Veeva Network

Veeva Network 22R2.0.1 Release Notes

August 2022

Contents

About these Release Notes **8**

Subscribe to release notifications 8

Browser requirements **8**

Release Note updates **8**

What's new **9**

Hierarchy Explorer **12**

Rollup counts for HCOs 12

Counts in Explore tabs 13

Smart filtering 14

Level column 18

Fullscreen 18

Health system search 18

Clear filters 19

Hierarchy levels 19

Tooltips 20

Network hashtags **20**

New hashtags 20

Predefined hashtags 20

Network hashtags in Veeva CRM 21

About Network hashtags 22

Example 1 - Searching for HCOs 22

Example 2 - Searching for HCPs 23

Example 3 - Viewing Account List 25

Hashtag support in Veeva CRM 26

Veeva CRM configuration 26
Network configuration 27
Pushing hashtags to Veeva CRM 27
Data lineage 29
Current source column 29
Legend 30
Show or Hide options 30
View sources 31
Sub-object details 32
Addresses 32
Parent HCO 33
Licenses 33
Inbox 34
Save as inbox view 34
Reports 35
Merged HCO and HCP reports 35
Merged HCOs query 35
Merged HCPs query 35
Save the reports 36
Reporting on hierarchies 37
Benefits 38
Enable the flat hierarchy table 39
Supported hierarchy paths 40
View the hierarchy table 41
Flattened hierarchy report examples 42

Customizations 49

Data quality 49

Data validation rules 49

Validating suspect match data 49

How it works 49

Enable the feature 50

Job validation rules 51

Enable job validation rules for source subscriptions 52

Enable job validation rules for Data Updater jobs 52

Job validation rules page 53

Creating validation rules 53

Edit rules 56

Copy rules 56

Delete rules 57

Triggered validation rules 58

Job errors 58

Predefined rules 61

Bulk updates 67

Logging 67

Exporting configurations 68

Data cleansing 70

Standardize field values 70

Data cleansing process 71

Create the data cleansing dictionary 71

Upload the file to Network 75

Apply the data cleansing function 76
Replacing special characters 78
Nulling existing values 78
Trimming white space 78
Delete the data cleansing dictionary 79
Data cleansing errors 79
File explorer 80
Smart tables 80
View smart tables 80
Add fields to smart tables 82
Add a new object 86
Download smart tables 89
View original file 90
Data model 90
Country support (22R2.0) 90
Localization 90
Data privacy opt out (22R2.0) 91
Opted-out countries 91
Network address inheritance 91
Country support (22R1.1) 91
Latin America 91
Asia Pacific 92
Data privacy opt out (22R1.1) 92
Opted-out countries 93
Cluster management 93

Primary address 94
Primary recalculation logic 94
Using standard logic 95
Custom logic 96
Supported fields 99
Logs....100
Field configurations....100
Data sources 101
DCR routing for third party systems....101
DCR processing 101
Allow change requests on locally managed fields....102
Parent HCO field....102
OpenData subscriptions 103
Export job error logs 103
Exporting the log file 104
Error log details 104
Source subscriptions 105
Transforming inbound data 105
Examples of data transformations 105
How transformation queries work 105
Supported source files....106
Example data transformation scenario 107
Upload source files as custom tables 107
Write and test the transformation query....109
Create a transformation query....110

Saved transformation queries 113
Configure source subscriptions to load files 114
Job details 119
Job errors 121
Exporting configurations 121
Source file column headers 122
Formatting case for column headers 122
Considerations for attributes 122
Integrations 122
SAP Concur authentication 122
Migrating existing Concur credentials 123
Creating Concur credentials 123
Audit 126
API 127
Version Update 127
API updates for hashtags 127
Search and Retrieve API 127
Match rule collections 129
Match API 129
Creating match rule collections 131
Edit country groups 137
Delete country groups 138

About these Release Notes

These Release Notes describe all features that are included in Veeva Network 22R2.0.

SUBSCRIBE TO RELEASE NOTIFICATIONS

You can receive email notifications about upcoming software releases and the supporting documentation:

- **Software releases and maintenance** - Go to trust.veeva.com. At the top of the page, click **Subscribe to Veeva Trust Site** and subscribe to the Veeva Network component.
- **Release Notes and Data Governance documents** - PDF files are posted on the Veeva Support website. To be notified when new documents are published, click the **Follow** button on that page or the Announcements section in the Network Community.

For more information, see About Network Releases in the *Veeva Network Online Help*.

Browser requirements

Veeva Network is tested and supported on the latest version of these browsers:

- Google Chrome™
- Apple® Safari®
- Microsoft® Edge

Veeva Network is not supported on mobile devices.

Release Note updates

The following item has been added to this release since the Sandbox Release Notes were published:

- **Merged HCO and HCP reports** – These legacy reports will be discontinued in a future release. For best results, begin using the provided SQL queries to replace these saved reports.

The Match Rule Collections feature (released in 22R1.1) was omitted from the Early Release Notes by mistake. It is included in the Sandbox Release Notes and the Production Release Notes.

All material in the Release Notes should be reviewed to ensure that updates to existing topics are noted.

What's new

The following key enhancements comprise the Veeva Network 22R2.0 major release.

			ST	DS	DM	AD
Hierarchy Explorer widget						
Rollup counts for HCOs	The widget now displays rollup counts for HCOs.	22R2.0	●	●	●	●
Smart filtering	Apply filters on the first level of a hierarchy so you can view the hospital hierarchies in the health system.	22R2.0	●	●	●	●
Level column	This column displays the level the HCO is at within the health system.	22R2.0	●	●	●	●
Fullscreen view	Hierarchy Explorer can be expanded to full browser size.	22R2.0	●	●	●	●
Health system search	View the list of available health systems in your Network instance.	22R1.1	●	●	●	●
Clear filters	The Active Filters dialog includes the option to clear all of the selected filters.	22R1.1	●	●	●	●
Hierarchy levels	The Hierarchy Level column now displays in the exported file.	22R1.1	●	●	●	●
Tooltips	Tooltips display when you hover over links	22R1.1	●	●	●	●
Network hashtags						
New hashtags	Several predefined hashtags have been added for the HCP and HCO objects for all countries.	22R2.0	●	●	●	●
Hashtags in CRM	Network hashtags can be configured to display in Veeva CRM.	22R2.0				●
Data Lineage						
Current Source column	A new column identifies the current source for fields on the record.	22R1.1	●	●	●	●
Sub-object details	Addresses, licenses, and parent HCOs now contain a summary so you can easily identify the correct sub-object.	22R1.1	●	●	●	●
Inbox						
Save inbox views	Data Stewards can now quickly save changes to the current view as a new view.	22R2.0	●	●	●	●

			ST	DS	DM	AD
Reports						
Merged HCO and HCP reports	SQL queries are provided to replace these legacy reports.	22R2.0	●	●	●	●
Flattened hierarchies	A new table called flat_hierarchy is added to Network reports to help you report on relationship hierarchies.	22R1.1	●	●	●	●
Data Quality						
Validation rules	Data validation rules have been renamed to profile validation rules.	22R2.0		●	●	●
Validation rules for suspect match tasks	Administrators can configure profile validation rules to automatically run after suspect match tasks are processed.	22R2.0		●	●	●
Job validation rules	Validation rules can be applied to data loading jobs to prevent critical value changes to large numbers of records.	22R2.0			●	●
Data cleansing	Administrators and data managers can now define rules to cleanse and standardize data in fields.	22R1.1			●	●
File Explorer						
Smart tables	Open, view, and augment .csv files directly from the File Explorer.	22R1.1	●	●	●	●
Data Model						
Supported countries (22R2)	Veeva OpenData data models have been added for Egypt and Jordan.	22R2.0			●	●
Data privacy opt-out (22R2)	Veeva OpenData now manages HCP opt outs for Egypt and Jordan.	22R2.0			●	●
Network address inheritance	China can now be added to Network address inheritance configurations with other countries.	22R2.0			●	●
Supported countries (22R1.1)	Veeva OpenData data models have been added for countries in Latin America and Asia Pacific.	22R1.1			●	●
Data privacy opt-out (22R1.1)	Veeva OpenData now manages HCP opt outs for several additional countries in the Asia Pacific region.	22R1.1			●	●
Cluster management	This feature now supports cluster data for Australia, Czech Republic, Portugal, and Slovakia.	22R1.1			●	●
Primary addresses	Define custom recalculation logic based on field conditions for Unique Checkbox primary address configurations.	22R1.1			●	●

			ST	DS	DM	AD
Field configurations	The Default Value and NEX Rules fields are bigger so it is easier to view and manage those values.	22R1.1			●	●
Data sources						
DCR routing for third party systems	Third party systems can support data change requests (DCRs) for customer managed fields on unverified records.	22R1.1			●	●
Parent HCO	When the Parent HCO object is managed by a third party system, the parent_hco_vid__v field is automatically set as a DCR enabled field.	22R1.1			●	●
Veeva OpenData subscriptions						
Export job error logs	Administrators can export the job error log.	22R1.1			●	●
Source subscriptions						
Transformation queries	Administrators can transform inbound data before it is loaded in source subscriptions.	22R2.0			●	●
Source file column headers	Administrators and data managers can use a new property rule to format column headers to use either lowercase (default) or uppercase letters.	22R1.1			●	●
Integrations						
Concur Connector	The SAP Concur Connector is updated to use OAuth2 authentication.	22R1.1				●
API						
Version update	The Network API is updated to v27.0.	22R2.0		Developers		
API updates for hashtags	A new value is added to the returnHashtagsForType parameter to display hashtags that are specific to Veeva CRM.	22R2.0		Developers		
Match Rule Collections	Administrators can create custom match rules to use in the Network API.	22R1.1	Administrators and Developers			

Note: The System and Data Admin user has all of the capabilities of the System Administrator and Data Steward users. Features and enhancements that apply to those users also apply to the System and Data Admin user.

Data Governance - Specific updates for fields and reference data are provided in the *Veeva Network Data Governance* release notes for every minor and major Network release.

Hierarchy Explorer

Several enhancements have been added to Hierarchy Explorer to improve your user experience.

These enhancements are enabled by default.

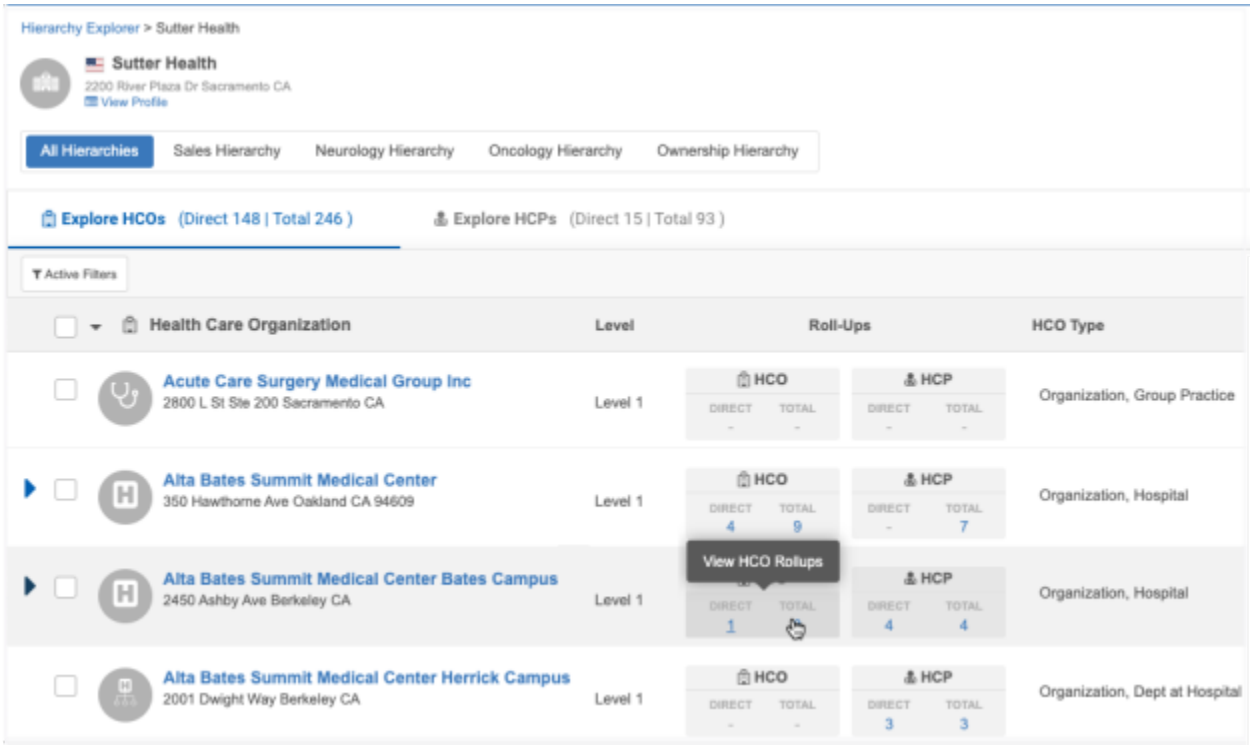
Note: The Hierarchy Explorer widget is available in Beta version. If you are interested in being an early adopter, contact Veeva Support.

Rollup Counts for HCOs

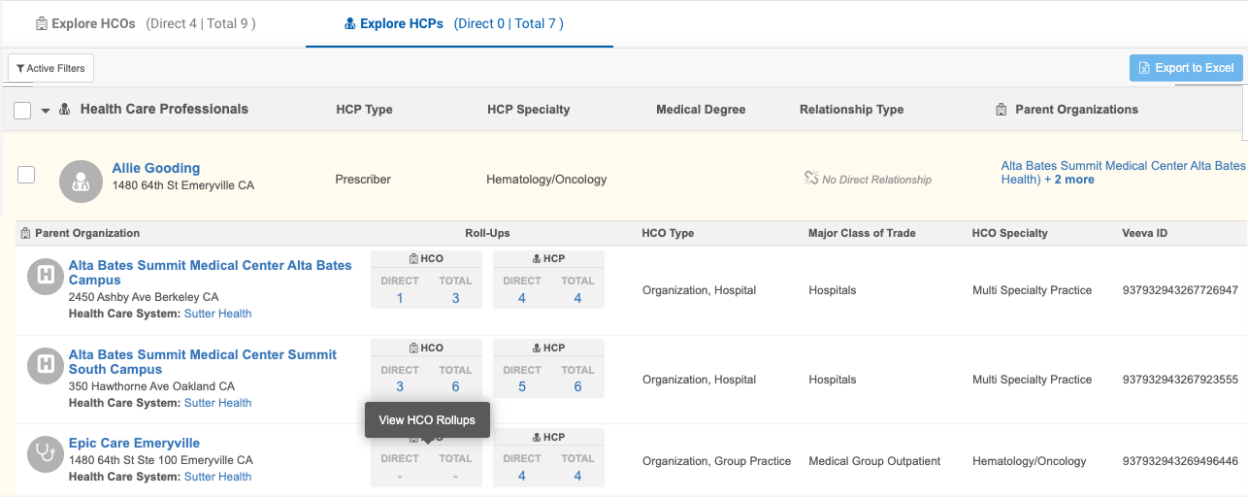
22R2

The Hierarchy Explorer widget now displays rollup counts for HCOs to help you understand the size of the health system.

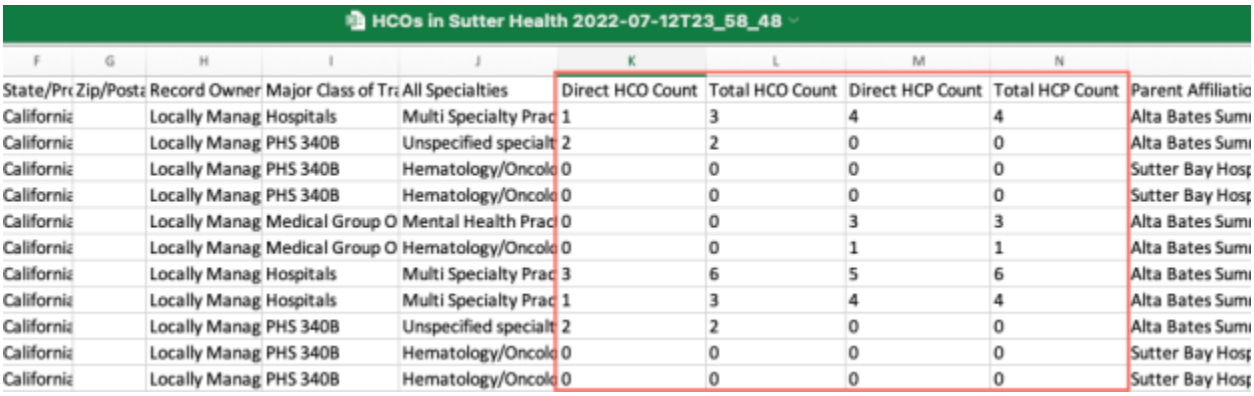
The **Direct** and **Total** counts display for each HCO in the list. Using the **Total** count, you can immediately see how many HCOs are indirectly related to the health system. Click a count to focus on that HCO's hierarchy.



The HCO rollup counts also display when you explore the **Parent Organizations** from the **Explore HCPs** tab.



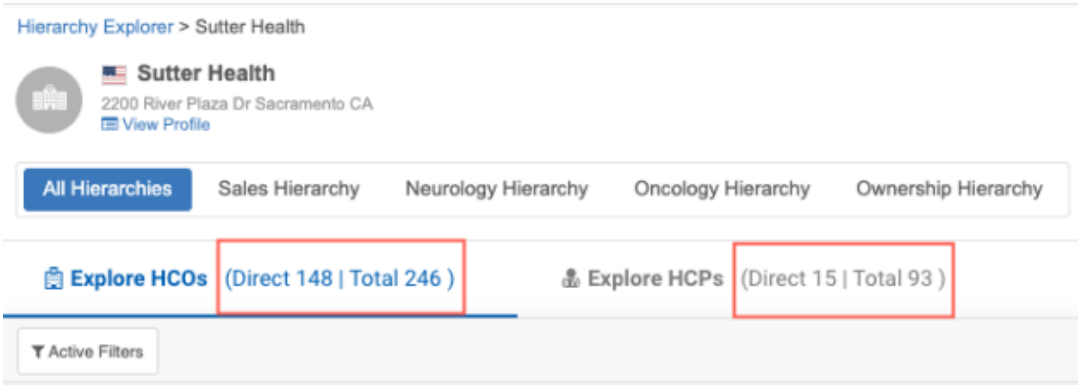
The direct and total HCO counts also display in the file when you export the hierarchy.



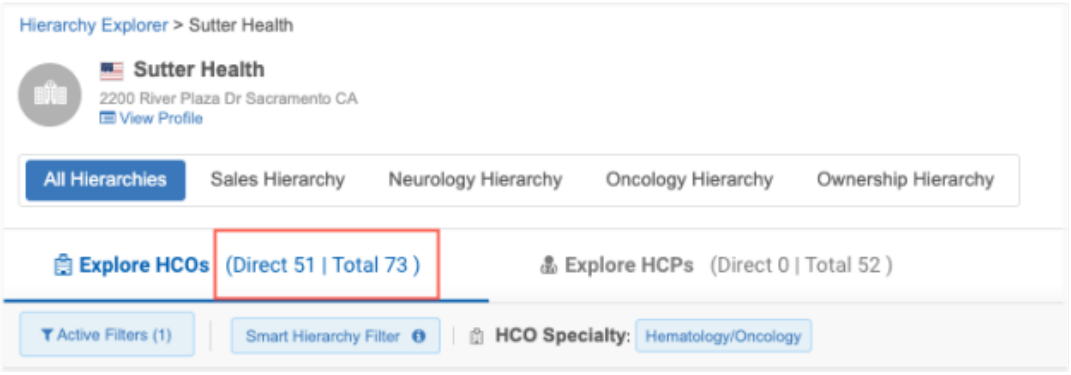
Counts in Explore tabs

The count of HCOs and HCPs display on the **Explore HCOs** and **Explore HCPs** tab headers so you can immediately understand the size of the health system without clicking into the tabs.

The **Direct** count indicates HCOs or HCPs that are immediately affiliated with the HCO. The **Total** count is the number of direct and indirect HCOs or HCPs.



If you filter the hierarchy, the counts are recalculated and apply to the filtered records.

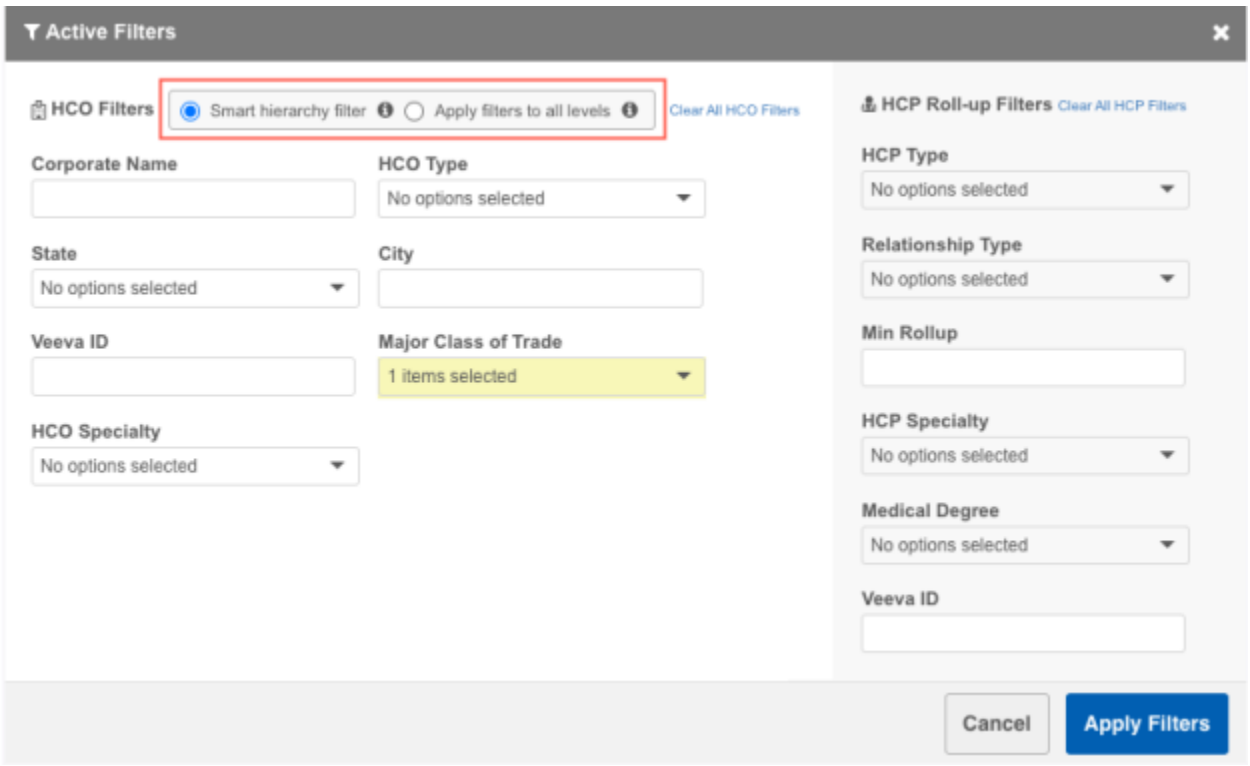


SMART FILTERING

22R2

You can now apply filters on the first level of a hierarchy so you can view the hospital hierarchies in the health system.

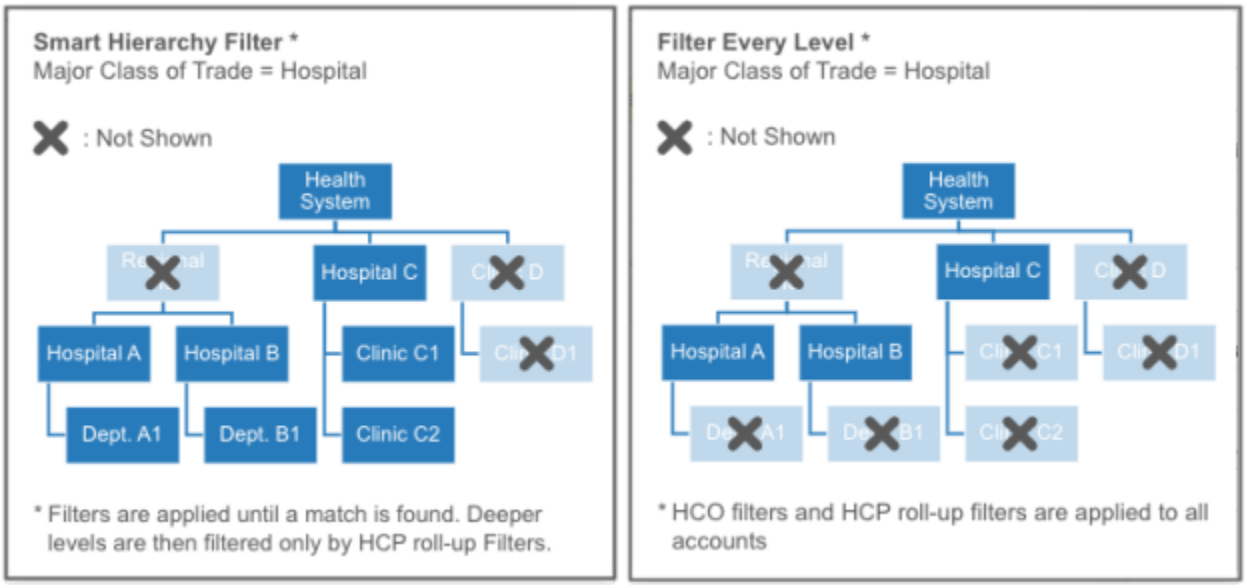
Currently, when you filter a hierarchy on hospitals, it filters every level and displays hospitals only; the hospital's children are not shown in the hierarchy. Using the new **Smart Hierarchy Filter**, you can see the hospital and all the HCOs in the hospital's hierarchy; for example, clinics and departments.



The **Active Filters** dialog contains the filtering options:

- **Smart hierarchy filter** - Filter on the first level of the hierarchy and display all HCOs in the hospital's hierarchy. This option is selected by default.
- **Apply filters to all level** - Filter on all levels of the hierarchy.

Click the tooltip to view an example for each filter option.

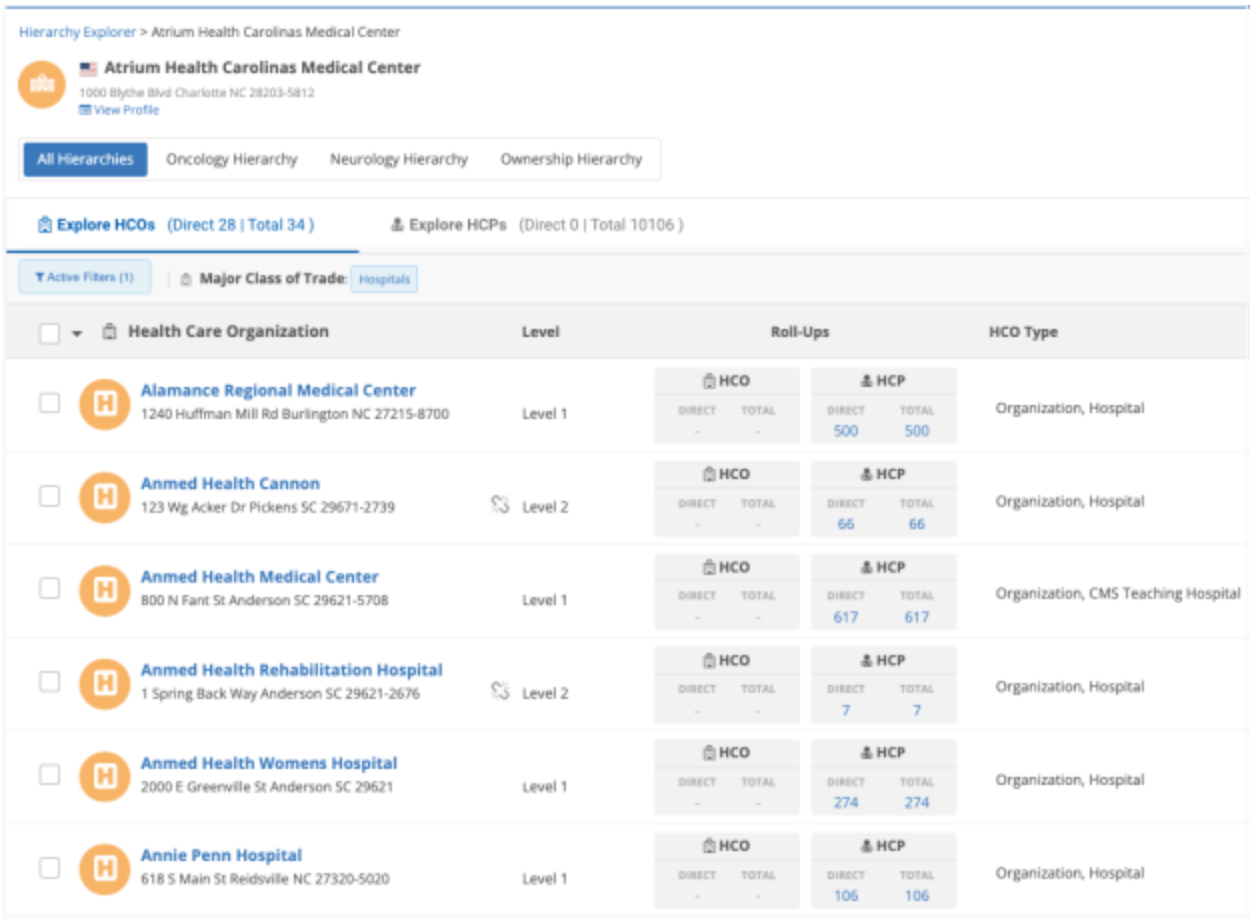


Example

We'll filter a health system to view hospital hierarchies. On the **Active Filter**, we'll select **Hospitals** for **Major Class of Trade**.

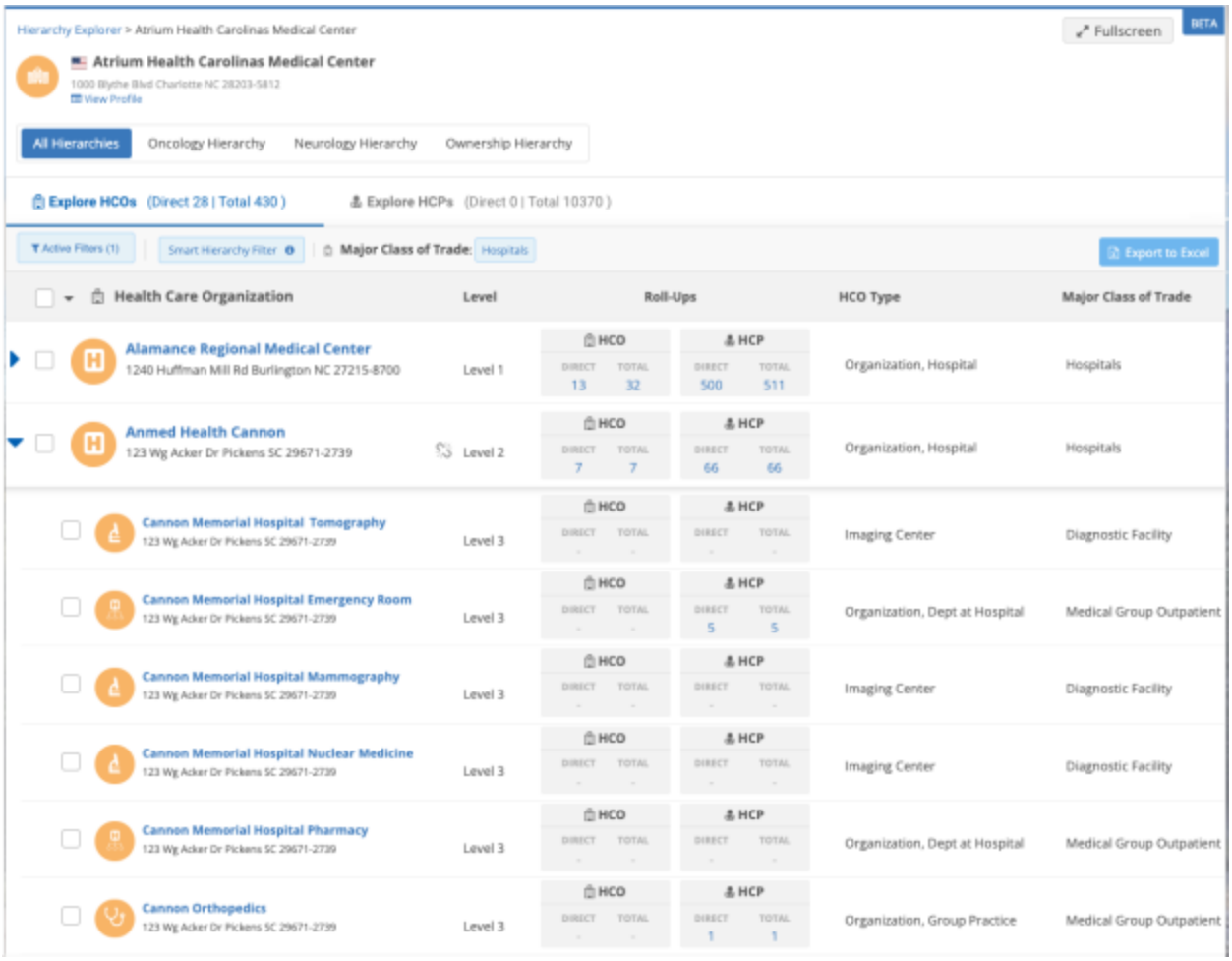
Apply filters to all levels

When we filter every level, Hierarchy Explorer displays hospitals in all levels of hierarchy. The hierarchy does not display child HCOs that are not hospitals.



Smart hierarchy filter

Using the smart filter, the first level of the health system hierarchy is filtered for hospitals and we can view all of the child HCOs in each hospital's hierarchy. In the example below, we can see departments and group practices in the HCO's hierarchy.



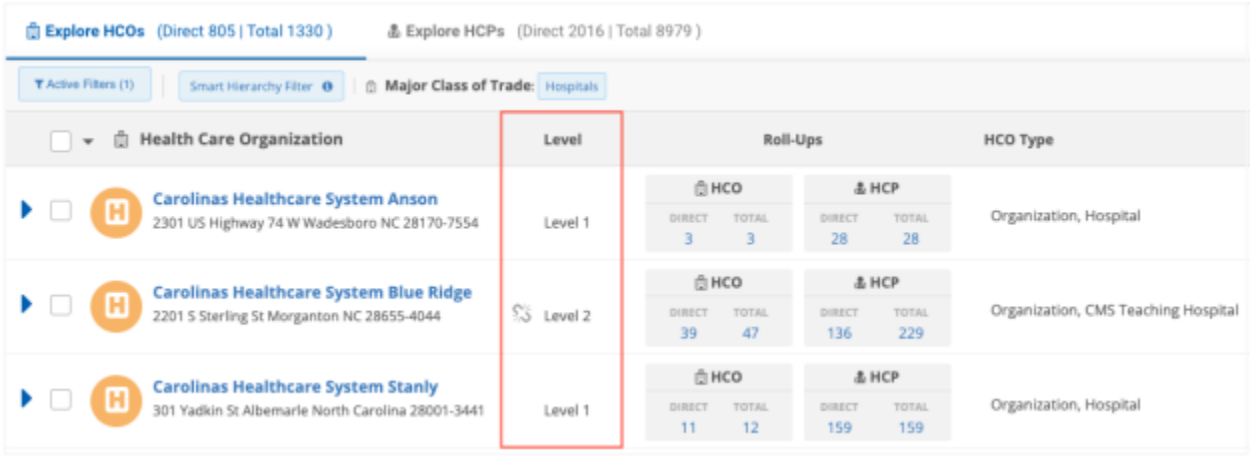
Navigating with filters applied

If you have applied smart filtering to a hierarchy, the filters will be removed when you click HCO or HCP rollup counts. If you return to the original hierarchy, the smart filtering is restored. The filtering remains if you are filtering all levels.

LEVEL COLUMN

22R2

The **Explore HCOs** tab contains a **Level** column so you can easily see what level the HCO is at within the health system.



FULLSCREEN

22R2

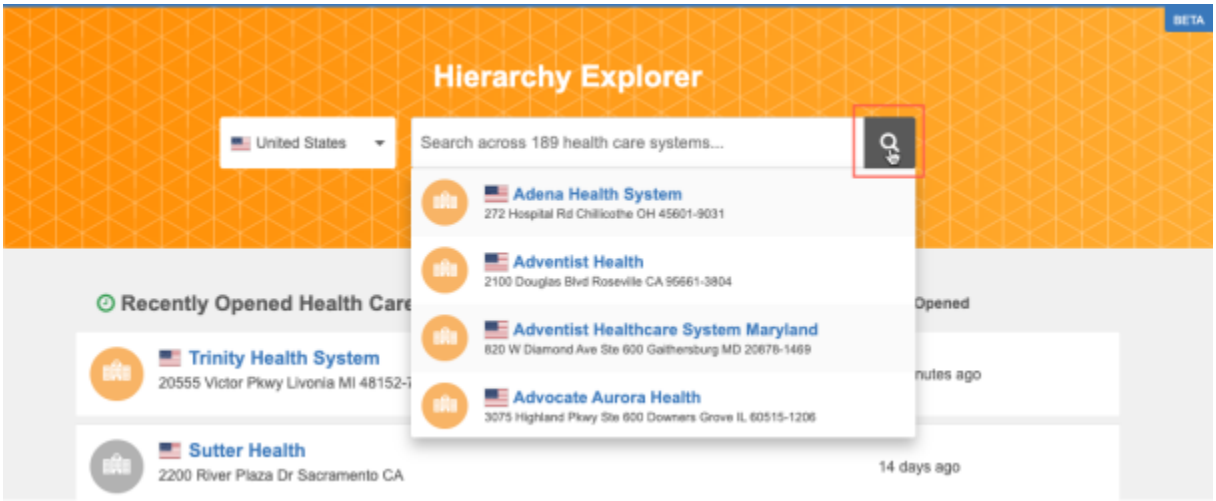
To give Hierarchy Explorer more screen space so you can better view the data, two enhancements are introduced:

- **Fullscreen mode** - Click **Fullscreen** in the header to expand the widget to your full browser screen and hide the Network Portal details at the top.
- **Hidden navigation pane** - If the application in the Network Portal has one widget only, the left navigation pane is removed by default.

HEALTH SYSTEM SEARCH

22R1.1

On the Hierarchy Explorer landing page, you can search for the health system that you want to explore. Click the **Search** icon to open the list of health systems that are available in your Network instance. The health systems are listed in alphabetical order.

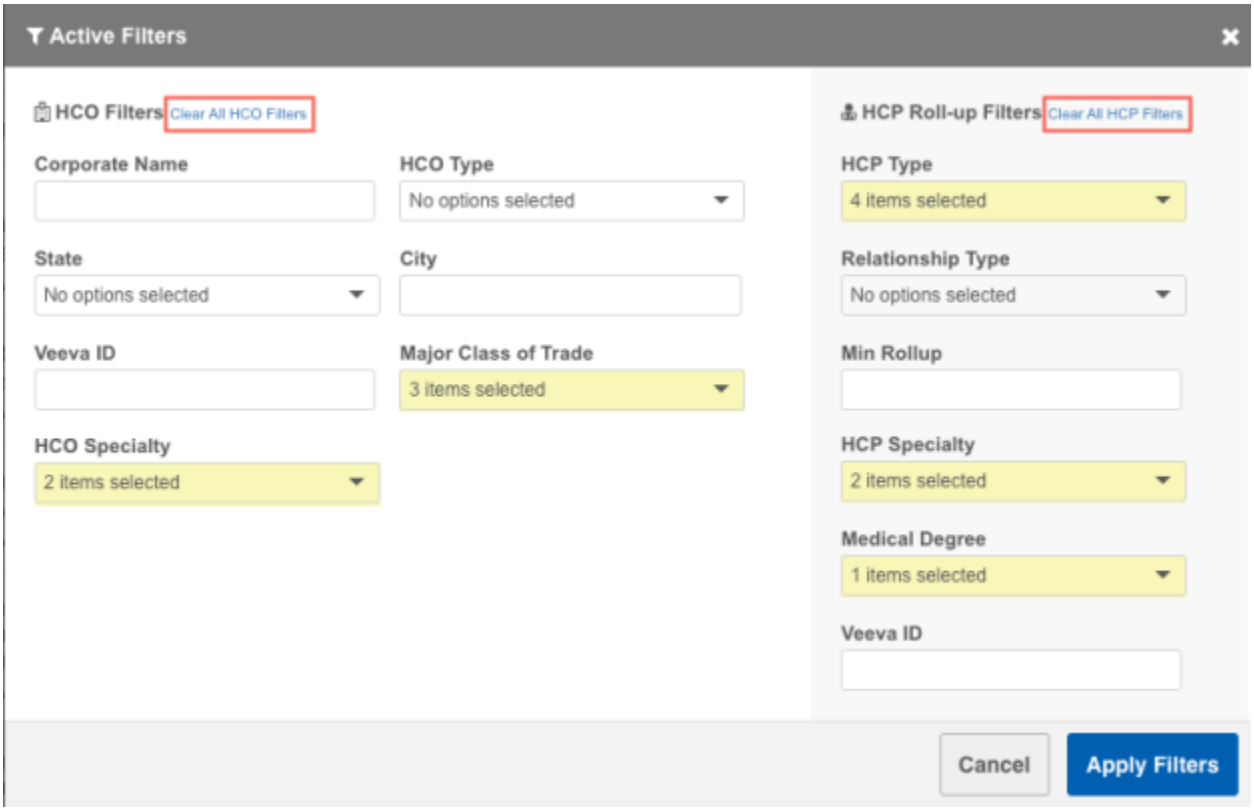


CLEAR FILTERS

22R1.1

The **Active Filters** dialog is updated to include the option to clear all of the selected filters.

Click **Clear All HCO Filters** or **Clear All HCP Filters** to reset your selections.

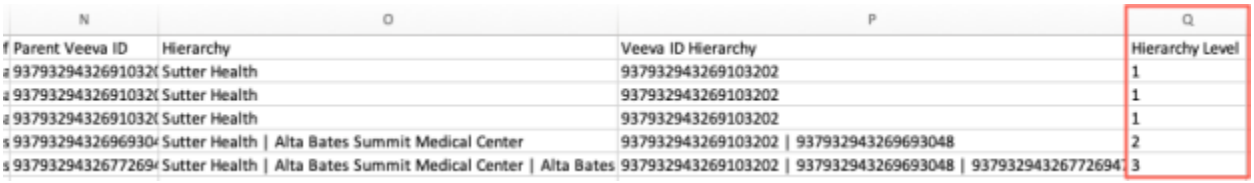


HIERARCHY LEVELS

22R1.1

When you download a hierarchy or selected records, the **Hierarchy Level** column now displays in the exported file.

Levels are the distance between the HCO at focus and the HCO in the hierarchy. For example, if the HCO is directly connected to the health system, the **Hierarchy Level** value is `1`.

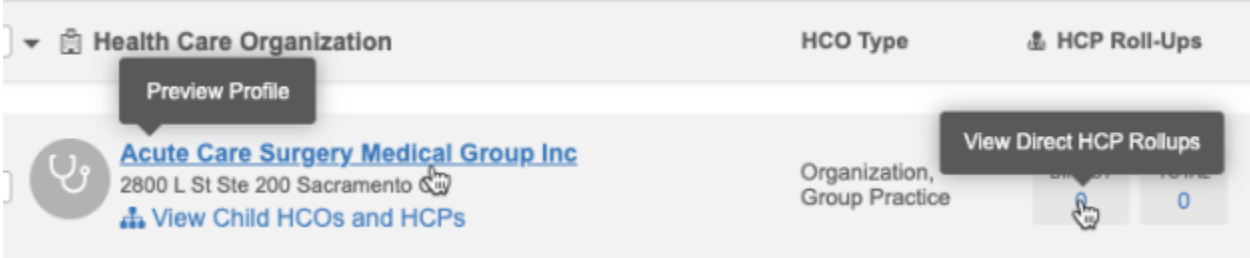


N	O	P	Q
Parent Veeva ID	Hierarchy	Veeva ID Hierarchy	Hierarchy Level
9379329432691032(	Sutter Health	937932943269103202	1
9379329432691032(	Sutter Health	937932943269103202	1
9379329432691032(	Sutter Health	937932943269103202	1
9379329432696930	Sutter Health | Alta Bates Summit Medical Center	937932943269103202 | 937932943269693048	2
9379329432677269	Sutter Health | Alta Bates Summit Medical Center | Alta Bates	937932943269103202 | 937932943269693048 | 937932943267726947	3

TOOLTIPS

22R1.1

Tooltips display when you hover over links to provide information about what happens when you click.



Network hashtags

NEW HASHTAGS

22R2

Predefined hashtags are added for all countries. These hashtags are enabled by default in your Network instance (Search and Profiles) and will be available in Network widgets and Veeva CRM by default.

Predefined hashtags

The following hashtags have been added to help you find HCPs and HCOs by leveraging specialties (for example, #allergy, #gastro, and so on).

Hashtag Name	Entity	Tooltip
#addiction	HCP, HCO	Record specializes in addiction medicine.
#allergy	HCP,HCO	Record specializes in allergy.
#anesth	HCP, HCO	Record specializes in anesthesiology.
#cardio	HCP, HCO	Record specializes in cardiology.
#dental	HCP, HCO	Record specializes in dentistry.
#derm	HCP, HCO	Record specializes in dermatology.
#diabetes	HCP, HCO	Record specializes in diabetes.
#emergency	HCP, HCO	Record specializes in emergency medicine.
#endo	HCP, HCO	Record specializes in endocrinology.
#gastro	HCP, HCO	Record specializes in gastroenterology.
#geriatric	HCP, HCO	Record specializes in geriatrics.
#hospice	HCP, HCO	Record specializes in hospice and palliative medicine.
#nephrology	HCP, HCO	Record specializes in nephrology.
#neuro	HCP, HCO	Record specializes in neurology.
#obgyn	HCP, HCO	Record specializes in pregnancy and female productive health.

Hashtag Name	Entity	Tooltip
#oncology	HCP, HCO	Record specializes in oncology.
#orthosurg	HCP, HCO	Record specializes in orthopedic surgery.
#oto	HCP, HCO	Record specializes in otology.
#pathology	HCP, HCO	Record specializes in pathology
#phsymed	HCP, HCO	Record specializes in physical medicine and rehabilitation.
#plasticsurg	HCP, HCO	Record specializes in plastic surgery.
#podiatry	HCP, HCO	Record specializes in podiatry.
#primarycare	HCP, HCO	Record specializes in primary care.
#proctology	HCP, HCO	Record specializes in proctology.
#psych	HCP, HCO	Record specializes in psychology.
#pulmonary	HCP, HCO	Record specializes in pulmonary.
#radiology	HCP, HCO	Record specializes in radiology.
#rheumatology	HCP, HCO	Record specializes in rheumatology.
#urology	HCP, HCO	Record specializes in urology.
#veterinary	HCP, HCO	Record specializes in veterinary.
#vision	HCP, HCO	Record specializes in vision.

