

Data Quality Studio on Dataverse

The Freemium & Premium apps

Table of Contents

1.	Purpose	5
2.	Audience	6
3.	Introduction to Data Quality Studio on Dataverse.....	7
4.	Data Quality Studio Freemium	8
4.1	Sign in.....	9
4.2	Duplicate Dashboard.....	9
4.3	Duplicate Detection Settings.....	10
4.4	AI Algorithm duplicate detection	11
4.5	Weight based duplicate detection	12
4.6	Duplicate check execution	14
4.7	Delivered setup examples	15
4.7.1	Set up AI Algorithm Detection setting	15
4.7.2	Set up Weight based Detection Setting	17
4.7.3	Fixed Quality Assessment on Contact first name starts with a capital.....	19
5.	Data Quality Studio Premium	20
5.1	Sign in.....	20
5.2	Set up real-time duplicate detection	20
5.3	Set up Data Patterns	22
5.3.1	Set up data pattern.....	22
5.4	Set up Web service	22
5.4.1	Set up Web service rule.....	24
5.5	Set up Configuration	25
5.5.1	Set up data pattern rule.....	25
5.5.2	Set up Web service validation rule	26
5.6	Policies and policy versions	27
5.7	Quality Assessment Projects	28
5.8	Quality Assessment Executions.....	30
5.9	Delivered set up example.....	31

5.9.1	How example is set up.....	31
6.	Frequently Asked Questions.....	33
6.1	General questions on AI.....	33
6.2	DQS on Dataverse questions.....	34

Document Information

Title	Data Quality Studio on Dataverse
Subtitle (Subject)	The Freemium & Premium apps
Solution/Suite	To-Increase Data Quality Studio
Category	Microsoft Power Platform
Author	Innovation Hub
Published Date	10/9/2023

© Copyright To-Increase B.V. All rights reserved.

The information in this document is subject to change without notice. No part of this document may be reproduced, stored or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of To-Increase B.V. To-Increase B.V. assumes no liability for any damages incurred, directly or indirectly, from any errors, omissions or discrepancies between the software and the information contained in this document.

1. Purpose

This document describes Data Quality Studio on Dataverse, both Freemium and Premium applications.

It starts with an introduction on Data quality management and moves to the Freemium application. For the Data Quality Studio on Dataverse Freemium application both AI algorithm based duplication detection and Weight based duplication detection are described.

Also the three delivered detection setups are mentioned and step-by-step explained.

Next all extra features of the Data Quality Studio on Dataverse Premium application are explained, including the Quality Assessments and their execution.

At the end some Frequently Ask Question pages are added.

Please contact To-Increase when you have questions, feedback on both of our applications or remarks on this documentation.

2. Audience

This document is primarily meant for (implementation) consultants familiar with Microsoft Power Platform, Dataverse and Data (Quality) Management.

End-users can also find useful information in this document and everyone should probably pay extra attention to chapter 4.7 as this chapter describes the setup step-by-step of the Detection Settings.

3. Introduction to Data Quality Studio on Dataverse

Data quality management is a set of practices that aim at maintaining a high quality of information. It goes all the way from the acquisition of data and the implementation of advanced data processes, to an effective distribution of data. It also requires a managerial oversight of the information you have. Effective data quality management is recognized as essential to any consistent data analysis, as the quality of data is crucial to derive actionable and - more importantly - accurate insights from your information.

Data quality refers to the assessment of the information, relative to its purpose and its ability to serve that purpose. The quality of data is defined by different, such as accuracy, completeness, consistency, or timeliness. That quality is necessary to fulfil the needs of an organization in terms of operations, planning, and decision-making.

While the digital age has been successful in prompting innovation far and wide, it has also facilitated what is referred to as the “data crisis” - low-quality data.

Today most of a company's operations and strategic decisions heavily rely on data, so the importance of quality is even higher. And indeed, low-quality data is the leading cause of failure for advanced data and technology initiatives. More generally, low-quality data can impact productivity, bottom line, and overall ROI.

An important consideration is that to uphold data integrity, it is advisable to establish guidelines that minimize the occurrence of redundant entries within the system. While model-driven apps and customer engagement applications like Dynamics 365 Sales and Dynamics 365 Customer Service come equipped with predefined duplicate detection rules for accounts and contacts, it's important to note that similar rules are not in place for other record categories. In cases where it becomes necessary to identify duplicates for different types of records, it's recommended to create a new set of duplicate detection rules tailored to those specific record types.

However, Microsoft's data duplication functionality flags records as duplicates only in case the entry fully matches that of another record. In case multiple fields are specified, the framework also does not allow to emphasize the importance of a duplicate of one of the fields in relation to the other fields.

That is why using AI algorithms and weight based thresholds will strengthen the product by ensuring a bigger potential of catching duplicates. Furthermore, providing flexibility in the configuration of the data duplication will allow for more scenarios to be supported.

Both are a part of our Freemium and Premium apps on Data Quality Studio on Dataverse.

The Premium app extends the Freemium app with data pattern rules and web service rules that can determine to validate if fields contain correct data matching the defined rules.

This ensures that the quality of the data is monitored, taking it to a higher level.

4. Data Quality Studio Freemium

The Freemium app contains two duplication detection types.
It also shows the available Premium features and will show a message when clicking on these Premium features.

The functionality you are trying to access is not available in the free version.

