Veeva Network

Veeva Network 23R2.1.1 Release Notes

October 2023

Contents

About these Release Notes

These Release Notes describe all features that are included in Veeva Network 23R2.1.

SUBSCRIBE TO RELEASE NOTIFICATIONS

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

Software releases and maintenance

- Veeva Trust Site

  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

The documents are posted in the following locations:

- Veeva Connect - Join the Network Community.

  To be notified as soon as the Release Notes are posted, go to your Veeva Connect profile and click **Settings**. On the Email Frequency page, expand the list and choose **Immediate**. Other notification options are **Daily** and **Weekly**.

- Veeva Product Support Portal

  Follow the Network Release Notes section to be notified when release documents are posted.

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 Release Notes were 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 23R2.1 minor release.

		ST	DS	DM	AD
Hierarchy Explorer widget					
Accessing Summary View	A button displays beside each record so you can view the record hierarchy in a pop-up.	●	●	●	●
Summary View enhancements	The view is updated to include additional details on ancestor records and the ability to export groupings.	●	●	●	●
HCP Summary View	A summary view is now available for HCP records.	●	●	●	●
Integrations					
Transformation rules	Create rules to transform field values (ex. override OpenData values) in the APIs and in files exported to Veeva CRM and other downstream systems.			●	●
Survivorship					
Time-based rules	Record level rules can be created to ensure that the most recent information wins survivorship.			●	●
Data maintenance jobs					
Job validation rules	Use rules to ensure that records are not mistakenly deleted or unsubcribed.			●	●
Source subscriptions					
Simulating updates	Run jobs in simulation mode so you can preview the outcome before committing the data.			●	●

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.

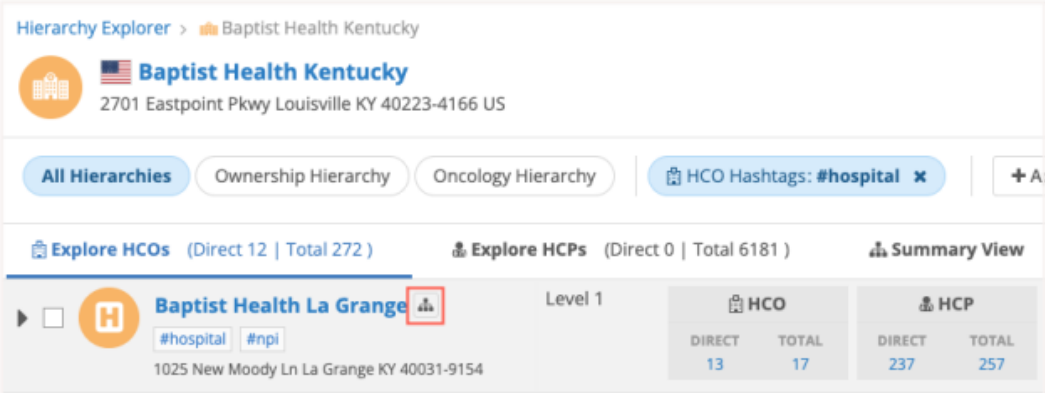
Hierarchy Explorer

The following enhancements have been added to the Hierarchy Explorer widget in this release.

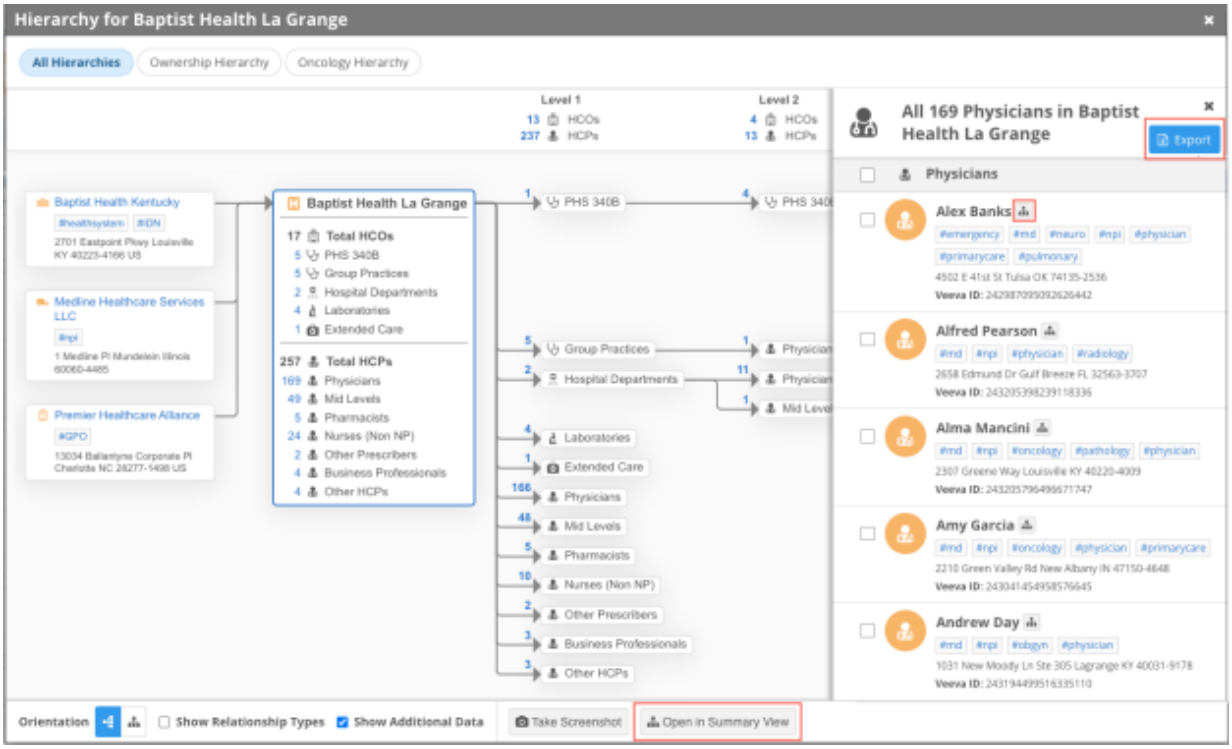
The Hierarchy Explorer widget is available by default in your Network instance. If it is enabled, these enhancements are enabled by default.

Accessing Summary View

A **Hierarchy** icon displays beside each record so you can quickly access the data to keep exploring and understanding the organization.



The Summary View opens in a pop-up, instead of the tab, with that record in focus.



Click on any of the group counts to open a pane and view the list.

All of the functionality in the Summary View tab is available in the pop-up except HCO/HCP filters. If you want to filter the accounts, use the **Open in Summary View** button to display the hierarchy in the tab.

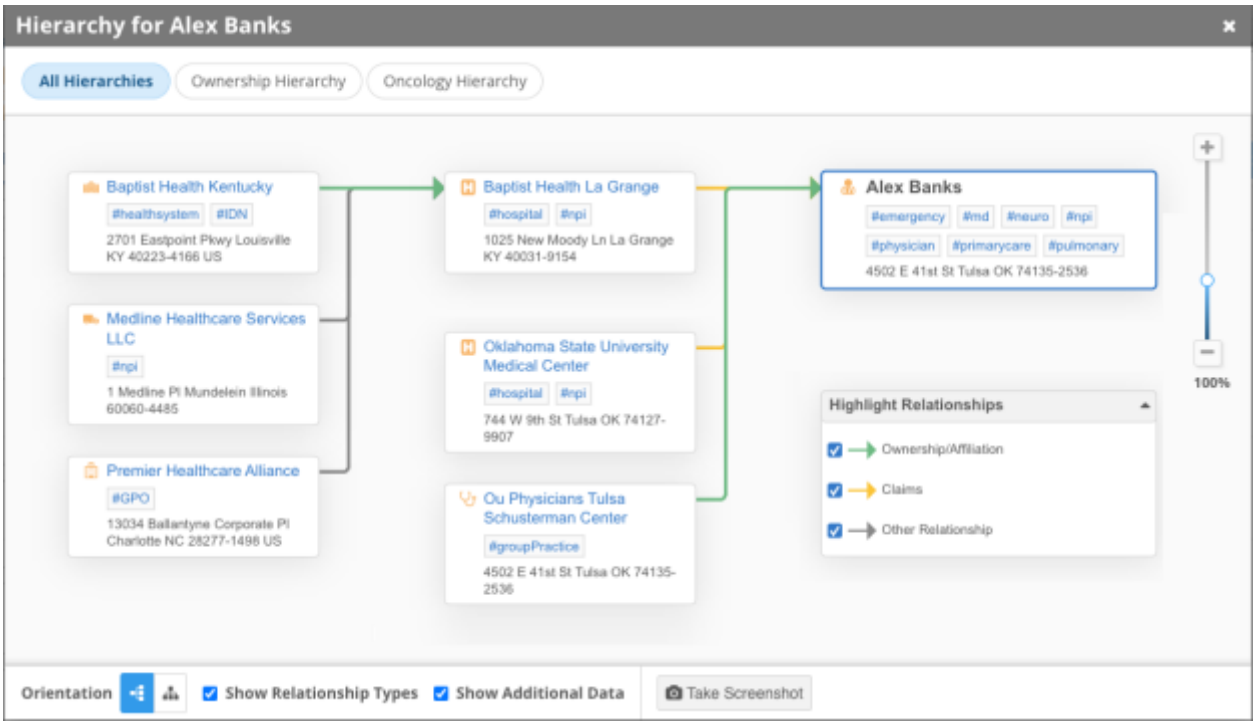
SUMMARY VIEW ENHANCEMENTS

The following updates are available in the Summary View tab and the Summary View popup.

- **Export** - The **Export** button is available when you click any of the HCP or HCO group counts so you can download the list.
- **Ancestors** - Hashtags and addresses display on the ancestor nodes to provide more information about these HCOs. These details are available when the **Show Additional Data** setting is on. The setting is on by default.

HCP SUMMARY VIEW

A summary view is now available for HCP records so you can see a diagram of the associated HCOs.



Tip: Use the custom hierarchies or **Show Relationship Types** settings to understand the different relationships for the HCP.

