Veeva Network

Veeva Network 22R1.1.1 Release Notes

June 2022

Contents

About these Release Notes

These Release Notes describe all features that are included in Veeva Network 22R1.1.

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

No features or enhancements have been added since the Early version of this document was published.

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 22R1.1 minor release.

		ST	DS	DM	AD
Network widgets					
Hierarchy Explorer	Several enhancements have been added to Hierarchy Explorer to improve your user experience.	●	●	●	●
Data Lineage					
Current source column	A new column identifies the current source for fields on the record.	●	●	●	●
Sub-object details	Addresses, licenses, and parent HCOs now contain a summary so you can easily identify the correct sub-object.	●	●	●	●
Reports					
Flattened hierarchies	A new table called flat_hierarchy is added to Network reports to help you report on relationship hierarchies.		●	●	●
File Explorer					
Smart tables	Open, view, and augment .csv files directly from the File Explorer.		●	●	●
Data quality					
Data cleansing	Administrators and data managers can now define rules to cleanse and standardize data in fields.			●	●
Data Model					
New countries supported	Veeva OpenData data models have been added for countries in Latin America and Asia Pacific.			●	●
Data privacy opt out	Veeva OpenData now manages HCP opt outs for several additional countries in the Asia Pacific region.			●	●
Cluster management	This feature now supports cluster data for Australia, Czech Republic, Portugal, and Slovakia.			●	●
Primary addresses	Define custom recalculation logic based on field conditions for Unique Checkbox primary address configurations.			●	●
Field configurations	The Default Value and NEX Rules fields are bigger so it is easier to view and manage those values.			●	●

		ST	DS	DM	AD
Data sources					
DCR routing for third party systems	Third party systems can support data change requests (DCRs) for customer managed fields on unverified records.			●	●
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.			●	●
Veeva OpenData subscriptions					
Export job error logs	Administrators can export the job error log.			●	●
Source subscriptions					
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.			●	●
Integrations					
Concur Connector	The SAP Concur Connector is updated to use OAuth2 authentication.				●
API					
Match rule collections	Administrators can create custom match rules to use in the Network API.		Developers		

Note: The System and Data Admin user has all 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.

Network widgets

HIERARCHY EXPLORER

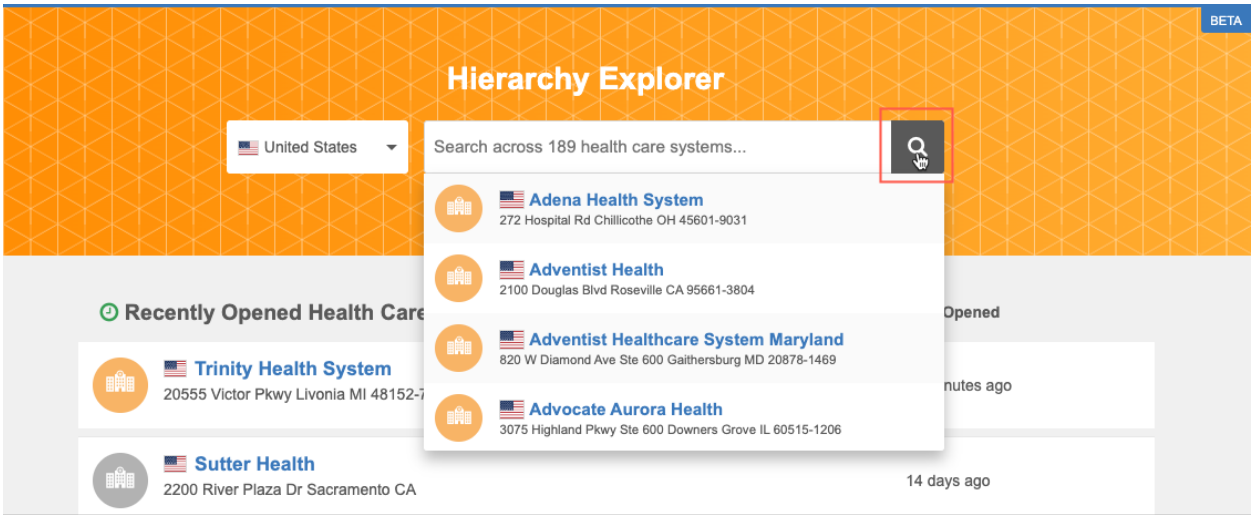
Several enhancements have been added to Hierarchy Explorer to improve your user experience.

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.

Health system search

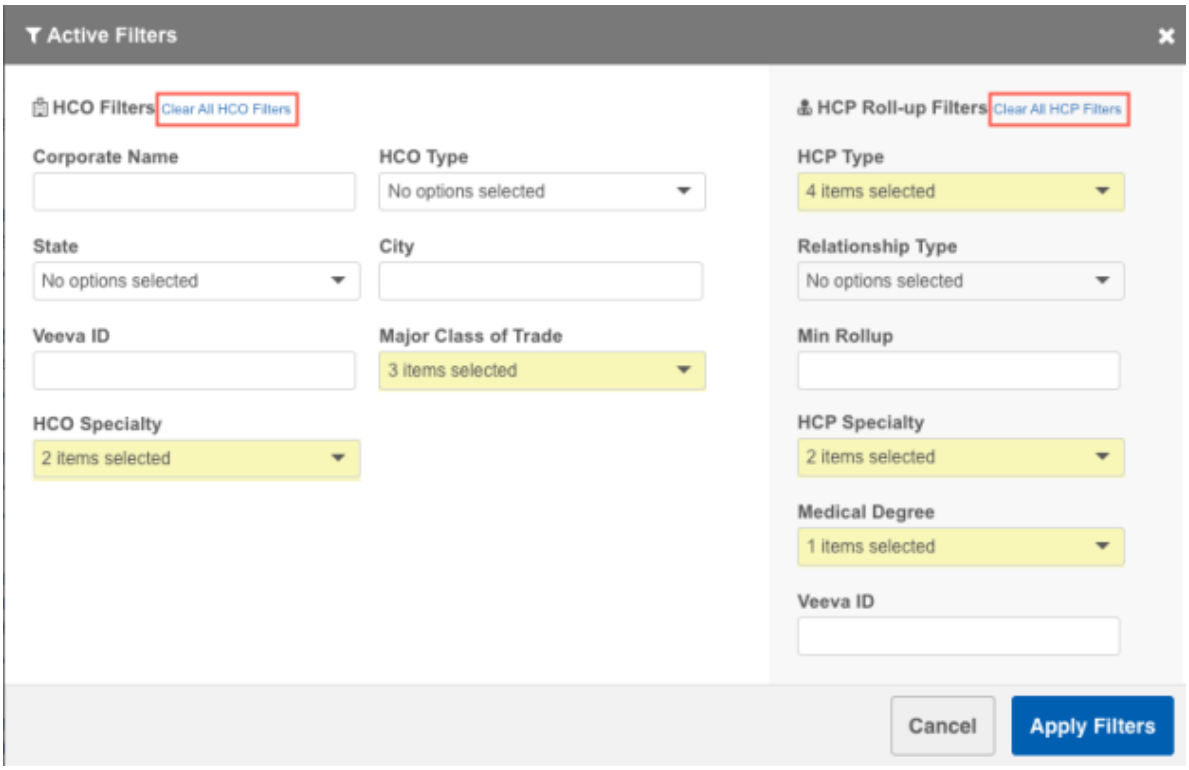
On the Hierarchy Explorer landing page, you can search for the health system that you want to explore. Click the **Search** icon to open the list of health systems that are available in your Network instance. The health systems are listed in alphabetical order.



Clear filters

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

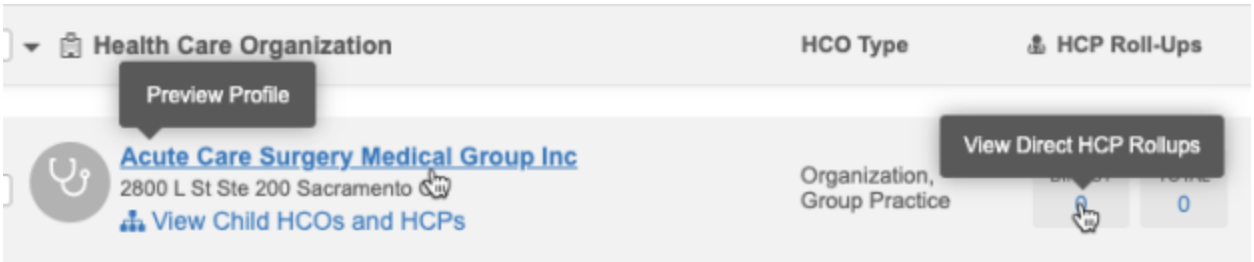
When you download a hierarchy or selected records, the **Hierarchy Level** column now displays in the exported file.

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 | 9379329432677269	3

Tooltips

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



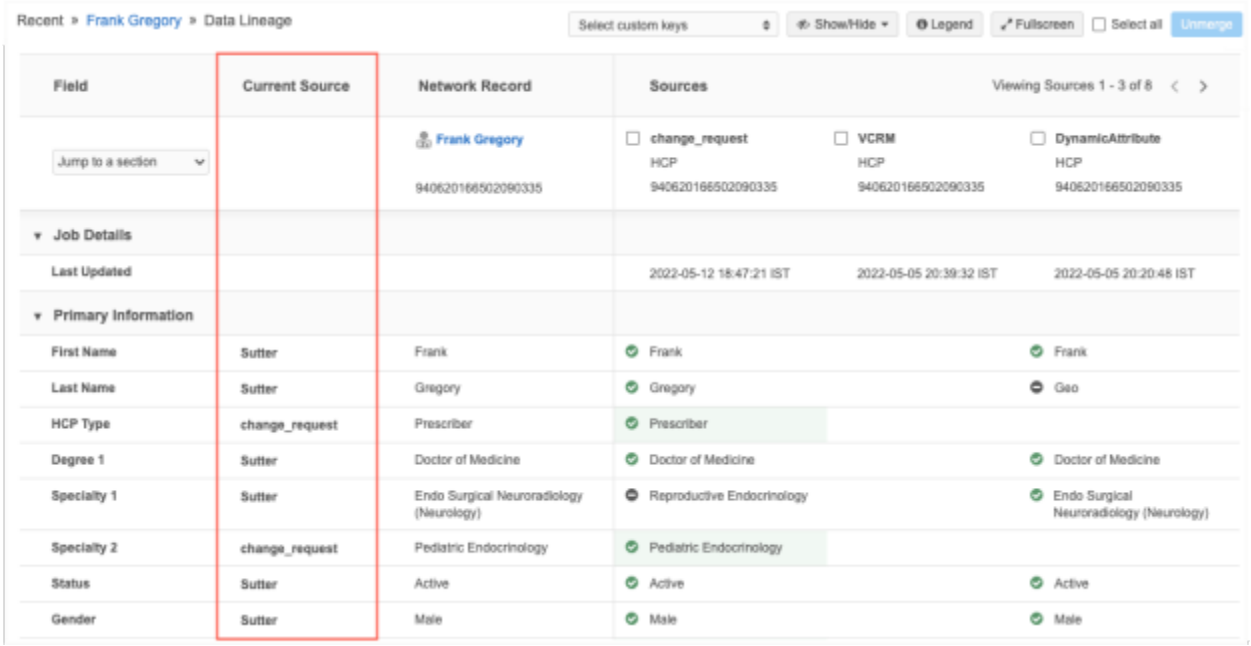
Data lineage

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

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

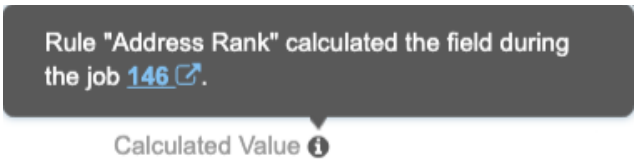


Other source values

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

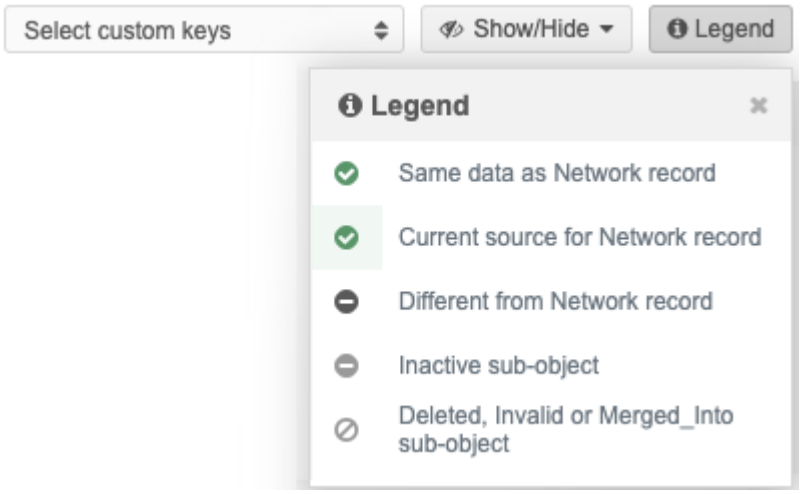
- **Updated by System** - System fields that are updated by Network. This value also displays for data that was calculated by Network using rules before these enhancements were released; that data is not backfilled for older values.
- **Calculated Value** - Displays on fields where the value is calculated 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

Several new icons have been added to fields to help you identify information about objects. Click the **Legend** button to see the explanation for 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.

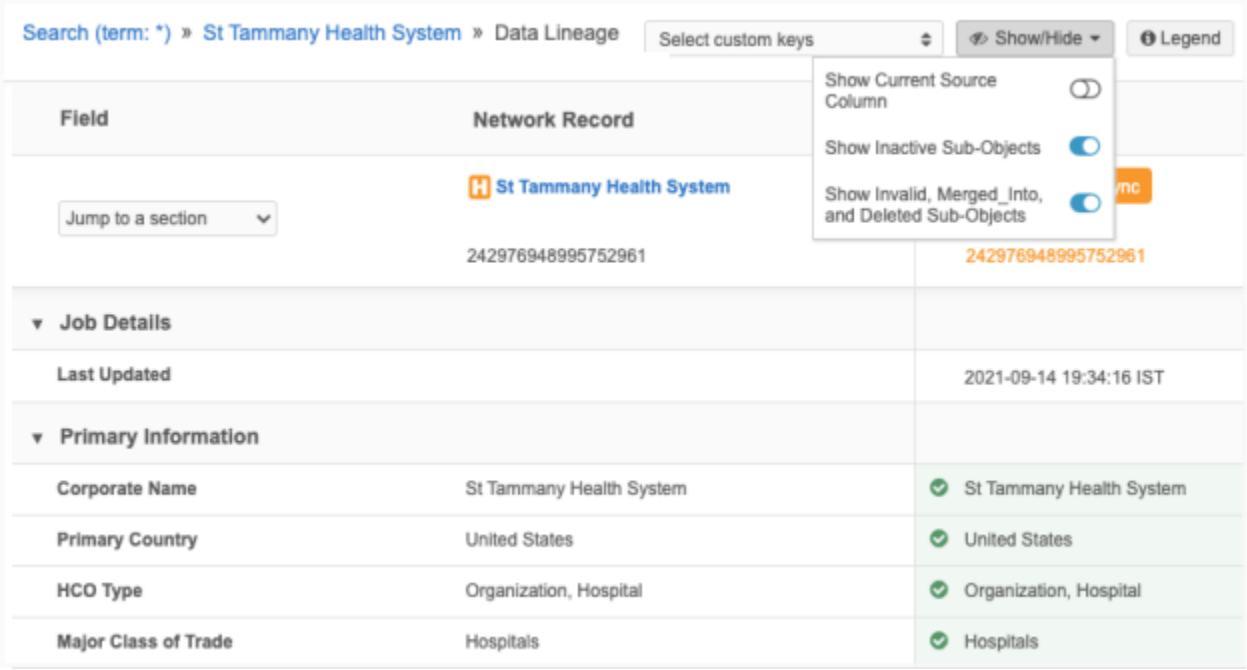
Inactive sub-object - The status of the sub-object is inactive.

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

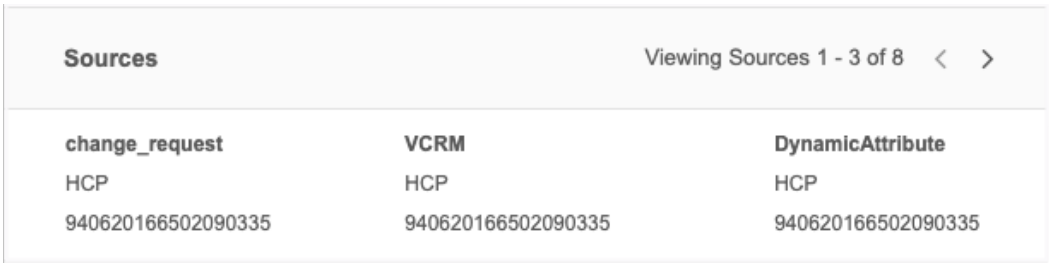
Click the **Show/Hide** button to toggle column and object options. By default, all options are enabled. Your preferences are retained the next time you open the Data Lineage page.