Enhance your data quality

Free

✓ Advanced historical duplicate detection

✓ AI recommended duplicate detection

✓ Limited data quality assessments

✓ Dual record merge

✓ Statistics dashboard

Current plan

Premium

✓ Real-time duplicate detection

✓ Advanced data governance

✓ Advanced data quality assessments

✓ Multi record merge

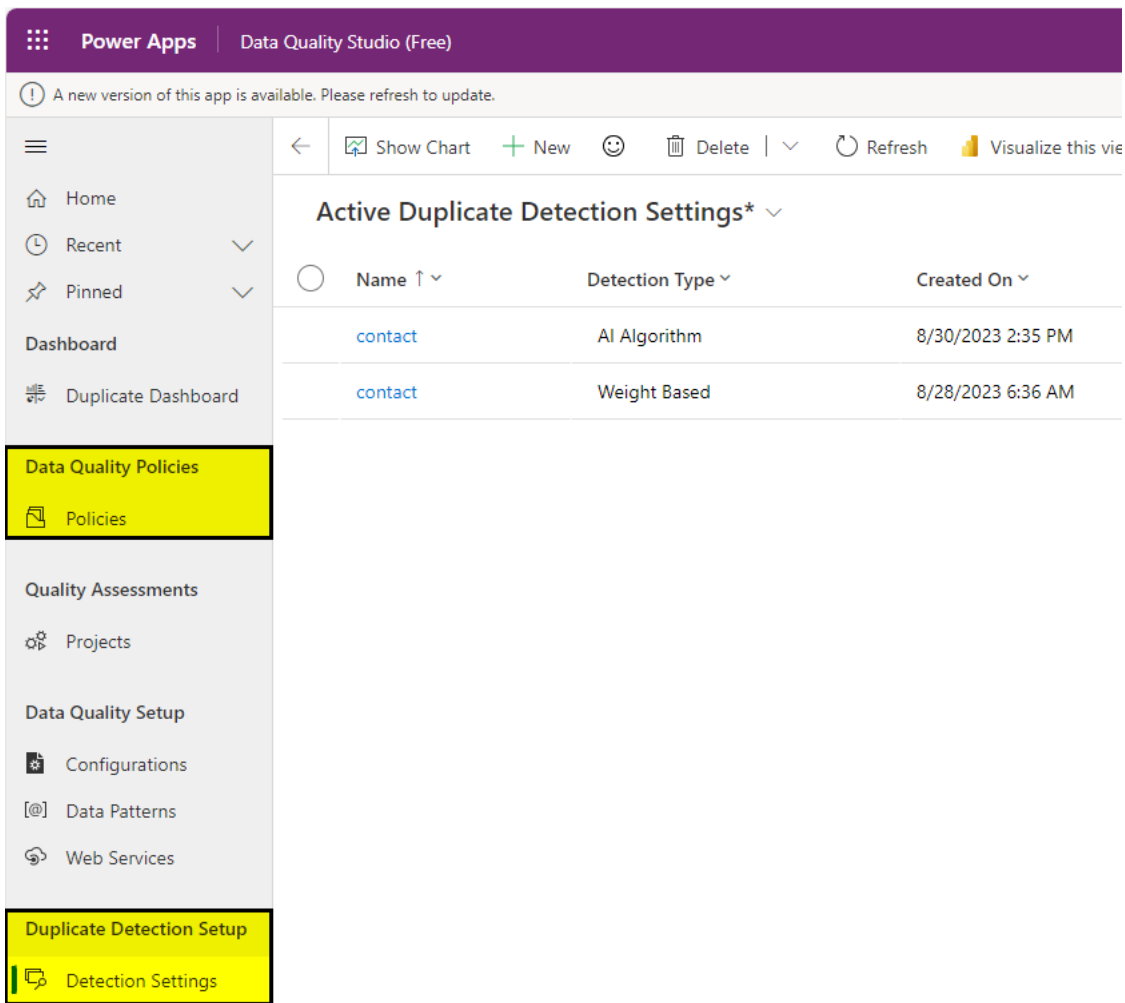
✓ Advanced interactive dashboard

✓ Support for long-running detection jobs

✓ Advanced data validation

Get started

And as extra contains a fixed Quality Assessment.



4.1 Sign in

Go to your Dynamics 365 environment and select the Freemium app

4.2 Duplicate Dashboard

The dashboard can be displayed on phone, tablet or in the Power App.



It will show three KPIs:

- The number of duplicates detected per detection type.
- The number of duplicates detected over time.
- The number of duplicates detected per table.

4.3 Duplicate Detection Settings

Duplicate detection settings determine when a record is considered a duplicate. They support two main detection types:

- **AI Algorithm:** when the user chooses the AI algorithm detection type, they will be prompted to select an AI algorithm when setting up the columns. The AI algorithm will be defaulted based on the selected column type (some work better for short strings, other work better for integers, etc.). When the AI algorithm detection type is chosen, all the columns specified in the duplicate detection setting should match for a record to be flagged as a duplicate.
- **Weight Based:** when the user chooses the Weight Based detection type, they will be prompted to define a threshold. For each of the columns they add, the user will be prompted to define a weight. Refer to the example below to understand how this duplicate detection mechanism works.

Active Duplicate Detection Settings*

 Name 	Detection Type 
contact	Weight Based
contact	AI Algorithm

NOTE:

Due to current limitations, AI Algorithms and Weight Based detections have a maximum runtime of 120 seconds in both Freemium and Premium.

It is unfortunately not possible to visualize the occurring error for the user.