Click an HCO name to open the Summary View for that HCO.

Integrations

TRANSFORMATION RULES

Administrators can create rules to transform field values in the APIs and in export files for downstream systems. You can also use transformation rules to override OpenData values before records are pushed to Veeva CRM.

Example use cases:

- Remove specific address types from Veeva CRM, for example, mailing addresses.
- Display only the Ownership Hierarchy in Veeva CRM (remove the noise of relationships for reps in CRM)
- Set fields (for example, Specialty or Medical Degree) with no value to a custom value.
- Limit the `postal_code__v` field to only five digits for US addresses
- Hide OpenData addresses in Veeva CRM if they have an ordinal greater than 10.
- Change all corporate names to uppercase characters
- Override OpenData's retired HCP status based on a target flag.

Note: Transformation rules can be applied to locally managed records and records managed by OpenData and third party data providers.

This feature is on by default in your Network instance.

Benefits

- Retain original data in Network but transform the data output for your downstream system.
- Each system can have its view of the data. Rules are assigned to specific systems, so you can customize data for different systems.
- Flexible integrations with Veeva CRM and other downstream systems. Using transformation rules reduces the need to create customizations for other systems.
- Network API support. Data can also be transformed using transformation queries in target subscriptions, but transformation queries are not supported for API requests.

Process overview

- Define a rule or set of rules for a specific source system. Rules are created using Network Expressions for a specific object and field.
- Apply the rules to target subscriptions or the API.
- Run the job to transform the data and then export it to the downstream system.

Veeva CRM integration

Use transformation rules to customize the data that is pushed to CRM.

By applying transformation rules on target subscriptions and to the integration user, the rules can then be applied when data is pushed to Veeva CRM.

This includes the following processes in CRM:

- Network Account Search
- DCR Inbound Process
- Network Bridge subscription

Filtering child account and address records pushed to Veeva CRM

Leveraging transformation rules, you can identify the addresses and parent affiliations that should be set to Inactive Veeva CRM. CRM functionality can then remove these Inactive records so they do not display in CRM.

Example

To ensure that Mailing type addresses are not pushed to CRM, create a NEX rule that changes the status of all mailing addresses to Inactive. Then, the CRM settings will delete those inactive addresses so they do not display on accounts in CRM.

To do this, use the following CRM settings and values:

- FILTER_INACTIVE_NETWORK_RECORDS_VOD - Set to `1`
- NETWORK_ADDRESS_DELETION_PROCESS_VOD - Set to `2`

For detailed information, see Handling Inactive Network Records in the *Veeva CRM Online Help*.

CRM considerations

Picklist values

If a transformation rule is applied to filter specific values from CRM, for example, remove mailing addresses or display the ownership hierarchy, then it is recommended to remove those values from being selected by users in CRM.

Examples

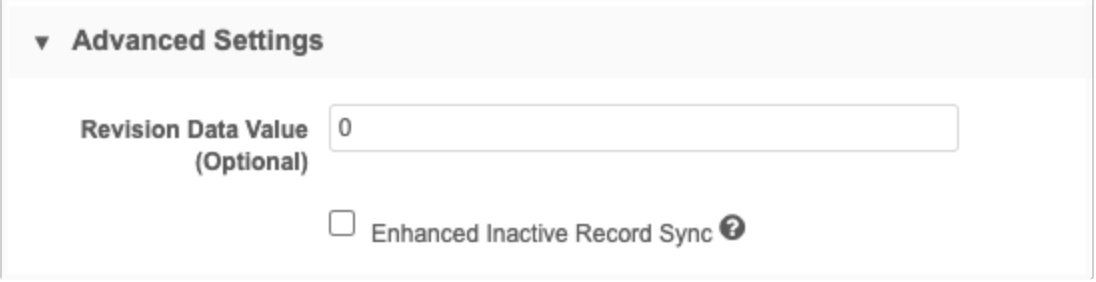
- If you are filtering out mailing addresses, ensure that Mail Only addresses cannot be selected by users in the picklists.
- If you are filtering to only display the Ownership hierarchy (Ownership, Affiliation), ensure that Claims relationship types cannot be selected.

Network Bridge updates

After transformation rules are applied to a the target subscription, run a Network Bridge job to ensure CRM is in sync.

Options

- **Update all records** - Run a full Bridge job. In the Network Bridge configuration, set the **Revision Data Value** setting to `0` to update all records.



- **Update affected records** - If you know the records that need to be updated in CRM, add the Veeva IDs (HCP/HCO) to the **Export by VID** option in the target subscription. When the next Bridge job runs, it updates those records.

  For example, if you know which HCPs records contain Mail Only addresses, add those VIDs to the **Export by VID** job so your next Bridge job removes those addresses.

Create a rule

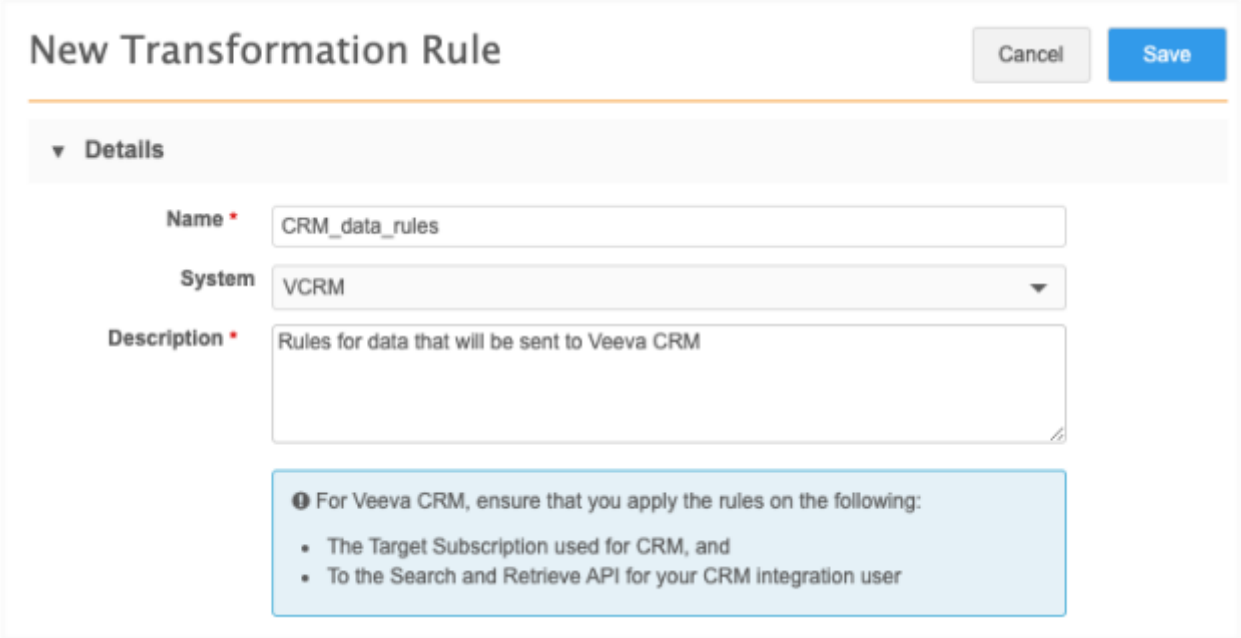
To add a transformation rule:

1. In the Admin console, click **System Interfaces > Transformation Rules**. Click **New Rule**.
2. Type a **Name** and **Description**.

   Names cannot contain spaces.
3. Select a **System**. All rules are associated to a specific source system.

   The system cannot be changed after the rule is saved.

   Tip: If you make a mistake, you can clone the rule and change the system.



Veeva CRM considerations

If this rule will be used to transform data before it is exported to CRM, ensure that the following settings are applied:

- System - Define the system for CRM
- Target subscription - Choose the subscription used for CRM.
- API - Choose your CRM integration user.

4. In the **NEX Rules** section, add the rules.

   For each NEX rule, define the following:

   - **Object** - All objects (except custom keys) are supported.
   - **Field** - The field. The list is filtered for the selected object.
   - **Code Description** - Define a meaningful description. This description will also display in associated target subscriptions.
   - **NEX Rule** - Type the Network Expression for transforming the data. The NEX function must return a value for the transformed data.

     Expand the field if you need additional space. Click **Verify** to validate the NEX syntax. A message displays at the top of the page to indicate the validation status.

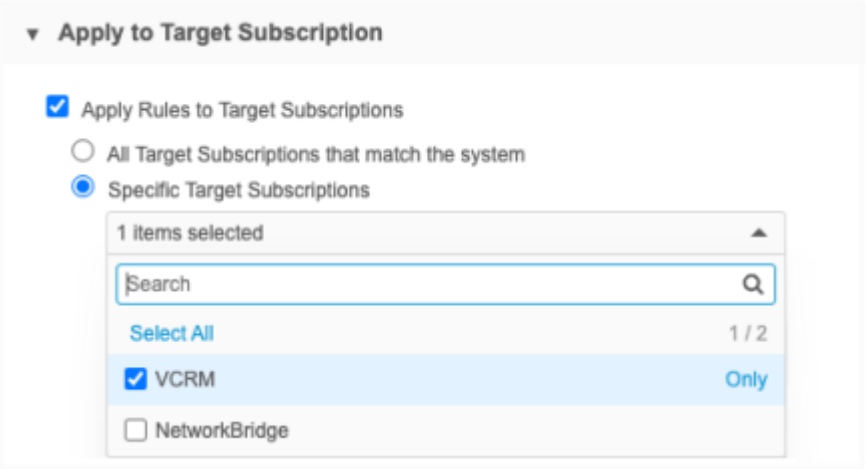
     See a list of examples in the *Example rules* section below. For help with creating network expressions, see the NEX functions and NEX operators topics in the *Veeva Network Online Help*.

   To create another rule for this system, click **Add Rule**.

Tip: Use the **Handle** icon to reorder the rules if the sequence of the rules matters. For example, you can chain together several rules to transform data for a specific field.

