</b> or <b>long</b> .                                                                                                                                                                                |
| <b>Group ID input field</b>      | The group ID field for the input records.                                                                                                                                                                                                                                    |
| <b>Score field</b>               | The score field for the input records.                                                                                                                                                                                                                                       |
| <b>Block size</b>                | Maximum number of records per transaction/request. Default is <b>100</b> . This is a tuning parameter. Generally, you can make this larger for narrow records and smaller for wide records.                                                                                  |
| <b>Use shared changeset</b>      | All MDM tools with this option selected cooperate around a single changeset. Pending changes across all such tools are gathered into a single shared changeset. When the last MDM tool in the project with this option selected finishes, the shared changeset is committed. |
| <b>Wait for changeset commit</b> | Select this option to wait for the changeset to commit at the end of the tool's processing. If not                                                                                                                                                                           |

|                              |                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                              | selected, the changeset commit will be requested but the tool/project will not wait for it to complete.                                                                                                                                                                                                                                             |
| <b>Commit wait seconds</b>   | If <b>Wait for changeset commit</b> is selected, the time to wait for a changeset commit to complete. If the commit does not complete within this time, the project fails and aborts; however, the commit will continue to process.                                                                                                                 |
| <b>Threads</b>               | Degree of parallel processing desired. Values up to 3 may help speed processing. Default is 1.                                                                                                                                                                                                                                                      |
| <b>Enable trigger output</b> | If selected, this option causes the tool to sprout a T "Trigger" connector. A single record will be sent to this connection after all operations are complete. This can be used to sequence other operations by connecting to downstream tools that support a Trigger input connector. See <a href="#">Trigger input and output</a> <sup>59</sup> . |

#### 6.12.2 Configure the MDM Set Match Groups tool

Before configuring an MDM Set Match Groups tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>56</sup>.

##### To configure the MDM Set Match Groups tool

1. Select the **MDM Set Match Groups** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>56</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. Select a **Match type**.
6. Select a **Primary key input field**.
7. Select a **Group ID input field**.
8. Select a **Score field**.
9. Optionally, configure processing options:
  - Change **Block size** from the default value of 100.

- Configure **Use shared changeset**, **Wait for changeset commit**, and **Commit wait seconds** options to specify how the changeset is handled.
- To enable parallel setting of match groups, specify **Threads** as a value greater than **1**.
- To support sequencing of later tools after setting match groups is complete, select **Enable trigger output**.

10. Set any desired [execution options](#)<sup>57</sup>.

6.13

## MDM Generate Match Group IDs tool

This tool gets the next available match group IDs for a table and match level. Use this tool ahead of the [MDM Set Match Groups tool](#)<sup>76</sup> to ensure that you are using non-conflicting and unique match group IDs that are guaranteed not to collide with any other match process.

6.13.1

### MDM Generate Match Group IDs tool parameters

The MDM Generate Match Group IDs tool has a single set of parameters in addition to the standard execution options: *Configuration*.

#### Configuration

|                             |                                                                                         |
|-----------------------------|-----------------------------------------------------------------------------------------|
| <b>Override</b>             | See <a href="#">MDM tool shared settings</a> <sup>57</sup> .                            |
| <b>Database</b>             | The name of the MDM database to query.                                                  |
| <b>Table</b>                | The name of the table to query.                                                         |
| <b>Detailed logging</b>     | If selected, sends detailed activity logs to the Message Viewer.                        |
| <b>Match type</b>           | The match type for this workflow (for example, <b>individual</b> or <b>household</b> ). |
| <b>ID count field</b>       | Match group ID field for the given table.                                               |
| <b>Set minimum ID value</b> | Generate match group IDs, starting at a given minimum value.                            |
| <b>Minimum ID value</b>     | The minimum ID value that will be generated.                                            |

6.13.2

### Configure the MDM Generate Match Group IDs tool

Before configuring an MDM Generate Match Group IDs tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>56</sup>.

### To configure the MDM Generate Match Group IDs tool

1. Select the **MDM Generate Match Group IDs** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>56</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. Select a **Match type**.
6. Select an **ID count field**.
7. Optionally, you may select **Set minimum ID** value and enter a **Minimum ID value**.
8. Set any desired [execution options](#)<sup>57</sup>.

6.14

## MDM Match Group Query tool

This tool reads records from the MDM database that are members of the list of match groups read from the input. It is useful for querying the existing members of match groups in anticipation of rematching triggered by identifying information changes.

6.14.1

### MDM Match Group Query tool parameters

The MDM Match Group Query tool has two sets of parameters in addition to the standard execution options: *Configuration* and *Options*.

#### Configuration

|                         |                                                                                                                                                                                                                                               |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Override</b>         | See <a href="#">MDM tool shared settings</a> <sup>57</sup> .                                                                                                                                                                                  |
| <b>Database</b>         | The name of the MDM database to query.                                                                                                                                                                                                        |
| <b>Table</b>            | The name of the table to query.                                                                                                                                                                                                               |
| <b>Detailed logging</b> | If selected, sends detailed activity logs to the Message Viewer.                                                                                                                                                                              |
| <b>Select data from</b> | Specifies which version of the data to query: <ul style="list-style-type: none"> <li>• <b>Current</b>: the most recently-committed data.</li> <li>• <b>Timestamp</b>: the latest changeset committed before a specified timestamp.</li> </ul> |

|                                         |                                                                                                              |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------|
|                                         | <ul style="list-style-type: none"> <li>• <b>Pending changeset:</b> a specified pending changeset.</li> </ul> |
| <b>Timestamp</b>                        | If <b>Select data from</b> is <b>Timestamp</b> , specifies the timestamp.                                    |
| <b>Changeset</b>                        | If <b>Select data from</b> is <b>Pending changeset</b> , specifies the changeset.                            |
| <b>Select input match group ID</b>      | The field that contains the match group ID values.                                                           |
| <b>Match type</b>                       | The match type for this workflow (for example, <b>individual</b> or <b>household</b> ).                      |
| <b>Include group ID field on output</b> | Add a field to the output record that contains the record's match group ID value.                            |

## Options

|                                   |                                                                                                                                                                                             |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Return complete match info</b> | Select to add a JSON document field to the output record containing match group IDs, match scores, and user overrides for all match types.                                                  |
| <b>Block size</b>                 | Maximum number of records per transaction/request. Default is <b>100</b> . This is a tuning parameter. Generally, you can make this larger for narrow records and smaller for wide records. |

### 6.14.2 Configure the MDM Match Group Query tool

Before configuring an MDM Match Group Query tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>56</sup>.

#### To configure the MDM Match Group Query tool

1. Select the **MDM Match Group Query** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>56</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. On the **Select data from** dropdown, choose the desired view of the data:

- **Current:** The most recently-committed data.
  - **Timestamp:** Data committed on or before the defined timestamp. If you select this option, use the **Timestamp** data picker to choose a specific timestamp.
  - **Pending changeset:** Data from the selected changeset. If you select this option, select a **Changeset** from the dropdown.
6. Select the **input match group ID** field.
  7. Select a **Match type**.
  8. Optionally, select **Include group ID field on output**.
  9. Optionally, click **Refresh sample data** to view a sample of the input data.
  10. Optionally, click the **Options** tab and configure processing options:
    - Select **Return complete match info** to return a JSON document containing match information for each record.
    - Change **Block size** from the default value of **100**.
  11. Set any desired [execution options](#)<sup>57</sup>.

6.15

## MDM Match Suppression Query tool

This tool queries the suppression information to support RedPoint Data Management-based matching:

- **Never-match overrides:** This is a series of primary-key pairs. These records must never be matched, and should not end up in the same group.
- **Always-match overrides:** This is a series of primary-key pairs. These records must always be matched and end up in the same group. In theory there should be no conflict between these and the never-match overrides, but if there is the never-match overrides take precedence.
- **In match review:** This is a series of primary-key values. These records should not be processed at all, because they are part of an ongoing match review workflow.

6.15.1

### MDM Match Suppression Query tool parameters

The MDM Match Suppression Query tool has two sets of parameters in addition to the standard execution options: *Configuration* and *Options*.

#### Configuration

|                 |                                                              |
|-----------------|--------------------------------------------------------------|
| <b>Override</b> | See <a href="#">MDM tool shared settings</a> <sup>57</sup> . |
| <b>Database</b> | The name of the MDM database to query.                       |
| <b>Table</b>    | The name of the table to query.                              |

|                             |                                                                                                                                          |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Detailed logging</b>     | If selected, sends detailed activity logs to the Message Viewer.                                                                         |
| <b>Description equals</b>   | Filter changesets by text contained in the changeset description. In this case, the string value must be an exact match.                 |
| <b>Description contains</b> | Filter changesets by text contained in the changeset description. In this case, the string value must be contained within a description. |

## Options

|                             |                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Enable trigger input</b> | If selected, this option causes the tool to sprout a T "Trigger" connector. A single record will be sent to this connection after all operations are complete. This can be used to sequence other operations by connecting to downstream tools that support a Trigger input connector. See <a href="#">Trigger input and output</a> <sup>59</sup> . |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 6.15.2 Configure the Match Suppression Query tool

Before configuring an MDM Match Suppression Query tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>56</sup>.

#### To configure the MDM Match Suppression Query tool

1. Select the **MDM Match Suppression Query** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>56</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. Optionally, you may filter changesets by text contained in the changeset description by entering a string value in the **Description equals** or **Description contains** box.
6. Optionally, click the **Options** tab and configure processing options:  
Add a T input connector by clicking **Enable trigger input**. For more information, refer to [Trigger input and output](#)<sup>59</sup>.
7. Set any desired [execution options](#)<sup>57</sup>.

## 6.16 MDM Start Match Review Workflow tool

This is an all-in-one tool for supporting match-override-review workflows. The tool assumes that the last-committed changeset contains the "accepted" match groups. These match groups are typically created by an upstream [MDM Set Match Groups tool](#)<sup>[76]</sup>. You can define the accepted match groups and initiate the review workflows in the same project if you set the tool priorities correctly.

### 6.16.1 MDM Start Match Review Workflow tool parameters

The MDM Start Match Review Workflow tool has a single set of configuration parameters in addition to the standard execution options: *Configuration*

#### Configuration

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Override</b>                  | See <a href="#">MDM tool shared settings</a> <sup>[57]</sup> .                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Database</b>                  | The name of the MDM database to query.                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Table</b>                     | The name of the table to query.                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Detailed logging</b>          | If selected, sends detailed activity logs to the Message Viewer.                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Workflow</b>                  | The review workflow to start.                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Match type</b>                | The match type for this workflow (for example, <b>individual</b> or <b>household</b> ).                                                                                                                                                                                                                                                                                                                                                     |
| <b>Review set ID field</b>       | This identifies the "review set", which will become a single workitem for review. Often, this consists of a single "loose" match group, that can be assumed to represent the consolidation of two or more "tight" match groups or singletons. However, there are corner cases where this logic isn't sufficient. For correct behavior, this ID should be the result of AO Associate Match IDs run to combine the loose and tight group IDs. |
| <b>Primary key field</b>         | The record primary key field, which must be an <b>integer</b> or <b>long</b> .                                                                                                                                                                                                                                                                                                                                                              |
| <b>Group ID field (proposed)</b> | The loose group ID that is being proposed, relative to the accepted tight group IDs.                                                                                                                                                                                                                                                                                                                                                        |
| <b>Score field</b>               | The score given to the entire review set, if it is accepted. Only the first score of each review set                                                                                                                                                                                                                                                                                                                                        |

|                                  |                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                  | is processed, although they should all be the same value within a review set.                                                                                                                                                                                                                                                                       |
| <b>Wait for changeset commit</b> | Select this option to wait for the changeset to commit at the end of the tool's processing. If not selected, the changeset commit will be requested but the tool/project will not wait for it to complete.                                                                                                                                          |
| <b>Commit wait seconds</b>       | Time to wait for changeset commit to complete. If the commit does not complete within this time, the project fails and aborts; however, the commit will continue to process.                                                                                                                                                                        |
| <b>Threads</b>                   | Degree of parallel processing desired. Values up to 3 may help speed processing. Default is 1.                                                                                                                                                                                                                                                      |
| <b>Enable trigger output</b>     | If selected, this option causes the tool to sprout a T "Trigger" connector. A single record will be sent to this connection after all operations are complete. This can be used to sequence other operations by connecting to downstream tools that support a Trigger input connector. See <a href="#">Trigger input and output</a> <sup>59</sup> . |

#### 6.16.2 Configure the MDM Start Match Review Workflow tool

Before configuring an MDM Start Match Review Workflow tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>56</sup>.

##### To configure the MDM Start Match Review Workflow tool

1. Select the **MDM Start Match Review Workflow** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>56</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. Select a **Workflow**.
6. Select a **Match type**.
7. Select a **Review set ID field**.
8. Select a **Primary key field**.

9. Select a **Group ID field (proposed)**.
10. Select a **Score field**.
11. Optionally, configure processing options:
  - Configure the **Wait for changeset commit** and **Commit wait seconds** options to specify how the changeset is handled.
  - To enable parallel processing, specify **Threads** as a value greater than **1**.
  - To support sequencing of later tools after starting the match review workflow is complete, select **Enable trigger output**.
12. Set any desired [execution options](#)<sup>57</sup>.

6.17

## MDM Diff Report tool

This tool generates a report of changes contained in the changeset shared by other tools that have the **Use shared changeset** option enabled.

6.17.1

### MDM Diff Report tool parameters

The MDM Diff Report tool has a single set of parameters in addition to the standard execution options: *Configuration*

#### Configuration

|                                                 |                                                                  |
|-------------------------------------------------|------------------------------------------------------------------|
| <b>Override</b>                                 | See <a href="#">MDM tool shared settings</a> <sup>57</sup> .     |
| <b>Database</b>                                 | The name of the MDM database to query.                           |
| <b>Table</b>                                    | The name of the table to query.                                  |
| <b>Detailed logging</b>                         | If selected, sends detailed activity logs to the Message Viewer. |
| <b>Return full record for delete operations</b> | If a record is to be deleted, return the full record.            |
| <b>Return old record for modify operations</b>  | If a record is to be modified, return the old record.            |
| <b>Return new record for modify operations</b>  | If a record is to be modified, return the new record.            |
| <b>Return field diffs for modify operations</b> | If a record is to be modified, return a list of field diffs.     |
| <b>Return match group IDs</b>                   | Return the match group IDs for all the match groups involved.    |

|                               |                                                                                                                   |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <b>Return match documents</b> | Select to return JSON documents containing match group IDs, match scores, and user overrides for all match types. |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------|

#### 6.17.2 Configure the MDM Diff Report tool

Before configuring an MDM Diff Report tool, you should have an MDM database connection defined in [tool connection settings](#)<sup>56</sup>.

##### To configure the MDM Diff Report tool

1. Select the **MDM Diff Report** tool, and then click the **Configuration** tab on the Properties pane.
2. Optionally, override [shared settings](#)<sup>56</sup>: open the **Connection settings** section, click **Override**, and then specify new values for the tool.
3. Click **Connect/Refresh** to validate the connection and populate the database list.
4. Select a **Database** and **Table**.
5. Optionally, configure processing options:
  - Select **Return full record for delete operations**.
  - Select **Return old record for modify operations**.
  - Select **Return new record for modify operations**.
  - Select **Return field diffs for modify operations**.
  - Select **Return match group IDs**.
  - Select **Return match documents**.
6. Set any desired [execution options](#)<sup>57</sup>.

7

## Reference

This section contains technical details on MDM's data models, matching, views and subscriptions, workflow, and support for GDPR/privacy mandates.

7.1

### Databases and data models

Currently, MDM only supports MongoDB NoSQL databases.

The [table options](#)<sup>88</sup>, [field options](#)<sup>89</sup>, and [field data types](#)<sup>90</sup> you configure in MDM collectively define your data model.

#### Note

In a well-formed database, the **system** user, or any user who is a member of any group that has the DBA role, can modify schemas.

##### 7.1.1 Table options

When you create a table in MDM, the following options are available:

| Table Option               | Details                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Type                       | <p><b>Versioned</b> tables have a primary key and retain historical versions of data. See <a href="#">Versioned and time-series tables</a><sup>91</sup> for more information.</p> <p><b>Timeseries</b> tables store transactional or event data, and share the primary key of an associated versioned table.</p>                                                                                                                                                                                                                                                                                                                                                                   |
| Primary Key                | <p><i>Required if Type is Versioned.</i> Field that holds the primary key value, which is used to uniquely identify records.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Sequence Field             | <p><i>Optional.</i> If <b>Type</b> is <b>Versioned</b>, you can select this to prevent overwriting of new data with old data. If enabled, replace and upsert operations will not create new versions of records where the sequence field would be smaller than the previous sequence field. The modify operation will also honor sequence semantics if the sequence field is part of the modification.</p> <p>The primary motivation for using a sequence value is when a service has multiple asynchronous sources where only the latest update is applied, and if updates come in out of order, the "previous" updates are not applied since they would just be overwritten.</p> |
| Enable DM Matching Support | <p><i>Optional.</i> If <b>Type</b> is <b>Versioned</b>, you can select this to enable matching operations in RedPoint Data Management. For information concerning RedPoint Data Management matching, refer to the <i>RedPoint Data Management User Guide</i>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Description                | <p><i>Optional.</i> Description of the table.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

### *Note*

Names in MDM are case-preserving but case-insensitive.

The simplified regular expression (*regex*) for a name is:

`[_a-zA-Z][_a-zA-Z0-9]*`

The formal definition is:

$^[\backslash\p{L}_-][\backslash\p{L}\backslash\p{Nd}_-]^*\$$

which supports the full unicode letter set, allowing tables and field names such as:

E?ñ

### 7.1.2 Field options

When you add a field to a table in MDM, the following options are available:

| Field Option    | Details                                                                                                                                                                                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field</b>    | Field name. Each field in a table must be unique. A name must start with an alphabetic character or underscore and all remaining characters must be alphanumeric or underscore. MDM treats <b>Field</b> in a case-insensitive manner.                                                        |
| <b>Nullable</b> | If selected, <b>Field</b> may contain null or empty values.                                                                                                                                                                                                                                  |
| <b>PII</b>      | If selected, <b>Field</b> contains Personally Identifiable Information (PII). For more information on PII and data visibility in MDM, refer to <a href="#">GDPR and privacy support</a> <small>[122]</small> .                                                                               |
| <b>Index</b>    | If selected, the field will be indexed (a table index will be added for <b>Field</b> in the database), which reduces query search times.<br><br>Be aware that indexing many fields may slow insert operations on the table. You should enable indexing for fields you expect to query often. |
| <b>Display</b>  | If selected, <b>Field</b> will be displayed in data views. Note that fields that are not display-enabled cannot be queried.                                                                                                                                                                  |
| <b>Type</b>     | See the <a href="#">Field data types</a> <small>[90]</small> table for a list of permissible field types in MDM.                                                                                                                                                                             |

| Field Option | Details                                                                                                            |
|--------------|--------------------------------------------------------------------------------------------------------------------|
| Size         | If <b>Type</b> is <b>text</b> , <b>integer</b> , or <b>float</b> , the size of the field (in characters or bytes). |
| Signed       | If <b>Type</b> is <b>integer</b> , whether the field is signed.                                                    |

### **Note**

Names in MDM are case-preserving but case-insensitive.

The simplified regular expression (*regex*) for a name is:

[ a-zA-Z][ a-zA-Z0-9]\*

The formal definition is:

`^[\\"p{L}_][\"p{L}\\"p{Nd}_]*$`

which supports the full unicode letter set, allowing tables and field names such as:

4

### 7.1.3 Field data types

When you add a field to a table in MDM, the following field data types are available:

| Field Type     | Details                                                                                                                                                                                                                                                                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>text</b>    | <b>String</b> consisting of Unicode characters. Size may be 1 to 1,000,000. <b>Null</b> and "" are treated as the same value.                                                                                                                                                                                                                   |
| <b>money</b>   | Fixed-precision number with four decimal places, with an effective range of -922,337,203,685,477.5808 to +922,337,203,685,477.5807.                                                                                                                                                                                                             |
| <b>integer</b> | Size is either 1, 2, 4, or 8 bytes. Signed is <b>true</b> or <b>false</b> . The combination { "size":8, "signed":false } is not allowed due to restrictions in the underlying database.                                                                                                                                                         |
| <b>float</b>   | Floating-point number. <ul style="list-style-type: none"><li>• If <b>Size</b> is 4 bytes, the range of values that can be stored is -3.402823466e+38 to 3.402823466e+38. The smallest allowable value is 1.175494351e-38F.</li><li>• If <b>Size</b> is 8 bytes, the range of values that can be stored is -1.7976931348623158e+308 to</li></ul> |

| Field Type                                    | Details                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                               | 1.7976931348623158e+308. The smallest allowable value is 2.2250738585072014e-308.                                                                                                                                                                                                                                                                                                                                          |
| <b>date</b><br><b>time</b><br><b>datetime</b> | <b>Date</b> , <b>Time</b> , and <b>DateTime</b> are all represented as text using 24-hour ISO formatting. <b>DateTime</b> is a local date/time without an explicit timezone, and is interpreted according to the usage context. It should be of the form: <ul style="list-style-type: none"> <li>• Date: <b>yyyy-MM-dd</b></li> <li>• Time: <b>HH:mm:ss</b></li> <li>• DateTime: <b>yyyy-MM-ddTHH:mm:ss.SSS</b></li> </ul> |
| <b>boolean</b>                                | Text characters or strings used as <b>True</b> and <b>False</b> values (0/1, true/false, T/F).                                                                                                                                                                                                                                                                                                                             |
| <b>table</b>                                  | See <a href="#">Table options</a> <sup>88</sup> .                                                                                                                                                                                                                                                                                                                                                                          |
| <b>linkto</b>                                 | See <a href="#">Linkage fields</a> <sup>94</sup> for more information.                                                                                                                                                                                                                                                                                                                                                     |

#### 7.1.4 Versioned and time-series tables

MDM supports the following kinds of tables:

- [Versioned tables](#)<sup>91</sup>
- [Time-series tables](#)<sup>92</sup>

##### 7.1.4.1 Versioned tables

*Versioned tables* are central to MDM and are used for the creation and maintenance of a *golden record*, typically comprised of entity data.

- All data or metadata that can be manipulated directly by MDM users are stored in versioned tables.
- Versioned tables have a primary key.
- Every metadata or data create/modify/delete operation performed on a versioned table is retained. See [Historical data model](#)<sup>95</sup> for more information.
- Every version of every record and table is associated with a changeset. See [Changesets](#)<sup>95</sup> for more information.
- You can view snapshots of the data at any point in time. See [Data view](#)<sup>99</sup> for more information.

#### 7.1.4.2 Time series tables

*Time series tables* can be thought of as child tables of versioned tables. They support the capture of many small timestamped records representing transactional or event data. Examples of such data are purchases, website visits, and instrumentation data. Time series tables are accessed more quickly than versioned tables and are useful when you need to insert large volumes of data. Time series tables lend themselves to views, which are lists of aggregation operations grouped by primary key and filtered by timestamp bounds (see [Views and subscriptions](#)<sup>[111]</sup> for more information).

- A time series table is associated with a versioned table. For example, `customer_transactions` would be associated with `customers`.
- A time series table shares the primary key of its associated versioned table, which is defined using a **link to** field type.
- A time series table has a timestamp field, which you can query to generate a historic view of the data.
- A time series table may not contain nested tables or structured fields.
- You cannot update time series data—only insert and append operations are supported.

## 7.2

### Nested tables and linkage fields

In MDM you can link a table field to either a sub-table or a field in another table. Sub-tables are called [nested tables](#)<sup>[92]</sup>, and linked fields are called [linkage fields](#)<sup>[94]</sup>.

Because of their simpler nature, we recommend that you use nested tables instead of linkage fields whenever possible. However, if you have static reference data that can be shared among multiple tables or multiple records, use linkage fields.

#### 7.2.1

### Nested tables

In MDM tables you can define a **table** field type that contains an array of sub-records. In effect, a **table** field references a sub-table within a table.

- This is the recommended method for referencing live data.
- Sub-documents don't have a life independent of the containing document.
- Sub-documents can't be shared (linked to) by other documents or other tables.
- Use cases:
  - addresses
  - email addresses
  - phone numbers
  - business codes

For example, you might have a list of addresses that you want to store within a customer table. You could create an **addresses** field of type **table** and add a different address string value to each individual sub-record. By doing this, you're actually storing an array of sub-documents in your customer document.