In Premium it is possible to run both via Azure Functions without a timeout.

4.4 AI Algorithm duplicate detection

The AI algorithm will be defaulted* based on the selected column type (some work better for short strings, other work better for integers, etc.).

The used AI algorithms are:

- Levenshtein, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other. See https://en.wikipedia.org/wiki/Levenshtein_distance
- JaroWinkler, the Jaro–Winkler similarity is a string metric measuring an edit distance between two sequences, The higher the Jaro–Winkler distance for two strings is, the less similar the strings are. See https://en.wikipedia.org/wiki/Jaro%E2%80%93Winkler_distance
- CosineSimilarity, an example to explain this algorithm, in information retrieval and text mining, each word is assigned a different coordinate and a document is represented by the vector of the numbers of occurrences of each word in the document. Cosine similarity then gives a useful measure of how similar two documents are likely to be, in terms of their subject matter, and independently of the length of the documents. See https://en.wikipedia.org/wiki/Cosine_similarity
- JaccardSimilarity, the Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. See https://en.wikipedia.org/wiki/Jaccard_index
- Metaphone, Metaphone 3 is used and improves phonetic encoding of words in the English language, non-English words familiar to Americans, and first names and family names commonly found in the United States. See <https://en.wikipedia.org/wiki/Metaphone>

*As mentioned the AI algorithm is defaulted based on the selected column type. When the default AI algorithm is changed afterwards this will have an effect on the operation of duplication detection, this duplication detection could be slower or will lead to worse outcomes.

contact - Saved

Duplicate Detection Setting

General Related ▾

Name *	Contact (contact) ▾
🔒 Detection Type *	AI Algorithm
○ Column ↑ ▾	
	firstname

firstname - Saved

Column

General Related ▾

🔒 Table	contact
🔒 Detection Type	AI Algorithm
Name *	First Name (firstname) ▾
🔒 Phonetic Hash Column	tidqs_metafirstname_hash
AI Algorithm	Metaphone

 Erik Jansen - Saved

Contact - Contact ▾

Summary Details Duplicate contacts (AI Based) Related ▾

 Show Chart  New Duplicate Entry  Add Existing Duplicate...  Refresh  Flow ▾  Run Report ▾  Excel Templates ▾  Export Duplicate Entries | ▾

Duplicate Entry Associated View ▾

○ Name ↑ ▾	Weight ▾	Matching Columns ▾	Record Url ▾	Created On ▾
contact			https://operations-d365-dw-tst.crm4.dyna...	9/5/2023 1:18 PM
contact			https://operations-d365-dw-tst.crm4.dyna...	9/5/2023 1:18 PM
contact			https://operations-d365-dw-tst.crm4.dyna...	9/5/2023 1:18 PM

4.5 Weight based duplicate detection

Suppose the following duplicate detection setting has been created:

Name	*	Contact (contact)
Detection Type	*	Weight Based
Threshold		70

☐	Column ↑ ↓	Weight ↓
	accountid	20
	firstname	30
	lastname	50

In the system, there exists an account with accountid = 1, firstname = "John" and lastname = "Doe".

Now, the user creates a new record with the details accountid=1, firstname = "John", lastname = "Smit". The system will then find that the accountid and firstname match an existing record in the system, so it will sum up the respective weights ($20 + 30 = 50$). Since this is below the specified threshold of 70, the system **will not** flag the newly created contact as a duplicate.

The user then creates a new record with details accountid=2, firstname = "John", lastname = "Doe". The system finds that the firstname and lastname match an existing record in the system, so it will sum up the respective weights ($30 + 50 = 80$). Since this is $\geq$ the specified threshold of 70, the system **will** flag the newly created contact as a duplicate.

Be aware that the total weight must be 100, when this total is not met, a message will pop-up.