- **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. This option does not display if the **Current Source** column is not enabled for your Network instance.
- **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

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.

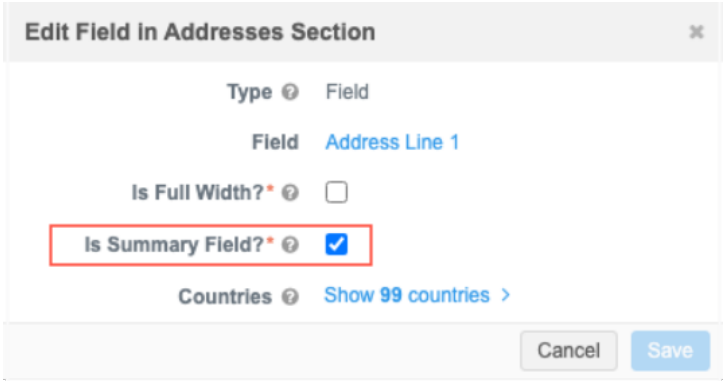
These enhancements are enabled by default in your Network instance.

Addresses

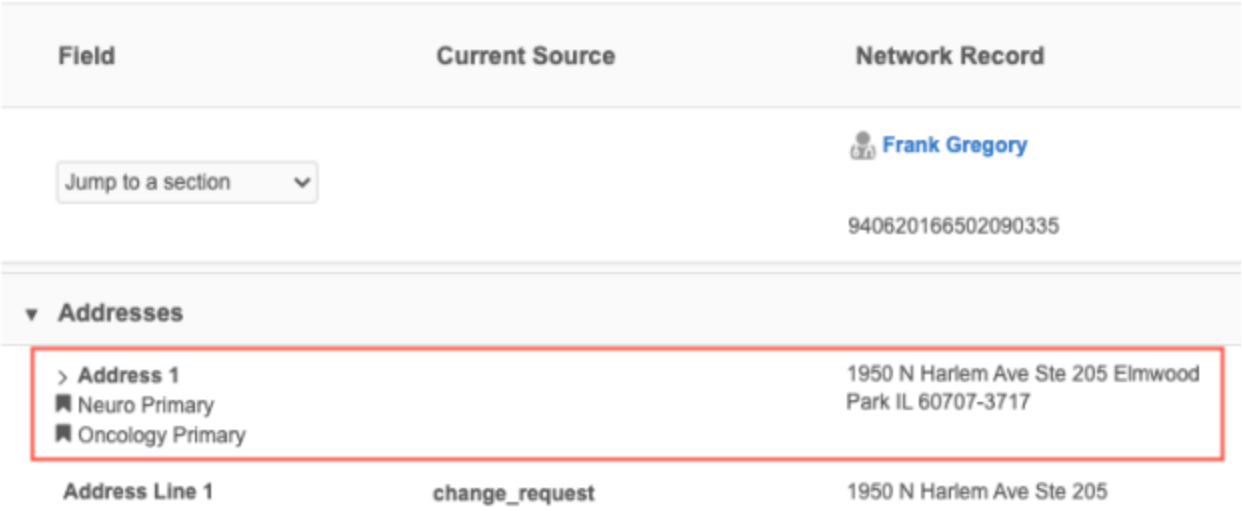
Address summaries contain the following details:

- 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

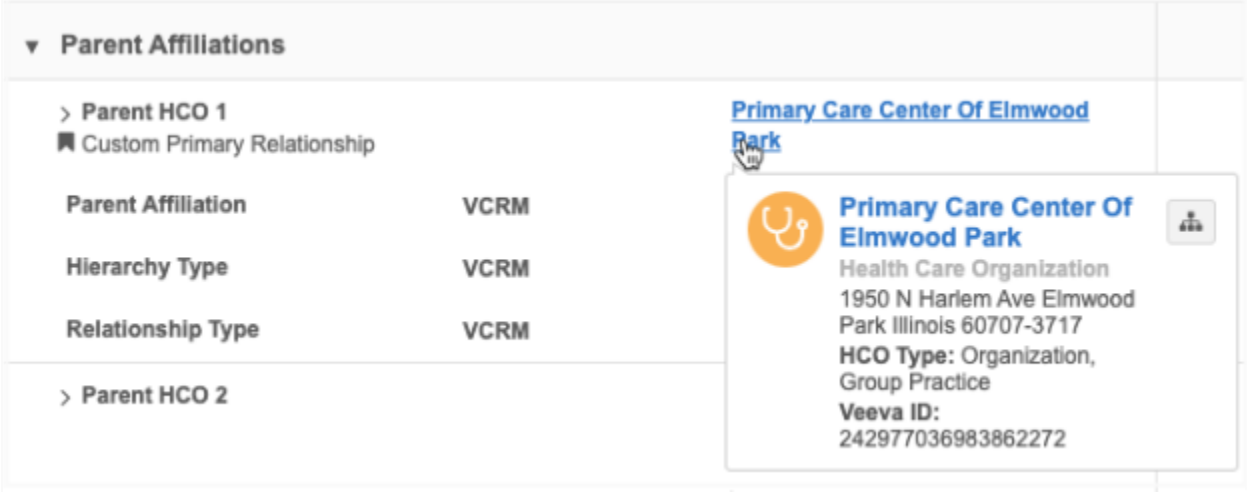
Addresses are sorted in the following order:

- 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

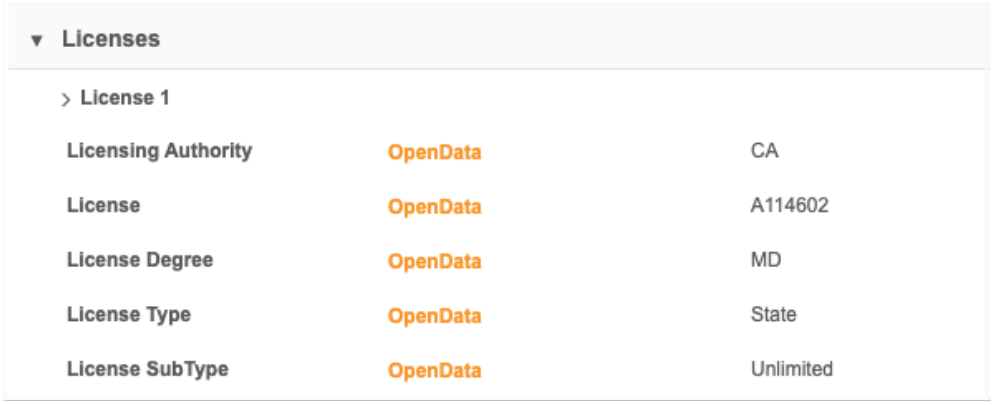
Parent HCOs are sorted in the following order:

- 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)

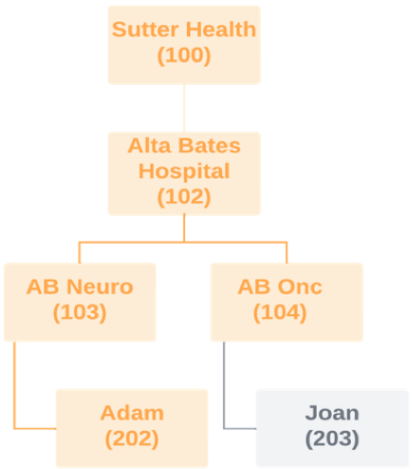
Reports

Reporting on hierarchies

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.



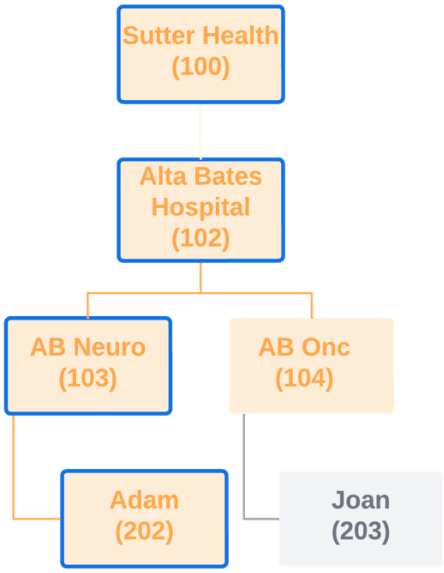
Entity VID	Entity Type	Parent HCO
Joan (203)	HCP	104 (AB Onc)
Adam (202)	HCP	103 (AB Neuro)
AB Onc (104)	HCO	102 (AB Hosp)
AB Neuro (103)	HCO	102 (AB Hosp)
AB Hospital (102)	HCO	100 (Sutter)

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 set an update schedule to populate the data.

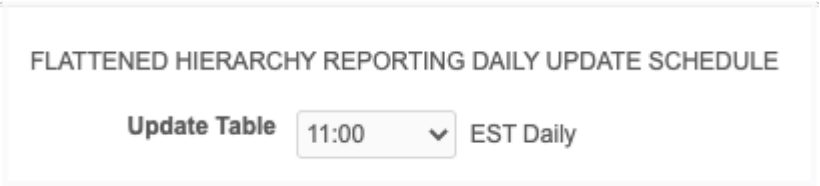
Note: You do not need to have the Hierarchy Explorer widget enabled to use this feature.

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.

Scheduling hierarchy updates

The `flat_hierarchy` reporting table is updated daily. Administrators can customize the time the table update occurs in their Network instance.



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.

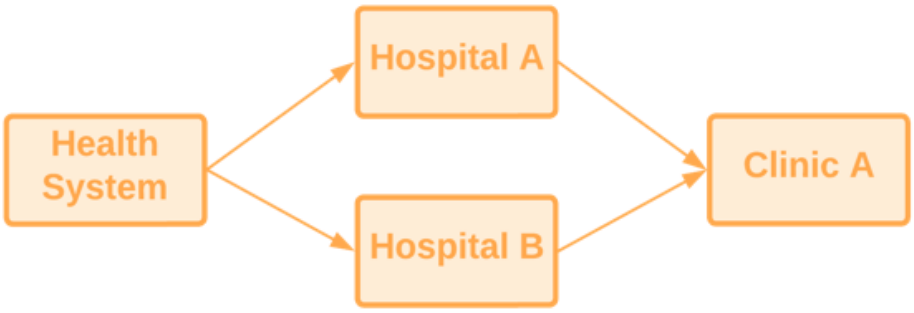
- **Inactive paths** - Paths are considered inactive for the following reasons:
  - HCP or HCP - Record status is not Active, they are candidate records, or HCPs have been opted out.
  - Relationships - The relationship status is not Active.
- **Invalid paths** - Paths are marked as invalid for the following 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:

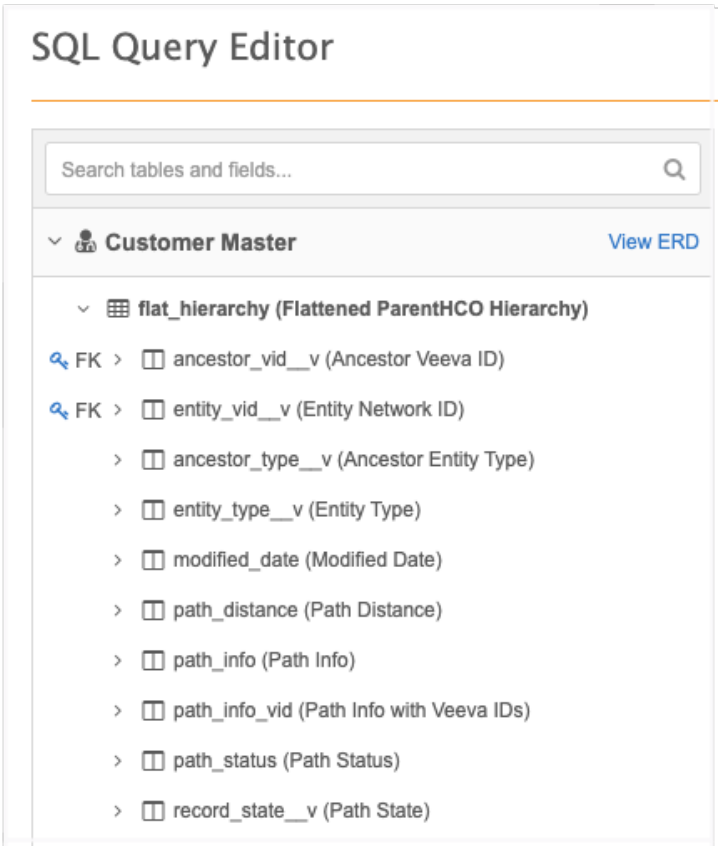
- **Health System → Hospital A → Clinic A**
- **Health System → Clinic A**)

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.

Field Name	Field Label	Field Type	Description
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.

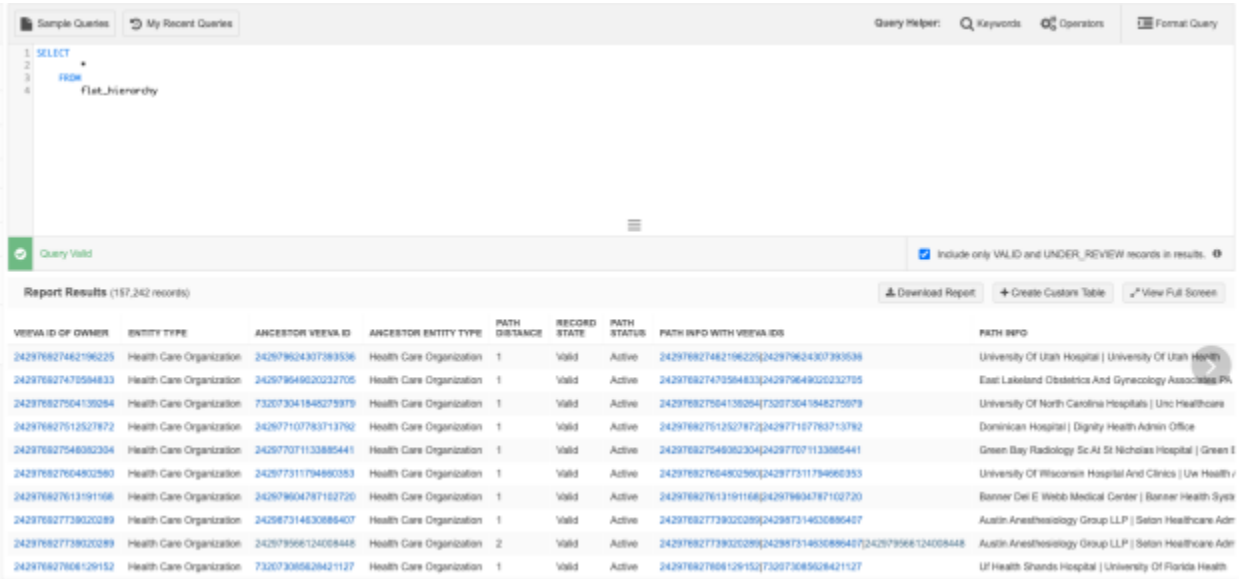
Example 1 - flat_hierarchy table structure

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

Report Results (2,407 records) | Download Report | + Create Custom Table | View Full Screen

VEEVA ID OF OWNER	ENTITY TYPE	ANCESTOR VEEVA ID	ANCESTOR ENTITY TYPE	PATH DISTANCE	RECORD STATE	PATH STATUS	PATH INFO WITH VEEVA IDS
242976927739020289	Health Care Organization	242979566124008448	Health Care Organization	2	Valid	Active	242976927739020289|242987314630886407|242979566124008448
242976931748774912	Health Care Organization	242979566124008448	Health Care Organization	1	Valid	Active	242976931748774912|242979566124008448
242976932361143296	Health Care Organization	242979566124008448	Health Care Organization	1	Valid	Active	242976932361143296|242979566124008448
242976933317444609	Health Care Organization	242979566124008448	Health Care Organization	3	Valid	Active	242976933317444609|242976938442884096|242977749554168832|2429
242976934550569984	Health Care Organization	242979566124008448	Health Care Organization	2	Valid	Active	242976934550569984|930192812526012348|242979566124008448

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 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

Report Results (39 records) | Download Report | + Create Custom Table | View Full Screen

Table | Chart

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.