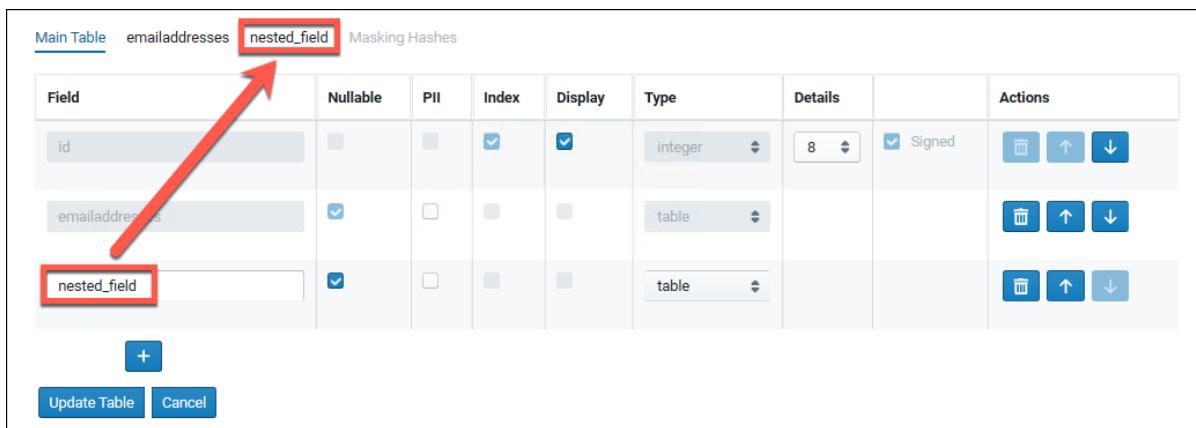
#### Note

We recommend that you limit the number of records in a nested table to less than 100, and overall record size to less than 10 KB. No record (including its sub-records) can exceed 2 MB in size. Using very large documents may cause problems for web services and browsers. See <https://docs.mongodb.com/manual/reference/limits/#bson-documents> for the current design limits of these applications and services.

#### 7.2.1.1 Add a nested field to a table

##### To add a nested field to a table

1. In the MDM Web App, select **Data Hub > Tables**.
2. Locate the desired table and click its **Schema** button .
3. Click  to add a field.
  - a. Enter a field name.
  - b. Select the field attributes (be sure to select the type **table**).  
The new field name appears as a tab at the top of the table grid.



| Main Table                                  | emailaddresses | nested_field                        | Masking Hashes           |                                     |                                     |         |                                  |                                            |                                                                                                                                                                                                                                                                   |
|---------------------------------------------|----------------|-------------------------------------|--------------------------|-------------------------------------|-------------------------------------|---------|----------------------------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                             |                |                                     |                          |                                     |                                     |         |                                  |                                            |                                                                                                                                                                                                                                                                   |
| Field                                       |                | Nullable                            | PII                      | Index                               | Display                             | Type    | Details                          |                                            | Actions                                                                                                                                                                                                                                                           |
| <input type="text" value="Id"/>             |                | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | integer | <input type="button" value="8"/> | <input checked="" type="checkbox"/> Signed |    |
| <input type="text" value="emailaddresses"/> |                | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | table   | <input type="button" value="8"/> |                                            |    |
| <input type="text" value="nested_field"/>   |                | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | table   | <input type="button" value="8"/> |                                            |    |

    **Update Table**    **Cancel**

A table nested field

4. Click the new field tab.
5. Click  to add a sub-field.
  - Enter a sub-field name.
  - Select the sub-field attributes.

| Main Table | emailaddresses                      | nested_field             | Masking Hashes           |                                     |      |         |                                                      |
|------------|-------------------------------------|--------------------------|--------------------------|-------------------------------------|------|---------|------------------------------------------------------|
| Field      | Nullable                            | PII                      | Index                    | Display                             | Type | Details | Actions                                              |
| home       | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | text | 10      | <span>trash</span> <span>up</span> <span>down</span> |
| cell       | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | text | 10      | <span>trash</span> <span>up</span> <span>down</span> |
| business   | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | text | 10      | <span>trash</span> <span>up</span> <span>down</span> |

**Nested field sub-fields**

- Click **Update Table** to finish.

#### 7.2.2 Linkage fields

In MDM tables you can define a **link to** (linkage) field type that contains the primary key of another table. You can also display a different field from the reference table other than the primary key. For example, you could link on country abbreviation ("FSM") but display the country's name ("Federated States of Micronesia").

The target field must be of the same type as the linked table's primary key, and both the linked table and the displayed field must exist.

- Recommended method for referencing:
  - Static reference data.
  - Entities that are related to the primary table (for example, a patient table might link to a table of primary care providers).
- Linked-to fields have a life independent of the linking document.
- Linked-to fields can be shared by other documents or other tables.
- Use cases:
  - transactions
  - history
  - friendship graphs
  - Static reference data such as North American Industry Classification System (NAICS) codes and lists of states or countries

#### Note

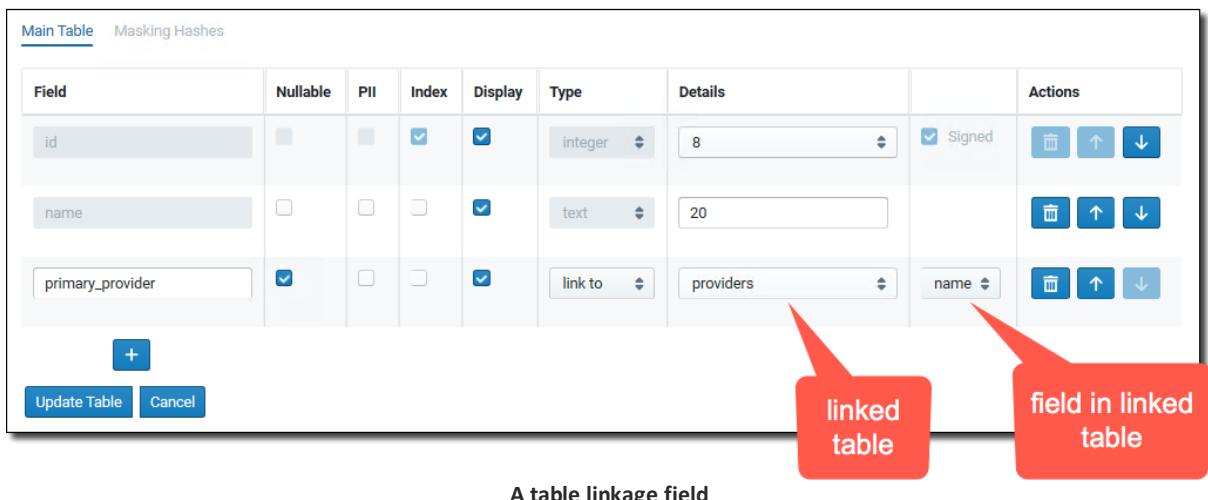
We recommend that you *do not* use linkage fields to decompose entity data into a normal form and spread it out across multiple tables. MDM uses a document database that makes using

contained sub-tables and sub-records preferable to using separate daughter tables and foreign key linkages to those sub-records.

#### 7.2.2.1 Add a linkage field to a table

##### To add a linkage field to a table

1. In the MDM Web App, select **Data Hub > Tables**.
2. Locate the desired table and click its **Schema** button .
3. Click  to add a field.
  - a. Enter a field name.
  - b. Select the field attributes (be sure to select the type **link to**).
  - c. In the **Details** column, select the MDM table to which you want to link.
  - d. In the column to the right of the **Details** column, choose the field that you want to display. The final column is the "displayed" column, not the linked table primary key column. The schema tells MDM so the user does not have to.



Main Table Masking Hashes

| Field            | Nullable                            | PII                                 | Index                               | Display                             | Type    | Details   |                                            | Actions |
|------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|---------|-----------|--------------------------------------------|---------|
| id               | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | integer | 8         | <input checked="" type="checkbox"/> Signed |         |
| name             | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | text    | 20        |                                            |         |
| primary_provider | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | link to | providers | <input type="checkbox"/> name              |         |

**A table linkage field**

4. Click **Update Table** to finish.

#### 7.3 Historical data model

The following topics cover how MDM handles data throughout the lifespan of a table.

##### 7.3.1 Changesets

In MDM, the data change paradigm is *pending changes are isolated until committed*.

For example, when a user makes a change to a record, the database isn't immediately updated. Instead, MDM creates a *changeset*, and a changeset must be committed for the changes to be implemented.

Think of a changeset as basket for holding related changes, with the changes being anything you can do to the system. You can also think of a changeset as a unique view of the data (whatever data you started with before you opened the changeset, plus the changes you made).

Creating a new table creates a changeset. Modifying a field in the table schema creates another changeset. Adding records to the table creates yet another changeset.

There are two kinds of changesets:

- Those comprised of metadata changes (such as creating or modifying a table).
- Those comprised of data changes (such as updating fields in a record).

After creation, a changeset may be viewed, deleted, or committed. A data changeset must be explicitly committed by a user, while a metadata changeset is automatically committed by MDM.

*Once a changeset is committed, the changes can't be undone or rolled back.*

### Notes

A separate changeset list is maintained for every database.

Creating or deleting a database does not result in a changeset. Database deletion physically purges all tables and history!

#### 7.3.1.1 Example: metadata changeset (create a table)

If user "dba" creates a new table **testTableAlpha**, the changeset is automatically committed because it is a metadata changeset.

| ID | Description                          | Category  | Status | Create... | Committ... | Date Created        | Date Committed          | View Det... | Commit | Discard |
|----|--------------------------------------|-----------|--------|-----------|------------|---------------------|-------------------------|-------------|--------|---------|
| 79 | Create table testTableAlpha metadata | committed | dba    | dba       |            | 12/12/2018, 11:37:4 | 12/12/2018, 11:37:42 PM | 2           |        |         |

Metadata changesets are automatically committed

#### 7.3.1.2 Example: data changeset (add a record to a table)

If user "bizuser" adds a record to a table through an existing workflow, the changeset is not automatically committed because it is a data changeset. Note that the changeset is created by user "system" because all data changesets are created by MDM on behalf of the user.

| ID          | Description                                       | Category | Status  | Created By | Committed By | Date Created   | Date Commi...  | View Details              | Commit                 | Discard                 |
|-------------|---------------------------------------------------|----------|---------|------------|--------------|----------------|----------------|---------------------------|------------------------|-------------------------|
| 80          | Created by workflow 'New Customer', workitemId=83 | data     | pending | system     | system       | 12/13/2018, 12 | 12/13/2018, 12 | <a href="#">View Hist</a> | <a href="#">Commit</a> | <a href="#">Discard</a> |
| <hr/>       |                                                   |          |         |            |              |                |                |                           |                        |                         |
| Previous    |                                                   |          |         |            |              |                |                |                           |                        |                         |
| Page 4 of 4 |                                                   |          |         |            |              |                |                |                           |                        |                         |
| 20 rows ▾   |                                                   |          |         |            |              |                |                |                           |                        |                         |
| Next        |                                                   |          |         |            |              |                |                |                           |                        |                         |

When a user adds a new record, the data changeset is created by MDM on behalf of the user, but is not automatically committed

Once the record has been reviewed by user "steward" and approved, the changeset is committed. Note that the changeset is committed by "system" because all data changesets are committed by MDM on behalf of the user.

| ID       | Description          | Category  | Status | Created By | Committed By | Date Created   | Date Commi...  | View Details              | Commit | Discard |
|----------|----------------------|-----------|--------|------------|--------------|----------------|----------------|---------------------------|--------|---------|
| 80       | Created by work data | committed | system | system     | system       | 12/13/2018, 12 | 12/13/2018, 12 | <a href="#">View Hist</a> |        |         |
| <hr/>    |                      |           |        |            |              |                |                |                           |        |         |
| Previous |                      |           |        |            |              |                |                |                           |        |         |

Once a new record has been approved by a data steward, the changeset is committed by user "system" (MDM) on behalf of the user

### 7.3.1.3 Conflict resolution

If multiple users (or systems) try to make a change to the same record(s) simultaneously, the mechanism for conflict resolution is as follows:

- If changes are compatible, the changes are merged automatically. For example, in one record a user changes the name, and another user changes the address.
- If changes are not compatible (for example, in one record a user changes the name to "Jane" and another user changes the name to "Mary Jane"), a *conflict resolution policy* is used. While somewhat configurable, generally this is:
  - For ETL operations, the latest change wins.
  - For interactive operations via workflow or the data hub, this results in a *conflict state*, which must be manually resolved.

### 7.3.1.4 Viewing changesets

You can view a list of a table's changesets (or a subset) from the **Data Hub > Changesets** page.

## Changesets

The screenshot shows a table of changesets with the following columns: ID, Description, Category, Status, Created By, Committed By, Date Created, Date Committed, View Details, Commit, and Discard. The table contains 10 rows of data. At the bottom, there are navigation buttons for Previous (labeled 12), Page 4 of 4 (labeled 13), 20 rows (labeled 14), and Next (labeled 12).

| ID | Description          | Category | Status    | Created By  | Committed By | Date Created    | Date Committed  | View Details              | Commit                                     | Discard                 |
|----|----------------------|----------|-----------|-------------|--------------|-----------------|-----------------|---------------------------|--------------------------------------------|-------------------------|
| 70 | Created by work data |          | pending   | system      |              | 12/12/2018, 12: |                 | <a href="#">View Hist</a> | <input checked="" type="checkbox"/> Commit | <a href="#">Discard</a> |
| 71 | Created by work data |          | committed | system      | system       | 12/12/2018, 8:0 | 12/12/2018, 8:0 | <a href="#">View Hist</a> |                                            |                         |
| 72 | Created by work data |          | pending   | system      |              | 12/12/2018, 8:0 |                 | <a href="#">View Hist</a> | <input checked="" type="checkbox"/> Commit | <a href="#">Discard</a> |
| 73 | Created by work data |          | pending   | system      |              | 12/12/2018, 8:1 |                 | <a href="#">View Hist</a> | <input checked="" type="checkbox"/> Commit | <a href="#">Discard</a> |
| 80 | Created by work data |          | committed | system      | system       | 12/13/2018, 12: | 12/13/2018, 12  | <a href="#">View Hist</a> |                                            |                         |
| 81 | Created by work data |          | pending   | system      |              | 12/13/2018, 12: |                 | <a href="#">View Hist</a> | <input checked="" type="checkbox"/> Commit | <a href="#">Discard</a> |
|    |                      |          | Previous  | Page 4 of 4 | 20 rows      |                 |                 | Next                      |                                            |                         |

The Data Hub > Changesets page displays a list of a table's changesets

### To view a table's changesets

1. **Filter**—Use the pull-down menu to filter the changeset table according to changeset status (pending, queued, committed, and so on). Click **Show Advanced Filter Options** to expand the **Filter** section and display more options:

The dialog box contains the following fields:

- Filter by:** A dropdown menu set to "All".
- Hide Advanced Filter Options** button.
- Between dates:** Two input fields labeled "Start date" and "End date".
- Description contains:** An input field labeled "b".
- Category:** A dropdown menu set to "All".

Dialog used to filter table changesets

- a. **Between dates**—Click inside the **Start date** and **End date** boxes to display a date picker popup. The popup also gives you time period options such as **today**, **past week**, **past year**, and so on.
- b. **Description contains**—Filter changesets by text contained in the changeset description.

- c. **Category**—Filter changesets by category (**data or metadata**). Changes involves adding, deleting, or modifying records are **data** changes. Changes that are not record-specific (such as adding a database, adding a table, or modifying a table schema) are **metadata** changes.
- 2. **ID**—Unique identifier assigned by MDM upon changeset creation.
- 3. **Description**—Short description of the changeset created by MDM.
- 4. **Category**—Values are **data** and **metadata**.
- 5. **Status**—Values are **pending**, **committed**, **error**, **committing**, and **queued**. **Queued** and **committing** are transitory states as a changeset goes from **pending** to **committed**. Sometimes a changeset will go into the **error** status if the changes cannot be resolved because of conflicting changes made and committed by other users.
- 6. **Created by**—Which user created the changeset. All data changesets are created by **system** on behalf of the user.
- 7. **Committed by**—Which user committed the changeset. All data changesets are committed by **system** on behalf of the user.
- 8. **Date Created / Date Committed**—The dates the changeset was created and committed, respectively.
- 9. **View Details**—Click a changeset's **View History** link to display the changeset's **table**, **operation**, and record **count**.
- 10. **Commit**—If the changeset has not already been committed, you can **Commit** it here.
- 11. **Discard**—If the changeset has not already been committed, you can **Discard** it here.
- 12. **Previous / Next**—Click **Previous** or **Next** to display additional changeset pages.
- 13. **Page**—Which page is currently displayed.
- 14. **Rows**—Set the number of table rows displayed per page.

### 7.3.2

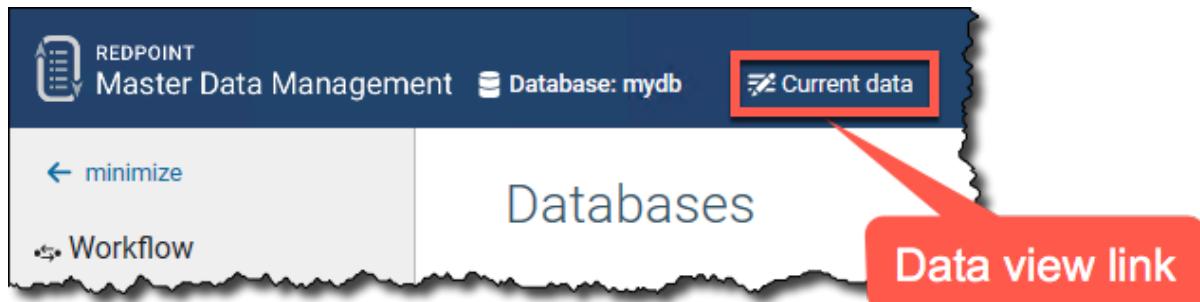
### Change history

From the moment a table is created, every change made to that table is recorded by a changeset. It follows that every table record has a change history, and you can [View that history](#) .

### 7.3.3

### Data view

The data view link at the top of the Web App describes the type of data currently displayed.



The data view link describes the type of data currently displayed

The data views are:

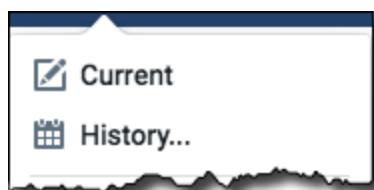
- [View current data](#)
- [View historical data](#)
- [Data inside an open changeset](#)

#### 7.3.3.1 View current data

**Current data** is the default data view.

To view current data if historical or changeset data is being displayed

1. On any Web App page, click the data view link.
2. On the menu, select **Current**. You are now viewing current table data.



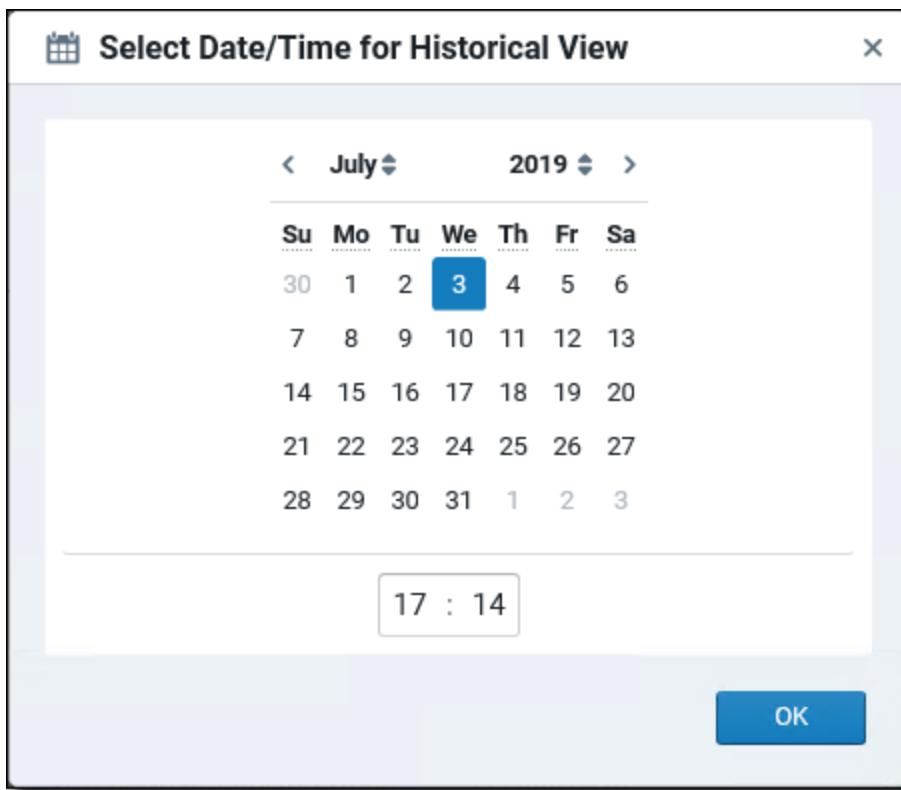
Select Current to view current table data

#### 7.3.3.2 View historical data

Since every record in a table has a change history, you can pick a historical date and view the state of the table at that point in time.

To view historical data

1. On any Web App page, click the data view link.
2. On the menu, select **History**.



Use the date picker to view table data from a specific date

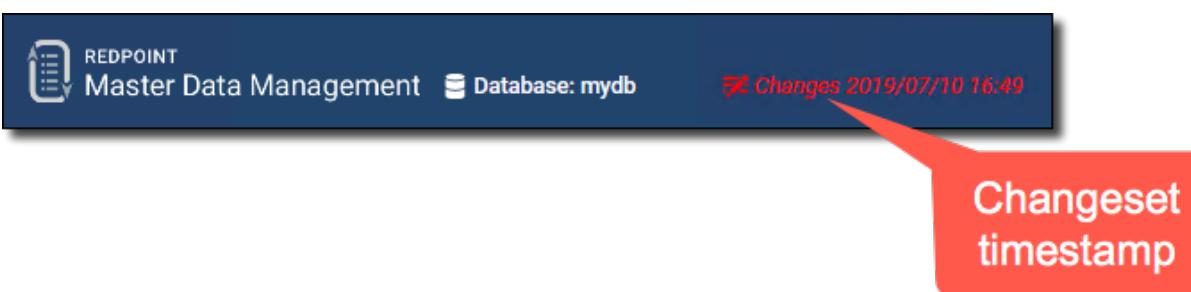
3. Use the date picker to choose a date (you can also select a time of day in 24-hour (**hh:mm**) format) and click **OK**.
4. You are now viewing table data from the chosen date. The data view link now displays the date from which the data was taken:



When MDM displays historical data, the data view link displays the date from which the data was taken

#### 7.3.3.3 Changeset data

If a changeset is currently open and uncommitted, the data view link displays that status:



In MDM, metadata changes (database create/delete, table create/edit/delete) are accomplished through an established workflow, and the workflow automatically creates and commits its own changeset.

Lower-level access users make data changes (table record create/edit/delete ) through an established workflow, and the workflow automatically creates and commits its own changeset.

However, high-level access users (such as those in the **dbas** or **system** groups) can explicitly open a new changeset, make changes directly to data, and explicitly commit the changeset. Explicitly opening a new changeset and then closing it is done through the data view link.

After opening a new changeset, you can:

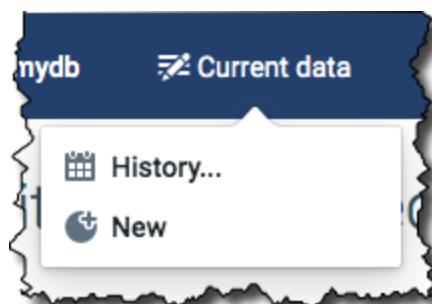
- Make direct changes to table data (outside of a workflow).
- Commit the changeset. All data changes made between opening and committing are included in the changeset.
- Discard the changeset (discard all data changes made after opening the changeset).
- Switch to another data view (current or historical). If you switch to a current or historical data view, the changeset remains open, and you can switch back to it by clicking the data view link and selecting **My uncommitted changes** on the menu.

#### 7.3.3.3.1 Open a new changeset with the data view link

Note that you can explicitly open only one changeset at a time.