⊗ **Name: Consolidated Weight is less than 100**

contact - Saved

Duplicate Detection Setting

General Related ▾

Name *	Contact (contact) ▾
Detection Type *	Weight Based
Threshold	59

Column ↑ ▾	Weight ▾
address1_line1	10
firstname	39
lastname	51

🔔 Duplicate Detection Completed

EJ Erik Jansen - Saved
Contact · Contact ▾

Summary Details **Duplicate contacts (Weight Based)** Related ▾

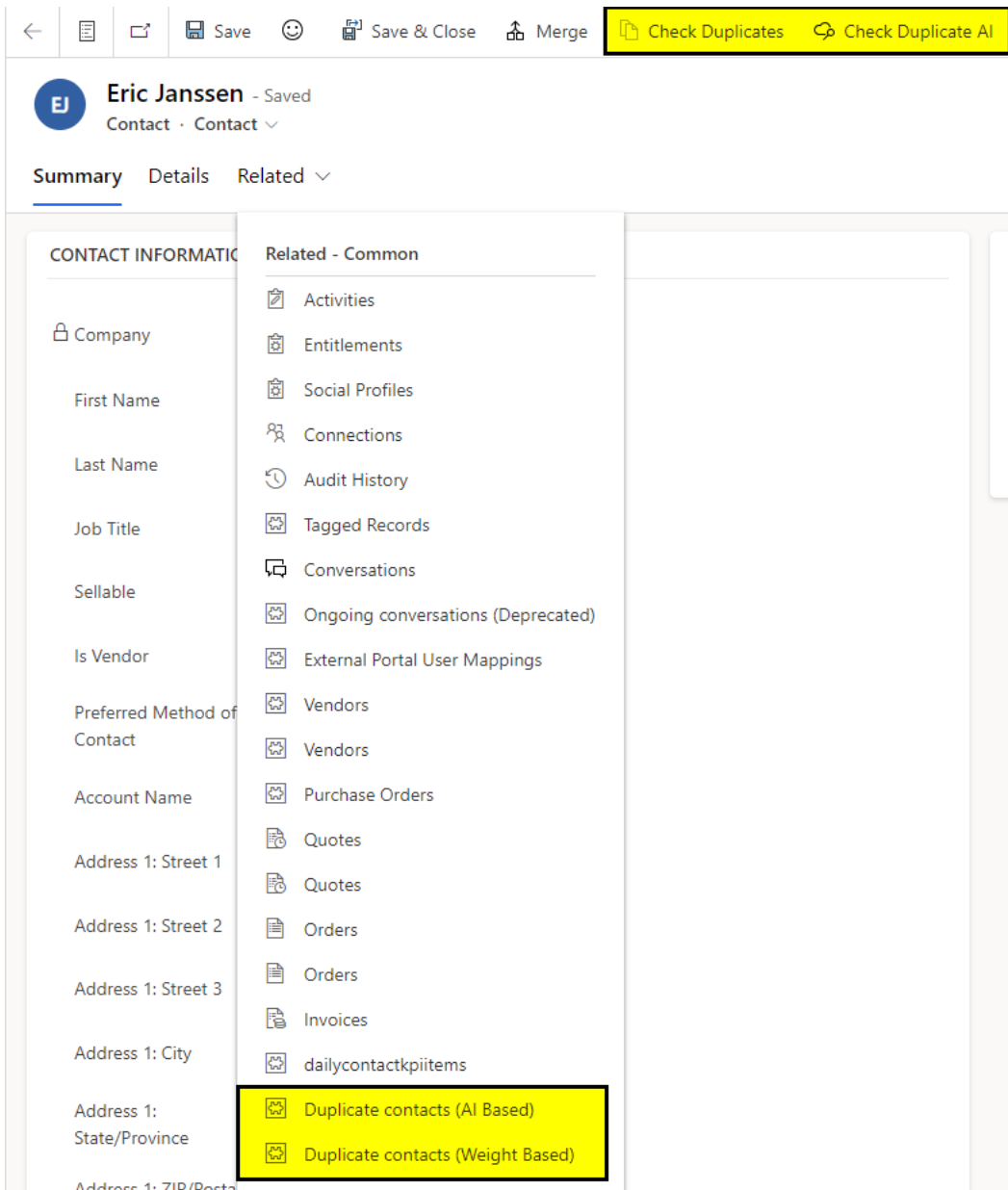
📊 Show Chart + New Duplicate Entry 📄 Add Existing Duplicate... 🔄 Refresh ⚙️ Flow ▾ 📄 Run Report ▾ 📄 Excel Templates ▾ 📄 Export Duplicate Entries | ▾

Duplicate Entry Associated View ▾

Name ↑ ▾	Weight ▾	Matching Columns ▾	Record Url ▾	Created On ▾
contact	90	lastname.firstname	https://operations-d365-dw-tst.crm4.dyna...	9/7/2023 11:45 AM
contact	61	lastname.address1_line1	https://operations-d365-dw-tst.crm4.dyna...	9/7/2023 11:45 AM

4.6 Duplicate check execution

Duplicate checks can be executed on any record in the system by clicking on the Check Duplicates (Weight Based) and Check Duplicate AI (AI algorithm) buttons in the ribbon bar. If duplicates are found, they will be listed in the related section of the record. From here, users can merge duplicates into one master record. Please note that this functionality is (for now) only supported for accounts, contacts, and leads.



4.7 Delivered setup examples

Herewith 2 examples which a user can set up, one AI algorithm and one Weight-based duplication detection, which both can be altered to the user needs.

Secondly a fixed Quality Assessment will be delivered based on First capital check for first name of a contact, to show the capabilities of the Premium application.

4.7.1 Set up AI Algorithm Detection setting

1. Go to Duplicate Detection Setup

2. Go to Detection Settings
3. Click on <+New>
4. In Name search for "Contact (contact)" via lookup
5. Click on "Contact (contact)" to select
6. Select for Detection Type: AI Algorithm
7. Click on <Save>

contact - Saved
Duplicate Detection Setting

General Related ▾

Name *	Contact (contact) ▾
🔒 Detection Type *	AI Algorithm

8. Click in Column part on <+New Column>
9. In Name search for "First Name (first name)" via lookup
10. Click on "First Name (first name)" to select
11. Automatically the default AI Algorithm will be filled
12. Click on <Save>

firstname - Saved
Column

General Related ▾

🔒 Table	contact
🔒 Detection Type	AI Algorithm
Name *	First Name (firstname) ▾
🔒 Phonetic Hash Column	tidqs_metafirstname_hash
AI Algorithm	Metaphone
Owner *	

13. Via return arrow you can return to the hierarchical structure
14. Set up will look like:

contact - Saved
Duplicate Detection Setting

General Related ▾

Name *	Contact (contact) ▾
🔒 Detection Type *	AI Algorithm

+ New Column

○ Column ↑ ▾

firstname

15. Now the AI algorithm duplication detection can be used by clicking on <Check Duplicate AI>

4.7.2 Set up Weight based Detection Setting

1. Go to Duplicate Detection Setup
2. Go to Detection Settings
3. Click on <+New>
4. In Name search for "Contact (contact)" via lookup
5. Click on "Contact (contact)" to select
6. Select for Detection Type: Weight Based
7. Threshold: 62