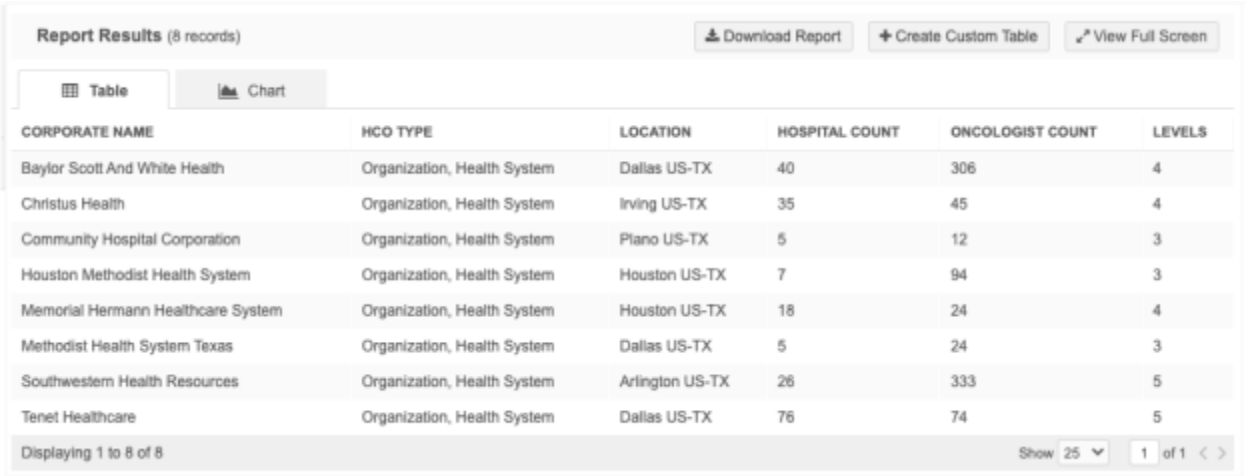
Query

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



Report Results (8 records) | Download Report | + Create Custom Table | View Full Screen

Table | Chart

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

Displaying 1 to 8 of 8 | Show 25 | 1 of 1

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.