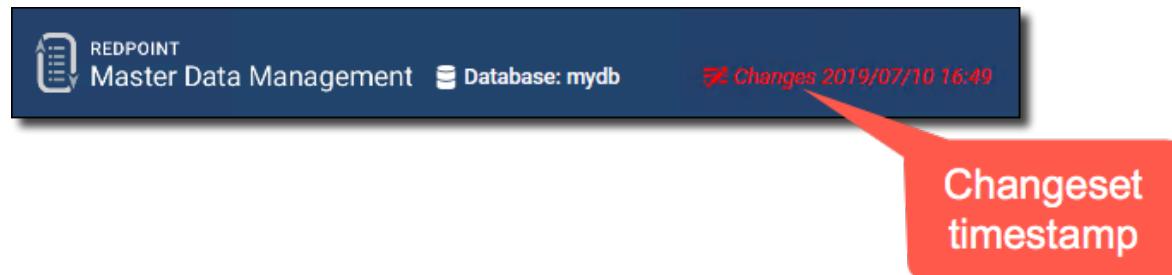
##### To open a new changeset with the data view link

1. On any Web App page, click the data view link.
2. On the menu, select **New**.



Select New to open a new changeset

3. A new changeset is created. Any changes you make beyond this point are included in the changeset. The data view link text changes to red, with the displayed date and time indicating when the changeset was opened:

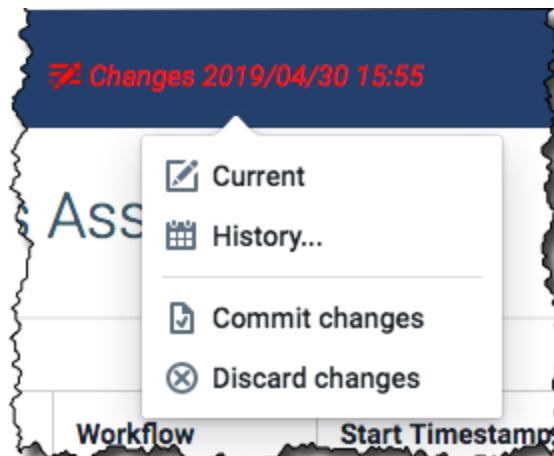


4. Make changes to the desired table. After opening a changeset, a table's data page will display a **Create Record** button and **Edit** and **Delete** buttons on each record row.

A screenshot of the REDPOINT Master Data Management software. The left sidebar includes options like "minimize", "Workflow", "Assigned to me", "Available", "Start New", "Started by me", "Data Hub", "Databases", "Tables", "Data", "Changesets" (which is selected and highlighted), "Views", and "Compliance". The main area is titled "Data" and shows a table for the "customers" table. The table has columns: COMPANY, ADDRESS, CITY, STATE, ZIP, PHONE, ID, emailaddress..., History, Edit, and Delete. There are three rows of data. Each row has "Edit" and "Delete" links at the end, which are highlighted with a red box. A "CREATE RECORD" button is located in the top right corner of the main area, also highlighted with a red box. A red timestamp box at the top right says "Changes 2019/05/03 11:39".

After opening a changeset, a table's data page displays a Create Record button, and Edit and Delete buttons on each record row

5. After making the desired table changes, click the data view link and select **Commit changes** or **Discard changes**.



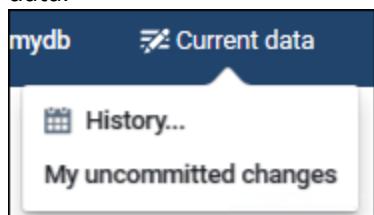
After making table changes, you can commit or discard the changes

Instead of immediately committing or discarding a changeset, you can also leave the changeset open and switch to the **Current** or a **History** view. If you switch to another view, the changeset will remain open until you switch back to it.

#### 7.3.3.3.2 Switch back to an open changeset

To switch back to an open changeset view from a current or a history view:

1. Click the data view link.
2. On the menu, select **My uncommitted changes**. You are now viewing the open changeset data.



3. At this point you can make further changes (and add to the changeset), commit the changeset, or discard the changeset.

#### 7.3.4 Versioned metadata

In MDM, table schemas are versioned. However, you cannot change a table schema in such a way as to render historical data invalid. Put another way, table representation continuity must be maintained for the entire table history. You can go from less representable to more representable, but not from more to less.

The rules for changing table metadata and maintaining valid historical data are:

- While you can add a new field, the new field must be able to accept a **null** value.  
If you look at a record's history, values of the added field before the field was added can simply be represented as null.
- You cannot change a field type to another type.

If you tried to change a field type from text to date, values from before the field change and after the field change could not be represented the same way. For example, the value "monday" cannot be represented as a date.

- Enlarging a field is allowed.  
A text field can be enlarged from 10 to 20 characters, because all field versions can be represented with 20 characters or less.
- Make a field value able to accept a **null** value is allowed.

Pre-modified fields can be represented as the original field type, while post-changed fields can be represented as the original field type or null.

- Deleted (dropped) fields are allowed.
  - Dropped fields still exist in historical data. After a field is dropped, all values can be represented as null.
  - A new field with the same name as a dropped field is distinct (treated as a different field).
  - Use a RedPoint Data Management ETL process to propagate and convert dropped fields to new fields.

7.4

## Integrating RedPoint Data Management and MDM

RedPoint Data Management supports a set of [connector tools](#)<sup>[56]</sup> that enable a variety of operations on MDM, including:

- Querying records with static query parameters
- Querying records that match a series of field values
- Loading new records or replacing existing records
- Bulk-updating records
- Bulk-deleting records
- Querying view results
- Support for various fuzzy-matching operations
- DBA operations such as creating and dropping tables

By using these connector tools, you can leverage the full power of RedPoint Data Management's ETL, hygiene, and matching capabilities. RedPoint Data Management is the preferred method for making batch changes to MDM.

In addition, you can create RedPoint Data Management-hosted web services using these same connector tools, which perform incremental queries from and changes to MDM. This is a powerful technique for exposing MDM as a data service, because it can support higher-level operations (such as "cleanse, match, and integrate new customer data") than can the raw MDM API.

For a detailed explanation of these tools, see [RedPoint Data Management's MDM tools](#)<sup>[56]</sup>.

7.4.1

### Settings and credentials

In order to connect RedPoint Data Management to MDM, you will need to know:

- The MDM authentication service URI. By default this is <http://localhost:9901/mdm>.
- The MDM data service URI. By default this is <http://localhost:12345/mdm>.

- The MDM username and password. The **system** username can perform all operations, and it is often used for ETL. The password for the system logon is initially set to **system** but this should be changed administratively.

You set the default and shared settings and credentials for MDM connector tools in the RedPoint Data Management repository under [/Settings/Tools](#), on the **MDM** tab. These settings can be overridden in each RedPoint Data Management MDM connector tool.

#### 7.4.2 Select databases and tables

In every RedPoint Data Management MDM connector tool, you must choose a database from the **Database** dropdown on the **Properties** pane. If this dropdown contains the message "ERROR\*\*\*", check and correct the settings and credentials. After selecting the database, most connector tools will also require that you select a **Table** on which to operate. Once you select the database and table, other settings are available and metadata will be automatically queried from MDM as needed.

#### 7.4.3 Query records

Use the MDM Input tool to read records from the MDM database. You can:

- Read an entire table.
- Query based on a simple form using field/value comparisons.

When querying records, you can also specify a timestamp to query historical data (for example, "the customer table at the end of last quarter").

#### 7.4.4 Query records matching input data

Use the MDM Key Query or MDM Dynamic Query tools to return all records matching a sequence of input values.

- MDM Key Query operates only on primary keys.
- MDM Dynamic Query supports queries on any field.

#### 7.4.5 Insert or replace records

Use the MDM Output tool to create, replace, or upsert records into the MDM database. Configure this tool for the desired **Mode**:

- **Create**: A record with a matching primary key must not already exist.
- **Replace**: A record with a matching primary key must already exist.
- **Upsert**: If a record with a matching primary key exists, it will be replaced (otherwise a new record will be inserted).

#### 7.4.6 Query views

Use the MDM Query View tool to return the results of a view in batch. It can read:

- Current data
- The latest changeset committed before a specified timestamp
- A specified pending changeset

#### 7.4.7 Extract historical data

Since every record in a table has a change history, it is possible to pick a historical date and view the state of the table at that time.

Use the MDM Input tool to read records from the MDM database. A changeset or timestamp can be specified to read from a specific point in history.

#### 7.4.8 Shared changesets

A typical RedPoint Data Management project may have several MDM connector tools that change the database, including:

- MDM Output tool
- MDM Update tool
- MDM Set Match Groups tool

All of these tools perform their operations on a *changeset*, which is automatically committed at the end of processing. However, each tool by default will create and commit its own changeset, isolating its changes from the changes of other tools in the project. Often, you will want all of the tools in a RedPoint Data Management project (or web service) to cooperate on a single set of changes. To enable that behavior, select the **Use shared changeset** option on all MDM tools in a project.

#### 7.4.9 Detect and process duplicates during load into MDM

When loading records into an MDM table, detecting record duplicates that already exist in the table gives you the option of deciding how you want to handle this situation.

1. If the input record does not have a unique primary key, use one of RedPoint Data Management's hashing functions (with a hashing algorithm such as MD5) to synthesize a primary key value from existing field values. Note that there are many ways a primary key value could be constructed; it is up to you to determine what makes the most sense for a given table.
2. If the input record does not have a sequence field (a timestamp or number), define a sequence field in the origin table that identifies a record version.
3. Use the MDM Key Query tool (with the embedded or constructed primary key value) to determine if the MDM table has a duplicate of the record being read.

4. If the record has no duplicate in the MDM table, use the MDM Output tool to insert the record.
5. If the record has a duplicate in the MDM table, you can:
  - Ignore the record just read.
  - Use the MDM Output tool to upsert the newly-read record into the MDM table. (The MDM Output tool recognizes sequence fields, and will automatically upsert a record in the MDM table if the record being read is the latest version.)

#### 7.4.10 Advanced matching

MDM matching is actually done in RedPoint Data Management. The following topics drill down into the mechanics of that process.

##### 7.4.10.1 Support for different match types

MDM supports multiple match types (for example, individual or household). Each match type supports match review in two modes:

- Review of automated decisions
- Manual review

By default, in either of these review modes, the match review workflow uses the match-type value **default**. A workflow can be configured to use a specific match-type value (such as **individual**).

Even if an MDM table supports multiple match types (for example, individual, household, and email), a workflow is bound to a single match type. A workflow cannot support multiple, specific match types at the same time.

##### 7.4.10.2 Match group IDs and match table primary keys

Match group IDs must be 8-byte integers. Even though match group ID isn't part of an MDM table schema definition, you can still extract them and set them via the RedPoint Data Management connector tools (and RedPoint Data Management deals only with 8-byte integer match IDs).

Also, you should create match tables with a primary key of type **integer**. Match table primary keys are stored in the **always** and **never** tables, and the RedPoint Data Management match tools deal only with 8-byte integer record IDs. If you create an MDM table that supports matching, things are easier if the primary key is of type **integer** (it avoids one additional key-mapping step in the project flow).

##### 7.4.10.3 Match-segmentation key fields

Match segmentation keys must be defined and managed between RedPoint Data Management and MDM as follows:

1. In the automatic matching logic you build in RedPoint Data Management, you define one or more segmentation keys. This is standard practice for RedPoint Data Management matching.

2. Segmentation keys are strings.
3. While RedPoint Data Management is entirely in charge of matching during the initial load, it must cooperate with MDM for incremental changes because new/modified records are potentially rematched against all match candidates and all records in the current match group.

To support this, you must define fields in your MDM table schema to hold the segmentation keys. This field should be indexed because:

- When RedPoint Data Management loads new records and matches them, it saves the segmentation keys along with the record data.
- When RedPoint Data Management processes incremental new/modified records, it calculates the segmentation keys for these new records and performs a dynamic query against the MDM table to pull all match candidates before doing the rematching. RedPoint Data Management also pulls the records for the existing match groups because they may also be affected and need to be rematched.

#### 7.4.10.4 Match information document

The MDM Input tool in RedPoint Data Management returns the defined fields (and any nested tables) in the MDM table schema by default. There is an option to return the match document as a JSON file that contains the match levels, scores, and overrides. This document is defined in [Matching sub-document](#)<sup>[153]</sup>.

#### 7.4.10.5 Handling match suppression

See [Automatic matching is performed using RedPoint Data Management/MDM integration](#)<sup>[109]</sup> for an explanation of how steward decisions are honored.

Call the MDM Match Suppression Query tool to retrieve the primary keys of never-match overrides, always-match overrides, and in-match-review-records.

- **Never-match overrides**—Add the never-match primary key pairs to the match macro itself (AO Business Match, for example).
- **Always-match overrides**—Treat these primary key pairs as hard matches. These key pairs go into a post-match process (AO Associate Match IDs).
- **In-match-review records**—Use a RedPoint Data Management join to exclude these records from matching.

#### 7.4.10.6 Automatic matching

Automatic matching is performed via RedPoint Data Management-MDM integration:

1. RedPoint Data Management executes automatic matching jobs, and you set match parameters inside RedPoint Data Management.
2. Match reviews are performed by humans (data stewards) in MDM.

When stewards do a match review, their decisions are stored in the **always** and **never** keys of the match document, and this records the user preferences. In order to keep RedPoint Data Management from rematching (or failing to match) combinations that the steward has set, the RedPoint Data Management MDM Match Suppression Query tool returns lists of "always match" and "never match" pairs of record IDs, as well as a list of "under review" record IDs.

#### **7.4.10.7 Batch matching**

In MDM, batch matching means either:

- Matching records in MDM with records from another source.
- Rematching all records already inside MDM.

In either case, load records first and then run the MDM matching process:

1. Read all records out of MDM.
2. Match.
3. Figure out new match group values and set them with the MDM Set Match Groups tool.
4. Don't generate new match groups every time—use AO Constant key to incorporate them into the matching process.

#### **7.4.10.8 Real-time matching**

To do real-time matching:

1. For the incoming record, compute the segmentation keys.
2. Take the segmentation keys and use the MDM Dynamic Query tool to pull matching records.
3. Query match groups (find every record that's in the current record's match group). Now you have every record to throw into the match hopper.
4. Do matching.
5. Put the results into AO Constant Key—have any match IDs changed?
6. For any match IDs that have changed:
  - a. Go back to MDM and generate new match group IDs.
  - b. Apply these to records to set new match group IDs.

#### **7.4.10.9 Match review is conducted via workflow**

Use the following workflow to support automatic match review from RedPoint Data Management projects (this flow can be seen in the **mdm\_matchtest\_with\_workflow.dlp** sample project):

1. Set up two classes of matching, tight and loose. The tight matches are always acceptable (low false positive rate), but loose matches should also be considered.
2. Perform both classes of matching in the same project (parallel flows).

3. Set the tight match groups for the data using the MDM Set Match Groups tool. These will be committed and become The Truth if no further action is taken.
4. Use AO Match Group Differences to find where the tight and loose results disagree.
5. Process the differences with the MDM Start Match Review Workflow tool. This tool does several things:
  - For each match group to review, it opens a changeset.
  - In the newly-opened changeset, it changes the match group IDs to be the loose IDs.
  - It does NOT commit the changeset (it remains pending).
  - It starts an Automatic Match Review workflow that compares the accepted/committed match groups to the proposed/pending match groups.

In this way, a single project may create many match review workitems.

6. When the steward pulls an MDM Match Review workitem from the queue, she is presented with a UI allowing her to choose between the workitem's accepted and proposed matches.

7.5

## Views and subscriptions

A *view* is a directed graph of data transformations (aka *projections*) that supports complex queries.

A view can be as simple as displaying a table. You can define more complex views by chaining together three basic operations:

- **Table:** Read data from a table.
- **Join:** Join two data sources together.
- **Aggregate:** Summarize groups of records, calculating a set of output variables according to rules you define.

Combining these basic operations allows you to make useful calculations. For example, you could create views to track customer value metrics over time:

- *N*-day total purchases
- Number of web visits in last *N* days
- Number of email-sent events
- Number of email-responded events

You can also create a *subscription* to a view that notifies you when the data in a view changes. See [Subscribing to a view](#)<sup>118</sup> for more information.

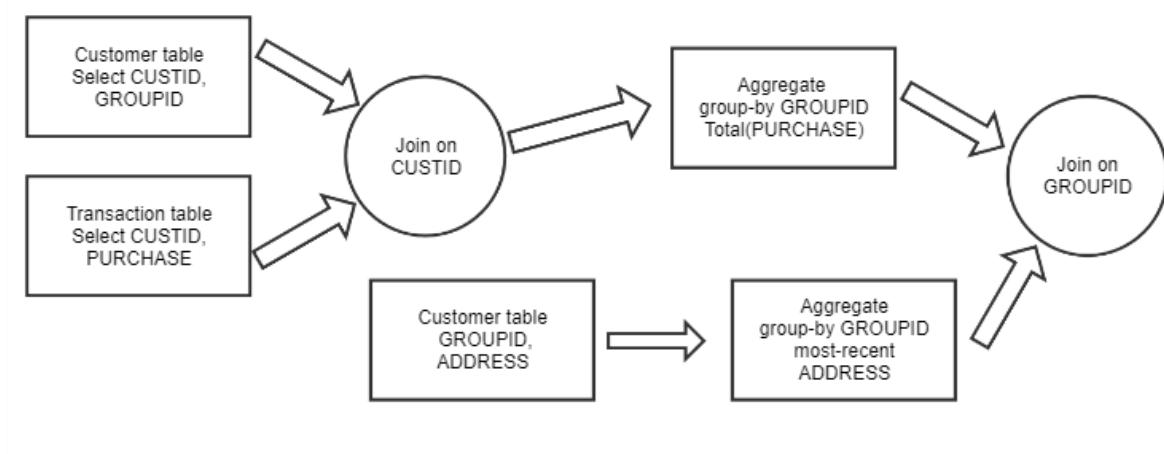
Views and subscriptions must be created and modified by a user assigned sufficient roles and permissions (for example, a user in the **dbas** group when using the demo database setup).

### 7.5.1 Defining a view

Under the covers, a view is a piece of code written in RedPoint Global Inc.'s proprietary, JSON-based Aggregation Expression Language (AGL) that is entered on the **Data Hub > Views** page. Currently, each view must be custom-written. See [Aggregation language](#)<sup>165</sup> for a detailed explanation of the language.

When you define a view, you're constructing a directed graph—an inverted tree. Inputs must be tables, which can be joined together and aggregated. Aggregations and tables may also be joined. For example, the view described in the graph below follows these steps:

1. Join customers and transactions by customer ID.
2. Aggregate the result on the customer group ID, and get the total purchase for the group of customers.
3. Join that to a different aggregation on customer group ID that gets the most recent address of the customer.



A view is described by a directed graph

Each step in the view graph is called a *projection*. Projections form the vertices of the graph.

#### 7.5.1.1 Types of projections

MDM supports the following projection types:

##### Aggregation

An aggregation projection has a single upstream vertex in the graph, and always starts with "group by" as in "group by customer ID". It consists of combining and chaining operations such as:

- [Select by date](#)<sup>165</sup>
- [Filter](#)<sup>165</sup>
- [Top](#)<sup>166</sup>
- [Sort](#)<sup>166</sup>

- [Unique](#)<sup>166</sup>
- [Count](#)<sup>161</sup>
- [Total](#)<sup>159</sup>, [Average](#)<sup>161</sup>, [Min and Max](#)<sup>159</sup>
- [Expression](#)<sup>157</sup>

These operations may be combined in any way that forms a legal [aggregation language](#)<sup>155</sup> expression, such as:

```
$Sum($AgeDays($timestamp, 0, 30).$Sort($PurchaseAmount, Descending).$Top(10),
$PurchaseAmount) / $Count($AgeDays($timestamp, 0, 60) + Math.PI)
```

## Join

A join projection has two or more upstream vertices in the graph. It is a simple join by join-key equality. There are no Cartesian joins (that is, many-to-many joins), but there are many-to-one joins. The join keys must be indexed in the upstream data.

## Table

A table view simply selects columns from a table and potentially maps them to new names. It passes primary key and indexed field metadata to the downstream projection.

### 7.5.1.2 View examples

The following examples assume that we have two tables:

The **Customers** table is versioned, with these fields:

- id
- text1
- match\_group\_id

The **Transactions** table is a time-series table, with these fields:

- timestamp
- cust\_id
- value
- department

#### 7.5.1.2.1 Simple table view

The following [aggregation language](#)<sup>155</sup> code creates a view that displays all fields of the **customer** table:

```
{
 "description": "Customer Table View. Subscribe to this to get customer changes",
 "projections": [
 {
 "projection": "customers",
 "type": "table",
 "table": {
 "table": "customers",
 "select_all": true
 }
 },
],
 "source": "customers"
}
```

The code is composed of the following parts:

| Part                      | Explanation                                                                                                                                                                          |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| "description"             | A description of what the view does.                                                                                                                                                 |
| "projections"             | The list of projections defines the data sources for data and for the chain of projections. Dependent projections are named first. This must be a tree with no dependency loops.     |
| "projection": "customers" | Name your projection something meaningful to you. Projection names must be unique inside this array of projections.                                                                  |
| "type": "table"           | Projections are tables, joins, or aggregations. In this case, this projection is simply the customer table.                                                                          |
| "table": "customers"      | Table projections simply load the data from the named database collection.                                                                                                           |
| "select_all": true        | Load all of the fields. If you want to limit which columns are loaded, set "select_all" to <code>false</code> and set the select field to an array of the column names to be loaded. |
| "source": "customers"     | Tell the view where in the projection graph to get its data. It must be the final projection in the graph.                                                                           |

Creating a subscription to this view will notify you of changes to the view (in this case, changes to the `customers` table). See [Subscribing to a view](#) for more information.

#### 7.5.1.2.2 Aggregating data from several tables to a view of transactions by customer

The following [aggregation language](#)<sup>155</sup> code creates a view of transactions by customer:

```
{
 "description": "customer_transactions",
 "projections": [
 {
 "projection": "trans_proj",
 "type": "table",
 "table": {
 "table": "transactions",
 "select_all": true
 }
 },
 {
 "projection": "cust_proj",
 "type": "table",
 "table": {
 "table": "customers",
 "select_all": true,
 "match_group_id": [
 "individual"
]
 }
 },
 {
 "projection": "join1",
 "type": "join",
 "join": {
 "inputs": [
 {
 "source": "trans_proj",
 "key": "cust_id",
 "map_all": true
 },
 {
 "source": "cust_proj",
 "key": "id",
 "map_all": false,
 "mapping": [
 {
 "in": "id",
 "out": "id"
 },
 {
 "in": "text1",
 "out": "text1"
 },
 {
 "in": "match_group_id",
 "out": "groupid"
 }
]
 }
]
 }
 },
 {
 "projection": "aggregation1",
 }
]
}
```

```

 "type": "aggregation",
 "aggregation": {
 "source": "join1",
 "group_field": "groupid",
 "sub_group_field": "department",
 "output_group_field": "group_n_dept",
 "output_fields": [
 {
 "field": "sumv1",
 "datatype": {
 "type": "float",
 "size": 8
 },
 "expression": "$sum($AgeDays($timestamp,5,10),$value1)"
 }
]
 },
 "source": "aggregation1"
}

```

The code is composed of the following parts:

| Part                   | Explanation                                                                                                         |
|------------------------|---------------------------------------------------------------------------------------------------------------------|
| "table":"transactions" | Get transactions from the time-series table named "transactions".                                                   |
| "table":"customers"    | Get customers from the versioned table named "customers".                                                           |
| "type":"join"          | Joining the customers to transactions makes a flattened record with customer information added on each transaction. |
| "source":"trans_proj"  | In this case, the transaction projection is one side of the join.                                                   |
| "key":"cust_id"        | The field used to match the other input.                                                                            |
| "map_all":true         | Use all fields and keep the same field names for the output.                                                        |
| "source":"cust_proj"   | The name of the second source for the join.                                                                         |
| "key":"id"             | Primary key field for this table. It should match the primary key field of the other table.                         |

| Part                                       | Explanation                                                |
|--------------------------------------------|------------------------------------------------------------|
| {"in": "match_group_id", "out": "groupid"} | Rename the individual match group ID field for the output. |

### 7.5.2 Subscribing to a view

You can create a subscription to a view that notifies you when the data in the view changes. For example, you could set up a subscription to a table that notifies you when any table records are added, modified, or deleted.

- Subscriptions provide real-time updates to view changes.
- Subscriptions support push operations to external databases.
- Each view can have multiple subscriptions

#### 7.5.2.1 Create a subscription

##### To create a subscription for a view

1. On the **Data Hub > Views** page, choose a view from the table and click **View**.

The screenshot shows a table with one row labeled '1: myview1'. To the right of the row, there are buttons for 'Created on 6/28/18', 'View' (which has a red box around it), 'Edit', and 'Delete'.

To create a subscription for a view, on the Data Hub > Views page choose a view from the table and click View

2. In the **Subscriptions** section, click the **Add** button

The screenshot shows a dialog box for 'myview1'. On the left, there's a JSON editor with some code. On the right, under 'Subscriptions', it says 'There are currently no subscriptions' and has an 'Add' button with a blue plus sign, which has a red arrow pointing to it.