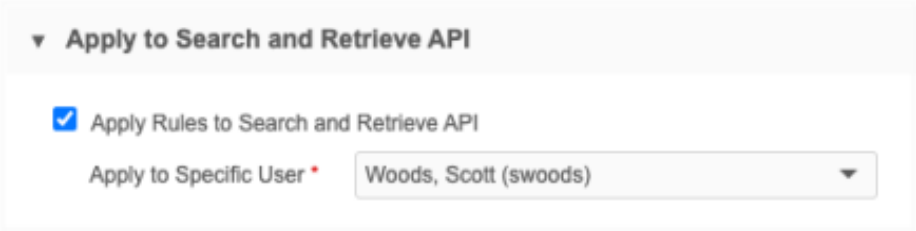
After a NEX rule is created, it can be enabled, disabled, or deleted. If a rule is not enabled, it will not be applied to the target subscription/API.

5. **Apply to Target Subscriptions** - Associate the rules to target subscriptions. Choose one of the following options:
    - **All Target Subscriptions that match the system** - Apply these rules to any target subscription that uses the source system.
    - **Specific Target Subscriptions** - Choose the specific target subscriptions that these rules apply to. All of the enabled target subscriptions that use this source system display in the list.



6. **Apply to Search and Retrieve API** - Apply the transformation rule on the API for a specific user.

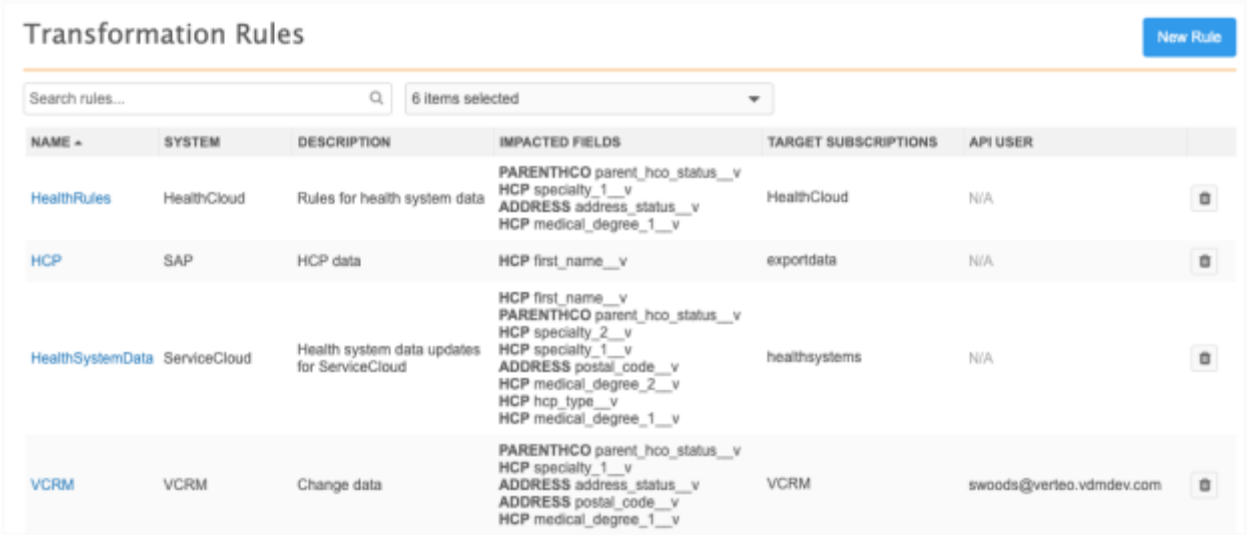
   Active users that have API access display in the list.



7. **Save** the rule.

   The NEX rule functions are validated. If there are issues, an error displays.

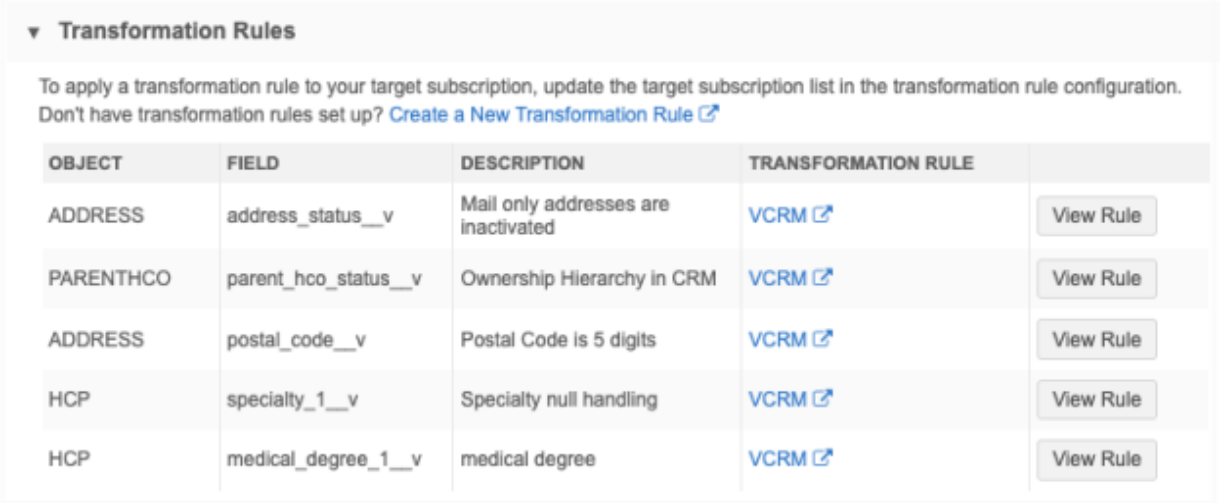
All rules in your Network instance display on the Transformation Rules page (**System Interfaces**).



Target subscriptions

On the target subscription configuration, the **Transformation Rules** section displays any rules that have been applied to the subscription.

Note: When the job runs, transformation rules are applied before transformation queries.



The following details display:

- **Object**, **Field**, **Description**
- **Transformation Rule** - Click the link to open the rule configuration.
- **View Rule** - Click to see the NEX rule. This is the current rule (from the Transformation Rules configuration page).

Target subscription job details

After the job runs, you can view the rules on the Job Details page.

The following details display in the **Transformation Rules** section:

- **Object**, **Field**, **Description**
- **NEX Rule** - The code as it existed when the job ran so you can understand the customizations that were made to the exported data if the NEX rule has since changed.
- **Transformation Rule** - The current name of the rule on the Transformation Rule configuration page. Click the name to open the configuration page. If the rule has since been deleted, the original rule name displays with a (Deleted) comment beside the name.

Job error log

If there is an issue with the NEX rule, errors will be logged. The job will fail in some cases; for example, if the NEX rule references a lookup table that does not exist in your Network instance.

Clone rules

After a transformation rule is saved, it can be cloned. This is helpful if you want to apply the same or similar rules to another source system.

When a rule is cloned, only the description and the NEX rules are copied to the new rule. You can create a name, select the source system and apply the rules to target subscriptions or the API.

Network API

Transformation rules can be applied to Retrieve and Search API requests.

Search API - Supported requests

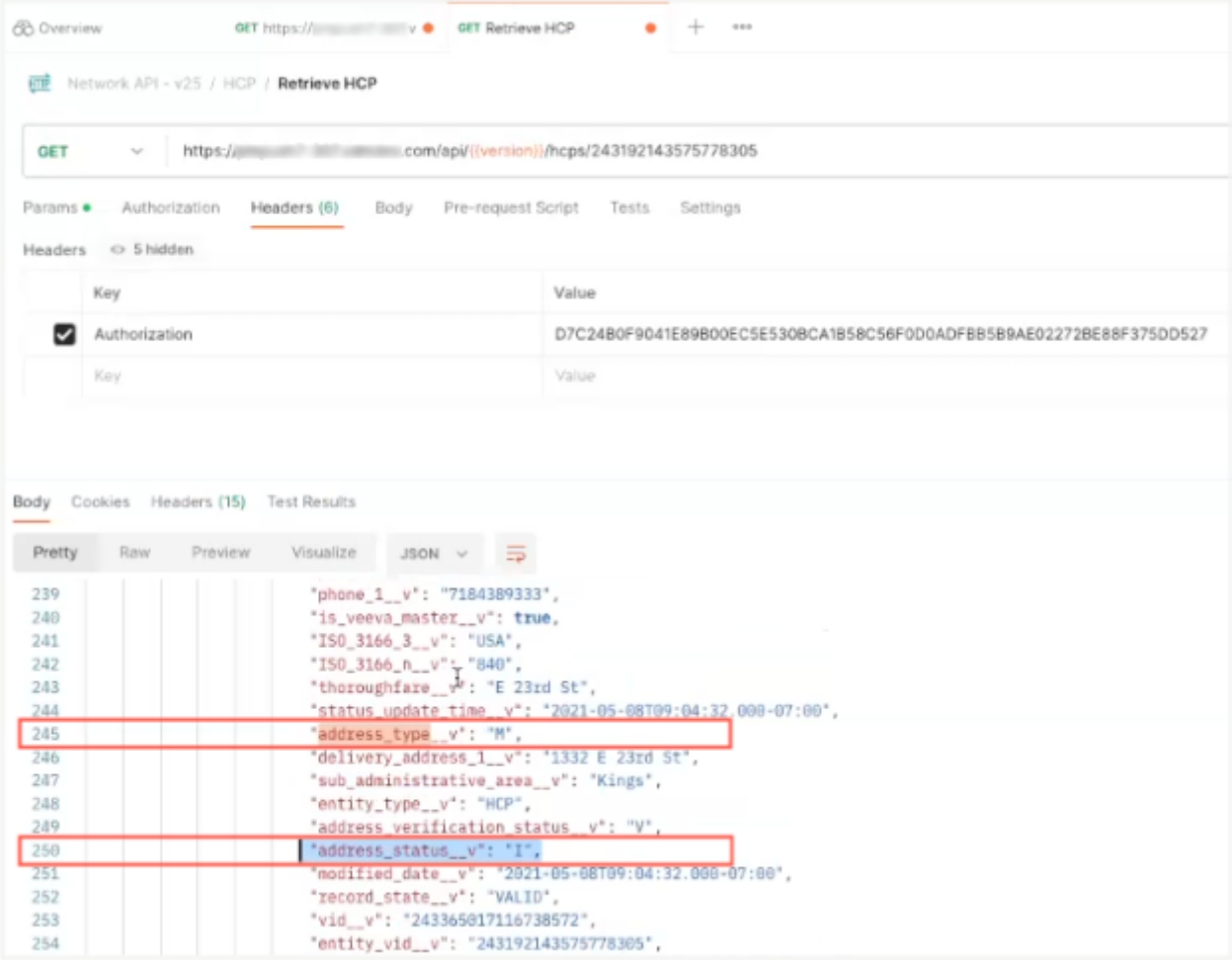
- Search API
- Search API + supplemental

Retrieve API - Supported requests

- Retrieve Entity
- Retrieve Child Entity
- Batch Retrieve Entity
- Batch Retrieve Child Entity
- Retrieve HCO
- Retrieve HCP
- Retrieve Change Request (IncludeEntity = True)
- Batch Retrieve Change Request (IncludeEntity = True)

Example request

During this Retrieve HCP request, the transformation rule sets the Mail Only address record status to Inactive.