New Duplicate Detection Setting - Unsaved

General

Name	*	Contact (contact) ▼
🔒 Detection Type	*	Weight Based
Threshold		62

8. Click on <Save>
9. Click in Column part on <+New Column>
10. Add "address1-line1" with weight 10

New Column - Unsaved

General

🔒 Table		contact
🔒 Detection Type		Weight Based
Name	*	Address 1: Address Type (address1_addr... ▼
Weight		10
Owner	*	

11. Click on <Save>
12. Click on return arrow (don't put any attention to the error)
13. Click in Column part on <+New Column>
14. Add "firstname" with weight 39

New Column - Unsaved

General

Table	contact
Detection Type	Weight Based
Name	* First Name (firstname) ✓
Weight	39
Owner	* [REDACTED]

15. Click on <Save>
16. Click on return arrow (don't put any attention to the error)
17. Click in Column part on <+New Column>
18. Add "lastname" with weight 51

New Column - Unsaved

General

Table	contact
Detection Type	Weight Based
Name	* Last Name (lastname) ✓
Weight	51
Owner	* [REDACTED]

19. Click on <Save>
20. Via return arrow you can return to the hierarchical structure
21. Set up will look like:

contact - Saved
Duplicate Detection Setting

General Related ▾

Name	*	Contact (contact) ▾
🔒 Detection Type	*	Weight Based
Threshold		62

○	Column ↑ ▾	Weight ▾
	address1_addresstypecode	10
	firstname	39
	lastname	51

22. Now the Weight-based duplication detection can be used by clicking on <Check Duplicates>

4.7.3 Fixed Quality Assessment on Contact first name starts with a capital

An Active Quality Assessment Project is set to use.

On the background this contains:

- A data pattern that checks if a word starts with a capital
- A data pattern configuration on Table "Contact (contact)" and Column "First Name (first name)"
- A policy with an active version of the data pattern configuration
- A project in Quality Assessments "contact first name assessments" with description "Assess whether all contacts' first names start with a capital"

The Quality Assessment can be executed by clicking on the <Execute> button.

This will check if all First names of Contacts will start with a Capital.

5. Data Quality Studio Premium

The Premium app also contains the Freemium features, and extends the Freemium app with real-time duplication detection, data patterns, web services validation and quality assessments.

5.1 Sign in

Go to your Dynamics 365 environment and select the Premium app

5.2 Set up real-time duplicate detection

By using Microsoft Power Platform standard duplication detection rules we can use our duplication detection features real-time.

Please note that this functionality only works for duplicate detection settings of type AI Algorithm and with the Metaphone algorithm selected.

Suppose the following duplicate detection setting has been created:

The screenshot shows the configuration for a duplicate detection setting named 'firstname'. The 'General' tab is selected. The configuration is as follows:

Property	Value
Table	contact
Detection Type	AI Algorithm
Name	* First Name (firstname)
Phonetic Hash Column	tidqs_metafirstname_hash
AI Algorithm	Metaphone

Then navigate to environment's Duplicate Detection rules in the Power Platform Admin Center, see for more information [Set up duplicate detection rules to keep your data clean - Power Platform | Microsoft Learn](#)

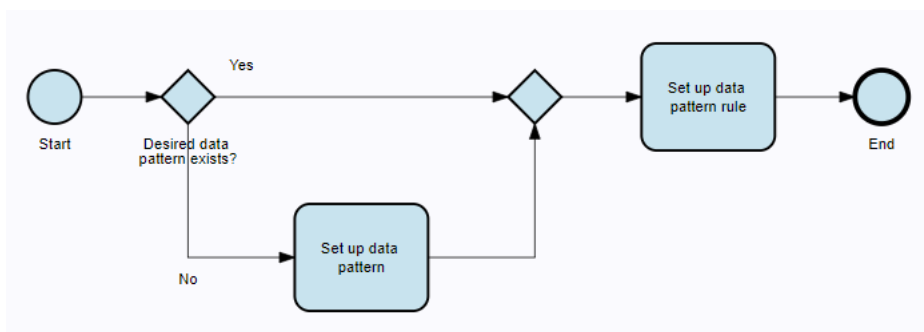
Here a new rule can be created with the following properties:

- Base record type : Table selected in the duplicate detection setting. Contact in this example.
- Matching record type : Table selected in the duplicate detection setting. Contact in this example.
- Field : MetaHash <column selected in the duplicate detection setting>. In this example MetaHash firstname.
- Criteria : Exact Match

5.3 Set up Data Patterns

Data patterns are using regular expressions, a regular expression, sometimes referred to as rational expression is a sequence of characters that specifies a match pattern in text. Usually such patterns are used by string-searching algorithms for "find" or "find and replace" operations on strings, or for input validation. Regular expression techniques are developed in theoretical computer science and formal language theory, see https://en.wikipedia.org/wiki/Regular_expression

The flow explains how to set up a data pattern and – rule.



5.3.1 Set up data pattern

First Characters Capital - Saved

Data Pattern

General Related ▾

Name	* First Characters Capital
Data Pattern	^[A-Z].*\$
Error Message	First letter should be capital

Test Cases

Input ↑ ▾	Indicator ▾	Expected Result ▾	Actual Result ▾
erik	✓	Fail	Fail
Erik	✓	Pass	Pass

5.4 Set up Web service

On a Dataverse table, you can use web service rules for these purposes:

1. Validate data
2. Validate and set data

Before you can set up a web service rule, set up the web service to be used.