NETWORK HASHTAGS IN VEEVA CRM

22R2

Administrators can configure Network hashtags to display in Veeva CRM. Hashtags can help CRM users find correct records faster in Network Account Search and throughout the CRM application (both Online and iPad). For example, hashtags help with the following:

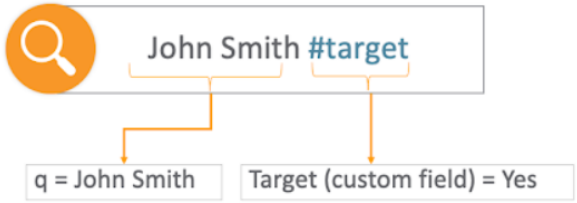
- Summarize a profile so you can identify HCPs and HCOs that have similar names
- View more accurate search results when hashtags are applied
- Identify HCPs and HCOs that are marked as key targets from market flags
- Help find the HCOs that have sales data associated to it
- Prevent add requests from being created for duplicate records because it's difficult to find existing accounts

Network hashtags are available by default. CRM administrators must configure some settings to support Network hashtags in Veeva CRM.

About Network hashtags

Hashtags are dynamically calculated to display on records based on the rules in the hashtag configuration. You do not have to update records to add hashtags.

For example, if you use a hashtag called `#target` with an HCP name in your search query, the hashtag is automatically applied to HCP records that meet the `#target` hashtag rule (the `target` custom field value is `Y`). Your search results will be filtered to display only the HCPs that meet the name and hashtag criteria.



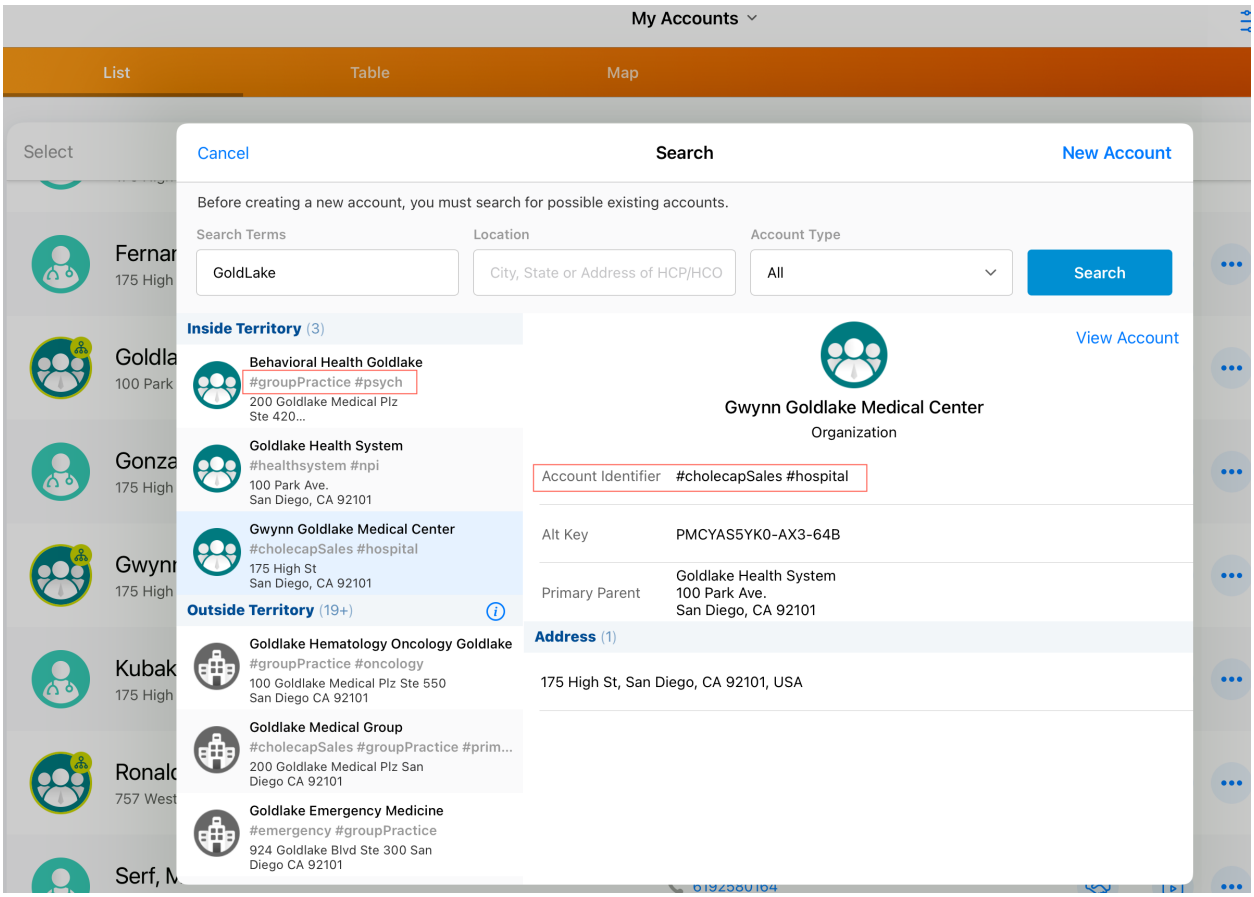
Network provides a number of predefined hashtags and Network administrators can create custom hashtags for your Network instance.

For more information, see the Network hashtags topic in the *Veeva Network Online Help*.

Example 1 - Searching for HCOs

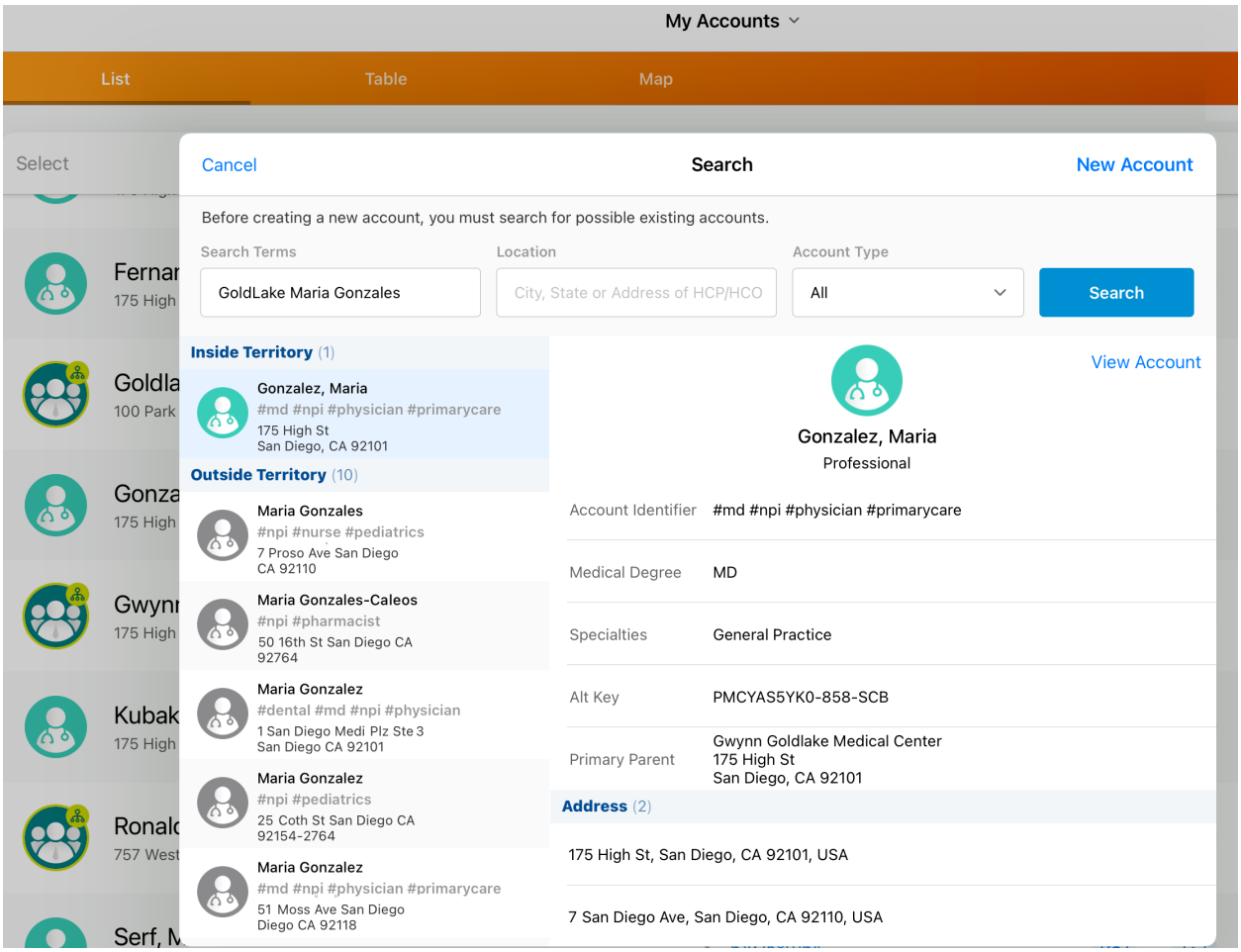
In this example, a CRM user searches for the UCLA HCO. The UCLA health system has a large number of locations. Using the Network hashtags below the HCO name, the user can quickly identify the type of HCO in the search results (for example `#hospital`, `#grouppractice`, `#healthsystem`).

Additionally, Network administrators can create hashtags based on sales data linked to the HCO. For example, the `#cholocapSales` hashtag indicates the HCO is a point-of-care location linked to 867 data.

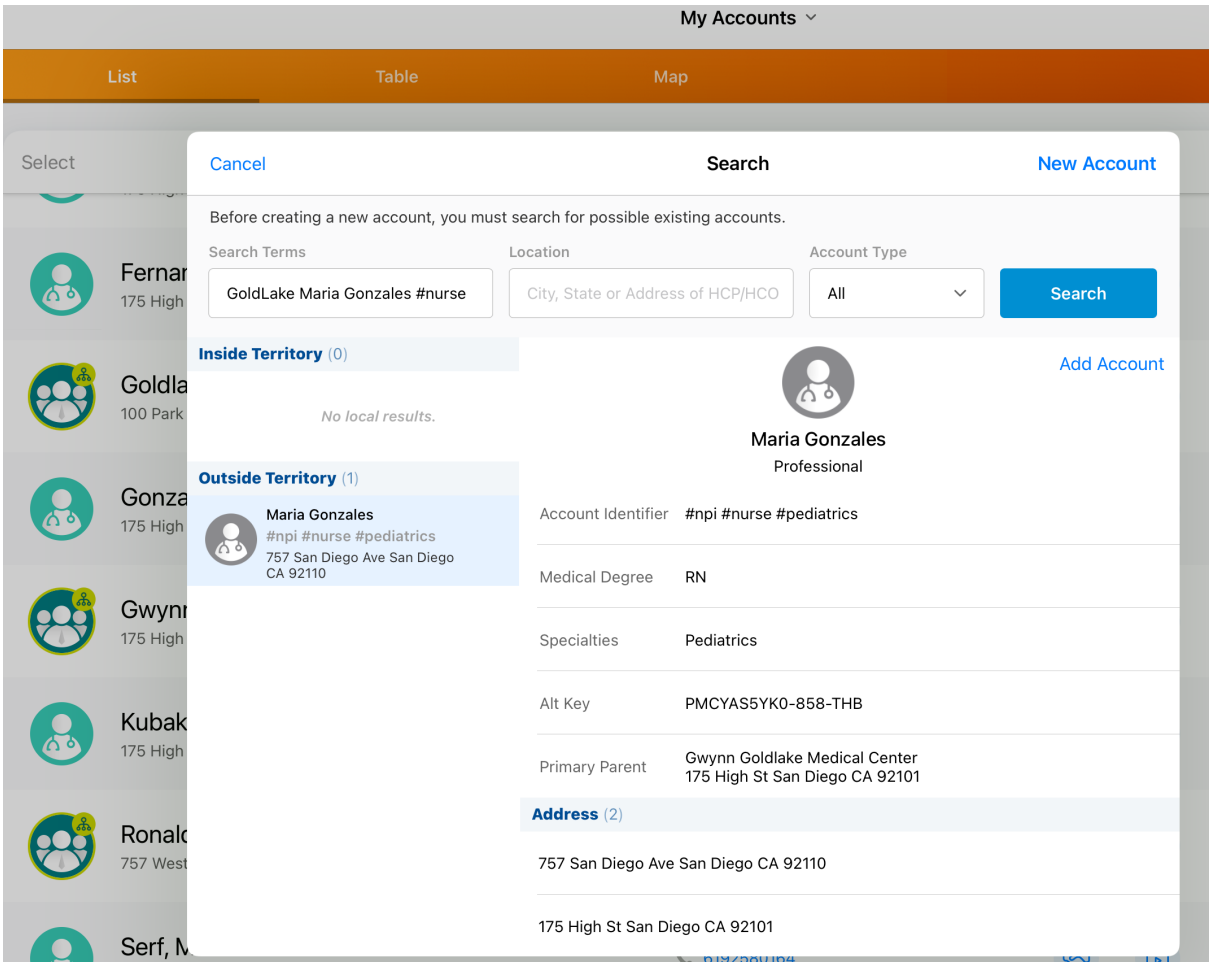


Example 2 - Searching for HCPs

The CRM user is looking for a nurse named Maria Gonzalez. This is a common HCP name and many results can be returned. The CRM user can use the hashtags to find the correct account.

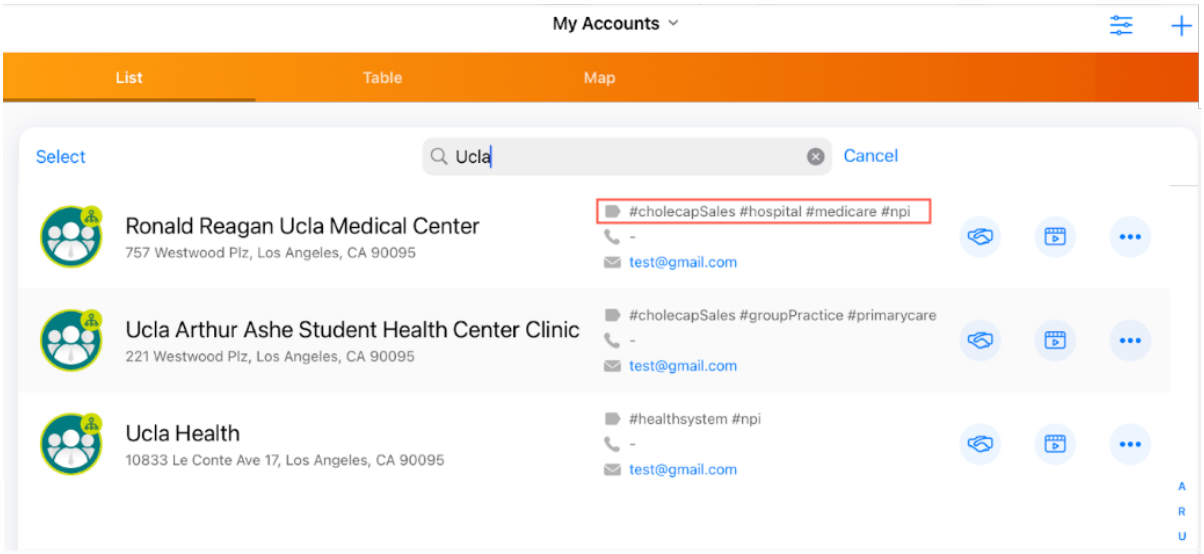


The CRM user adds the `#nurse` hashtag to the search terms. This makes their search more precise and significantly reduces the number of results that are returned.

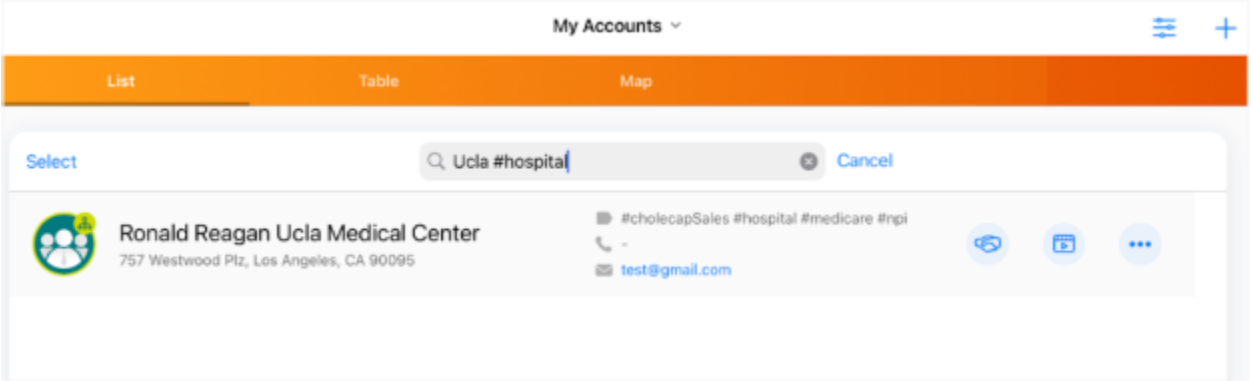


Example 3 - Viewing Account List

In this example, the CRM user can view the hashtags displayed under the account identifier in the Account List page.



From the account list, users can search using hashtags to find records. In this case, we can search for accounts that match `UCLA #hospital`. This will help filter the records within the account list.



Hashtag support in Veeva CRM

Hashtags can be configured to display in Network Account Search only or throughout the CRM application. If they are configured for the CRM application, they will display in the following areas:

- **Accounts** - Network Account Search, Account lists, Account details, and more
- **Calls** - Group Calls, Call Reports, Call Location, Attendee Search
- **My Schedule** - Agenda, Week, Scheduler, Map View
- **Events Management** - Add Attendee, Attendee Details, Outside Territory Search
- **Stakeholder Navigator** - HCP and HCO accounts

Veeva CRM configuration

Network hashtags can be used in Network Account Search and Veeva CRM. They are supported for CRM Online and iPad.

To support hashtags, CRM administrators must set the `NETWORK_MANAGED_ACCOUNT_IDENTIFIER_vod` setting to one of the following options:

- **1** - Allow Network hashtags to display in Network Account Search.
- **2** - Allow Network hashtags to display in Network Account Search and throughout CRM.

  This option syncs the hashtags to the Account Identifier field.

For detailed information and additional configuration, see the Network Hashtags in CRM topic in the *Veeva CRM Online Help*.

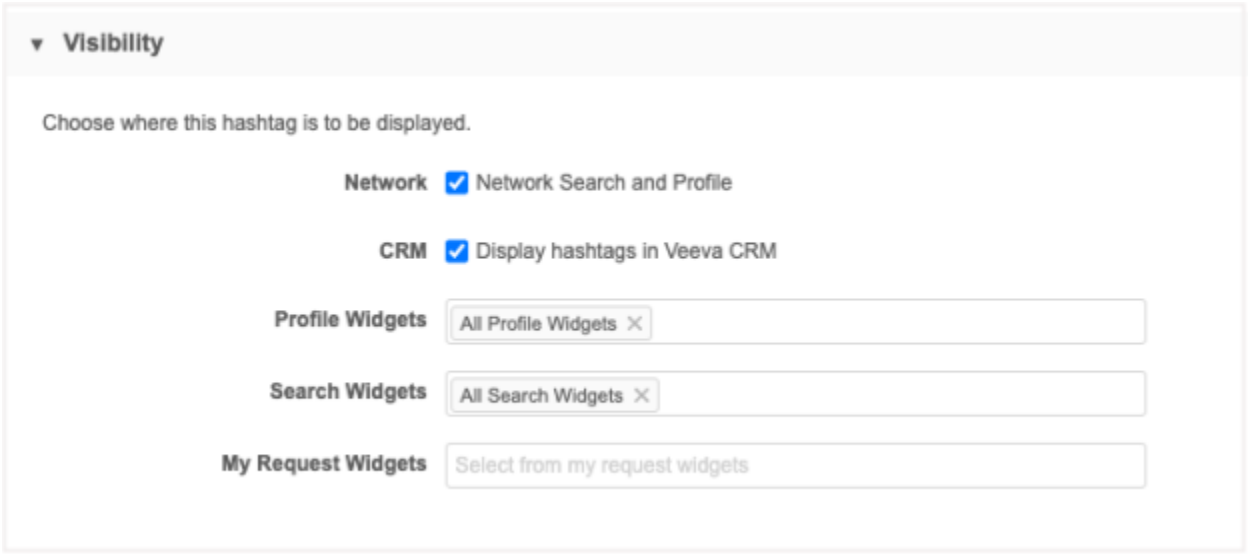
Note: No Network-CRM field mappings are required to support hashtags.

Network configuration

Each hashtag configuration includes an option to display the hashtag on accounts in Network Account Search and Veeva CRM. The option is enabled by default for all new hashtags and for all existing hashtags, except for the `#crm` and `#candidate` hashtags.

To enable or disable a hashtag for CRM:

1. In the Admin console, click **Data Model > Network Hashtags**.
2. Open an existing hashtag or create a new hashtag.
3. In the **Visibility** section, select (enable) or clear (disable) the **CRM** option.
4. **Save** your changes.



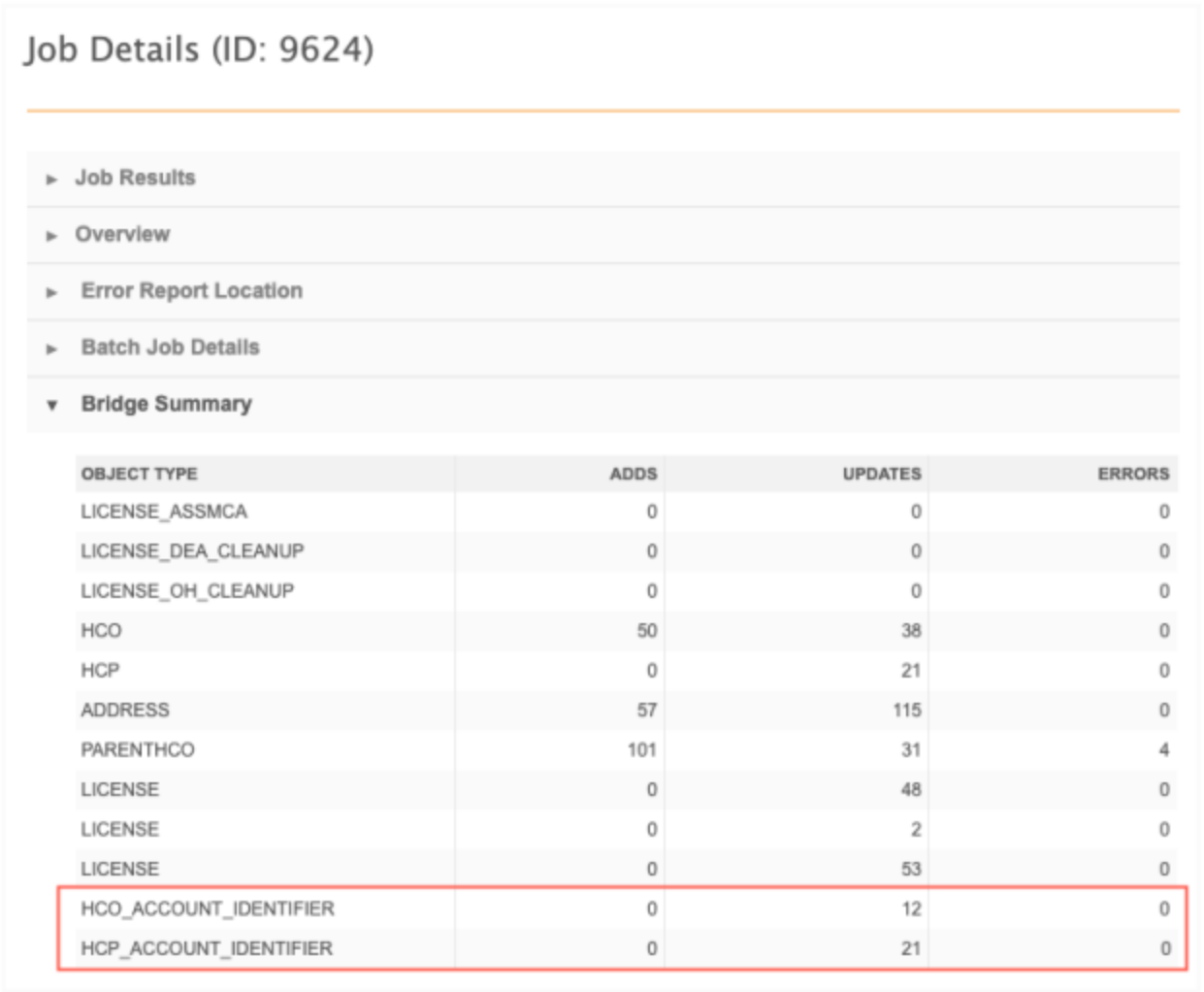
Pushing hashtags to Veeva CRM

When the Network Bridge runs, it checks to see if the CRM org is configured to support hashtags and it will update the Account Identifier field. No updates are required for the Network Bridge configuration to support pushing hashtags to CRM.

A hashtag file is generated for the hashtags that are updated.

- Single country bridge - Only the records from that country and that are in that CRM org are in the file.
- Multi-country bridge - Records from multiple countries and that are in that CRM org are in the file.

On the Job Details page, you can see the number of HCP_ACCOUNT_IDENTIFIER and HCO_ACCOUNT_IDENTIFIER updates that occurred during the Bridge job.



Job Details (ID: 9624)

▸ Job Results

▸ Overview

▸ Error Report Location

▸ Batch Job Details

▾ Bridge Summary

OBJECT TYPE	ADDS	UPDATES	ERRORS
LICENSE_ASSMCA	0	0	0
LICENSE_DEA_CLEANUP	0	0	0
LICENSE_OH_CLEANUP	0	0	0
HCO	50	38	0
HCP	0	21	0
ADDRESS	57	115	0
PARENTHCO	101	31	4
LICENSE	0	48	0
LICENSE	0	2	0
LICENSE	0	53	0
HCO_ACCOUNT_IDENTIFIER	0	12	0
HCP_ACCOUNT_IDENTIFIER	0	21	0

Data lineage

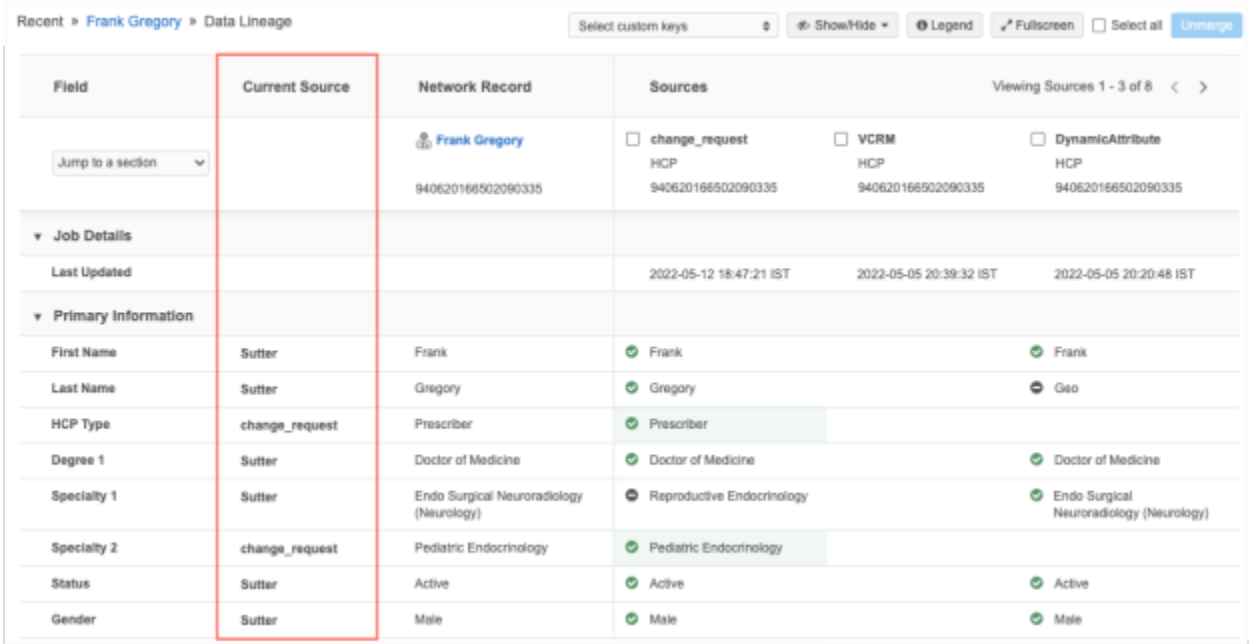
Several updates have been made to the data lineage page to help you identify the winning source for field values and to provide field and sub-objects details.

These enhancements are enabled by default for your Network instances.

CURRENT SOURCE COLUMN

22R1.1

The **Current Source** column is added to display the name of the source that provided the winning value for each field.

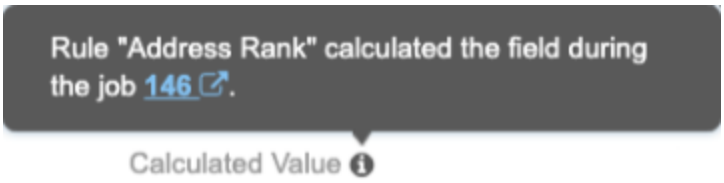


Other source values

Some fields display values other than a source name if the field value was updated by Network.

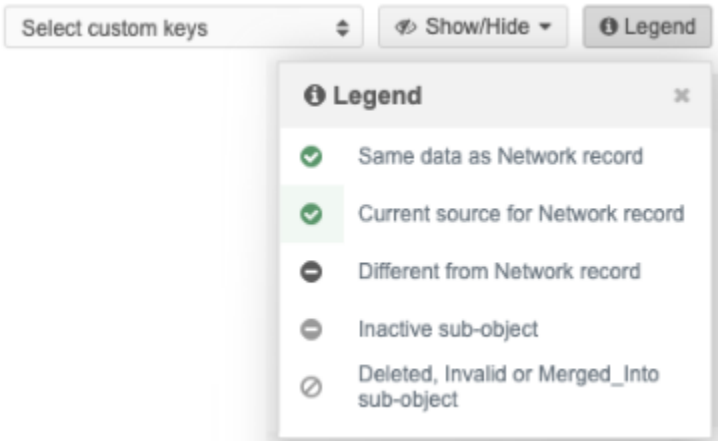
- **Updated by System** - System fields that are updated by Network. This value also displays for data that was calculated by Network using rules before these enhancements were released; that data is not backfilled for older values.
- **Calculated Value** - Displays on fields where the value is calculated by Network rules (for example, fields updated by NEX rules, default values, or primary values when a source file does not contain a value).

  Hover to view a tooltip that identifies the rule and job that calculated the value. Administrators and data managers can click the job ID to navigate to it for more details.



Legend

Icons display to help you identify information about objects and field values. Click **Legend** to view a description of each icon.



Same data as Network record - The field value from the source is the same as the Network record.

Current source for Network record - The field value is the current source for the Network record. The background is highlighted in green.

Different from Network record - The field value is different than the current value on the Network record.

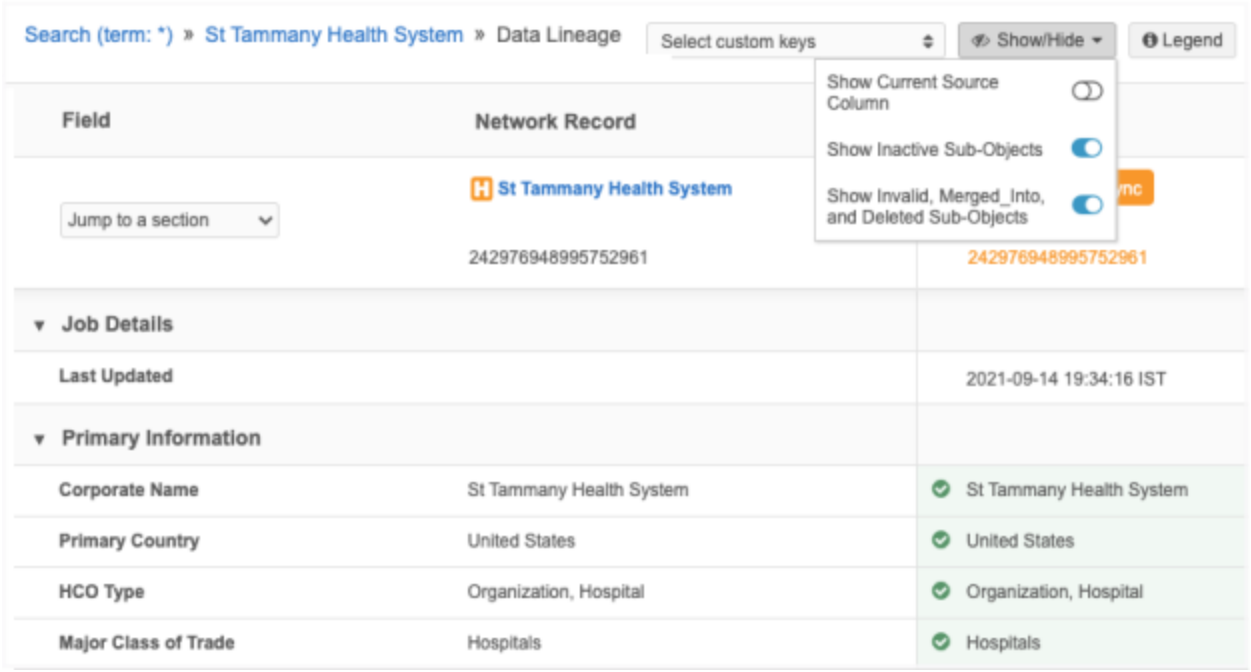
Inactive sub-object - The status of the sub-object is inactive.

Deleted, Invalid or Merged_Into sub-object - The record state of the sub-object.

Note: If the **Current Source** column is not enabled for your Network instance, the **Legend** does not display.

Show or Hide options

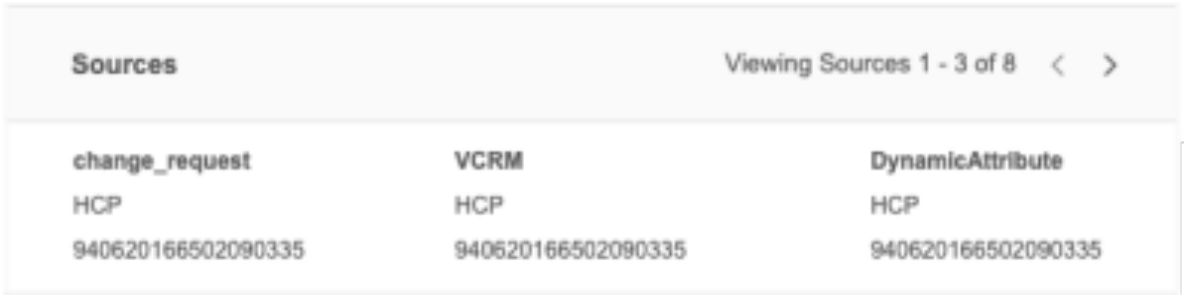
Click the **Show/Hide** button to toggle column and object options. By default, all options are enabled. Your preferences are retained the next time you open the Data Lineage page.



- **Show Current Source Column** - Display or hide the column. If you hide the column, the current source remains highlighted in green in its respective source column.
- **Show Inactive Sub-Objects** - Display or hide inactive sub-objects. When the objects display, they can be identified by the **Inactive** icon.
- **Show Invalid, Merged_Into, and Deleted Sub-Objects** - Display or hide sub-objects with these record states. When the objects display, they can be identified by the **Invalid** icon.

View sources

The **Sources** section now displays pagination so you can navigate to the next page if there are multiple sources. The page numbers update as you click through the sources.



SUB-OBJECT DETAILS

22R1.1

Addresses, licenses, and parent HCOs now contain a summary so you can easily view relevant information about the sub-object. Additionally, these sub-objects are now sorted so the active objects display before the inactive objects. Inactive objects display so you have a history of specific addresses. You can remove inactive sub-objects using the **Show/Hide** button.

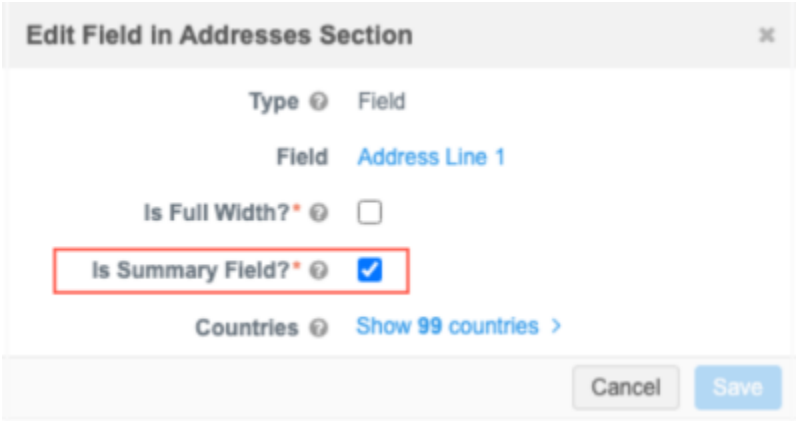
These enhancements are enabled by default in your Network instance.

Addresses

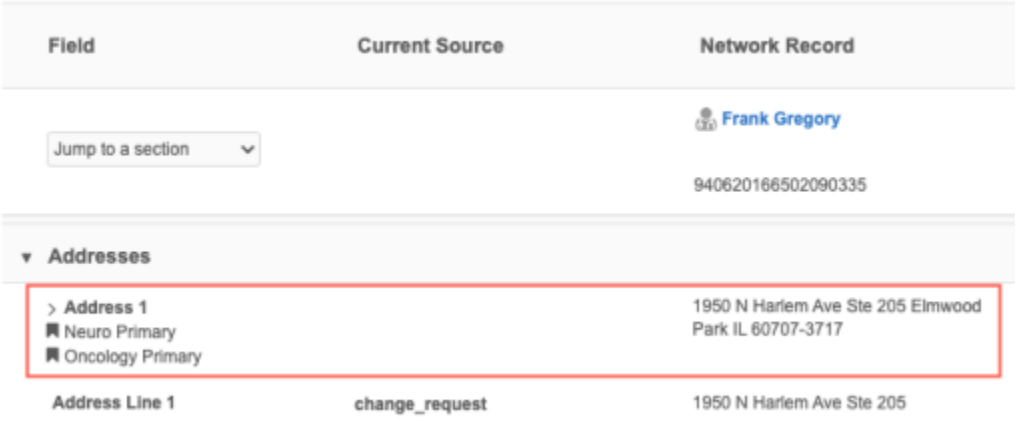
Address summaries contain the following details:

- Formatted address
- Fields that are defined as **Is Summary Field** on profile layouts.

  For example, in the profile layout, the **Is Summary Field** option is selected for Address Line 1. This means that the field will display in the summary header on the record profile, and it will also display in the address summary on the Data Lineage page.



- Status or State (hidden if the address is Active and Valid)
- Primary address flag (if applicable)



Sort order

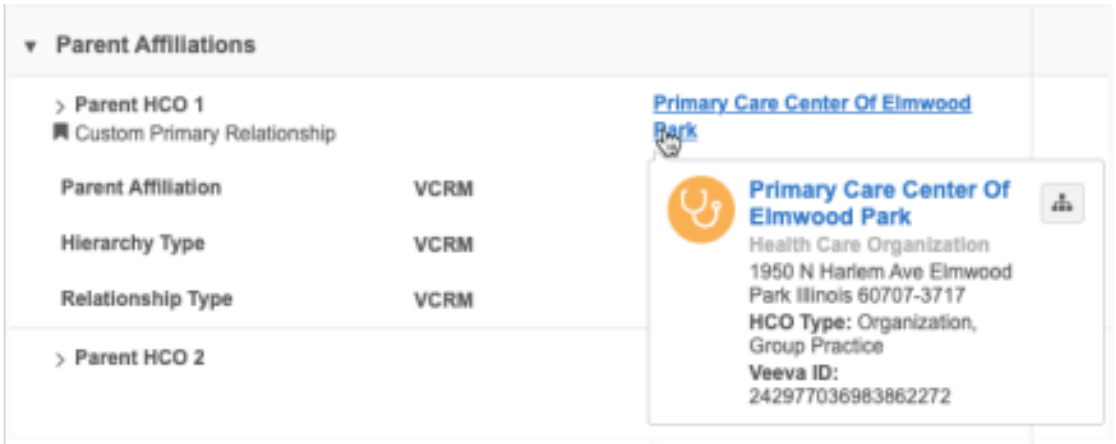
Addresses are sorted in the following order:

- Rank
- Status (Active before Inactive)
- Record state (ordered by Valid, Merged_Into, Merge_Inactivated, Merge_Added, Invalid, and then Deleted)

Parent HCO

Parent HCO summaries contain the following details:

- HCO corporate name (click the link to view the business card)
- Fields that are defined as **Is Summary Field** on profile layouts.
- Status or State (hidden if the Parent HCO is Active and Valid)
- Primary affiliation flag (if applicable)



Sort order

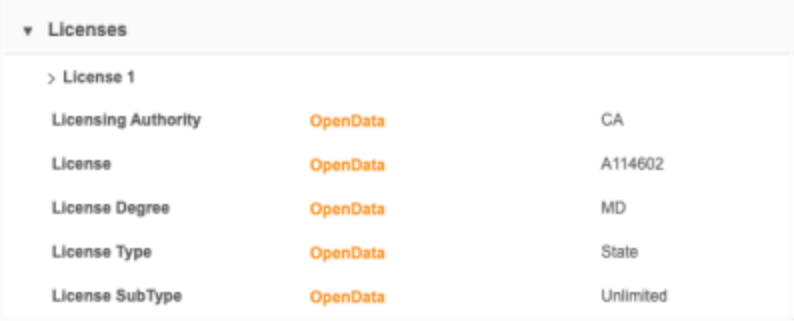
Parent HCOs are sorted in the following order:

- Status (Active before Inactive)
- Record state (ordered by Valid, Merged_Into, Merge_Inactivated, Merge_Added, Invalid, and then Deleted)

Licenses

License summaries contain the following details:

- Fields that are defined as **Is Summary Field** on profile layouts.



Sort order

Licenses are sorted in the following order:

- Status (Active before Inactive)
- Record state (ordered by Valid, Merged_Into, Merge_Inactivated, Merge_Added, Invalid, and then Deleted)

Inbox

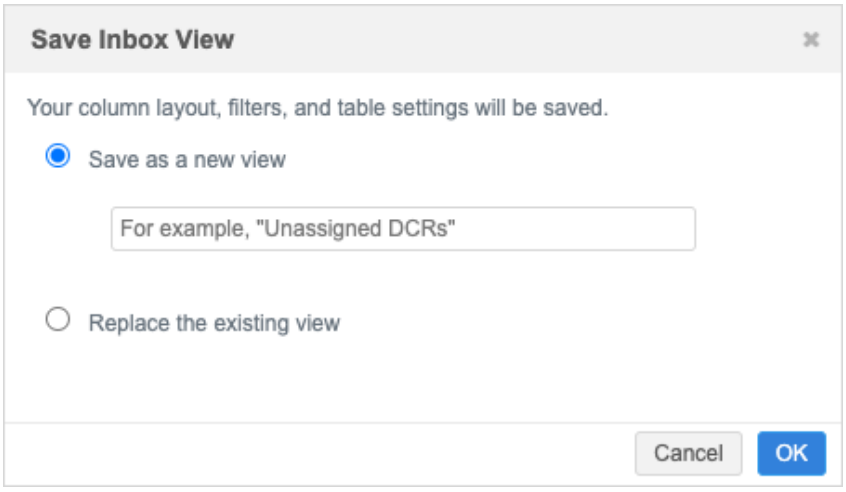
SAVE AS INBOX VIEW

22R2

When Data Stewards make changes to their inbox view, they can now save the changes as a new view. Previously, when Data Stewards clicked **Save changes**, the current inbox view was updated with the change. Now, a pop-up displays so the updates can be quickly saved as a new view.

On the **Save Inbox View** pop-up, Data Stewards can choose one of the options:

- **Save as a new view** - Create a new view. The column layout, filters, and table settings from the current view will be saved to the new view. This is the default option.
- **Replace the existing view** - Update the current view with these changes.



This enhancement is enabled by default in your Network instance.

Reports

MERGED HCO AND HCP REPORTS

22R2

The following Saved Reports will be discontinued in a future release:

- Merged HCO Report
- Merged HCP Report

These reports are legacy reports that do not use the current Network reporting warehouse.

For best results, begin using the provided SQL queries to report on merged HCOs and HCPs. You can save the queries as saved reports.