The Subscriptions section Add button

3. On the popup dialog, set the subscription configuration options and click **Done**.

**Create New Subscription**

Enter description

ENABLED

DELIVERY TYPE

Web Service

WEB SERVICE URL

RETRY ON DELIVERY FAILURE START/MAX INTERVAL(SECONDS)

2      60

AGE OUT WINDOW START/END TIMES

0 : 00 : 00      23 : 59 : 59

MAX CHANGES PER MESSAGE

100

NOTIFICATION OPTIONS

RETURN FULL RECORD ON DELETE  
 RETURN OLD RECORD ON MODIFY  
 RETURN NEW RECORD ON MODIFY  
 RETURN DIFF ON MODIFY

CANCEL      DONE

**Subscription configuration options**

- a. Select to enable (start) the view.
- b. Description (name) of the view.
- c. **Web Service**—Notifications of data changes are sent via the internet (currently the only option).

- d. The URL that will receive the notifications.
- e. After a notification delivery failure, number of seconds ( $x$ ) to wait before trying to resend a notification.
- f. Number of seconds ( $y$ ) to keep trying to resend a notification. Taken together, this means after a notification failure, wait  $x$  seconds before trying to resend the notification, then keep trying to resend the notification for  $y$  seconds. If this time period expires, wait  $x$  seconds before trying again. Continue this cycle indefinitely until a notification is successfully sent.
- g. Some aggregation rules have a filter such as **\$AgeDays(datefield, N)** (which means "select everything up to  $N$  days old"). Of course, this selection changes daily and these record choices must be recalculated. The **Age-out** selector lets you specify a daily (24-hour) window as to when MDM is allowed to do this potentially computationally-expensive process. Values are 24-hour time in the form **HH:MM:SS**. If no values are entered, the form defaults to **start="00:00:00"** and **end="23:59:59"**. If **start > end** the calculation happens between PM and AM (for example, 23:00 - 01:00). If **start < end** the calculation happens during a contiguous chunk during the same day (for example, 00:00 - 04:00).
- h. Maximum number of record changes to include in a notification message. If the number of record changes at a given time exceeds this value, multiple messages will be sent. The max value is **1000**.
- i. By default, a notification simply tells you what records were deleted or modified. Select one or more of these options to include whole records (or the diff of records) in the notification message.

#### 7.5.3 Querying MDM view data with a DM connector

Inside a RedPoint Data Management project, you can use the MDM Query View tool to pull all of a view's output in batch. This tool allows you to:

- Export the view data to some other format/database.
- Validate the results by hand.
- Validate the view using processing steps in the RedPoint Data Management project.

#### 7.6

## Advanced workflow

A *workflow* is a process whereby a user of a specific group (or groups) performs a task on a specific database and table (such as "a business user adds a new customer to table x on database y").

Currently, workflows are custom-created for each customer. If you want to create a new workflow, contact RedPoint Global.

For some users (especially users with low-level access rights), a workflow is the only way to change a database. Higher-level users (such as those in the **dbas** and **system** groups) can [change metadata directly](#)<sup>[39]</sup>.

### 7.6.1 Standardized workflows

There is a small set of workflows that most organizations use:

#### Data steward review of business user changes

In these workflows, a business user can create new records or change existing ones. The new or modified record is forwarded to a data steward, who can approve, reject, or bounce the record back to the originating business user in order to make changes (the data steward can include comments for the business user, directing them as to what changes should be made). Once the data steward approves the record, the record's enclosing workitem is committed to the database.

See [Enter a new record](#)<sup>[43]</sup> and [Edit an existing record](#)<sup>[46]</sup> for detailed descriptions of these workflows.

#### Data steward review of marginal matches

In the automated match review workflow, If a RedPoint Global Data Management ETL process has automatically identified potential duplicate records, a data steward can review these match groups and edit, accept, or reject them. For example, you could review individual match level for new customers. See [Review automated matches](#)<sup>[49]</sup> for a detailed description of this workflow.

In the manual match review workflow, a data steward can manage duplicate records in a table by querying the table to identify likely duplicate records and manually grouping them into match groups. For example, you could review and change current customer matches at individual match level. See [Review manual matches](#)<sup>[52]</sup> for a detailed description of this workflow.

### 7.6.2 Workflow roadmap

All workflows follow a similar series of steps:

- **Validation rules:** After creation, MDM validates an initial workitem before it is placed in the workitem queue. For example, if a RedPoint Data Management ETL process has automatically identified potential duplicate records and forwarded the information to MDM, MDM will validate this RedPoint Data Management data before it places the resulting workitem in the MDM workitem queue.
- **Dashboard notification:** Once the initial workitem is created and added to the workitem queue, the MDM dashboard will indicate to users in the appropriate group or groups that a workitem is ready to be claimed. For example, when a business user adds a new customer, the workitem is added to the queue and MDM notifies users in the **data steward** group that a new workitem is available to be claimed by them.
- **Escalation:** After a workitem is claimed and processed by an appropriate user, the workitem is escalated to the next level. For example, if a data steward reviews a new customer record and rejects it, MDM kicks the record back to the originating business user so that user can make requested changes.
- **Quarantine:** Once a workitem has travelled through a workflow, it remains in the workitem queue until it is explicitly committed. Only after a workitem is committed are the changes made to the table. For example, once a new customer record has been reviewed by a data

steward and approved, MDM commits the changeset. (The changeset is committed by "system" because all data changesets are committed by MDM on behalf of the user).

7.7

## GDPR and privacy support

The increase of data breaches and data-related threats has led to the enactment of numerous statutes related to data privacy. The GDPR (General Data Protection Regulation) is the most significant of these regulations, and has served as a model for laws in other jurisdictions.

Implemented in May 2018, the GDPR significantly increased the data privacy rights of consumers and the requirements on companies that solicit and retain customer identities. It requires that any organization that manages or stores the personal data of EU citizens offer individuals the right to request and obtain any Personally Identifiable Information (PII), and in certain circumstances also have their data erased. It also levies hefty financial penalties for non-compliance.

7.7.1

### Data visibility

In MDM, data visibility is controlled via options set in the table schema:

- The **Display** option must be selected for a field to be visible in any view of the table. Fields that are not display-enabled cannot be queried via the web interface.
- The **PII** option can be selected and a masking hash applied to remove sensitive field data. In a masked record, all columns marked with PII are erased. When viewing masked records, PII fields are indicated by using asterisks (\*) as placeholders for the removed values. Those values have been removed from the database.

7.7.2

### Masking erases sensitive data fields

Data disclosure rules tend to be more relaxed for *pseudonymized* data (information that has been secured via data masking, encryption, and hashing). This masking process lets you retain data while “de-identifying” it. In MDM, you can select columns containing PII and apply a masking hash. If a request is made to mask an individual’s data, you can quickly locate the requested record and mask it. The record will remain in the system, but columns marked as PII cannot be queried, and their field values will be displayed as asterisks (\*).

7.7.3

### Erasing (purging) permanently deletes records

The **Erasure** operation permanently deletes entire records from a table, throughout the entire history of the table.

The **Verify Erasure** operation lets you confirm whether a record was masked or purged by querying the value of a hashed PII field.

7.7.4

### Masking and purging create hash entries

When you mask or purge a record, MDM creates a unique hash value for each mask hash defined for the table. When you choose fields for your mask hashes, ensure that those fields used to

generate the mask hash are, in combination, unique for the record or set of records you want masked or erased.

For example, in table **customers** you have defined three mask hashes:

1. **name** (hashes the **CONTACT** field)
2. **address** (hashes the **ADDRESS** field)
3. **name+address** (hashes both the **CONTACT** and **ADDRESS** fields)

When you mask or purge a record in table **customers**, MDM creates three hash values for the record (one for each of the defined mask hashes).

#### 7.7.5

#### Choose masking or purging based on customer needs

If a customer requests that their PII be secured, mask their record. If a customer requests that you delete all their information, purge their record.

#### 7.7.6

#### Masking and purging

| CONTACT | COMPANY         | ADDRESS        | CITY   | STATE | ZIP   | PHONE | ID  | emailaddresses | History                      |
|---------|-----------------|----------------|--------|-------|-------|-------|-----|----------------|------------------------------|
| *****   | KIRBY'S ELECTRI | 179 COUNTRY RD | AGAWAM | MA    | 01001 | ***** | 115 | 0 records      | <a href="#">View History</a> |

Data record with masked fields

In general, record masking is done if a customer asks for their private data to be masked, or if a company's policy requires it.

Record masking is usually available to users with low-level permissions (such as a business user).

Record purging is usually available only to users with high-level permissions (such as a data steward).

Some additional notes on masking and purging records:

- A masked record does not appear at all in Data Hub queries unless you define a query on a field that is *not* configured as PII and select the **Include masked records** box. If you do so, masked records are displayed in the data grid with asterisks (\*) instead of data in PII fields.
- You cannot query or view masked records from a workflow workitem.

- You can either mask or purge a record, but not both.

#### 7.7.6.1 Identifying PII in table schemas

The masking and purging functions are available only for tables that have been configured with one or more PII columns with associated masking hashes. Field hash values are needed so that MDM can verify whether a record has been masked or purged without having to retain the PII data in MDM.

PII is defined as any data that could potentially identify a specific individual. Any information that can be used to distinguish one person from another and can be used for de-anonymizing anonymous data is considered PII. Under the GDPR, PII may include information like transaction histories and IP addresses in addition to sensitive personal and financial data. You should consult with your organization's Privacy Officer to ensure that you identify PII in your data assets and comply with applicable laws, regulations, and policies.

You should know the following about your data:

- **Which fields contain PII?** Typical fields to mask are:
  - name
  - phone number
  - address
  - email address
  - account number
  - Social Security number
  - gender
  - birth date
- **Which PII fields do I want to use for querying masked or erased records?** Typical mask hash query fields are:
  - account number
  - the individual's full name
  - primary email address
  - phone number

If your organization looks people up using a special code, make a hash of that code.

#### 7.7.6.2 Hashing

Hashing supports future verification that a record has been masked or purged.

When a record is masked or purged, the record's query fields are hashed to create a verifiable record of the masking or purge, and the original PII fields are erased.

## Multi-field vs. single-field hashes

Multi-field hashes are not as common as single-field hashes, but may be necessary if you need more than one field to correctly identify a person (for example, full name is expressed as **First Name** and **Last Name** fields). A multi-field hash narrows the possible matches so you can make sure you've identified the right person. Do not use insufficiently unique fields (such as **First Name**) as a mask.

## Single vs. multiple hashes

This depends on your data and how your organization looks up people. Hashes on email, name, account number, and phone number fields are typical, though some organizations may need only account number or a name-and-address multi-field hash. Remember that each hash should correspond to one person/contact.

### 7.7.6.3 Record history for masking and purging

When a record is masked, PII is erased from all versions of the record (current and historical).

When a record is purged, all versions of the record (current and historical) are permanently deleted from the table. You cannot query purged records, except to verify that they have been erased.

### 7.7.6.4 Configure a table for masking and purging

#### To configure a table for masking and purging

1. Go to the **Data Hub > Tables** page.
2. Locate the desired table and click its **Schema** button.
3. On the **Edit Table** page, locate each **Field** you want to designate as **PII** and select its **PII** box.

| Main Table | emailaddresses | Masking Hashes                      |                                     |                          |                                     |        |         |  |         |
|------------|----------------|-------------------------------------|-------------------------------------|--------------------------|-------------------------------------|--------|---------|--|---------|
| Field      |                | Nullable                            | PII                                 | Index                    | Display                             | Type   | Details |  | Actions |
| CONTACT    |                | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | text ▾ | 30      |  |         |
| COMPANY    |                | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | text ▾ | 30      |  |         |
| ADDRESS    |                | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | text ▾ | 30      |  |         |
| CITY       |                | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | text ▾ | 40      |  |         |

To designate a record field as PII, select the field's PII box on the Edit Table page

4. Click the **Masking Hashes** tab.
5. Click **Create Mask Hash**.

Main Table emailaddresses Masking Hashes

Mask hashes are used to encrypt personal information in records for GDPR (General Data Protection Regulation) compliance.

+ Create Mask Hash

|       |        |         |
|-------|--------|---------|
| name  | fields | CONTACT |
| phone | fields | PHONE   |

Description Enter description Hash Fields Search... DONE CANCEL

Update Table Cancel

To create a mask hash for a table, on the table's Edit Table page, click the Masking Hashes tab and click Create Mask Hash

6. For each mask hash you want to create:
  - a. Enter a **Description** for this hash.
  - b. Click inside the **Hash Fields** box and select one or more PII-enabled fields.
  - c. Click **Done**.
7. Click **Update Table**.

#### 7.7.6.5 Mask or purge a record

To mask or purge a record

1. On the **Compliance > Erasure** page, select a **Target Table**.
2. Execute a query to locate the customer record.

Erasure

Target Table: customers

Record Query

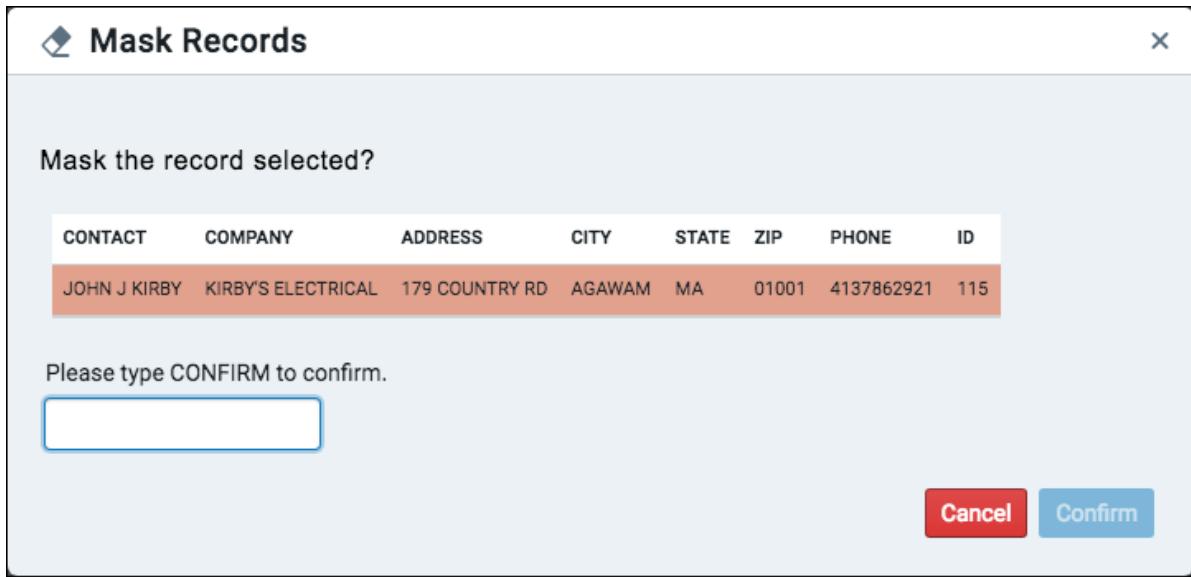
| Query field | Operation | Values       | Delete |
|-------------|-----------|--------------|--------|
| CONTACT     | =         | JOHN J KIRBY | x      |

|      | CONTACT      | COMPANY            | ADDRESS        | CITY   | STATE | ZIP   | PHONE      | ID  |
|------|--------------|--------------------|----------------|--------|-------|-------|------------|-----|
| Mask | JOHN J KIRBY | KIRBY'S ELECTRICAL | 179 COUNTRY RD | AGAWAM | MA    | 01001 | 4137862921 | 115 |

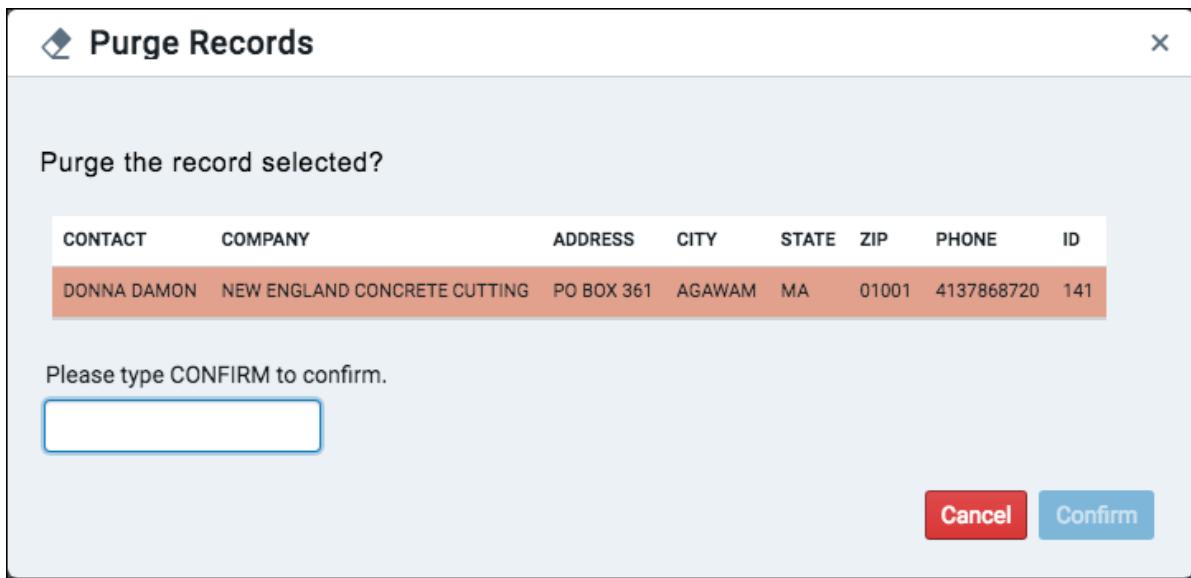
To mask or purge a record, first execute a query to locate the record

3. Find the record that you want to mask or purge and click its **Mask** or **Purge** link.

4. Examine the **Mask/Purge Records** popup to confirm that the selected record is correct. Enter "confirm" and click **Confirm**. **Warning!**: When a record is purged, all versions of the record (current and historical) are permanently deleted from the table.



Mask Records confirmation dialog



Purge Records confirmation dialog

#### 7.7.6.6 Verify a record was masked or purged

The **Verify Erasure** page lets you query tables with a generated mask hash for one or more PII fields and confirm whether a record has been masked or purged. The query contains one or more values that are checked against a table containing hashes for records that have been masked or purged. MDM disassociates the masked record from the hash code, so the query returns not the masked record, but the hash information showing what action was taken (masking or erasure) and when the action occurred.

The verify masked/purged record operation is usually available to users with low-level permissions (such as a business user).

The verify masked/purged record operation only confirms if a particular record was masked or purged. The record itself cannot be located this way.

### To verify that a record was masked or purged

1. On the **Compliance > Verify Erasure** page, select a **Target Table**.

**Verify Erasure**

Select a hash mask and fill in PII fields in order to see if the record was masked or purged.

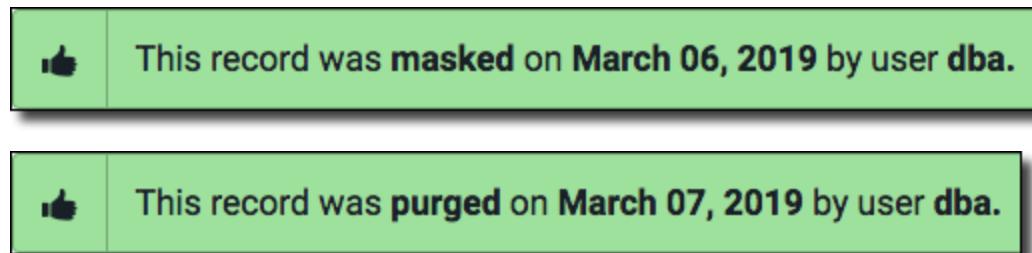
|                  |                |
|------------------|----------------|
| Target Table     | customers      |
| Target Mask Hash | name + address |
| Field            | Value          |
| CONTACT          | JOHN J KIRBY   |
| ADDRESS          | 179 COUNTRY RD |

**Verify**

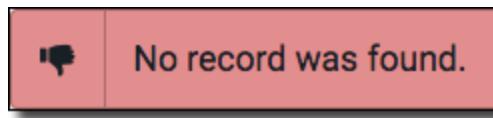
Compliance > Verify Erasure page

2. Select a **Target Mask Hash**. (You must have already defined at least one mask hash.)
3. For the selected mask hash, enter the field value(s) of the target record.
4. Click **Verify**.

If a matching hash value is found, a success message is displayed:



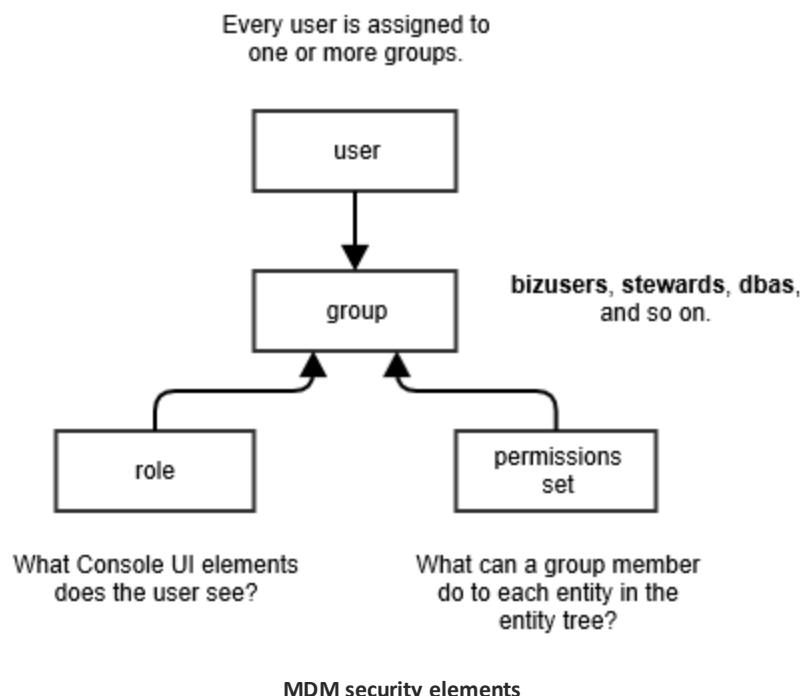
If no matching hash value is found, a failure message is displayed:



7.8

## Security

MDM security is constructed using the following elements:



7.8.1

### Users and groups

Each person who logs on to MDM is an individual *user*. Each user is assigned to one or more groups. The system administrator defines the users and groups that are appropriate for the site.

For example, this user belongs to the **everyone**, **bizusers**, and **stewards** groups:

The screenshot shows the 'Add User' dialog box. It includes fields for User Name (jsaxon), Email (jsaxon@example.com), Full Name (Jane Saxon), User Description (A data steward), and User Groups (everyone, bizusers, stewards). There is also a 'Set Password' button and 'Save' and 'Cancel' buttons at the bottom.

Each user is assigned to one or more groups. This user belongs to the **everyone**, **bizusers**, and **stewards** groups.

The following groups are the default configuration when MDM is initialized with the demo database:

- **bizusers**: Low-privilege group designed for call centers and field reps.
- **stewards**: Medium-privilege users with authority to make data decisions.
- **dbas**: High-privilege users that can make data model changes and bulk data changes.
- **officers**: Not assigned to any role in particular; reserved for future high-privilege users.

For example, one could have a business user "srogers" who belongs to both the **everyone** and **bizusers** groups, and a data steward "nromanoff" who belongs to the **everyone** and **stewards** groups.

MDM security features can be modified by users in the **dbas** and **system** groups.

#### 7.8.1.1 Special users and groups

The following users and groups are built into MDM and cannot be modified or deleted:

## Everyone group

Users are automatically assigned to the **everyone** group. A user cannot be removed from the **everyone** group, and the **everyone** group cannot be deleted. The **everyone** group is typically granted read-only data privileges.

## System group

A user assigned to the **system** group has permission to do anything in MDM. This group cannot be deleted.

## System user

The **system** user always belongs to the **system** group and cannot be deleted or removed from the **system** group.