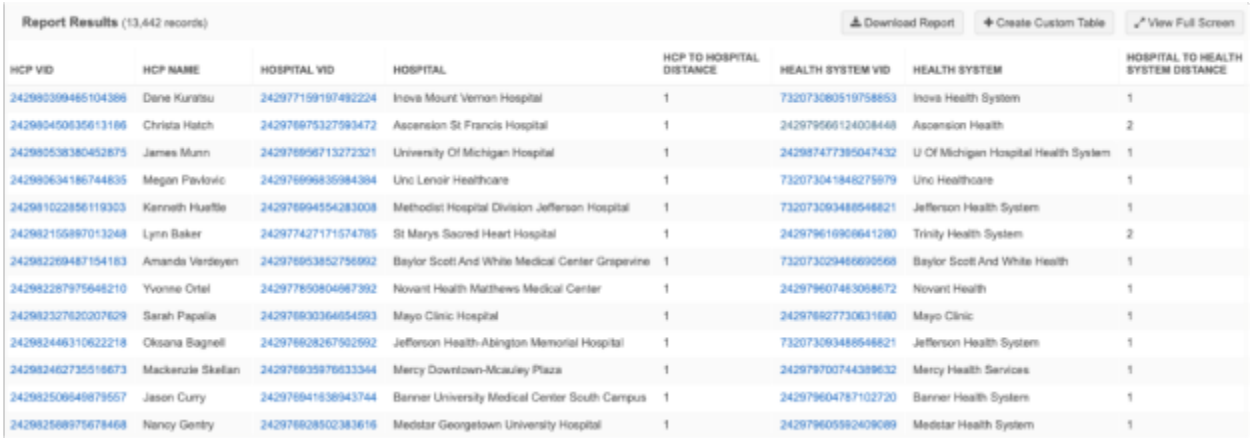
Query

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	242976996835984384	Unc Lenoir Healthcare	1	732073041848275879	Unc Healthcare	1
242981022856119303	Kenneth Hueftle	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
242982287975646210	Yvonne Ortel	242977650804967392	Novant Health Matthews Medical Center	1	242979607483068672	Novant Health	1
242982327620207629	Sarah Papalia	242976930364654593	Mayo Clinic Hospital	1	242976827730831680	Mayo Clinic	1
242982446310622218	Oksana Bagnell	242976928267502592	Jefferson Health-Abington Memorial Hospital	1	732073093488546821	Jefferson Health System	1
242982462735518673	Mackenzie Skellan	242976935976633344	Mercy Downtown-Mcauley Plaza	1	242979700744389632	Mercy Health Services	1
242982506649879557	Jason Curry	242976941638943744	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')
```

File explorer

SMART TABLES

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.

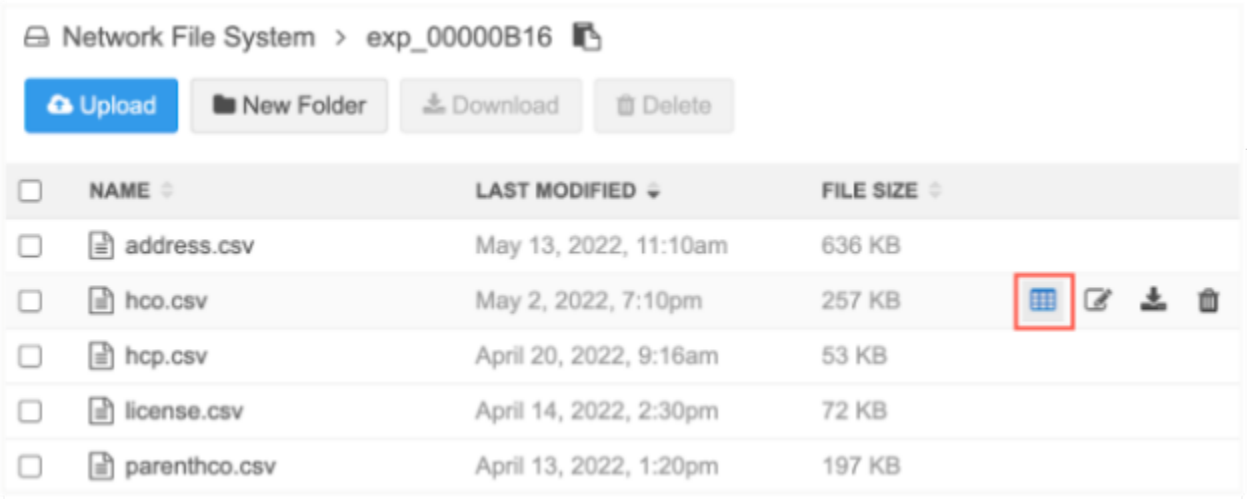
Note: Users must have FTP access to use the File Explorer.

View smart tables

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

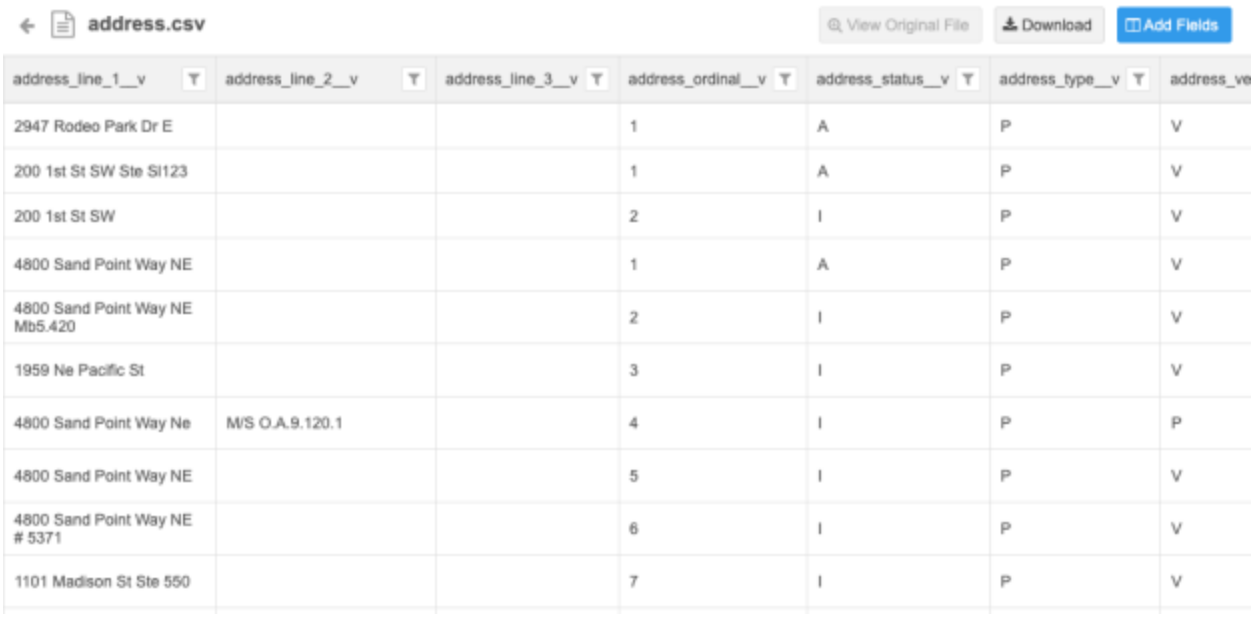
- Beside any .csv file, click the **Table** icon. The icon displays only on .csv file.

  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.



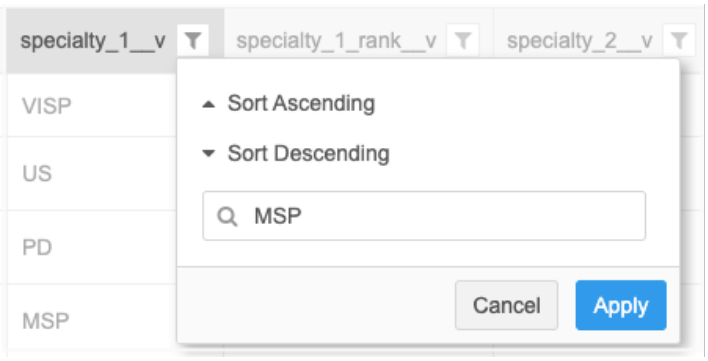
address_line_1__v	address_line_2__v	address_line_3__v	address_ordinal__v	address_status__v	address_type__v	address_ve
2947 Rodeo Park Dr E			1	A	P	V
200 1st St SW Ste SI123			1	A	P	V
200 1st St SW			2	I	P	V
4800 Sand Point Way NE			1	A	P	V
4800 Sand Point Way NE Mb5.420			2	I	P	V
1959 Ne Pacific St			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.

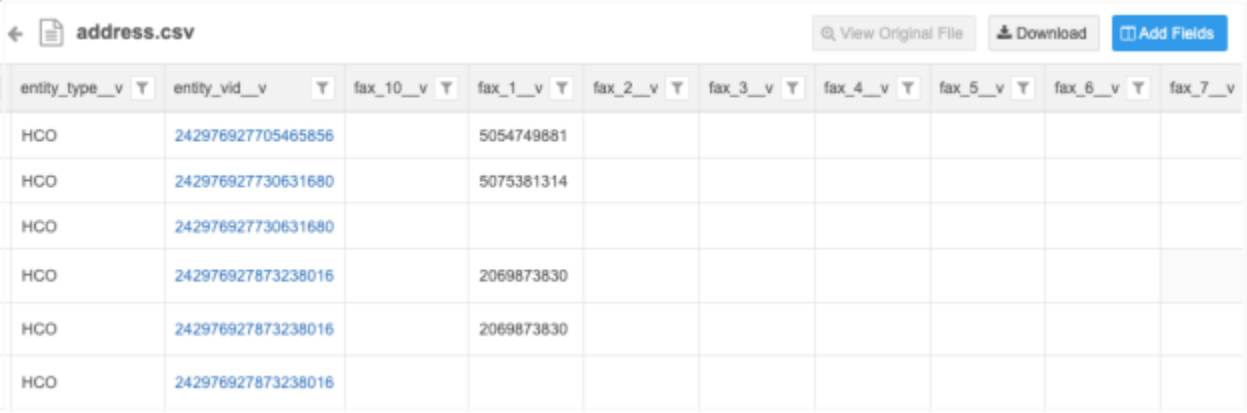
You can also filter the column by searching for specific data; for example, in a Specialty column, you can search for a specific specialty and filter the column. You can filter the smart table on multiple columns. The **Filter** icon in the column header is highlighted to identify sorted and filtered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



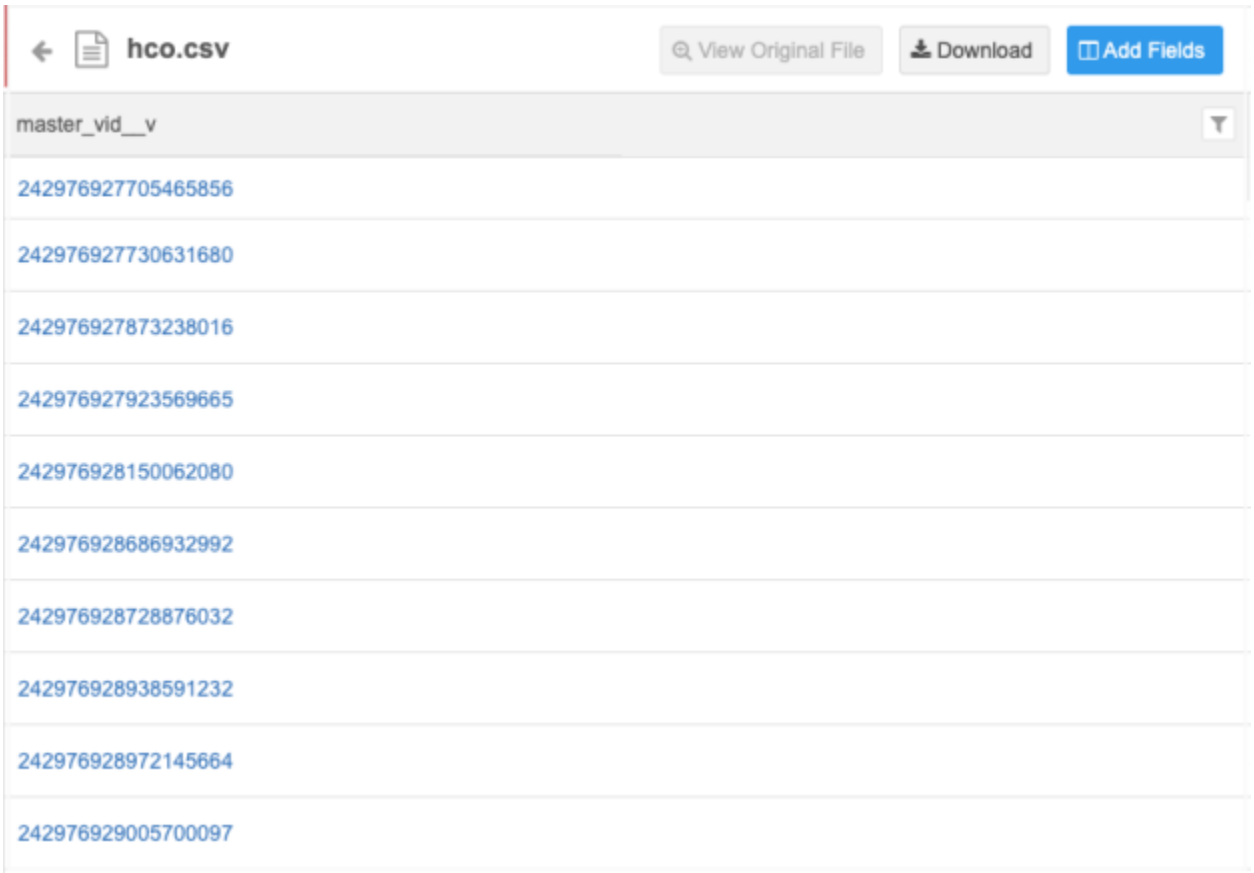
entity_type__v	entity_vid__v	fax_10__v	fax_1__v	fax_2__v	fax_3__v	fax_4__v	fax_5__v	fax_6__v	fax_7__v
HCO	242976927705465856		5054749881						
HCO	242976927730631680		5075381314						
HCO	242976927730631680								
HCO	242976927873238016		2069873830						
HCO	242976927873238016		2069873830						
HCO	242976927873238016								

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.

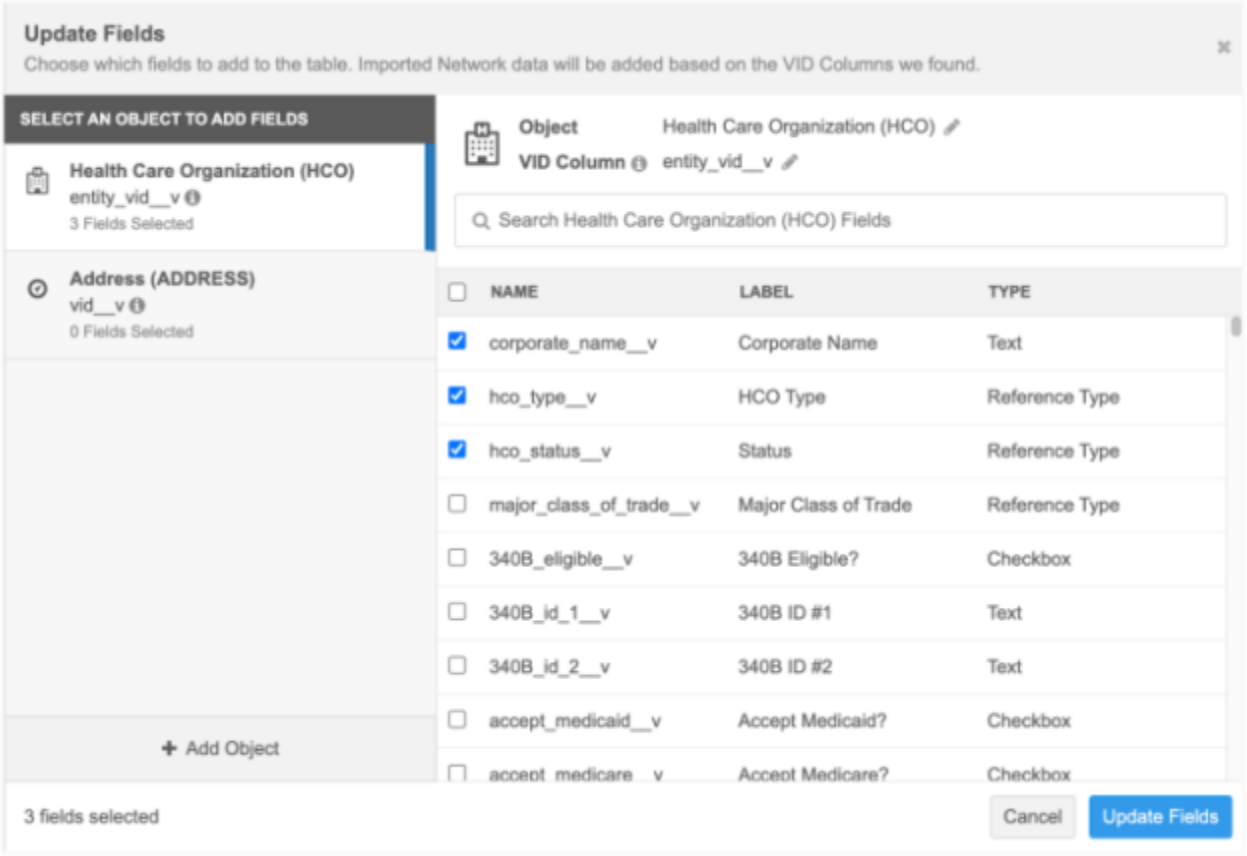


Network uses the VID columns to map the Network fields that can be joined to the file. For example, when Network identifies a VID column as an HCO object, the HCO fields are available to add to the file. If a VID column isn't identified, default field mappings are not available.

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.

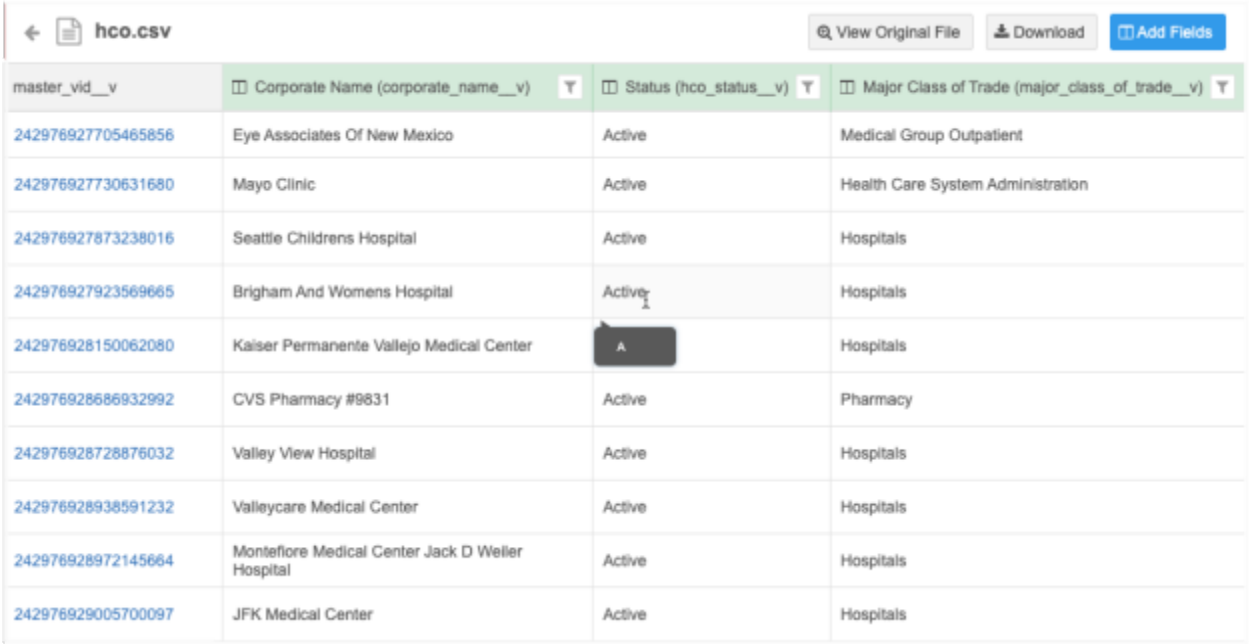
3. Select the fields that you want to add.

   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:

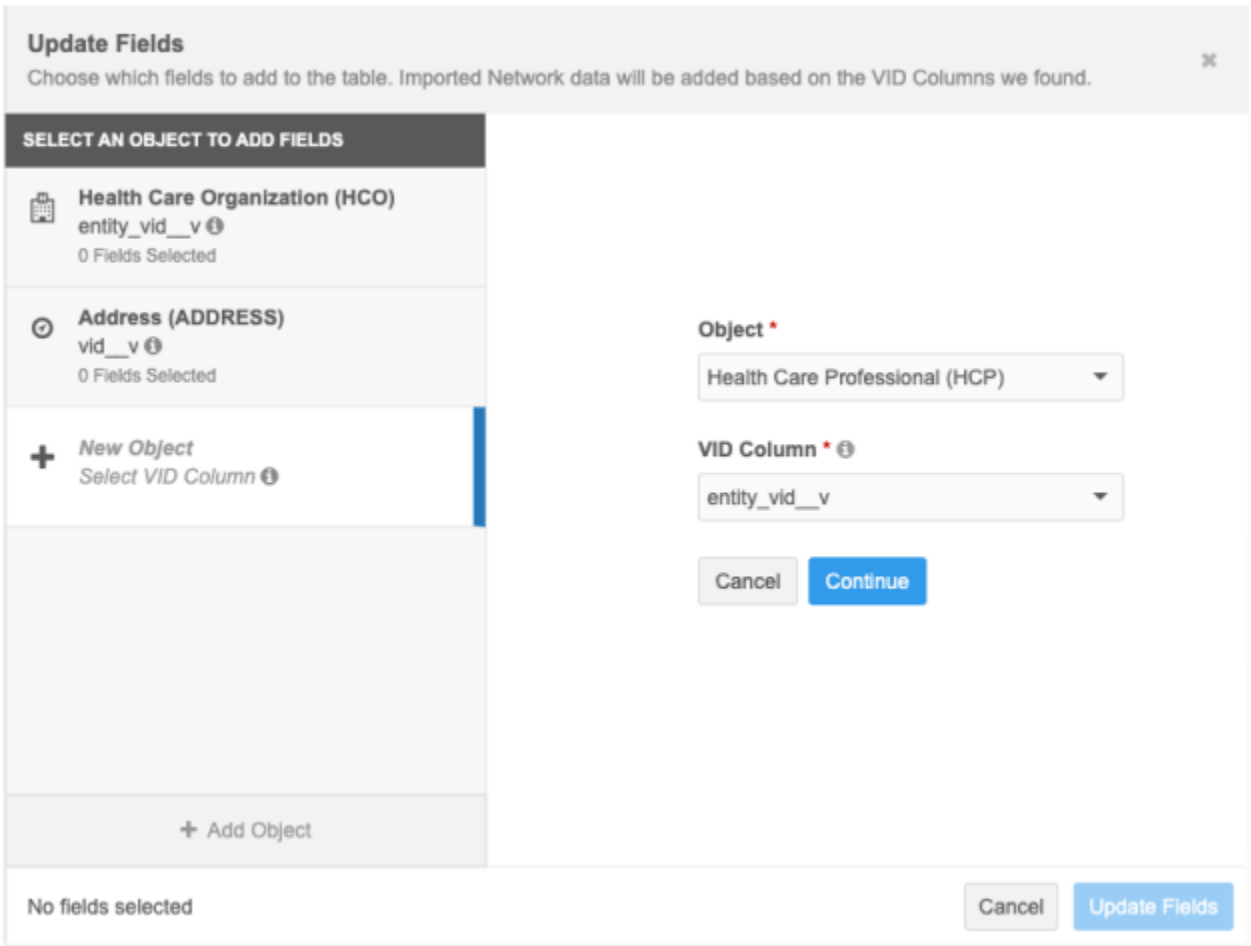
1. **Click Add Fields** and clear the selected fields.
2. Click **Update Fields** to save your changes.

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.



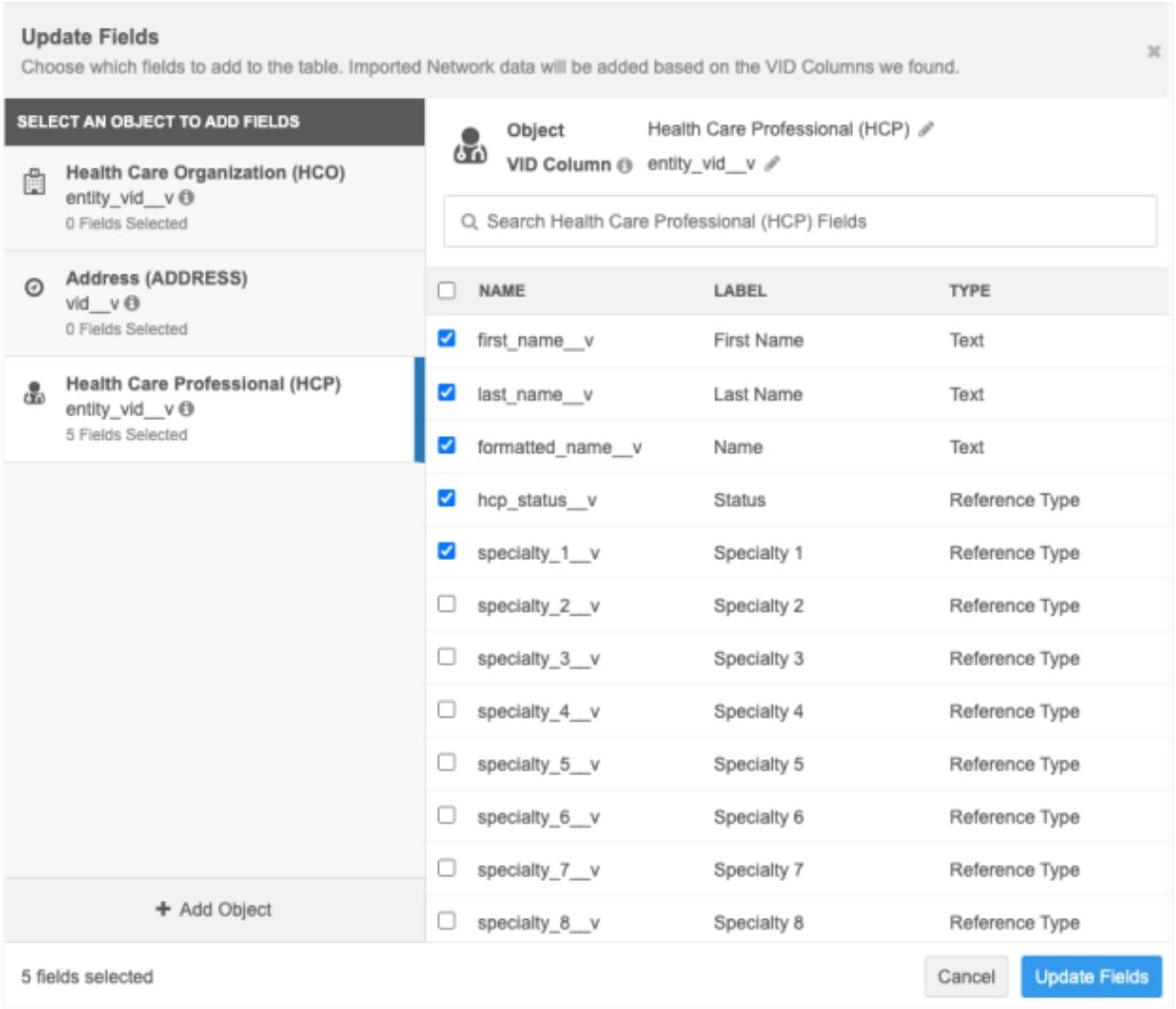
To add an object:

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

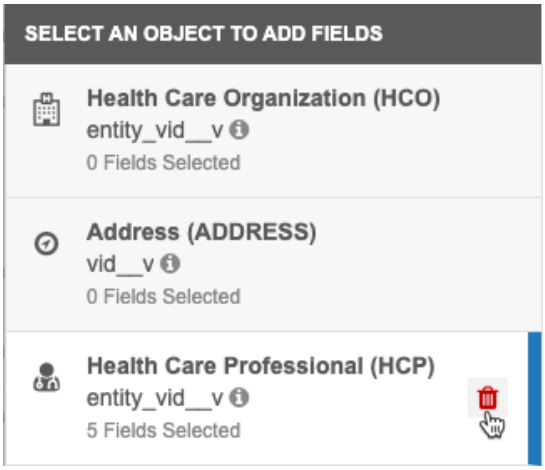
   You can add an object more than once but each **Object/VID Column** mapping must be unique. An object cannot be added and mapped to the same VID column.
4. Click **Continue**.
5. When the object is added, select the fields to add to the smart table.



   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.

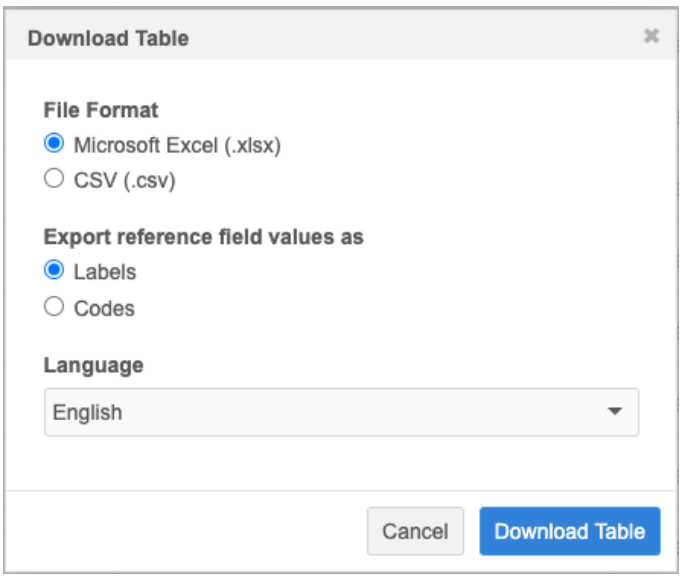


Click **Update Fields** to save the changes. The object and fields are removed from the smart table.

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.

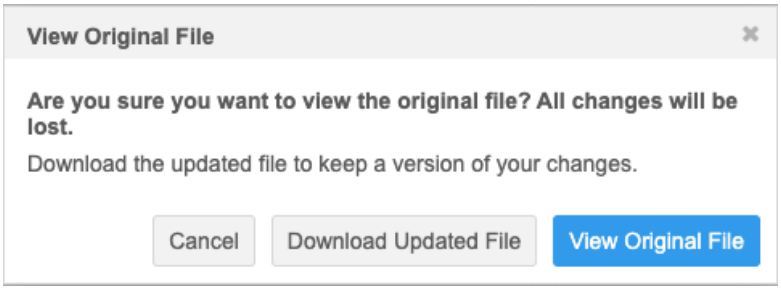
View original file

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.

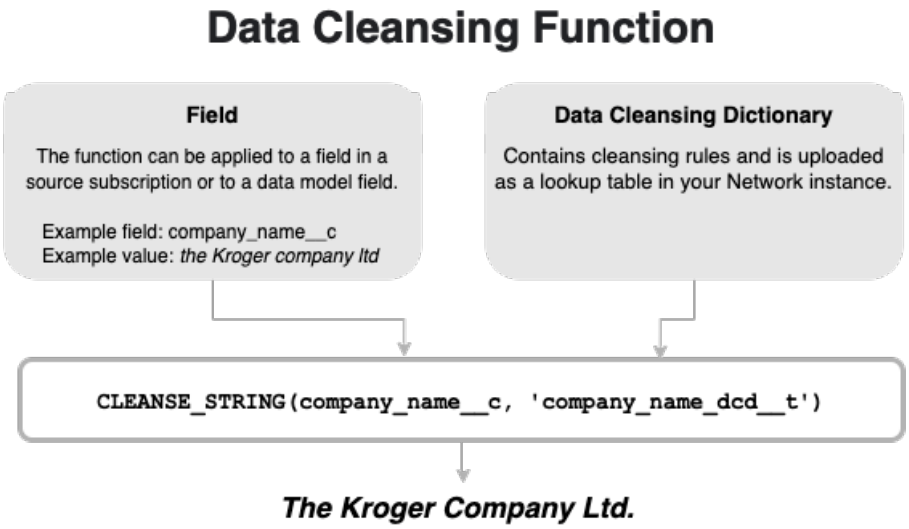