Merged HCOs query

```
SELECT
           hco_surviving.vid__v AS "Veeva ID (Surviving)",
           hco_non_surviving.vid__v AS "Veeva ID (Non-Surviving)",
           hco_revision.created_at AS "Timestamp"

     FROM
         hco_revision
           INNER JOIN hco AS hco_non_surviving ON hco_revision.vid__v =
hco_non_surviving.vid__v
           INNER JOIN hco AS hco_surviving ON
hco_non_surviving.record_merged_vid__v = hco_surviving.vid__v
     WHERE
         hco_revision.record_state__v = 'MERGED_INTO'
         and hco_revision.created_at >= 'insert_date_here' -- example
'2022-01-30'
```

Merged HCPs query

```
SELECT
           hcp_surviving.vid__v AS "Veeva ID (Surviving)",
           hcp_non_surviving.vid__v AS "Veeva ID (Non-Surviving)",
         hcp_revision.created_at AS "Timestamp"

     FROM
         hcp_revision
           INNER JOIN hcp AS hcp_non_surviving ON hcp_revision.vid__v =
hcp_non_surviving.vid__v
           INNER JOIN hcp AS hcp_surviving ON
hcp_non_surviving.record_merged_vid__v = hcp_surviving.vid__v
     WHERE
         hcp_revision.record_state__v = 'MERGED_INTO'
         and hcp_revision.created_at >= 'insert_date_here' -- example
'2022-01-30'
```

Save the reports

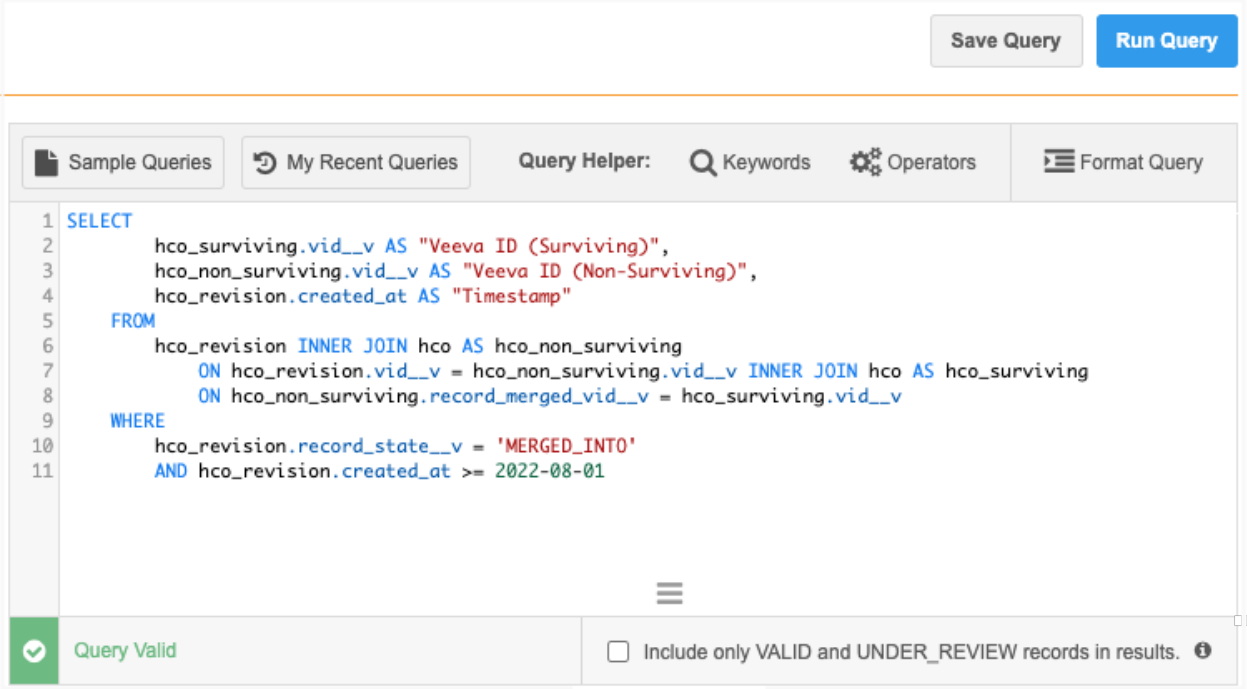
Save these queries as saved reports. You can schedule saved reports to run or you can run them manually.

To save a report:

1. On the Network menu bar, click **Reports > SQL Query Editor**.
2. Paste the query in the query box.

   Customize the query by defining the date parameter and include any additional columns.

   Example



3. Clear the **Include only VALID and UNDER_REVIEW records in results** option.
   This ensures that records with the `record_state__v` field set to MERGED_INTO display in the results.
4. Click **Save Query**. The button is available only if the query is valid.
5. In the **Save Report As** dialog, type a name and description.
6. On the report configuration page, you can choose to schedule the report, share it with other users, define download options, and make changes to the SQL query.

   For more details , see the Saved report options topic in the *Veeva Network Online Help*.

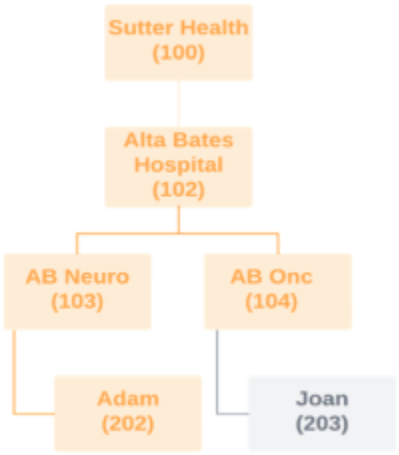
REPORTING ON HIERARCHIES

22R1.1

A new table called `flat_hierarchy` is added to Network reports to help you report on relationship hierarchies. The table updates hierarchies once a day and displays all of the relationships and levels so you can see how HCPs and HCOs rollup to HCOs.

Existing hierarchy reporting

Using the existing `parenthco` table, you can view each entity and its direct parent HCO. You can create a SQL query and join the table with other tables, but this can be complex.



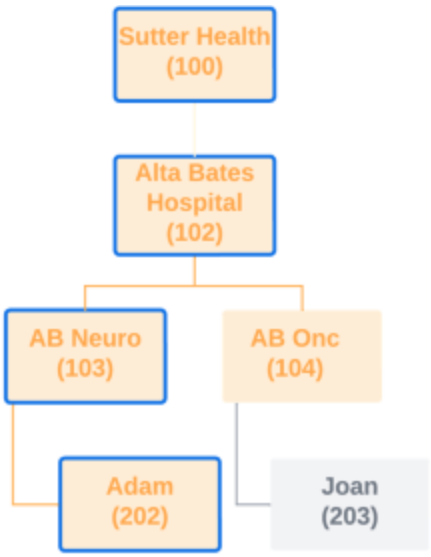
Entity VID	Entity Type	Parent HCO
Joan (203)	HCP	104 (AB Onc)
Adam (202)	HCP	103 (AB Neuro)
AB Onc (104)	HCO	102 (AB Hosp)
AB Neuro (103)	HCO	102 (AB Hosp)
AB Hospital (102)	HCO	100 (Sutter)

Flattened hierarchy reporting

Using the new `flat_hierarchy` table, you can view all the relationships and understand how HCPs and HCOs roll up to direct and indirect HCOs.

Example

For the HCP, Adam, a row is populated for every HCO that he's connected to so you can see all the relationships and the levels to understand how he rolls up to each HCO.



Entity VID	Entity Type	Ancestor VID	Distance
Adam (202)	HCP	AB Neuro (103)	1
Adam (202)	HCP	AB Hospital (102)	2
Adam (202)	HCP	Sutter (100)	3
Joan (203)	HCP	AB Onc (104)	1
Joan (203)	HCP	AB Hospital (102)	2
Joan (203)	HCP	Sutter (100)	3
AB Onc (104)	HCO	AB Hospital (102)	1
AB Onc (104)	HCO	Sutter (100)	2
AB Neuro (103)	HCO	AB Hospital (102)	1
AB Neuro (103)	HCO	Sutter (100)	2
AB Hospital (102)	HCO	Sutter (100)	1

This feature is available by default. Administrators must enable the `flat_hierarchy` table by setting a schedule to populate the data for their Network instance.

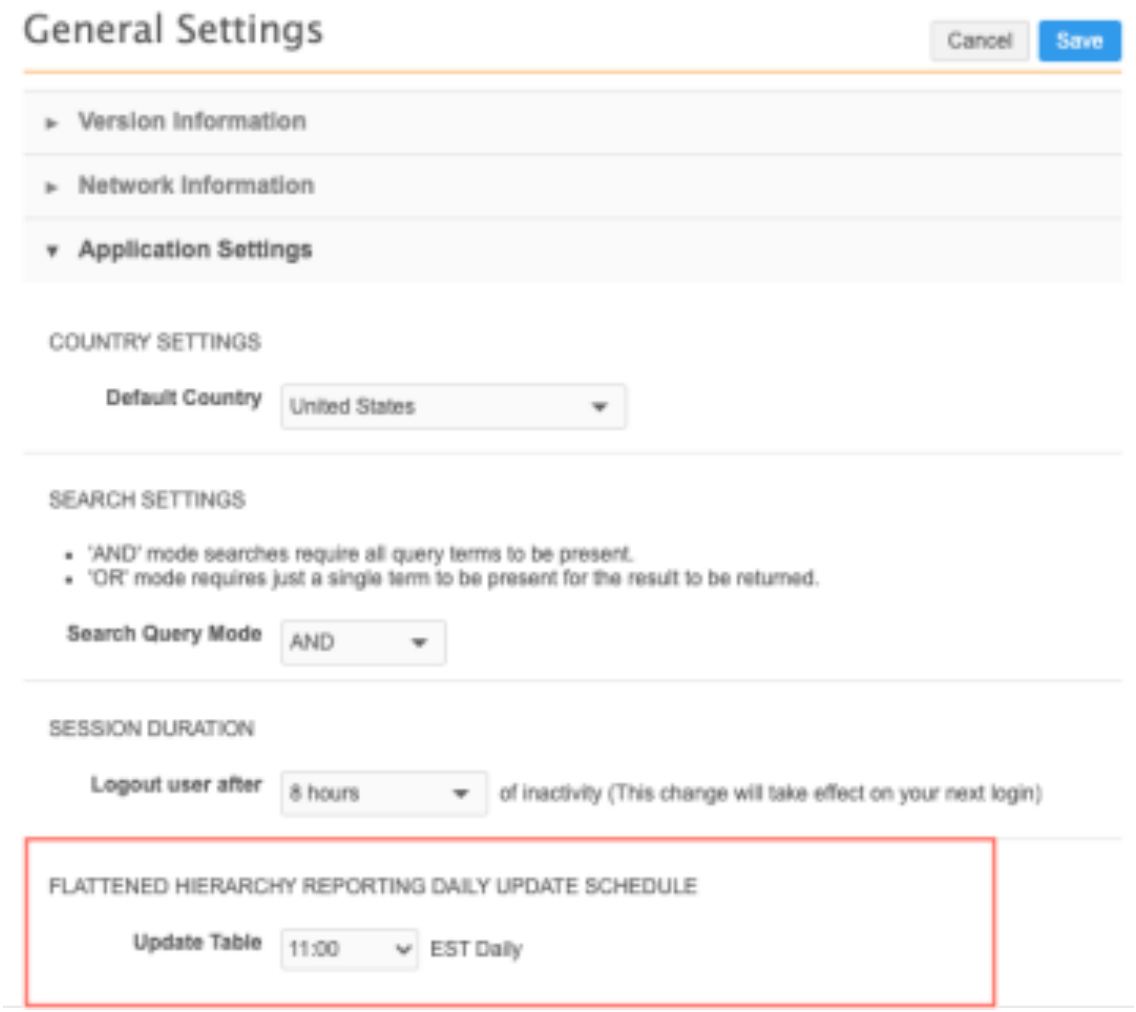
Note: You do not need to have the Hierarchy Explorer widget enabled to use this feature.

Benefits

- **Explore targets** - Find the HCPs in a specific health system or understand the health systems you should target for specific therapeutic areas. For example, you can produce a report for health systems that have a large number of HCPs that are gastroenterologists.
- **Maintain hierarchies and tracking changes** - Compare old versions of your custom hierarchy to the new version to find breakages. You can also use it to understand how data source updates impact the hierarchy. For example, if a data source inactivates an HCO, you can see how that change impacts a hierarchy.
- **Roll up sales and interactions to the health system** - Allow analytics teams to easily roll up the interactions and sales data to the health system or hospital.
- **Skip levels of the hierarchy when reporting** - Data Managers and analytics teams can skip levels of the hierarchy using SQL.
- **Export** - Export the report to your data warehouse so you can use it for business purposes like incentive compensation. This can be done using transformation queries.

Enable the flat hierarchy table

The `flat_hierarchy` reporting table is not enabled until Administrators set the schedule to update the data for their Network instance. Until the schedule is set, there is no data for the reporting database to return.



1. In the Admin console, click **Settings > General Settings**.
2. In the **Flattened Hierarchy Reporting Daily Update Schedule** section, expand the **Update Table** list and choose the time.
3. **Save** your changes.

Supported hierarchy paths

The flattened hierarchies support HCPs, HCOs, and ParentHCO relationships.

The report displays only active and valid relationship paths. Paths that do not meet this criteria are removed from the hierarchy.

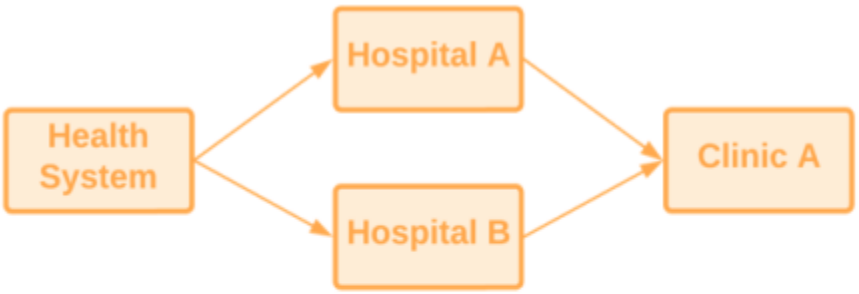
- **Inactive paths** - Paths are considered inactive for the following reasons:
  - HCP or HCP - Record status is not Active, they are candidate records, or HCPs have been opted out.
  - Relationships - The relationship status is not Active.
- **Invalid paths** - Paths are marked as invalid for the following reasons:
  - HCP/HCP - Record state is not Valid or Under Review (Invalid/Merged_Into/Deleted) or the record is unsubscribed (record state is Deleted).
  - Relationships - The record state is not Valid or Under_Review or the parent of the relationship is not in your Network instance.

Shortest path

The shortest path between two entities displays in the report. Duplicate short paths do not display to prevent duplicate entities in roll up counts.

Example 1

Between **Health System** and **Clinic A**, there are two paths with a distance value of `2`; however, the table displays only one row.



Child	Ancestor	Distance	Path
Clinic A	Health System	2	Clinic A | Hospital A | Health System
Clinic A	Hospital A	1	Clinic A | Hospital A
Clinic A	Hospital B	1	Clinic A | Hospital B
Hospital A	Health System	1	Hospital A | Health System
Hospital B	Health System	1	Hospital B | Health System

Example 2

Between **Health System** and **Clinic A** there are two paths:

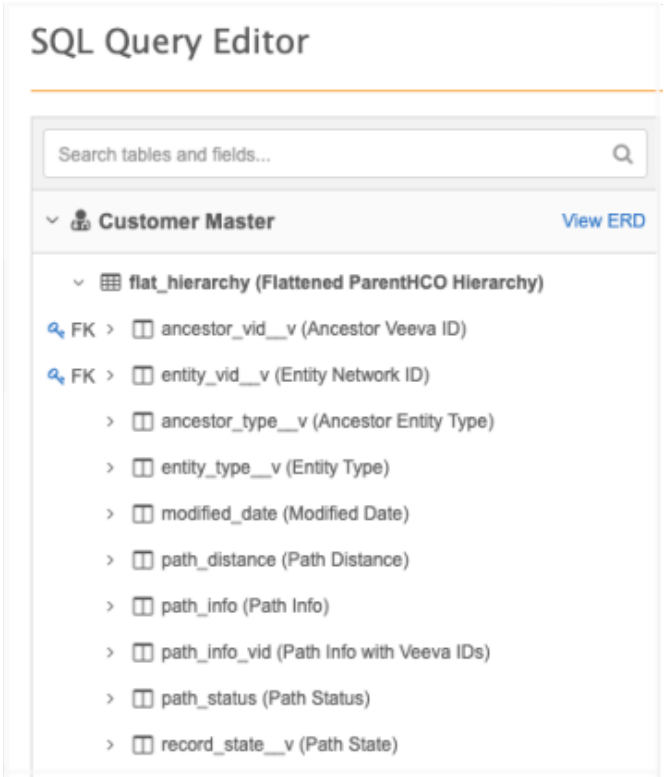
- **Health System → Hospital A → Clinic A**
- **Health System → Clinic A**)

The flat hierarchy table displays only the shortest path (**Health System → Clinic A**) to prevent duplicate counts when rolling up the data.

Child	Ancestor	Distance
Clinic A	Health System	1
Hospital	Health System	1
Clinic A	Hospital A	1

View the hierarchy table

The `flat_hierarchy` table is in the **Customer Master** section in the SQL Query Editor (**Reports**).



The table contains the following fields:

Field Name	Field Label	Field Type	Description
entity_vid__v	Entity VID	VID	Veeva ID of the child record.
entity_type__v	Entity Type	Reference	Type of the child record.
ancestor__vid__v	Ancestor Veeva ID	VID	Veeva ID of the ancestor record.
ancestor_type__v	Ancestor Entity Type	Reference	Type of the ancestor record. This value is always HCO.
path_distance	Path Distance	Number	Number of hops from the child to the ancestor. Direct relationships have 1 as the length.
record_state__v	Record State	Reference	State of the path. The value is always Valid. Paths that are not valid do not display in the reporting table.
path_status	Path Status	Reference	Status of the path. The value is always Active. Paths that are inactive do not display in the reporting table.
modified_date	Modified Date	Date Time	The last modified date for any entity or relationship in the path.
path_info	Path Info	Text	Path from the child record to the ancestor using names. The names are separated by the pipe (|) character.
path_info_vid	Path Info with Veeva IDs	Text	Path from the child record to the ancestor HCO using Veeva IDs. The VIDs are separated by the pipe (|) character.

Flattened hierarchy report examples

Review the following examples to see how you can use the `flat_hierarchy` table.

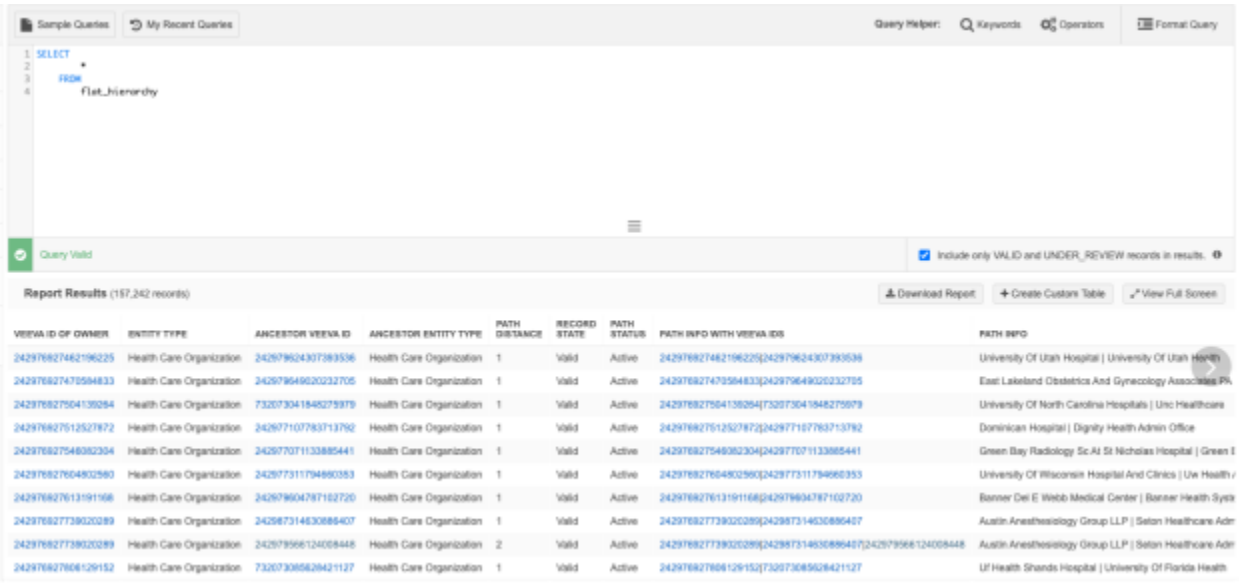
Example 1 - flat_hierarchy table structure

Use this basic query to view the structure of the table.

Query

```
select * from flat_hierarchy
```

Results



Notes about path columns

PATH DISTANCE	RECORD STATE	PATH STATUS	PATH INFO WITH VEEVA IDS	PATH INFO
1	Valid	Active	242976927462196225|242979624307393536	University Of Utah Hospital | University Of Utah Health
1	Valid	Active	242976927470584833|242979649020232705	East Lakeland Obstetrics And Gynecology Associates PA | Baptist Memorial Healthcare Corporation
2	Valid	Active	242976927739020289|242987314630886407|242979566124008448	Austin Anesthesiology Group LLP | Seton Healthcare Admin | Ascension Health
1	Valid	Active	242976927806129152|732073085628421127	Uf Health Shands Hospital | University Of Florida Health
1	Valid	Active	242976927864849408|242979701709079552	UPMC Pinnacle Lancaster | UPMC Pinnacle

- **Path Distance** - Indicates the level that the entity is from the parent HCO. For example, a value of `1` means that the entity is directly connected to the parent HCO. A value of `2` means that there is another HCO between the entity and the parent HCO.
- **Path Info with Veeva IDs** - Displays the Veeva IDs of the entities in the path separated by the pipe (|) character. The entity's ID displays first and then the parent HCO ID displays.
- **Path Info** - Displays the names of the entities in the path separated by the pipe (|) character.

Example 2 - Query an entire health system

Report on all of the HCPs and HCOs in a specific health system.

Note: Previously, this data would require a SQL query that joined an average of five different tables.

Query

In this example, we are reporting on all of the HCPs and HCOs in the Ascension Health system.

```
SELECT
        *
    FROM
        flat_hierarchy
    WHERE
        ancestor_vid__v = 242979566124008448
```

Results

Example 3 - Hierarchy levels and counts for a health system

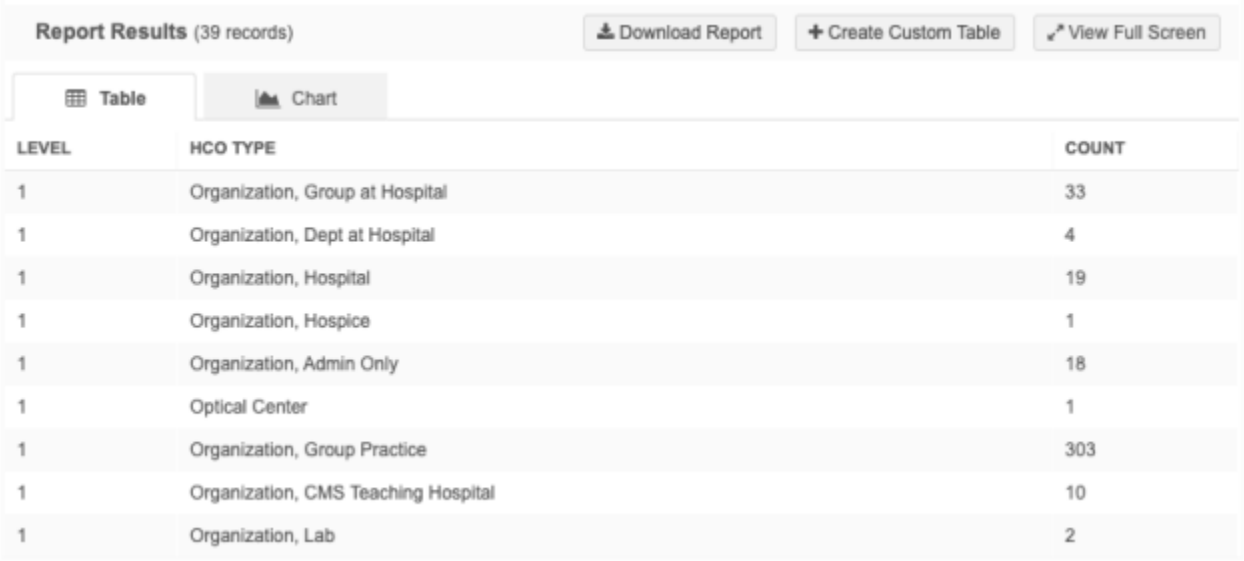
Use this query to report on the levels and counts of HCOs in a health system.

Query

This query joins the `flat_hierarchy` table to the `hco` table.

```
SELECT
        path_distance AS "Level",
        hco_type__v,
        COUNT (*)
    FROM
        flat_hierarchy LEFT JOIN hco
            ON entity_vid__v = hco.vid__v
    WHERE
        ancestor_vid__v = 242979566124008448
    GROUP BY
        hco_type__v,
        "Level"
    ORDER BY
        "Level" ASC
```

Results



LEVEL	HCO TYPE	COUNT
1	Organization, Group at Hospital	33
1	Organization, Dept at Hospital	4
1	Organization, Hospital	19
1	Organization, Hospice	1
1	Organization, Admin Only	18
1	Optical Center	1
1	Organization, Group Practice	303
1	Organization, CMS Teaching Hospital	10
1	Organization, Lab	2

Example 4 - Health system for HCPs

Use this query to find the health system where HCPs are affiliated.

Query

This query joins the `flat_hierarchy` table to the `hco` table.

Note: This query is difficult to create without the `flat_hierarchy` table; it requires joining on several tables.

```
SELECT
          entity_vid__v,
          LISTAGG (
              DISTINCT corporate_name__v,
              '| '
          ) AS "Health Systems"
      FROM
          flat_hierarchy JOIN hco
              ON ancestor_vid__v = hco.vid__v
      WHERE
          hco_type__v = '4:37'
          AND entity_type__v = 'HCP'
      GROUP BY
          entity_vid__v
```

Results

VEEVA ID OF OWNER	HEALTH SYSTEMS
242979996023391235	Wellspan Health System
242980399465104386	Inova Health System
242980450635613186	Ascension Health
242980538380452875	University Of Michigan Hospital Health System
242980634186744835	Unc Healthcare
242980778579854338	Ascension Health
242981022856119303	Jefferson Health System
242981099377001474	Banner Health System
242981539023946766	Baylor Scott And White Health

Example 5 - Rollup counts for HCPs and HCOs

Use this query to display roll up counts for health systems for a specific place and therapeutic area.

Query

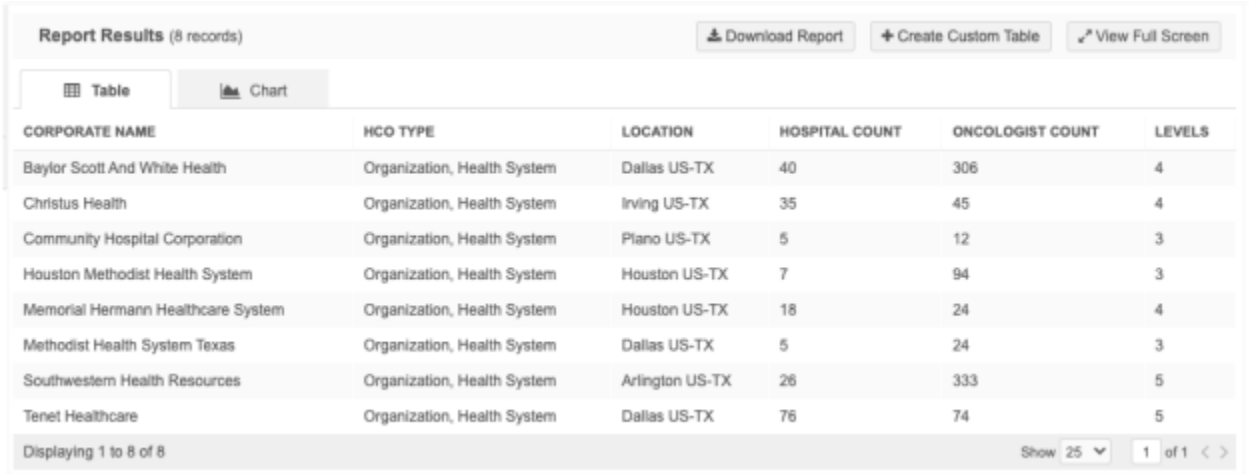
In this example, we want to return the rollup counts for health systems in Texas that have HCPs that are MDs and that specialize in oncology.

This query joins the `flat_hierarchy` table to the `hco` and `hcp` tables. The query is more complex but it is much easier to do now than it previously was using the `parenthco` table.

```
SELECT
         hco_anc.corporate_name__v,
         hco_anc.hco_type__v,
         locality__v || ' ' || administrative_area__v AS "location",
         SUM (
             CASE
                 WHEN hco_ent.major_class_of_trade__v = '32'
                 THEN 1
                 ELSE 0
             END
         ) AS "Hospital Count",
         SUM (
             CASE
                 WHEN hcp.primary_specialty_group__v = 'G-ON'
                 AND medical_degrees IN (
                     'MD',
                     'DO'
                 )
                 THEN 1
                 ELSE 0
             END
```

```
        ) AS "Oncologist Count",
        MAX( path_distance ) AS "Levels"
    FROM
        flat_hierarchy JOIN hco hco_anc
            ON hco_anc.vid__v = ancestor_vid__v JOIN address
            ON ancestor_vid__v = address.entity_vid__v
        AND address_ordinal__v = 1 LEFT JOIN hcp
            ON hcp.vid__v = flat_hierarchy.entity_vid__v LEFT JOIN hco
hco_ent
            ON hco_ent.vid__v = flat_hierarchy.entity_vid__v
    WHERE
        hco_anc.hco_type__v = '4:37'
        AND administrative_area__v = 'US-TX'
    GROUP BY
        hco_anc.corporate_name__v,
        "location",
        hco_anc.hco_type__v
    HAVING
        "Oncologist Count" > 5
        AND "Hospital Count" > 1
    ORDER BY
        hco_anc.corporate_name__v
```

Results



CORPORATE NAME	HCO TYPE	LOCATION	HOSPITAL COUNT	ONCOLOGIST COUNT	LEVELS
Baylor Scott And White Health	Organization, Health System	Dallas US-TX	40	306	4
Christus Health	Organization, Health System	Irving US-TX	35	45	4
Community Hospital Corporation	Organization, Health System	Plano US-TX	5	12	3
Houston Methodist Health System	Organization, Health System	Houston US-TX	7	94	3
Memorial Hermann Healthcare System	Organization, Health System	Houston US-TX	18	24	4
Methodist Health System Texas	Organization, Health System	Dallas US-TX	5	24	3
Southwestern Health Resources	Organization, Health System	Arlington US-TX	26	333	5
Tenet Healthcare	Organization, Health System	Dallas US-TX	76	74	5

Example 6 - Skip hierarchy levels

Analytics teams may want to roll up sales and interactions to the closest hospital an HCP is affiliated to and then to the Health System the hospital is affiliated with.

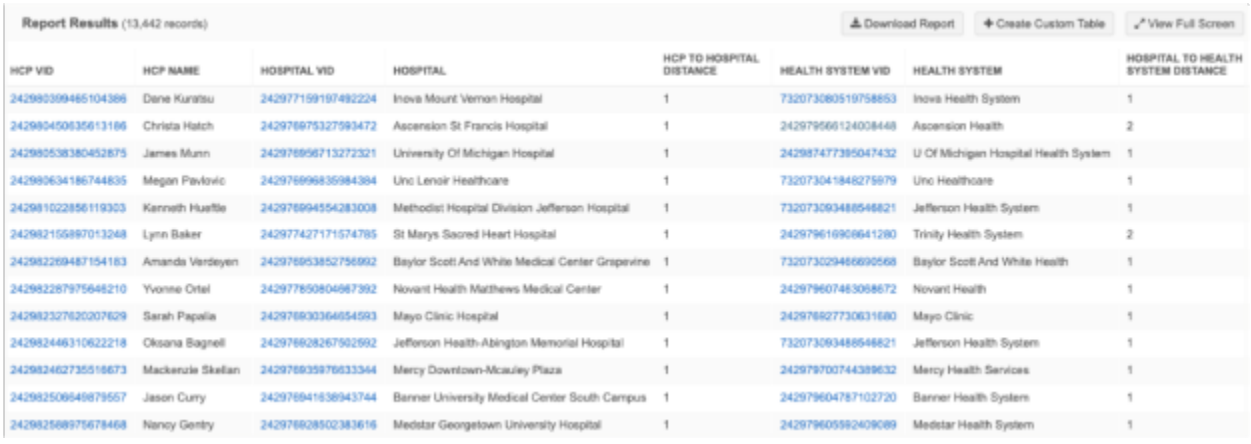
We can leverage the flat hierarchy to skip levels from the HCP to the hospital to the health system.

Query

This query involves a number of joins because we're pulling the names of the HCP, hospital, and health system. However, the number of joins on the relationships is reduced significantly.

```
SELECT
          flat_hospital.entity_vid__v as "HCP VID",
          formatted_name__v as "HCP Name",
          flat_hospital.ancestor_vid__v as "Hospital VID",
          hospital.corporate_name__v as "Hospital",
          flat_hospital.path_distance as "HCP to Hospital Distance",
          flat_healthsys.ancestor_vid__v as "Health System VID",
          healthsystem.corporate_name__v as "Health System",
          flat_healthsys.path_distance as "Hospital to Health System
Distance"
     FROM
          flat_hierarchy flat_hospital JOIN hco hospital
              ON hospital.vid__v = flat_hospital.ancestor_vid__v JOIN hcp
              ON flat_hospital.entity_vid__v = hcp.vid__v LEFT JOIN
flat_hierarchy flat_healthsys
              ON flat_healthsys.entity_vid__v =
flat_hospital.ancestor_vid__v LEFT JOIN hco healthsystem
              ON healthsystem.vid__v = flat_healthsys.ancestor_vid__v
     WHERE
          flat_hospital.entity_type__v = 'HCP'
          AND hospital.hco_type__v IN (
              '4:6',
              '4:35',
              '32_23',
              '32_22',
              '1_7',
              '1_3',
              '32_11',
              '4_56',
              '1_21',
              '1_10',
              '1_1'
          )
          and healthsystem.hco_type__v = '4:37'
```

Results



Report Results (13,442 records) — Download Report | + Create Custom Table | View Full Screen

HCP VID	HCP NAME	HOSPITAL VID	HOSPITAL	HCP TO HOSPITAL DISTANCE	HEALTH SYSTEM VID	HEALTH SYSTEM	HOSPITAL TO HEALTH SYSTEM DISTANCE
242980399465104386	Dane Kuratsu	242977159197492224	Inova Mount Vernon Hospital	1	732073080519758853	Inova Health System	1
242980450635613186	Christa Hatch	242976975327593472	Ascension St Francis Hospital	1	242979566124008448	Ascension Health	2
242980538380452875	James Munn	242976956713272321	University Of Michigan Hospital	1	242987477395047432	U Of Michigan Hospital Health System	1
242980634186744835	Megan Pavlovic	242976996836984384	Unc Lenoir Healthcare	1	732073041848275879	Unc Healthcare	1
242981022856119303	Kenneth Huefile	242976994554283008	Methodist Hospital Division Jefferson Hospital	1	732073093488546821	Jefferson Health System	1
242982155897013248	Lynn Baker	242977427171574785	St Marys Sacred Heart Hospital	1	242979619908641280	Trinity Health System	2
242982269487154183	Amanda Verdeyen	242976953852756992	Baylor Scott And White Medical Center Grapevine	1	732073029466890568	Baylor Scott And White Health	1
242982287975648210	Yvonne Ortel	242977650804967392	Novant Health Matthews Medical Center	1	242979607483068672	Novant Health	1
242982327620207629	Sarah Papalia	242976930364654593	Mayo Clinic Hospital	1	242976827730631680	Mayo Clinic	1
242982446310622218	Oksana Bagnell	242976928267502592	Jefferson Health-Abington Memorial Hospital	1	732073093488546821	Jefferson Health System	1
242982462735518673	Mackenzie Skellan	242976893976633344	Mercy Downtown-Mcauley Plaza	1	242979700744389632	Mercy Health Services	1
242982509649879557	Jason Curry	242976941636943744	Banner University Medical Center South Campus	1	242979604787102720	Banner Health System	1
242982568975678468	Nancy Gentry	242976928502383616	Medstar Georgetown University Hospital	1	242979605592409089	Medstar Health System	1

Customizations

Flattened hierarchies can be customized to flatten certain relationships only. For example, instead of flattening all active parentHCO relationships, you can flatten the “Ownership Hierarchy” (relationships where the relationship type is either `Affiliation` or `Ownership`).

To enable a customization on the flattened hierarchy, contact Veeva Support.

In the support ticket, provide the filter you want applied on the parentHCO. For example, for the "Ownership Hierarchy", submit the following filter:

```
relationship_type__v in ('7356','2')
```

Data quality

DATA VALIDATION RULES

22R2

Data Validation Rules have been renamed to Profile Validation Rules. Validation rules for data loading jobs are introduced in version 22R2.0. This name change helps to differentiate the two types of validation rules that Network now provides.

- **Profile Validation Rules** - Apply to individual records and are used as data quality checks; for example, to ensure that mandatory fields are populated or that HCPs have an address.
- **Job Validation Rules** - Apply to data loading jobs to prevent unintended changes to large numbers of records.

VALIDATING SUSPECT MATCH DATA

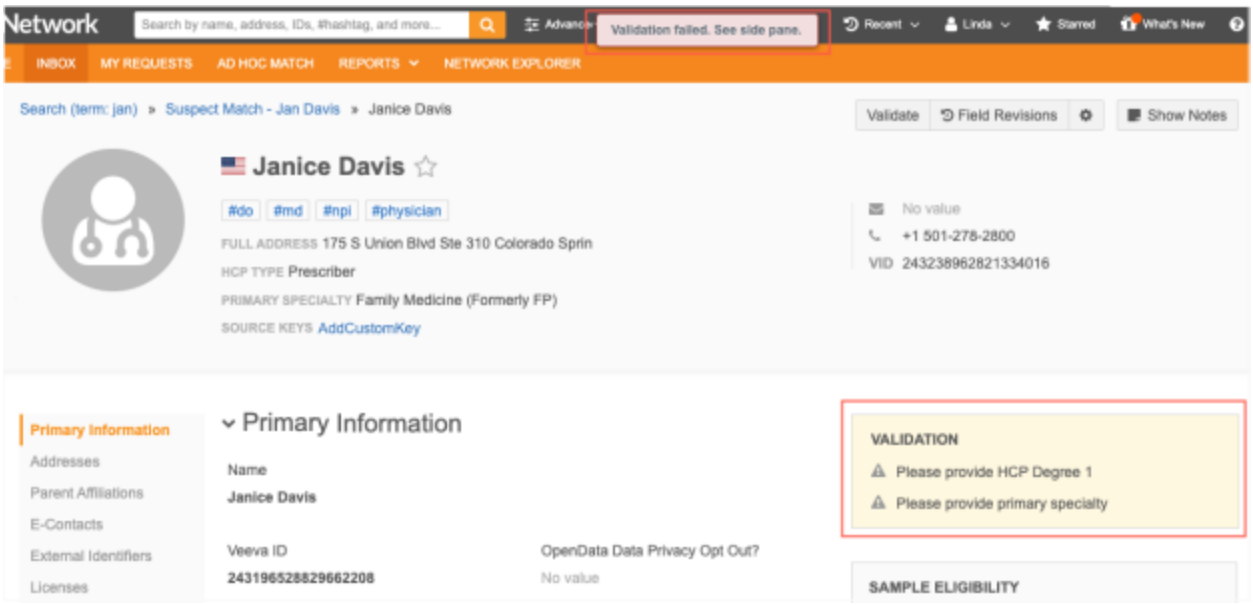
22R2

Administrators can configure profile validation rules (previously called *data validation rules*) to run immediately after suspect match tasks are processed. Data Stewards no longer need to click **Validate** on the profile to run the validation rules.

This feature is not enabled by default.

How it works

After Data Stewards apply a suspect match task, the profile page opens to display the surviving record and profile validation rules will run. If any of the validation rules fail, the **Validation** box displays the issues. If no validation rules fail, a success message displays.



Note: If Data Stewards navigate from the profile page without resolving the validation issues, it does not prevent the merge. The records are merged at this point, but the data issues will remain until they are resolved.

Enable the feature

Administrators can enable this enhancement for their Network instance.

1. In the Admin console, click **Settings > General Settings**.
2. Select **Edit**.
3. In the **Application Setting** section, under **Validation Rules**, select the **Automatically run profile validation rules after a suspect match task is processed**.
4. Save your changes.

Whenever suspect matches are processed in your Network instance, the applicable profile validation rules will run.

JOB VALIDATION RULES

22R2

Administrators can create and manage data validation rules for source subscription and data updater jobs. Job validation rules will fail the data loading job if critical value changes try to occur on a large number of records.

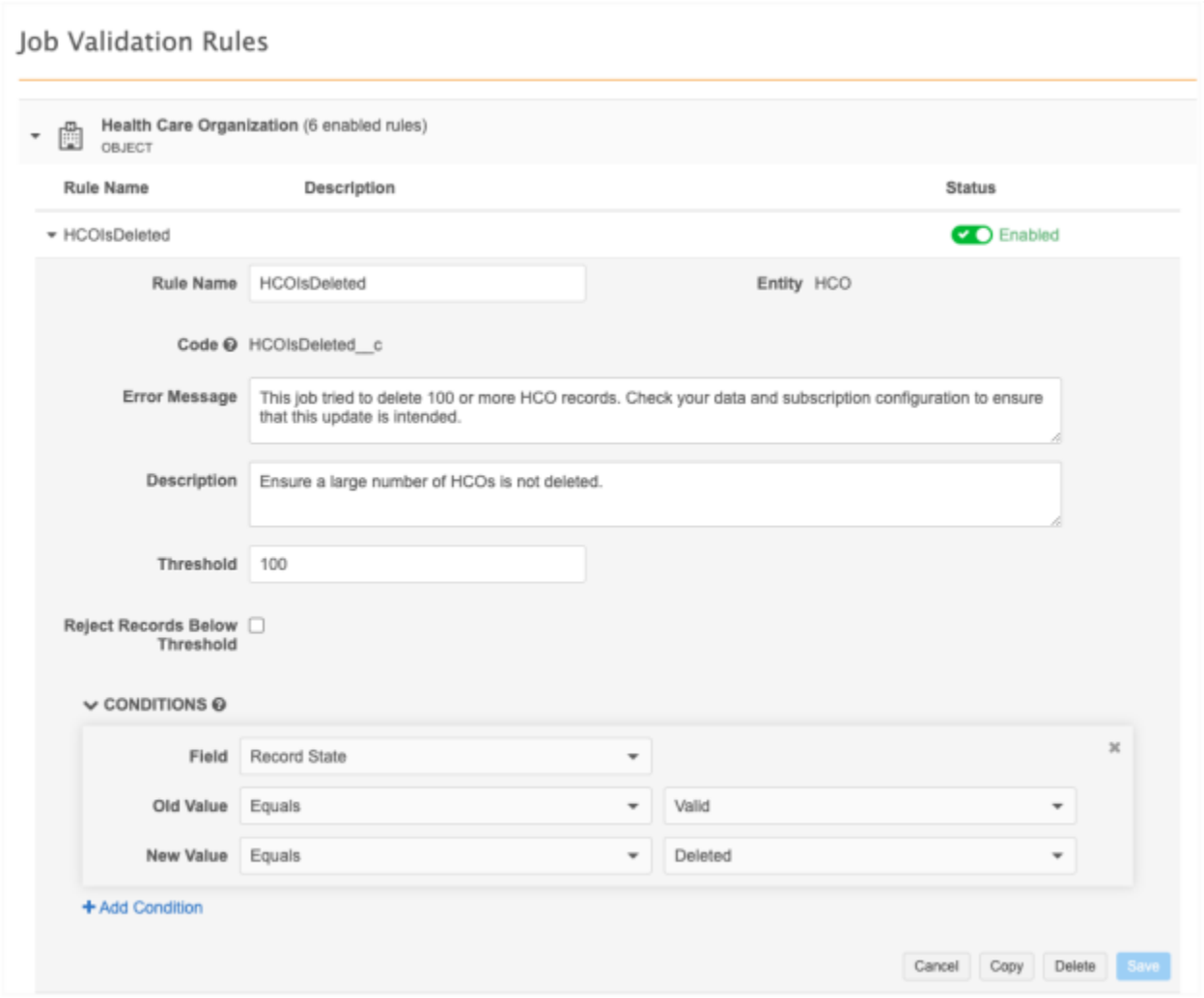
Examples of critical value changes

- updating mandatory name fields to NULL
- deleting, invalidating, and inactivating records
- opting out records

If a validation rule is set and a defined number of records violate the rule, the job will fail before the data is loaded and these changes take effect.

Note: Job validation rules reject certain changes for existing records; they do not ensure data quality for newly loaded records. Validation for newly added records is not supported.

Job Validation Rules apply to source subscriptions and Data Updater jobs.



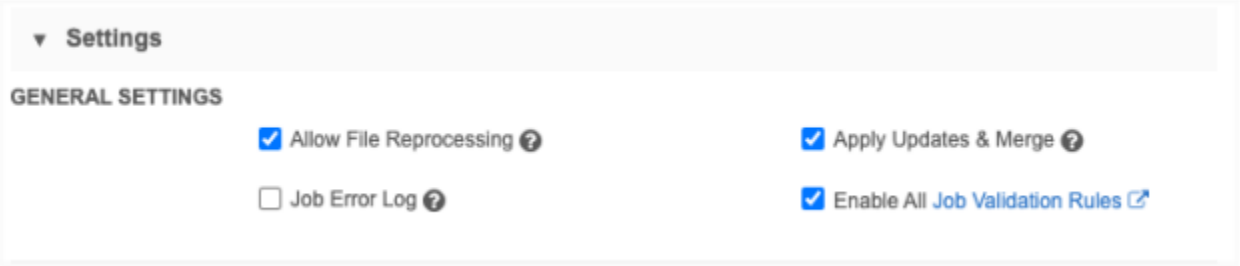
This feature is available by default in your Network instance.

A set of predefined rules are provided for Veeva standard objects. These rules are enabled by default but they do not impact existing data loading jobs until Administrators enable the rule settings for these jobs.

Enable job validation rules for source subscriptions

The job validation rules setting is enabled on new source subscriptions by default. The setting is not enabled on existing subscriptions to ensure the rules have no impact until Administrators choose to enable it.

Job validation rules are supported for source subscriptions that load data from all source systems, including third party data providers.



Important: Job validation rules run during the merge stage of the source subscription job, so the rules are only applied if the **Apply Updates & Merge** setting is also selected.

To set the option for an existing source subscription:

1. In the Admin console, click **System Interfaces > Source Subscription**.
2. Select an existing subscription.
3. In the **General Settings** section, select **Enable All Job Validation Rules**.

   To review the rules, click the **Job Validation Rules** link to navigate to that page.

Job validation rules are applied if the source feed contains fields that are part of a rule.

Enable job validation rules for Data Updater jobs

Administrators can enable job validation rules to run on data updater jobs. By default, this option is not enabled so it does not impact data updater jobs.

1. In the Admin console, click **Settings > General Settings**.
2. Select **Edit**.
3. In the **Application Settings** section, select **Apply all Enabled Job Validation Rules to Data Updater**.
4. **Save** your changes.

The job validation rules are applied if fields defined in the rules are part of the data feed.