### 7.8.1.2 User operations

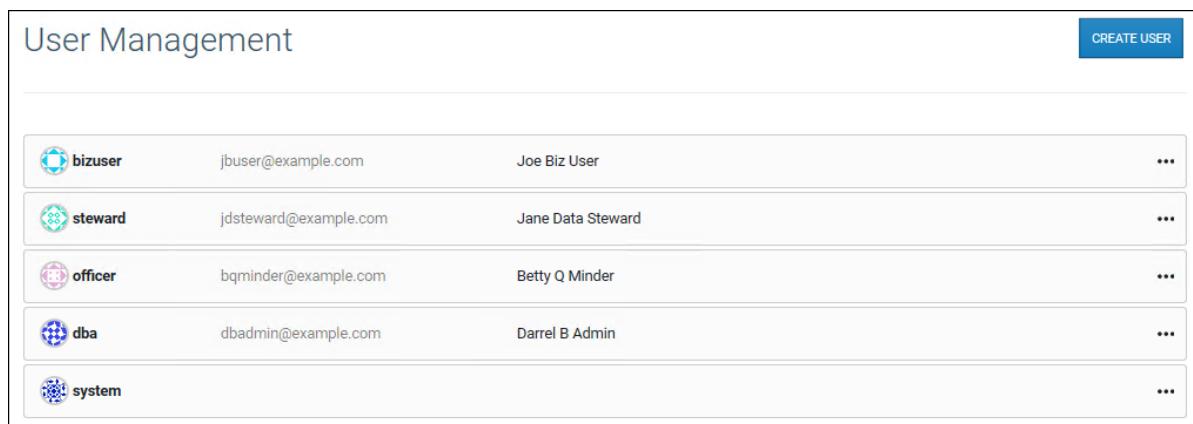
The following operations can be performed on users:

- [Browse users](#)<sup>131</sup>
- [Create a user](#)<sup>132</sup>
- [Edit a user](#)<sup>133</sup>
- [Delete a user](#)<sup>135</sup>

#### 7.8.1.2.1 Browse users

##### To browse users

1. In the **Administration** section, click **Users**.
2. The **User Management** page displays a list of system users.



The screenshot shows the 'User Management' page with a header 'User Management' and a 'CREATE USER' button. Below the header is a table listing five users:

| Icon    | Username              | Email             | Name | Actions |
|---------|-----------------------|-------------------|------|---------|
| bizuser | jbuser@example.com    | Joe Biz User      | ...  |         |
| steward | jdsteward@example.com | Jane Data Steward | ...  |         |
| officer | bqminder@example.com  | Betty Q Minder    | ...  |         |
| dba     | dbadmin@example.com   | Darrel B Admin    | ...  |         |
| system  |                       |                   | ...  |         |

The User Management page displays a list of system users

#### 7.8.1.2.2 Create a user

##### To create a user

1. In the **Administration** section, click **Users**.
2. On the **User Management** page, click **Create User**.

The screenshot shows the 'User Management' page. At the top right is a blue 'CREATE USER' button. Below it is a table listing five users:

| Icon    | Username              | Email             | Name | Actions |
|---------|-----------------------|-------------------|------|---------|
| bizuser | jbuser@example.com    | Joe Biz User      | ...  |         |
| steward | jdsteward@example.com | Jane Data Steward | ...  |         |
| officer | bqminder@example.com  | Betty Q Minder    | ...  |         |
| dba     | dbadmin@example.com   | Darrel B Admin    | ...  |         |
| system  |                       |                   | ...  |         |

**The User Management page allows you to create a new user**

3. Enter the user information, deleting or adding **User Groups** as necessary. Click **Search** for a list of user groups.

**Add User**

|                           |                                              |
|---------------------------|----------------------------------------------|
| User Name (required)      | jqpublic                                     |
| Email (required)          | jqpublic@gmail.com                           |
| Full Name                 | Jane Q Public                                |
| User Description          | An everyday user                             |
| User Groups               | <b>everyone</b> <b>bizusers</b><br>Search... |
| <b>Set Password</b>       |                                              |
| <b>Save</b> <b>Cancel</b> |                                              |

The Add User dialog allows you to enter new user information, including deleting or adding associated user groups

4. Click **Set Password** to define the user password. *You cannot save the user unless you set a password.*
5. Click **Save** to finish.

#### 7.8.1.2.3 Edit a user

##### To edit a user

1. In the **Administration** section, click **Users**.
2. On the **User Management** page, click the menu icon **•••** next to the desired user and select **Edit**.

The screenshot shows a "User Management" page with a table of users. The columns are "User Type", "Email", and "Name". The rows are:

- bizuser jbuser@example.com Joe Biz User
- steward jdsteward@example.com Jane Data Steward
- officer bqmindr@example.com Betty Q Minder
- dba dbadmin@example.com Darrel B Admin
- system system@example.com

A context menu is open over the second row (steward), showing "Edit" and "Delete" options.

The User Management page allows you to edit the information for an existing user

3. Edit the user information, deleting or adding **User Groups** as necessary. Click **Search** for a list of user groups.

The screenshot shows the "Edit User" dialog with the following fields:

- User Name (required): bizuser
- Email (required): jbuser@example.com
- Full Name: Joe Biz User
- User Description: A business user
- User Groups: everyone, bizusers (with a search bar below)

Buttons at the bottom include "Change Password" (blue), "Save" (blue), and "Cancel" (red).

The Edit User dialog allows you to edit an existing user's information, including deleting or adding associated user groups

4. If necessary, click **Change Password**.
5. Click **Save** to finish.

#### 7.8.1.2.4 Delete a user

##### To delete a user

1. In the **Administration** section, click **Users**.
2. On the **User Management** page, click the menu icon **•••** next to the desired user and select **Delete**.

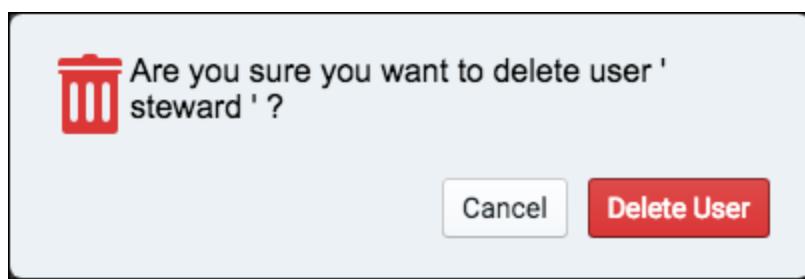
The screenshot shows the 'User Management' page. At the top right is a blue 'CREATE USER' button. Below it is a table listing five users:

|  | bizuser | jbuser@example.com    | Joe Biz User      |
|--|---------|-----------------------|-------------------|
|  | steward | jdsteward@example.com | Jane Data Steward |
|  | officer | bqminder@example.com  | Betty Q Minder    |
|  | dba     | dbadmin@example.com   | Darrel B Admin    |
|  | system  |                       |                   |

A context menu is open over the 'steward' row, showing options: 'Edit' (with a pencil icon) and 'Delete' (with a trash can icon). A hand cursor is hovering over the 'Delete' option.

The User Management page allows you to delete an existing user

3. Click **Delete User**.



MDM will ask you for confirmation before deleting an existing user

#### 7.8.1.3 Group operations

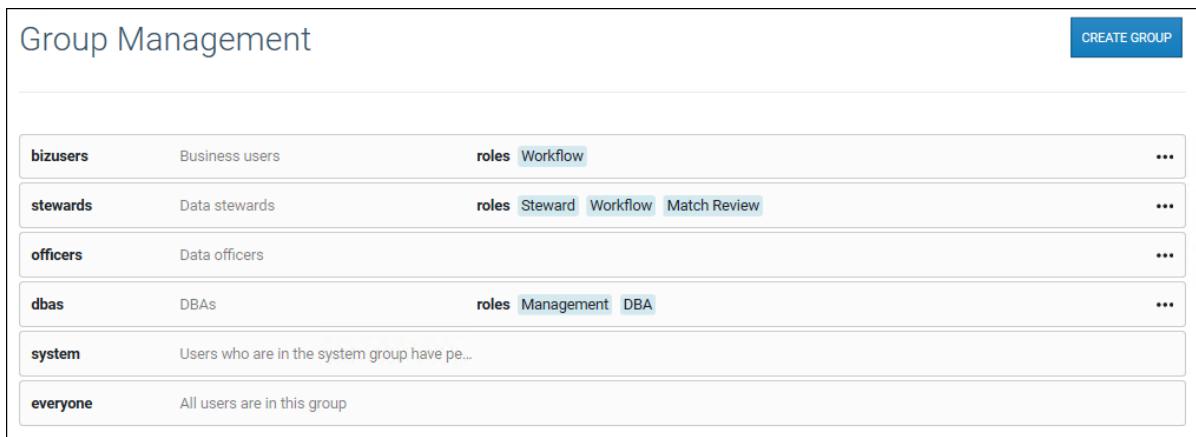
The following operations can be performed on groups:

- [Browse groups](#) [136]
- [Create a Group](#) [136]
- [Edit a group](#) [137]
- [Delete a group](#) [139]

#### 7.8.1.3.1 Browse groups

##### To browse groups

1. In the **Administration** section, click **Groups**.
2. The **Group Management** page displays a list of system groups.



The screenshot shows the 'Group Management' page with a header 'CREATE GROUP'. Below the header is a table listing system groups:

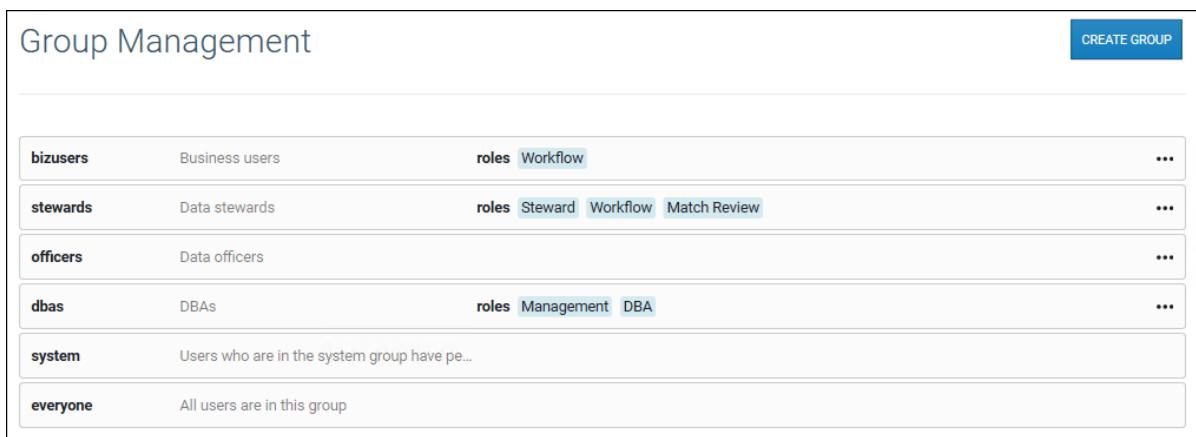
| Group Name | Description                                  | Roles                               | Actions |
|------------|----------------------------------------------|-------------------------------------|---------|
| bizusers   | Business users                               | roles Workflow                      | ...     |
| stewards   | Data stewards                                | roles Steward Workflow Match Review | ...     |
| officers   | Data officers                                |                                     | ...     |
| dbas       | DBAs                                         | roles Management DBA                | ...     |
| system     | Users who are in the system group have pe... |                                     |         |
| everyone   | All users are in this group                  |                                     |         |

The Group Management page displays a list of all system groups

#### 7.8.1.3.2 Create a group

##### To create a group

1. In the **Administration** section, click **Groups**.
2. On the **Group Management** page, click **Create Group**.

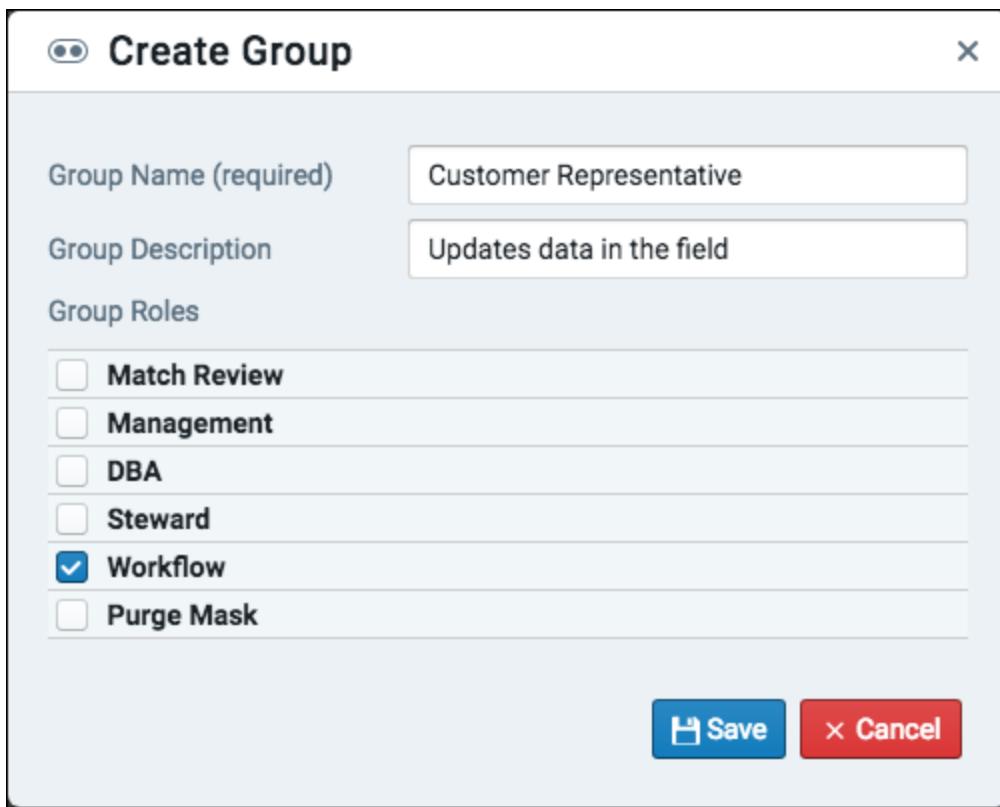


The screenshot shows the 'Group Management' page with a header 'CREATE GROUP'. Below the header is a table listing system groups, identical to the one in the previous screenshot:

| Group Name | Description                                  | Roles                               | Actions |
|------------|----------------------------------------------|-------------------------------------|---------|
| bizusers   | Business users                               | roles Workflow                      | ...     |
| stewards   | Data stewards                                | roles Steward Workflow Match Review | ...     |
| officers   | Data officers                                |                                     | ...     |
| dbas       | DBAs                                         | roles Management DBA                | ...     |
| system     | Users who are in the system group have pe... |                                     |         |
| everyone   | All users are in this group                  |                                     |         |

The Group Management page allows you to create a new group

3. Enter the **Group Name** and (optionally) a **Group Description**, and then select the **Group Roles**.



The Create Group dialog allows you to define a new group's information and choose the associated group roles

4. Click **Save** to finish.

#### 7.8.1.3.3 Edit a group

##### To edit a group

1. In the **Administration** section, click **Groups**.
2. On the **Group Management** page, click the menu icon **•••** next to the desired group and select **Edit**.

The screenshot shows a list of groups in the RedPoint Master Data Management system:

| Group Name | Description                                  | Roles                           |
|------------|----------------------------------------------|---------------------------------|
| bizusers   | Business users                               | Workflow                        |
| stewards   | Data stewards                                | Steward, Workflow, Match Review |
| officers   | Data officers                                |                                 |
| dbas       | DBAs                                         | Management, DBA                 |
| system     | Users who are in the system group have pe... |                                 |
| everyone   | All users are in this group                  |                                 |

A context menu is open over the "stewards" group, showing options: Edit (with a pencil icon) and Delete (with a trash bin icon). A cursor arrow points to the "Edit" option.

The Group Management page allows you to edit the information for an existing group

3. Edit the **Group Description** and **Group Roles** as necessary.

The Edit Group dialog is displayed, showing the following fields:

- Group Name (required):** customer representative
- Group Description:** Updates data in the field
- Group Roles:** A list of checkboxes:
  - Match Review
  - Management
  - DBA
  - Steward
  - Workflow
  - Purge Mask
- Buttons:** Save (blue button) and Cancel (red button)

The Edit Group dialog allows you to edit an existing group's description and associated group roles

4. Click **Save** to finish.

#### 7.8.1.3.4 Delete a group

##### To delete a group

1. In the **Administration** section, click **Groups**.
2. On the **Group Management** page, click the menu icon next to the desired user and select **Delete**.

The screenshot shows the 'Group Management' page with a list of groups:

| Group    | Description                                  | Roles                           | Action         |
|----------|----------------------------------------------|---------------------------------|----------------|
| bizusers | Business users                               | Workflow                        |                |
| stewards | Data stewards                                | Steward, Workflow, Match Review | Edit<br>Delete |
| officers | Data officers                                |                                 |                |
| dbas     | DBAs                                         | Management, DBA                 | ...            |
| system   | Users who are in the system group have pe... |                                 |                |
| everyone | All users are in this group                  |                                 |                |

The Group Management page allows you to delete an existing group

#### 7.8.2 Group roles

Group roles control what each user can access in the MDM Web App. Roles are attached to groups, and each role defines a set of user interface elements that are displayed for members of a given group. Groups can have multiple roles. The **stewards** group shown below is assigned the **Steward**, **Workflow**, and **Match Review** roles.

The screenshot shows the 'Group Management' page with the 'stewards' group selected. The roles assigned to this group are:

| Group    | Description    | Roles                           |
|----------|----------------|---------------------------------|
| bizusers | Business users | Workflow                        |
| stewards | Data stewards  | Steward, Workflow, Match Review |

Group roles control what each user can access in the MDM Web Console. Roles are attached to groups, and each role defines a set of user interface elements that are displayed for members of a given group.

As an example, user "tstark" is a member of the **dbas** group (which is assigned the **DBA** role). User "bbanner" is a member of the **bizusers** group (which is assigned the **workflow** role). For "tstark", the Web App displays both the **Create Database** and **Refresh** buttons, but "bbanner" only sees a database **Refresh** button.

The following table lists the user interface elements available to each role:

| Role              | Permissions                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Management</b> | Manage (add/remove/update) these entities: <ul style="list-style-type: none"> <li>• users</li> <li>• groups</li> <li>• processes</li> <li>• workflows</li> <li>• permissions</li> <li>• id generators</li> <li>• exports</li> </ul>                                                                                                                                          |
| <b>DBA</b>        | Manage metadata (data model) editing: <ul style="list-style-type: none"> <li>• create database</li> <li>• delete database</li> <li>• create table</li> <li>• modify table</li> <li>• delete table</li> <li>• specify associations and linkages</li> </ul>                                                                                                                    |
| <b>Steward</b>    | Ad-hoc data Create, Read, Update, and Delete (CRUD) and changeset management: <ul style="list-style-type: none"> <li>• create record</li> <li>• delete record</li> <li>• modify record</li> <li>• create/select pending changeset</li> <li>• commit pending changeset</li> <li>• review and manage/undo pending operation</li> <li>• conflict view and resolution</li> </ul> |
| <b>Workflow</b>   | Workflow menu is available to: <ul style="list-style-type: none"> <li>• Start a new workflow</li> </ul>                                                                                                                                                                                                                                                                      |

| Role                | Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <ul style="list-style-type: none"> <li>• View and claim available workflows</li> <li>• Continue work on assigned workflow</li> <li>• View workflows user has started</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Match Review</b> | <p>Enable ad-hoc (no workflow) interfaces related to record matching:</p> <ul style="list-style-type: none"> <li>• suspect review</li> <li>• force match</li> <li>• break apart match</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Purge Mask</b>   | <p>Allow PII (Personally Identifiable Information) actions. These actions address requirements of the <a href="#">GDPR (General Data Protection Regulation)</a> <small>[122]</small>.</p> <p>These actions include:</p> <ul style="list-style-type: none"> <li>• Masking PII fields in records.</li> <li>• Purging (removing) PII records.</li> <li>• Determining if a record with PII data has been masked or purged, and if so, by whom and when.</li> </ul> <p>For more information on GDPR, refer to <i>Right of access by the data subject</i> and <i>Right to erasure</i> (articles 15 and 17 of the GDPR code). <a href="#">This source</a> provides a concise reference, and the complete regulation can be found <a href="#">here</a>.</p> |

A user implicitly has the roles of all groups to which the user belongs. A user whose groups are not assigned any roles can see only the common Web App elements available to everyone.

### 7.8.3 Permissions and entities

Permissions define what operations a member of a group can do to each entity in the system.

The MDM entities are:

- System
- Database
- Table

- View
- Process
- Workflow

In the MDM permissions model:

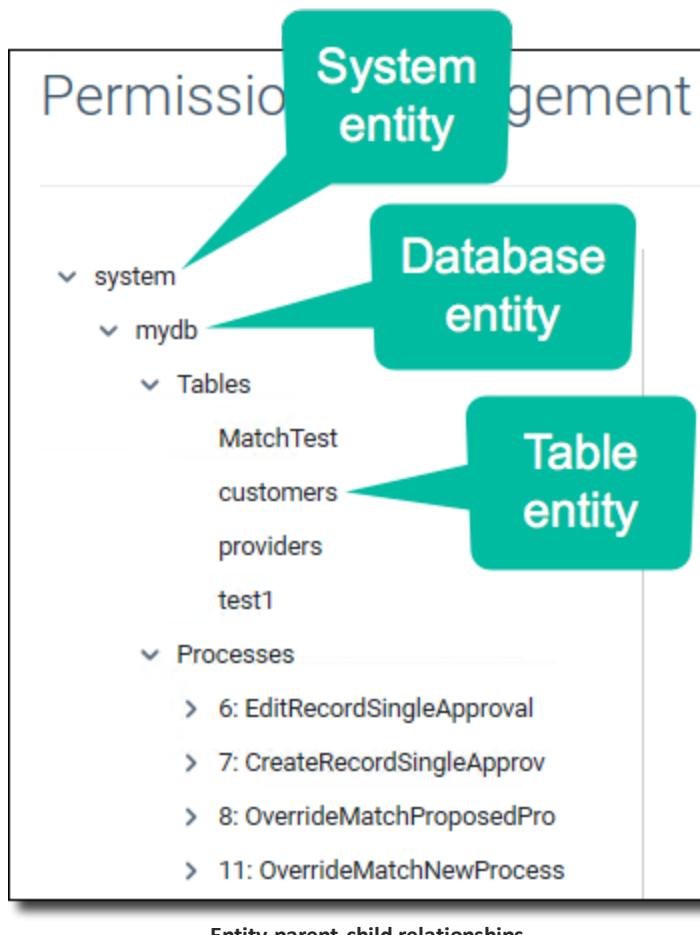
- Permissions (to operate on entities) are assigned to groups.
- A user is allowed to do anything that any of her groups is allowed to do.
- For a given group, entity permissions are inherited by the entity's children, but may be explicitly overridden.

Note that **permissions are not roles!** Group roles control which user interface a user sees in the Web App, while permissions control operations on entities. Permissions and roles should align to avoid confusion.

The **system** group is assigned all permissions. A user in the **system** group can do anything.

#### 7.8.3.1 Permissions inheritance

The root entity of the permissions hierarchy is **system**. All other entities are children of **system**. In the example below, **mydb** is a child entity of **system**, and **customers** is a child of both **system** and **mydb**.



Entity parent-child relationships

Permissions assigned to a group for an entity are inherited by children of the entity, but these inherited permissions may be overridden.

Unless overridden, group permissions assigned for an entity are inherited for the entity's children according to this hierarchy:

- system
  - database
    - table
      - column (*future enhancement*)
    - process
      - workflow
    - view
      - subscription

In the example below, the permissions assigned to the **bizusers** group for the **Database** entity are displayed, as well as the permissions inherited for the child entities.

## Permissions Management

The screenshot shows the RedPoint Permissions Management interface. On the left, there is a tree view of entities under the 'system' category. The 'HealthCare' entity is expanded, showing its children: 'Tables', 'Processes', and 'Views'. A red arrow points from the text 'In the example below...' to the 'Tables' node under 'HealthCare'. To the right of the tree view, there is a detailed view for the 'Database: HealthCare' entity. At the top, there are two tabs: 'bizusers' (which is selected) and 'stewards'. Below the tabs is a checkbox labeled 'Override Inherited Permissions'. The main area is divided into two sections: 'Permissions for this Entity' and 'Permissions for Children of this Entity'. Under 'Permissions for this Entity', there is a section for 'Database' with a list of permissions: delete\_database, create\_table, query\_table, create\_changeset, query\_changeset, create\_id\_generator, create\_process, query\_process, create\_view, and query\_view. Most of these permissions have a checked checkbox. Under 'Permissions for Children of this Entity', there is a section for 'Table' with a list of permissions: read\_table, modify\_table, delete\_table, and query\_record. All of these permissions have a checked checkbox. At the bottom of the detailed view, it says 'Bizusers permissions for the HealthCare database'.

system

- > mydb
- < b **HealthCare** ↗
- > Tables
- > Processes
- > Views

**Database: HealthCare**

bizusers   stewards  

Override Inherited Permissions

**Permissions for this Entity**

**Database**

- delete\_database
- create\_table
- query\_table
- create\_changeset
- query\_changeset
- create\_id\_generator
- create\_process
- query\_process
- create\_view
- query\_view

**Permissions for Children of this Entity**

**Table**

- read\_table
- modify\_table
- delete\_table
- query\_record

Bizusers permissions for the HealthCare database

Note that you can add and remove permissions for groups at any level of a parent-child entity tree.