Data quality

DATA CLEANSING

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:

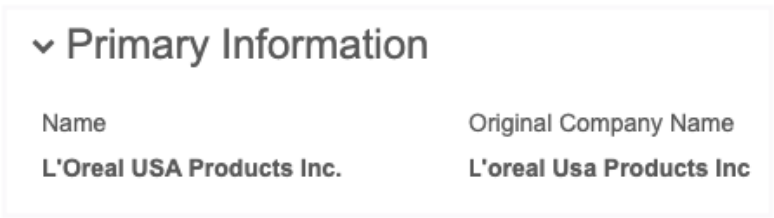
1. Profile your data to understand where data issues are and understand what cleansing rules are needed to fix these data issues.
2. Create cleansing rules in a .csv file. This is your data cleansing dictionary.
3. Upload the .csv file as a lookup table in your Network instance.
4. Create a NEX rule 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_sensititive** - 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, `thousand`, 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.

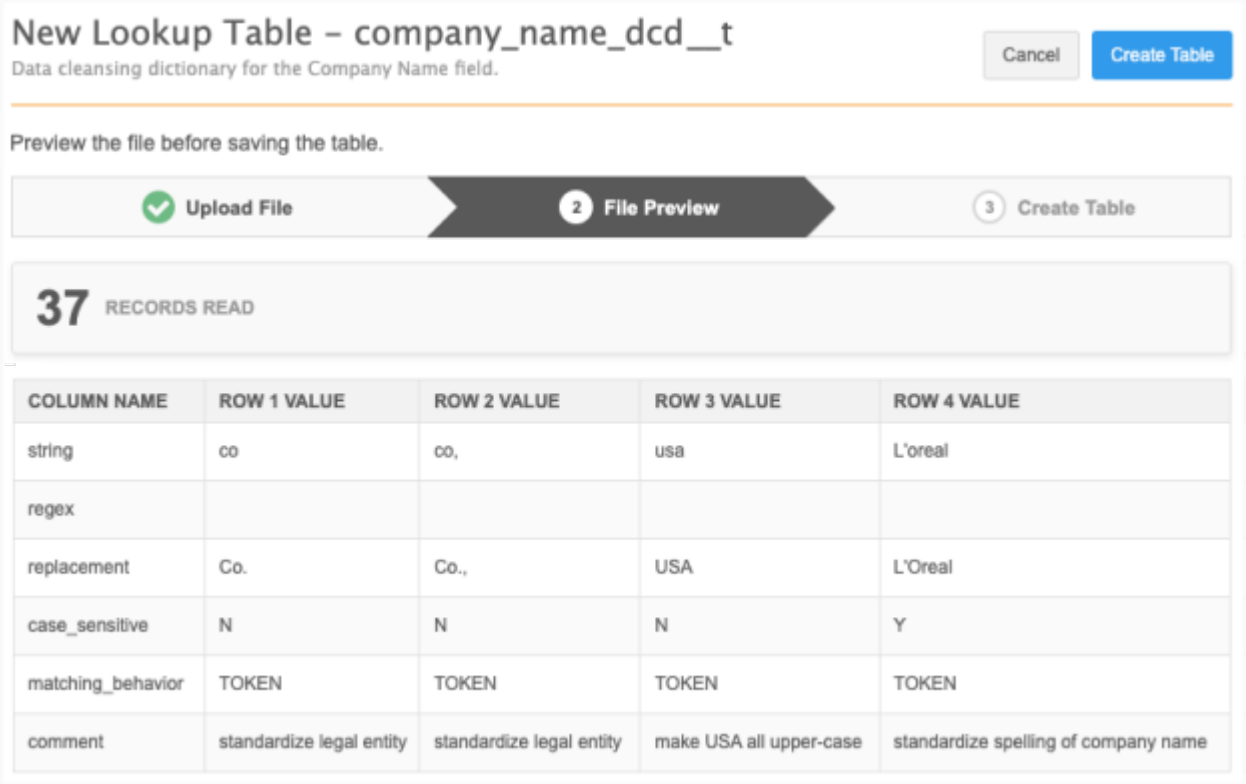
Upload the file to Network

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.

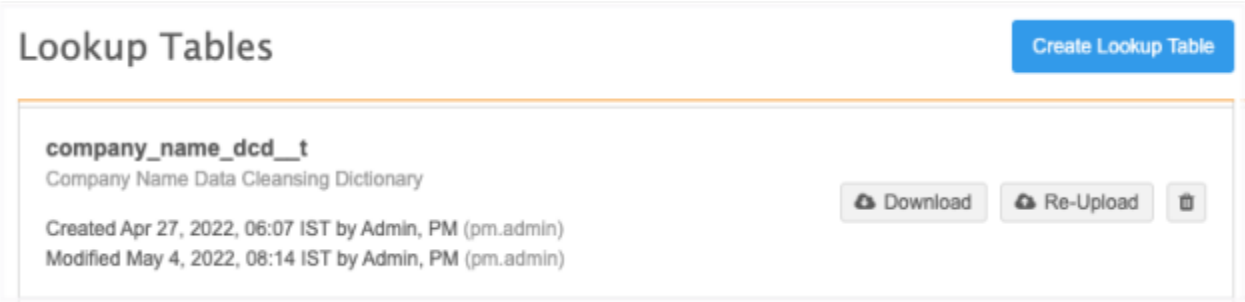
   The file is validated to ensure that it complies with supported lookup table requirements, but the file is not checked to ensure that the mandatory data cleansing columns are included.

   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

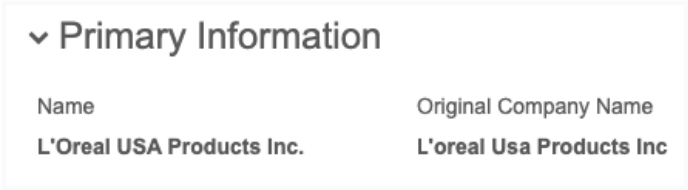
```
CLEANSE_STRING(<field_name>, '<data_cleansing_dictionary_name>')
```

- **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

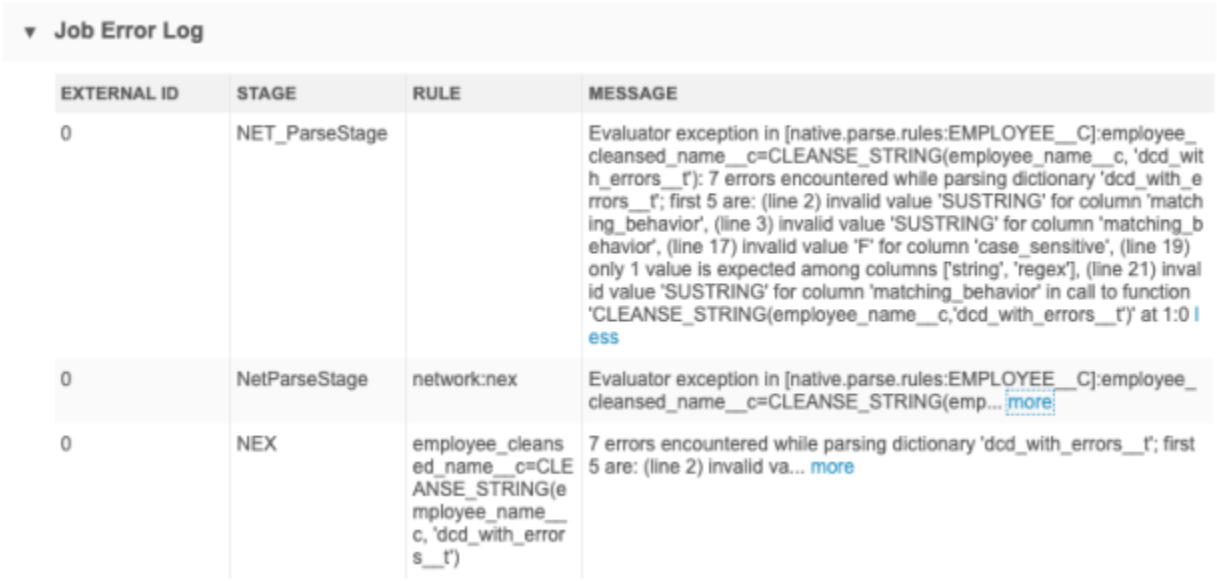
When you upload the .csv file as a lookup table, the data cleansing rules are not validated for consistency and completeness. If mandatory information is missing or if there are ambiguities, the data cleansing function will throw a run-time error with details about these issues.

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.



▼ Job Error Log

EXTERNAL ID	STAGE	RULE	MESSAGE
0	NET_ParseStage		Evaluator exception in [native.parse.rules:EMPLOYEE__C]:employee_cleansed_name__c=CLEANSE_STRING(employee_name__c, 'dcd_with_errors__t'): 7 errors encountered while parsing dictionary 'dcd_with_errors__t'; first 5 are: (line 2) invalid value 'SUSTRING' for column 'matching_behavior', (line 3) invalid value 'SUSTRING' for column 'matching_behavior', (line 17) invalid value 'F' for column 'case_sensitive', (line 19) only 1 value is expected among columns ['string', 'regex'], (line 21) invalid value 'SUSTRING' for column 'matching_behavior' in call to function 'CLEANSE_STRING(employee_name__c,'dcd_with_errors__t')' at 1:0 less
0	NetParseStage	network:nex	Evaluator exception in [native.parse.rules:EMPLOYEE__C]:employee_cleansed_name__c=CLEANSE_STRING(emp... more
0	NEX	employee_cleansed_name__c=CLEANSE_STRING(employee_name__c, 'dcd_with_errors__t')	7 errors encountered while parsing dictionary 'dcd_with_errors__t'; first 5 are: (line 2) invalid va... more

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.

Data model

COUNTRY SUPPORT

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

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

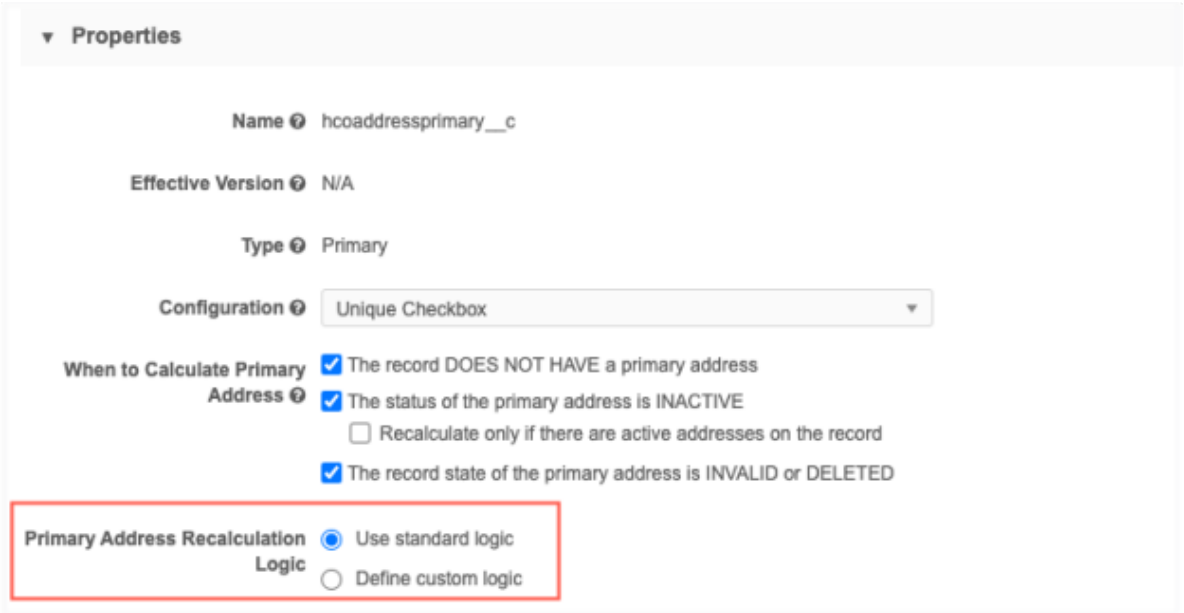
Customers can enrich addresses by adding cluster codes from third party cluster providers. In this release, Network has included support for the following countries for the cluster provider, IQVIA™.

- 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

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.

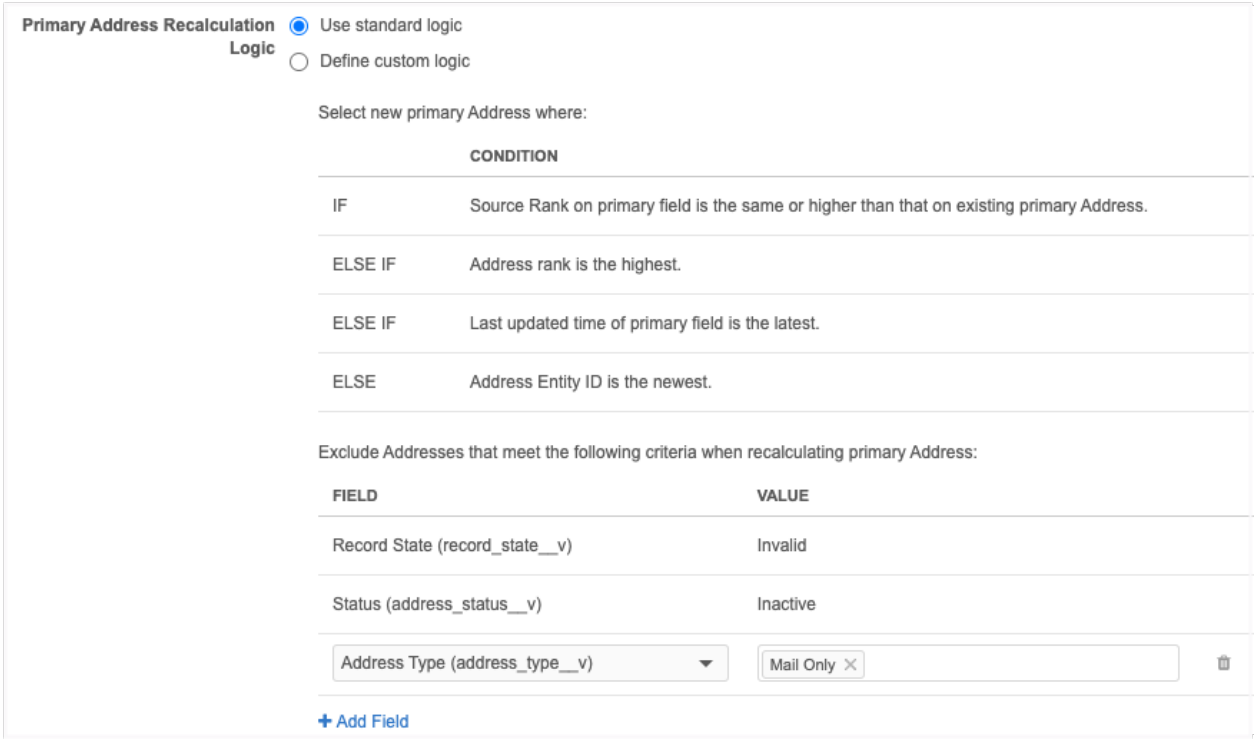
1. On a primary address field that uses the Unique Checkbox configuration, select an option for calculating primary address. For example, select **The status of the primary address is INACTIVE**.
2. In the **Primary Address Recalculation Logic** section, choose one of the options:
    - **Use standard 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.

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	24324400374 5596484	9321845422743 24385	9329263649201 87752	2432440037455 29263	2432440037455 96476

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:

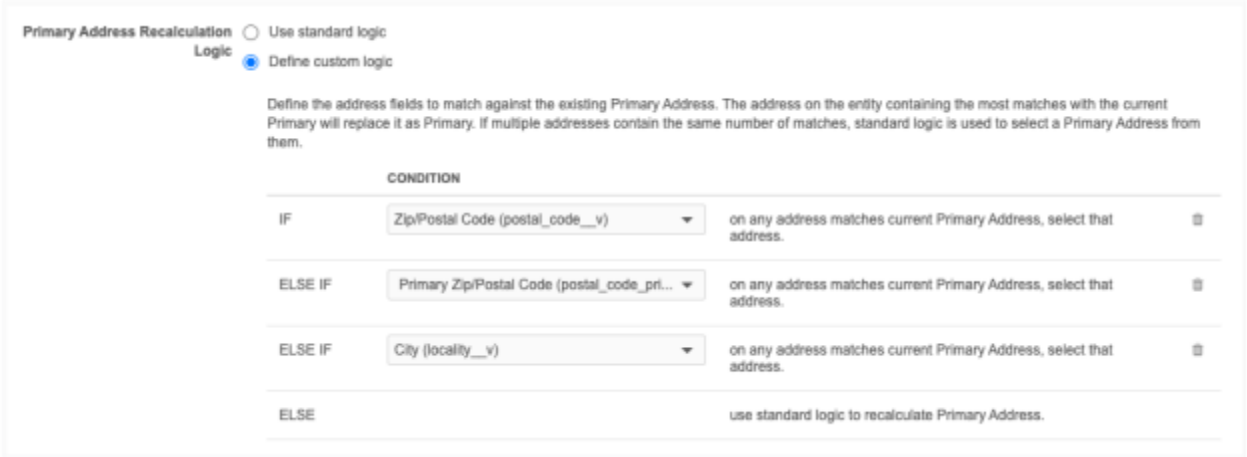
1. In the **Condition** section, expand the list and choose an address field.
2. To add another field to match on, click **Add Condition**. Select the field.

   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	2432440037455 96484	93218454227 4324385	9329263649201 87752	24324400374552 9263	243244003745 596476