Support for fields with null values

In the Network API, if a field is empty, the record is not returned in the JSON. For transformation rules, the field is returned in the response entity JSON because the transformation rule adds a value.

Example

The `specialty_1__v` field value is NULL, so it is not returned in the Retrieve HCP response. A transformation rule runs that changes the NULL value to "UNSPECIFIED__c". Now, the Specialty field will be returned in the JSON: "specialty_1__v":"UNSPECIFIED__c".

Hashtag considerations

Hashtags reflect the untransformed values. They do not change for transformed values.

If you search using hashtags, the results returned are based on the original value of the data.

Example transformation rules

Object	Description	Field	NEX Rule
Address	Deactivate mailing addresses	address_status__v	`if(addres_type__v =='M', 'I',address_status__v)`
	Limit US zip codes to 5 digits	postal_code__v	`if(country__v == 'US', left(postal_code__v,5),postal_code__v)`
ParentHCO	Deactivate relationship types that are not Ownership or Affiliation	parenthco_status__v	`if(not(relationship_type__v in ['2','7356','12']), 'I', parent_hco_status__v)`
HCP	Override retired HCP status	hcp_status__v	`if(hcp_status__v == 'R' && status_override__c == 'Y', 'A', hcp_status__v)`
	When Specialty is null, then set it to a custom reference type (UNSPECIFIED__c)	specialty_1__v	`if(ISNULL(specialty_1__v), 'UNSPECIFIED__c', specialty_1__v)`
	When Medical Degree is null, then set it to a custom reference type (UNSPECIFIED__c)	medical_degree_1__v	`if(ISNULL(medical_degree_1__v), 'UNSPECIFIED__c', medical_degree_1__v)`
	Set record type Note that this uses a lookup table.	record_type__v	`LOOKUP('type_mapping__t', 'record_type', hcp_type: hcp_type__v)`

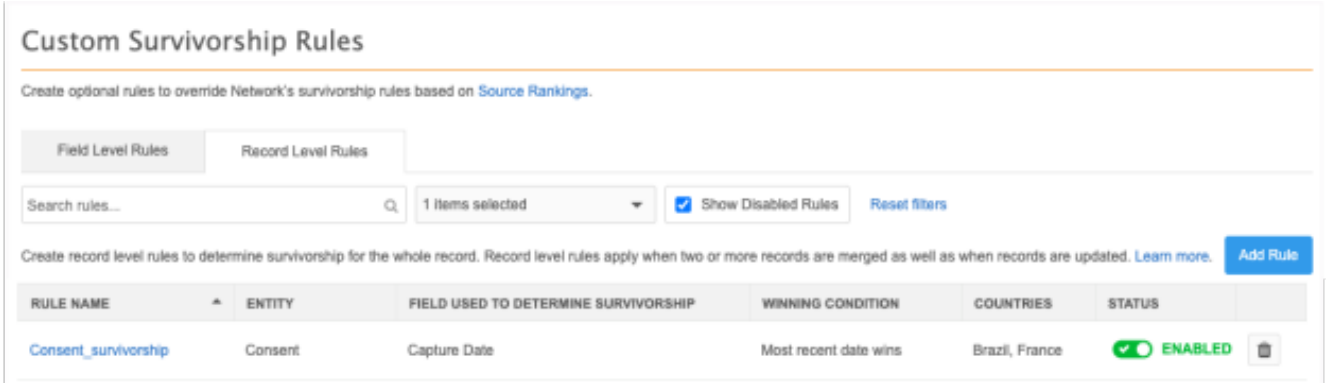
Note: For rules that change an object status to inactive (for example, addresses or relationships), if it is pushed to Veeva CRM, some CRM settings will then remove those records. For more information, see the *Veeva CRM integrations* section above.

Survivorship

TIME-BASED SURVIVORSHIP RULES

Administrators can define custom rules that determine survivorship based on the most recent update. This can simplify survivorship when the most recent information is the most valuable and should win; for example, when you are tracking HCP channel consent.

These record-level rules are applied to data updates (from source subscriptions only) and merges. They override all source rankings and merge rules that are defined in your Network instance.



This feature is on by default. Administrators and Data Managers can create rules for your Network instance.

Supported objects

Record level rules can be applied to all sub-objects and relationship objects that are enabled in your Network instance.

Main objects and custom keys are not supported.

Requirements

Rules are based on custom date or date/time fields. The dates are compared between records to determine survivorship.

- Sub-objects and relationships objects must have a custom field of this type so that record level rules can be created.
- Sources must use these custom fields to specify when the record was last updated.

About survivorship

Survivorship in Network is typically applied using the following methods:

- **Source rankings** - Survivorship is determined by the highest ranked source. Source systems can be ranked at the instance level, object level, and field level.
- **Merge instructions** - Survivorship is determined by one of the following instructions:
  - **Exclusive merge** - Field values are retained from the winning record unless the winner has no value.
  - **Inclusive merge** - Source rankings are used to determine which value should win.

  Used when merges occur through suspect match tasks, Find Suspect Match and in data updates or bulk merge using source subscriptions.

- **Custom merge rules** - Custom field-level rules that override regular survivorship rules. They are applied only to merges; they are not applied during data load.

When these time-based record level rules are applied, they override all of these survivorship methods listed above.

Supported actions

Record level survivorship rules are applied when records are updated using source subscriptions or when records are merged.

- **Updating records** - The date/time is compared between the incoming data and the existing record in Network.

Survivorship scenario	Update Accepted or Dropped
Data in the source feed is more recent than what Network currently has	Accepted
Data in Network is more recent than the data in the source feed	Dropped
Data in the source feed and the current date are the same	Accepted

  Rules do not apply to updates from change requests or to edits made directly on the Profile page.

- **Merging records** - When two records (Veeva IDs) are merged, the data with the most recent date survives the merge.

  Example: If the data with the most recent date/time is on the losing record, it is added to the winning record.

Note: Record level survivorship is typically based on the recentness of the data, but there is an option to choose the oldest date as the winner.

Create a time-based survivorship rule

Add a rule to determine survivorship based on the date or time of the field value.

Prerequisite

Record level rules support the following types of custom fields: **Date (no time)** and **Date and time**. Ensure that the sub-object or relationship object has a custom field of this type so you can create the rule.

Add the rule

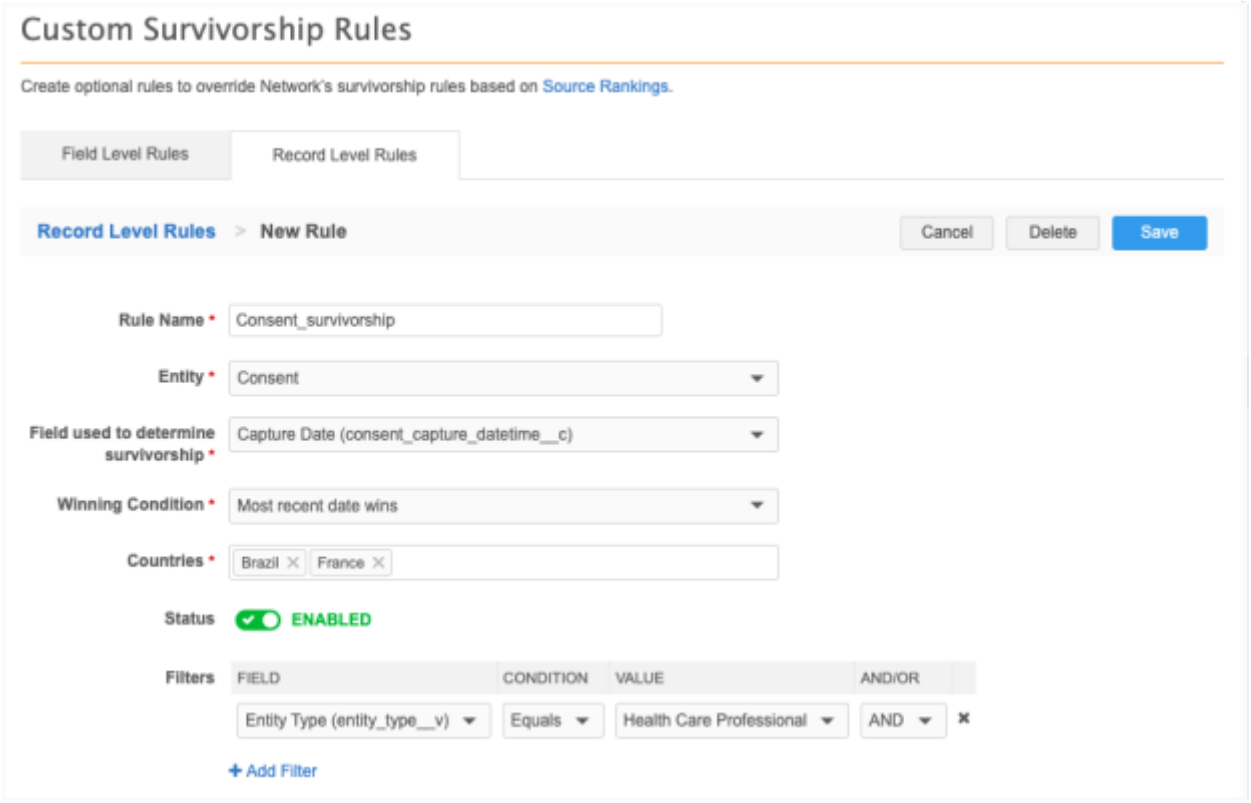
1. In the Admin console, click **Data Model > Custom Survivorship Rules**.