Remember that permissions are assigned on a group basis. For example, let's say there are two groups assigned permissions to work with tables: **bizusers** and **stewards**. We see that the **bizusers** group has the following table permissions:

**Table: matchtest**

bizusers   stewards  

---

**Override Inherited Permissions**

**Permissions for this Entity**

**Table**

- `read_table`
- `modify_table`
- `delete_table`
- `query_record`
- `create_record`
- `modify_record`
- `delete_record`
- `mask_record`
- `purge_record`

Table permissions for the bizusers group

The **stewards** group has a slightly different permissions list:

**Table: matchtest**

bizusers **stewards** 

**Override Inherited Permissions**

**Permissions for this Entity**

**Table**

- `read_table`
- `modify_table`
- `delete_table`
- `query_record`
- `create_record`
- `modify_record`
- `delete_record`
- `mask_record`
- `purge_record`

Table permissions for the stewards group

#### 7.8.3.2 How permissions are interpreted for each entity

Permissions are interpreted for each entity as follows:

| Entity Type | Permission                    | Action allowed                            |
|-------------|-------------------------------|-------------------------------------------|
| system      | <code>create_database</code>  | Create a new database.                    |
| database    | <code>delete_database</code>  | Delete a database.                        |
| database    | <code>create_table</code>     | Create a new table.                       |
| database    | <code>query_table</code>      | Query the list of tables in the database. |
| database    | <code>create_changeset</code> | Create a changeset on this database.      |

| Entity Type | Permission          | Action allowed                                                                                                     |
|-------------|---------------------|--------------------------------------------------------------------------------------------------------------------|
| database    | query_changeset     | Query changesets on this database.                                                                                 |
| database    | create_export       | Create an export to external RDBMS.                                                                                |
| database    | create_id_generator | Create a new ID generator.                                                                                         |
| database    | create_view         | Create a new computed view.                                                                                        |
| database    | query_export        | Query list of exports.                                                                                             |
| database    | create_process      | Create a new process in the database.                                                                              |
| database    | query_process       | Query the list of defined processes.                                                                               |
| database    | query_view          | Query the list of defined computed views.                                                                          |
| table       | read_table          | Read table schema and other details.                                                                               |
| table       | modify_table        | Change the table's schema.                                                                                         |
| table       | delete_table        | Delete this table.                                                                                                 |
| table       | query_record        | Query records in this table. Also grants the right to determine if a record has had its PII data masked or purged. |
| table       | create_record       | Create new record in this table.                                                                                   |
| table       | modify_record       | Modify records in this table. This combined with create_records allows upsert.                                     |
| table       | delete_record       | Delete records from this table.                                                                                    |
| table       | purge_record        | Permanantly expunge records or record versions from the database.                                                  |

| Entity Type | Permission          | Action allowed                                                                         |
|-------------|---------------------|----------------------------------------------------------------------------------------|
| column      | query_column        | Include this column in query output (future enhancement).                              |
| column      | modify_column       | Change the value of this column (future enhancement).                                  |
| database    | query_operation     | Query the operations performed in a changeset.                                         |
| database    | undo_operation      | Undo operations of a changeset.                                                        |
| database    | commit_changeset    | Commit a changeset.                                                                    |
| database    | discard_changeset   | Discard a changeset.                                                                   |
| database    | manage_conflict     | Query and resolve conflicts.                                                           |
| process     | read_process        | Read a process definition.                                                             |
| process     | modify_process      | Replace an existing process.                                                           |
| process     | delete_process      | Delete a process.                                                                      |
| process     | create_workflow     | Create a workflow based on this process.                                               |
| process     | query_workflow      | Query which workflows are available to start, which workitems are assigned, and so on. |
| workflow    | read_workflow       | Read a workflow definition.                                                            |
| workflow    | modify_workflow     | Modify a workflow definition.                                                          |
| workflow    | delete_workflow     | Delete the workflow.                                                                   |
| view        | read_view           | Read details of a view.                                                                |
| view        | modify_view         | Modify a view.                                                                         |
| view        | delete_view         | Delete a view.                                                                         |
| view        | create_subscription | Create a new subscription to a computed view.                                          |

| Entity Type  | Permission          | Action allowed                                                         |
|--------------|---------------------|------------------------------------------------------------------------|
| view         | query_subscription  | Query the subscriptions for a computed view.                           |
| subscription | read_subscription   | Read details of a subscription.                                        |
| subscription | modify_subscription | Modify a subscription.                                                 |
| subscription | delete_subscription | Delete a subscription.                                                 |
| database     | mask_record         | Clear PII fields in a record and note who masks the record and when.   |
| database     | purge_record        | Remove a record with PII data and note who purged the record and when. |

#### 7.8.3.3 System-only permissions

There are some things that can only be done by users in the **system** group, so there is no explicit permission type for them:

- Manage users
- Change permissions of the **system** object

#### 7.8.3.4 Synchronization and versions

There is no versioning of permissions—they apply to all versions of data and metadata simultaneously. When a database is deleted, everything is deleted (including all history) and permissions for the database (and all child entities of the database) are removed with it.

#### 7.8.3.5 Permissions and groups

The following topics relate to permissions and groups:

##### 7.8.3.5.1 Permission entries

A permission entry is a set of permissions for an entity and all inheriting descendants. Because of this, it is useful to think of setting permissions on an entity as actually setting the default permissions for an entire *entity tree*. An entity tree consists of the entity and all of its current and future descendants. The root of every entity tree is the **system** entity. Some things to note about permission entries:

- Each permission entry is for a single group.
- Each permission entry applies the given permissions to that entity and all descendants.

- Each permission entry must specify all desired permissions for the given entity and its child entities. Because of this requirement, permission overrides are an all-or-nothing proposition: if any permission is changed from its inherited default, all permissions must be specified.
- Permissions that aren't applicable to the given entity type are ignored (but may be passed on to child entities). For example, when setting permissions on a database, you will also want to specify the default permissions for every table, process, workflow, view, and subscription.
- Inherited permissions are taken from the closest ancestor in the permissions tree.

Typically, the **system** entity has a set of permission entries (one for each group) to establish the default permissions for each group. You can gain finer-grained control by overriding permissions for a particular group and entity combination. Note that deleting a permission entry does *not* remove any descendant permission entries.

In another example, suppose that database **hospitalRecords** has a permission entry containing **modify\_record** for group **stewards**. Given that setting, all table records in that database can be modified by users in the **stewards** group.

However, if the administrator wants to limit access to certain tables (for example, **hospitalRecords.no\_touch**), she may override permissions for the **no\_touch** table, and remove the **modify\_record** permission, thus preventing stewards from being able to modify that specific table.

#### 7.8.3.5.2 Most-permissive logic

There may be multiple permission entries for an entity that apply to the user attempting to perform an operation because a user may belong to multiple groups. If this is the case, the *effective* set of permissions for a user on an entity consists of the union of the permissions on that entity for all of the user's groups.

#### 7.8.3.6 Change entity permissions

By default, child entities inherit the permissions of their parent entity. For example, **system** entity permissions are inherited by all database entities. Database permissions are inherited by their table entities, and so on down the hierarchy.

If you don't want to use inherited permissions for a portion of the permissions tree, you can override an entity's permissions.

- Children of the overridden entity will inherit the new permissions unless they also override permissions.
- Overrides are done per-group. Therefore, some groups may see overridden permissions and others may not.
- A user that does not belong to a group with sufficient privileges to modify permissions will still be able to view permissions but not change or override them.

## To change permissions on an entity

The screenshot shows the 'Permissions' section of the administration interface. On the left, a tree view lists entities: 'system' (expanded), 'mydb' (expanded), 'Tables' (expanded), 'customers' (selected), 'matchtest' (highlighted in grey), 'Processes' (expanded), 'Views' (expanded), and '2: 120 Loyalty Point Sum' (selected). A red circle with the number '2' is placed over '2: 120 Loyalty Point Sum'. To the right, under 'View: 120 Loyalty Point Sum', there is a group named 'bizusers' with a trash icon, and a 'stewards' group with a gear icon. A red circle with the number '3' is placed over the 'Override Inherited Permissions' checkbox. Below this, the 'Permissions for this Entity' section shows checkboxes for 'read\_view', 'modify\_view', 'delete\_view' (selected), 'create\_subscription' (selected), and 'query\_subscription'. A red circle with the number '4' is placed over the 'delete\_view' checkbox. The 'Permissions for Children of this Entity' section shows checkboxes for 'read\_subscription', 'modify\_subscription' (selected), and 'delete\_subscription'. A red circle with the number '5' is placed over the 'modify\_subscription' checkbox.

The process of changing permissions for an entity (steps 2 through 5 below)

1. In the **Administration** section, click **Permissions**.
2. Select the entity on the tree for which you want to change inherited permissions.
3. Select a group.
4. Select or clear entity permissions. (If the permissions are not selectable, select **Override Inherited Permissions**.)  
**WARNING!** Once you override inherited permissions on an entity, you cannot switch back to inherited permissions. Your only option at this point is to destroy the entire group for that entity.
5. Select or clear permissions on the entity's children.

### 7.8.3.7 Adding and deleting permission entry groups

Adding or deleting a permission entry group for an entity means that you are granting or revoking a group's permissions on an entity (and possibly its children).

For example, if you grant permissions for the **officers** group to a database entity, you are explicitly granting any user who is a member of the **officers** group officer-level permissions for the database entity and its children (because of [inheritance](#)<sup>[142]</sup>). If you decide that **officers** should be granted different permissions for certain tables of that database, you could go to a child table of that database entity and delete the **officers** group permissions entry. This would revoke officer-specific permissions from the table.

### To add a permission entry for a group

Note that you can only add groups that have been previously defined for your MDM site workflows.

1. In the **Administration** section, click **Permissions**.
2. Select an entity in the tree and click the group add button .
3. On the **Create Permission Group** popup, select a group name and click **Create**.

### To delete a permission entry for a group

1. In the **Administration** section, click **Permissions**.
2. Select an entity in the tree:

**Permissions**

- system
  - mydb
    - Tables
      - MatchTest
      - customers
      - test1
    - Processes
      - 6: EditRecordSingleApproval
      - 7: CreateRecordSingleApproval
      - 8: OverrideMatchProposedProcess
      - 11: OverrideMatchNewProcess
    - Views
  - HealthCare
    - Tables
    - Processes
    - Views
      - 1: 30 Purchase Sum
      - 2: 120 Loyalty Point Sum
      - 3: Flattened Customer with

**Table: MatchTest**

| bizusers                 | dbas                     | stewards                 |                                     |
|--------------------------|--------------------------|--------------------------|-------------------------------------|
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

Override Inherited Permissions

**Permissions for this Entity**

**Table**

|                                     |               |
|-------------------------------------|---------------|
| <input checked="" type="checkbox"/> | read_table    |
| <input checked="" type="checkbox"/> | modify_table  |
| <input checked="" type="checkbox"/> | delete_table  |
| <input checked="" type="checkbox"/> | query_record  |
| <input checked="" type="checkbox"/> | create_record |
| <input checked="" type="checkbox"/> | modify_record |
| <input checked="" type="checkbox"/> | delete_record |
| <input type="checkbox"/>            | mask_record   |
| <input type="checkbox"/>            | purge_record  |

Deleting the bizuser's permission entry for the MatchTest table

- On the row of groups to the right of the entity tree, click the group's associated delete icon . (If the group doesn't have an associated delete icon, select **Override Inherited Permissions**.)
- On the **Delete Group** popup, click **Delete**.

7.9

## Matching sub-document

Tables created with **matching** set in the schema will create documents with an optional **rpmdm\_match sub-document** . This sub-document contains the entirety of the matching state for a record, including:

- Automatic matching performed inside MDM
- Matching performed externally by RedPoint Data Management

- Match overrides performed by stewards
- Match groups

#### 7.9.1 The sub-document

The matching sub-document takes the following form:

```
"rpmdm_match":{
 type:{
 "type":"string",
 "group":{"id":long, "score":float},
 "auto": [
 {"other":"value", "score":float}
 ...
],
 "always":["value",...],
 "never":["value",...]
 },
 ...
}
```

Each sub-document in the **rpmdm\_match** document represents a separate *match type*.

The tags inside each element are:

| Tag          | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>type</b>  | The match type. This is used as the key into the map and is repeated within the match sub-doc. The default value is "default" (if nothing else is specified).                                                                                                                                                                                                                                                                                                                                                                |
| <b>group</b> | The match group to which this record has been assigned, if any: <ul style="list-style-type: none"> <li>• <b>id</b> is the 8-byte integer match group ID. Match group IDs are automatically generated and have no relation to record primary keys.</li> <li>• <b>score</b> is a number between 0.0 and 1.0 (inclusive) and indicates the quality of the match group. This score is synthesized from a combination of the inter-record match scores and a measurement of the match group inter-connection strength.</li> </ul> |
| <b>auto</b>  | An array of matches to other records that were made through automated matching. Each entry contains:                                                                                                                                                                                                                                                                                                                                                                                                                         |

| Tag           | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <ul style="list-style-type: none"> <li>• <b>other</b>: The primary key of the other record. This should not contain any primary keys contained in the always or never arrays.</li> <li>• <b>score</b>: The match score produced by the matching system. The value is between 0.0 and 1.0, and indicates the quality of the match. This score has no direct interpretation, although it is generally related to the edit distance between this record and other records of the group, and higher scores are better matches.</li> </ul> |
| <b>always</b> | <p>An array of primary keys indicating "always match" to other records. These are the result of manual force-match operations. This record will be grouped with all other records in the <b>always</b> list unless:</p> <ul style="list-style-type: none"> <li>• Doing so would place this record in a group with one or more records from the <b>never</b> list, or</li> <li>• The resulting group would break limits on group size or topology (see the specification of grouping rules for details).</li> </ul>                    |
| <b>never</b>  | <p>An array of primary keys indicating "never match" to other records. These are the result of manual break/split operations. This record will never be added to a group containing another record in this list.</p> <p><b>Note:</b> <b>never</b> takes priority over <b>always</b>.</p>                                                                                                                                                                                                                                              |

Note that **auto**, **always**, and **never** are symmetric (both records in the pairs should contain the same information).

7.10

## Aggregation language

MDM's [views](#)<sup>111</sup> support *aggregation projections* that can perform complex calculations on groups of records, summarizing each group down to a single output record. You specify the aggregation rules using a JSON document in which each computation is an expression loosely based on Java. This aggregation language is not simple, but it is extremely powerful.

#### 7.10.1 Expression syntax

Aggregating expressions are used within the aggregation projections of computed views. An aggregating expression is a declaration of the ultimate value of an output variable, *not* code that performs iterative operations as input variables are processed. An aggregating expression may reference input variables only through *aggregators*. It may also reference constants and Java methods (for example, `Math.abs`) and operations (for example, `+, -, *, /`).

In addition to aggregating expressions, there are expressions used as the terms for aggregators. These are simple expressions over input and output variables, without the aggregators.

#### 7.10.2 Special identifiers

A MDM aggregating expression is really a Java expression. We evaluate these expressions by transforming them into Java classes, compiling them, and running the compiled code. MDM uses a simple technique to distinguish between the things that MDM cares about, and the rest of the expression that is passed through to Java. To accomplish this, the aggregating language prefixes MDM identifiers with `$`:

- Input record field names
- Output record field names
- Temporary variable names
- Computed field names
- Selectors and aggregators
- MDM-provided functions

For example, in the following expression:

```
$Sum($Filter($TYPE==3), Math.abs($PURCH_QTY) / 2)
```

MDM recognizes these elements:

- `$Sum` is an aggregator.
- `$Filter` is a selector.
- `$TYPE` is an input field.
- `$PURCH_QTY` is an input field.

By contrast, `Math` and `abs` are not recognized or processed by MDM, but are passed through to Java. The aggregation library allows only a small documented set of Java imports such as `Math` and `Joda types`. Internally, MDM generates Java code and compiles it into a class that carries out the computations.

All `$` identifiers are case-insensitive and treated as lower case regardless of how they are input. For example, the following entities are treated as identical:

```
$FILTER
```

```
$filter
$FiLTER
```

#### 7.10.3 Backquoted strings

To make it easier for users to use double quotes in expressions, backquotes are treated as double quotes inside expression strings. For example, within an expression string, `hello` is the same as `\"hello\"`. Because we parse these expressions to include a backquote in a backquoted string, you need to escape the backquote that will still be converted into a double quote. To create the backquote character, use the unicode escape sequence \u0060 instead.

#### 7.10.4 Expression types

MDM's aggregation language uses two expression types: *aggregating expressions*, and *scalar expressions*.

##### Aggregating expressions

An aggregating expression is an *expression over aggregators*. It doesn't compute directly on field values, only on aggregated field values. All assignments to output fields or temporary variables are aggregating expressions. Aggregating expressions operate implicitly on "the group of records," but the group of records was defined in the computed view itself (such as "all transactions for a customer ID"). For example:

```
$Sum($Filter($TYPE==3), Math.abs($PURCH_QTY)) / 2.0
```

The above expression means "Find all records in the group where TYPE==3, total absolute value of the PURCH\_QTY field, and divide the result by 2.0." Note that this expression has only three elements as far as Java is concerned:

- `$Sum(...)` is an aggregator.
- `/` is a Java divide operator.
- `2.0` is a Java floating-point constant.

Of course, *inside* the `$Sum(...)` aggregator are other items that are not themselves aggregators, as discussed below.

##### Scalar expressions

A scalar expression computes over record variables. It can be a Boolean condition such as:

```
$PURCH_QTY > 0
```

It can be a scalar value such as:

```
$PURCH_AMOUNT / 100
```

It can involve Java language elements, such as:

```
Math.sqrt(Math.abs($PURCH_AMT)) + $FIRSTNAME.length()
```

It can be arbitrarily complex. However, it can only access the fields of one record at a time. Which record it is accessing is controlled by the aggregating machinery and is not the concern of the expression. Scalar expressions are used in several places:

- The *expression* argument to `Sum()`, `Min()`, and `Max()`.
- The *condition* argument to `Count()`.
- The *condition* in `Filter()`.

#### 7.10.4.1 Scalar versus aggregating usage

The two expression types (*aggregating expressions*, and *scalar expressions*) must be used in the appropriate context:

- [Selectors](#)<sup>164</sup> consume scalar expressions.
- [Aggregators](#)<sup>159</sup> consume scalar expressions.
- An output field must be assigned an aggregating expression.
- A [temporary variable](#)<sup>168</sup> must be assigned an aggregating expression.
- A [computed field](#)<sup>168</sup> must be assigned a scalar expression.
- An input field referenced as \$FIELD can only be used in scalar expressions.
- A computed field referenced as \$FIELD can only be used in scalar expressions.
- A temporary variable referenced as \$VARIABLE can be used in scalar or aggregating expressions.

For example, if you assign a temporary variable:

```
$NumRecordsInGroup = $Count($All())
```

You can later say:

```
$Filter(condition).$Sum($PURCHASE) / $NumRecordsInGroup
```

or:

```
$Filter(condition).$Sum($PURCHASE / $NumRecordsInGroup)
```

Similarly, if you assign a computed field:

```
$Rank = $SOURCE.equals("A") ? 1 : $SOURCE.equals("Z") ? 2 : 3
```

You can later say:

```
$First($Filter(condition).$Sort($Rank, descending), $VALUE)
```

But you cannot say:

```
$Filter(condition).$Sum($PURCHASE) / $Rank
```

### 7.10.5 Aggregators

An aggregator is a function that takes an *input set* and an expression:

`Function(selector, expression)`

The selector is represented as a chain of *selectors*, which taken together specify an ordered set of input records to process. The expression can be any Java expression over input fields and computed fields. For example, this expression counts the number of records where PROGRAM\_ID is equal to 3:

```
$Count($Filter($PROGRAM_ID==3))
```

This expression computes the average value of TRANSACTION over records where TYPE is equal to 10:

```
$Average($Filter($TYPE==10), $TRANSACTION)
```

The following aggregators are supported:

- [\\$Sum with optional type](#) [159]
- [\\$Min and \\$Max](#) [159] and [\\$Min and \\$Max with type](#) [160]
- [\\$Count](#) [161]
- [\\$Average](#) [161]
- [\\$First](#) [161] and [\\$First with type](#) [162]
- [\\$Nth](#) [163] and [\\$Nth with type](#) [163]

#### 7.10.5.1 \$Sum with optional type

The sum of the given *scalar-expression* over all records in *selector*.

Syntax

```
$Sum(selector, scalar-expression [, type])
```

Remarks

MDM supports BigDecimal, long, double, and String types. Summing dates and times is not currently supported. If no type is specified, \$Sum() returns a value of type double.

The sum of a string is the concatenation of the all of the string values. For example, \$Sum(..., *expression returning* ["a", "b", "c"]) produces the string abc.

The sum of an empty set is zero for numbers and the empty string ("") for strings.

#### 7.10.5.2 \$Min and \$Max

The minimum or maximum value of the given *scalar-expression* over all records in *selector*.

## Syntax

**\$Min( selector, scalar-expression )**

**\$Max( selector, scalar-expression )**

## Remarks

Returns a value of double. The minimum or maximum value of an empty set is zero (**0.0**).

### 7.10.5.3 \$Min and \$Max with type

The minimum or maximum value of the given *scalar-expression* over all records in *selector*.

## Syntax

**\$Min( selector, scalar-expression, type )**

**\$Max( selector, scalar-expression, type )**

## Remarks

Returns a value whose type is determined by the *type* argument. Choose one of the following keywords:

- BigDecimal
- Boolean
- Double
- Long
- LocalDate
- LocalDateTime
- LocalTime
- String

The *type* keywords are case-insensitive but are replaced with the case-sensitive version listed above. All of the date types are Joda types. The minimum or maximum value of an empty set depends on the type:

| Type       | Default value |
|------------|---------------|
| BigDecimal | 0             |
| Boolean    | false         |
| Double     | 0.0           |
| Long       | 0L            |

| Type          | Default value       |
|---------------|---------------------|
| LocalDate     | 1600-01-01          |
| LocalDateTime | 1600-01-01 00:00:00 |
| LocalTime     | 00:00:00 (Midnight) |
| String        | ""                  |

### Why have a type?

There are times you will be working with a data type other than double. Because these functions are used in arbitrary scalar expressions and our compiler is not part of the Java compiler, we cannot determine the correct type at compile time and instead require a hint from the user. This only matters if a default value is required. For the case of \$Min and \$Max, this is for the empty set.

#### 7.10.5.4 **\$Count**

The number of records in selector, returning a value of type long.

Syntax

**\$Count( selector )**

#### 7.10.5.5 **\$Average**

The average of the given *scalar-expression* over all records in *selector*.

Syntax

**\$Average( selector, scalar-expression )**

Remarks

Returns a value of type double. The average of the empty set is zero.

#### 7.10.5.6 **\$First**

Calculates *scalar-expression* for the first record in *selector*.

Syntax

**\$First( selector, scalar-expression )**

Remarks

This is typically done using \$Sort() as the selector to choose the "best value of the group" based on the ranking established in \$Sort.

Do not confuse this with the \$Top(1) selector. While they seem similar, they have completely different purposes:

- **\$First:** This aggregator evaluates its expression for the first record in the group, and returns it.
- **\$Top(1):** This selector reduces the record group to a single record and makes it available to aggregators.

If the selector returns the empty set, **\$First** will return the double value 0.0.

#### 7.10.5.7 **\$First with type**

Calculates *scalar-expression* for the first record in *selector*.

Syntax

**\$First( selector, scalar-expression, type )**

Use this version of **\$First** if your scalar expression will return a value with a type other than double. Returns a value whose type is determined by the *type* argument. Choose one of the following keywords:

- BigDecimal
- Boolean
- Double
- Long
- LocalDate
- LocalDateTime
- LocalTime
- String

The *type* keywords are case-insensitive but are replaced with the case-sensitive version listed above. All of the date types are Joda types. If the *selector* returns the empty set and a default value is needed, the *type* argument determines that default value as follows.

| Type          | Default value       |
|---------------|---------------------|
| BigDecimal    | 0                   |
| Boolean       | false               |
| Double        | 0.0                 |
| Long          | 0L                  |
| LocalDate     | 1600-01-01          |
| LocalDateTime | 1600-01-01 00:00:00 |

| Type      | Default value       |
|-----------|---------------------|
| LocalTime | 00:00:00 (Midnight) |
| String    | ""                  |

#### 7.10.5.8 **\$Nth**

Calculates *scalar-expression* for the nth record in *selector*.

Syntax

**\$Nth( selector, offset, scalar-expression )**

Remarks

The *offset* argument must be a constant integer value. Zero is not a valid offset.

If *offset* is positive, it is the one-based offset of element starting at the front of the record set. For example, 1 gets the first element, 4 gets the 4th element, and so on.