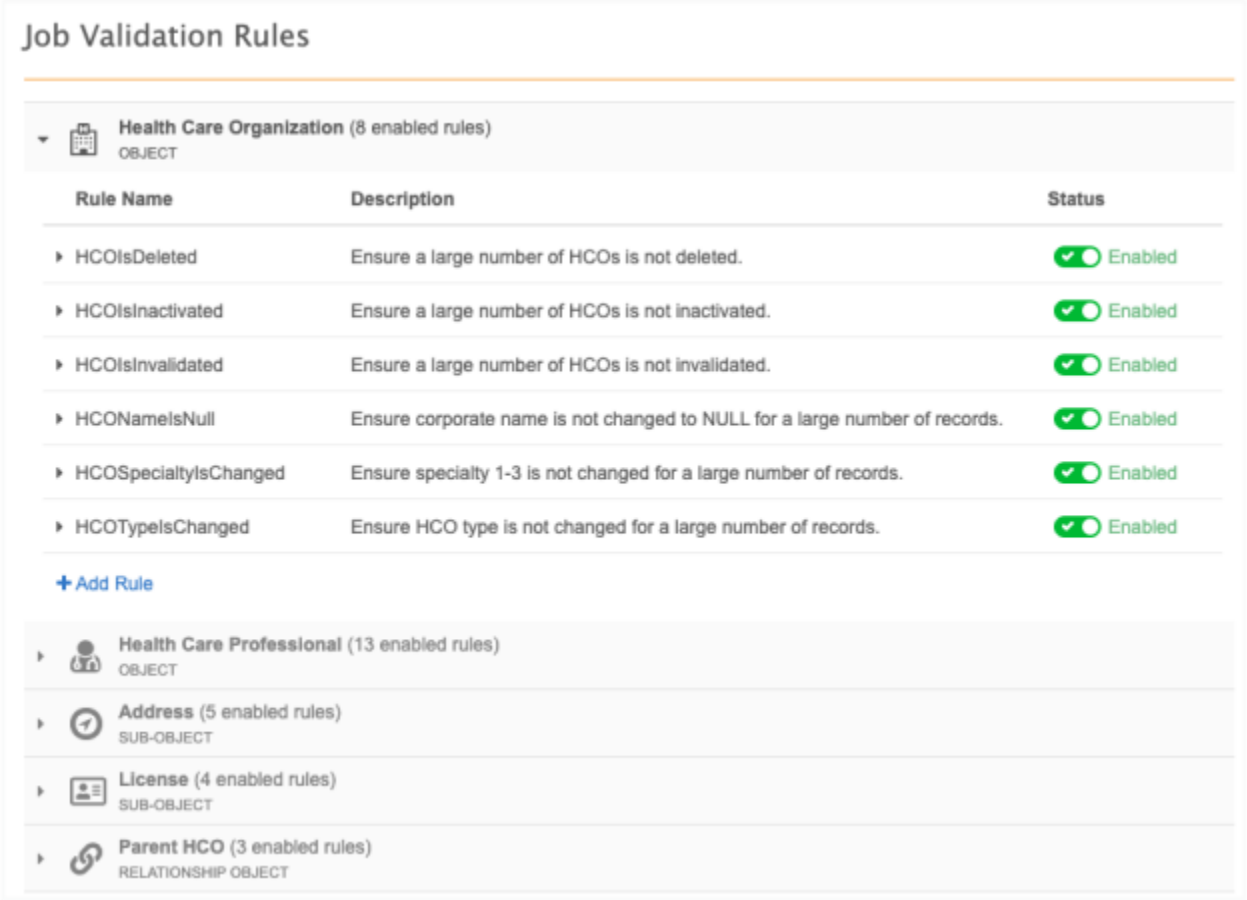
Job validation rules page

Network provides a set of predefined job validation rules for each Veeva standard object.

To view the rules in the Admin console, click **System Interfaces > Job Validation Rules**.

The page contains a section for each enabled object in your Network instance. The objects are listed alphabetically by main object and then sub-objects and relationship objects.

The predefined rules are enabled by default but they do not impact data loading jobs until they are enabled to run for a source subscription or they are enabled for data updater jobs. For a list of the provided rules, see the *Predefined rules* section below.



Creating validation rules

Create validation rules to prevent critical field values changes.

Rules apply to all countries. If you create a rule using a field that is not available in a country and you run a job for that country, the job validation rule is not applied.

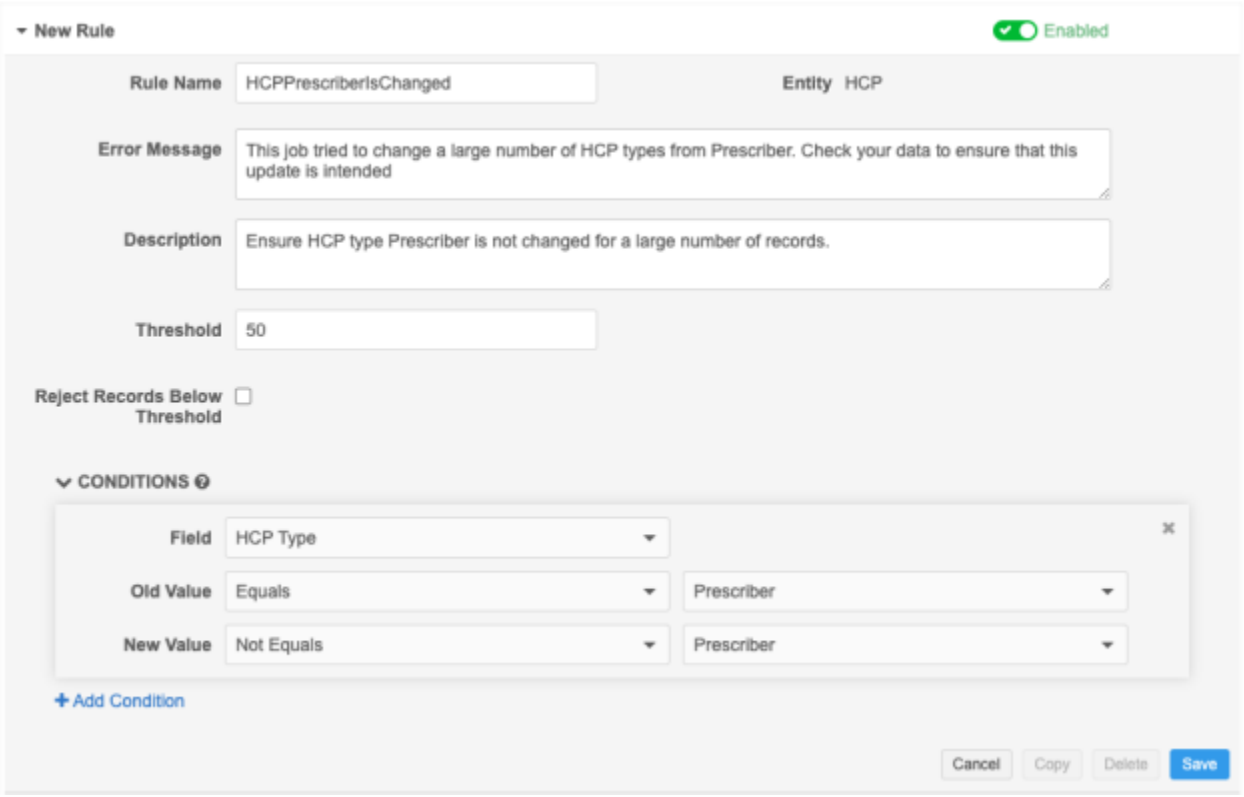
To create a rule:

1. In the Admin console, click **System Interfaces > Job Validation Rules**.
2. On the Job Validation Rules page, all objects that are enabled in your Network instance display. Expand an object and click **Add Rule**.

   Example

   Create a validation rule to ensure that a large number of HCP types are not changed from **Prescriber**.

   The rule will check that the `hcp_type__v` field value does not change from **Prescriber** to any value that is not **Prescriber**.



3. In the **New Rule** section, define the following settings:
   - **Rule Name** - Type the name of the rule. The name must be unique.
   - **Error Message** - Type a message that Administrators and Data Managers will see if the rule triggers the job to fail.
   - **Description** – Type a description of the rule. This displays on the Job Validation Rules page.
   - **Threshold** - Define the number of records to meet the rule criteria for the job to fail. The number must be between 1 and 20,000.
   - **Reject Records Below Threshold** - Select this option to reject any number of records below the threshold that meet the rule criteria. If this option is not set, records that meet the rule conditions below the threshold will be updated during the job.

     Tip: Select this option when the change will result in bad data quality; for example, if the update changes the HCP name to Null.

- **Conditions** - Identify the field and values as the criteria for this rule. For most rules, this is a comparison between the old (existing value) and the new (incoming) value.
  - **Field** - Expand the list and choose the field. Only the fields for the object display.
  - **Old Value** - The existing field value on the record. Expand the list to choose the operator (**Equals**, **Not Equals**, **Find**, and so on) and then type the field value.

    The available operators depend on the selected field.

    Not all operators required a value. For example, the **Is Changed**, **Is Null** and **Is Not Null** operators do not require values.
  - **New Value** - The incoming value of the field that will update the record. Expand the list to choose the condition and then type the field value.

  Supported operators

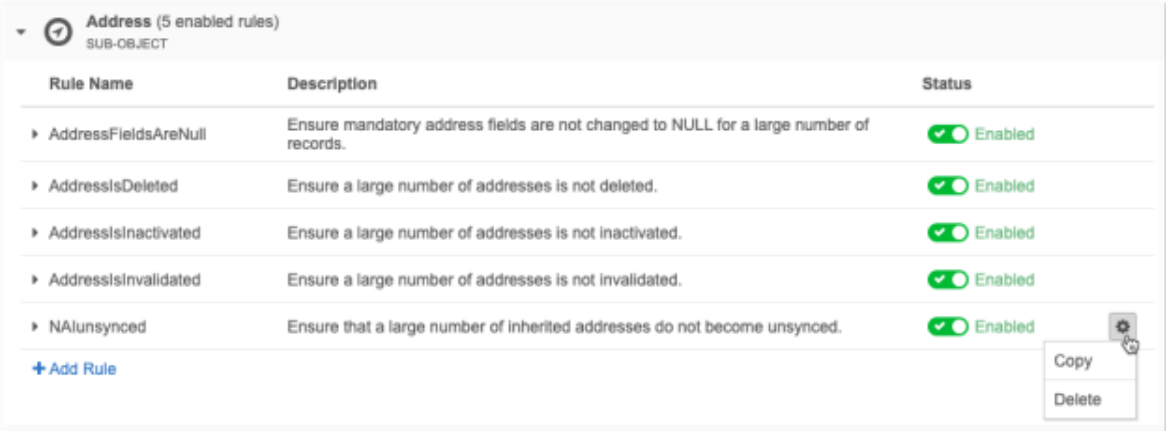
  The following operators are available, depending on the selected field type:

  - **Equals / Not Equals** - Values are not case-sensitive.
  - **Find / Not Find** - Use this operator for substring matching. Think of it as "contains / not contains". For example, if you specify the FIND operator value as "abc" then the rule is violated for all fields that contain the substring "abc" ("abcxyz", "xyzabcxyz", "xyzabc", and so on). Values are not case-sensitive.
  - **In / Not In** - Used for string and reference fields. It specifically looks for values in fields and it ignores blanks or nulls. Values are not case-sensitive.
  - **Is Null / Is Not Null** - Use to check when there is no value or when existing values are removed and are not replaced. This is helpful to ensure that records contain critical values (for example, HCP name).
  - **Is Changed** - Used to determine if a field value is changed. If this operator is used, the **New Value** is not required.
  - **Match Regular Expression / Not Match Regular Expression** - Can be used to check if the old or new value match a certain expression. (Network uses Java Regular Expressions). Values are case sensitive.
  - **Less Than / Greater Than** - Can be used for fields containing numeric values; for example, fields with rankings or dates.
  - **Between** - Can be used for fields containing numeric values; for example, fields with rankings or dates.
    This operator looks at the values between what's specified, not including what's specified. For example, if you provide the values of *1* and *3*, the rule will only fire/fail for a value of *2*. If you provide values of *1* and *2*, the rule will never fire because there is no value between *1* and *2*.

  Click **Add Condition** to define another field. Each rule can link multiple conditions using the AND or the OR operator. All conditions must be joined through the same operator; you cannot mix AND and OR within the same rule.

4. **Save** the rule.

The rule displays in the object section. It is enabled by default.



Edit rules

Predefined and custom rules can be edited; for example, you can raise or lower the threshold. All of the rule properties can be changed except the object and the **Code**. The **Code** is used for exporting configurations to other Network environments.

Copy rules

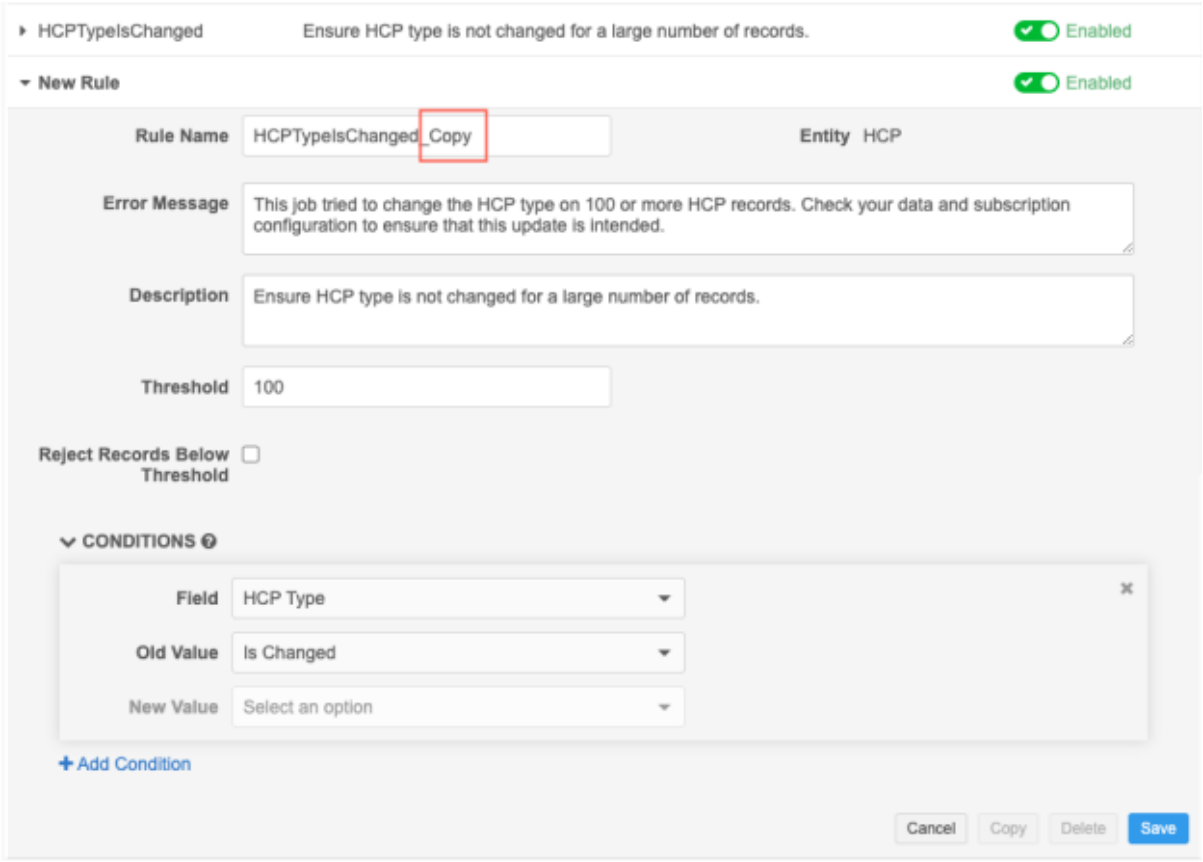
Copy a validation rule to create a similar rule.

To copy a rule:

1. Expand a rule and click **Copy**.

   You can also copy a rule from the row in the list view. Click **Options** **> Copy**
2. In the **Copy Rule** pop-up, click **Yes, Copy**.

   The copied rule opens so you can edit it. The **Rule Name** is appended with `_Copy`.



3. When you have finished editing the rule, select **Save**.

   The rule is enabled by default.

Delete rules

All predefined and custom rules can be deleted.

To delete a rule:

1. Expand a rule and click **Delete**.

   You can also delete a rule from the row in the list view. Click **Options** > **Delete**.

2. In the **Delete Job Validation Rule** pop-up, click **Yes, Delete**.

   The rule is removed from your Network instance.

Triggered validation rules

Validation rules run on source subscription and data updater jobs if the source feed or file contains the fields defined in the rule conditions.

The job outcome depends on the configuration of any triggered rules.

Failed jobs

A rule is violated and the job fails if the number of records that meet that rule condition is equal or higher than the defined threshold. For example, if the rule threshold is 100 and the job tries to change a critical value for 100 or more records, the job fails and no records are updated.

Rejected records

If a rule threshold is not met and the job completes, individual records that violate the rule might be rejected depending on the **Reject Records Below Threshold** rule setting. Each rule contains this setting so you can determine the number of acceptable records to be updated.

Example

A rule threshold is 100 records and 3 records met the rule conditions.

- **Reject Records Below Threshold** option is not set - The 3 records are updated by the completed job.
- **Reject Records Below Threshold** option is set - The 3 records are rejected by the completed job.

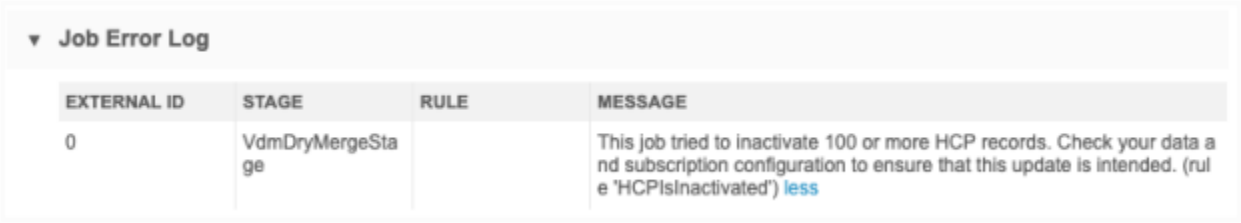
Job errors

Errors are logged when a job fails because of a validation rule and when records are rejected because of a validation rule.

Source subscription job error log

The source subscription error log contains entries for the following issues:

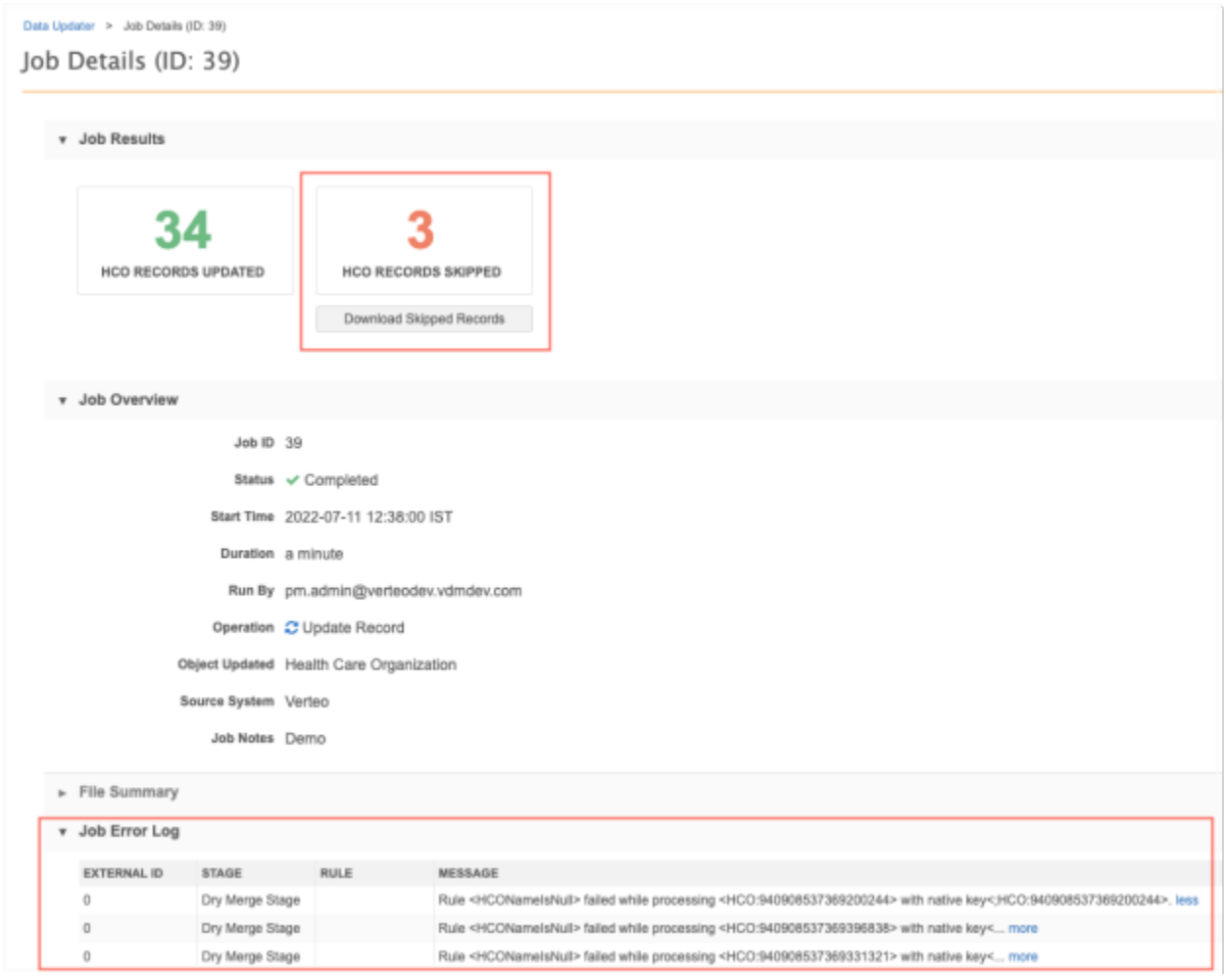
- Job fails because a validation rule threshold was met. The log entry contains the rule name and error message defined in the rule.



- Job completes but records were rejected because the validation rule was configured to reject records below the rule threshold. The error log contains the rule and the record that the job tried to change so you can investigate the update.

Data updater jobs

For data updater jobs, if the job completes but a number of records below the threshold were rejected, they will not be updated and will display as **Skipped Records** on the Job Details page. The **Job Error Log** will display the job validation rule that failed.



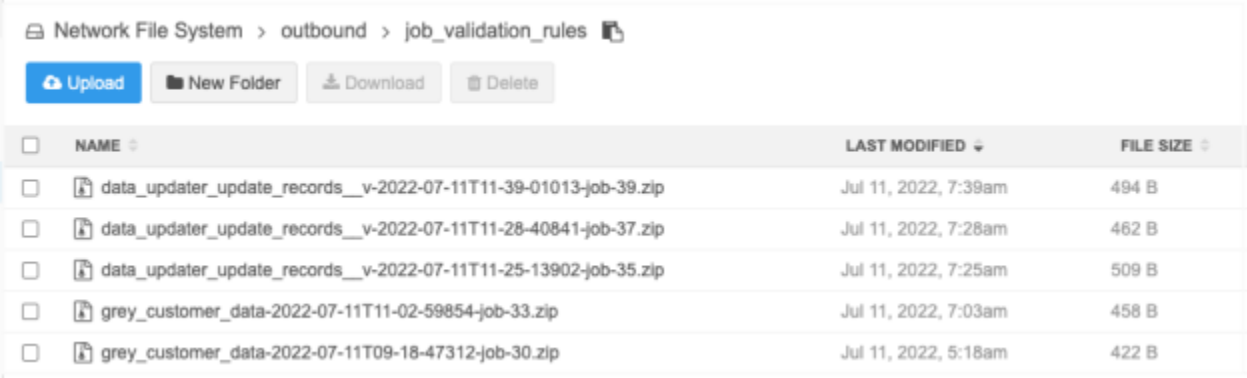
Job validation error log

All records that met *any* rule conditions are logged in a separate log file so you can investigate the records that caused the job rule violations. This includes records that violated rules without hitting the rule thresholds (records that were updated). The logged records stop near the defined threshold; for example, if the threshold is 100, the log might contain 100 or more records.

The log file is located in the `outbound/job_validation_rules` folder. All job validation rule logs for source subscriptions and data updater jobs are contained in this folder.

Log file naming convention:

- Source subscription jobs - `<subscription_name>-<timestamp>-job-<job_ID>.zip`
- Data updater jobs - `data_updater_update_records__v-<timestamp>-job-<job_ID>.zip`



Download the file to review information about the records and rules that were violated.

Example log file

This job failed because a large number of HCPs were inactivated. The log file displays all of the critical changes that the job made based on the rules that were triggered.

This log displays the records that violated the `HCPIsInactivated` rule; this is the rule that caused the job to fail. It also displays some records that violated the `HCPTypeIsChanged` validation rule. These records did not trigger the job to fail because it was below the threshold, but they are still logged so you can see that a critical value would be changed.

grey_customer_data-2022-07-13T14-14-15461-job-43

VID	Type	Parent VID	Parent Type	Native Key	Rule Name	Rule ID	Old Value(s)	New Value(s)
940908537368086121	HCP	940908537368086121	HCP	HCP:463722913342686248	HCPTypeIsChanged	HCPTypeIsChanged__c	hcp_type__v:P	hcp_type__v:N
940908537368086113	HCP	940908537368086113	HCP	HCP:463722913132971009	HCPIsInactivated	HCPIsInactivated__c	hcp_status__v:A	hcp_status__v:I
940908537368086112	HCP	940908537368086112	HCP	HCP:463722913065862145	HCPIsInactivated	HCPIsInactivated__c	hcp_status__v:A	hcp_status__v:I
940908537368086114	HCP	940908537368086114	HCP	HCP:463722913216857089	HCPIsInactivated	HCPIsInactivated__c	hcp_status__v:A	hcp_status__v:I
940908537368086118	HCP	940908537368086118	HCP	HCP:463722913300743174	HCPTypeIsChanged	HCPTypeIsChanged__c	hcp_type__v:P	hcp_type__v:N
940908537368086116	HCP	940908537368086116	HCP	HCP:463722913258800129	HCPIsInactivated	HCPIsInactivated__c	hcp_status__v:A	hcp_status__v:I
940908537368086115	HCP	940908537368086115	HCP	HCP:463722913216857095	HCPIsInactivated	HCPIsInactivated__c	hcp_status__v:A	hcp_status__v:I

The log files contain the following columns:

- **VID** - The Veeva ID.
- **Type** - The object type.
- **Parent VID** - Veeva ID of the parent object.
- **Parent Type** - Type of parent object.
- **Native Key** - If the source file contains a native key, it displays so you can identify which incoming record experienced the error.
- **Rule Name** - The job validation rule name.
- **Rule ID** - The rule code (automatically defined by Network)
- **Old Value** - The existing field value on the record.
- **New Value** - The new incoming value from the source feed or file.

Predefined rules

These following validation rules are available in your Network instance. They are enabled by default. You can edit these rules to customize them for your requirements.

HCO rules (6 rules)

Name	Description	Condition	Error	Default Threshold	Reject Records?
HCOIsDeleted	Ensure a large number of HCOs are not deleted.	The record_state__v field is changed from VALID to DELETED.	This job tried to delete 100 or more HCO records. Check your data and subscription configuration to ensure that this update is intended.	100	No
HCOIs Inactivated	Ensure a large number of HCOs are not inactivated.	The hco_status__v field is changed from ACTIVE.	This job tried to inactivate 100 or more HCO records. Check your data and subscription configuration to ensure that this update is intended.	100	No
HCOIs Invalidated	Ensure a large number of HCOs are not invalidated.	The record_state__v HCO field is changed from VALID to INVALID.	This job tried to invalidate 100 or more HCO records. Check your data and subscription configuration to ensure that this update is intended.	100	No

Name	Description	Condition	Error	Default Threshold	Reject Records?
HCONameIsNull	Ensure corporate name is not changed to NULL for a large number of records.	Existing value in the corporate_name__v field is changed to NULL.	This job tried to set corporate name to null on 100 or more HCO records. Check your data and subscription configuration to ensure that this update is intended.	100	Yes
HCOSpecialtyIsChanged	Ensure specialty 1-3 is not changed for a large number of records.	Existing value in any of the following fields is changed: speciality_1__v speciality_2__v speciality_3__v	This job tried to change the specialty 1-3 on 100 or more HCO records. Check your data and subscription configuration to ensure that this update is intended.	100	No
HCOTypeIsChanged	Ensure HCO type is not changed for a large number of records.	The hco_type__v field is changed.	This job tried to change the HCO type on 100 or more HCO records. Check your data and subscription configuration to ensure that this update is intended.	100	No

HCP rules (7 rules)

Name	Description	Condition	Error	Default Threshold	Reject Records?
HCPIsDeleted	Ensure a large number of HCPs are not deleted.	The record_state__v HCP field is changed from VALID to DELETED.	This job tried to delete 100 or more HCP records. Check your data and subscription configuration to ensure that this update is intended.	100	No

Name	Description	Condition	Error	Default Threshold	Reject Records?
HCPIs Inactivated	Ensure a large number of HCPs are not inactivated.	The hcp_status__v is changed from ACTIVE.	This job tried to inactivate 100 or more HCP records. Check your data and subscription configuration to ensure that this update is intended.	100	No
HCPIs Invalidated	Ensure a large number of HCPs are not invalidated.	The record_state__v HCP field is changed from VALID to INVALID.	This job tried to invalidate 100 or more HCP records. Check your data and subscription configuration to ensure that this update is intended.	100	No
HCPIsOpted Out	Ensure a large number of HCPs are not opted out.	The data_privacy_opt_out__v field is changed from N to Y.	This job tried to opt out 100 or more HCP records. Check your data and subscription configuration to ensure that this update is intended.	100	No
HCPNameIs Null	Ensure first name or last name is not changed to NULL for a large number of records.	Existing value in the first_name__v or last_name__v fields is changed to NULL.	This job tried to set first name or last name to Null on 100 or more HCP records. Check your data and subscription configuration to ensure that this update is intended.	100	Yes
HCPSpecialty IsChanged	Ensure specialty 1-3 is not changed for a large number of records.	Any of the following fields are changed: speciality_1__v speciality_2__v speciality_3__v	This job tried to change the specialty 1-3 on 100 or more HCP records. Check your data and subscription configuration to ensure that this update is intended.	100	No

Name	Description	Condition	Error	Default Threshold	Reject Records?
HCPTypeIsChanged	Ensure HCP type is not changed for a large number of records.	The hcp_type__v field is changed.	This job tried to change the HCP type on 100 or more HCP records. Check your data and subscription configuration to ensure that this update is intended.	100	No

Address rules (4 rules)

Name	Description	Condition	Error	Default Threshold	Reject Records?
AddressFieldsAreNull	Ensure mandatory address fields are not changed to NULL for a large number of records.	Existing values in the following fields are changed to NULL: address_line_1__v postal_code__v locality__v country__v	This job tried to set mandatory address fields to null on 100 or more address records. Check your data and subscription configuration to ensure that this update is intended.	100	Yes
AddressIsDeleted	Ensure a large number of addresses are not deleted.	The record_state__v field is changed from VALID to DELETED.	This job tried to delete 100 or more address records. Check your data and subscription configuration to ensure that this update is intended.	100	No
AddressIsInactivated	Ensure a large number of addresses are not inactivated.	The address_status__v field is changed from ACTIVE.	This job tried to inactivate 100 or more address records. Check your data and subscription configuration to ensure that this update is intended.	100	No

Name	Description	Condition	Error	Default Threshold	Reject Records?
AddressIs Invalidated	Ensure a large number of addresses are not invalidated.	The record_state__v address field is changed from VALID to INVALID.	This job tried to invalidate 100 or more address records. Check your data and subscription configuration to ensure that this update is intended.	100	No

License rules (4 rules)

Name	Description	Condition	Error	Default Threshold	Reject Records?
LicenseIs Deleted	Ensure a large number of licenses are not deleted.	The record_state__v field is changed from VALID to DELETED.	This job tried to delete 100 or more license records. Check your data and subscription configuration to ensure that this update is intended.	100	No
LicenseIs Inactivated	Ensure a large number of licenses are not inactivated.	The license_status__v field is changed from ACTIVE.	This job tried to inactivate 100 or more license records. Check your data and subscription configuration to ensure that this update is intended.	100	No
LicenseIs Invalidated	Ensure a large number of licenses are not invalidated.	The record_state__v license field is changed from VALID to INVALID.	This job tried to invalidate 100 or more license records. Check your data and subscription configuration to ensure that this update is intended.	100	No

Name	Description	Condition	Error	Default Threshold	Reject Records?
LicenseIs Null	Ensure license number is not changed to NULL for a large number of records.	An existing license number is changed to NULL.	This job tried to set license number to null on 100 or more license records. Check your data and subscription configuration to ensure that this update is intended.	100	Yes

ParentHCO rules (4 rules)

Name	Description	Condition	Error	Default Threshold	Reject Records?
ParentHCOIs Deleted	Ensure a large number of parentHCOs are not deleted.	The record_state__v field is changed from VALID to DELETED.	This job tried to delete 100 or more parentHCO records. Check your data and subscription configuration to ensure that this update is intended.	100	No
ParentHCOIs Inactivated	Ensure a large number of parentHCOs are not inactivated.	The parent_hco_status __v field is changed from ACTIVE.	This job tried to inactivate 100 or more parentHCO records. Check your data and subscription configuration to ensure that this update is intended.	100	No
ParentHCOIs Invalidated	Ensure a large number of parentHCOs are not invalidated.	The record_state__v parentHCO field is changed from VALID to INVALID	This job tried to invalidate 100 or more parentHCO records. Check your data and subscription configuration to ensure that this update is intended.	100	No

Name	Description	Condition	Error	Default Threshold	Reject Records?
ParentHCO TypeIs Changed	Ensure relationship type is not changed for a large number of records.	The relationship_type__v field is changed.	This job tried to change the relationship type on 100 or more parentHCO records. Check your data and subscription configuration to ensure that this update is intended.	100	No

Bulk updates

If you are intentionally updating a large number of records, for example, you are inactivating more than 1000 addresses, you will not want the job validation rule to run.

There are three options to perform the bulk update:

- Increase the threshold of the job validation rule. For example, if there are 1000 addresses to be activated, temporarily increase the rule threshold to 2000. This will impact any jobs started by other users.
- Temporarily disable the individual job validation rule on the Job Validation Rule page. This will impact all other source subscription and data updater jobs that are running while the rule is disabled.
- Disable all job validation rules in the source subscription. This means that all other job validation rules that can prevent other critical value changes during the job are disabled as well.

All of these actions are tracked in the System Audit Log.

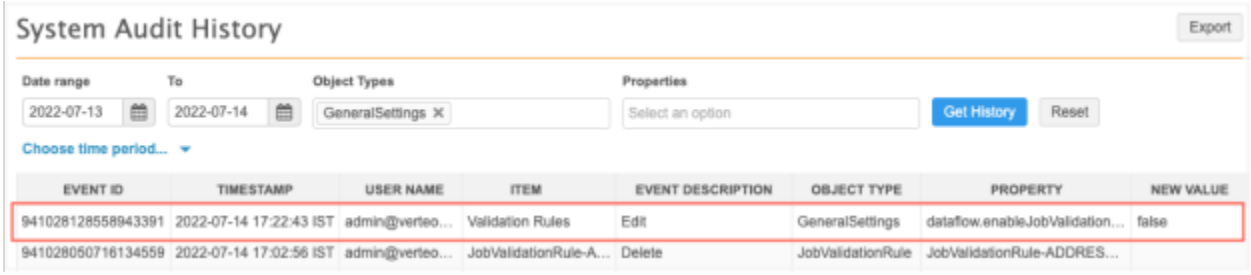
Logging

The System Audit Log contains any changes that are made to job validation rules.

The log tracks the following changes:

- Enable or disable the setting to apply job validation rules to Data Updater jobs.

  Search by the `GeneralSettings` **Object Types** to filter the log records.

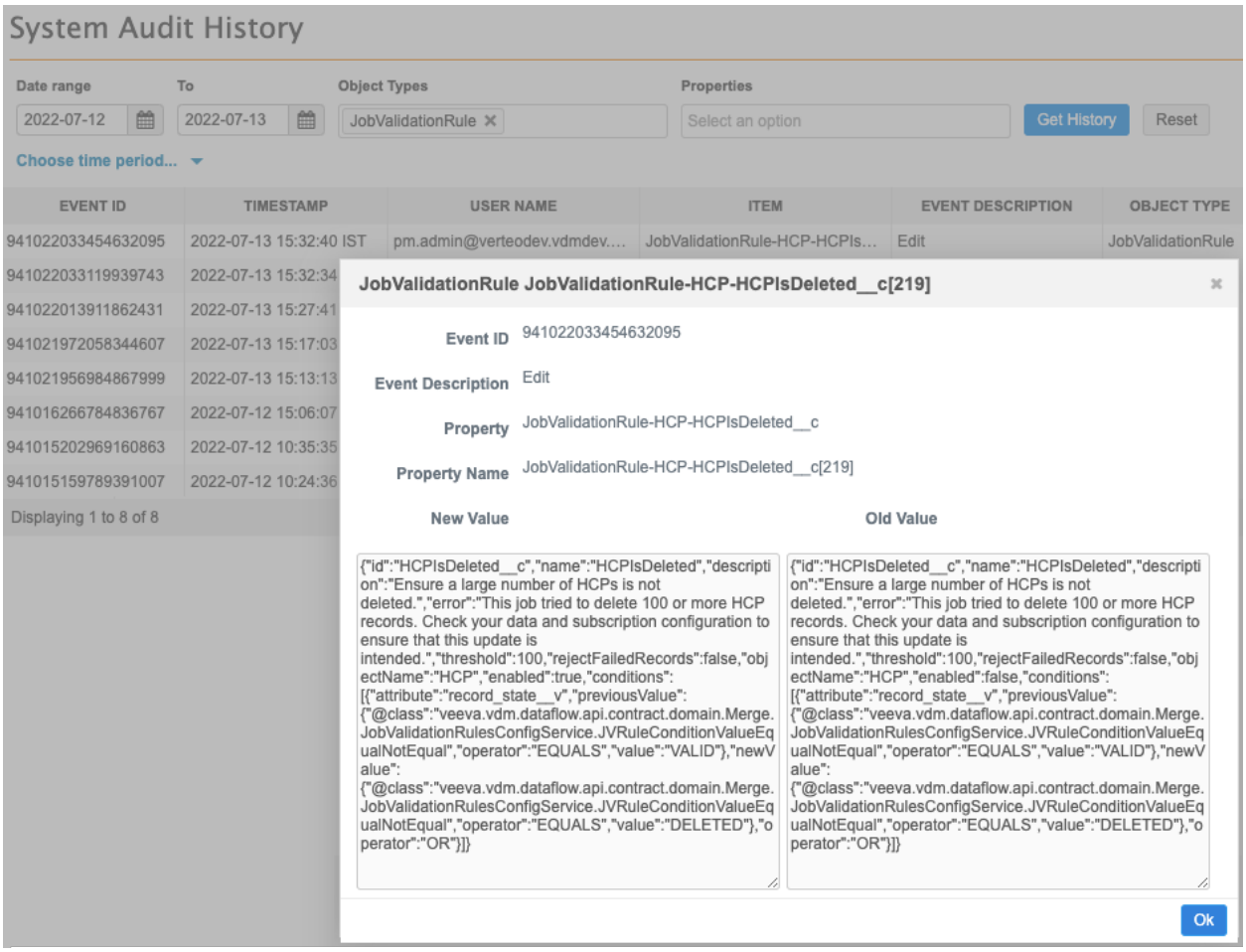


- Change job validation rule settings for a subscription.

  Search by the `Subscription` **Object Types** to filter the log records. Open the log to view the change.

- Create, edit, or delete job validation rules.

  Search for the `JobValidationRule` **Object Types** to filter the log records. Open the record to review the JSON so you can see the details of the change.



Exporting configurations

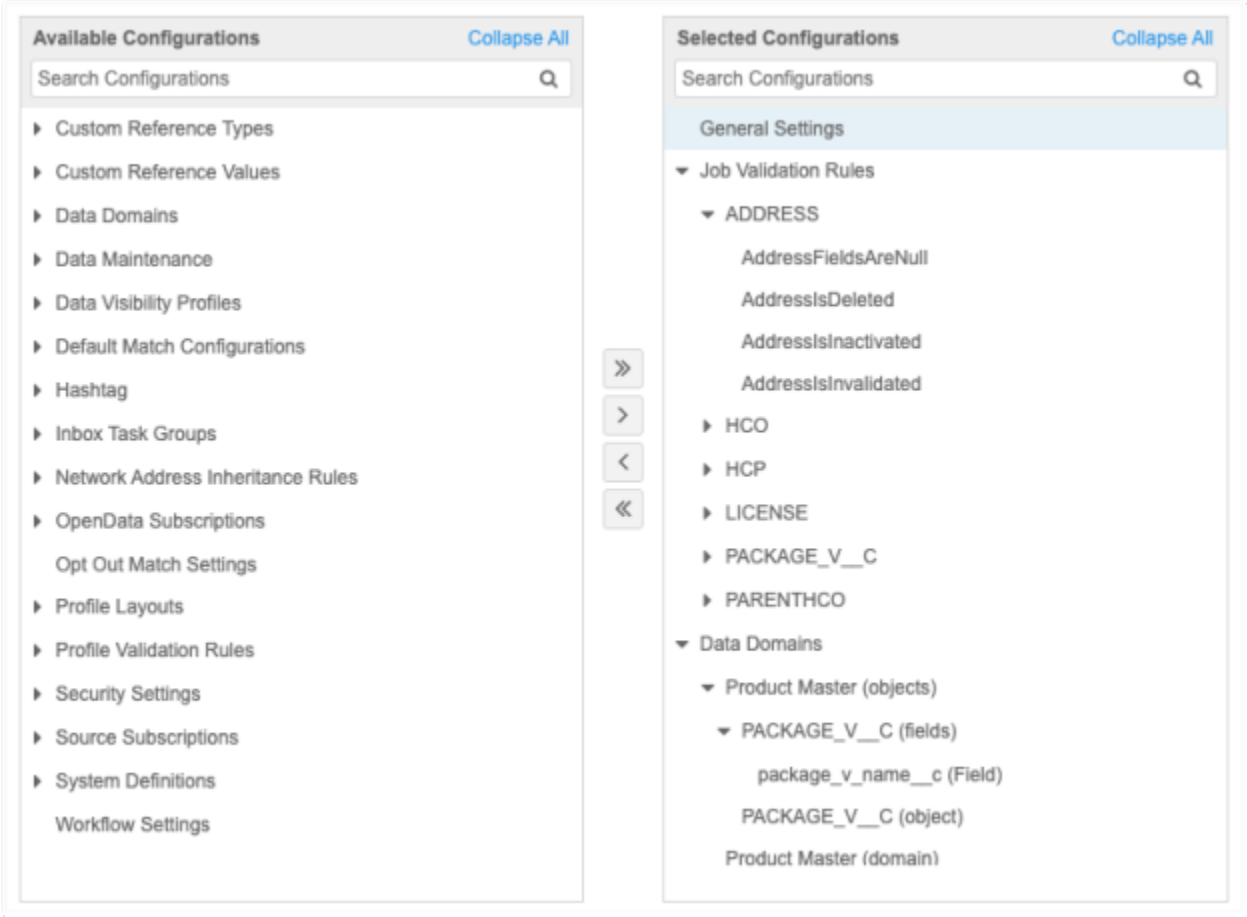
Administrators can export the job validation rules to a target environment using configuration packages.

All of the job data validation rules and the setting to enable job validation rules for the data updater can be exported.

- **Job validation rules** - Export one rule, all rules for an object, or the **Job Validation Rules** section to move all of the rules.

  If you include job validation rules for custom objects, the custom fields and objects for those rules are automatically added to the export package.

- **Data updater setting** - Select the **General Settings** configuration to export the setting.



For more information about exporting configurations, see the Managing configurations topic in the *Veeva Network Online Help*.

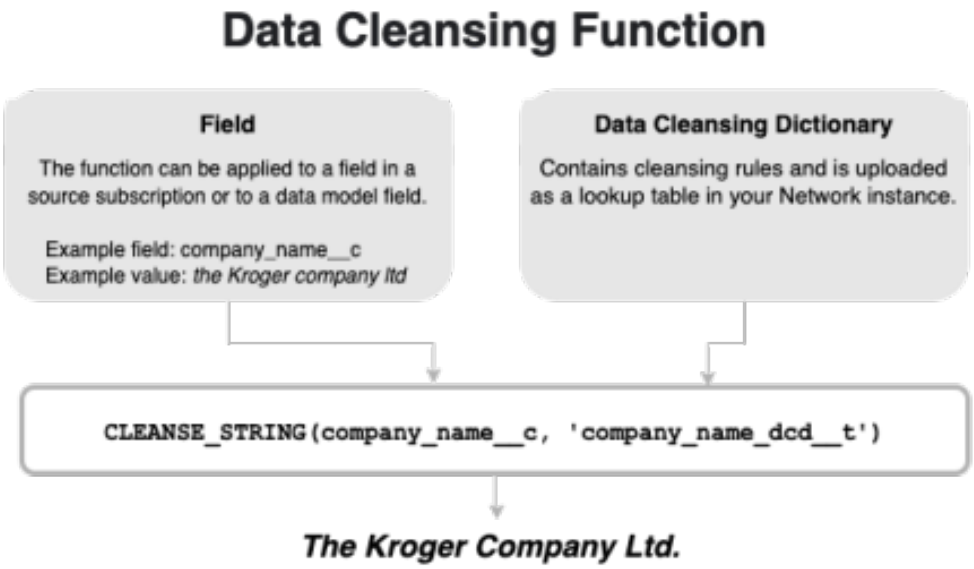
DATA CLEANSING

22R1.1

Administrators and data managers can now define rules to cleanse and standardize data in fields.

Addresses are cleansed using Network's third party cleansing tool, but previously, there was no efficient way to cleanse and standardize other fields. Now you can create a data cleansing dictionary to do things like remove noise characters and words and replace strings with better strings.

Use the data cleansing dictionary with a new Network expression function in a source subscription or a data model field to update specific field values.



This feature is available by default in your Network instance.

Standardize field values

Cleansing fields helps you to improve the values. This can be helpful for matching and to ensure high data quality in general.

Example use cases:

- **Standardize letter case** - Make the first character of each word uppercase (for example, cleanse **the Kroger company** as **The Kroger Company**).
- **Standardize names** - Standardize the spelling or capitalization of names (for example, cleanse **L'oreal** to **L'Oreal** or **usa** to **USA**).
- **Formatting legal entities** - Use the correct format for all entities (for example, cleanse **co** to **Co.**)
- **Remove noise characters or noise words** - Remove commas, dashes, parentheses, and so on.
- **Add or remove spaces** - Example: Add a space between the number and the measurement (for example, **2 mg**).
- **Remove values** - Blank out placeholder values ("unknown", "not available", and so on) that users submit.

- **Remove special control characters** - Replace characters like tab, returns, line feed, and so on.

Data cleansing process

Cleansing field values involves the following steps:

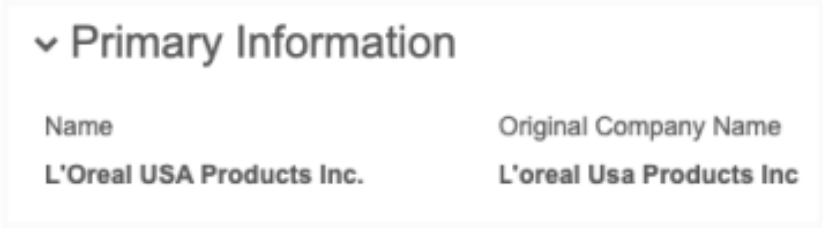
1. Profile your data to understand where data issues are and understand what cleansing rules are needed to fix these data issues.
2. Create cleansing rules in a .csv file. This is your data cleansing dictionary.
3. Upload the .csv file as a lookup table in your Network instance.
4. Create a NEX rule in your source subscription or in a data model field.