   Previously, this tab was called **Custom Merge Rules**.

2. Select the **Record Level Rules** tab.

   The **Field Level Rules** tab contains any custom merge rules that were previously created in your Network instance.

3. Click **Add Rule** and define the details for the rule.



4. **Rule Name** - Define a name.

   Supported characters: alphanumeric (a-z, A-Z, 0-9), hyphens (-), and underscores (_). The name cannot contain spaces.

5. **Entity** - Choose the sub-objects or relationship object that the rule applies to.

6. **Field used to determine survivorship** - Select the custom field that will be used to determine survivorship. Only custom date and date/time fields display.
7. **Winning Condition** - Choose either **Most recent date wins** or **Oldest date wins** .
8. **Countries**- Select the applicable countries.
9. **Status** - The rule is on by default.
10. **Filters** - Add filters to the rule if you want it to apply to selected records only.
    - **Field** - Choose the field for this filter. The fields in the drop-down list apply to the object selected for this rule.
    - **Condition** Choose one of the available conditions.
    - **Value** - If the **Condition** that you chose requires a value, type the value. The value might be free text (for example, type a name) or a list of options.
    - **And/Or** - Choose the operator if there is more than one filter.
11. **Save** your changes.

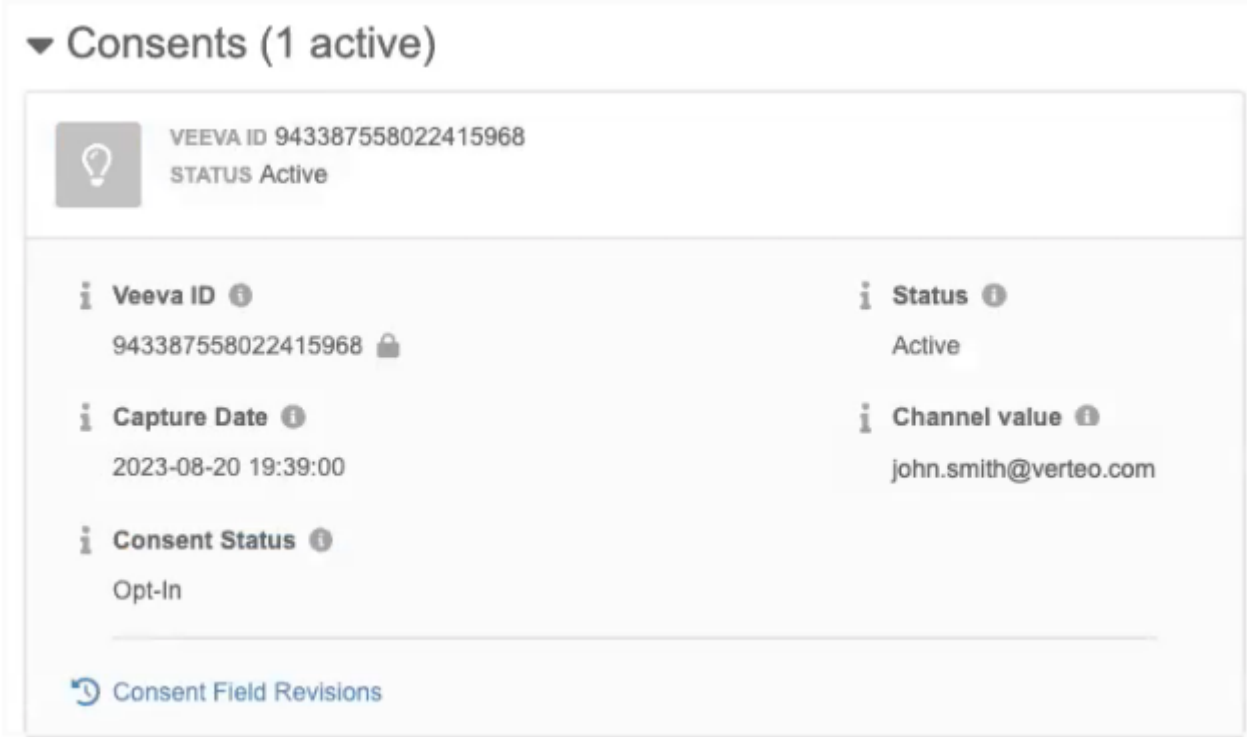
The rule will run when records that include that sub-object are updated or merged.

Rules can be edited or deleted.

Example - Updating data

In this example, a Consent custom sub-object stores HCP channel consent. A source subscription is loading updates for the Consent object for three HCP records. When records are updated, we want the most recent consent information to survive.

Example consent record



Records in Network - Current consent information for the HCPs

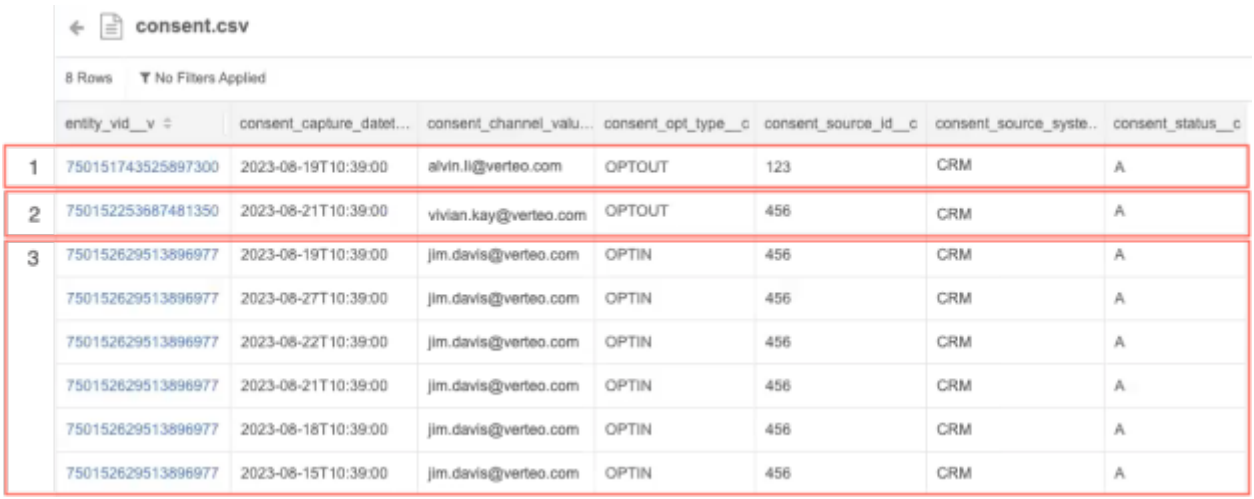
These records were originally loaded to Network using the Salesforce Marketing Cloud (SFMC) source system. Note that Record 3 currently has no Consent record.

Record	HCP Email	Consent Capture Date	Consent Status	Source System
1	alvin.li@verteo.com	2023-08-20 19:39:00	Opt-In	SFMC
2	vivian.kay@verteo.com	2023-08-20 19:39:00	Opt-In	SFMC
3	jim.davis@verteo.com	NULL	NULL	NULL

Source file - Updates to consent information of the HCP

A source file from Veeva CRM contains updates to the Consent status for Records 1 and 2 and contains new consent records for Record 3.

Note: In this example, the SFMC system is ranked higher than the CRM system.


consent.csv

8 Rows | No Filters Applied

	entity_vid__v	consent_capture_datet...	consent_channel_valu...	consent_opt_type__c	consent_source_id__c	consent_source_syste..	consent_status__c
1	750151743525897300	2023-08-19T10:39:00	alvin.li@verteo.com	OPTOUT	123	CRM	A
2	750152253687481350	2023-08-21T10:39:00	vivian.kay@verteo.com	OPTOUT	456	CRM	A
3	750152629513896977	2023-08-19T10:39:00	jim.davis@verteo.com	OPTIN	456	CRM	A
	750152629513896977	2023-08-27T10:39:00	jim.davis@verteo.com	OPTIN	456	CRM	A
	750152629513896977	2023-08-22T10:39:00	jim.davis@verteo.com	OPTIN	456	CRM	A
	750152629513896977	2023-08-21T10:39:00	jim.davis@verteo.com	OPTIN	456	CRM	A
	750152629513896977	2023-08-18T10:39:00	jim.davis@verteo.com	OPTIN	456	CRM	A
	750152629513896977	2023-08-15T10:39:00	jim.davis@verteo.com	OPTIN	456	CRM	A

Results - Updates to the consent information of the HCPs

Record-level survivorship rules determine the Consent Status value based on the most recent Consent Capture Date. The source rankings are overruled.

Record	HCP Email	Consent Capture Date	Consent Status	Source System
1	alvin.li@verteo.com	2023-08-20 19:39:00	Opt-In	SFMC
2	vivian.kay@verteo.com	2023-08-21 10:39:00	Opt-Out	CRM
3	jim.davis@verteo.com	2023-08-27 10:39:00	Opt-In	CRM

- **Record 1 - Update dropped**: The incoming consent capture date was older than the current consent capture date.
- **Record 2 - Update accepted**: The incoming consent capture date was more recent than the current consent capture date.
- **Record 3 - Record created**: The consent data from the source feed is used to create the consent record on the HCP.

  The source feed contained multiple consent records with the same email address. This can be very typical for data extracted from Veeva CRM when the consent status is updated several times creating multiple records in the Veeva CRM database.