Validate data

Validate if a new or changed field value in a Dataverse table matches a value in another internal or external source.

Example:

Use a web service rule to validate a ZIP code and house number combination.

For request type 'POST', the request JSON is, for example:

```
{  
  
  "code":3905 NR,  
  
  "houzenumber":2  
  
}
```

Validate and set data

Validate if a new or changed field value in a Dataverse table matches a value in another internal or external source.

Get a field value from another internal or external source and set this value in a Dataverse table.

Example:

Use a web service rule to first validate if a ZIP code and house number combination is valid. If valid, based on the ZIP code, set address data like street name, city name, and province name.

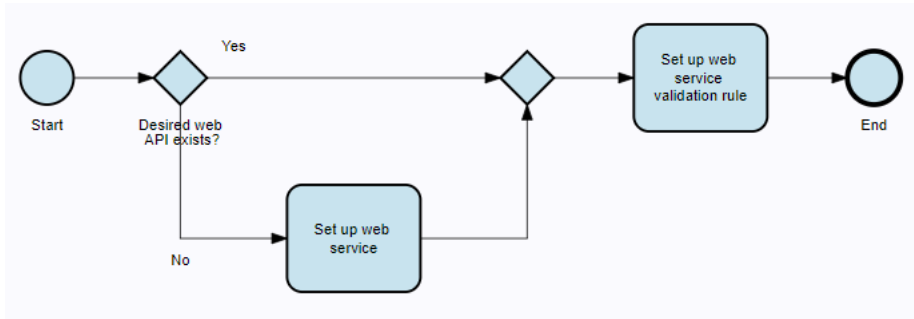
For request type 'POST', the response JSON is, for example, when successful:

```
{  
  
  "response":true,  
  
  "street":Kazemat,  
  
  "city":Veenendaal,
```

"province":Utrecht

}

This flow explains how to set up web service and – validation rules.



5.4.1 Set up Web service rule

You can find the required web service settings, to be used for the web service rule, in the technical documentation as provided by the web service provider.

The web service supports GET and POST API commands.

Language Detection API - Saved
Web Service

General Related

Name	Language Detection API
Url	https://prod-48.westeurope.logic.azure.com:443/workflows/38ed157c6ad2444b5ce6c962d32d77/triggers/manual?api-version=2016-06-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=MF6z_ndmL4dy8DXp2D5YevU_Jfun5gM0uIMZLOE
Authentication Type	Request Header
Parameter Name	api-key
Parameter Value	ok
Request Type	GET POST
Error Message	Language should be english
Is Paid Web Service?	No
Owner	

Test Cases

Test Cases	Input Json	Expected Output	Actual Output
+ New Web Service Test + Add Existing Web Serv... Refresh Flow			

Or

Melissa Address Search - Saved
Web Service

General Related ▾

Name	* Melissa Address Search
Url	* https://address.melissadata.net/v3/WEB/GlobalAddress.do?GlobalAddressId=YBNDFOH9TECAQyEWX5H3**
Authentication Type	* Request Header
Parameter Name	* dummy
Parameter Value	* dummy
Request Type	GET POST
Error Message	Address Not Found
Is Paid Web Service?	* No
Owner	* [Avatar]

Test Cases

Input Item	Expected Output	Actual Output

+ New Web Service Test
 Add Existing Web Serv...
 Refresh
 Flow ▾
 :

A flag can be set that indicates whether the web service is a paid one. This flag is used while executing quality assessment projects to limit costs.

Note: With a web service rule, you can only set field values in Dataverse if first a validation is successfully done.

5.5 Set up Configuration

Based on the set data patterns and web services configurations (rules) can be created.

Active Configurations* ▾

Table ↑ ▾	Validation Type ▾	Column ▾	Created On ▾
account	Data Pattern	name	9/8/2023 6:21 AM
contact	Web Api	address1_line1	9/8/2023 8:37 AM
incident	Web Api	title	9/8/2023 7:14 AM

5.5.1 Set up data pattern rule

General configuration:

account - Unsaved Configuration

General Conditions Related ▾

Table	*	Account (account) ▾
Column	*	Account Name (name) ▾
Validation Type	*	Data Pattern
Pattern	*	[@] First Characters Capital
Validate Business Unit		DEMFB

When the field “Validate in business unit” is populated the configuration will only be triggered on records created/updated in the specified business unit.

When the field “Validate in business unit” is not populated, the configuration will be triggered on records created/updated in all business units.

5.5.2 Set up Web service validation rule

Total overview

contact - Saved
Configuration

General Conditions Related ▾

Column	*	Address 1: Street 1 (address1_line1) ▾
Validation Type	*	Web Api
Web API	*	Melissa Address Search
Validate Business Unit		---
Owner	*	ME

Request Parameters

Name ▾	Parameter Type ▾	Bound ▾	Value ▾
a1	Request	Yes	address1_line1
ctry	Request	No	NL
format	Request	No	json
postal	Request	Yes	address1_postalcode

1 - 4 of 4 Page 1

Response Parameters

Name ▾	Parameter Type ▾	Bound ▾	Value ▾	Success Values ▾
Records.AddressLine2	Response	Yes	address1_line2	
Records.AdministrativeArea	Response	Yes	address1_stateorprovince	
Records.Results	Response	No		AC02.AC03.AC16.AE11.AV13.GS02
Records.SubAdministrativeArea	Response	Yes	address1_city	

1 - 4 of 4 Page 1

General configuration:

contact - Saved Configuration

General Conditions Related ▾

Column	*	Address 1: Street 1 (address1_line1) ▾
Validation Type	*	Web Api
Web API	*	 Melissa Address Search
Validate Business Unit		---

When the field “Validate in business unit” is populated the configuration will only be triggered on records created/updated in the specified business unit.