If the NEX function is used on a source subscription, the incoming data is cleansed when the job runs. If the NEX rule is applied to a data model field, then the rule triggers and cleanses the data every time the record is updated (for example, in a source subscription or through a DCR).

Important considerations

When you are creating cleansing rules and defining the NEX rule, consider these key practices:

- Profile the data and understand the issues so you can define the correct cleansing rules.
- Test the rules to ensure that they cleanse your data in the right way and do not make any unexpected replacements.
- Store the cleansed value and the original (raw) value in separate fields so you have a record of the change. This will help you to troubleshoot any unexpected replacements.



Create the data cleansing dictionary

In any spreadsheet application, create a .csv file that contains your data cleansing rules. This is your data cleansing dictionary. Each cleansing rule addresses a specific issue.

Example data cleansing dictionary

This data cleansing dictionary, `company_name_dcd`, contains rules to standardize a field called **Company Name**. It has rules for specific company names (for example, L'Oreal and RB Health) but it also has rules to standardize capitalization and legal entities for all company names.

company_name_dcd

	A	B	C	D	E	F
1	string	regex	replacement	case_sensitive	matching_behavior	comment
2	co		Co.	N	TOKEN	standardize legal entity
3	co,		Co.,	N	TOKEN	standardize legal entity
4	usa		USA	N	TOKEN	make USA all upper-case
5	L'oreal		L'Oreal	Y	TOKEN	standardize spelling of company name
6	Rb Health		RB Health	Y	SUBSTRING	standardize spelling of company name
7		(, inc$\| inc$)	, Inc.	N	SUBSTRING	standardize legal entity
8		(, ltd$\| ltd$)	, Ltd.	N	SUBSTRING	standardize legal entity
9		(, llc$\| llc$)	, LLC.	N	SUBSTRING	standardize legal entity
10		(, inc.$\| inc.$)	, Inc.	N	SUBSTRING	standardize legal entity
11		(, ltd.$\| ltd.$)	, Ltd.	N	SUBSTRING	standardize legal entity
12		(, llc.$\| llc.$)	, LLC.	N	SUBSTRING	standardize legal entity
13		a(.*)	A$1	Y	TOKEN	make first character (a to z) of each word uppercase
14		b(.*)	B$1	Y	TOKEN	make first character (a to z) of each word uppercase
15		c(.*)	C$1	Y	TOKEN	make first character (a to z) of each word uppercase
16		d(.*)	D$1	Y	TOKEN	make first character (a to z) of each word uppercase
17		e(.*)	E$1	Y	TOKEN	make first character (a to z) of each word uppercase
18		f(.*)	F$1	Y	TOKEN	make first character (a to z) of each word uppercase
19		g(.*)	G$1	Y	TOKEN	make first character (a to z) of each word uppercase
20		h(.*)	H$1	Y	TOKEN	make first character (a to z) of each word uppercase
21		i(.*)	I$1	Y	TOKEN	make first character (a to z) of each word uppercase
22		j(.*)	J$1	Y	TOKEN	make first character (a to z) of each word uppercase
23		k(.*)	K$1	Y	TOKEN	make first character (a to z) of each word uppercase
24		l(.*)	L$1	Y	TOKEN	make first character (a to z) of each word uppercase
25		m(.*)	M$1	Y	TOKEN	make first character (a to z) of each word uppercase
26		n(.*)	N$1	Y	TOKEN	make first character (a to z) of each word uppercase
27		o(.*)	O$1	Y	TOKEN	make first character (a to z) of each word uppercase
28		p(.*)	P$1	Y	TOKEN	make first character (a to z) of each word uppercase
29		q(.*)	Q$1	Y	TOKEN	make first character (a to z) of each word uppercase
30		r(.*)	R$1	Y	TOKEN	make first character (a to z) of each word uppercase
31		s(.*)	S$1	Y	TOKEN	make first character (a to z) of each word uppercase
32		t(.*)	T$1	Y	TOKEN	make first character (a to z) of each word uppercase
33		u(.*)	U$1	Y	TOKEN	make first character (a to z) of each word uppercase
34		v(.*)	V$1	Y	TOKEN	make first character (a to z) of each word uppercase
35		w(.*)	W$1	Y	TOKEN	make first character (a to z) of each word uppercase
36		x(.*)	X$1	Y	TOKEN	make first character (a to z) of each word uppercase
37		y(.*)	Y$1	Y	TOKEN	make first character (a to z) of each word uppercase
38		z(.*)	Z$1	Y	TOKEN	make first character (a to z) of each word uppercase

Mandatory columns

The file must contain the following columns with a value:

- **string** or **regex** - Each rule must include either a string or a regular expression (regex).

  Only one of these columns can be populated for a rule. If you have a value in both columns, an error occurs.

  - **string** - Means that the value is interpreted as a string constant.
  - **regex** - Means that the value is interpreted as a regular expression.

These two alternative columns are there to make it easier for you if you want to match and replace just on a string. In that case, enter the value into the **string** column. You do not have to take care of escaping any characters that have a special meaning in regular expressions because everything in that column is treated as a string constant.

However, if you want to use regular expressions to do some advanced matching and replacing, then enter the expression into the **regex** column. Everything in this column is treated as a regular expression, meaning that you might have to escape some characters depending on your specific expression and use case.

Example

If you specify a dot (.) character in the string column, then this means only every occurrence of a dot is replaced (for example, if you replace (`.`) with an empty string ('), `n.a.` is cleansed to `na`).

However, in regular expression, the dot (.) character has a completely different meaning; it represents any character or number. So, if you replace the dot (.) character as a regular expression in the **regex** column with an empty string (') , all characters are replaced ("").

- **replacement** - The value that you want the matching string or substring to be replaced with. The value can also be empty if you are stripping the field of the value.

Optional columns

The following columns can be included in the file:

- **case_sensitive** - Supported values are **Y** or **N**. If the column is not included in the file, the default value is **N**. If you want the rule to match only on a specific case, specify **Y** in the column.

  Case applies to both **string** column values and **regex** column values.

  Example

  These regex rules are similar, but one is case sensitive.

  ```
  regex,replacement,case_sensitive
   [a-z],x,Y
   [a-z],x,N
  ```

  - The first rule replaces only all lowercase characters with `x`.
  - The second rule replaces all characters (regardless of letter case) with `x`.
- **matching_behavior** - Identify the type of string or substring to match for replacement. If the column is not included in the file, the default matching behavior is **SUBSTRING**.
  - **SUBSTRING** - Every occurrence of the matching string is replaced.
  - **TOKEN** - Occurrences are replaced only if they are words separated by other words through white spaces. A TOKEN is a sequence of non-white space characters separated by white spaces (blanks, tabs, and so on). If a word is separated by a dash (-), comma (,), parentheses (), or other character, it is not treated as a TOKEN.

Tip: Document your cleansing rules by including an optional column to describe each rule.

Example rules

Example 1

This cleansing rule standardizes all occurrences of `USA` as a token.

string	regex	replacement	case_sensitive	matching_behavior
usa		USA	N	TOKEN

Column values

- **string** - Match the entire string `usa`.
- **replacement** - Replace the matching string with `USA`.
- **case_sensitive** - Replace any occurrence of `usa` regardless of letter case. This includes `usa`, `Usa`, `USa`, and so on.
- **matching_behavior** - TOKEN means only match when `usa` is an entire word (it is separated by white space). A string that includes `usa` with other characters; for example, `tho`**usa**`nd`, would not be cleansed as `thoUSAnd`.

Example 2

This cleansing rule standardizes the legal entity `Inc.`.

string	regex	replacement	case_sensitive	matching_behavior
	(, inc$\| inc$)	, Inc.	N	SUBSTRING

Column values

- **regex** - Match any occurrence of `,inc` or `inc` at the end of the string.
- **replacement** - Replace the matching string with `, Inc.`.
- **case_sensitive** - Replace any occurrence of the matching substring regardless of letter case.
- **matching_behavior** - SUBSTRING means match any occurrence of the substring.

Sequence of the rules

The sequence of the columns does not matter, but the sequence of the rules does matter. Depending on the sequence, you could get a different result.

Example

In this example, we want to cleanse the string `n.a.`.

Dictionary example 1

	A	B	C	D	E
1	string	regex	replacement	case_sensitive	matching_behavior
2	.				SUBSTRING
3	n.a.				SUBSTRING

Result: With this sequence of cleansing rules, the output is `na`.

Dictionary example 2

A	B	C	D	E
string	regex	replacement	case_sensitive	matching_behavior
n.a.				SUBSTRING
.				SUBSTRING

Result: With this sequence of cleansing rules, the output is an empty string.

Ensure that the rules are ordered so that you get the expected output.

Upload the file to Network

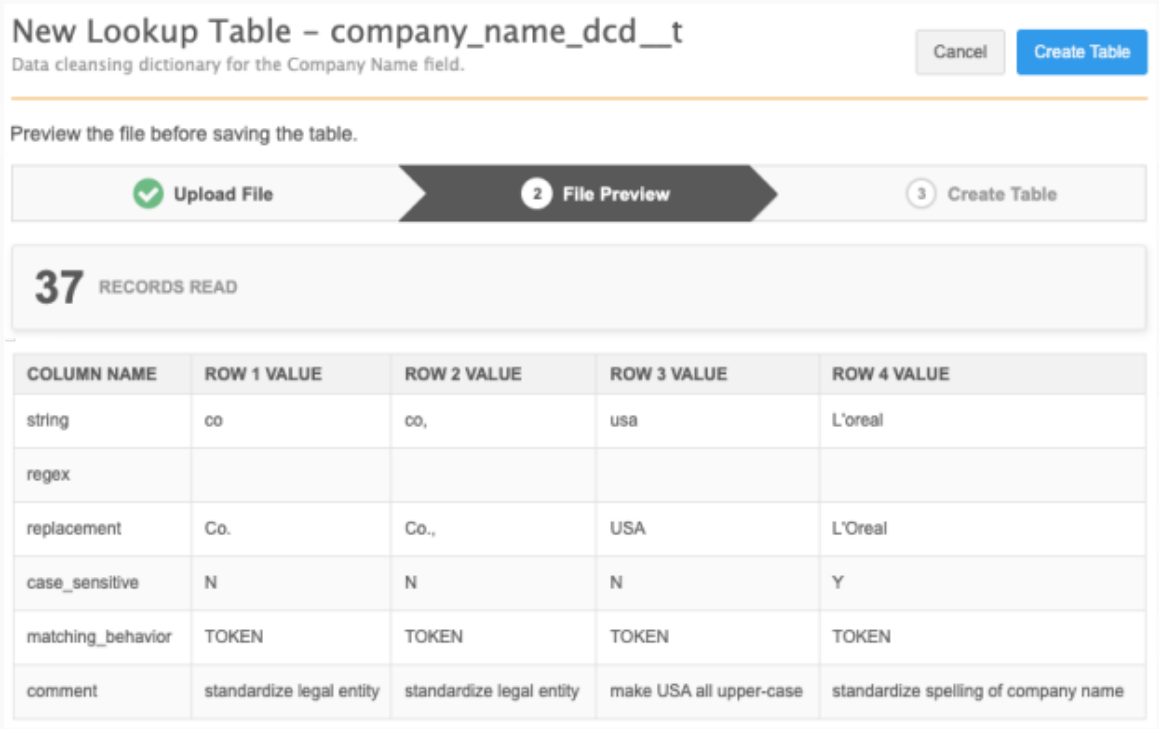
When you have created the data cleansing rules, upload the .csv file as a lookup table.

1. In the Admin console, click **Data Model > Lookup Tables**.
2. Click **Create Lookup Table**.
3. Type the **Table Name** and **Description**, confirm whether the file contains third party data, and upload the .csv file.

   The file is validated to ensure that it complies with supported lookup table requirements, but the file is not checked to ensure that the mandatory data cleansing columns are included.

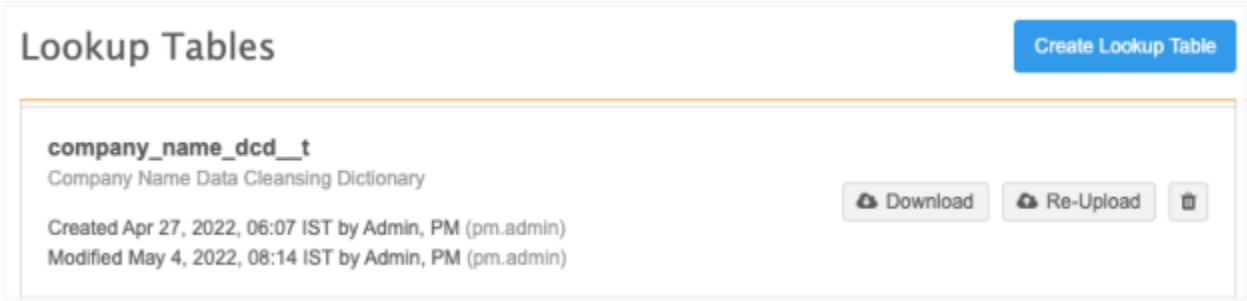
   For more information about lookup tables, see Create a lookup table.

4. On the **File Preview** tab, review the data and click **Create Table**.



COLUMN NAME	ROW 1 VALUE	ROW 2 VALUE	ROW 3 VALUE	ROW 4 VALUE
string	co	co,	usa	L'oreal
regex				
replacement	Co.	Co.,	USA	L'Oreal
case_sensitive	N	N	N	Y
matching_behavior	TOKEN	TOKEN	TOKEN	TOKEN
comment	standardize legal entity	standardize legal entity	make USA all upper-case	standardize spelling of company name

The `__t` suffix is automatically appended to the table. Remember to add the suffix when you reference the data cleansing dictionary file in the NEX rule.



Edit the dictionary

To change the data cleansing rules, download the lookup table. When the changes are complete you can re-upload the .csv file.

Apply the data cleansing function

The data cleansing function can be applied to a NEX Rule in a source subscription or a on data model field.

NEX rule syntax

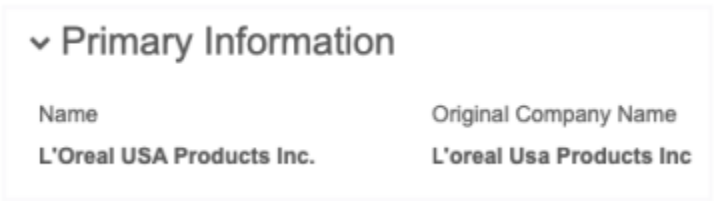
```
CLEANSE_STRING(<field_name>, '<data_cleansing_dictionary_name>')
```

- **field_name** - The field that will be cleansed.
- **data_cleansing_dictionary_name** - The name of the dictionary that you added as a lookup table.

Example rule

```
 [
     "company_original_name__c = company_name__c",
     "company_name__c = CLEANSE_STRING(company_name__c,
'company_name_dcd__t')"
 ]
```

This rule means that first the existing company name will be saved to a second field, `company_original_name__c`, and then cleanse the company name. The cleansed name will be stored in the `company_name__c` field.

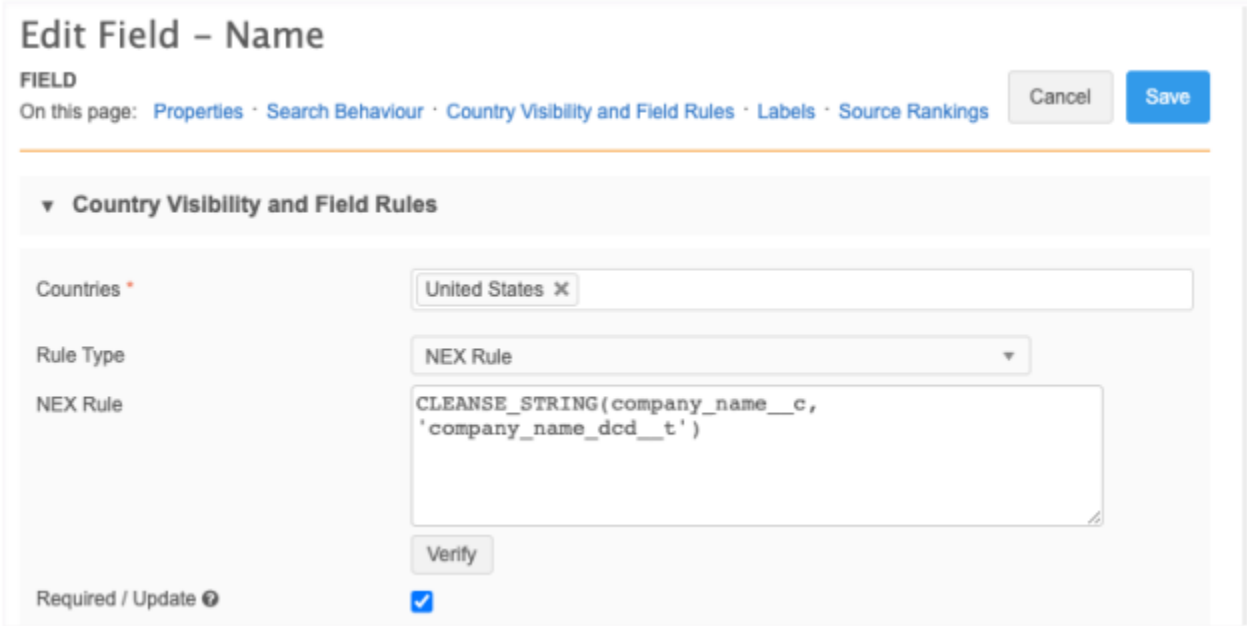


To apply data cleansing for a field, add it to a source subscription or a data model field.

- Source subscription - In the **Network Expression Rules** section, add the rule.

  You will get better match results if you cleanse and standardize the values of the fields that you use in matching (for example, fields like hospital names, company names, product names, and so on). To ensure that the fields are cleansed and standardized before matching, add the rule to the earliest possible stage (**File Preparation** or **Transformation** rule points).

- **Data model field** - Expand the **Country Visibility and Field Rules** section and add the NEX rule.



Replacing special characters

You can remove special control characters like tab, returns, line feeds, and so on by replacing them with a defined string (for example, an empty string).

Considerations

- All special characters must be defined in the **regex** column (not in the 'string' column).
- You can replace any special characters supported by Java Regex; only encodings that are supported by Java RegEx can be used. For more information, see https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html.
- All backslashes (\) must be escaped with another backslash (for example: `"\\t"` instead of `"\t"`)
- The value in the **regex** column must be in quotes (").

dcd__t.csv

```
regex,replacement,matching_behavior,case_sensitive
"\\t",,substring,y
"\\u0009",,substring,y
```

Nulling existing values

If the final outcome of the cleansing is an empty string, then the function will return a `NULL` value.

Example

This data cleansing rule expects to replace field values that are `n/a` with an empty string.

```
string,replacement
 n/a,
```

With this cleansing rule, the data cleansing function will return `NULL`, not an empty string.

To nullify any existing field values through the data cleansing function in a source subscription, add the following property to the **Advanced Mode**:

```
    "feed.retain.null": "true"
```

Trimming white space

By default, the string cleansed by the function is trimmed; the whitespace on the left and right of the cleansed string is removed.

Delete the data cleansing dictionary

You can remove the dictionary from the Lookup Tables page. When you remove a data cleansing dictionary, you should also remove the NEX rule. Run time errors will occur for any NEX rule that references a deleted data cleansing dictionary.

Data cleansing errors

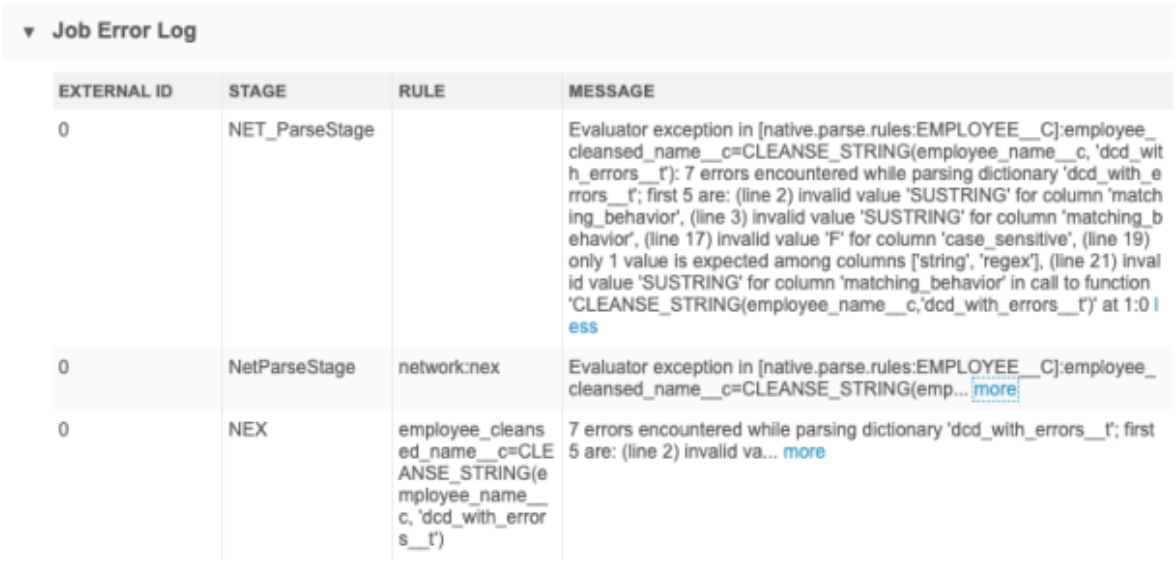
When you upload the .csv file as a lookup table, the data cleansing rules are not validated for consistency and completeness. If mandatory information is missing or if there are ambiguities, the data cleansing function will throw a run-time error with details about these issues.

Errors will occur in the following situations:

- An expected column occurs more than once in the data cleansing dictionary (for example, two columns named **string**).
- Values are present in both the **string** and **regex** columns.
- The **string** and **regex** columns are both missing from the dictionary (at least one of them is required).
- The **replacement** is missing from the dictionary.
- The lookup table referenced by the data cleansing function does not exist.
- Unsupported values are present in the columns.

Source subscription

When the data cleansing function is called and an error occurs, the errors display in the **Job Error Log** section on the Job Details page. The first five errors display.



In this example the following errors occurred:

- *Substring* is spelled incorrectly.
- The **case_sensitive** column contained an unsupported value (**F**). Only **Y** or **N** are supported.
- A rule contained values in both the **string** and **regex** columns.

File explorer

SMART TABLES

22R1.1

Use smart tables to open, view, and augment .csv files directly from File Explorer. This enables you to share and view files within Network. Smart tables are helpful for users who do not have robust reporting and Network data model knowledge. You can easily add Network data in bulk to the file and download the changes. For example, if a file contains a list of Veeva IDs (VIDs) but you need more data to analyze the content, you can add Network fields to the file to view that data.

This feature is enabled by default in your Network instance if the File Explorer feature is enabled.

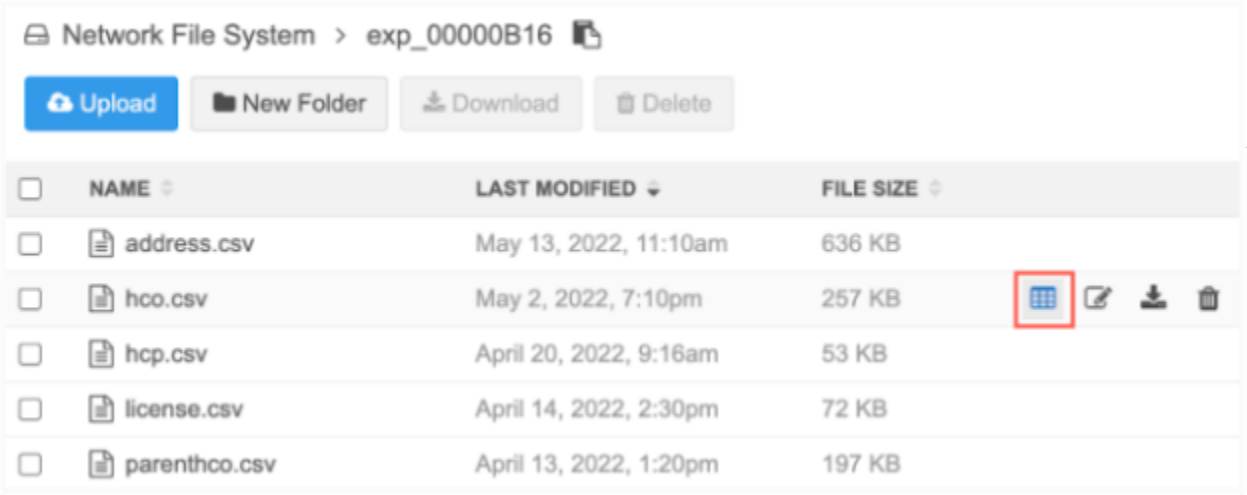
Note: Users must have FTP access to use the File Explorer.

View smart tables

Smart tables are supported for .csv files. You can open these files directly from File Explorer.

- Beside any .csv file, click the **Table** icon. The icon displays only on .csv file.

  You can also double-click on a file to open it.



The file loads and displays all the file columns and the first 1000 rows. The first row in the file displays as a header row for the smart table.

The name of the file displays at the top of the table. Click the **Back** arrow to return to the File Explorer.

address.csv

View Original File | Download | Add Fields

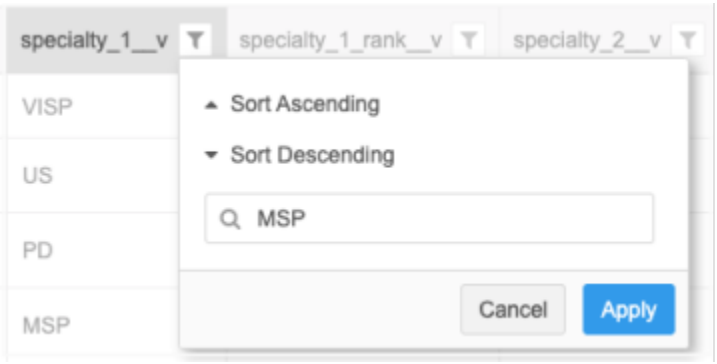
address_line_1__v	address_line_2__v	address_line_3__v	address_ordinal__v	address_status__v	address_type__v	address_ve
2947 Rodeo Park Dr E			1	A	P	V
200 1st St SW Ste Sl123			1	A	P	V
200 1st St SW			2	I	P	V
4800 Sand Point Way NE			1	A	P	V
4800 Sand Point Way NE Mb5.420			2	I	P	V
1959 Ne Pacific St			3	I	P	V
4800 Sand Point Way Ne	M/S O.A.9.120.1		4	I	P	P
4800 Sand Point Way NE			5	I	P	V
4800 Sand Point Way NE # 5371			6	I	P	V
1101 Madison St Ste 550			7	I	P	V

If the .csv file is empty or the file cannot be read, a message displays.

Sorting and filtering columns

Smart tables can be sorted by one column at a time. Columns can be sorted in ascending or descending order.

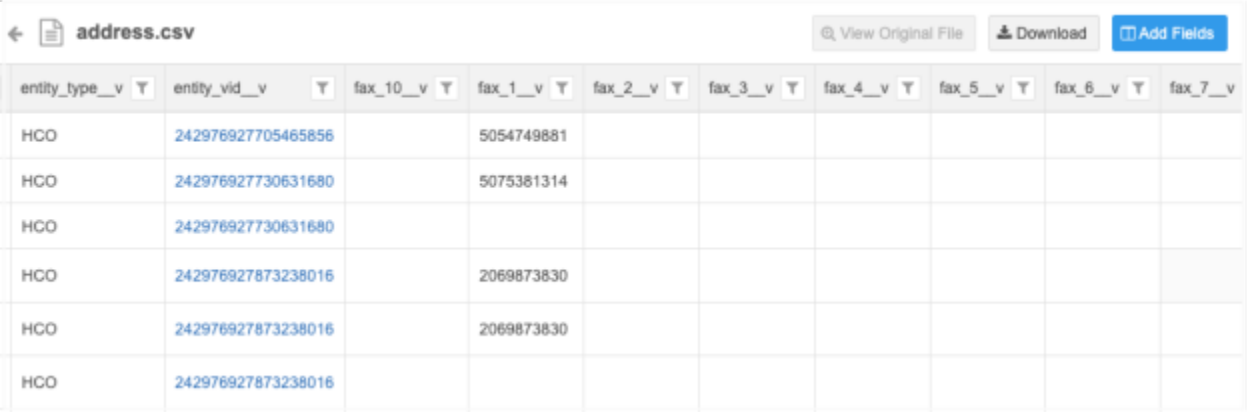
You can also filter the column by searching for specific data; for example, in a Specialty column, you can search for a specific specialty and filter the column. You can filter the smart table on multiple columns. The **Filter** icon in the column header is highlighted to identify sorted and filtered columns.



The first 1000 rows are displayed in the smart table view, but any sort or filter is applied to the entire column. You can download the file to view all the sorted or filtered data.

VID columns

Network tries to identify records based on fields that contain 18-digit numbers, which is typically a Veeva ID (VID) field. The VID value will display as a hyperlink. For example, for an address file, the **entity_vid__v** column contains an 18-digit number so Network identifies this as a VID field and adds a link so you can navigate directly to the profile page.

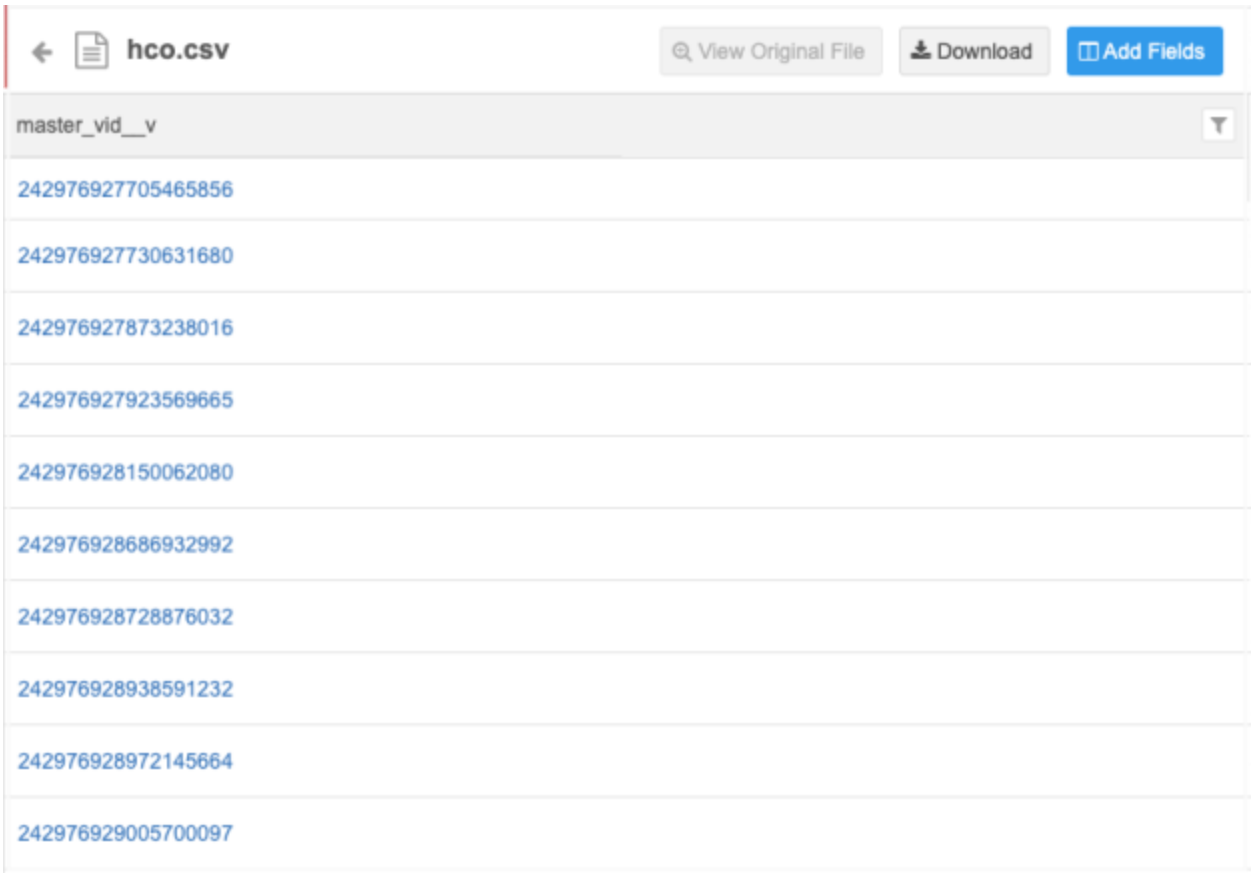


If the link to the record doesn't work or you do not have access to the profile, an error displays. For example, if you click a VID in the **vid__v** column of an address file, no record will be found. Instead, click a VID in the **entity_vid__v** column to open the associated HCP or HCO record.

Add fields to smart tables

You can extend the smart table with Network data. For example, if you receive a file that is a list of Veeva IDs, it would be helpful to see more information about each record without clicking the VID link and navigating to each record profile. In this case, you can add fields to view more relevant Network data for these records.

Note: Any changes that you make to the smart table are available locally. They can be downloaded, but they are not saved to the file in File Explorer.

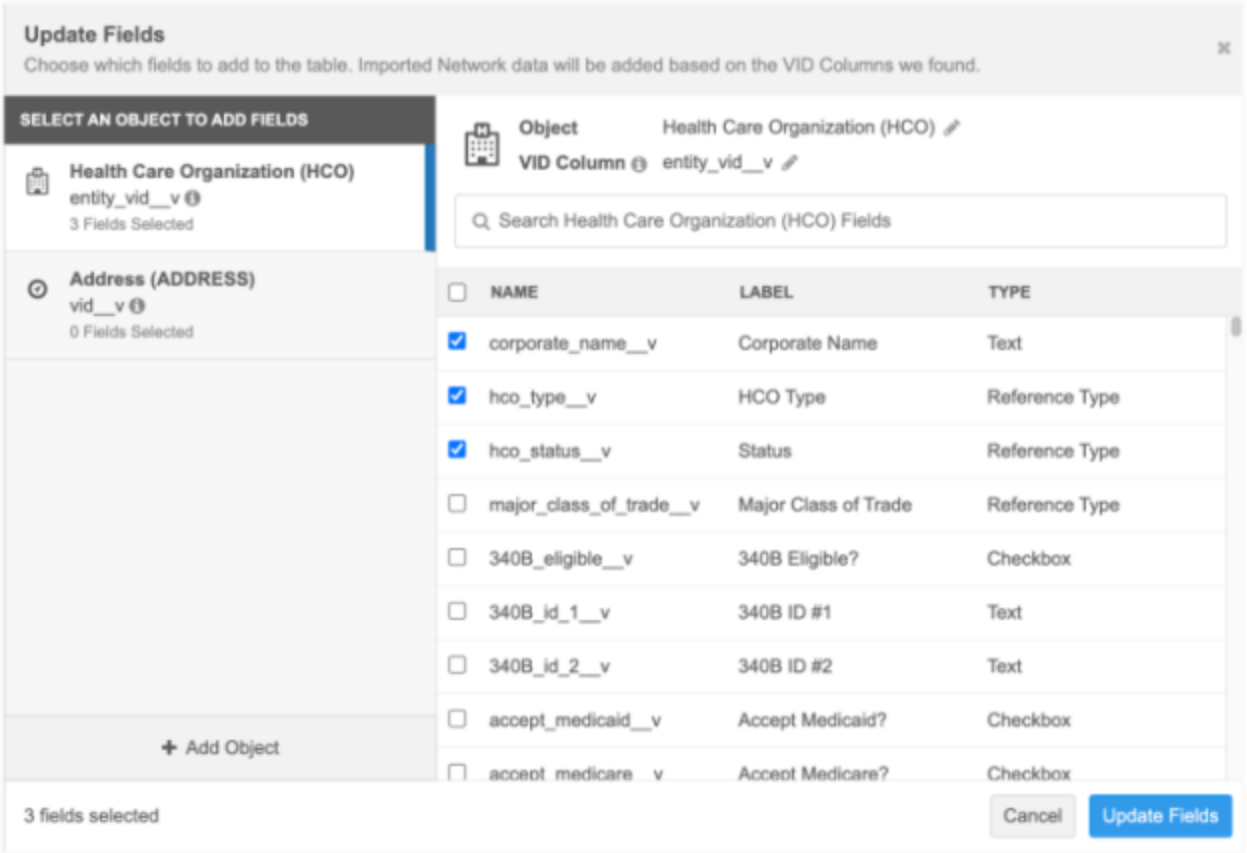


Network uses the VID columns to map the Network fields that can be joined to the file. For example, when Network identifies a VID column as an HCO object, the HCO fields are available to add to the file. If a VID column isn't identified, default field mappings are not available.

Add fields

To add fields to the smart table:

1. Click **Add Fields**.
2. In the **Update Fields** dialog, you can see the object and the VID column that is used to identify the fields that you can add to the file.



Only fields that are enabled are available. Disabled fields, set fields (for example, Set of Licenses), and restricted fields are not available for smart tables.

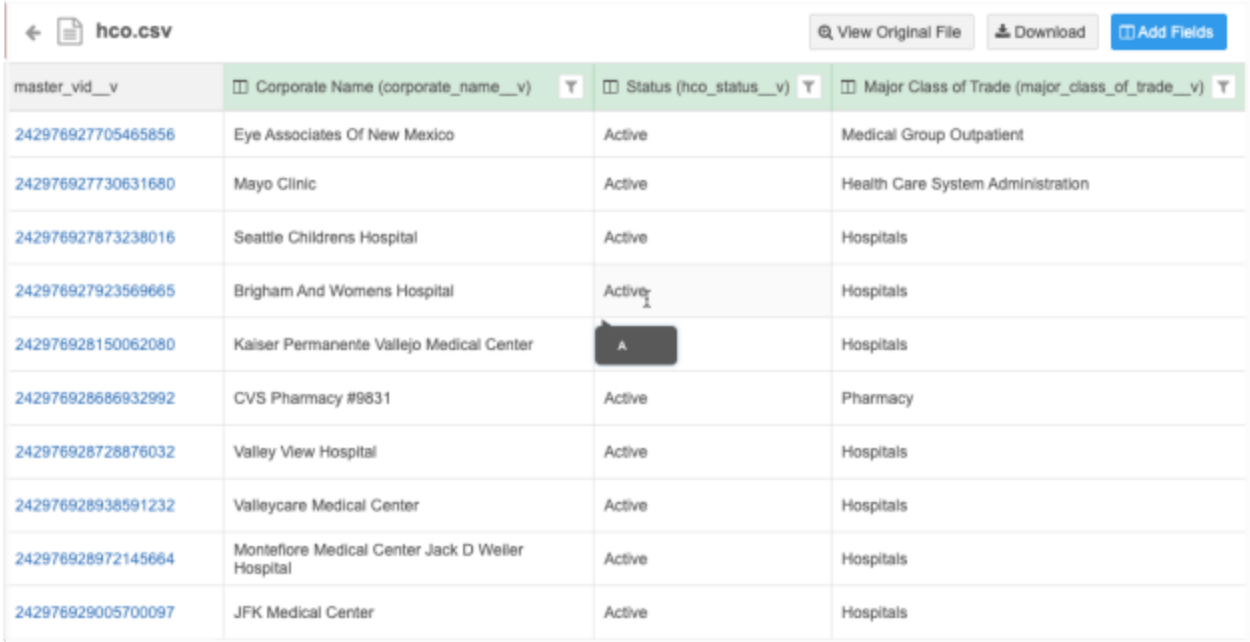
3. Select the fields that you want to add.

   The order of the fields are prioritized by their usefulness and relevance. For example, for HCO objects, the Corporate Name, HCO Type, HCO Status, and Major Class of Trade display at the beginning of the list. Then, the remaining fields are listed alphabetically. You can also use the **Search** field to find fields.

   As you select fields, the count displays under the object in the left pane. The total count of selected fields for all objects displays at the bottom of the dialog.

4. Click **Update Fields** to add the selected fields to the smart table.

   The new columns are added and highlighted in green. Network populates the columns with the field data.



master_vid__v	Corporate Name (corporate_name__v)	Status (hco_status__v)	Major Class of Trade (major_class_of_trade__v)
242976927705465856	Eye Associates Of New Mexico	Active	Medical Group Outpatient
242976927730631680	Mayo Clinic	Active	Health Care System Administration
242976927873238016	Seattle Childrens Hospital	Active	Hospitals
242976927923569665	Brigham And Womens Hospital	Active	Hospitals
242976928150062080	Kaiser Permanente Vallejo Medical Center	A	Hospitals
242976928686932992	CVS Pharmacy #9831	Active	Pharmacy
242976928728876032	Valley View Hospital	Active	Hospitals
242976928938591232	Valleycare Medical Center	Active	Hospitals
242976928972145664	Montefiore Medical Center Jack D Weiler Hospital	Active	Hospitals
242976929005700097	JFK Medical Center	Active	Hospitals

New columns

Fields are added alphabetically after the original columns but some fields are prioritized. The prioritized fields are added (if selected) and then the remaining fields display in alphabetical order. Main object fields display before sub-object fields.

Prioritized fields

Display Order	HCP Fields	HCO Fields	Address Fields
1	first_name__v	corporate_name__v	address_line_1__v
2	last_name__v	hco_type__v	address_line_2__v
3	formatted_name__v	hco_status__v	address_line_3__v
4	hcp_type__v	major_class_of_trade__v	locality__v
5	hcp_status__v		administrative_area__v
6	specialties 1-10 (specialty_1__v, specialty_2__v, ...)		postal_code__v
7			country__v

Field data considerations

- The column is populated with data that corresponds to the field and record (using the VID).
- The data that is available in the reporting database is added to the smart table. Field data is not calculated when it is added.
- The data that displays is based on the **Report Results** permission in your user account. If the permission is Restricted, the data is limited by your assigned data visibility profile.
- Data can be added for invalid records.
- When reference type fields are added, the label displays in the column. Hover over the label to view the reference code.
- A maximum of 40 reference type fields across all objects can be added to a smart table.

Remove fields

To remove fields:

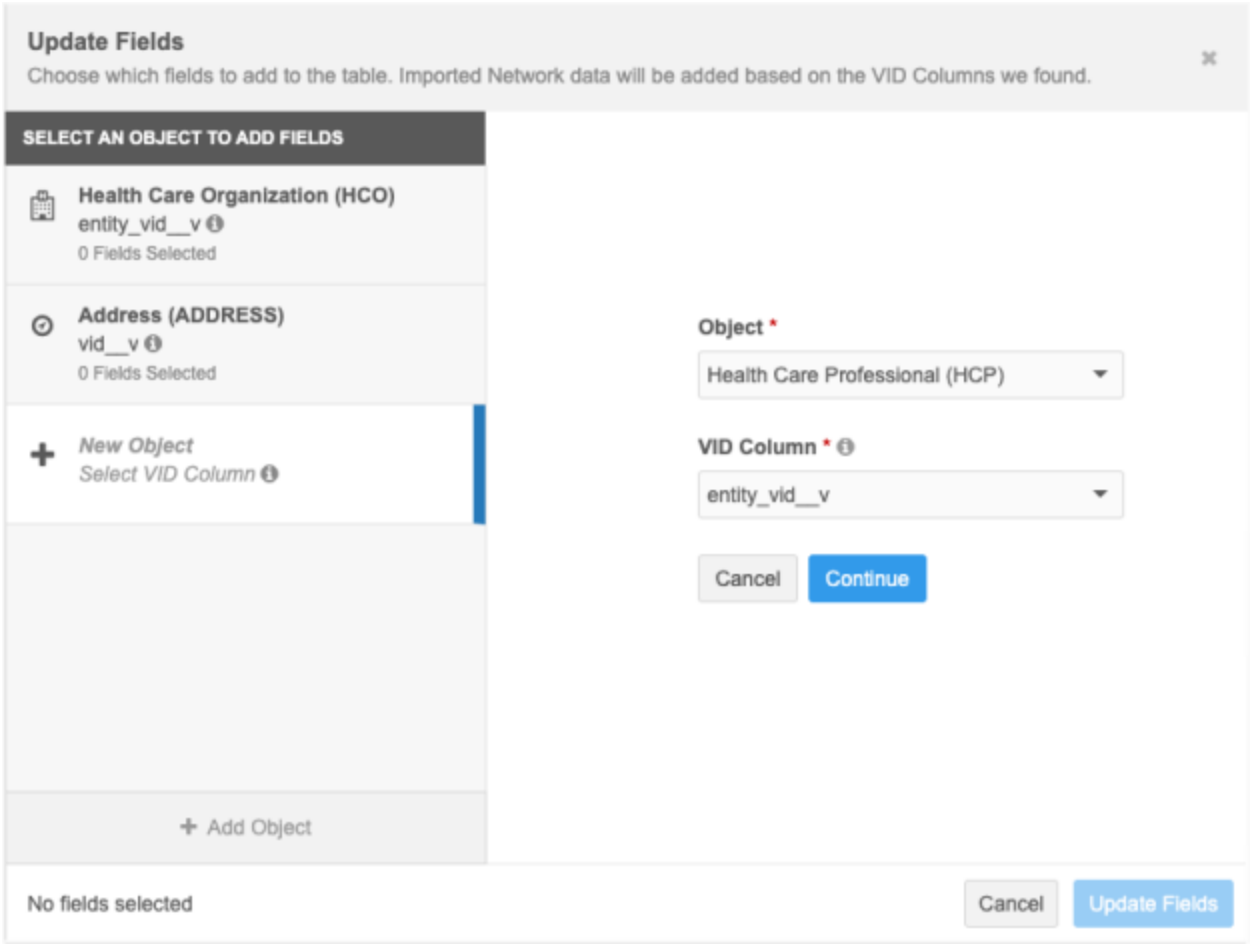
1. **Click Add Fields** and clear the selected fields.
2. Click **Update Fields** to save your changes.

The columns are removed from the smart table.

Add a new object

There might be cases where there are additional objects in the file, but they were not added to the smart table. This typically happens because only the first 50 rows of the VID columns are used to look up Network objects. For example, if you have a file of addresses and Network identifies HCO objects only but you know that some of the addresses belong to HCP objects, you might need to add the HCP object to the file.

Add the object so you can add the fields associated with the object to the smart table.