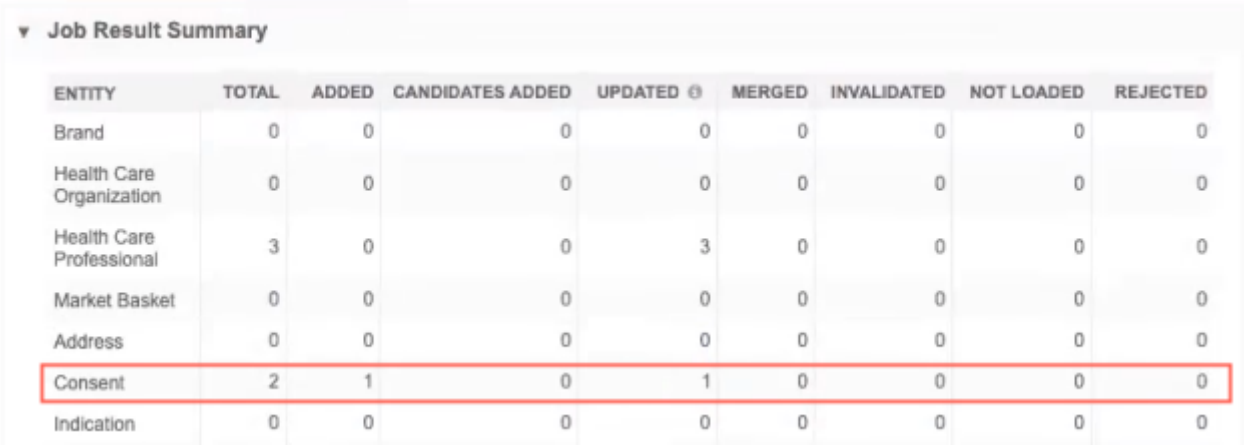
  These records were deduplicated during data load; the **Source Dedup** setting was on in the subscription for the Consent object.

Job details

You can see the outcome of the updates in the **Job Summary Section**.

In this example, one record was added (Record 3) and one record was updated (Record 2).

Note: The consent update that was dropped for Record 1 is not recorded. Updates that are dropped for record level survivorship are not logged in the job details.



Job Result Summary

ENTITY	TOTAL	ADDED	CANDIDATES ADDED	UPDATED	MERGED	INVALIDATED	NOT LOADED	REJECTED
Brand	0	0	0	0	0	0	0	0
Health Care Organization	0	0	0	0	0	0	0	0
Health Care Professional	3	0	0	3	0	0	0	0
Market Basket	0	0	0	0	0	0	0	0
Address	0	0	0	0	0	0	0	0
Consent	2	1	0	1	0	0	0	0
Indication	0	0	0	0	0	0	0	0

Data loading considerations

Timezones

It is highly recommended to specify the timezone when loading data.

Record level survivorship rules always compare the incoming date/time with the current date/time on the record to determine whether to accept or drop the incoming update.

In the Network database, all date timestamps are stored in Coordinated Universal Time (UTC). For example: `2023-03-25T19:30:00Z`. This can be different from the timezone that displays on record profiles. On the Profile page, the date/time reflects the time zone of your local machine.

Example: If your local computer is set to Central European Summer Time (UTC+02:00) and data is loaded at 2023-03-25T19:30:00Z (UTC), the date and time that displays on the profile page is 2023-03-25T21:30:00Z (UTC+02:00).

Examples

Review these examples to understand if the incoming update will be accepted or rejected based on the defined timezone.

Current Date/Time in Network	Incoming Date/Time	Result	Details
2023-03-25T-19:30:00Z (UTC)	**2023-03-25T-19:15:00-04:00** EDT (UTC -04:00)	Accepted	**2023-03-25T-19:15:00** EDT equals **2023-03-25T-23:15:00** UTC. The incoming date/time is more recent than the current date/time.
2023-03-25T-19:30:00Z (UTC)	**2023-03-25T-21:15:00+02:00** CEST (UTC +02:00)	Dropped	**2023-03-25T-21:15:00** CEST equals **2023-03-25T-19:15:00** UTC. The incoming date/time is less recent than the current date/time.
2023-03-25T-19:30:00Z (UTC)	**2023-03-25T-19:35:00**	Accepted	No timezone was specified in the incoming update, so UTC is the default. The incoming and current date/time are in the same timezone and the incoming date/time is more recent than the current date/time.

Note: The timezone offset from UTC time must be specified in the source feed to indicate the different timezone. For example, if the data reflects Central European Summer Time (UTC+02:00), the date in the source file must include **+02:00**.

Source dedupe

Always enable the **Source Dedupe** setting for the object to ensure that duplicate records are not created.

In most cases, the record level survivorship rules will ensure that only one record survives. However, if an initial load contains duplicates in the source feed, duplicate records will be created if the **Source Dedupe** setting is not on.

Null values

Record level survivorship rules are based on date fields. If the date field in Network or an incoming data feed is empty or null, survivorship is handled using the following behavior:

Date in Feed	Date in Network	Result	Details
2023-03-04T-09:00:00Z	2023-03-04T-09:00:00Z	Accepted	The incoming and existing date/time is the same.
2023-03-04T-09:00:00Z	empty/null	Accepted	The incoming update has a date/time so it wins survivorship over the value in Network that has no date/time.
empty/null	2023-03-04T-09:00:00Z	Dropped	The incoming record did not specify a date/time so the date/time defined in Network wins survivorship.
empty/null	empty/null	Accepted	The incoming record wins survivorship because neither record has a date/time defined.

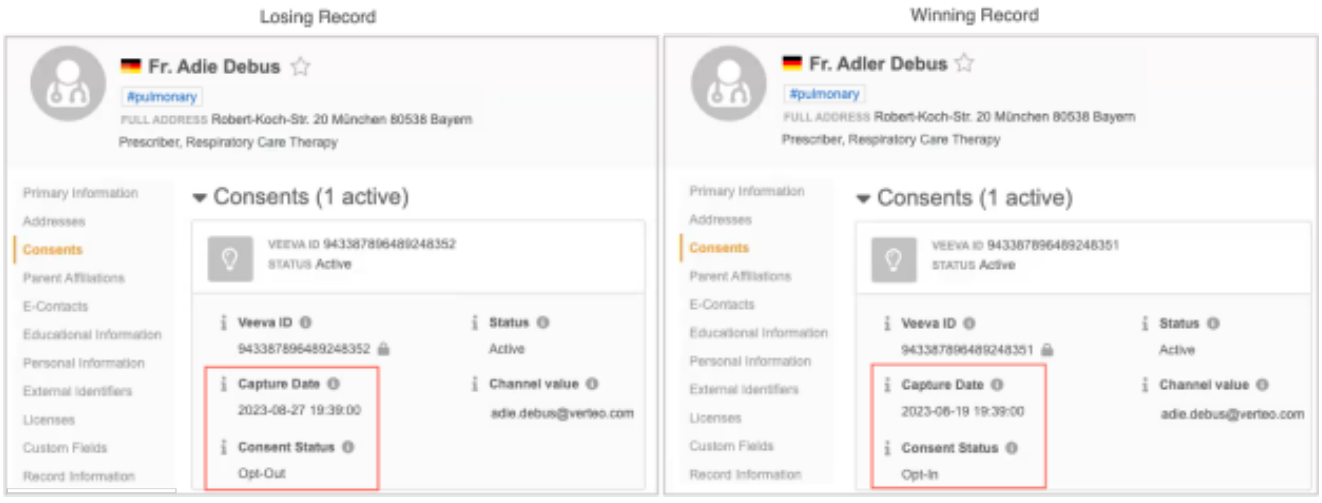
Example - Merging records

Records can be merged in several ways; for example, using Find Suspect Match, through the Data Updater, and bulk merge using Source subscriptions. Merges can also occur when records are merged by OpenData. When merges performed by OpenData are pushed down to your Network instance, record-level survivorship rules apply only to data managed in custom objects.

In this example, two locally managed HCP records will be merged using the Data Updater feature. The HCP records include consent data. A record level survivorship rule is applied to the Consent object to ensure that the latest consent capture date wins survivorship.

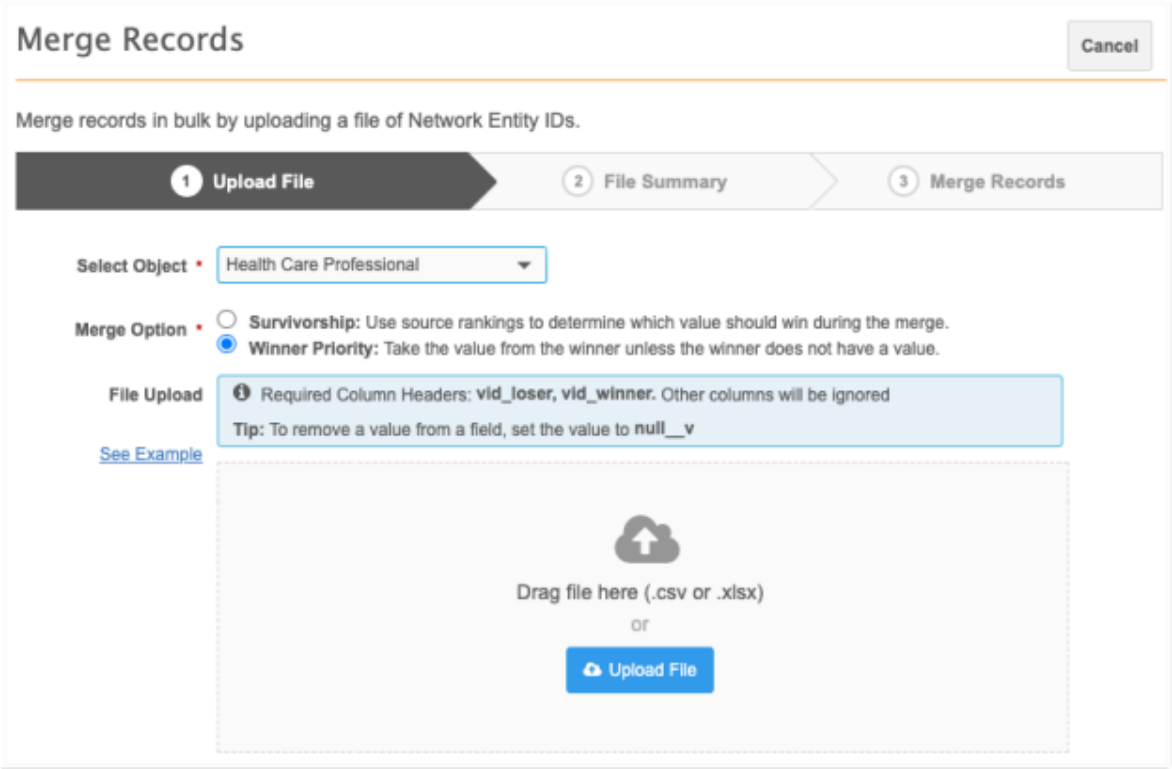
Records being merged

The losing record has the most recent consent Capture Date, so that consent information should survive on the winning record.