When the field “Validate in business unit” is not populated, the configuration will be triggered on records created/updated in all business units.

Request parameters:

Request Parameters			
☐ Name ↑ ▾	Parameter Type ▾	Bound ▾	Value ▾
a1	Request	Yes	address1_line1
ctry	Request	No	NL
format	Request	No	json
postal	Request	Yes	address1_postalcode

Response parameters:

Response Parameters				
☐ Name ↑ ▾	Parameter Type ▾	Bound ▾	Value ▾	Success Values ▾
Records.AddressLine2	Response	Yes	address1_line2	
Records.AdministrativeArea	Response	Yes	address1_stateorprovince	
Records.Results	Response	No		AC02,AC03,AC16,AE11,AV13,GS02
Records.SubAdministrativeArea	Response	Yes	address1_city	

5.6 Policies and policy versions

Policies allow for grouping of configurations and can contain 0 to N versions, which adhere to the following rules. Please note that some of the mentioned processes are implemented asynchronously, which means it might take a few seconds for the result to show up in the UI.

- There can only be one active version at a time.
- Each version is decorated with a unique number, the Version Number, which automatically increments with each version creation.
- When creating a new policy version, the user can specify a Source version. If this is filled in, all the configurations should be copied from the source version onto the newly created version. The new policy version automatically becomes active.
- If a policy version gets deactivated, the configurations included will be deactivated as well. The same goes for policy version activations.

Example on First name capital:

First Name Capital - Saved

Policy

General Related ▾

Name

* First Name Capital

Versions

Version number ↑ ▾

Name ▾

Status ▾

1

V1

Active

V1 - Saved

Policy Version

General Related ▾

Name

+

V1

Source version

Owner

*

oJ

Validation rules

Table

↑

▼

Column

▼

Validation Type

▼

Status

▼

contact

firstname

Data Pattern

Active

5.7 Quality Assessment Projects

Configurations will be triggered during data creation or real-time updates. This allows for data quality enforcement during manual entry and data import scenarios. However, this approach does not validate already existing data in the system. That's where quality assessment projects come into play. With the help of quality assessment projects, users can run configurations on existing data in their system and manually correct faulty records. Quality assessment projects adhere to the following rules:

- They can contain 0 to N policies. During quality assessment executions, all configurations specified in the policies' active versions will be triggered.
- The following parameters limit the number of records being queried. This is important because Dataverse has a limit in place that restricts us from running jobs that last longer than 120 seconds.
 - **Records created after** only queries records that have been created after the specified datetime. If no datetime is specified, the system's minimum datetime will be used.
 - **Records created before** only queries records that have been created before the specified datetime. If no datetime is specified, the system's maximum datetime will be used.
 - **Exclude Paid Web Services** makes sure that paid webservices will not be executed. Whether a webservice is a paid one is controlled by a flag on the webservice itself.
- Quality assessment projects can be executed by a click on the Execute button in the ribbon bar. This is an asynchronous process, so users can navigate away from the screen. If they remain in the screen itself, they will see a notification once the project finished executing.

Example on First name capital:

The screenshot displays the configuration interface for a Quality Assessment Project named 'First name capital detection'. The ribbon bar at the top includes buttons for 'Save', 'Execute' (highlighted in yellow), 'Save & Close', 'Merge', 'Check Duplicates', 'Check Duplicate All', 'New', 'Deactivate', 'Delete', 'Refresh', and 'Check Access'. The project is currently 'Saved'.

The 'General' tab is active, showing the following configuration details:

- Name:** First name capital detection
- Description:** ---
- Records Created After:** 8/1/2023
- Records Created Before:** ---
- Exclude Paid Web Services:** No
- Owner:** [User Avatar]

Below the configuration details is a 'Policies' section with a table listing the active policies for this project:

Name	Status
First Name Capital	Active

Results					+ New Quality As
☐ Error message ¹ ▾	Table ▾	Column ▾	Review status ▾	Record URL ▾	
First letter should be capital For name	account	name	Pending	https://operations-d365-1	
First letter should be capital For name	account	name	Pending	https://operations-d365-1	
First letter should be capital For name	account	name	Pending	https://operations-d365-1	
First letter should be capital For name	account	name	Pending	https://operations-d365-1	
First letter should be capital For name	account	name	Pending	https://operations-d365-1	
First letter should be capital For name	account	name	Pending	https://operations-d365-1	

5.9 Delivered set up example

DQS on Dataverse Premium application will be delivered with one Quality Assessment based on First capital check for last name of a contact.