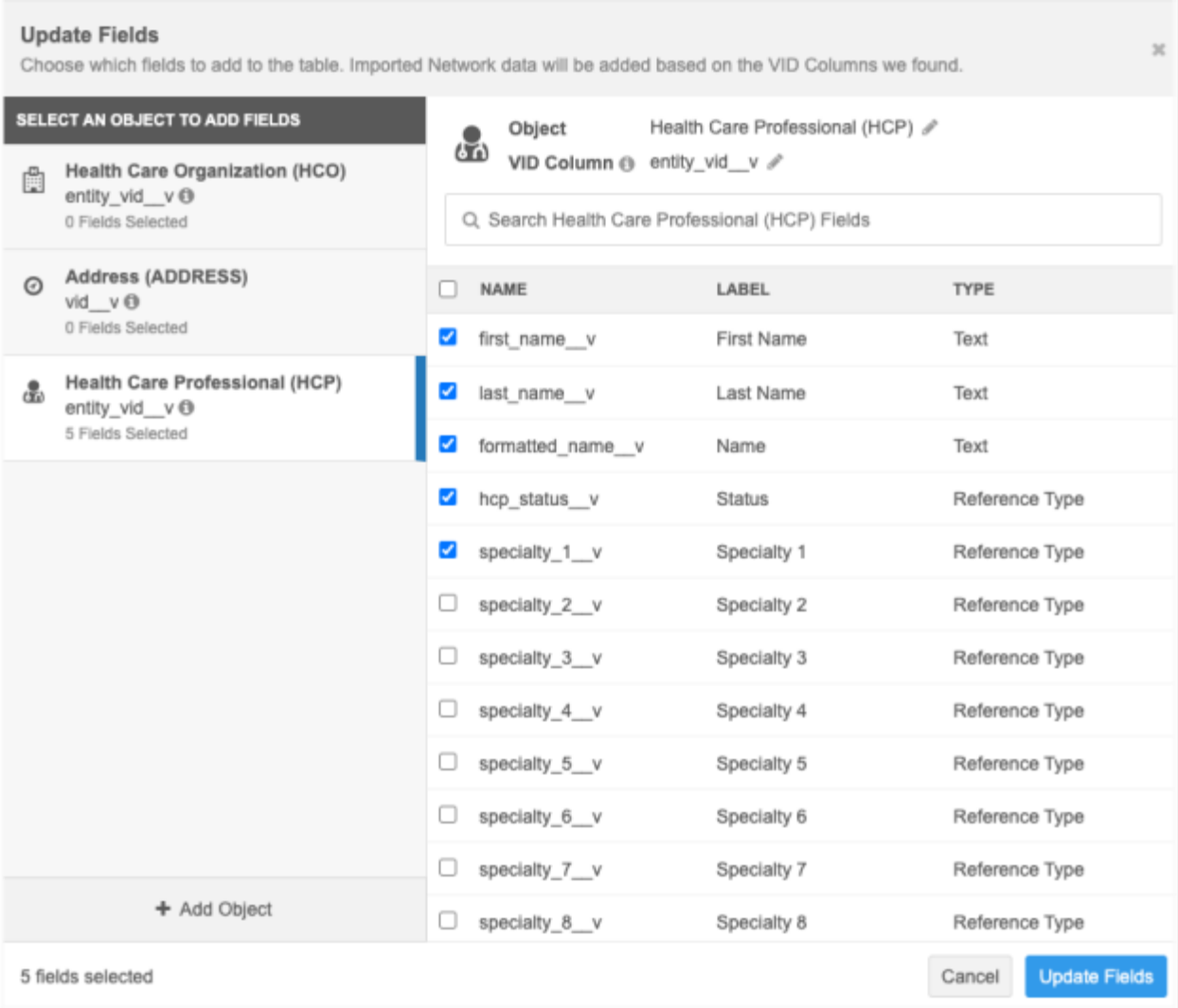
To add an object:

1. In the **Update Fields** dialog, click **Add Object**.
2. Expand the **Object** list and select the object.

   All enabled objects in your Network instance are supported in smart tables.
3. Expand the **VID Column** list and select the column that contains VID values. Network uses this field to join to the object table and identify all the fields.

   All fields (columns), except empty headers, from the original file display in the list

   You can add an object more than once but each **Object/VID Column** mapping must be unique. An object cannot be added and mapped to the same VID column.
4. Click **Continue**.
5. When the object is added, select the fields to add to the smart table.

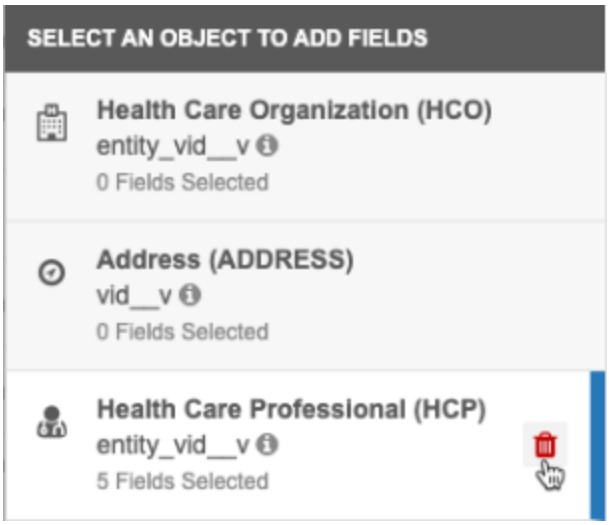


You can also edit the **Object** and **VID Column** fields to change the object type and ID field.

6. Click **Update Fields** to view the new object fields (columns) in the smart table.

Remove objects

To remove an object, click the **Trash** icon that displays when you hover over the object. The object and fields are removed from the file preview.

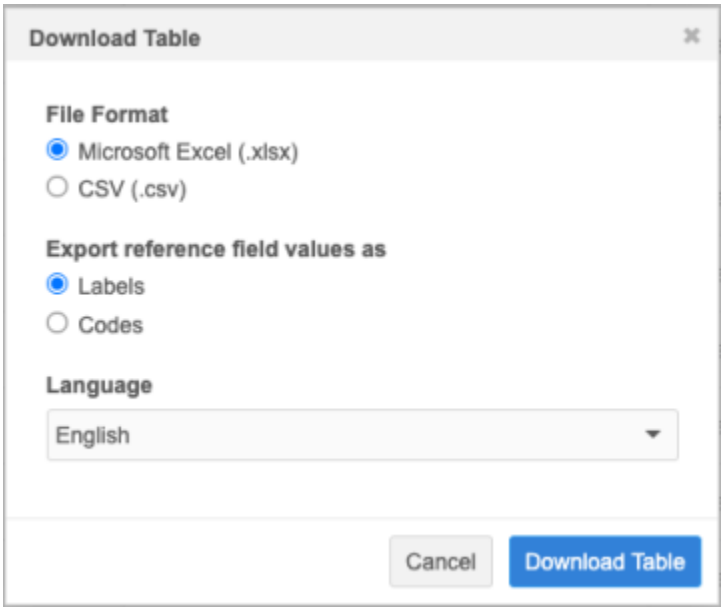


Click **Update Fields** to save the changes. The object and fields are removed from the smart table.

Download smart tables

You can download the smart table to your local directory as a .csv file. Any changes that you made are downloaded; for example, if you sorted a column or added columns, those updates are included in the downloaded file.

1. To download the smart table, click **Download**.
2. In the **Download Table** dialog, choose the file format and the type of reference field values to export. If you export reference values as labels, the user's language is selected by default.
3. Click **Download Table**.



Large files can take time to download.

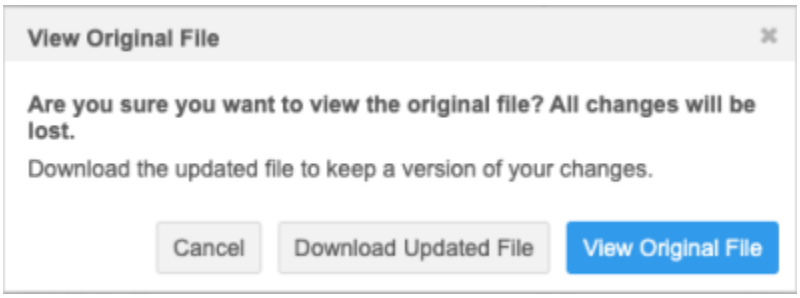
View original file

After you have made changes to the smart table, you can remove them.

1. At the top of the smart table, click **View Original File**.

   This button is dimmed until changes are made to the smart table.

2. In the **View Original File** pop-up, you can download the modified file to retain the changes locally before you revert to the original file.



You can also click the **Back** arrow beside the file name to exit the file and return to the File Explorer.

Data model

COUNTRY SUPPORT (22R2.0)

22R2

Veeva OpenData data models have been added for the following countries :

- Egypt (EG)
- Jordan (JO)

The data models are based on the Other Countries (ZZ) data model. The data models also include additional fields so they are consistent with other European OpenData data models.

The activated reference codes are based on the reference codes that are activated for Other Countries (ZZ), along with additional reference codes supported by the European OpenData team.

Localization

English (en) translations will be used for the Network UI, data model fields, and reference data for both countries.

DATA PRIVACY OPT OUT (22R2.0)

22R2

Veeva OpenData now manages HCP opt outs in the following countries:

- Egypt (EG)
- Jordan (JO)

Two data model fields have been enabled for these countries for the HCP object:

- `data_privacy_opt_out__v`
- `data_privacy_opt_out_date__v`

Records that are opted-out by Veeva OpenData do not display and cannot be accessed in downstream systems. This ensures data privacy for opted-out HCPs to satisfy regional regulatory requirements.

Opted-out countries

To review the list of opted-out countries, in the Admin console:

1. Click **Data Model > Data Domains** and choose the **Customer Master** domain.
2. Select the **Health Care Professional** object and find the `data_privacy_opt_out__v` field in the **Fields** section.
3. Click the field to review the list of opted-out countries that are managed by Veeva OpenData.

NETWORK ADDRESS INHERITANCE

22R2

China can now be added to Network address inheritance configurations with other countries. Previously, China required a separate country group configuration because the `address_type__v` field was not included in the China data model. This meant that the **Parent Address Type** setting was not supported for China . This setting is used to copy the address from a parent HCO relationship.

Now, if the `address_type__v` field is enabled for the address object, the **Parent Address Type** setting is supported and China can be added to a **Country Group** with other countries.

COUNTRY SUPPORT (22R1.1)

22R1.1

Veeva OpenData data models have been added for countries in Latin America and Asia Pacific.

Latin America

Veeva OpenData data models have been added for the following countries:

- Bolivia (BO)
- Paraguay (PY)
- Uruguay (UY)

These data models are duplicates of the Mexico data model. They are based on the Other Countries (ZZ) data model and include additional fields so they are consistent with other Latin American OpenData data models.

The activated reference codes are based on the reference codes that are activated for Other Countries (ZZ), along with additional reference codes supported by the Latin America OpenData team.

Localization

Spanish (es) translations will be used for the Network UI and data model fields. Spanish-Mexico (es-MX) translations will be used for reference data.

Asia Pacific

Veeva OpenData data models have been added for the following countries:

- India (IN)
- Indonesia (ID)
- Malaysia (MY)
- Philippines (PH)
- Thailand (TH)
- Vietnam (VN)

The data models are based on the Other Countries (ZZ) data model. The data model also includes additional fields so they are consistent with other Asia Pacific OpenData data models.

The activated reference codes are based on the reference codes that are activated for Other Countries (ZZ), along with additional reference codes supported by the Asia Pacific OpenData team.

Localization

English (en) translations will be used for the Network UI and data model fields. Australian English (en-AU) translations will be used for reference data.

DATA PRIVACY OPT OUT (22R1.1)

22R1.1

Veeva OpenData now manages HCP opt outs in the following countries:

- India (IN)
- Indonesia (ID)
- Malaysia (MY)
- Philippines (PH)
- Thailand (TH)
- Vietnam (VN)

Two data model fields have been enabled for these countries for the HCP object:

- `data_privacy_opt_out__v`
- `data_privacy_opt_out_date__v`

Records that are opted-out by Veeva OpenData do not display and cannot be accessed in downstream systems. This ensures data privacy for opted-out HCPs to satisfy regional regulatory requirements.

The `opt_out__v` field is also enabled for these countries. The field is used to guide marketing usage of records in these countries.

Opted-out countries

To review the list of opted-out countries, in the Admin console:

1. Click **Data Model > Data Domains** and choose the **Customer Master** domain.
2. Select the **Health Care Professional** object and find the `data_privacy_opt_out__v` field in the **Fields** section.
3. Click the field to review the list of opted-out countries that are managed by Veeva OpenData.

CLUSTER MANAGEMENT

22R1.1

Customers can enrich addresses by adding cluster codes from third party cluster providers. In this release, Network has included support for the following countries for the cluster provider, IQVIA™.

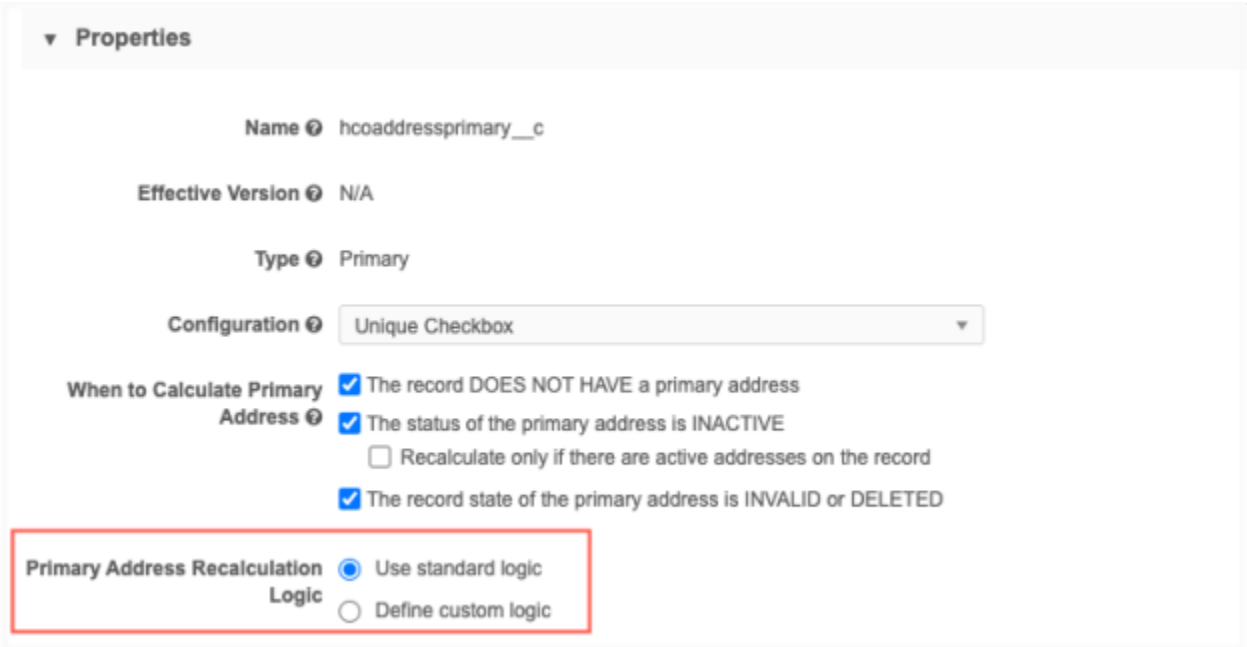
- Australia
- Czech Republic
- Portugal
- Slovakia

A TPA must be signed with the third party cluster provider to use the cluster management feature. For more information, see the topic called Managing clusters in the *Veeva Network Online Help*.

PRIMARY ADDRESS

22R1.1

Administrators and data managers can now include their own conditions for recalculating primary addresses for Unique Checkbox primary fields. This is helpful to ensure that the address that is the most relevant for your business purposes is selected as primary. For example, an address that is outside of the sales territory could be calculated as the new primary because the existing primary recalculation logic does not consider postal codes. You can now specify the fields that you want Network to consider during the primary address recalculation.



This enhancement is enabled by default in your Network instance. It is supported only for Unique Checkbox primary address configurations.

Primary recalculation logic

The Primary Address Recalculation Logic displays when you select a primary calculation option.

1. On a primary address field that uses the Unique Checkbox configuration, select an option for calculating primary address. For example, select **The status of the primary address is INACTIVE**.
2. In the **Primary Address Recalculation Logic** section, choose one of the options:
    - **Use standard logic** - Standard logic considers the following: source rank, address rank, last updated date and time, and the highest (newest) Veeva ID.
    - **Define custom logic** - Define the fields that you want Network to use for recalculating a primary. This is selected by default.

Review the following sections to understand each recalculation logic option.

Using standard logic

If you select **Use standard logic**, Network continues to recalculate primary addresses using the following conditions (in this order) to match against any existing primary address:

- **Source rank** - The rank on the primary field is the same or higher than the existing primary address (rank of 1 is highest).
- **Address rank** - The address ordinal value that is the lowest.
- **Date and time** - The address where the primary field was last updated.
- **Veeva ID** - The newest (highest) address entity ID.

Exclude addresses

Network automatically excludes records that are Invalid and Inactive. You can define additional address criteria to exclude from the primary address recalculation. A maximum of three custom exclude criteria can be added. For example, exclude all addresses that are the `Mail Only` type.

- To exclude addresses, click **Add Field**. Expand the list to choose the **Field** and then select or type the field **Value** to exclude.

Standard logic recalculation

When Network recalculates the primary address using standard logic, the following steps are taken:

1. **Run Inactive/Invalid logic** - Exclude addresses that are Inactive or Invalid.

2. **Run the exclude logic** - Remove any addresses for primary consideration based on the exclude criteria you have defined.
3. **Run the standard condition logic** - Network recalculates the best primary address based on the order of the standard conditions: source rank, address rank, last updated date and time, and the highest (newest) Veeva ID

Example

Address data	Current Primary	Address 1	Address 2	Address 3	Address 4
City	Fort York	Fort York	Fort York	Fort York	Fort York
Postal Code	45678	45678	45678	45678	456701
Address Type	Professional	Professional	Professional	Mail Only	Professional
Address Status	Active	Active	Active	Inactive	Active
Source Rank	3	3	3	3	3
Address Rank (ordinal)	2	2	3	1	1
Date/Time	2022-01-12 10:53:06	2022-04-10 09:22:11	2022-05-12 15:03:25	2022-02-08 02:44:11	2022-04-30 12:45:06
Veeva ID	243244003745596484	932184542274324385	932926364920187752	243244003745529263	243244003745596476

Result

Address 4 is recalculated as the primary address because it has the lowest ranked ordinal.

The addresses all had the same source ranking (3), so Network used the next condition (address ordinal) to find the best primary address.

Note that **Address 3** was excluded because it is Inactive.

Custom logic

If you select **Define custom logic**, Network uses the conditions that you define to recalculate primary addresses. For example, to ensure that the new primary address is located in the same sales territory, you can add postal code as a condition to match to the current primary address.

A maximum of three conditions can be defined. Networks matches the conditions you define to the current primary to recalculate the new primary address.

Note: The custom recalculation logic runs when the existing primary address defined on a record becomes disqualified.

To define custom logic:

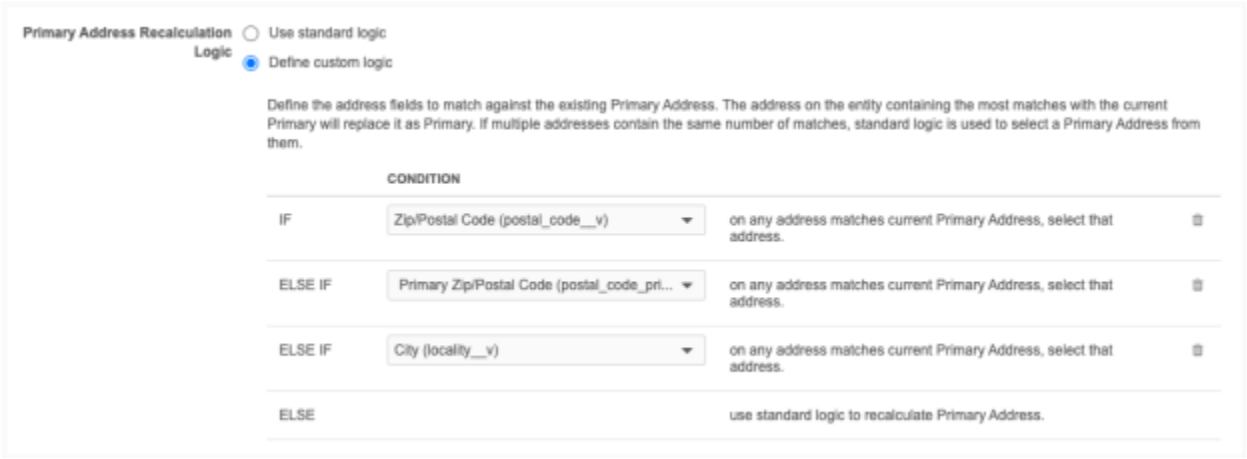
1. In the **Condition** section, expand the list and choose an address field.
2. To add another field to match on, click **Add Condition**. Select the field.

   A maximum of three conditions can be defined.
3. To exclude address criteria from the recalculation logic click **Add field**. Define the **Field** and **Value** that you want to exclude. A maximum of three custom exclude criteria can be added. Invalid and Inactive addresses are automatically excluded.

Custom logic recalculation

When Network recalculates the primary address using custom logic, the following steps are taken:

1. **Run Inactive/Invalid logic** - Remove addresses that are Inactive or Invalid.
2. **Run the exclude logic** - Remove any addresses for primary consideration based on the exclude criteria you have defined.
3. **Run the custom condition logic** - If the first condition is met, then the address with the condition is the new primary address. If multiple addresses match the conditions, the address with the most matches is the new primary address. If the condition is not met, move on to the next condition.
4. **Run standard logic** - If multiple addresses match the current primary with the same number of matches, Network uses the standard logic conditions as a tie-breaker to recalculate the new primary address. Only the addresses that matched the custom conditions are considered.



Example 1

The custom logic conditions match on the following fields:

- Postal code - Supports 9-digit postal codes.
- Primary postal code - Supports 5-digit postal codes.

  Tip: When you use postal code as a condition, define a condition for both the `postal_code__v` and `primary_postal_code__v` fields to ensure that Network considers 5-digit and 9-digit postal codes.

- City

Address data	Current Primary	Address 1	Address 2	Address 3	Address 4
City	Fort York	Fort York	Fort York	Fort York	Fort York
Postal Code	45678-1234	45678-1234	45673-9865	45678-5462	45670-5476
Postal Code Primary	45678	45678	45673	45678	45670
Address Status	Active	Active	Active	Inactive	Active
Source Rank	3	3	3	3	3
Address Rank (ordinal)	2	2	3	1	1
Date/Time	2022-01-12 10:53:06	2022-04-10 09:22:11	2022-05-12 15:03:25	2022-02-08 02:44:11	2022-04-30 12:45:06
Veeva ID	243244003745596484	932184542274324385	932926364920187752	243244003745529263	243244003745596476

Result: Address 1 is recalculated as the primary address.

- Postal code is the same
- City is the same

Note that **Address 3** was excluded because it is Inactive.

Example 2

This example uses the same custom logic conditions as Example 1 but it has a different outcome because multiple addresses match the custom conditions.

The conditions match on the following fields:

- Postal code - Supports 9-digit postal codes.
- Primary postal code - Supports 5-digit postal codes.
- City

Address data	Current Primary	Address 1	Address 2	Address 3	Address 4
City	Fort York	Fort York	Fort York	Fort York	Fort York
Postal Code	45678-1234	45678-1234	45678-1234	45678-5462	45670-5476
Postal Code Primary	45678	45678	45678	45678	45670
Address Status	Active	Active	Active	Inactive	Active
Source Rank	3	3	3	3	3
Address Rank (ordinal)	2	2	3	1	1
Date/Time	2022-01-12 10:53:06	2022-04-10 09:22:11	2022-05-12 15:03:25	2022-02-08 02:44:11	2022-04-30 12:45:06
Veeva ID	243244003745596484	932184542274324385	932926364920187752	243244003745529263	243244003745596476

Result: Address 1 is recalculated as the primary address.

Address 1 tied with **Address 2** because they matched to the current primary on the custom conditions:

- Postal code is the same
- City is the same

The standard logic ran to break the tie. **Address 1** was selected because it had a lower address rank.

Note that **Address 3** was excluded because it is Inactive.

Supported fields

The following fields are available to use as conditions or to use to exclude addresses:

- Active address fields

  System fields (`created_date__v`, `vid__v`, `record_state__v`, and so on) are not supported

Logs

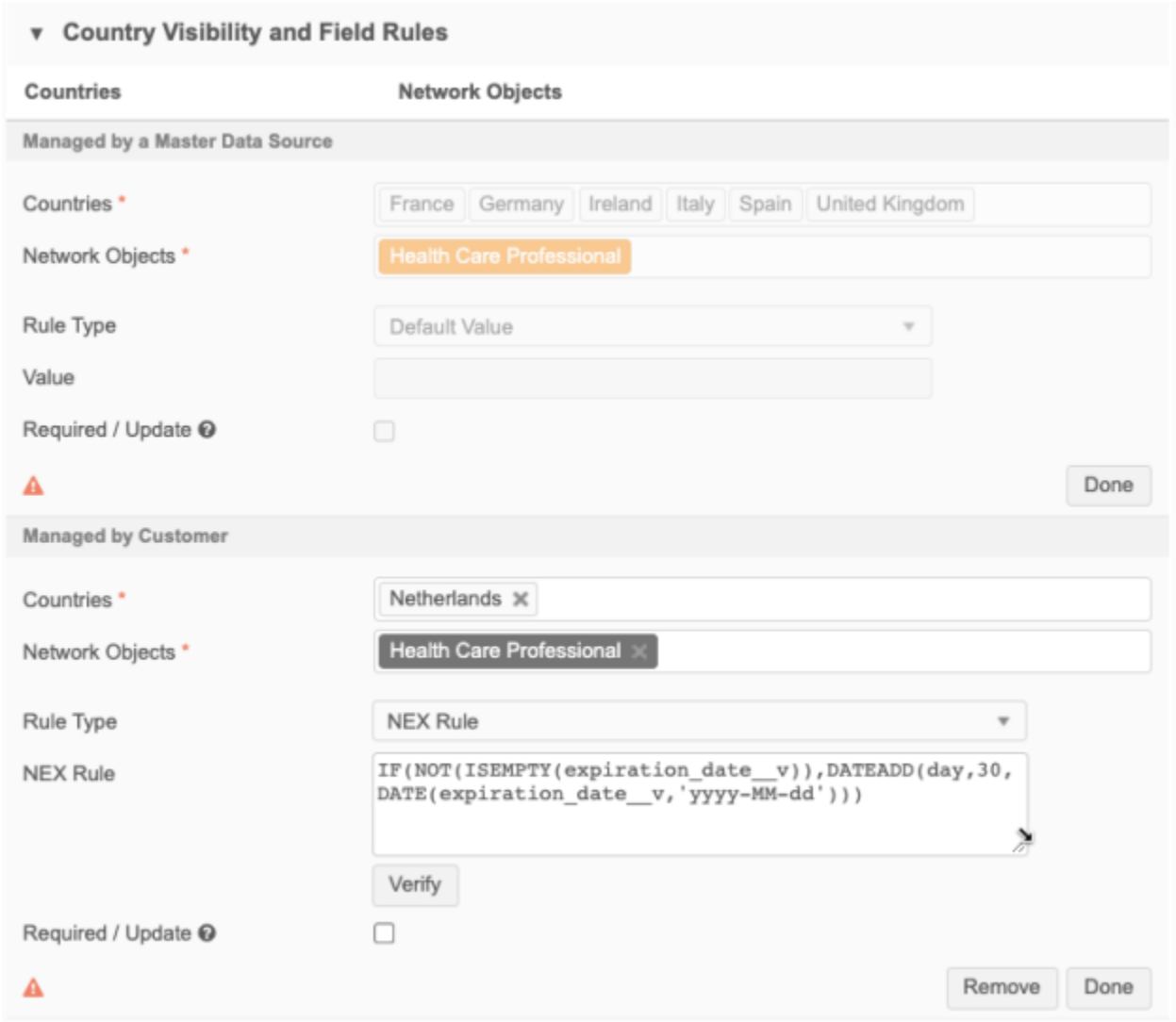
Changes to the primary address configuration are tracked in the System Audit Log.

FIELD CONFIGURATIONS

22R1.1

Enhancements have been made to data model field configurations so it is easier to manage default values and NEX rules for your Network instance.

In the **Country Visibility and Field Rules** section, the input boxes are bigger for the **Default Value** and **Nex Rule** types. The **NEX Rule** box can be resized so you can view the entire rule.



These enhancements are enabled by default in your Network instance. They apply to the **Managed by Customer** section for standard fields and custom fields.

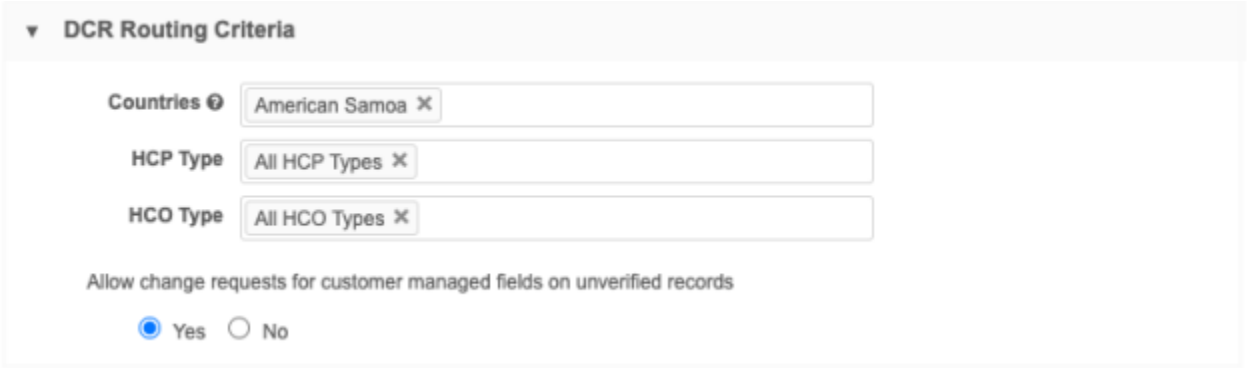
Tip: The NEX Rule box in source subscription configurations can also be resized.

Data sources

DCR ROUTING FOR THIRD PARTY SYSTEMS

22R1.1

Administrators and data managers can configure third party source systems to support data change requests (DCRs) for customer managed fields on unverified records. This will route change requests for customer managed fields to local data stewards. This is helpful when you want to add information to a new third party record but the record is still pending approval from the third party data provider. For example, if you want to add an HCP's email address on an unverified third party record, the DCR is rejected because the third party data provider is still processing the add request. Now, you can submit the DCR and local data stewards can process the request. When the third party record is approved, the local updates are merged.



This enhancement is available by default in your Network instance. Administrators must enable support for these DCRs in the third party system configuration.

DCR processing

When you enable support for DCRs on customer-managed fields, a DCR can be processed by local data stewards even when the add request has not yet been approved by third party data stewards.

Third party add request is approved

When local data stewards approve the DCR for customer-managed fields, an unverified (Under_Review) local record is created. When the add request is approved, the unverified local record is merged into the newly created third party record.

To ensure that local DCRs are processed quickly, administrators can configure the field-level change procedure on the data model field to automatically accept customer-managed field changes.

When auto-accept isn't enabled, the following issues can occur:

- Local tasks that are not processed in chronological order could overwrite newer data with older change requests. These local DCRs are not In Queue tasks so they might not be processed in order by local data stewards.
- If the local task is processed after the add request is approved by third party stewards, the local task will be repointed to the new third party record.

Third party add request is rejected

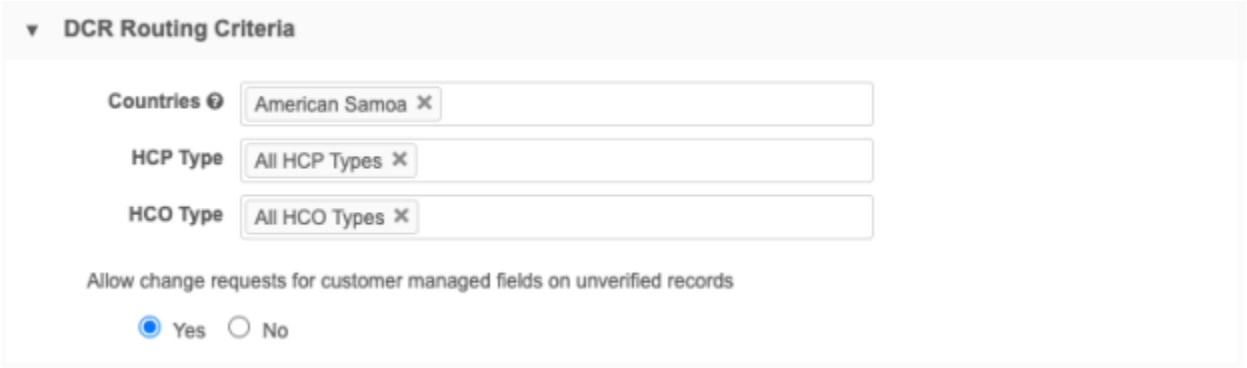
If the add request is not approved, any pending local DCR tasks are rejected (Invalid and Merged_Into records cannot be updated). If the local DCRs have already been approved, the unverified (Under_Review) local record is invalidated and any record profile changes are removed.

Allow change requests on locally managed fields

To configure these changes:

1. In the Admin console, click **System Interfaces > Systems**.
2. Click the name of a third party system.
3. In the **DCR Routing Criteria** section, click **Yes** below **Allow change requests for customer managed fields on unverified records**.

   When **No** is selected, the change request will be automatically rejected. The resolution note that is applied to the rejected task is `R-00014 System rejected - you are trying to update an Under Review record that is currently locked. Please resubmit your change request when the record is open for changes.`



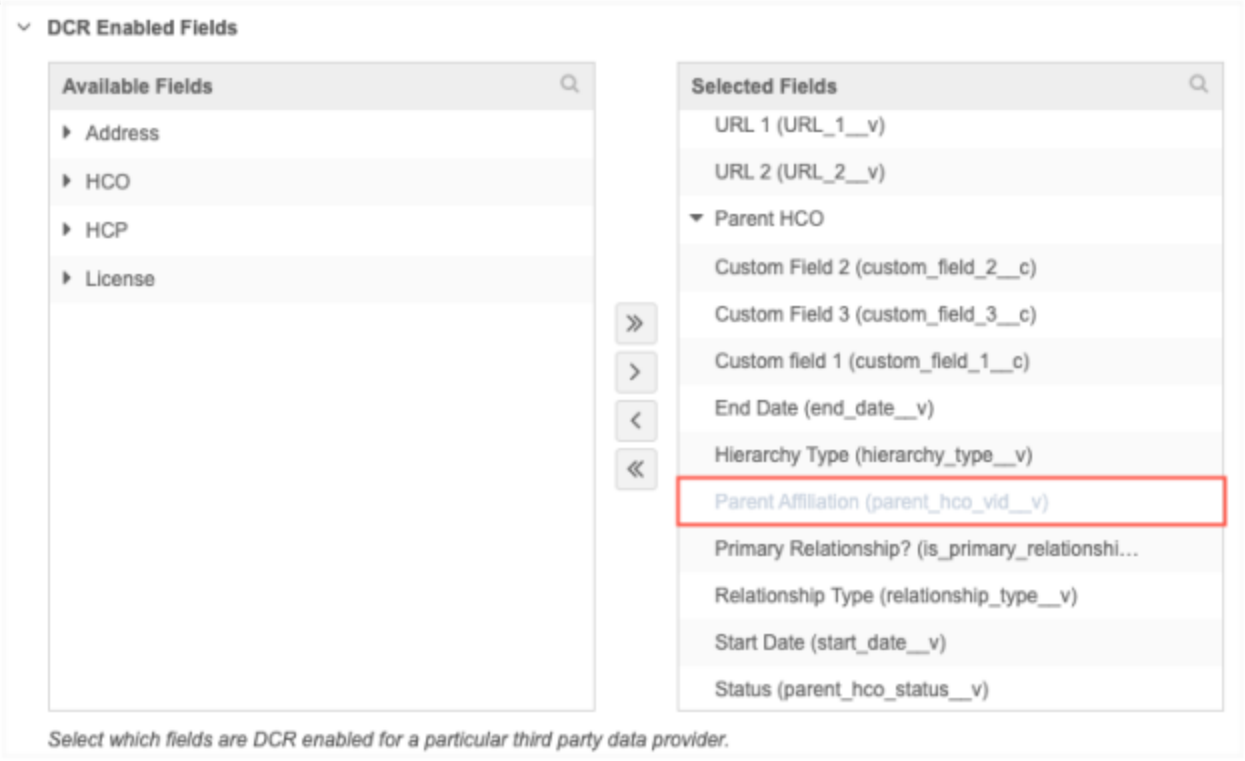
4. In the **DCR Enabled Fields** section, move the fields that you will allow to be processed by local data stewards to the **Selected Fields** column.
5. **Save** your changes.

PARENT HCO FIELD

22R1.1

After the `parent_hco_vid__v` field is moved to the **Selected Fields** pane, it cannot be removed unless the other Parent HCO fields are moved back to the **Available Fields** pane.

This behavior currently exists for HCP and HCO objects; when those objects are defined in the DCR Routing Criteria for a third party system, the HCO/HCP Type fields and the Primary Country field are automatically selected.



This enhancement is enabled by default in your Network instance.

OpenData subscriptions

EXPORT JOB ERROR LOGS

22R1.1

Administrators can now export the job error log after an OpenData country subscription runs to further investigate the issues.

This enhancement is enabled by default in your Network instance.

Exporting the log file

The log is available on any country subscription job that had errors.

To export the log:

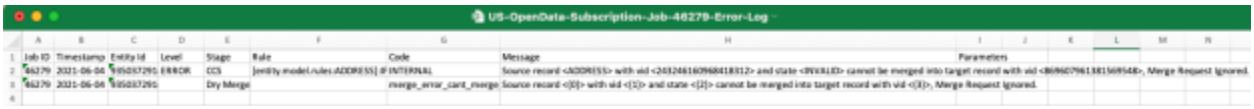
1. In an OpenData country subscription, click a job **ID** in the **Job History** section.
2. On the Job Details page, if errors occurred, the **Job Error Log** section displays. Click the **Export Error Log** button.

   The log downloads as a Microsoft® Excel® file to your local computer. The name format of the downloaded file is: `<country_code>-OpenData-Subscription-<Job_ID>-Error-Log.xlsx`.

   For example, an error log for a US OpenData subscription would be `US-OpenData-Subscription-Job-46279-Error-Log.xlsx`.

Error log details

The log contains a maximum of 500,000 rows. The errors do not display in any specific order.



The following columns are included in the log file:

- **Job ID** - The ID of the OpenData subscription job.
- **Timestamp** - The data and time the job ran.
- **Entity ID** - The Veeva ID (VID) of the object or sub-object.

  In some cases, an error occurs and the record cannot be loaded/processed so no VID can be found and cannot be displayed here. In these cases, the column will show the best 'identifier' information it can. For example, it might display `Address:7` to indicate an error was encountered for the seventh address record on a new, incoming record.

  In other cases, the error is due to something about the job and the fact it couldn't run or it was canceled. In those cases, the column will list the job ID in the first column (`@job:46279`).

- **Level** - The issue level: WARN, INFO, or ERROR.
- **Stage** - The processing stage that the error occurred.
- **Rule** - The Network rule that triggered the error. For example, `[entity.model.rules:ADDRESS]:IF( address_type__v == 'M', REJECT()).`
- **Code** - The Network error code.
- **Message** - An explanation of the error.
- **Parameters** - VID and object information, when applicable.

Source subscriptions

TRANSFORMING INBOUND DATA

22R2

Network can now transform data before loading it into Network. Previously, transformation queries were available to use for data that you were exporting to downstream systems. Now you can use the transformation queries to pre-process data before it is processed and loaded during source subscription jobs.

This feature is enabled by default in your Network instance.

Examples of data transformations

Use transformation queries to pre-process inbound data. For example:

- **Process the output of a report**

  Example: Run a report to find HCPs that are flagged for opt out, but that are not opted out yet. The output of that query would then be the input file for the source subscription to opt out these HCPs.

- **Process dynamic attributes from Veeva CRM**

  Example: This requires pivoting the data that is coming from CRM (converting rows into columns).

- **Simplify data feeds and model maps**

  Example: If a source system provides a single file containing many entities, split the data into individual files so the model map can be simplified.

- **Define field values based on certain conditions**
- **Split or concatenate fields**

How transformation queries work

Use transformation queries for the following activities:

- Transform files before they are loaded by source subscriptions.
- Use a report to generate the input for your source subscription.
- Join incoming source files loaded by source subscriptions with tables (custom tables, reporting tables, lookup tables) from the reporting database.

Note: If you use the query to read from the reporting database, all applicable records from all countries and with all record states will be included in the inbound file; data visibility profiles are not considered and records are not filtered by their record state.

Query process overview

Transformation queries are SQL based (SELECT statements). Test them in the SQL Query Editor, create the query on the Transformation Query page, and then link them to source subscriptions.

1. Upload source files as custom tables so you can test your query.
2. Write and test your query in the SQL Query Editor (**Reports**).
3. Create a transformation query (**System Interfaces**). Copy and paste the tested SQL query from the SQL Query Editor into the transformation query configuration.
4. Link the transformation query to a source subscription.

   When the source subscription runs, the following process occurs:

   - **Apply queries** - Transformation queries preprocess the source files from the inbound FTP folder.
   - **Process files** - The source files and query output file are processed by the source subscription, as defined by the source subscription, and the data is loaded into Network.

Supported source files

Source files are created as temporary tables when they are read by transformation queries during the subscription job. File names and column names (header rows) are validated to ensure that they comply with the required naming conventions.

- Contain only alphanumeric characters (0-9, a-z, A-Z) and underscore (_) characters.
- File names must start with an alphabetic (a-z, A-Z) or underscore character. Column names can start with a number (0-9).
- Must be between 1 and 100 characters.
- Source files must use delimited formats. Files with fixed length formatting will be skipped.
- Source file names can be reserved SQL keywords (for example, `order.csv` and `new.csv`).
- Column names can be reserved SQL keywords but they must be in double quotes("") in the transformation query. (for example, `select "select" from test_table_csv`).
- Column names must be unique in each source file.

Source files are not validated until the source subscription job runs.

- If the column names do not comply with the naming requirements, the job will fail.
- Source files that uses a fixed length format are skipped by transformation queries.

Example data transformation scenario

We'll use the following example to walk through the steps for this feature.

Scenario

We need to split up a complex source file from our data provider, Pharma. The file contains multiple product entities (Market Basket, Brand, Product) in one row. We'll use transformation queries to create individual files for each entity so the model map is simplified in the source subscription.

Example file

pharma_data

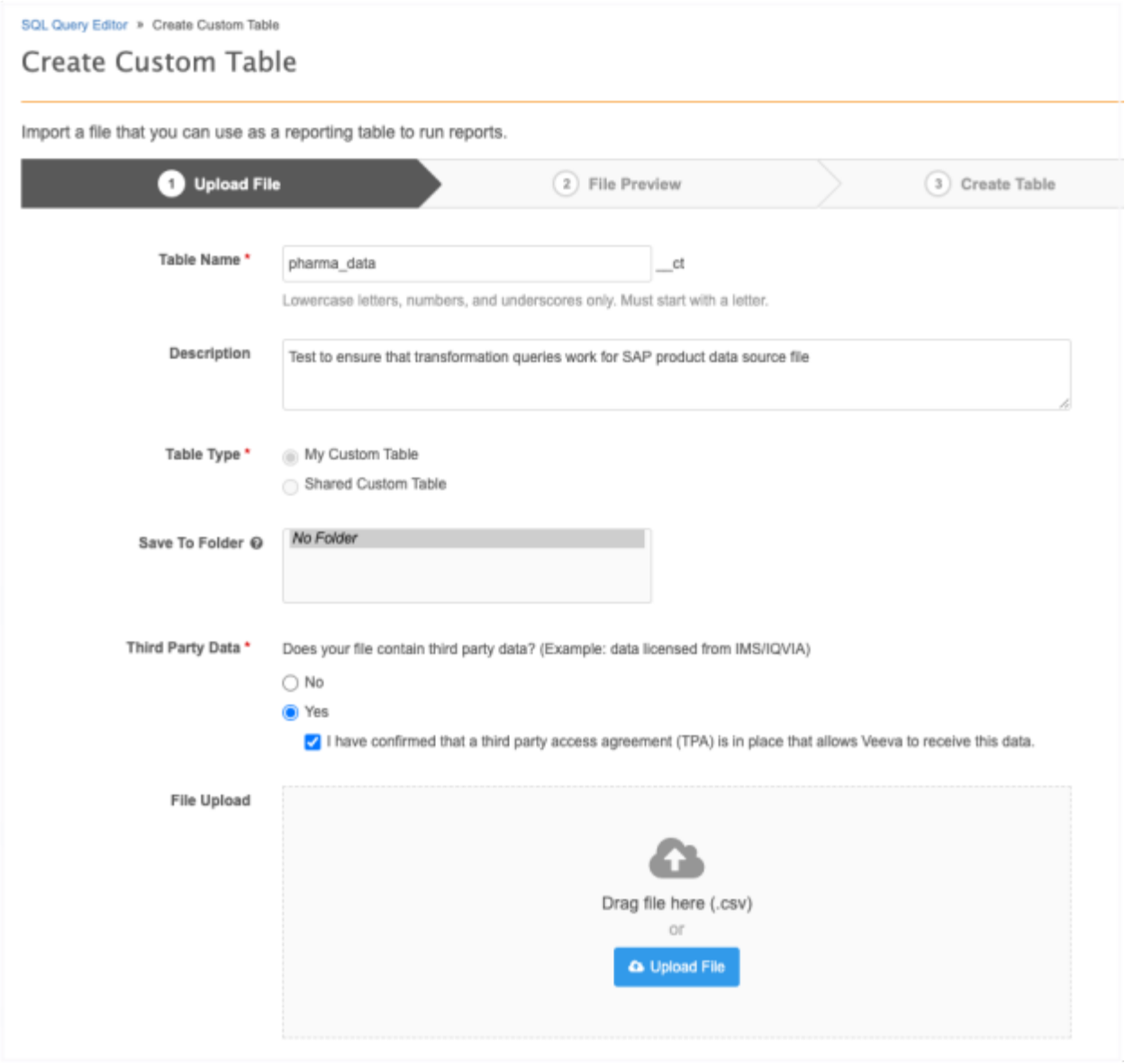
mid	marketbasket	bid	brand	parent_mid	pid	product	parent_bid
m1	Cholesterol Reducer	b1	Cholecap	m1	p1	Cholecap Tablet 25mg	b1
m1	Cholesterol Reducer	b1	Cholecap	m1	p2	Cholecap Tablet 50mg	b1
m1	Cholesterol Reducer	b1	Cholecap	m1	p3	Cholecap Tablet 75mg	b1
m1	Cholesterol Reducer	b1	Cholecap	m1	p4	Cholecap Tablet 100mg	b1
m2	Neurology	b2	Restolar	m2	p5	Restolar Liquid 5mg	b2
m2	Neurology	b2	Restolar	m2	p6	Restolar Liquid 10mg	b2
m2	Neurology	b2	Restolar	m2	p7	Restolar Liquid 15mg	b2
m2	Neurology	b2	Restolar	m2	p8	Restolar Liquid 20mg	b2

Upload source files as custom tables

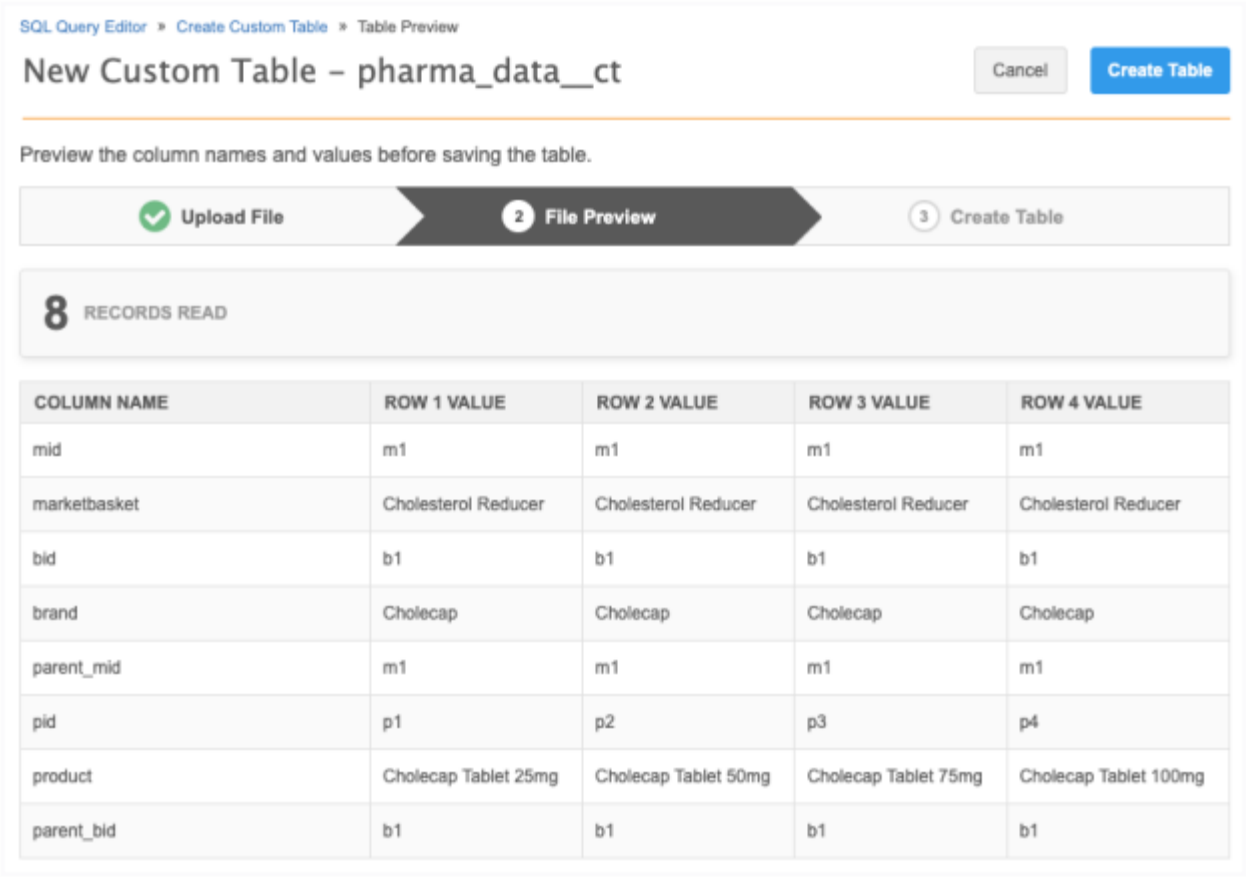
Transformation queries for inbound files cannot be tested until runtime because the inbound table doesn't exist yet. To test the query to ensure that it is valid, create a custom table based on the source file.