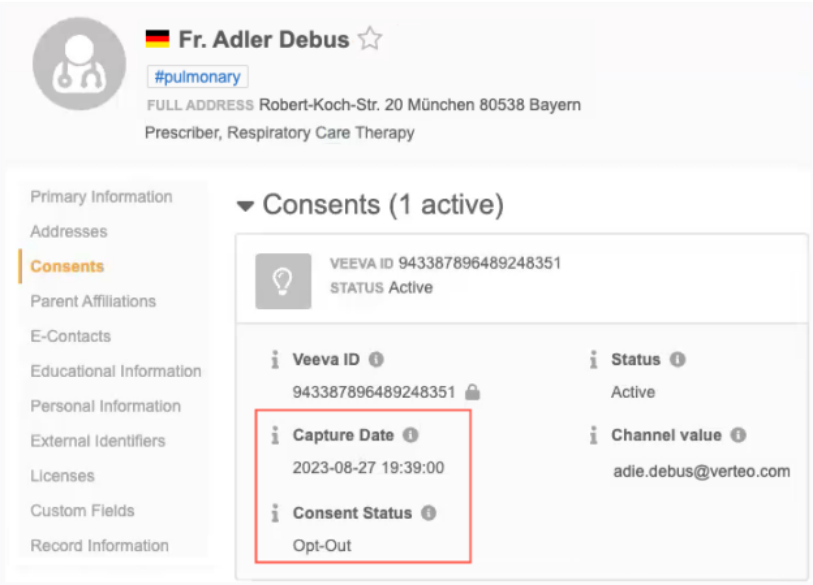
Data Updater job

A **Merge Option** must be defined for each job. In this example, we'll select **Winner Priority** instruction; this means all values from the winning HCP record will win during the merge unless the winner does not have a value. However, for the Consent object, the record level survivorship rule will override the **Winner Priority** instruction to ensure that the most recent consent information survives.



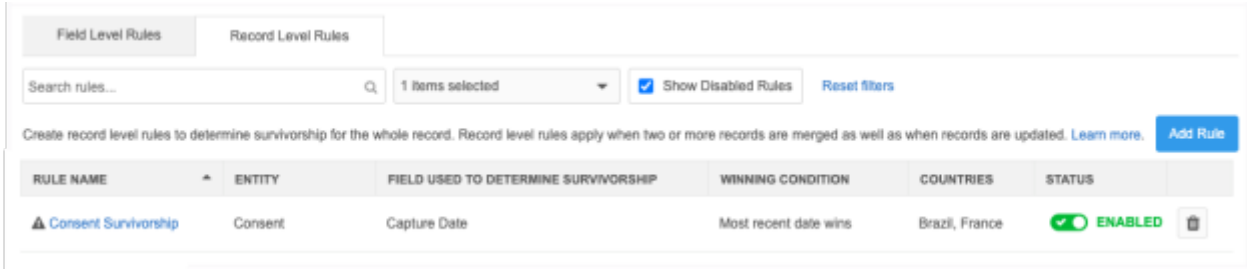
Results

When the merge completes, the most recent consent data from the merge loser is moved to the winning record.



Inactive fields

When a field that is used in a record level survivorship rule is deactivated, an **Alert** ⚠ icon displays beside the rule name on the Record Level Rules tab.



On the rule configuration, the field is highlighted and the message "This field is inactive" displays.

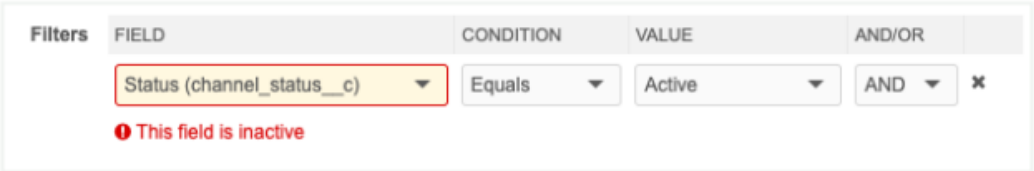
Rule behavior

The rule will not be applied when the field is inactive. Survivorship is determined by the usual methods.

Filter behavior

If a field that is used in a rule filter is deactivated, the rule will be applied but the filter condition will not apply.

For example, when a rule filter specifies that the `channel_status__c` field value must be Active but that field is deactivated, the rule will apply to all Consent records regardless of their value for the `channel_status__c` field.



Data maintenance subscriptions

JOB VALIDATION RULES

Job validation rules can now be applied to selected data maintenance subscriptions to ensure that critical updates occur as expected. Use the rules to check if a job is unintentionally inactivating or deleting a lot of records. This is particularly helpful if you have automated the process for unsubscribing or soft-deleting records.

Validation rules can be applied to the following data maintenance subscriptions:

- Delete Locally Managed HCP/HCO
- Delete Custom Object Records
- Unsubscribe OpenData Records

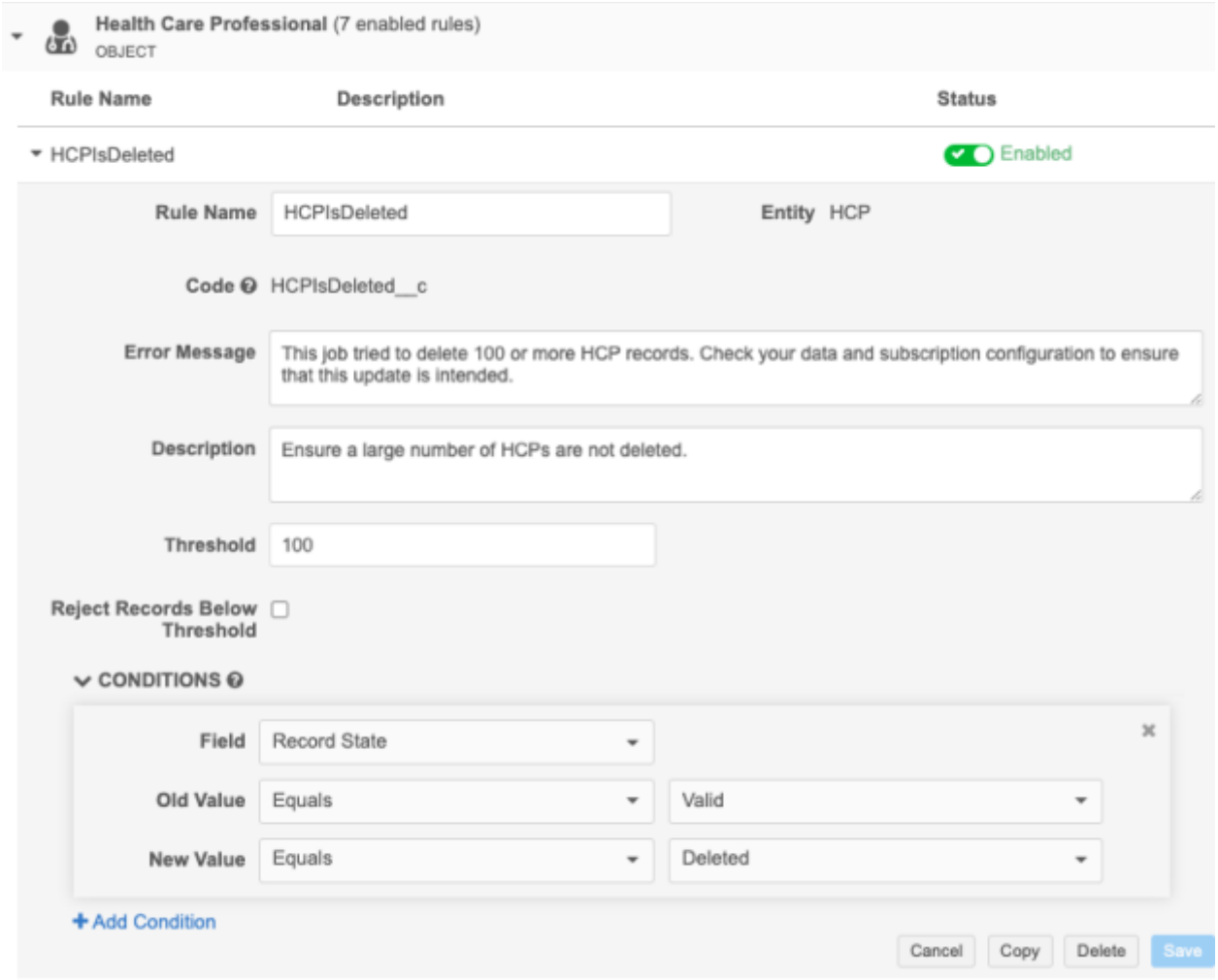
> **Tip:** Unsubscribe from Third Party Records jobs already support job validation rules because these jobs are run using NEX rules in source subscriptions.

This enhancement is on by default in your Network instance.

Job validation rules

A set of default validation rules are provided for Veeva standard objects (**System Interfaces > Job Validation Rules**). These rules are on by default.