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

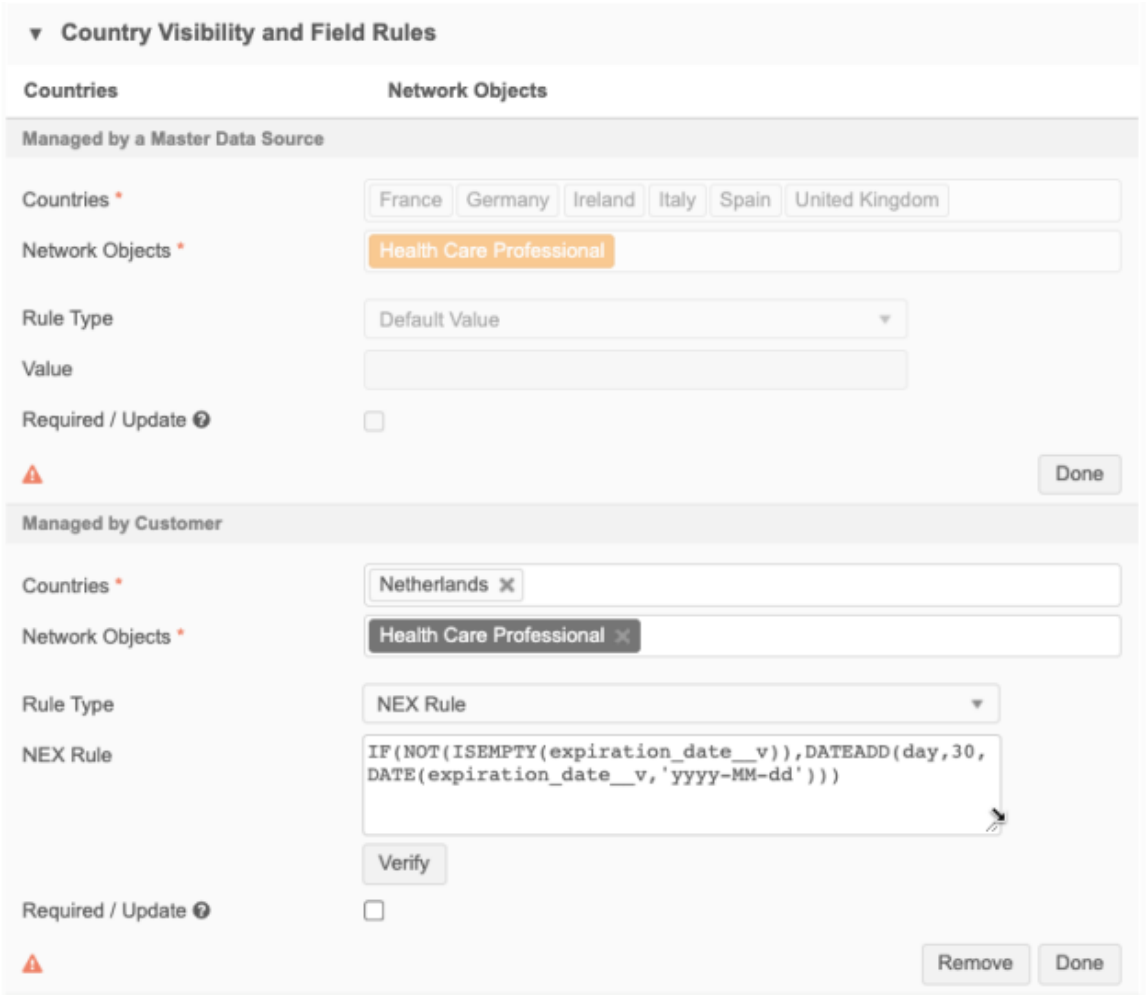
Logs

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

FIELD CONFIGURATIONS

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.



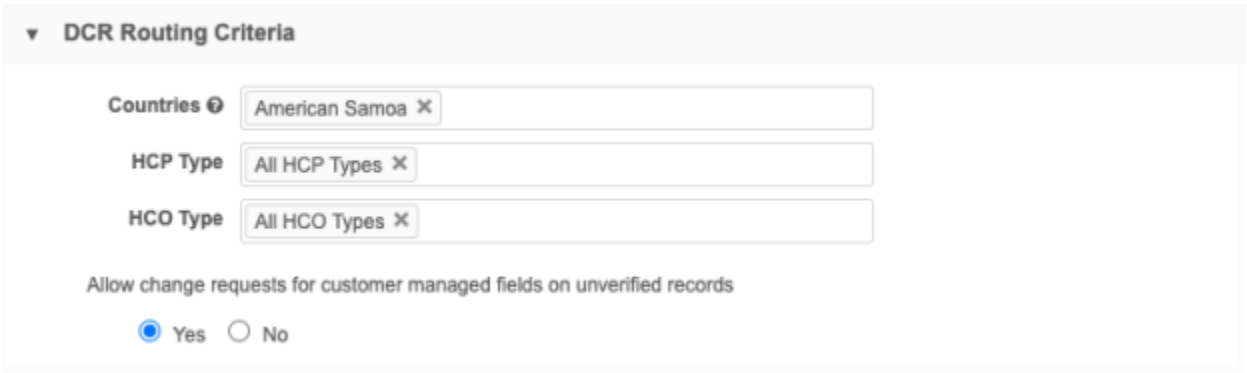
These enhancements are enabled by default in your Network instance. They apply to the **Managed by Customer** section for standard fields and custom fields.

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

Data sources

DCR ROUTING FOR THIRD PARTY SYSTEMS

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 to auto-accept customer owned fields 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

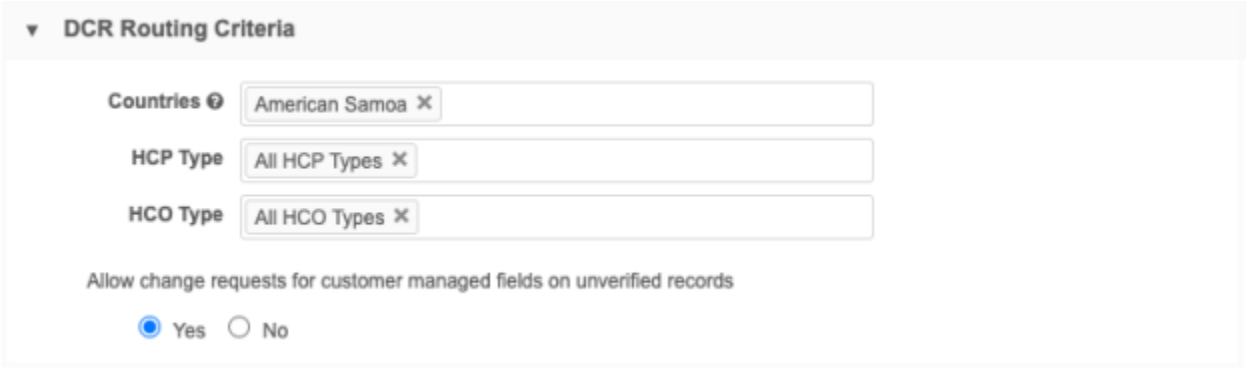
If the add request is not approved, any pending local DCR tasks are rejected (Invalid and Merged_Into records cannot be updated). If the local DCRs have already been approved, the unverified (Under_Review) local record is invalidated and any record profile changes are removed.

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**.

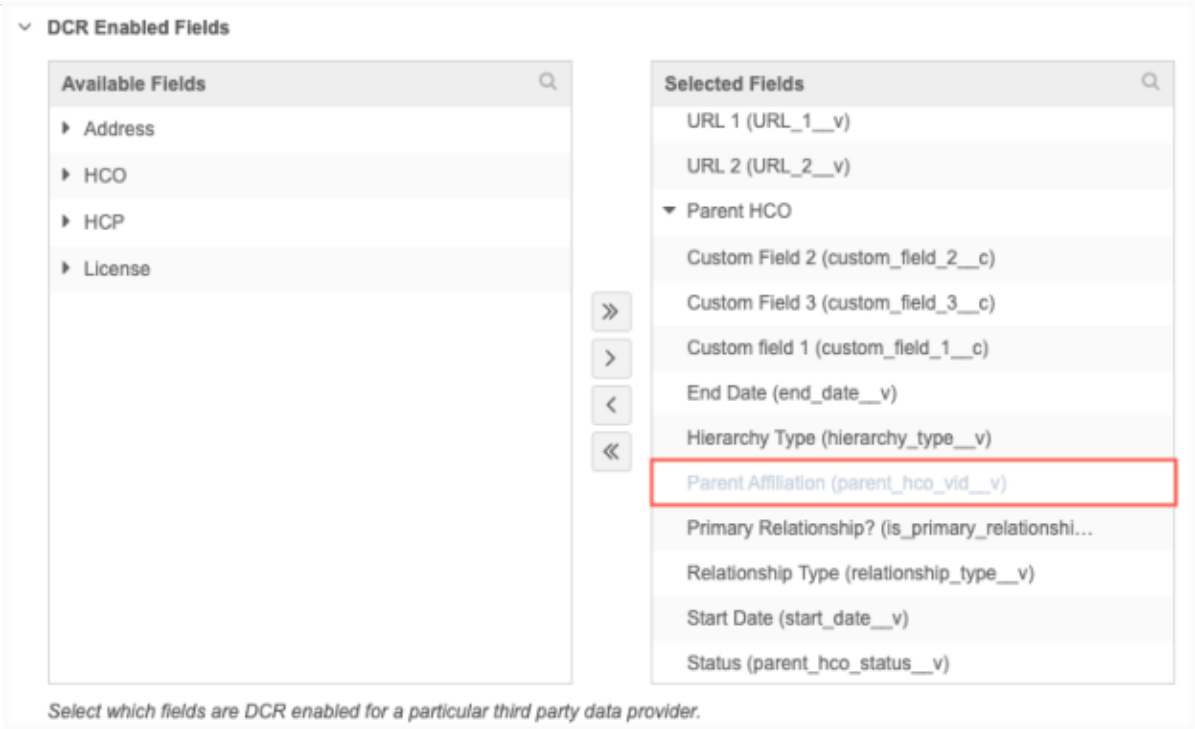
   When **No** is selected, the change request will be automatically rejected. The resolution note that is applied to the rejected task is `R-00014 System rejected - you are trying to update an Under Review record that is currently locked. Please resubmit your change request when the record is open for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

When the Parent HCO object is selected as a **DCR Enabled Field** on a third party system, the `parent_hco_vid__v` field is now automatically moved to the **Selected Fields** pane. This field is mandatory for the Parent HCO object integration.



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

Administrators can now export the job error log after an OpenData 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:

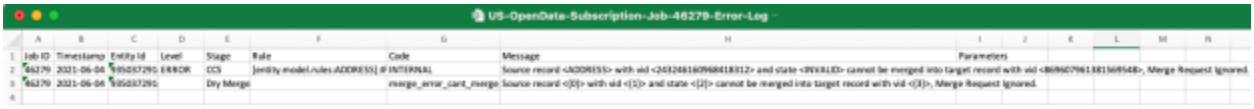
1. In an OpenData country subscription, click a job **ID** in the **Job History** section.
2. On the Job Details page, if errors occurred, the **Job Error Log** section displays. Click the **Export Error Log** button.

   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

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. 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 for 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, 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, the 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

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.

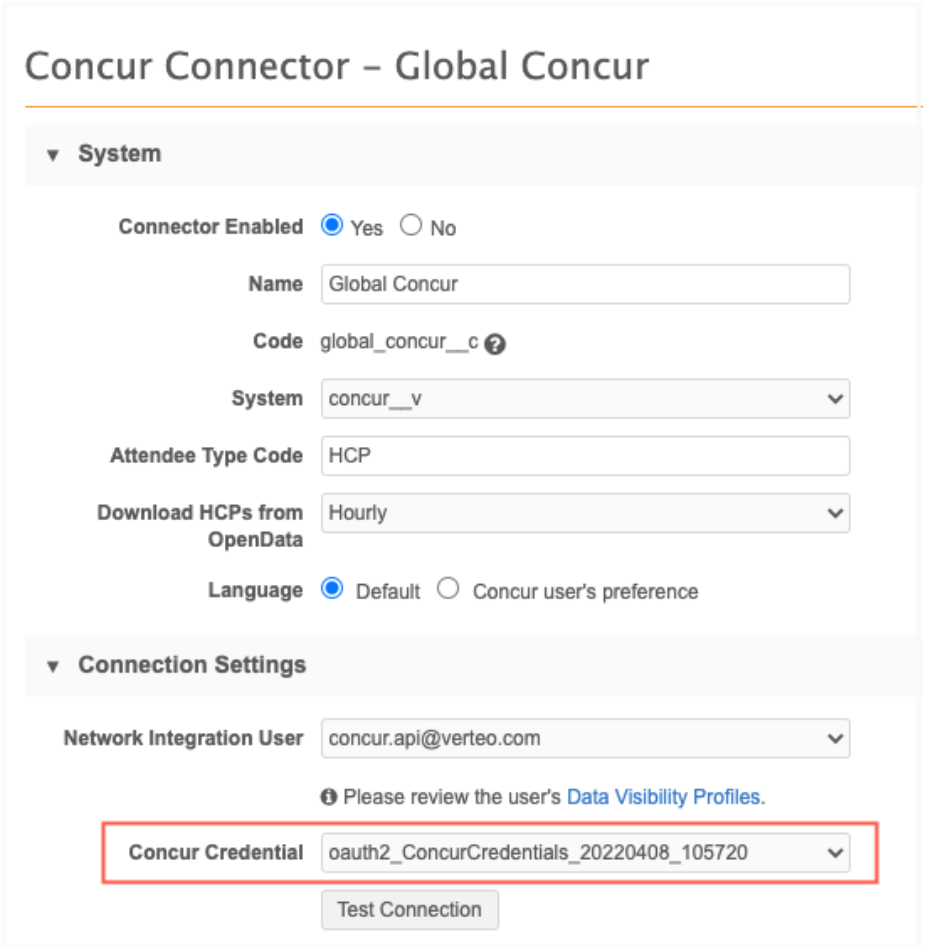
Existing customers will be automatically migrated to use OAuth2 authentication. New customers can create credentials that use OAuth2 to connect their Network instance to SAP Concur.

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.



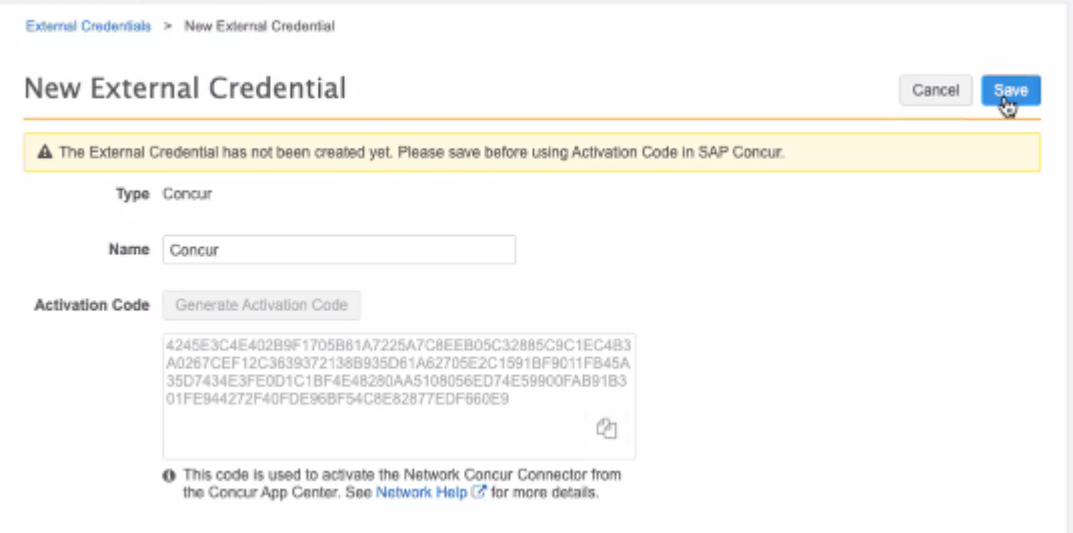
Creating Concur credentials

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:

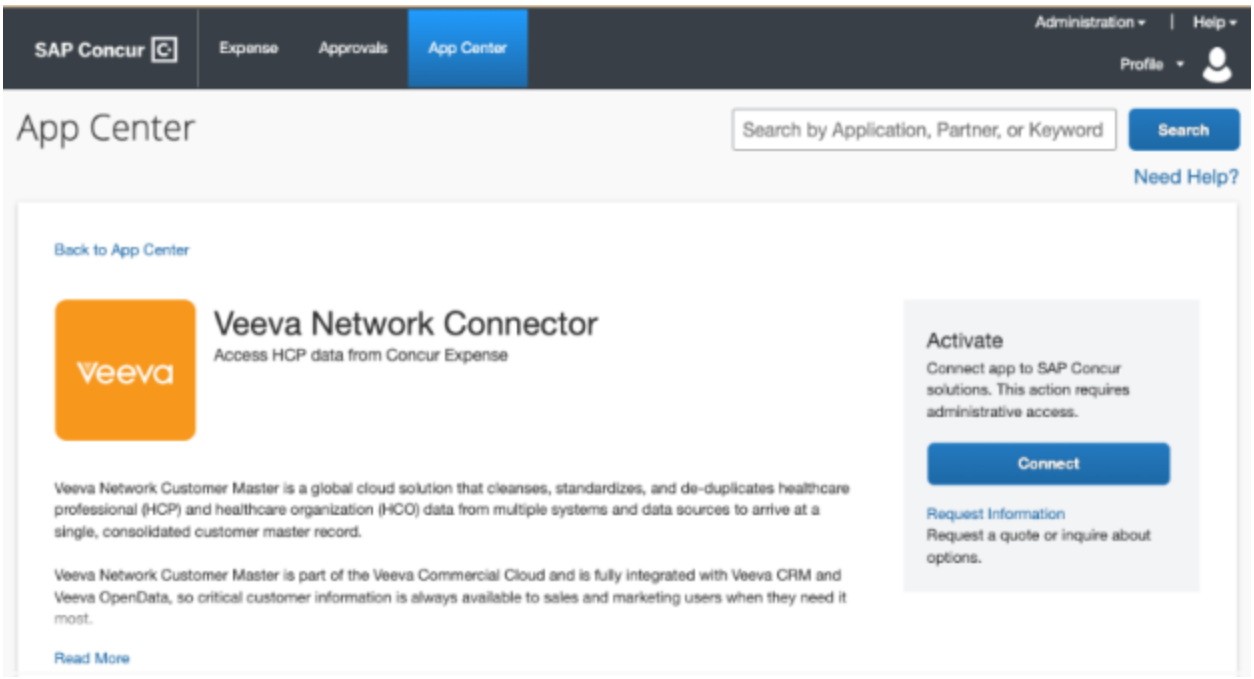
1. In the Admin Console, click **Settings > External Credentials**.
2. Click **Add Credentials** and select **Concur** from the list. Click **Continue**.
3. On the New External Credential page, type a **Name** for the Concur OAuth2 credentials.



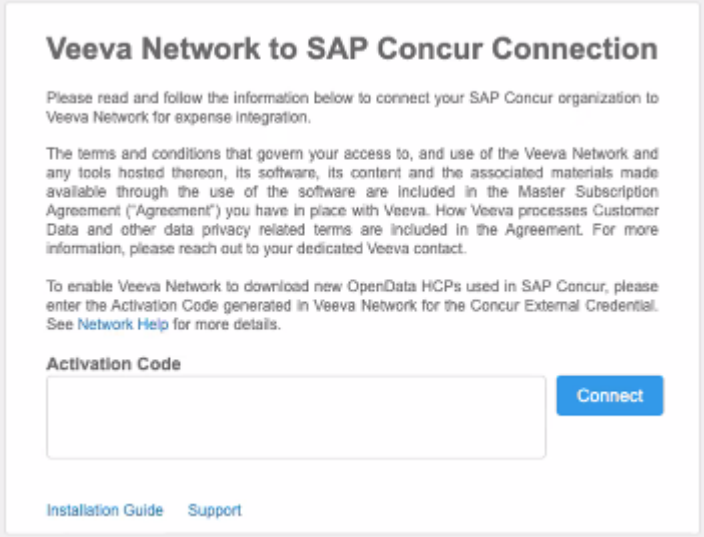
4. Click **Generate Activation Code**.

   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.



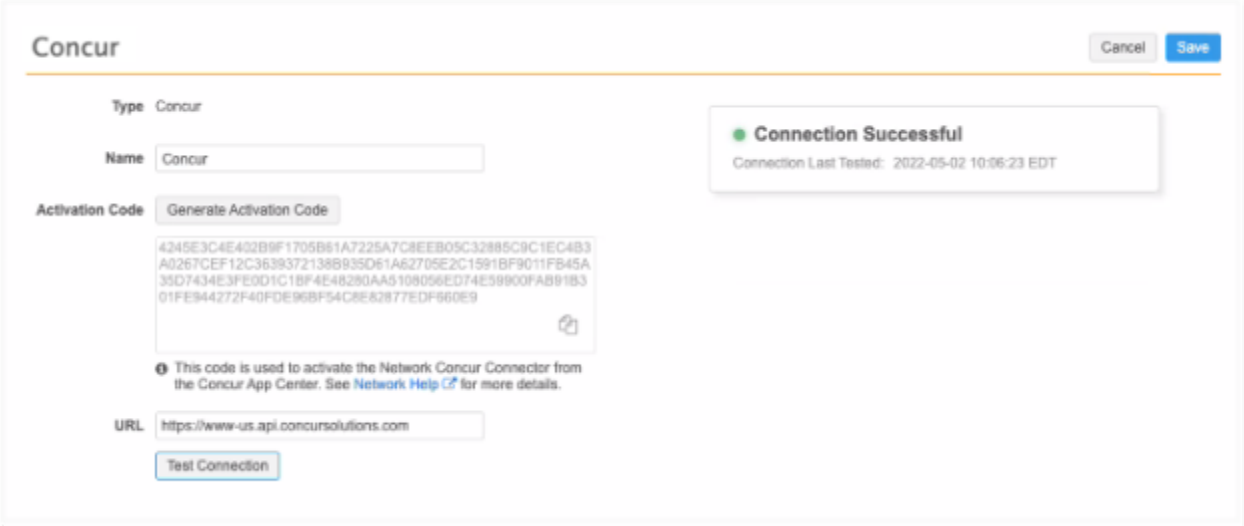
9. On the Veeva Network Connector app listing page, click **Connect**.
10. In the pop-up, accept the Terms and Conditions.
11. In the landing page that is hosted by Network, paste the **Activation Code** that you generated in Network. Click **Connect**.



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**.



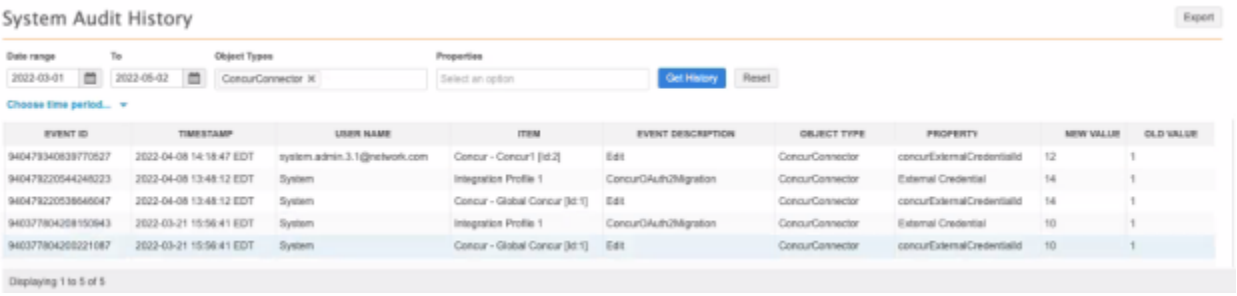
When the Connection Successful displays, your Network instance is authenticated to SAP Concur.

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`.

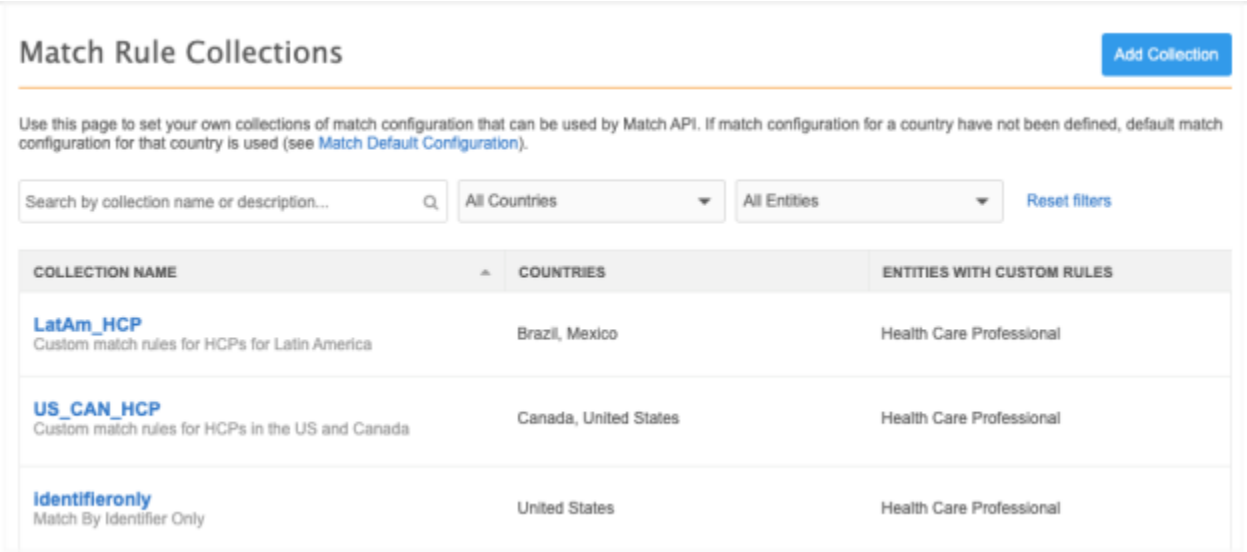


API

MATCH RULE COLLECTIONS

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"
   },
   ...
 }
```