To create a custom table:

1. On the Network menu bar, click **Reports > SQL Query Editor**.
2. In the tree view, beside **My Custom Tables**, click **Create > Custom Table**.
3. On the **Upload File** step, define the required information and upload the sample file.



4. On the **File Preview** step, review the data. Click **Create Table**.



COLUMN NAME	ROW 1 VALUE	ROW 2 VALUE	ROW 3 VALUE	ROW 4 VALUE
mid	m1	m1	m1	m1
marketbasket	Cholesterol Reducer	Cholesterol Reducer	Cholesterol Reducer	Cholesterol Reducer
bid	b1	b1	b1	b1
brand	Cholecap	Cholecap	Cholecap	Cholecap
parent_mid	m1	m1	m1	m1
pid	p1	p2	p3	p4
product	Cholecap Tablet 25mg	Cholecap Tablet 50mg	Cholecap Tablet 75mg	Cholecap Tablet 100mg
parent_bid	b1	b1	b1	b1

5. The table is added to the Custom Table section in the tree view.

Write and test the transformation query

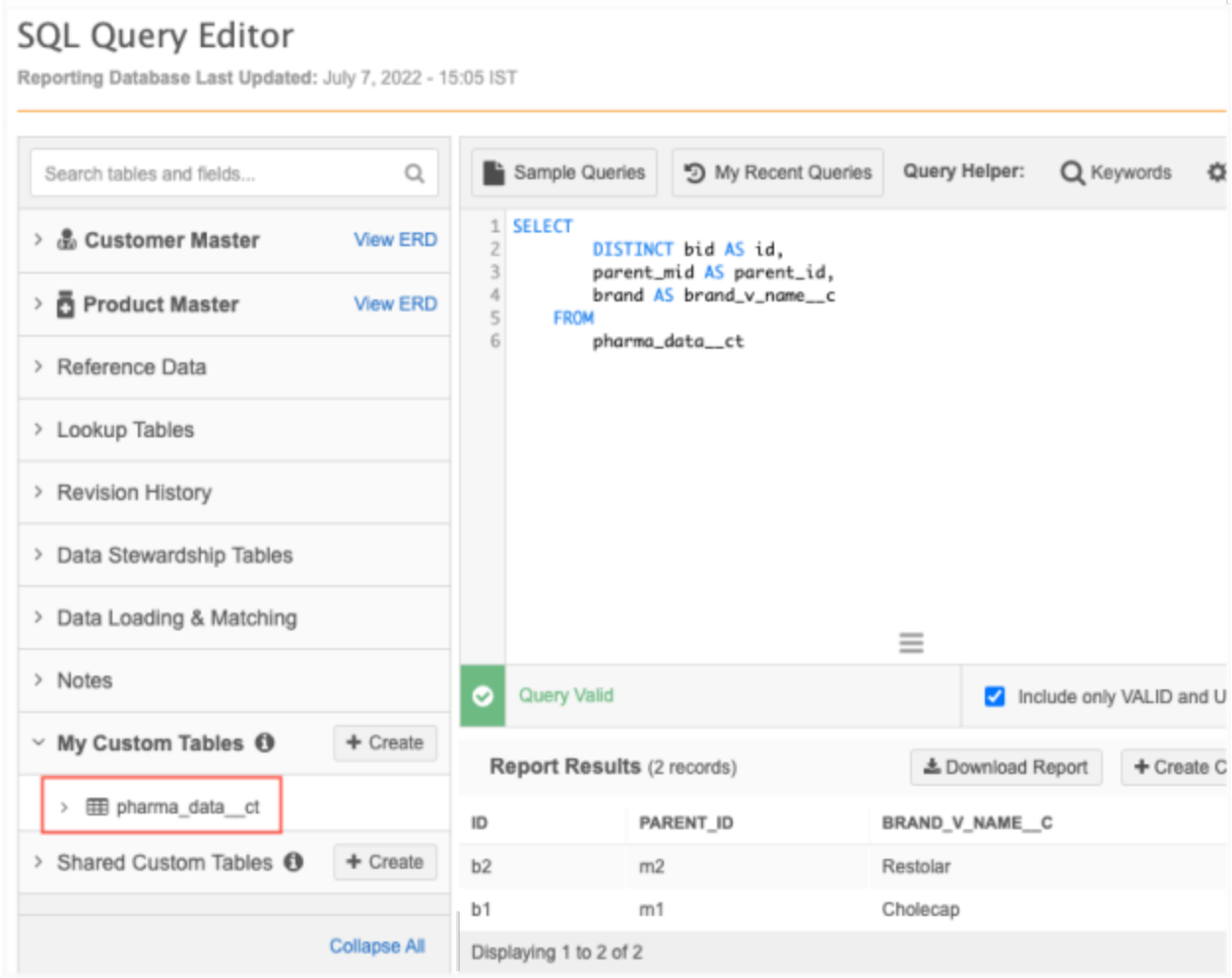
In the SQL Query Editor, write the transformation query. Use the custom table so the SQL syntax and field names can be validated and to ensure that it performs the data transformation as expected.

Example query

This query transforms the data in the incoming feed by creating a separate file for the Brand entity:

```
SELECT
        DISTINCT bid AS id,
        parent_mid AS parent_id,
        brand AS brand_v_name__c
    FROM
        pharma_data__ct
```

Note that instead of the source file name, `pharma_data.csv`, the query uses the custom table name, `pharma_data_ct`, to ensure that the query is valid. When we create the transformation query in the next step, we'll replace the custom table name with the source file name.

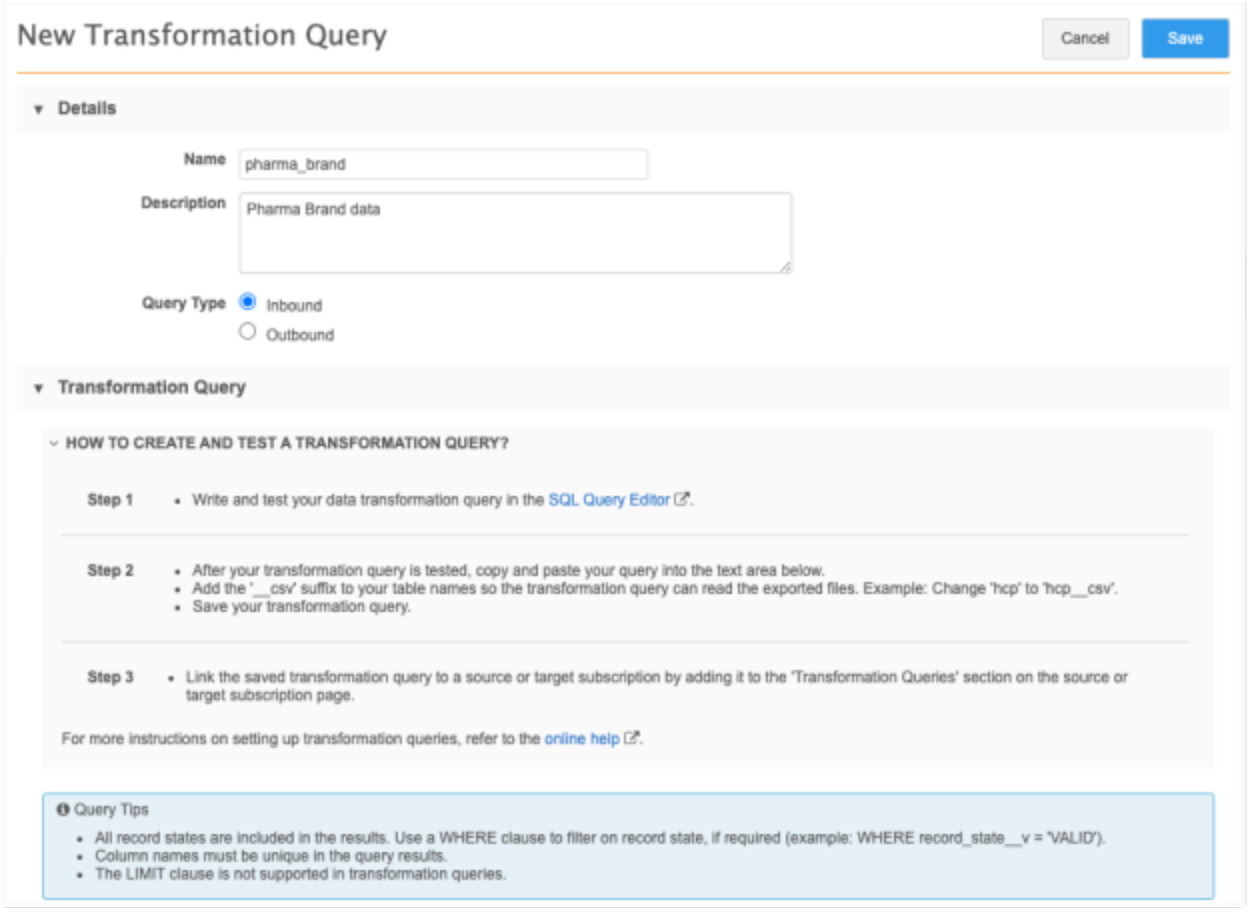


Create a transformation query

After the SQL query is tested, create the transformation query.

For our example scenario, we'll create a transformation query for each entity (Market Basket, Brand, and Product) that we want to separate from the `pharma_data.csv` file.

1. In the Admin console, click **System Interfaces > Transformation Queries**.
2. Click **Add Query**



3. Type a **Name** and **Description**.

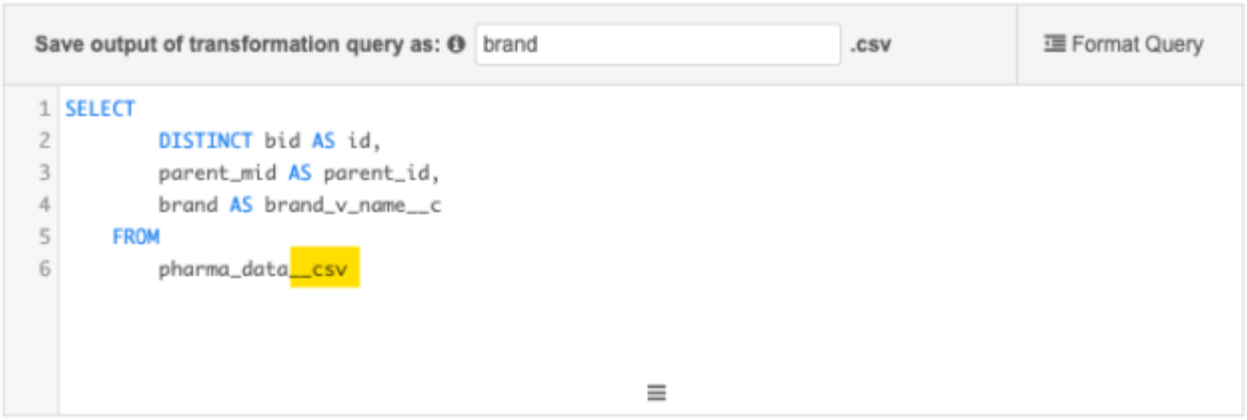
   Using our example, we'll create the transformation query to separate the Brand entity data into its own file.

4. Choose the **Inbound Query Type**.

   The **Outbound** query type is used to post-process data before it is exported using a target subscription.

   Note: After the query is saved, the **Query Type** cannot be changed.

5. Paste the tested SQL query into the query box.



Query considerations

- **Table names** - If you tested the query using a custom table, replace the `__ct` suffix on the table name with the `__csv` file extension if you want to pre-process a feed loaded by the source subscription. To read directly from a table in the reporting database, do not add the `__csv` extension.

  Inbound queries can query from any table as long as it exists in reporting or ends with the `__csv` extension.

- **Unique column names** - The source file must contain unique column names. Identical column names are valid in the SQL Query Editor for reports but must be changed here so the transformation query can be saved.
- **Record state** - All record states are included in the results if the transformation query reads directly from the table in the reporting database. You can use a WHERE clause to filter records; for example, to only include valid records in the results, add `WHERE record_state__v = VALID` to your query.
- **Number and date fields** - When transformation queries process data that is loaded by the source subscription, all fields are internally treated as `varchar` fields. So, if your transformation query is intended to aggregate numbers or to do calculations on number or data fields, use the `CAST` function to convert the data type from `varchar` to the correct data type. This prevents issues because the export file processing uses `varchar` as the data type. For more information about `CAST`, see SQL functions in the *Veeva Network Online Help*.
- **`LIMIT` clause** - The `LIMIT` clause is not supported for transformation queries. If you want to limit the number of rows returned, wrap another query around the query that has the `LIMIT` clause.

6. Type a name for the transformation query output file. The file name is automatically appended with `.csv`.

Supported file naming conventions:

- can contain numbers (0-9), letters (a-z, A-Z), and underscore (_) characters
- must begin with a letter or underscore (_)
- cannot contain more than 100 characters
- cannot be any SQL keyword
- cannot be `reference.csv`

Note: The query output file will not be saved to the `inbound` FTP folder where the source subscription reads the source files. It can be downloaded from the Job Details page after the source subscription job runs.

7. **Save** your query.

   When the transformation query is saved, Network validates the query; for example, to ensure that column names are unique, to check if the query contains the LIMIT clause, or if the table names and output files are valid. If warnings display, fix the issues and save your changes.

For our example, we'll create transformation queries for the Market Basket and Product entities also.

Market Basket query

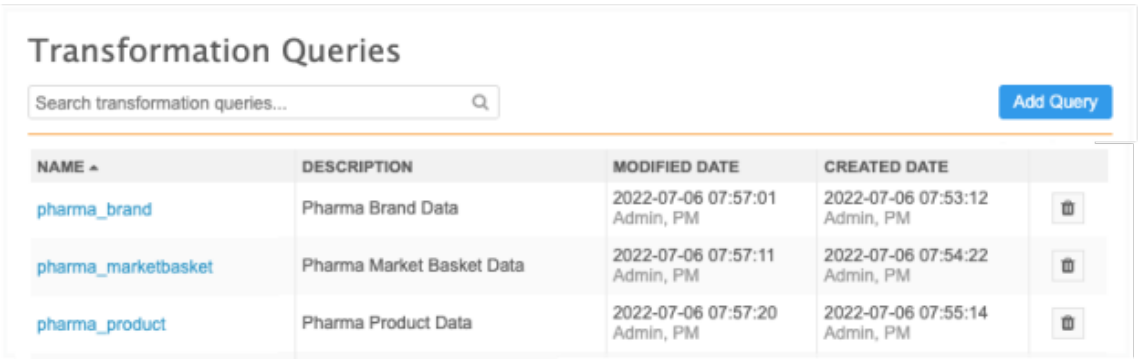
```
SELECT
          DISTINCT mid AS id,
          marketbasket AS marketbasket_v_name__c
     FROM
          pharma_data__csv
```

Product query

```
SELECT
          DISTINCT pid AS id,
          product AS product_v_name__c,
          parent_bid AS parent_id
     FROM
          pharma_data__csv
```

Saved transformation queries

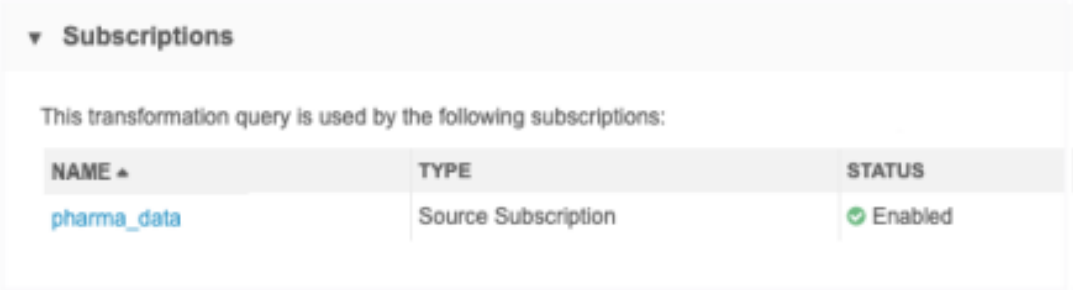
After you save a transformation query, it is listed on the Transformation Queries page. Each query row contains the name, description, and the modified and created dates by user.



Edit transformation queries

Click the transformation query to open it and view the details or make changes. The **Query Type** and **Code** cannot be edited.

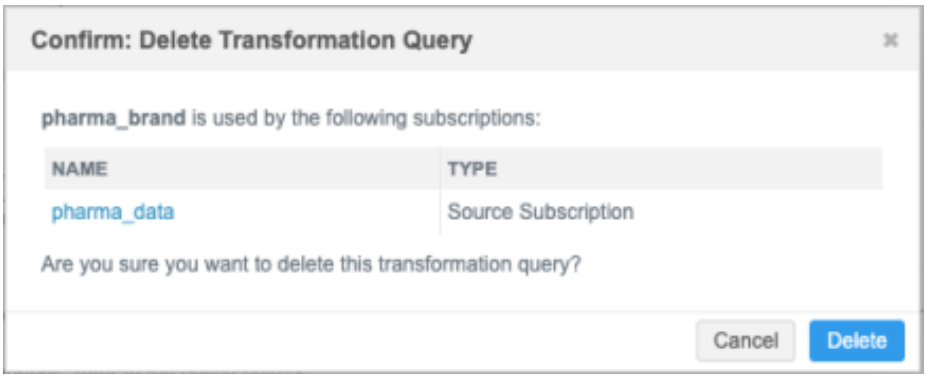
The **Subscriptions** section displays any source subscriptions that use that transformation query.



Delete transformation queries

If the query is no longer needed or was created by mistake, click the **Delete** icon to remove it from your Network instance.

The dialog displays any linked subscriptions . Click **Delete** to confirm.



When a query is deleted, any related source subscriptions are also updated to remove the link to the query.

Configure source subscriptions to load files

After the transformation query is created, add it to a source subscription so you can pre-process the inbound files.

Create or edit a source subscription

1. In the Admin console, click **System Interfaces > Source Subscriptions**.
2. Open an existing subscription or click **Add Subscription** to create a new one.

> **Note:** Transformation queries are supported for **Classic Mode** only. If you create a subscription using **Wizard Mode**, you can add it to the subscription after the configuration is saved.

3. **Details** section
    - Type a **Name** and **Description** for the subscription.
    - **System** - Choose the source system.
    - **Status** - The subscription is enabled by default.
4. **Settings** section
    - **Apply Updates and Merge** - Enable this option to update records based on data loaded by this subscription. If this option is not set, the subscription runs without changes to existing data. Transformation queries run during the parse stage of the job, so you can clear this option to test your subscription and transformation queries and view the results in the job log.
    - Accept all of the other default options in this section.

    For detailed information about the settings in these sections see the Add a source subscription topic in the *Veeva Network Online Help*.
5. **Source Files** section
    - **Network FTP Path** - Identify the inbound folder where the source files are located.

      Note: If the transformation query reads directly from the reporting database, you still need to specify an inbound folder, but it will be empty.

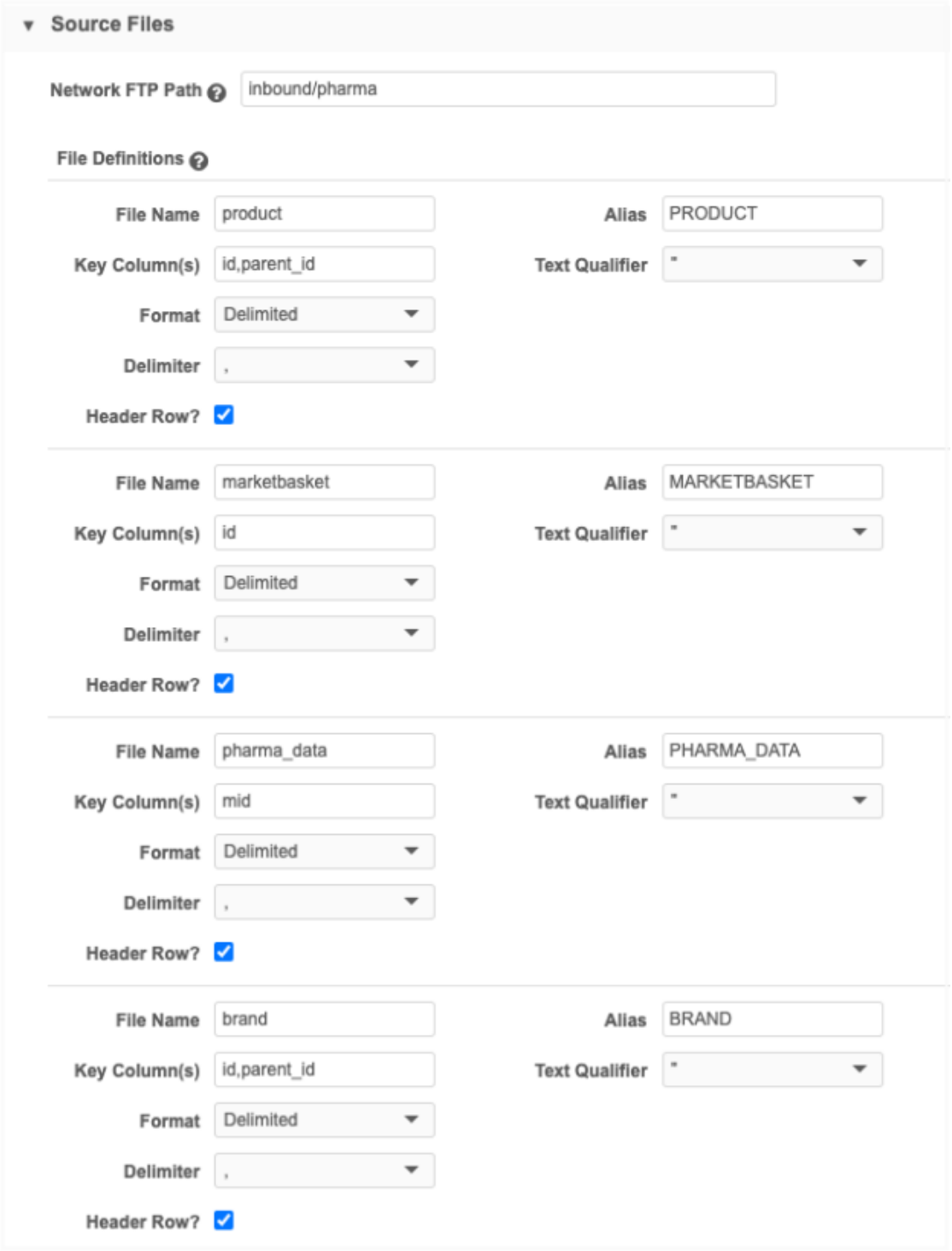
    - **File Definitions** - Click **Add File** to provide the following information about each file that you want to process through the source subscription.

      Note: Include all of the source files that will be processed by transformation queries (even if it isn't used by the model map) so Network knows which delimiter and text qualifier to use when parsing the file.

        - **File name** - The name of the file. Do not include the `.csv` extension.
        - **Alias** - The simplified name of the file that is used to reference it in the model map.
        - **Key Columns** - Type the keys from the source system, delimited by commas (,).
        - **Text Qualifier**: Select a qualifier to use for the beginning and end of text.
        - **Format**: Choose **Delimited**. Files with **Fixed Length** formats are not supported for data transformations; they will be skipped during the job
        - **Delimiter**: Select the delimiter that is used to separate text in the source file.
        - **Header Row?**: Select the checkbox to indicate that a comma separated list of headers exist for the header row.

Example file definitions

In this example, we've added definitions for the source file (pharma_data.csv) and a file for each of the output files for entities (Market Basket, Brand, Product) that we want created from the transformation queries.

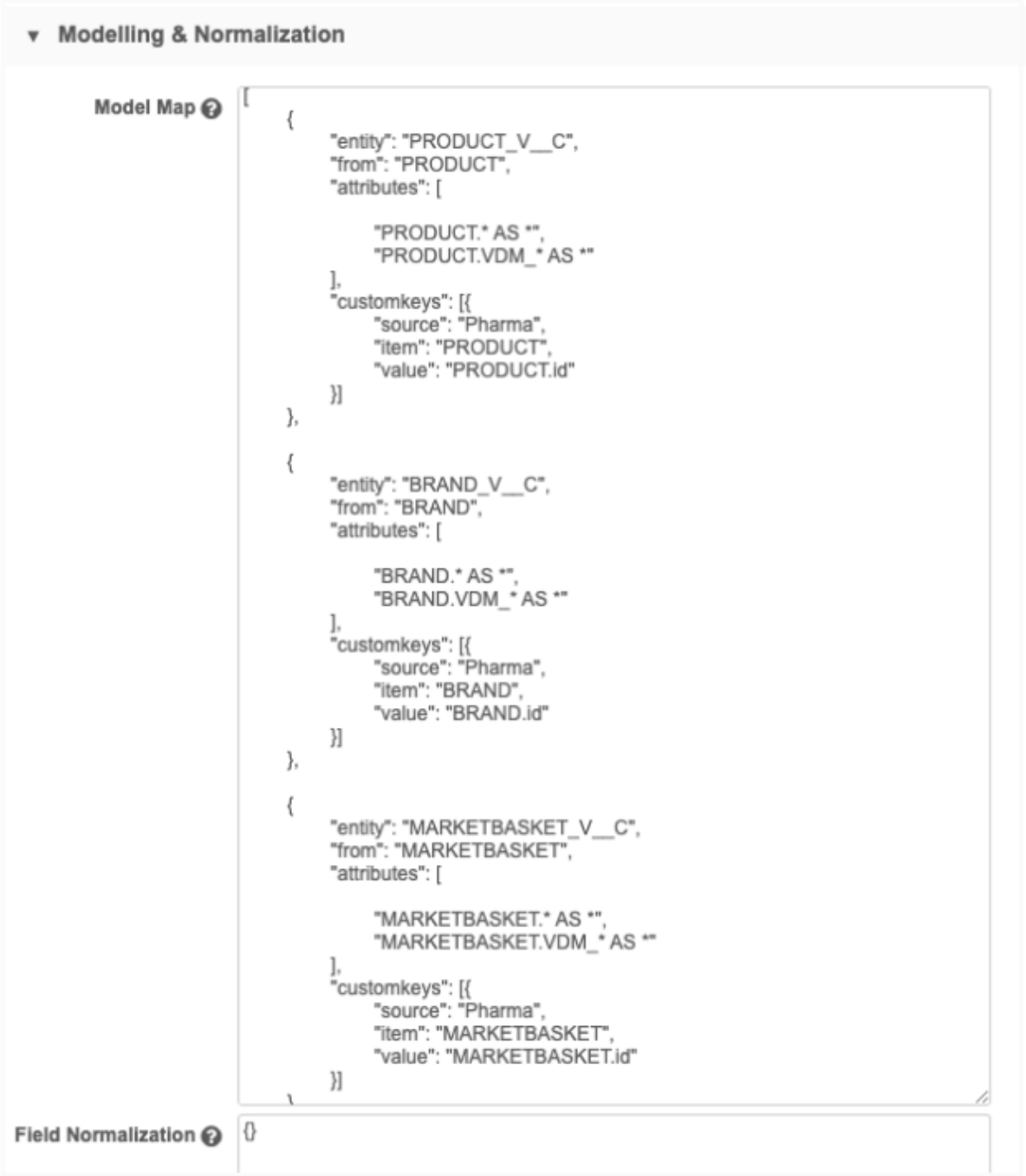


If the output file is not used in the model map (for example, if the output file of the first transformation query is the input file for the next transformation query) the file definitions for the query output files are not mandatory.

If the file definitions for the query output file are not defined, double quotes (") are used as the default text qualifier and comma (,) is used as the default delimiter.

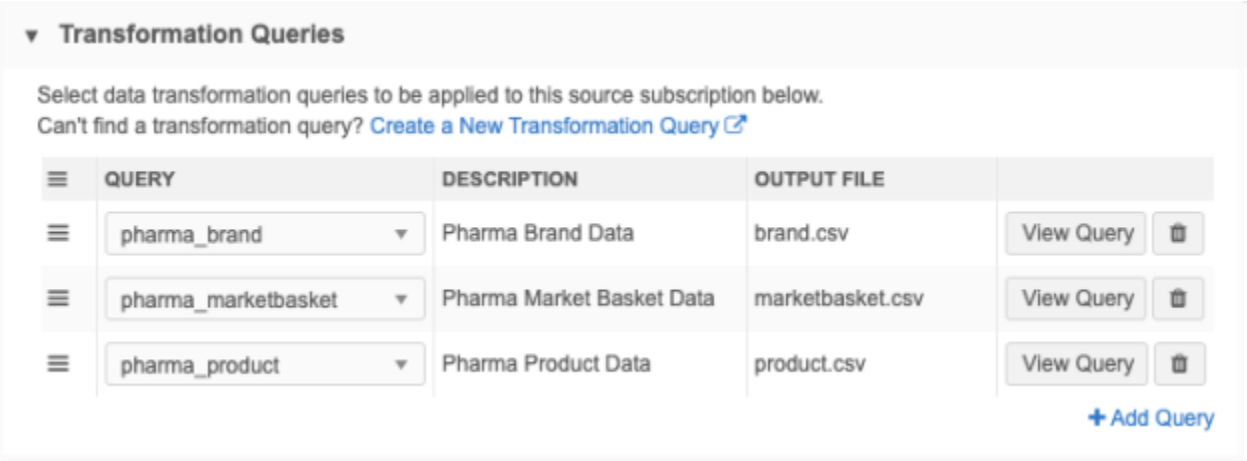
6. **Modelling & Normalization** - Add a model map and any field normalizations. The transformation query output files are used in the model map; the original source file is not.

   Example model map



7. **Network Expression Rules** - Add any NEX rules that you want to apply to the subscription job.
8. **Match Configuration** - Define any match rules by country for this subscription.
9. **Transformation Queries** section - Click **Add Query** to add a transformation query to the subscription. Only inbound queries display in the list.

You can apply multiple queries to each subscription. The queries will always run in sequence as they are listed in the subscription. Change the order of the queries using the **Handle** icon.



10. **Job Trigger Configuration** - Optionally add a schedule and any emails or jobs that should be triggered for this subscription. For example, you can trigger a target subscription to export the data to your downstream system.
11. **Advanced properties** (*optional*) - Transformation queries make all column headers lower case. For example, if your source file has a column "AMS_id", then this column will change to "ams_id" when processed and outputted by a transformation query. This can be relevant for key columns since they are case sensitive by default. To avoid data loading issues, click **Advanced Mode** and add the following advanced property to force all incoming column headers to be lower case:

```
"parser.attribute.rules": "network:force_lowercase_attributes"
```

12. **Save** your changes.

Data Flow View

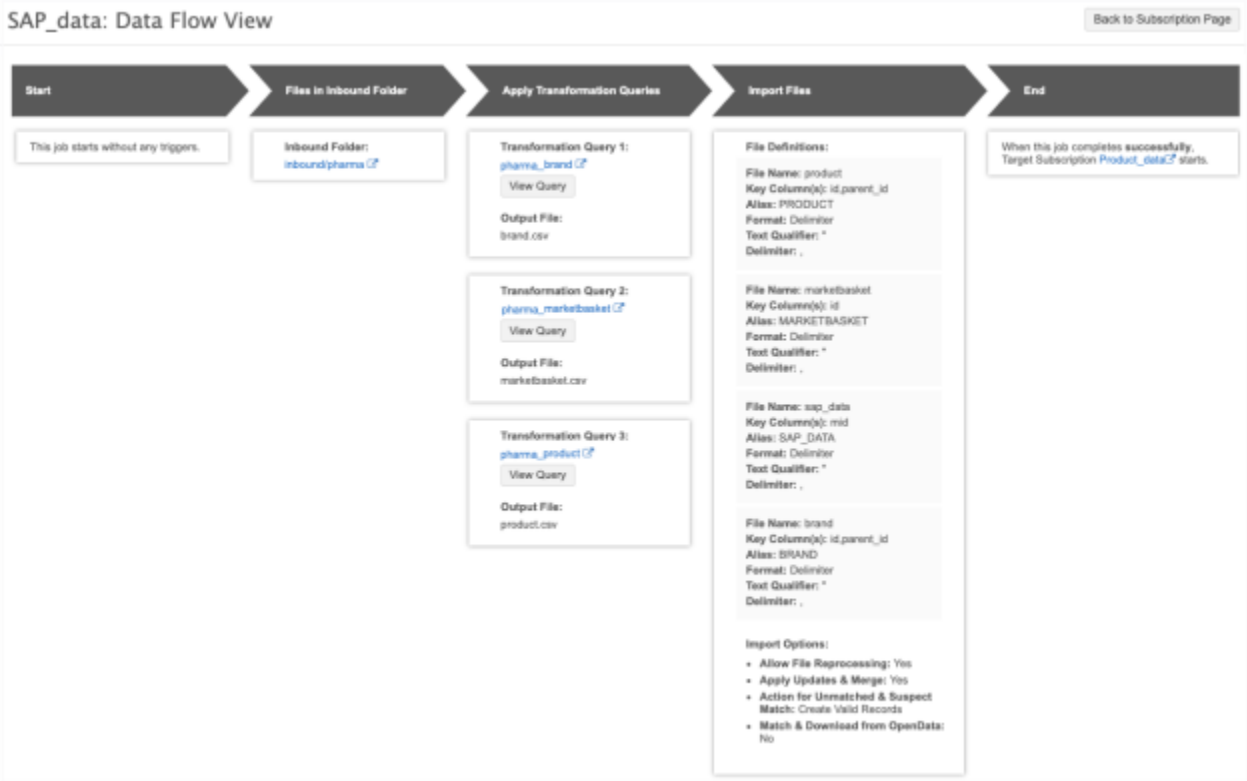
After you save a source subscription, a new **Data Flow View** is added to the **Details** section to visualize all of the steps of the job.

Click the **Data Flow View** thumbnail to open the view.

These stages of the job are defined:

- **Start** - Indicates if the job has been triggered by another job.
- **Files in Inbound Folder** - The file path of the `inbound` folder. Click the link to open the File Explorer in a new tab.
- **Apply Transformation Queries** - Details about each query that is applied to the subscription. If there are multiple queries, they are listed in the order that they run. Click the query name to open the transformation query configuration. Click **View Query** to see the query. This step does not display if transformation queries are not applied.
- **Import Files** - The file definitions for each file that is loaded. The **Import Options** section defines the options that are set in the source subscription configuration.

- **End** - Indicates if this job triggers email notification or another job.



To return to the job configuration, click **Back to Subscription Page**.

Unsaved changes

If you have made changes to the source subscription configuration, the **Data Flow View** does not reflect those changes until the subscription is saved. A message displays if you open the view before saving the subscription.

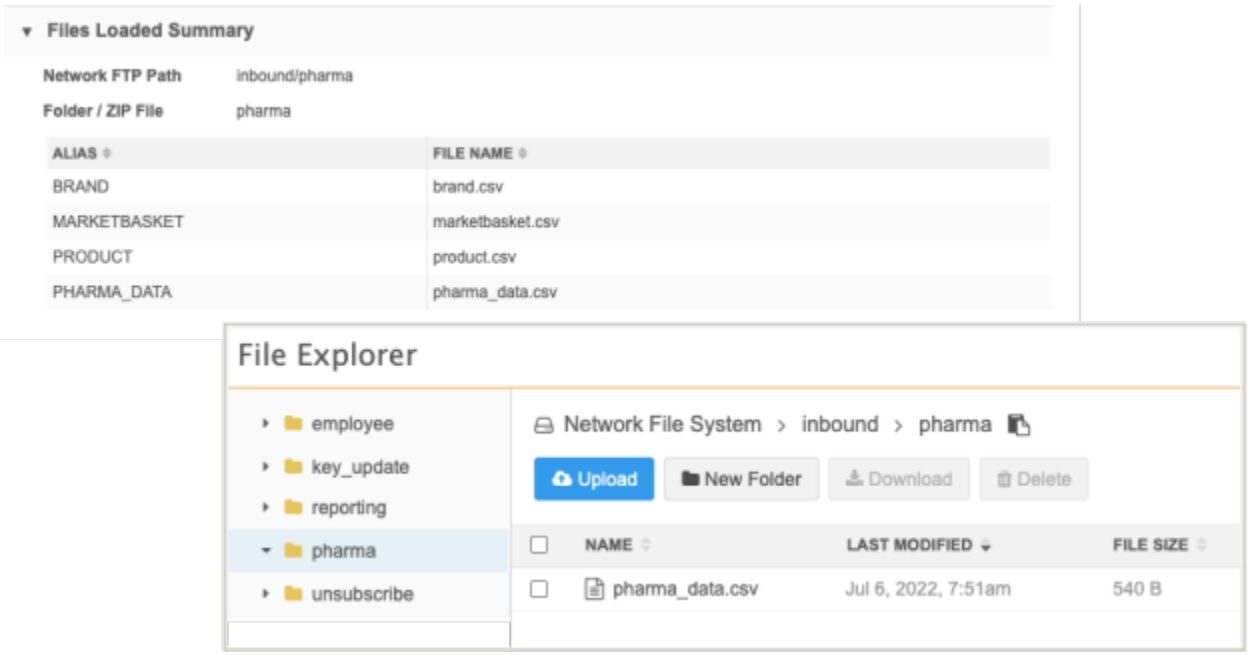
Job details

After a source subscription completes, review the Job Details page.

Files Loaded Summary

This section displays the files that were processed during the job. The query output files are listed in this section but they are not added to the inbound FTP folder with the original source file. You can download the output files in the **Transformation Queries** section on the Job Details page.

Example

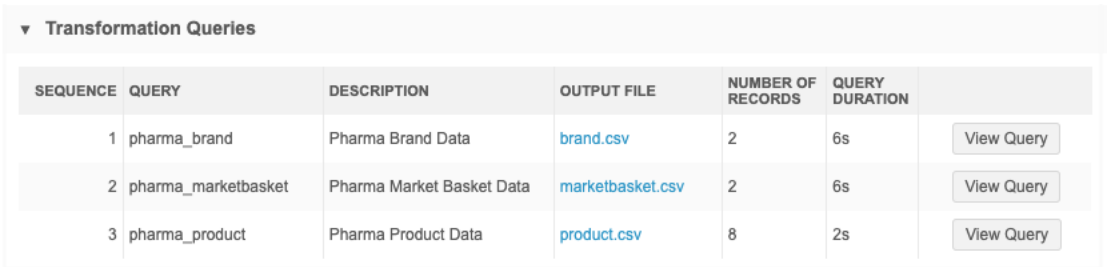


Transformation Queries

This section displays if one or more queries was applied to the job. If multiple transformation queries were applied, they display in the sequence that they ran.

Details

- **Query** - The transformation query name.
- **Description** - A description of the query.
- **Output File** - The file created by the query. Click the file name to download it.
- **Number of Records** - The number of records that were returned by the transformation query.
- **Query Duration** - The runtime of the query during the job. For example, `2s` means that the query runtime was 2 seconds.
- **View Query** - Click to display a snapshot of the query that was applied when this job ran. It might be different than the query that is currently saved in Network. Viewing the query as it was at run-time can help you to troubleshoot any issues that might have occurred.

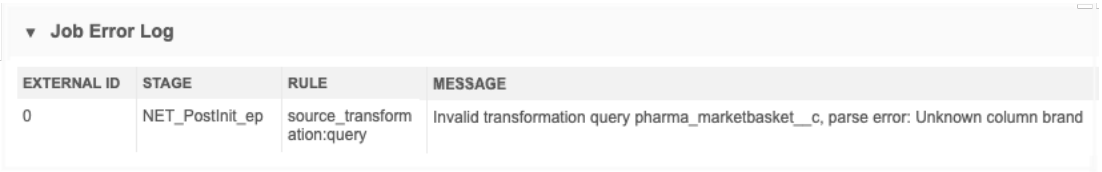


Job errors

Source subscriptions fail if the transformation query creates a runtime error because of an invalid or incorrect SQL statement. An error is logged to help you to troubleshoot.

Example

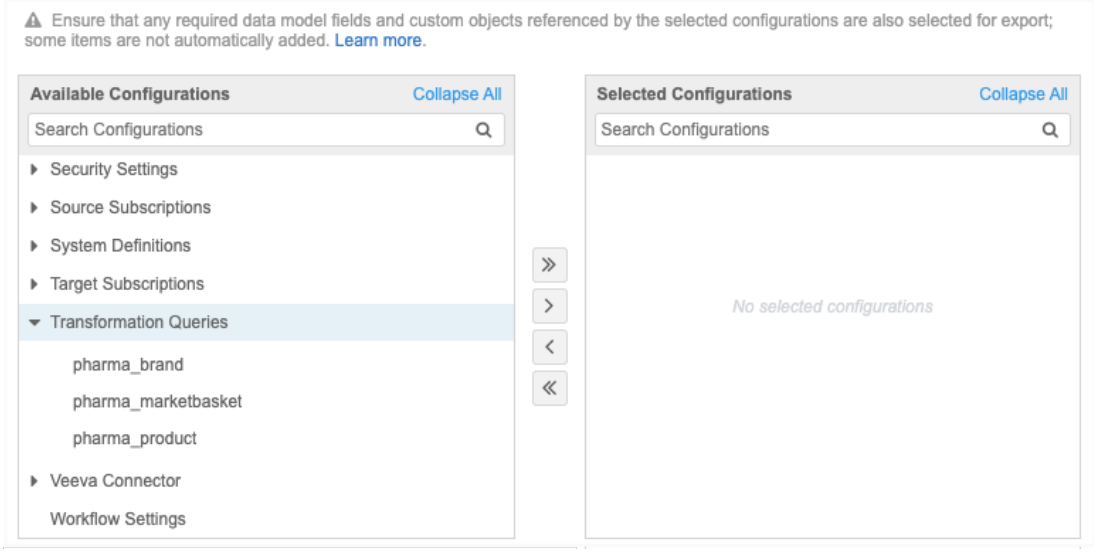
If the source file names have unsupported characters or duplicate column names, an `Unknown column` error occurs.



Exporting configurations

Administrators can move transformation queries to target environments using export packages (**Settings > Configuration Export**). For example, you might create and test transformation queries in your Sandbox instance and then move them to your Production instance.

When you create a package, expand the **Transformation Queries** category and choose the queries that you want to add to the package.



SOURCE FILE COLUMN HEADERS

22R1.1

Administrators and data managers can use a new property rule to format the column headers that are loaded to use either lowercase (default) or uppercase letters. Currently, the column headers for source subscriptions are case-sensitive which can cause data loading issues.

This feature is not enabled by default.

Formatting case for column headers

To add case formatting to column headers:

1. Open the source subscription configuration and click **Advanced Mode**.
2. In the **Module Properties** field, add the following property:

   ```
   "parser.attribute.rules": "network:force_lowercase_attributes"
   ```

   or

   ```
   "parser.attribute.rules": "network:force_uppercase_attributes"
   ```

3. **Save** your changes.

When the subscription runs and the files are loaded, the column header values will be formatted to lowercase or uppercase letters.

Considerations for attributes

If you reference column headers in the source subscription configuration (for example, in NEX rules, Model Map, Field Normalization, or Transformation Queries sections), manually edit those references so they use the same case as the column headers in the file (uppercase or lowercase).

Integrations

SAP CONCUR AUTHENTICATION

22R1.1

The SAP Concur Connector will be updated to use OAuth2 authentication in this release. The Network Concur Connector is currently using an API for Veeva OpenData downloads that SAP Concur is decommissioning on June 30, 2022. Using OAuth2, Network can access SAP Concur and continue using that function.