If *offset* is negative it is the one-based offset of element starting at the back of the record set. For example, -1 gets the last element, -2 gets the first from the last element, -3 the second from the last element, and so on.

The default return value is of type double.

#### 7.10.5.9 **\$Nth with type**

Calculates *scalar-expression* for the nth record in *selector*.

Syntax

**\$Nth( selector, offset, scalar-expression, type )**

Use this version of **\$Nth** if your scalar expression will return a value with a type other than double. Returns a value whose type is determined by the *type* argument. Choose one of the following keywords:

- BigDecimal
- Boolean
- Double
- Long
- LocalDate
- LocalDateTime
- LocalTime
- String

The *type* keywords are case-insensitive but are replaced with the case-sensitive version listed above. All of the date types are Joda types. If the *selector* returns the empty set and a default value is needed, the *type* argument determines that default value as follows.

| Type          | Default value       |
|---------------|---------------------|
| BigDecimal    | 0                   |
| Boolean       | false               |
| Double        | 0.0                 |
| Long          | 0L                  |
| LocalDate     | 1600-01-01          |
| LocalDateTime | 1600-01-01 00:00:00 |
| LocalTime     | 00:00:00 (Midnight) |
| String        | ""                  |

#### 7.10.6

#### Selectors

Selectors are pseudo-functions that can be chained together to specify the *selector* upon which the [aggregators](#) <sup>159</sup> are evaluated in an expression.

Selectors are chained together with a period character . and applied left-to-right. For example:

`$Filter($TYPE==4).$Sort($DATE,descending).$Top(5)`

selects the most recent five records where TYPE==4.

Selectors are automatically cached and shared within a set of aggregation rules. If the same chain of selectors is used repeatedly, code is generated automatically to avoid computing the set more than once per group.

The following selectors are supported:

- [\\$All](#) <sup>165</sup>
- [\\$Filter](#) <sup>165</sup>
- [\\$AgeDays](#) <sup>165</sup>
- [\\$Sort](#) <sup>166</sup>
- [\\$Top](#) <sup>166</sup>
- [\\$Unique](#) <sup>166</sup>
- [\\$MostCommon](#) <sup>166</sup>

**7.10.6.1 \$All**

Designates all input records of a record group.

Syntax

**\$All**

**7.10.6.2 \$Filter**

Filters the record set, retaining only records for which *condition-expression* evaluates to true.

Syntax

**\$Filter( condition-expression )**

Remarks

The *condition-expression* must evaluate to true/false.

**7.10.6.3 \$AgeDays**

Selects records whose *\$FIELD* is between *low* and *high* days old.

Syntax

**\$AgeDays( \$FIELD, low, high )**

Remarks

The \$AgeDays selector is a special kind of filter, selecting records whose *\$FIELD* (which must be Date or DateTime type) is between *low* and *high* days old (inclusive) relative to the current date. The low and high values are literal integers. Use of this filter triggers automatic re-computation of affected aggregations during a specified "middle of the night" window as records age in and out of the day range. For example:

`$AgeDays($PURCHASE_DATE,0,30)`

selects records whose PURCHASE\_DATE is between 0 and 30 days old.

To express "up to 30 days old," you should use `$AgeDays($FIELD,0,30)`, not `$AgeDays($FIELD,1,30)`. One day old means "yesterday" whereas zero days old means "today."

The time is never considered when performing the age calculation; it is stripped from both the current date/time and the field value. Only the date portion is considered. For example:

- If the current time is `2020-01-01T23:59:59` and the date field value is `2020-01-02T00:00:00`, the field is "one day old."
- If the current time is `2020-01-01T00:00:00` and the date field value is `2020-01-01T23:59:59`, the field is "zero days old."

#### 7.10.6.4 **\$Sort**

Sorts the record set by a list of input field names and key orders.

Syntax

**\$Sort(** *field1, order1 [, field2, order2... ]* **)**

Remarks

Each *order* is the keyword either *ascending* or *descending*.

#### 7.10.6.5 **\$Top**

Retains only the first N records of a record set.

Syntax

**\$Top(** *N* **)**

Remarks

N must be a constant integer  $\geq 1$ . For example:

**\$Sort(\$DATE,descending).\$Top(3)**

selects the three newest records.

#### 7.10.6.6 **\$Unique**

Reduces the record set by discarding records that are duplicated across the input variables *uniqueVar1 ... uniqueVarN*.

Syntax

**\$Unique(** *uniqueVar1 [, uniqueVar2...], [tieBreakField1, tieBreakOrder1,...]* **)**

Remarks

If additional consideration must be given to which records survive deduplication, a list of "tie breakers" may also be specified, each of which contains an input variable and an order. Note that ascending order keeps the lowest-valued records, and descending order keeps the highest-valued records. For example:

**\$Unique(\$SOURCE,\$DATEOFBIRTH,descending)**

Will select a group of records that are unique by the SOURCE field, breaking ties by choosing the most recent DATEOFBIRTH.

#### 7.10.6.7 **\$MostCommon**

Reduces the record set by selecting only records that have the most common value or values.

## Syntax

**\$MostCommon( *field* [ , *N* ] )**

## Remarks

The \$MostCommon selector is limited to using the first \${rpmdm.subscription.max\_delta\_size} records. If an aggregation has more than this number of records, only the first delta size (default 100,000) records are considered.

If *N* is not specified, it defaults to one. If *N* is specified using a positive integer value, records which have the *N* most common values are selected. *N* should be small, but is not constrained. The selector creates a histogram of the field value of all input records and then chooses the most commonly-occurring value(s) of that field.

- In case of ties, the record order is used as a tie breaker.
- The order of the input record is not preserved. The result is ordered from most common to least common. For a given value of the field, the records are left in the order found. For example, given values a, b, c, d, c, d, b, c the records are reordered to c, c, c, b, b, d, d, a. If *N* is less than 4, the result is ccc, cccbb, or cccbbdd, corresponding to values of N=1,2,3 respectively.
- If *N* is larger than the number of distinct values for the field, the input is reproduced but sorted as described above.
- The algorithm runs in O(*n*) or O(*n*) + O(log *n*) if *n* is small, where *n* is the number of input records.

For example, given this input:

| IndividualID | XactID | Qty | Amount | ProductID |
|--------------|--------|-----|--------|-----------|
| 100          | 1      | 1   | 17.45  | 535       |
| 100          | 2      | 2   | 1.56   | 432       |
| 100          | 3      | 1   | 463.00 | 535       |
| 100          | 4      | 4   | 34.17  | 145       |
| 100          | 5      | 1   | 100.00 | 532       |

running the selector

**\$MostCommon(\$ProductID)**

records are selected containing ProductID=535 (occurs twice vs all the other values) which:

| IndividualID | XactID | Qty | Amount | ProductID |
|--------------|--------|-----|--------|-----------|
| 100          | 1      | 1   | 17.45  | 535       |

| IndividualID | XactID | Qty | Amount | ProductID |
|--------------|--------|-----|--------|-----------|
| 100          | 3      | 1   | 463.00 | 535       |

`$MostCommon($ProductID, 2)` would select the following records in the following order:

| IndividualID | XactID | Qty | Amount | ProductID |
|--------------|--------|-----|--------|-----------|
| 100          | 1      | 1   | 17.45  | 535       |
| 100          | 3      | 1   | 463.00 | 535       |
| 100          | 2      | 2   | 1.56   | 432       |

#### 7.10.7 Temporary variables

A *temporary variable* is a scalar value that you can compute once and reuse later. This is helpful if you find yourself typing the same sub-expression multiple times. Some rules of thumb to remember:

- Temporary variables (like output fields) are an aggregation result. They are not computed per-record (see "computed fields" below for per-record computations).
- Temporary variable calculations are done before any output field aggregations.
- Temporary variable calculations are done in order. If there are any dependencies, put them in the right order!

Examples

```
$Last30Count = $Count($AgeDays($PURCH_DATE, 0,30))$outputField = $Last30Count > 12$anotherOutput = $Last30Count <
```

#### 7.10.8 Computed fields

A *computed field* acts like another field that is appended to each input record, except that its value is derived from the other input fields using a [scalar expression](#)<sup>157</sup>. For example, to compute a custom Rank field on each record that you can use later in [\\$Sort](#)<sup>166</sup>, you might try a conditional expression chain in Java:

```
$Rank = $SOURCE.equals("A") ? 1 : $SOURCE.equals("B") && $PURCHASE > 1000 ? 2 : 3
```

Refer to computed fields in later scalar expressions just like you would any other input field (using `$VARIABLE`).

#### 7.10.9 Java library support

Only the following Java and Joda objects are supported:

- `java.lang.Math`

- `java.lang.StrictMath`
- `java.math.BigDecimal`
- `java.util`
  - `java.util.ArrayList`
  - `java.util.HashSet` (although these classes are needed by the aggregation compiler and would be unusual to use in an expression)
- `org.joda.time`
  - `LocalDate`
  - `LocalDateTime`
  - `LocalTime`

#### 7.10.10 Support functions for expressions

MDM's aggregation language offers the following built-in functions that simplify some Java constructs to aid in string comparison and the coercion of scalar expressions to a particular type:

- [`\$Compare\(v1, v2\)`](#)  170
- [`\$Equals\(v1, v2\)`](#)  170
- [`\$StrCompare\(s1, s2\)`](#)  170
- [`\$StrEquals\(s1, s2\)`](#)  171
- [`\$DaysDiff\(dt1, dt2\)`](#)  171
- [`\$ToBigDecimal`](#)  171
- [`\$.ToBoolean`](#)  171
- [`\$ToDate`](#)  172
- [`\$ToDate`](#)  172
- [`\$toDouble`](#)  172
- [`\$toFloat`](#)  172
- [`\$toInteger`](#)  173
- [`\$toLong`](#)  173
- [`\$toString`](#)  173
- [`\$toTime`](#)  174

#### 7.10.10.1 **\$Compare**

Compares the expressions  $v1$  and  $v2$ .

Syntax

**\$Compare(  $v1$ ,  $v2$  )**

Remarks

Compares the expressions  $v1$  and  $v2$ , promoting variables to the "biggest type" (for example, in int versus long, the int is cast to long) and returns:

- a negative integer if  $v1 < v2$
- 0 if  $v1 == v2$
- a positive integer if  $v1 > v2$

Strings use a case-sensitive comparison. See [\\$StrCompare](#) [170].

#### 7.10.10.2 **\$Equals**

Compares two variables for equal value.

Syntax

**\$Equals(  $v1$ ,  $v2$  )**

Remarks

Compares two variables and returns *true* if they are equal and *false* otherwise. Strings use a case-sensitive comparison. See [\\$StrCompare](#) [170] for case-insensitive comparison. When used on values with differing types the variables with "smaller" types are promoted to larger types. For example, int becomes long, Date compared to DateTime will be promoted to DateTime, and so on.

#### 7.10.10.3 **\$StrCompare**

Compares the string expressions  $s1$  and  $s2$ .

Syntax

**\$StrCompare(  $s1$ ,  $s2$  )**

Compares the string expressions  $s1$  and  $s2$  (ignoring case) and returns:

- a negative integer if  $s1$  lexically precedes  $s2$
- 0 if  $s1 == s2$
- a positive integer if  $s1$  lexically follows  $s2$

For example, `$StrCompare("FooBIE!", "foobie!")` returns 0.

If  $s1$  or  $s2$  is not a string, it is converted to string before comparing.

#### 7.10.10.4 **\$StrEquals**

Compares two string expressions.

Syntax

**\$Equals( *v1, v2* )**

Remarks

**\$StrEquals( *s1, s2* )**

Compares the string expressions *s1* and *s2* (ignoring case) and returns *true* if they are lexically equal and *false* otherwise. For example, `$StrEquals("FooBIE!", "foobie!")` returns *true*. If *s1* or *s2* is not a string, it is converted to string before comparing.

#### 7.10.10.5 **\$DaysDiff**

Returns the difference in days between *dt1* and *dt2*.

Syntax

**\$DaysDiff( *dt1, dt2* )**

Remarks

The arguments *dt1* and *dt2* must be LocalDate or LocalDateTime instances. The returned value is of type integer. This number is positive if *dt2* is later than *dt1*, and negative if *dt2* is earlier than *dt1*.

For example, if *dt1* is 2000-01-01 and *dt2* is 2000-01-02:

`$DayDiff( $JAN_01_2000, $JAN_02_2000)`

returns 1.

#### 7.10.10.6 **\$ToBigDecimal**

Converts a scalar value to type BigDecimal.

Syntax

**\$ToBigDecimal( )**

Remarks

For numeric types, this is a one-to-one map. For non-numeric types, the value is first converted to a string and then to a BigDecimal. A run-time exception may occur if the input is not a number or its [\\$ToString](#)<sup>173</sup> value is not a number.

#### 7.10.10.7 **\$ToBoolean**

Converts a scalar value to Boolean.

## Syntax

**\$ToBoolean( )**

## Remarks

Numeric values of `0` (zero) return false. Numeric values other than zero return true. Non-empty strings return true (for example, `$toBoolean("false")` is true). Non-numeric, non-string instances return *false* if their [\\$ToString](#)<sup>173</sup> value is an empty string and *true* otherwise.

### 7.10.10.8 **\$ToDate**

Convert a LocalDate, LocalDateTime, or String instance to a LocalDate instance.

## Syntax

**\$ToDate( )**

## Remarks

Converting strings with invalid date formats or types other than LocalDate and LocalDateTime will cause run-time errors.

### 7.10.10.9 **\$ToLocalDateTime**

Convert a LocalDate, LocalDateTime, or String instance to a LocalDateTime instance.

## Syntax

**\$ToLocalDateTime( )**

## Remarks

Converting strings with invalid date formats or types other than LocalDate and LocalDateTime will cause run-time errors.

### 7.10.10.10 **\$toDouble**

Convert a Numeric, Boolean, or String value to a Double.

## Syntax

**\$toDouble( )**

## Remarks

Numeric values are cast to double. Boolean becomes 1.0/0.0 for true/false. String values are parsed. Invalid strings are treated as 0.0.

All other classes (for example, Dates) cause run-time errors.

### 7.10.10.11 **\$toFloat**

Convert a Numeric, Boolean, or String value to a Float.

## Syntax

**\$ToFloat( )**

## Remarks

Numeric values are cast to 32-bit float. Boolean becomes 1.0/0.0 for true/false. String values are parsed. Invalid strings are treated as 0.0.

All other classes (for example, Dates) cause run-time errors.

### 7.10.10.12 \$ToInteger

Convert a Numeric,Float, Boolean, or String value to an Integer.

## Syntax

**\$ToInteger( )**

## Remarks

Numeric values are cast to integer. Floating point values are rounded as per java.lang.Math.round(). (For example, **\$ToInteger(0.5)** is 1.) Boolean becomes 1 / 0 for true / false. String values are parsed. Invalid strings are treated as zero.

All other classes (for example, Dates) cause run-time errors.

### 7.10.10.13 \$ToLong

Convert a Numeric, Float, Boolean, or String value to a Long.

## Syntax

**\$ToLong( )**

## Remarks

Numeric values are cast to long integer. Floating point values are rounded as per java.lang.Math.round(). For example, **\$ToInteger(0.5)** is 1. Boolean becomes 1 / 0 for true / false. String values are parsed. Invalid strings are treated as zero.

All other classes (for example, Dates) cause run-time errors.

### 7.10.10.14 \$ToString

Converts the value to a String.

## Syntax

**\$ToString( )**

## Remarks

Runs the .toString() method on the instance.

#### 7.10.10.15 \$ToTime

Converts a LocalTime, LocalDate, LocalDateTime, or String instance to a LocalTime instance.

Syntax

**\$ToTime( )**

Remarks

Convert a LocalTime, LocalDate, LocalDateTime, or String instance to a LocalTime instance as per the Joda cast operation. Converting values of other types causes a run-time exception.

### 7.10.11 Aggregation expression notes

The following topics offer more in-depth information on aggregation expressions:

- [Rules and conventions](#)<sup>174</sup>
- [Java language considerations](#)<sup>175</sup>
- [MDM type to Java type mappings](#)<sup>175</sup>
- [Null values and Date/DateTime fields](#)<sup>176</sup>
- [Selecting multiple fields as a group](#)<sup>176</sup>
- [Multiple aggregators in an expression](#)<sup>177</sup>
- [Making decisions](#)<sup>177</sup>
- [Complex ranking](#)<sup>178</sup>

#### 7.10.11.1 Rules and conventions

When thinking about expressions, observe the following rules and conventions:

- Expressions have a passive role. They do not "loop" over records; they are handed records one at a time, or handed groups of records for aggregation.
- Expressions are declarative. Thus, they should not have any side effects (opening files, writing to the Web App, changing global variables, and so on). MDM restricts access to most things that could have side effects.
- Numeric variables that are null on input are given the value zero (0).
- Date fields that are null on input are given start of epoch default value 1600-01-01.
- Time fields that are null on input are given the default value 00:00:00.000.
- DateTime fields that are null on input are given the default value 1600-01-01T00:00:00.000.
- String fields that are null on input are given the value "" (empty string).

- Because nulls are mapped to default values, special variables of the form \$isXXXValid (where XXX is the input field name) are automatically generated so that rules can test for null explicitly.
- Java comma expressions (statement, statement) are not supported.

#### 7.10.11.2 Java language considerations

Because expressions are essentially Java with placeholders for the special aggregating functionality, you can use most language constructs that you can use in Java itself. In addition, fields in records become data fields in Java classes that have correct semantics and rich types. Specifically:

- Mathematical operators: + - \* / %
- Bitwise operators: << >> >>> & | ~ ^
- Logical operators: && || ! == != < > >= <=
- The conditional expression: condition ? expr1 : expr2
- Parenthetical grouping: ( )
- Bracket operators: [ ]
- The functions of the Math and StrictMath packages
- Methods of LocalDate, LocalTime, and LocalDateTime fields. Note, for example, that `LocalDate.now()` returns the current date.
- Do not compare strings using = or ==. Use `$Equals`<sup>[170]</sup> or `$StrEquals`<sup>[171]</sup>.
- Do not compare strings using < or >. Use `$Compare`<sup>[170]</sup> or `$StrCompare`<sup>[170]</sup>.
- Aggregation language does not support comma expressions. For example, `while( x >>>= 1, x == 0)` is unsupported.
- A backquote is treated as a double quote. Use \u0060 to create the ` character in a string.

#### 7.10.11.3 MDM type to Java type mappings

| MDM type                                               | Java type        | Description                                                              |
|--------------------------------------------------------|------------------|--------------------------------------------------------------------------|
| all integer types (8, 16, 32, 64, signed and unsigned) | long (64 bits)   | 64 bit unsigned is not allowed.<br>All integer types are coerced to long |
| all floating point types (32, 64)                      | double           |                                                                          |
| text (string)                                          | java.lang.String |                                                                          |
| boolean                                                | boolean          | true, false (no null)                                                    |

| MDM type | Java type                   | Description                                                   |
|----------|-----------------------------|---------------------------------------------------------------|
| date     | org.joda.time.LocalDate     | date with no timezone                                         |
| time     | org.joda.time.LocalTime     |                                                               |
| datetime | org.joda.time.LocalDateTime | date/time with no timezone                                    |
| binary   | byte[]                      | This is not useful for aggregations, in general               |
| money    | BigDecimal                  | Roughly equivalent to a decimal with precision=20 and scale=4 |

#### 7.10.11.4 Null values and Date/DateTime fields

Newest by Date/DateTime

Because null Date/DateTime fields are replaced by "epoch" dates at 1600-01-01, finding the newest records generally works as expected without special consideration for null. For example:

```
$Sort($DATEOFBIRTH,descending).$Top(3)
```

selects the three newest records.

Oldest by Date/DateTime

Because null Date/DateTime fields are replaced by "epoch" dates at 1600-01-01, finding the oldest records needs extra filtering if nulls may be present in the data and you don't want to select them. For example:

```
$Filter(!$isDATEOFBIRTHNull).Sort($DATEOFBIRTH,ascending).$Top(3)
```

selects the three oldest records.

#### 7.10.11.5 Selecting multiple fields as a group

This is useful when you have one ranking criterion that governs multiple fields as a group. These aggregation rules often look like "choose the best address record using these ranking criteria, then choose STNAME, STNUM, UNIT, UNITNBR, CITY, STATE, and ZIP all at once." It looks more difficult than it actually is.

The first thing is to establish the ranking criteria, which is typically some combination of source priority and recency, such as:

```
$Filter(!$isADDRESSNull && !$isCITYNull &&
 !$isZIPNull).$Sort($DATE,descending,$SOURCEPrio,descending)
```

Then you repeat that same ranking criteria for each field:

```
STNUM : $First($Filter(!$isADDRESSNull && !$isCITYNull &&
!$isZIPNull).$Sort($DATE,descending,$SOURCEPRIO,descending), $STNUM)STNAME
: $First($Filter(!$isADDRESSNull && !$isCITYNull &&
!$isZIPNull).$Sort($DATE,descending,$SOURCEPRIO,descending), $STNAME)...
ZIP : $First($Filter(!$isADDRESSNull && !$isCITYNull &&
!$isZIPNull).$Sort($DATE,descending,$SOURCEPRIO,descending), $ZIP)
```

This looks inefficient, but it is not because the `$Filter().$Sort()` selector chain is cached and reused for all of these expressions.

#### 7.10.11.6 Multiple aggregators in an expression

Recall that the aggregating expression syntax allows for more than one aggregator to be embedded in a single expression. This is useful when you need to express different aggregations over the same (or different) record sets.

Example: Find ratio of offers to responses

```
$Count($Filter($EVENT_TYPE.equals("offer")) / $Count(All(),
$EVENT_TYPE.equals("response"))
```

Example: Find difference between highest and lowest purchase

```
Math.abs($Max($Filter($PURCHASE > 0), $PURCHASE) - $Min($Filter($PURCHASE > 0),
$PURCHASE))
```

Example: Find ratio of current and past YTD sales

This is somewhat complex, because the Year To Date concept is tricky to express using the `LocalDate` interface.

Filter for this year:

```
$Filter($DATE.getYear() == LocalDate.now().getYear())
```

Filter for last year YTD:

```
$Filter($DATE.getYear() == LocalDate.now().getYear() - 1 && $DATE.getDayOfYear() <=
LocalDate.now().getDayOfYear())
```

Combined expression:

```
$Sum($Filter($DATE.getYear() == LocalDate.now().getYear()), $PURCHASE) /
$Sum($Filter($DATE.getYear() == LocalDate.now().getYear() - 1 && $DATE.getDayOfYear() <=
LocalDate.now().getDayOfYear()), $PURCHASE)
```

**Note:** This construct will not trigger nightly re-computation of affected aggregations.

#### 7.10.11.7 Making decisions

You can create expressions that make decisions based on aggregations using a Java *conditional expression*.

For example, "if total purchases this year exceed 10000, then A else B":

```
$Sum($Filter($DATE.getYear() == LocalDate.now().getYear()) > 10000 ? "A" : "B")
```

### 7.10.11.8 Complex ranking

Sometimes your ranking criteria for the [\\$Sort](#)<sup>166</sup> selector is more than a simple calculation over fields. In this case, use a chain of *conditional expressions* to map conditions to ranks:

```
condition1 ? 1 : condition2 : 2 : condition3 ? 3 ... : N
```

For example:

```
$Sort($SOURCE.equals("A") ? 1 : $SOURCE.equals("B") && $CODE == 3 ? 2 :
$SOURCE.equals("B") && $CODE == 9 ? 3 : $ALTBIZ > 10 ? 4 : 5, descending)
```

To use this ranking in multiple places, you may find it easier to assign it to a [computed field](#)<sup>168</sup>:

```
$RANK = $SOURCE.equals("A") ? 1 : $SOURCE.equals("B") && $CODE == 3 ? 2 :
$SOURCE.equals("B") && $CODE == 9 ? 3 : $ALTBIZ > 10 ? 4 : 5
```

Then later you can use the RANK field wherever you need it:

```
$Sort($RANK, descending)
```

## 7.11 User configurable properties

In MDM, user-configurable properties are set in:

- [MDM core services](#)<sup>178</sup>
- [MDM UI configuration file](#)<sup>186</sup>

### 7.11.1 MDM core services

MDM core services options can be set via:

- [MDM core services options](#)<sup>178</sup>
- [MDM core services properties file](#)<sup>179</sup>

#### 7.11.1.1 MDM core services options

These are options that can be set on the MDM core services command-line or via environment variables. These are different than the settings found in the [MDM core services properties file](#)<sup>179</sup>. Certain settings are available only via command-line options and environment variables, because the loading of properties files happens too late during startup.