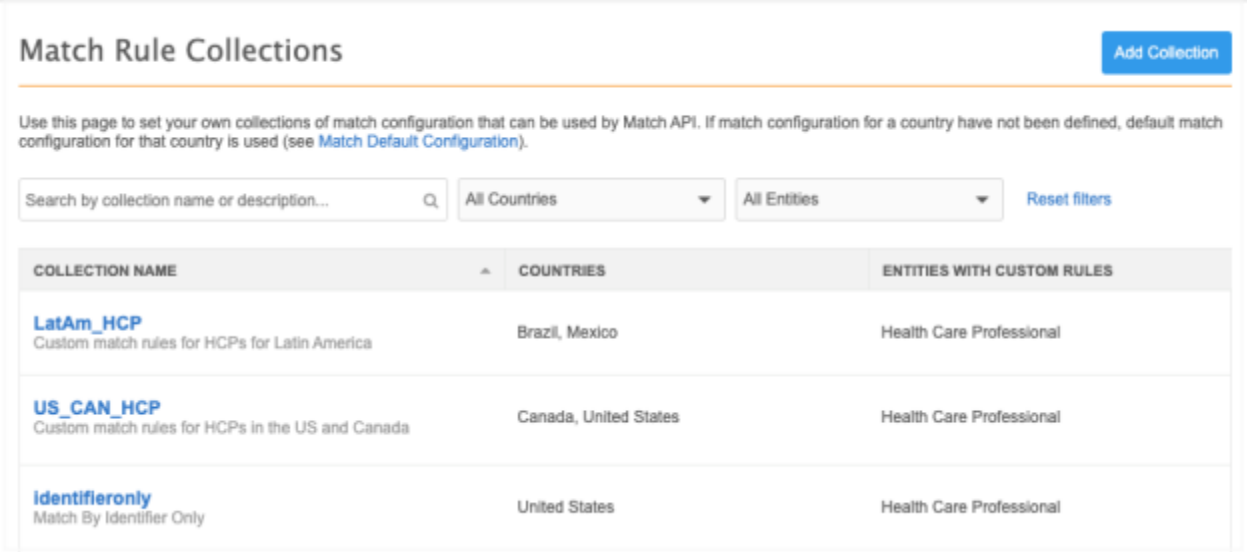
Errors

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:

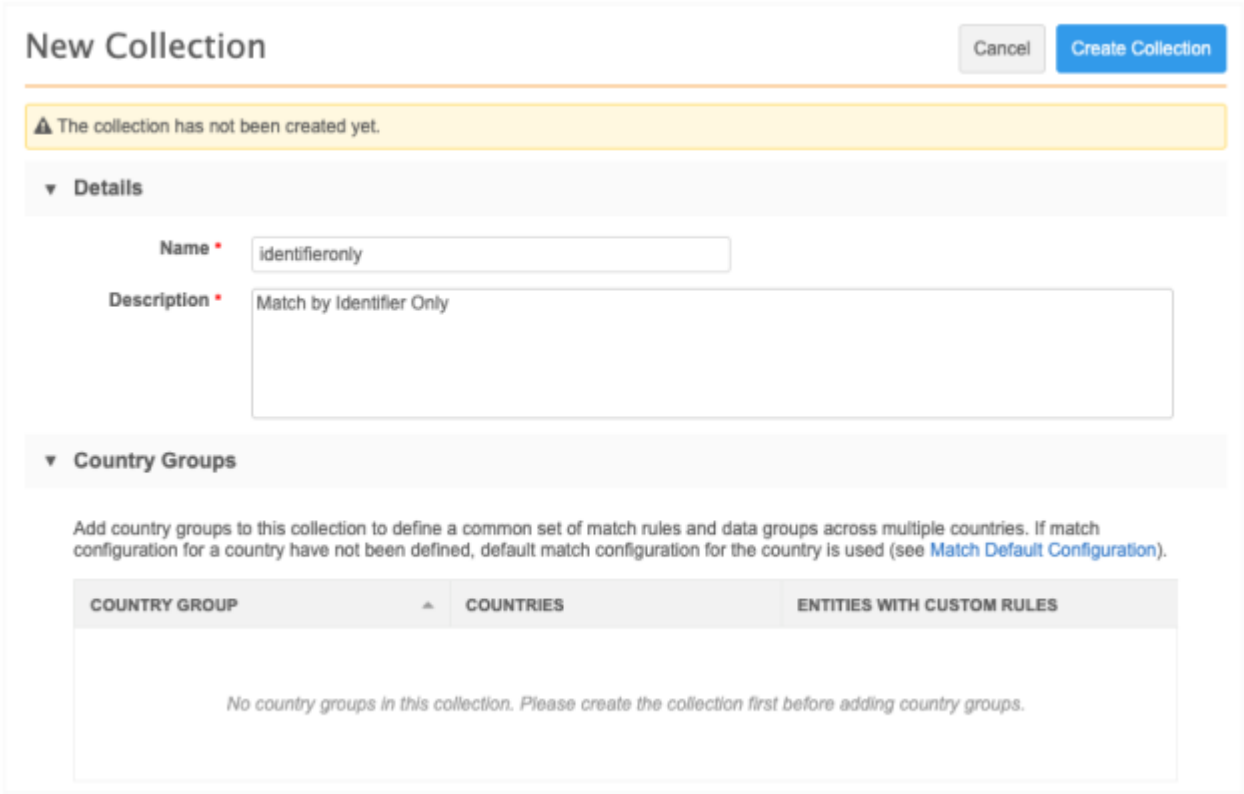
- **Collection Name** - The name and description.
- **Countries** - The countries in all of the country groups defined in the collection.
- **Entities with Custom Rules** - The objects that have custom rules defined. Objects in the collection that use the default match rules do not display in this column.