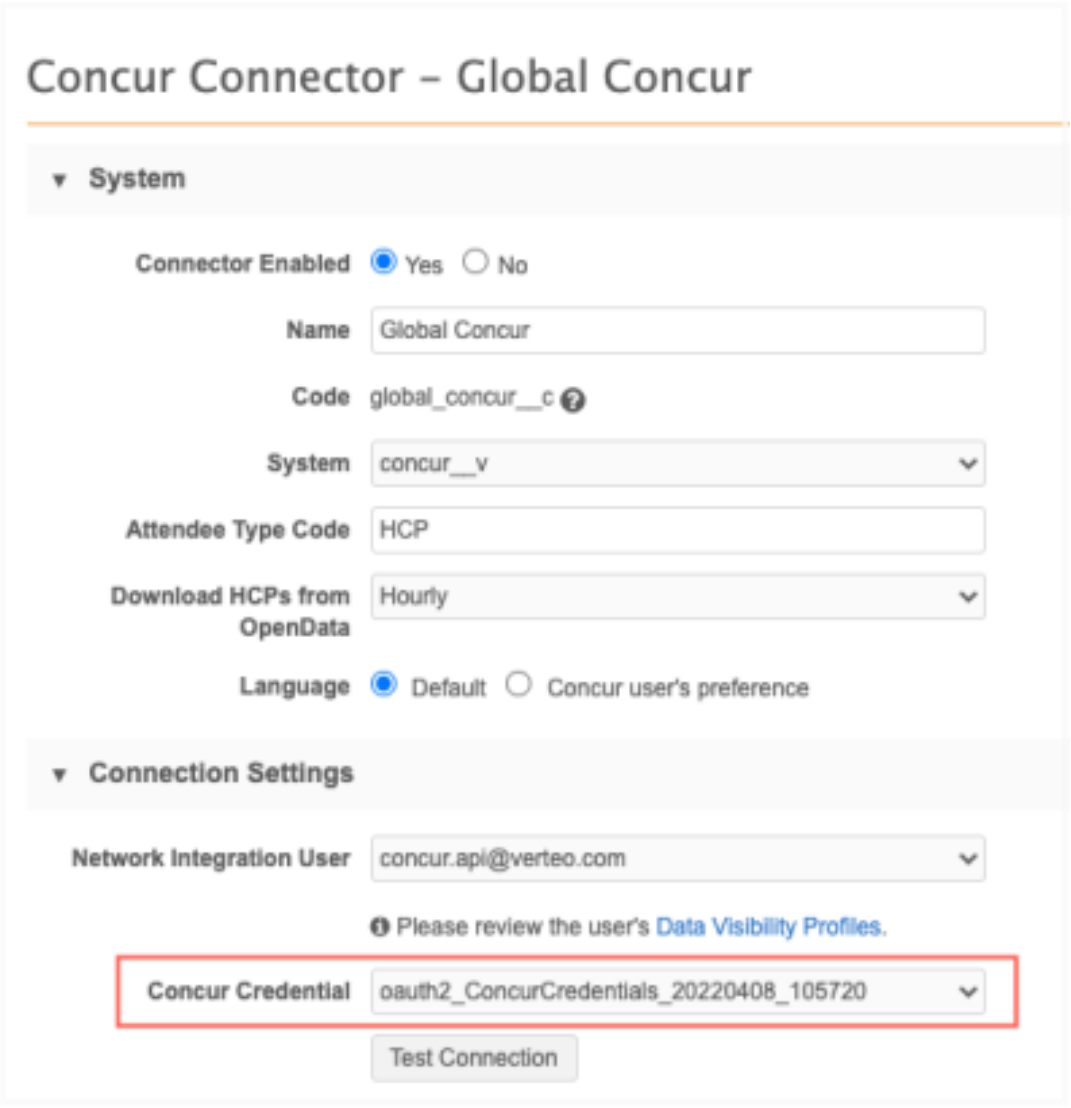
Existing customers will be automatically migrated to use OAuth2 authentication. New customers can create credentials that use OAuth2 to connect their Network instance to SAP Concur.

This enhancement is available by default in your Network instance.

Migrating existing Concur credentials

Network will automatically migrate existing customers to OAuth2 when the Concur Connector is used the first time after the release.

On the Concur Connector configuration, the **Concur Credential** displays the OAuth2 authentication credential.



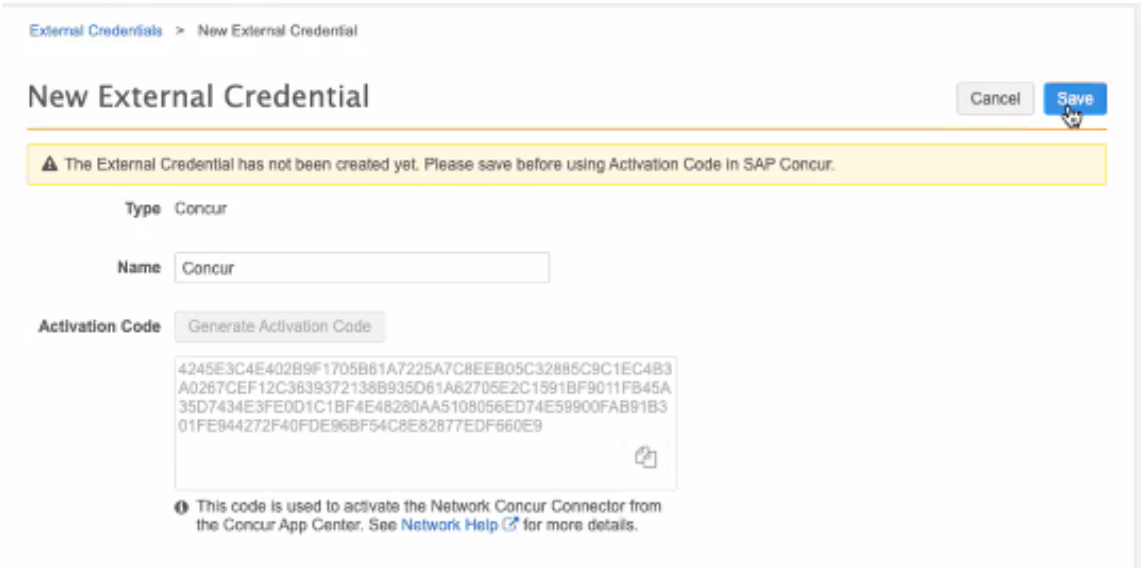
Creating Concur credentials

When you create a Concur credential an activation code is generated. Add the code to SAP Concur to connect it to your Network instance.

Note: The credentials are specific to each Network instance, so this must be done on all your Sandbox and Production instances.

To create credentials to connect to SAP Concur:

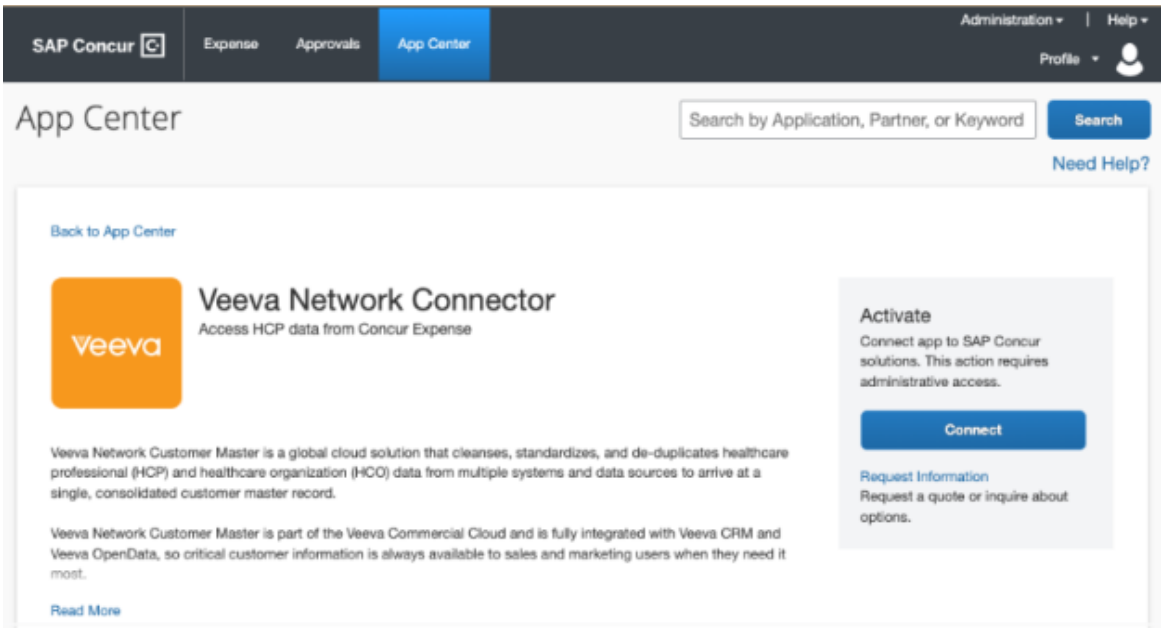
1. In the Admin Console, click **Settings > External Credentials**.
2. Click **Add Credentials** and select **Concur** from the list. Click **Continue**.
3. On the New External Credential page, type a **Name** for the Concur OAuth2 credentials.



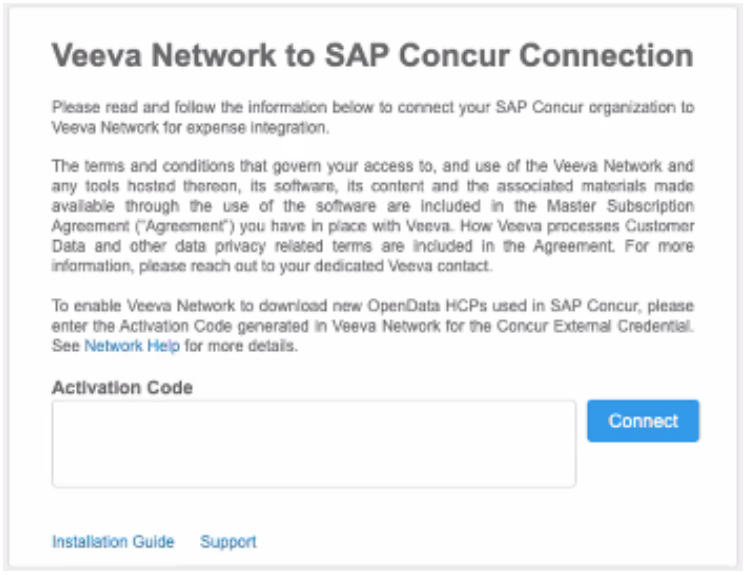
4. Click **Generate Activation Code**.

   The activation code displays in the box. Click the **Copy** icon to copy the code to your clipboard so you can paste it after you log into SAP Concur.

5. **Save** your changes.
6. Log into SAP Concur with your Web Services Admin user credentials.
7. In the **App Center**, search for *Veeva Network*.
8. Select the existing **Veeva Network Connector** app.



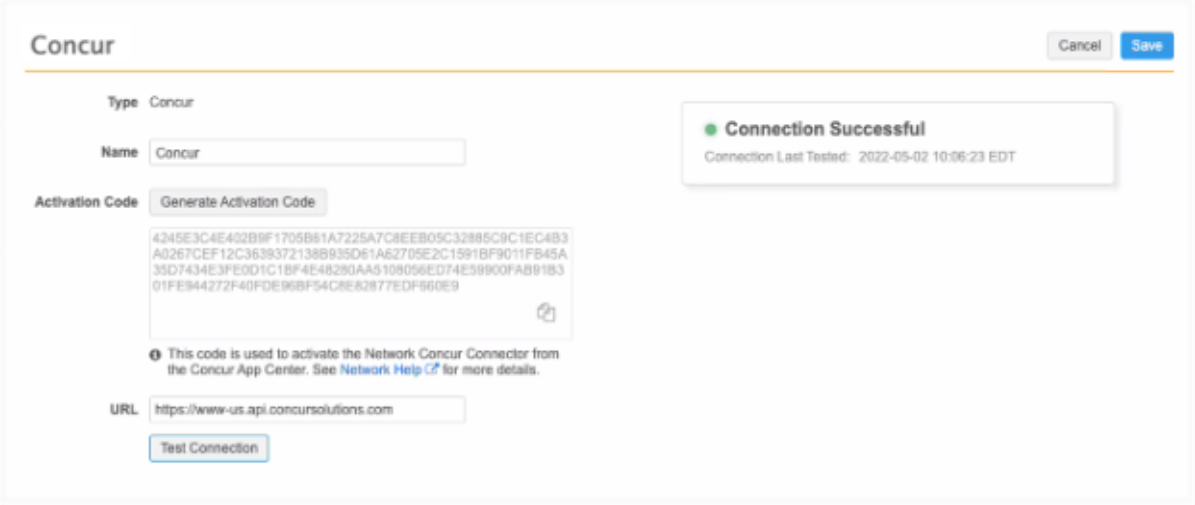
9. On the Veeva Network Connector app listing page, click **Connect**.
10. In the pop-up, accept the Terms and Conditions.
11. In the landing page that is hosted by Network, paste the **Activation Code** that you generated in Network. Click **Connect**.



12. A message displays when SAP Concur successfully connects to Network using the activation code. Click **Done**.
13. In Network, return to the External Credentials page for Concur. The **URL** field is now populated. The URL depends on the region that your SAP Concur is hosted.

Region	URL
USA	https://us.api.concursolutions.com
EMEA	https://emea.api.concursolutions.com
China	https://cn.api.concursolutions.com

14. Click **Test Connection**.



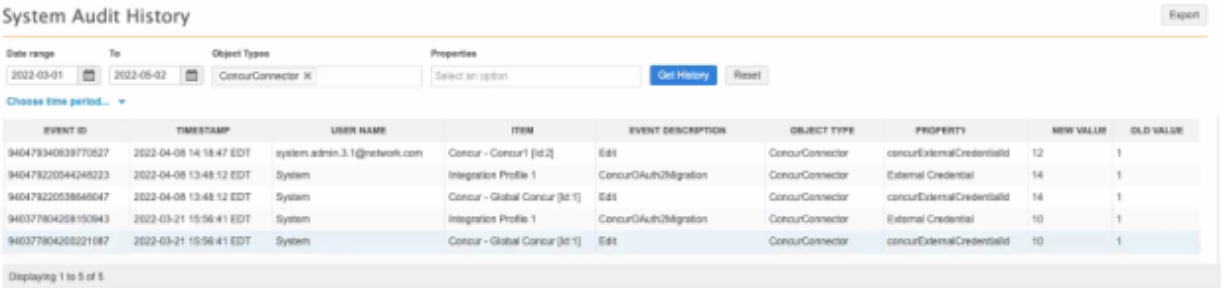
When the Connection Successful displays, your Network instance is authenticated to SAP Concur.

Repeat this process for all of your Network instances.

Audit

When the migration to OAth2 occurs, it is automatically logged in the System Audit Log.

To review the migration in the log, look for `ConcurOAuth2Migration` in the Event Description column. The **New Value** is `10` the **Old Value** is `1`.



API

VERSION UPDATE

22R2

The Network API is updated to v27.0.

The Network API version is updated for every major release. Any additional changes are documented in this section of the Release Notes.

As with all version updates, Integration Users should continue to use v26.0 until there is a change for v27.0 that they want to apply.

For more information about the Network API, see the *Veeva Network API Reference* at http://developer.veevanetwork.com.

API UPDATES FOR HASHTAGS

22R2

The following updates are supported for Network API v25 and later.

Search and Retrieve API

A new value is added to the following parameter to display hashtags that are specific to Veeva CRM.

Parameter

Parameter Name	Value
returnHashtagsForType	CRM

Response

Hashtags that meet the following conditions are returned:

- Record meets hashtag rule criteria
- Hashtag is active
- Hashtag is available in Veeva CRM.

Supplemental results

The `returnHashtagsForType` parameter supports hashtags for the `supplemental` parameter for the Search API.

The `supplemental` parameter returns related parent HCOs of the entities found in the supplemental results. If you use the supplemental parameter with the ALL or ONE value, the hashtags automatically display in the entity block.

Example response

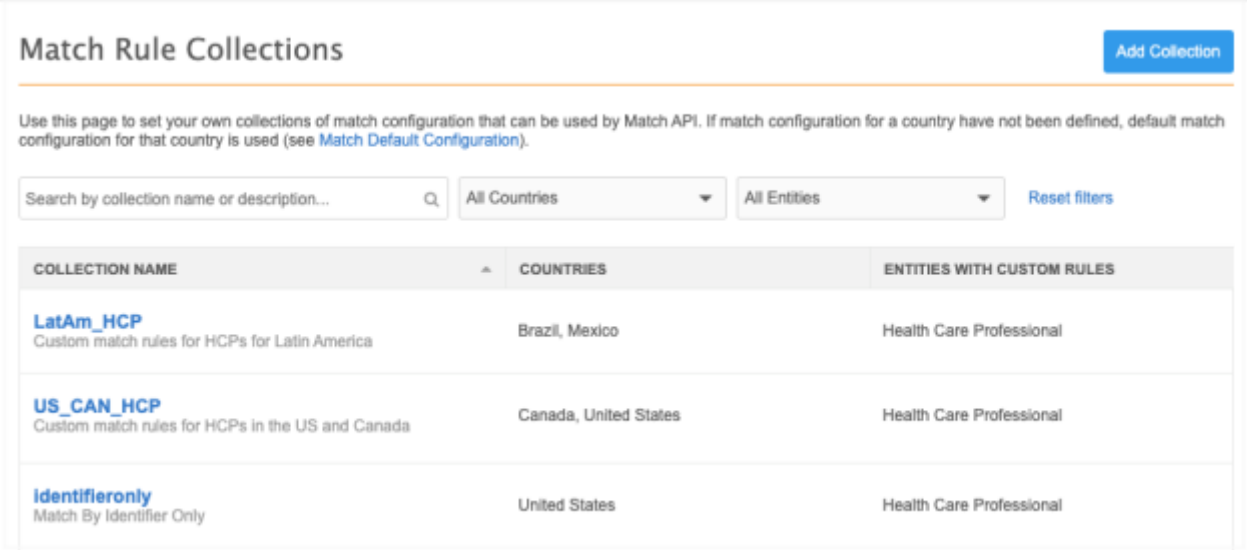

```
"supplementalResults": [
     {
       "specialty_1__v": "PD",
       "specialty_1__v_value__u": "Pediatrics",
       "hco_type__v": "4:4",
       "hco_type__v_value__u": "Organization, Group Practice",
       "340B_eligible__v": "N",
       "340B_eligible__v_value__u": "No/False",
       "master_vid__v": "242977361002234881",
       "record_owner_type__v": "VOD",
       "record_owner_type__v_value__u": "OpenData",
       "lab_services__v": "U",
       "lab_services__v_value__u": "Unknown/No response",
       "hco_status__v": "A",
       "hco_status__v_value__u": "Active",
       "accept_medicare__v": "U",
       "accept_medicare__v_value__u": "Unknown/No response",
       "specialty_3__v": "PM",
       "specialty_3__v_value__u": "Physical Medicine & Rehabilitation",
       "count_all_locn_medstaff__v": 0,
  .....
       ],
       "hashtags": [
         {
           "name": "#groupPractice",
           "color": "gray",
           "tooltip": "HCO is a group practice",
           "tooltipLabel": null,
           "tooltipValue": null,
           "tooltipHasOtherValue": false
         },
         {
           "name": "#pediatrics",
           "color": "gray",
           "tooltip": "Record specializes in pediatrics",
           "tooltipLabel": null,
           "tooltipValue": null,
           "tooltipHasOtherValue": false
         }
       ],
       "account_link_set__v": [],
       "licenses__v": [],
       "parent_hcos__v": []
     }
   ]
 }
```

MATCH RULE COLLECTIONS

22R1.1

Administrators can create match rule collections to use in the Match API. The Network Match API was introduced in version 21R3.0 to enable you to match data immediately for a single record. This is helpful when you need to verify data in real-time, for example, when you are registering HCPs in a portal. You can match the data using an API call instead of trying to verify the data using Network Search which requires multiple searches, filtering, and field queries.

By default, the Match API uses the country default match rules for your Network instance. Now you can create and use match rule collections that contain country groups that share custom match rules and data groups. Match rule collections are independent of subscriptions and other Network processes, so you can customize them for specific purposes.



This feature is enabled by default for your Network instance.

Match API

A new parameter enables Integration users to use custom match rules in the Match API to match data immediately for a single record.

Parameter

- **matchRuleCollection** - The name of the match rule collection to be used for matching.

  This is an optional parameter. If it is not specified, the country default match rules are used.

Response

The response includes the following:

- **matchSetup** - Information about the match configuration used.

Example response

```
{
   "status": "SUCCESS",
   "matchSetup": {
     "country": "US",
     "entity": "HCP",
     "dataGroups": [
         {
             "instance" : 20301
             "first_name__v=james & is_externally_mastered__v!=true &
primary_country__v=us": 1000,
             "first_name__v=james & is_externally_mastered__v!=true &
last_name__v=johnson & primary_country__v=us": 345,
             "addresses__v.locality__v=new york &
is_externally_mastered__v!=true & last_name__v=johnson &
primary_country__v=us": 235,
             "addresses__v.locality__v=new york &
is_externally_mastered__v!=true & primary_country__v=us": 1000,
             "addresses__v.locality__v=new york &
addresses__v.postal_code_primary__v=10027 & is_externally_mastered__v!=true
& primary_country__v=us": 1000,
             "is_externally_mastered__v!=true & npi_num__v=1962593913 &
primary_country__v=us": 1
         },
         {
             "instance" : 301
             "first_name__v=james & is_externally_mastered__v!=true &
primary_country__v=us": 1000,
             "first_name__v=james & is_externally_mastered__v!=true &
last_name__v=johnson & primary_country__v=us": 1000,
             "addresses__v.locality__v=new york &
is_externally_mastered__v!=true & last_name__v=johnson &
primary_country__v=us": 1000,
             "addresses__v.locality__v=new york &
is_externally_mastered__v!=true & primary_country__v=us": 1000,
             "addresses__v.locality__v=new york &
addresses__v.postal_code_primary__v=10027 & is_externally_mastered__v!=true
& primary_country__v=us": 1000,
             "is_externally_mastered__v!=true & npi_num__v=1962593913 &
primary_country__v=us": 1
         }
     ]
     "includeMasterResults": true,
     "addressCleansing": true,
     "matchRuleCollection": "verteoMatch"
   },
   ...
 }
```

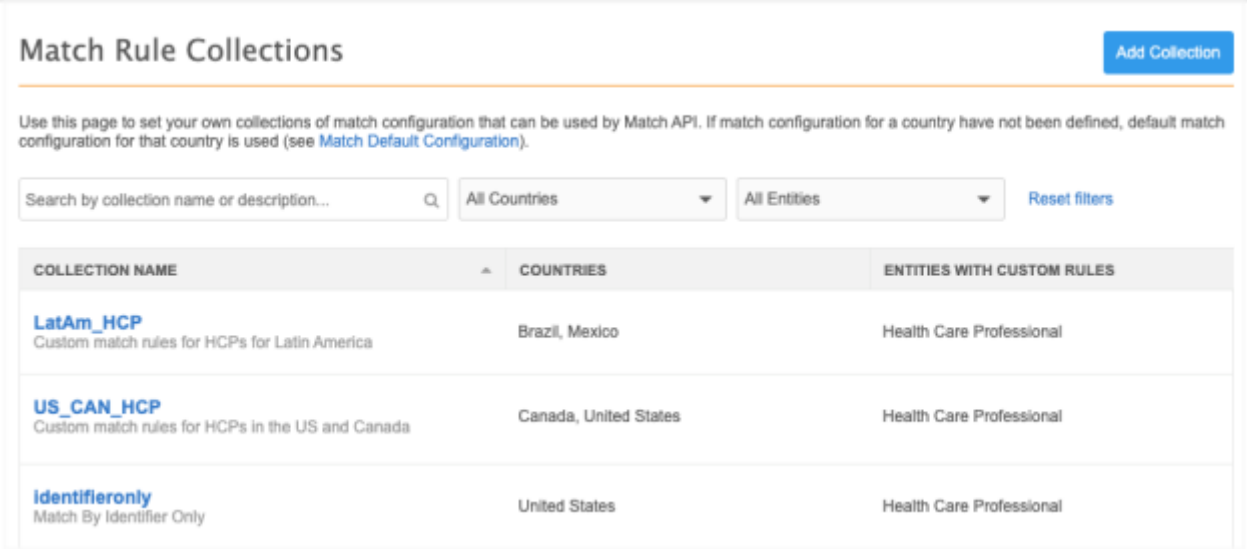
Errors

The `responseStatus` returns a FAILURE message if the Match API request is submitted with an invalid `matchRuleCollection`.

Creating match rule collections

Administrators can create custom match rules to use in the Match API.

In the Admin console, a new tab called **Match Rule Collections** displays in the **System Interfaces** menu.



This page displays the match rule collections that you have created in your Network instance.

Each row contains the following details:

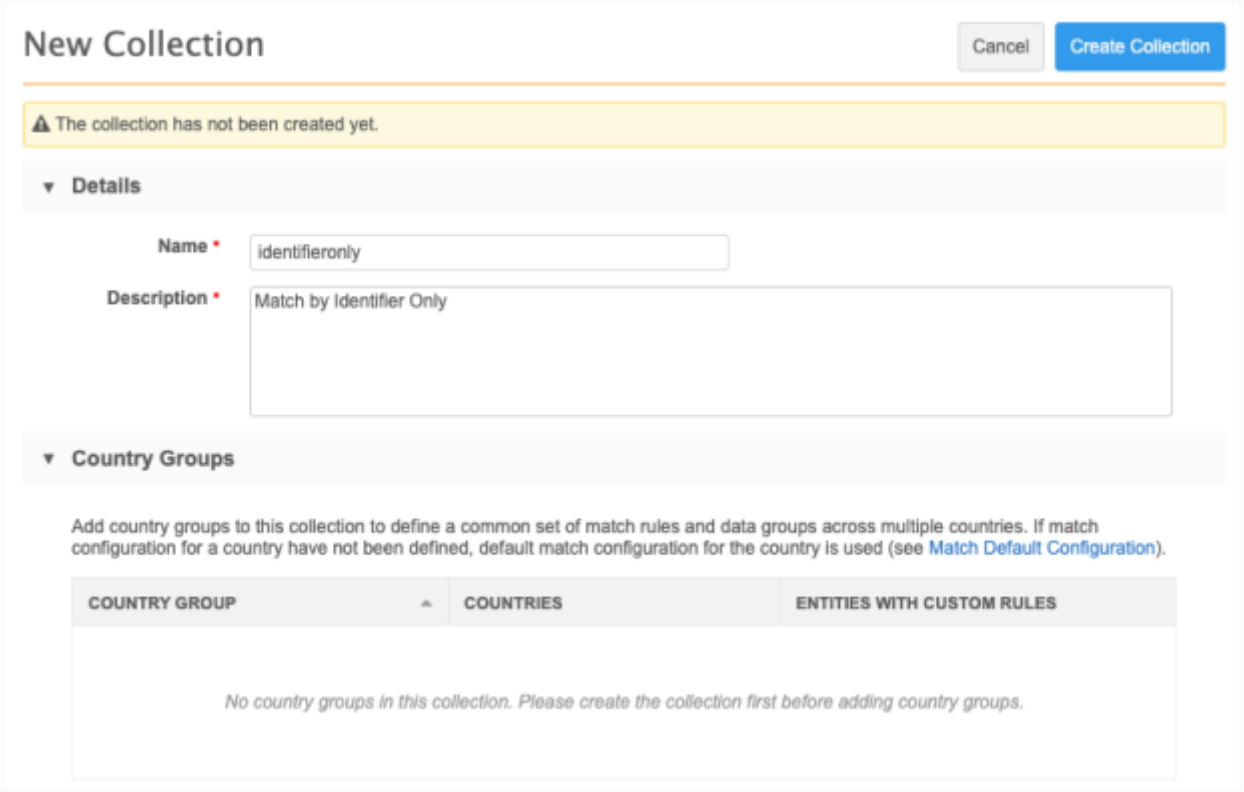
- **Collection Name** - The name and description.
- **Countries** - The countries in all of the country groups defined in the collection.
- **Entities with Custom Rules** - The objects that have custom rules defined. Objects in the collection that use the default match rules do not display in this column.

Each collection contains custom match rules and data groups for country groups.

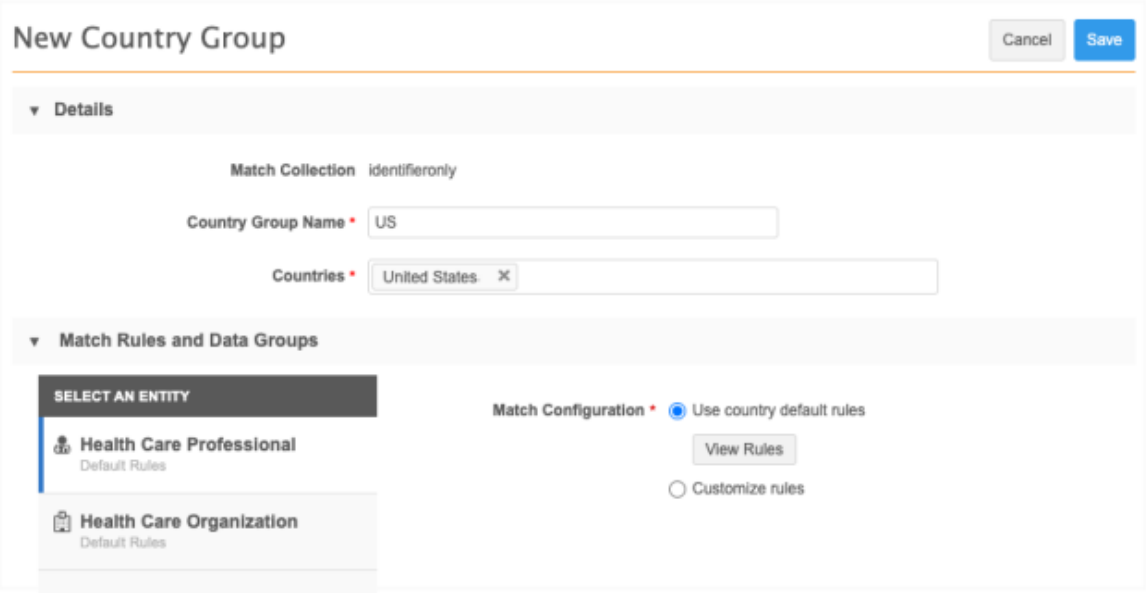
To create a match rule collection:

1. In the Admin console, click **System Interfaces > Match Rule Collections**.
2. On the Match Rule Collections page, click **Add Collection**.
3. In the **Details** section, define a **Name** and **Description**.

   The name must be unique. Letters, numbers, and underscore (_) characters are supported for the name.

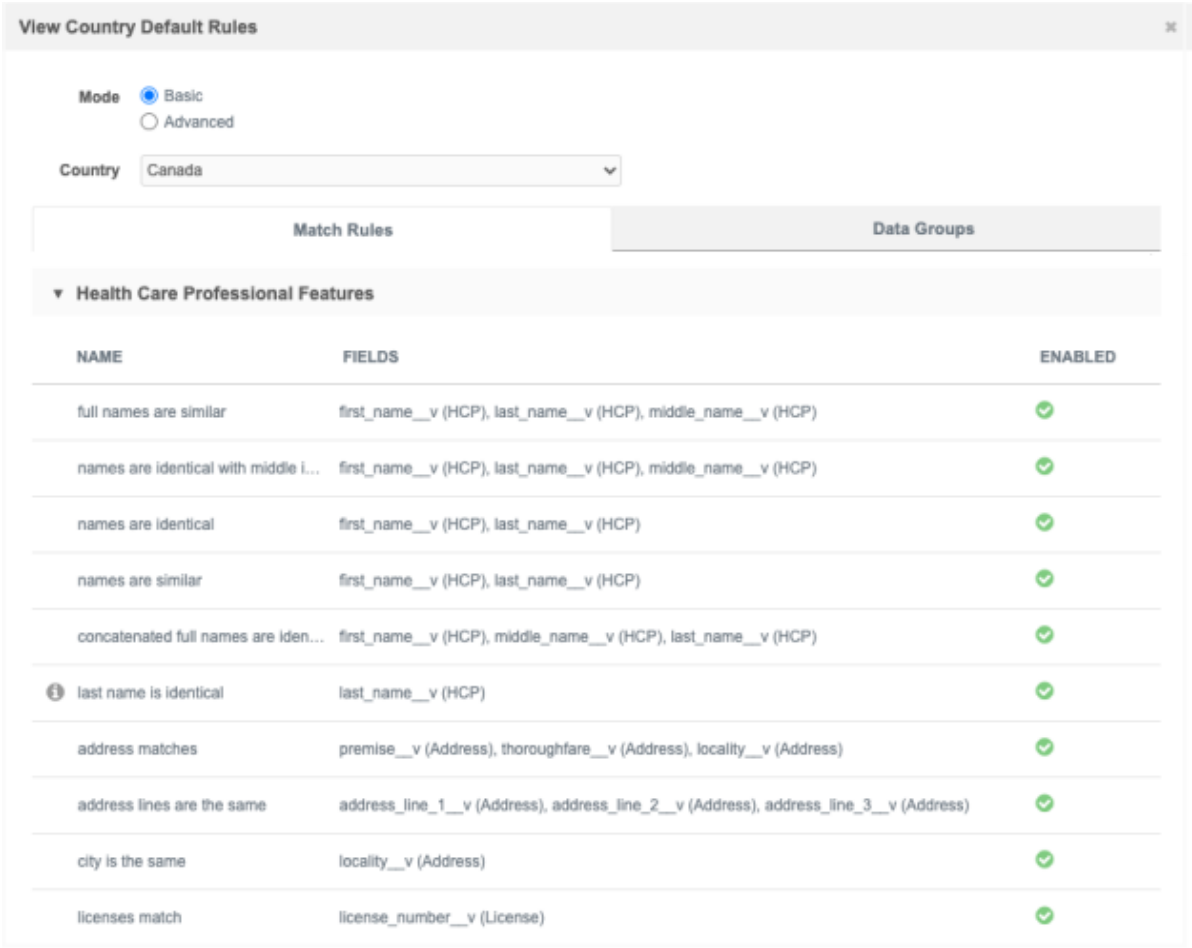


4. Click **Create Collection** to save it. The collection must be saved before you can add country groups.
5. Click **Add Country Group** . Match collection rules are specific to country groups. Country groups are countries that use the same match rules and data groups.
6. On the New Country Group page, define the following:
    - **Country Group Name** - The name of the group.
    - **Countries** - Add the countries that apply to this group. A country can belong in one country group for each match rule collection.



7. In the **Match Rules and Data Groups** section, select the entity that will use custom match rules and data groups. For example, click **Health Care Professional**.
8. Beside **Match Configuration**, the option **Use country default rules** is defined by default. Click **View Rules** to see a read-only preview of the current default rules.

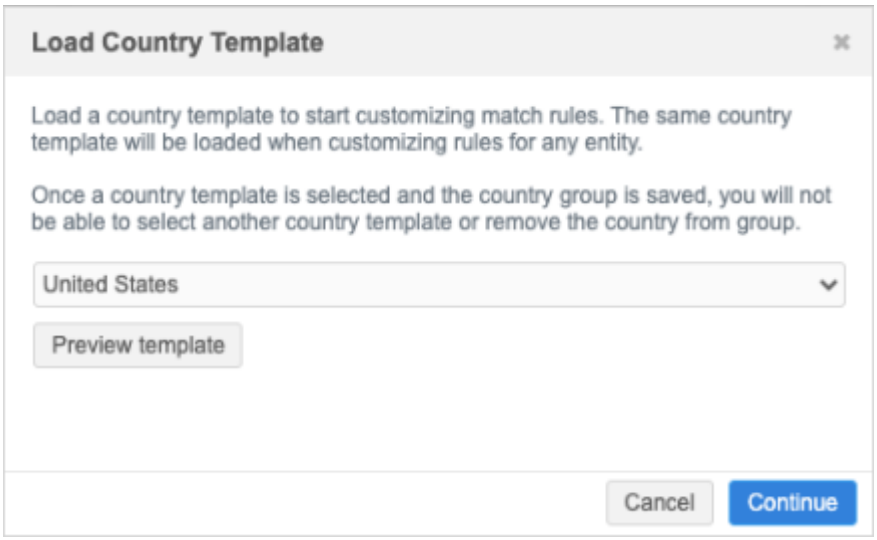
   If the country group contains multiple countries, use the **Country** list to see each country's default rules.



These are the rules that are currently the default for your Network instance for the country (the rules in the Match Default Configuration). They are not necessarily the default rules that Network provided when you received your Network instance. If you have made any changes to those rules, those changes are considered the default rules for your Network instance now.

9. Choose **Customize rules** to start defining the custom match rules.
10. On the **Load Country Template** pop-up, select a country from the list. The default rules for the selected country will be used as a starting point for the custom match rules. The countries that are defined for the country group are available in the list.

    The template (country default match rules) that you choose will be applied to each object in the country group if you customize rules for other objects.

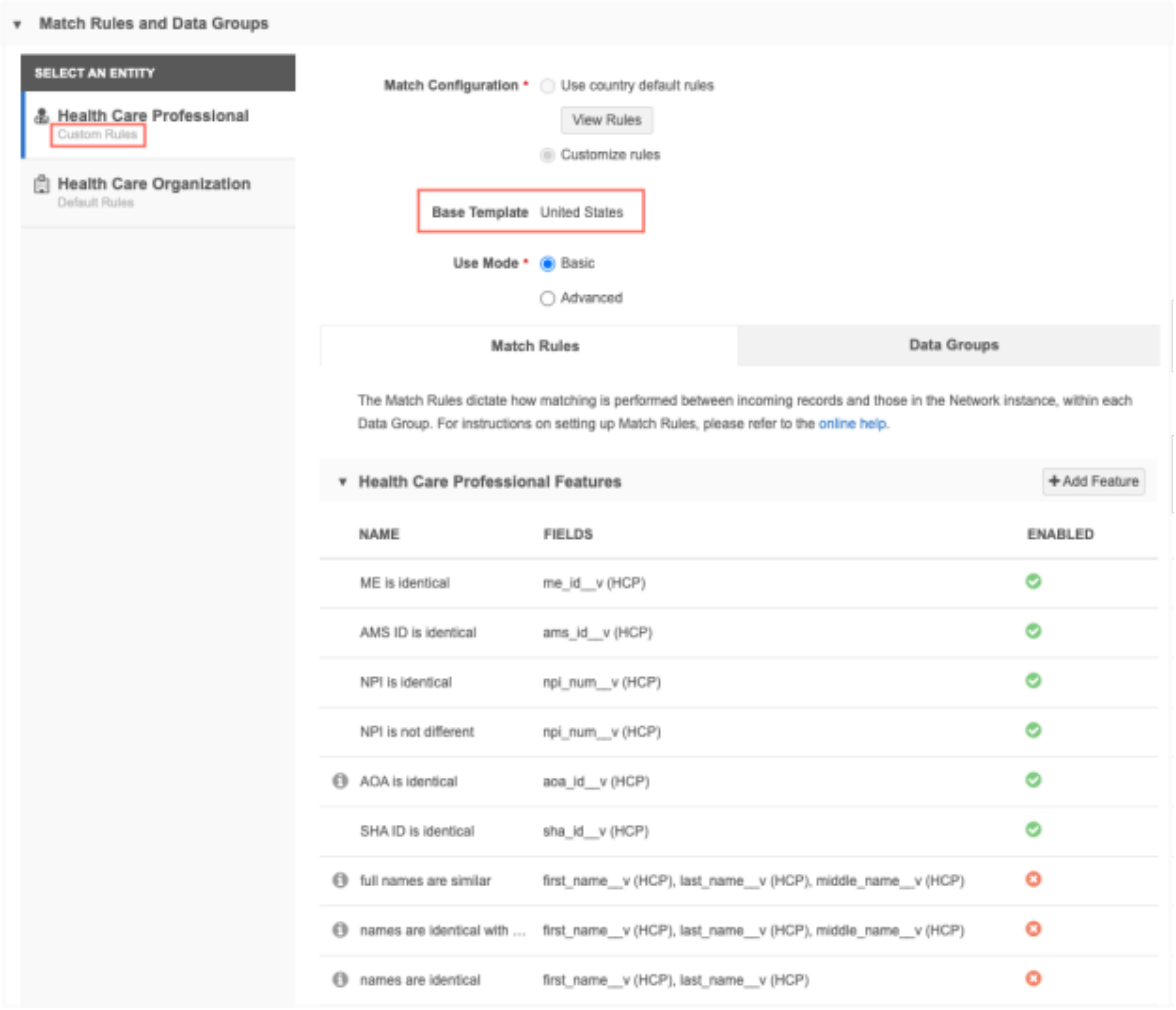


Note: After you save the country group, the template cannot be changed and the country cannot be removed from the country group.

Click **Preview** to open and review a read-only version of the default match rules and data groups for that country. Close the pop-up and click **Continue**.

11. The **Match Rules** and **Data Groups** tabs display. Customize the match rules and data groups for the selected object. For example, if you want the match collection to match by IDs only, you might disable all of the match rules that are not for IDs.

    By default, **Basic** mode is selected for HCPs and HCOs. You can edit the rules in **Basic** or **Advanced** mode. Only **Advanced** mode is supported for custom objects.



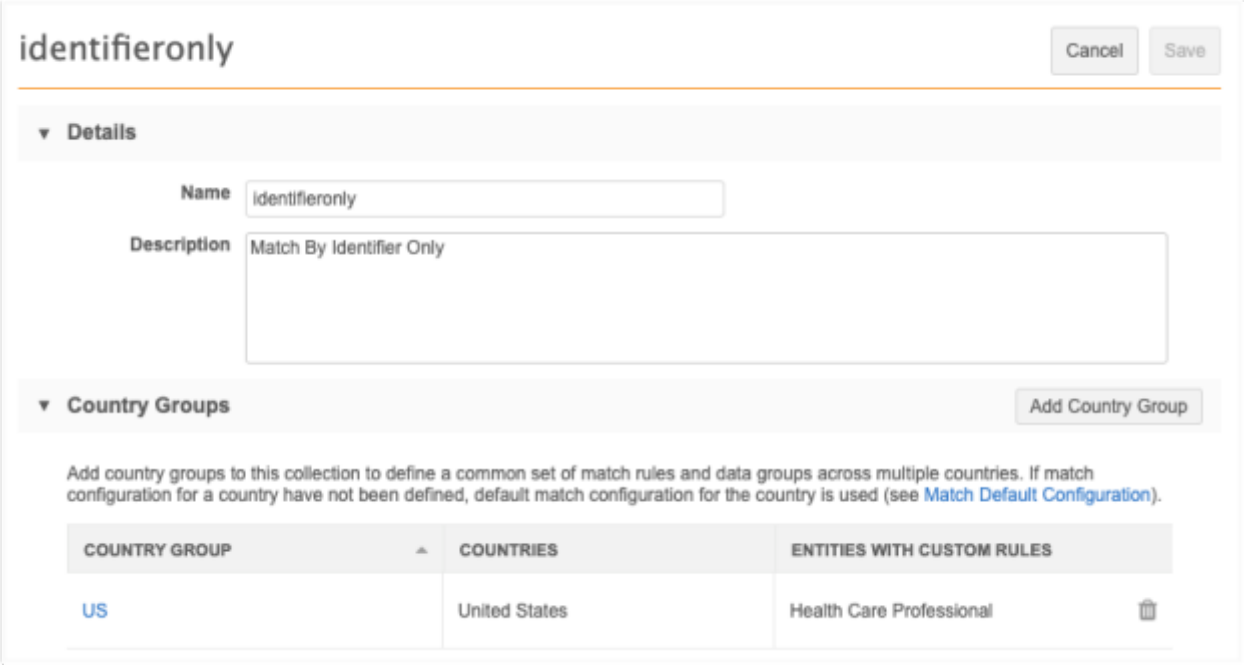
Notice that the **Custom Rules** label displays below the selected object. Also, the country group is updated to display country that the **Base Template** uses.

12. Save the country group.

Note: If you have customized rules for an object, after the match rule collection is saved, the entities cannot be set back to use Default Rules.

Considerations

- To save the country group, at least one object must have custom rules defined. If all of the objects in the match rule collection are set to use default rules, a message displays: `No changes in match rules detected. Please customize an entity's match rules before saving a country group.`
- If the country group contains multiple countries and you customize a rule for an object but there isn't a rule for that object in one of the countries, a message displays: `At least one field is not available or not enabled in this country`. For example, if you use the default match rules for Mexico as your template and there isn't a rule for that object for Brazil, the message displays. The rule must be removed before the country group can be saved.



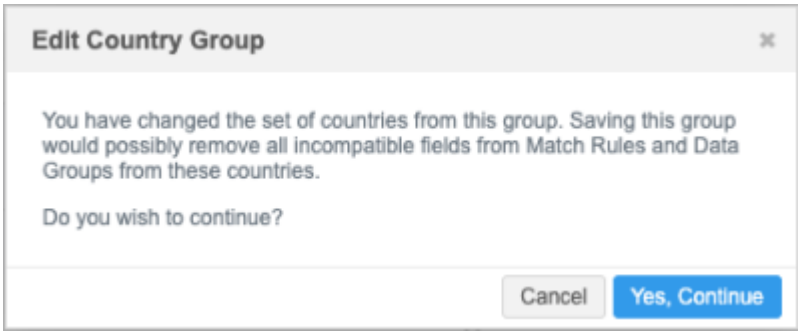
13. *Optional* Add more country groups to the match rule collection.
14. Save the match rule collection.

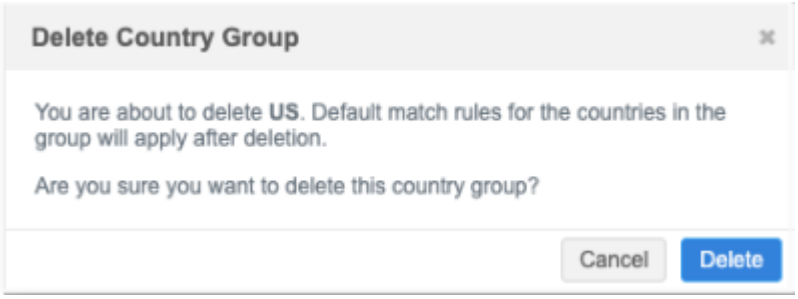
Edit country groups

Country groups can be edited or deleted.

- Add or remove countries to the country group.

  When you save the country group, a pop-up displays. Click Change to confirm that you want to update the country group. Network validates the features, feature sets, and data groups to update the fields used in the rules.



Delete country groups

You can delete a country group from the match group selection. If you delete a country group, the default match rules for the countries in the group will apply. A message displays to confirm that you want to delete the group.