5.9.1 How example is set up

- 1. Set up a Data Pattern**
- Go to Data Pattern in Data Quality Setup
- Click on <+New>
- Name: Start with a capital
- Data Pattern: `^[A-Z].*$`
- Error Message: Should start with a capital
- Click on <Save>
- In addition Test cases can be added
- Click on <+ New Test Case>
- Add: Lastname
- Add: lastname
- System will show expected and actual result
- 13. Set up a Configuration**
- Go to Configurations in Data Quality Setup
- Click on <+New>
- In Table search for "Contact (contact)" via lookup
- Click on "Contact (contact)" to select
- In Column search for "Last Name (lastname)" via lookup
- Click on "Last Name (lastname)" to select
- Validation Type: Data Pattern
- Pattern: Start with a capital
- Validate Business Unit: is optional
- 23. Set up a Data Quality Policy**
- Go to Policies
- Click on <+New>

26. Name: Sales & marketing – Sample
27. Click on <Save>
28. Add a Version
29. Click on <+ New Policy Version>
30. Version number will be generated automatically
31. Name: Initial version
32. Add Validation Rule
33. Click on <+ New Configuration>
34. Add Contact configuration
35. Click on <Save>
- 36. Set up a Quality Assessment Project**
37. Go to Projects
38. Add new Quality Assessment Project via <+New>
39. Name: Contacts last name assessment
40. Description: Assess whether all contacts' last names start with a capital
41. Optional Records Created After date can be mentioned
42. Optional Records Created Before date can be mentioned
43. Exclude Paid Web Services can be set to YES or NO
44. Add a Existing Policy via <Add Existing Policy>
45. Add Sales & Marketing – Sample
46. Click on <Save>
47. Use <Execute> button to use the Quality Assessment
48. Results will be shown in Results part

6. Frequently Asked Questions

6.1 General questions on AI

Q1: What is artificial intelligence?

A: Artificial intelligence is a branch of computer science concerned with creating machines that can think and make decisions independently of human intervention. Those AI programs can do complex tasks that were previously only able to be done by humans. Some AI programs can complete simple tasks, some more complex. Some can take in data to learn and improve, completely without the touch of a human developer.

Q2: What is an AI algorithm?

A: The definition of an algorithm is “a set of instructions to be followed in calculations or other operations.” This applies to both mathematics and computer science. So, at the essential level, an AI algorithm is the programming that tells the computer how to learn to operate on its own.

An AI algorithm is much more complex than what most people learn about in algebra, of course. A complex set of rules drive AI programs, determining their steps and their ability to learn. Without an algorithm, AI wouldn't exist.

Q3: How do AI algorithms work?

A: While a general algorithm can be simple, AI algorithms are by nature more complex. AI algorithms work by taking in training data that helps the algorithm to learn. How that data is acquired and is labeled marks the key difference between different types of AI algorithms.

At the core level, an AI algorithm takes in training data (labeled or unlabeled, supplied by developers, or acquired by the program itself) and uses that information to learn and grow. Then it completes its tasks, using the training data as a basis. Some types of AI algorithms can be taught to learn on their own and take in new data to change and refine their process. Others will need the intervention of a programmer in order to streamline.

Q4: What AI algorithms do we normally use?

A: We mainly use:

- Data entry and classification
- Advanced or predictive analytics
- Search engines (Google, Yahoo, Bing, etc.)
- Digital assistants (Siri, Alexa, etc.)
- Robotics (assembly machines, self-driving cars, etc.)

Q5: What kind of types AI algorithms do we have?

A: There are three major categories of AI algorithms: supervised learning, unsupervised learning, and reinforcement learning. The key differences between these algorithms are in how they're trained, and how they function.

Under those categories, there are dozens of different algorithms.

- **Supervised learning**, these work by taking in clearly-labeled data while being trained and using that to learn and grow. It uses the labeled data to predict outcomes for other data. The name “supervised learning” comes from the comparison of a student learning in the presence of a teacher or expert. Building a supervised learning algorithm that actually works takes a team of dedicated experts to evaluate and review the results, not to mention data scientists to test the models the algorithm creates to ensure their accuracy against the original data and catch any errors from the AI.
- **Unsupervised learning** algorithms mean, in comparison to supervised learning. Unsupervised learning algorithms are given data that isn't labeled. Unsupervised learning algorithms use that unlabeled data to create models and evaluate the relationships between different data points in order to give more insight to the data.
- **Reinforcement learning** algorithms, which learn by taking in feedback from the result of its action. This is typically in the form of a “reward”. A reinforcement algorithm is usually composed of two major parts: an agent that performs an action, and the environment in which the action is performed. The cycle begins when the environment sends a “state” signal to the agent. That queues the agent to perform a specific action within the environment. Once the action is performed, the environment sends a “reward” signal to the agent, informing it on what happened, so the agent can update and evaluate its last action. Then, with that new information, it can take the action again. That cycle repeats until the environment sends a termination signal. There are two types of reinforcement the algorithm can use: either a positive reward, or a negative one.

6.2 DQS on Dataverse questions

Q1: Is the data in my Dataverse used to train other AI models?

A: No, our applications run only in your Dataverse environment.

Q2: You use 5 different AI algorithms, why?

A: The used AI algorithms are best suited; they create the best outcome.

Q3: Why are the used AI algorithms defaulted?

A: The used AI algorithm is based on the selected column to perform better output.

Q4: What happens when I change the defaulted AI algorithm to another?

A: When the default AI algorithm is changed afterwards this will influence the operation of duplication detection, this duplication detection could be slower or will lead to worse outcomes.

Q5: I don't see any outcome produced?

A: Requests have a limit of 120 seconds, when the request does take longer, the request will be stopped after 120 seconds. This is a timeout limitation made by Microsoft.