Command-line options (these options have no *values*, they are bare options)

| Option         | Meaning                                                             |
|----------------|---------------------------------------------------------------------|
| -t             | Set "trace" level logging. This is the most detailed logging level. |
| --traceLogging | Set "trace" level logging. This is the most detailed logging level. |

| Option         | Meaning                                                                                  |
|----------------|------------------------------------------------------------------------------------------|
| -d             | Set "debug" level logging. This is less detailed than "trace" but more than the default. |
| --debugLogging | Set "debug" level logging. This is less detailed than "trace" but more than the default. |

Alternatively, these environment variables may be set. The value of the environment variable is not important.

For example on Linux systems:

```
setenv MDM_JAVA_TRACE_LOGGING
```

| Option                 | Meaning                                                                                  |
|------------------------|------------------------------------------------------------------------------------------|
| MDM_JAVA_TRACE_LOGGING | Set "trace" level logging. This is the most detailed logging level.                      |
| MDM_JAVA_DEBUG_LOGGING | Set "debug" level logging. This is less detailed than "trace" but more than the default. |

If there are conflicting log-level options, the highest level will be used.

#### 7.11.1.2 MDM core services properties file

The following properties are recognized by the MDM core services. These are typically stored in a Spring properties resource, but can be overridden on the command-line (if running using `mdm-quickstart` application), or by supplying a properties override file in one of these locations:

- Windows: `C:/Program Files/RedpointMDM/redpointmdm.properties`
- Linux: `/usr/local/share/redpointmdm/redpointmdm.properties`

##### 7.11.1.2.1 Sample file

You can start with `redpointmdm.properties` as a template:

```
#Uncomment lines that you want to override

#rpmdm.authentication.systemsecret=secret
#rpmdm.cache.block_cache_size=10485760
#rpmdm.cache.max_age=300
#rpmdm.cache.cursor_cache_size=32
#rpmdm.testing.mock_users=false

#rpmdm.mongodb.host=localhost
#rpmdm.mongodb.port=27017
#rpmdm.mongodb.user=admin
#rpmdm.mongodb.password=redpoint
#rpmdm.mongodb.authdb=admin
#rpmdm.mongodb.database=#rpmdm_system
#rpmdm.mongodb.ssl=false
#rpmdm.mongodb.flavor=mongodb
#rpmdm.mongodb.query_limit_in_values=1000
#rpmdm.mongodb.network_error_retry_seconds=30
#rpmdm.mongodb.uri=

#rpmdm.service.host=0.0.0.0
#rpmdm.service.port=12345
#rpmdm.service.secure=false
#rpmdm.service.logging=false
#rpmdm.testing.mock_users=false
#rpmdm.mongodb.database=#rpmdm_system
#rpmdm.workflow.clear_persistent_store=false

#rpmdm.authentication.url=http://localhost:9901/mdm
#rpmdm.usermodel.url=http://localhost:9903/mdm
#rpmdm.permission.url=http://localhost:9908/mdm

#rpmdm.aggregation.source-directory=
#rpmdm.views.no_load=false
#rpmdm.views.clear=false
#rpmdm.subscription.max_delta_size=100000
```

#### 7.11.1.2.2 File format

These are text files, with each line containing:

property=value

| Property                 | Description                          | Default                                                                |
|--------------------------|--------------------------------------|------------------------------------------------------------------------|
| rpmdm.mongodb.uri        | The MongoDB connection string "URI". | mongodb://admin:redpoint@localhost:27017/rpmdm_system?authSource=admin |
| rpmdm.authentication.url | URL to the authentication service.   | http://localhost:9901/mdm                                              |
| rpmdm.usermodel.url      | URL to the user/group service.       | http://localhost:9903/mdm                                              |

| Property                                 | Description                                                                                                                                                                      | Default                                                                                                                          |
|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| rpmdm.permission.url                     | URL to the permissions service.                                                                                                                                                  | http://localhost:9908/mdm                                                                                                        |
| rpmdm.mongodb.flavor                     | What flavor of MongoDB is this?<br>One of: <ul style="list-style-type: none"><li>• mongodb</li><li>• cosmosdb</li><li>• atlas</li></ul>                                          | mongodb                                                                                                                          |
| rpmdm.mongodb.query_limit_in_values      | Maximum number of values that can be processed in a single \$in term. Currently this has been reduced to accommodate Cosmos DB which has a lower \$in clause limit than MongoDB. | 490 (Cosmos limit 500.<br>MongoDB limit is the 16MB of Bson document size so ~100K to ~1.5M depending on the size of the values) |
| rpmdm.mongodb.network_wait_seconds       | Initial wait time in seconds after a network exception occurs talking to MongoDB.                                                                                                | 1                                                                                                                                |
| rpmdm.mongodb.network_max_wait_seconds   | Maximum wait time between tries to contact MongoDB.                                                                                                                              | 32                                                                                                                               |
| rpmdm.mongodb.network_total_wait_seconds | Maximum total time we try to contact MongoDB from a network error before we give up.                                                                                             | 160                                                                                                                              |
| rpmdm.mongodb.max_write_size             | The maximum number of records which can be written in a single bulkWrite style operation.                                                                                        | 10000                                                                                                                            |
| rpmdm.mongodb.max_awaits                 | The maximum number of times we can await for an database operation can fail.                                                                                                     | 100                                                                                                                              |
| rpmdm.cosmos.ru_wait_ms                  | Initial wait time in milliseconds to wait before retrying an single operation that failed due to (over) resource utilization. This property only applies to Cosmos DB.           | 50                                                                                                                               |
| rpmdm.cosmos.ru_max_wait_ms              | Maximum wait time between retries when an operation failed due to resource usage. This                                                                                           | 8000                                                                                                                             |

| Property                           | Description                                                                                                                                                                                                                                                  | Default  |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                    | property only applies to Cosmos DB.                                                                                                                                                                                                                          |          |
| rpmdm.cosmos.ru_total_wait_ms      | Maximum total wait time before failing an operation that has repeatedly failed due to resource usage limitations. This property only applies to Cosmos DB.                                                                                                   | 30000    |
| rpmdm.cosmos.ru_max_bulk_wait_ms   | Time to wait for a bulk operation to complete before changing strategy from a single bulk operation to a chunked record operation mechanism. The other ru_wait values listed above are used for each sub-operation. This property only applies to Cosmos DB. | 3000     |
| rpmdm.db.max_unique_db_attempts    | Maximum number of attempts to create a unique database name.                                                                                                                                                                                                 | 10       |
| rpmdm.authentication.system_secret | Secret token to pass to get_system_token authentication call.                                                                                                                                                                                                | secret   |
| rpmdm.cache.block_cache_size       | Maximum size of result blocks to cache, in bytes.                                                                                                                                                                                                            | 10485760 |
| rpmdm.cache.cursor_cache_size      | Maximum number of open query cursors to cache.                                                                                                                                                                                                               | 32       |
| rpmdm.cache.max_age                | Retention period, in seconds, for cached blocks and cursors.                                                                                                                                                                                                 | 300      |
| rpmdm.cache.disable                | Prevent the server from caching user requests regardless of the request "nocache" field value. Used for debugging.                                                                                                                                           | false    |
| rpmdm.views.no_load                | Do not load views from database. Normally used only for testing.                                                                                                                                                                                             | false    |
| rpmdm.views.clear                  | Clear views from database upon startup. Normally used only for                                                                                                                                                                                               | false    |

| Property                                 | Description                                                                                                                                                                                                                                                                                                            | Default            |
|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
|                                          | testing.                                                                                                                                                                                                                                                                                                               |                    |
| rpmdm.workflow.badge_update_rate_seconds | Number of seconds between workflow badge updates via websockets                                                                                                                                                                                                                                                        | 5                  |
| rpmdm.workflow.clear_peristent_store     | For testing only:<br>Clear all processes, workflows, and workitems before starting.                                                                                                                                                                                                                                    | false              |
| rpmdm.aggregation.source-directory       | Root folder where auto generated aggregation Java files are saved.<br>If empty, no files are saved.                                                                                                                                                                                                                    | ./generated-source |
| rpmdm.subscription.max_delta_size        | The largest chunk of changes that can be processed using "delta" calculations.<br>Larger changes will induce a batch recomputation of the view.<br>Prevents excessive memory use.<br>This metric is for the number of record keys in a set.<br>Larger data structures like records will count as more than one "unit". | 100000             |
| rpmdm.service.host                       | Host (bound interface) of published MDM service.<br>If this is "localhost", only local clients be able to connect.<br>If this is "0.0.0.0", publish on all IP addresses.                                                                                                                                               | 0.0.0.0            |
| rpmdm.service.port                       | Port of published service.                                                                                                                                                                                                                                                                                             | 12345              |
| rpmdm.service.secure                     | True to use https.                                                                                                                                                                                                                                                                                                     | false              |
| rpmdm.service.logging                    | True to log requests and responses to MDM core services.                                                                                                                                                                                                                                                               | false              |
| rpmdm.proxy.logging                      | True to log requests and responses that MDM core services make to other services.                                                                                                                                                                                                                                      | false              |
| rpmdm.service.cors_allow_origin          | Controls CORS requests for security.                                                                                                                                                                                                                                                                                   | *                  |

| Property | Description                                             | Default |
|----------|---------------------------------------------------------|---------|
|          | Set as Access-Control-Allow-Origin header in responses. |         |

The following properties are used only for testing:

| Property                                             | Description                                                                                                                                  | Default |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|---------|
| rpmdm.testing.mock_users                             | Create fake users and groups for testing.                                                                                                    | false   |
| rpmdm.testing.mongodb_proxy.enable                   | Enable the connection proxy.                                                                                                                 | false   |
| rpmdm.testing.mongodb_proxy.host                     | Network-fail testing proxy will publish proxy to this host.                                                                                  |         |
| rpmdm.testing.mongodb_proxy.port                     | Network-fail testing proxy will publish proxy on this port.<br>If replica sets or shards are in use, this will be the first port of several. |         |
| rpmdm.testing.mongodb_proxy.connection_timeout_ms    | Override the connection timeout in the MongoDB URI.                                                                                          |         |
| rpmdm.testing.mongodb_proxy.socket_timeout_ms        | Override the socket timeout in the MongoDB URI.                                                                                              |         |
| rpmdm.testing.mongodb_proxy.server_select_timeout_ms | Override the server-selection timeout in the MongoDB URI.                                                                                    |         |
| rpmdm.testing.mongodb_proxy.init_delay_ms            | Delay in milliseconds before connection proxy opens at initialization time                                                                   | 0       |

To override a property on the command-line of `mdm-quickstart`, use `--property=value`, for example:

```
--rpmdm.mongodb.uri=mongodb://redpoint:admin@dbserver.example.com:27017/rpmdm_system?
authSource=admin
```

#### 7.11.1.2.3 Properties for unit testing

In order to set these properties while running the unit tests, independently of the production properties, use these file paths:

- Windows: `C:/Program Files/RedpointMDM/redpointmdm.test.properties`

- Linux: `/usr/local/share/redpointmdm/redpointmdm.test.properties`

### 7.11.2 MDM UI configuration file

In the MDM user interface (UI), configuration is handled via the `redpoint_default.js` file, located in the base `mdm-ui` directory (in the same place as `app.js`). It can be overridden and modified. The configuration is stored as plain JSON.

A custom configuration can be imported from any location, and multiple import locations may also be used.

#### 7.11.2.1 Properties list

The file `redpoint_default.js`, located in the top level directory of the JavaScript services ("JS Services"), contains all of the default configuration of ports and hosts and so on. These should not be modified by administrators. Instead, create another file that contains the desired values (as a JSON object). The file name is operating system dependent:

`/usr/local/share/redpointmdm/redpoint_properties.js` (Unix)  
`C:/Program Files/RedpointMDM/redpoint_properties.js` (Windows)

If `$NODE_ENV` is not set or is any value other than "test", the startup will read the user properties file and merge it into the default properties file, overriding the defaults.

```
module.exports = {
 /* Don't run permissions service */
 "RUN_PERM": false,
 /* Set the system password to something better than 'system' */
 "authentication": {
 "system_password": "M0nk3yBadge3comm$ntbADger67%**"
 }
};
```

| Section | Property | Description                                 | Default                    |
|---------|----------|---------------------------------------------|----------------------------|
| imports |          | An array of valid imports.                  | [ "./ui-vol/config.json" ] |
| mdm     | host     | The MDM http(s) web service host name.      | localhost                  |
| mdm     | port     | The MDM web service port.                   | 9090                       |
| mongodb | host     | The MongoDB service hostname or IP address. | localhost                  |
| mongodb | port     | The MongoDB service port number.            | 27017                      |

| <b>Section</b> | <b>Property</b> | <b>Description</b>                                                                                        | <b>Default</b> |
|----------------|-----------------|-----------------------------------------------------------------------------------------------------------|----------------|
| mongodb        | user            | MongoDB service user name the services should use to log into MongoDB.                                    | admin          |
| mongodb        | password        | MongoDB service password. This is in plaintext.                                                           | redpoint       |
| mongodb        | authdb          | Database containing user authentication information. This is deprecated as of MongoDB v3.6.               | admin          |
| mongodb        | mdm_db          | The MDM system database to use.                                                                           | rpmdm_system   |
| mongodb        | uri             | Full MongoDB connection string to override all other information (user, password, host, port, and so on). |                |
| authentication | host            | Authentication service hostname or IP address.                                                            | localhost      |
| authentication | port            | Authentication service port number.                                                                       | 9901           |
| authentication | systemsecret    | Secret token to pass to get_system_token authentication call.                                             | secret         |
| user           | host            | User service hostname or IP address.                                                                      | localhost      |
| user           | port            | User service port number.                                                                                 | 9903           |
| permission     | host            | The Permission service hostname.                                                                          | localhost      |

| Section    | Property       | Description                                                                                             | Default   |
|------------|----------------|---------------------------------------------------------------------------------------------------------|-----------|
| permission | port           | The Permission service port number.                                                                     | 9908      |
| rpmdm      | host           | The MDM service hostname.                                                                               | localhost |
| rpmdm      | port           | The MDM service port number.                                                                            | 12345     |
|            | RUN_WEB_SERVER | Set to true to run it, false to not run the web service. This is at the top level and not in a section. | true      |
|            | RUN_PERM       | Set to true to run the permissions service on this node. This is at the top level and not in a section. | true      |
|            | RUN_USER       | Set to true to run the user service on this node. This is at the top level and not in a section.        | true      |

#### 7.11.2.2 Connection to Cosmos DB

The following is an example `redpoint_properties.js` file that connects to an Azure Cosmos DB where the host and user are named "rpdmdev-mongodb". Replace every instance of `rpdmdev-mongodb` with your own host and username. Use your password for the password placeholders.

```
module.exports = {
 "mongodb" : {
 "host": "rpdmdev-mongodb.documents.azure.com",
 "port": 10255,
 "user": "rpdmdev-mongodb",
 "password": "<bin-hex encoded string provided by Azure that looks like qwGaKrERrdd78h5o...==",
 "options": "?ssl=true&replicaSet=globaldb",
 "uri" : "mongodb://rpdmdev-mongodb:qwGaKrERrdd78h5o...==@rpdmdev-mongodb.documents.azure.com:10255/?ssl=true&replicaSet=globaldb"
 }
};
```

#### 7.11.2.3 Usage and internals

Internally, the file `configHandler.ts` determines how imports and overrides function.

For example, if I have a `config.json` in my ui-vol with the following:

```
{
 "mdm": {
 "host": "coolermdm.io",
 "port": 8080
 }}
This will override the "mdm" section of the config, but everything else will remain intact.
Note that invalid JSON will result in errors.
For development, the choice of whether to use config.json or getConfig() depends on your use case. getConfig() is dynamic and requires async/await or promises. config.json is the plain JSON file and does not evaluate imports.
```

# Index

## - A -

add  
     fields 40  
 aggregation  
     expressions 157, 158  
     projections 155  
 aggregation language 155  
     aggregators 159  
     backquoted strings 157  
     complex ranking 178  
     computed fields 168  
     expression syntax 156  
     expression types 157, 158  
     functions 169  
     Java language considerations 175  
     Java library support 168  
     making decisions 177  
     MDM type to Java type mappings 175  
     multiple aggregators in an expression 177  
     null values and date/datetime fields 176  
     rules and conventions 174  
     selecting multiple fields as a group 176  
     selectors 164  
     special identifiers 156  
     temporary variables 168  
 aggregators 159  
     average 161  
     count 161  
     first 161, 162  
     min/max 159, 160  
     nth 163  
     sum 159  
 always key 109  
 architecture 10  
 assistance  
     customer 11

## - C -

change history 99, 100  
 changesets 101  
     conflict resolution 97

    data 95, 96  
     metadata 95, 96  
     open 102  
     view 97  
 client 25  
 column entity 142  
 components 10  
 configuration 12, 21, 178, 179, 185  
 conflict resolution 97  
 contact RedPoint Data Management 11  
 create  
     databases 39  
     tables 40  
 customer assistance 11

## - D -

data  
     hub 29  
     masking 122  
     view 99, 100, 101, 102  
     visibility 122  
 data models  
     advanced topics 87  
     changes 40  
     historical 95  
 databases  
     advanced topics 87  
     change 30  
     create 39  
     delete 39  
     entity 142  
     select 30  
     supported 87  
 delete  
     databases 39  
     fields 40  
     tables 40  
 diagrams  
     component 10  
     console 27  
     security 129  
 Docker commands 19

## - E -

edit an existing record 46

enter a new record 43  
entities 141  
    add a permission entry for a group 151  
    change permissions 150  
    child 142  
    column 142  
    database 142  
    delete a permission entry for a group 151  
    effective set of permissions for a user 150  
    multiple permission entries 150  
    parent 142  
    permissions 146  
    process 142  
    subscription 142  
    system 142  
table 142  
trees 149  
view 142  
workflow 142  
erase record 122

## - F -

field data types  
    boolean 90  
    date 90  
    datetime 90  
    float 90  
    integer 90  
    linkto 90  
    money 90  
    table 90  
    text 90  
    time 90  
field options  
    Display 89  
    Field 89  
    Index 89  
    Nullable 89  
    PII 89  
    Signed 89  
    Size 89  
    Type 89

fields  
    add 40  
    add linkage 95  
    change order 40  
    computed 168

data types 90  
delete 40  
linkage 92, 94  
masking 122  
options 89  
timestamp 92  
visibility 122

## - G -

GDPR 122  
getting help 12  
golden record 8, 91  
groups 129  
    browse 136  
    create 136  
    delete 139  
    edit 137  
    everyone 130  
    roles 139  
    special 130  
    system 130, 141, 149

## - H -

hashing 122, 124  
historical data model 95  
history  
    view historical data 100  
    view records 36

## - I -

installation  
    components 12  
    Linux 18  
    prerequisites 14  
    Windows 14

## - K -

keys  
    always 109  
    never 109

**- L -**

linkage fields 92, 94  
adding 95

log

off 27  
on 26

**- M -**

masking 122, 123, 125, 126  
data 122  
fields 122

match document 109  
always key 109  
never key 109

match groups  
review automatically generated 49  
review manually generated 52

matching  
advanced 108  
automatic 109  
automatic match review workflow 110  
batch 110  
document 109  
match group IDs 108  
match review modes 108  
match segmentation keys 108  
match suppression 109  
match table primary keys 108  
multiple match types 108  
real-time 110  
review automated matches 49  
review manual matches 52  
sub-document 153, 154  
type 154

MDM Delete tool 68  
configure 70  
parameters 68

MDM Diff Report tool 86  
configure 87  
parameters 86

MDM Dynamic Query tool 72  
configure 74  
parameters 72

MDM Generate Match Group IDs tool 79

configure 79  
parameters 79

MDM Input tool 59  
configure 61  
parameters 60

MDM Key Query tool 70  
configure 71  
parameters 70

MDM Match Group Query tool 80  
configure 81  
parameters 80

MDM Match Suppression Query tool 82, 109  
configure 83  
parameters 82

MDM Output tool 62  
configure 65  
parameters 62

MDM Query View tool 74, 120  
configure 76  
parameters 75

MDM Set Match Groups tool 76, 110  
configure 78  
parameters 77

MDM Start Match Review Workflow tool 84, 110  
configure 85  
parameters 84

MDM tools  
MDM Match Suppression Query 109  
MDM Query View 120  
MDM Set Match Groups 110  
MDM Start Match Review Workflow 110  
shared settings 56  
tool execution settings 57

MDM Update tool 66  
configure 68  
parameters 66

mdmscript 20

metadata  
modify 39, 104  
versioned 40, 104

MongoDB  
trigger input 59  
trigger output 59

**- N -**

nested tables 92  
adding 93

never key 109

## - O -

overview 8

## - P -

permissions 141

entries 149

hierarchy 142

inheritance 142

synchronization 149

versioning 149

phone support 11

PII 124

port assignments 21

prerequisites 14

privacy 122

problems 12

process entity 142

product support 11

projections 112

aggregation 112

join 112

table 112

types 112

properties

MDM core services 178, 179, 185

MDM core services options 178

MDM core services properties file 179

MDM UI configuration file 185

user-configurable 178, 179, 185

purging 122, 123, 125, 126

## - Q -

queries

elements 33

example 35

how to construct 33, 34

operations 33

query MDM view data 120

record 32

term 33

## - R -

records

display masked 34

edit existing 46

enter new 43

erasure 122

golden 8, 91

hashing 122

history 125

masking 122, 123, 125, 126

purging 122, 123, 125, 126

query 32

verify erasure 122

verify masking 127

verify purging 127

view history 36

review

automated matches 49

manual matches 52

roles

ui elements 139

## - S -

scalar expressions 157, 158

schemas

add fields 40

change field order 40

delete fields 40

modify 40, 104

modify attributes 40

scripting tool 20

security 129

selectors 164

agedays 165

all 165

filter 165

mostcommon 166

sort 166

top 166

unique 166

services access 21

shared tool settings 59

site settings

tools 59

sub-document 153, 154  
 subscriptions 111  
     entity 142  
 support  
     phone 11  
     product 11  
     technical 11  
     telephone 11  
 support functions 169  
     compare 170  
     daysDiff 171  
     equals 170  
     strCompare 170  
     strEquals 171  
     toBigDecimal 171  
     toBoolean 171  
     toDate 172  
     toDateTime 172  
     toDouble 172  
     toFloat 172  
     toInteger 173  
     toLong 173  
     toString 173  
     toTime 174  
 system entity 142  
 system requirements 12

**- T -**

table options  
     Description 88  
     Display 122  
     Enable DM Matching Support 88  
     PII 122  
     Primary Key 88  
     Sequence Field 88  
     Type 88  
 tables  
     add fields 40  
     add nested 93  
     change field order 40  
     change history 99  
     configure for masking or purging 125  
     create 40  
     delete 40  
     delete fields 40  
     entity 142  
     modify 40

modify attributes 40  
 modify metadata 104  
 modify schema 104  
 nested 92  
 options 88  
 query records 32  
 schemas 104  
 search for from data page 31  
 search for from tables page 30  
 time series 92  
 versioned 91, 104  
 technical support 11  
 telephone support 11  
 temporary variables 168  
 time series tables 92  
 tool settings 59  
 trigger input 59  
 trigger output 59  
 troubleshooting 12

**- U -**

users 129  
     browse 131  
     create 132  
     delete 135  
     edit 133  
     special 130  
     system 130

**- V -**

verify  
     record was erased 122  
     record was masked 127  
     record was purged 127  
 versioned tables 91  
 views 111  
     changeset data 101, 102  
     create subscription 118  
     current data 100  
     data 99, 100, 101, 102  
     defining 112  
     entity 142  
     examples 113, 115  
     historical data 100  
     query MDM view data 120

views 111  
subscribing to 118

## - W -

web app  
administration section 129  
compliance section 126, 127  
components 27  
data hub section 29  
diagram 27  
log off 27  
log on 26  
MDM 25  
workflow section 42

workflows  
create 120  
dashboard notification 121  
definition 41, 120  
edit an existing record 46, 121  
enter a new record 43, 121  
entity 142  
escalation 121  
quarantine 121  
review automated matches 49, 121  
review manual matches 52, 121  
roadmap 121  
standardized 121  
steps 121  
validation rules 121  
workitems 41

workitems 41

RedPoint Global Inc.

36 Washington Street, Suite 120  
Wellesley Hills, MA 02481  
+1 781 725 0250 Phone  
+1 781 235 3739 Fax  
[www.redpointglobal.com](http://www.redpointglobal.com)

Copyright © 2019 RedPoint Global Inc. This document is unpublished and the foregoing notice is affixed to protect RedPoint Global Inc. in the event of inadvertent publication.

All rights reserved. No part of this document may be reproduced in any form, including photocopying or electronic transmission, without prior written consent of RedPoint Global Inc. The information contained in this document is confidential and proprietary to RedPoint Global Inc. and may not be used or disclosed except as expressly authorized in writing by RedPoint Global Inc.

Product names mentioned in this document may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.