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

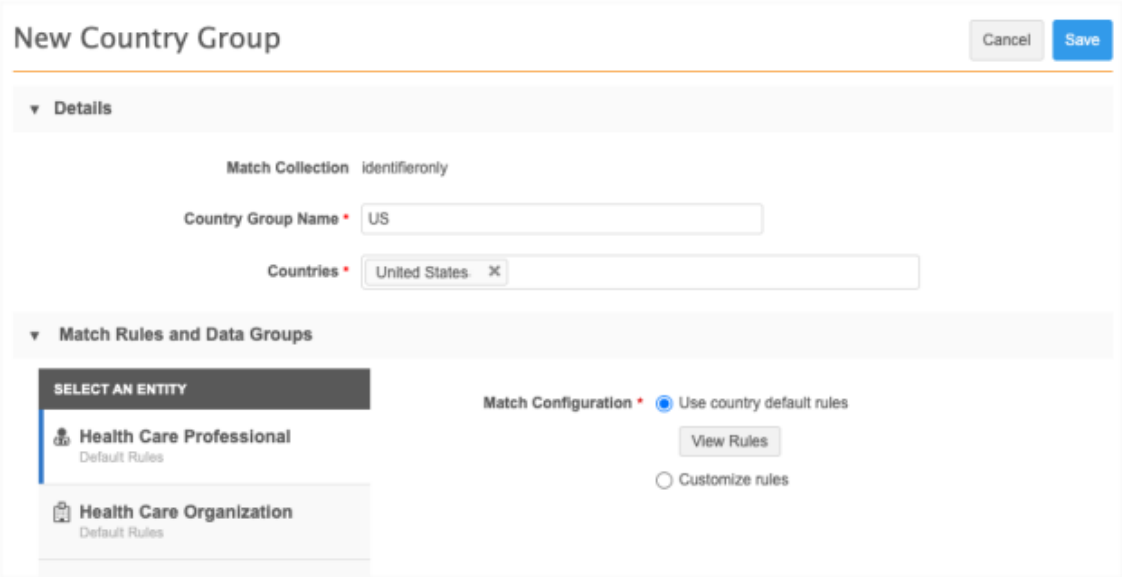
To create a match rule collection:

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.

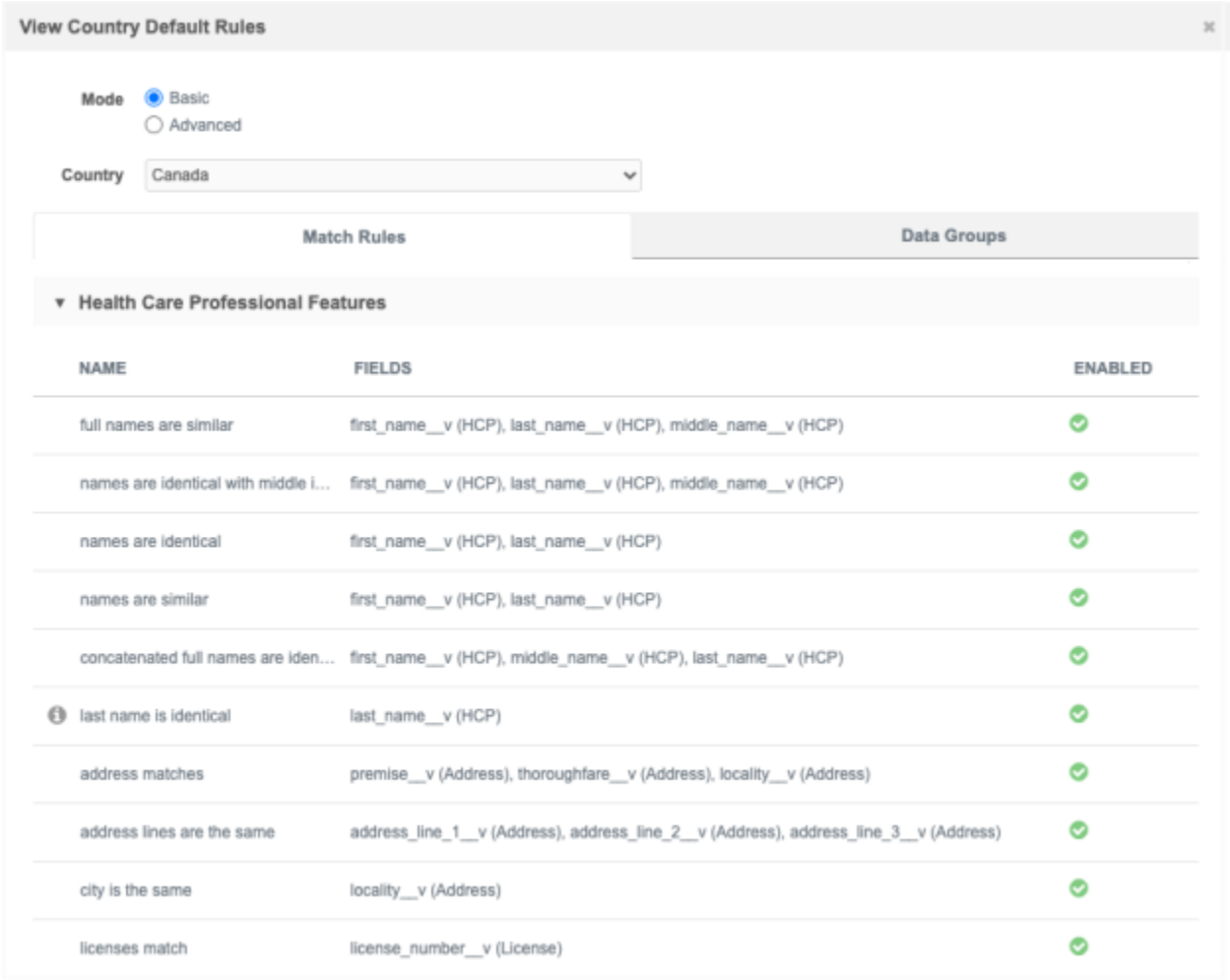


4. Click **Create Collection** to save it. The collection must be saved before you can add country groups.
5. Click **Add Country Group** . Match collection rules are specific to country groups. Country groups are countries that use the same match rules and data groups.
6. On the New Country Group page, 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.

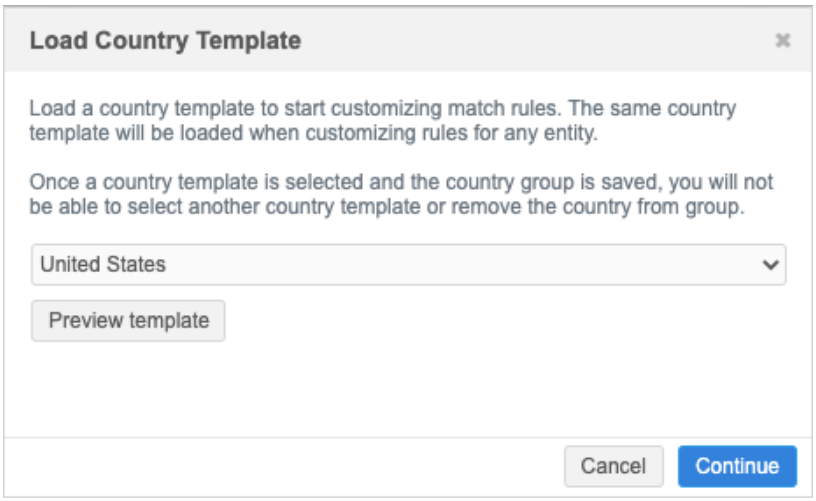
   If the country group contains multiple countries, use the **Country** list to see each country's default rules.



   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.

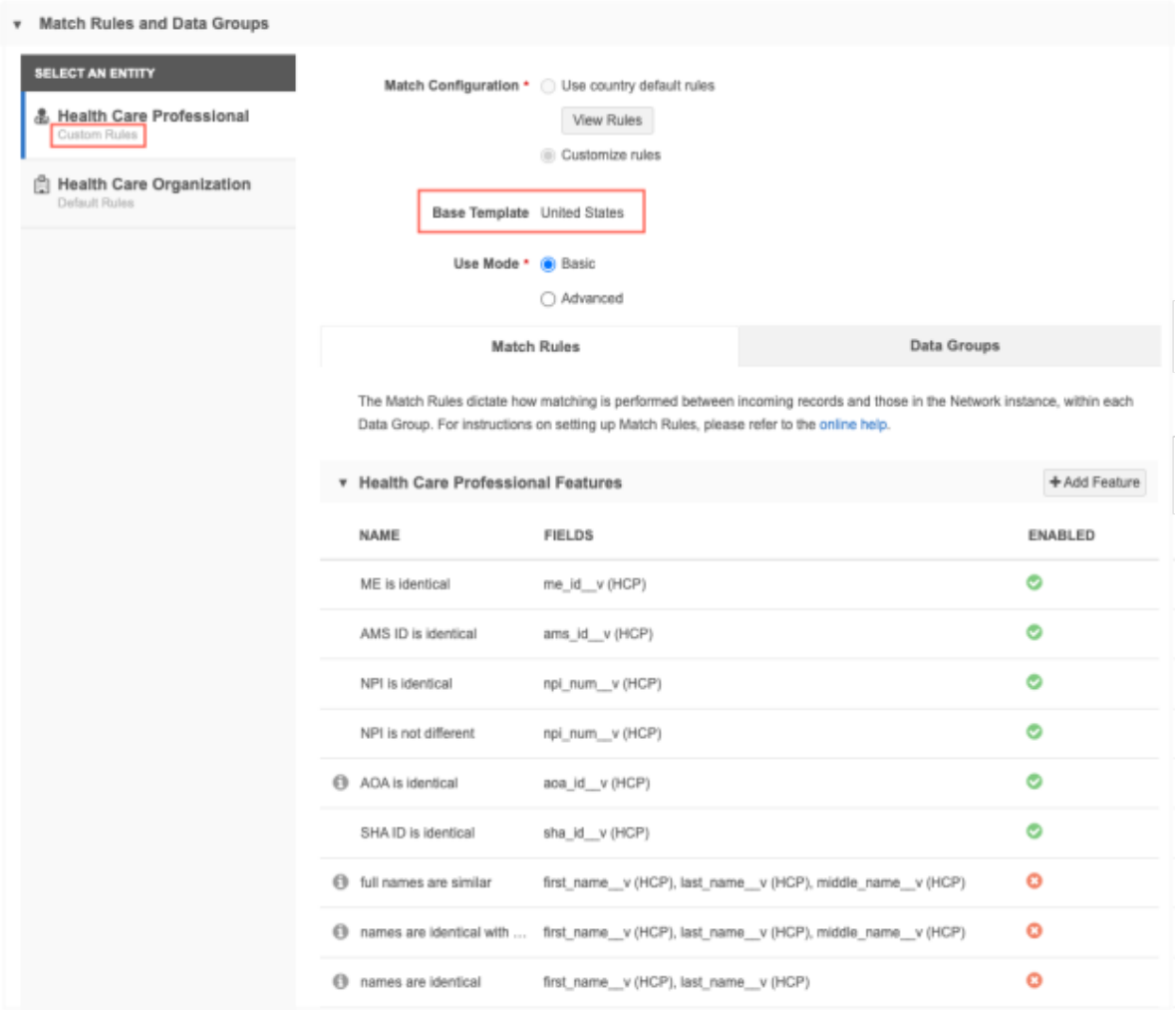


Note: After you save the country group, the template cannot be changed and the country cannot be removed from the country group.

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.



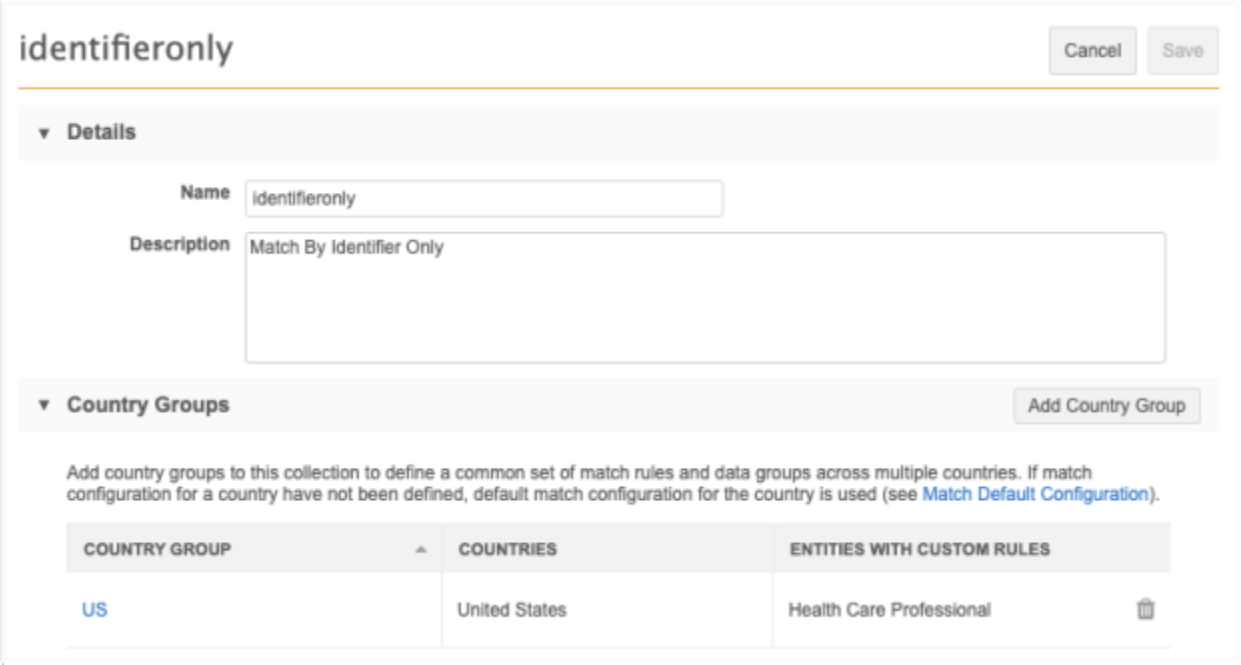
Notice that the **Custom Rules** label displays below the selected object. Also, the country group is updated to display country that the **Base Template** uses.

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.



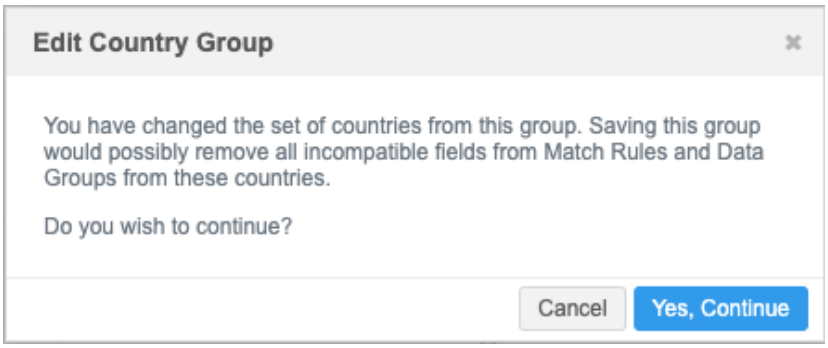
13. *Optional* Add more country groups to the match rule collection.
14. Save the match rule collection.

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.