For data maintenance subscriptions, the applicable default job validation rules are the rules that check for records being deleted. For example, if job validation rules are applied, the **HCPIsDeleted** rule will fail a **Delete Locally Managed HCP/HCO** job if the job tries to delete 100 or more HCP records.



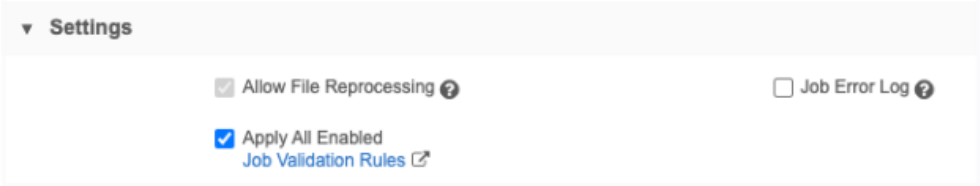
Create rules

Rules can be added for Veeva objects and for custom objects that are enabled in your Network instance.

For details, see Job validation rules in the *Veeva Network Online Help*.

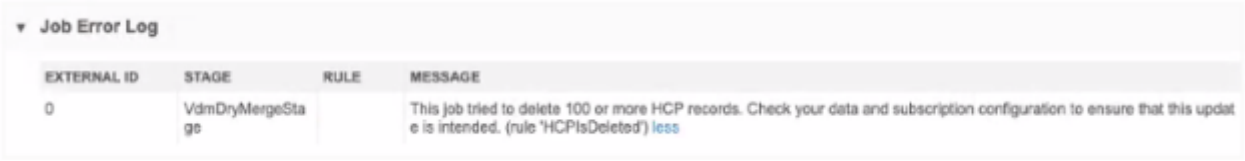
Apply job validation rules

The supported data maintenance subscriptions contain a new setting called **Apply All Enabled Data Validation Rules**. Select the setting to apply job validation rules to the job.



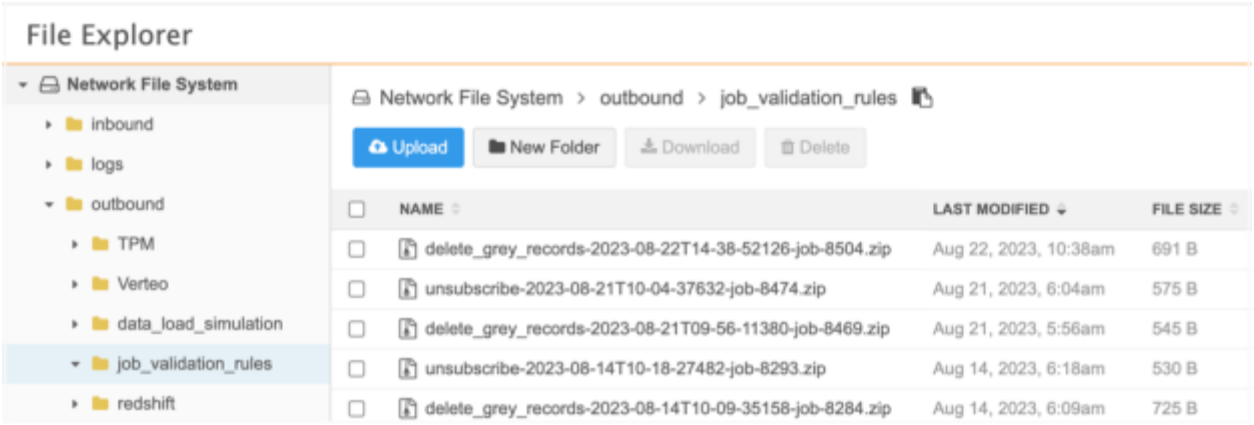
Failed rules

If a job validation rule threshold is met, the data maintenance job fails and no changes are made to the data. Open the Job Details page to see an error and the rule name in the **Job Error Log**.



Logs

When job validation rules are applied to data maintenance subscriptions, log files are created. View the log files in the **outbound > job_validation_rules** directory in File Explorer. A .zip file is created for each job.

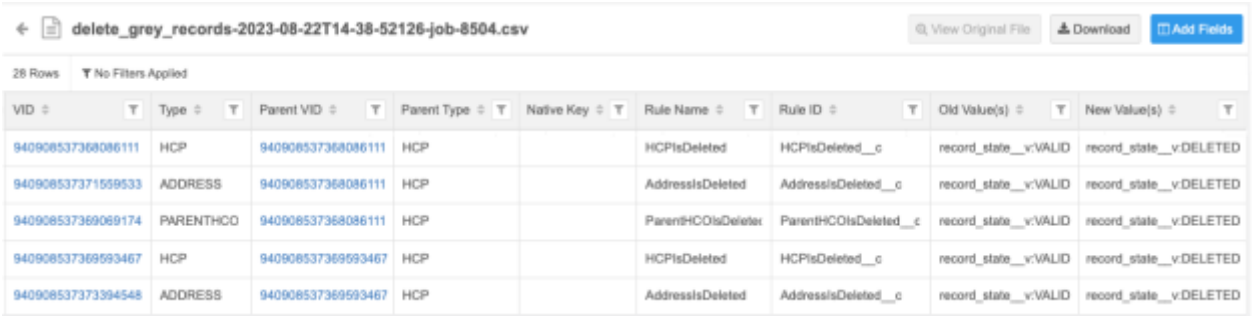


Open the .zip file to review the .csv file for the job.

The .zip file and .csv file have the following naming convention: <subscription_name>-<timestamp>-job-<job ID>.

Example job log

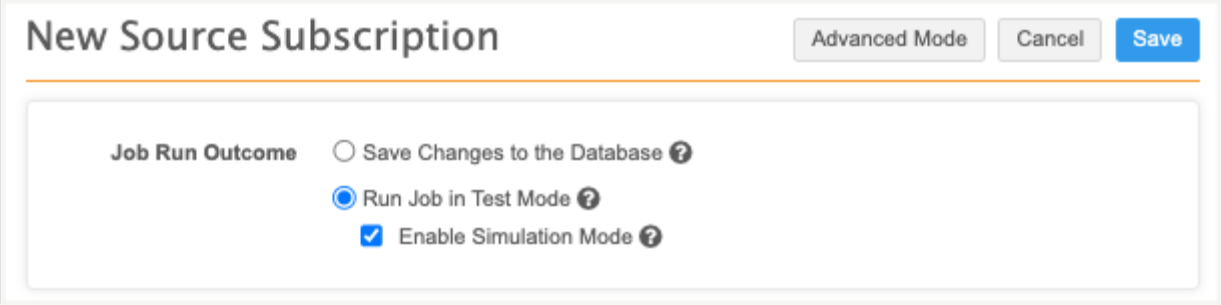
In the file, you can review the changes that the job tried to make and the rule that was used to detect the change.



Source subscriptions

SIMULATING DATA UPDATES

Administrators and Data Managers can now test data updates in their Production instance. Source subscription jobs can be run in simulation mode so you can preview the outcome before committing the data to the database. This ensures that you have more control and confidence over the quality of your data operations.

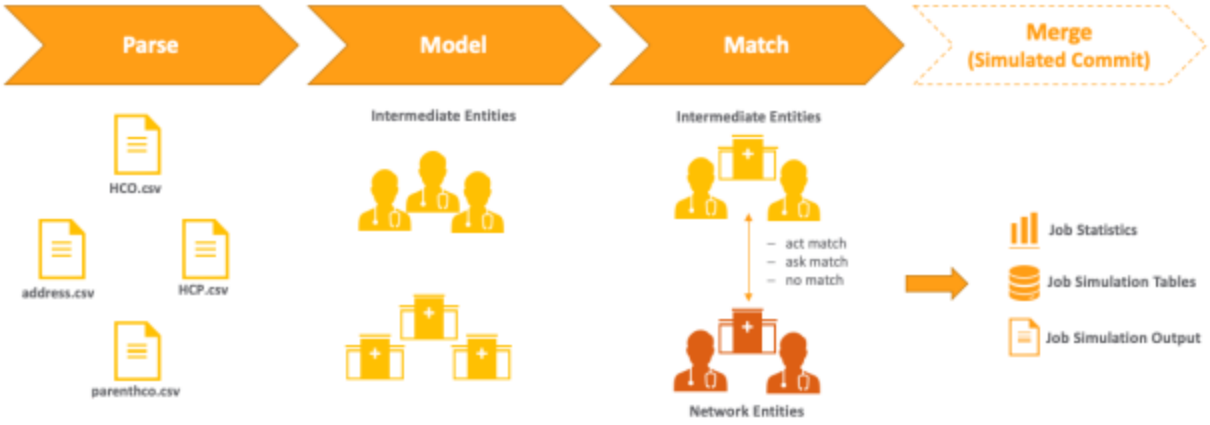


This feature is enabled by default in your Network instance.

About simulation mode

Previously, data loading jobs could be tested if the **Apply Updates & Merges** setting was not enabled for a subscription. The job stopped after the match stage, so the data wasn't committed in the merge stage (the last stage of the job). Some statistics were available on the Job Details to provide an idea of what the job would do.

Now, when you run a job in simulation mode, the data is simulated during the merge stage. You can preview the results without committing any data to the database.



Simulated data

Examples of outcomes that you can test with simulated data:

- **Changes to critical fields** - For example, address status, relationship type, HCO/HCP type, and so on.
- **Rejected records** - Learn about records that will be dropped from the feed because of data issues. Records that are rejected are not included in the simulation output, but an error displays in the **Job Error Log** so you can investigate the issue.
  For example, if a reference code is not mapped to Network, a message displays in the log so you can fix the issue. Job errors for rejected records do not display in **Test Mode**.
- **Network expression rules** - Test NEX rules in the simulated data to ensure the outcome is expected before committing it to the database.
  All rule points are supported for simulation mode.
- **Network Address Inheritance** - Review addresses that are synced from Parent HCOs in the output.
- **Repointing relationships**
- **Bulk merging entities**

Reviewing simulated data

Use simulation mode to preview the updates before committing the data to the database.

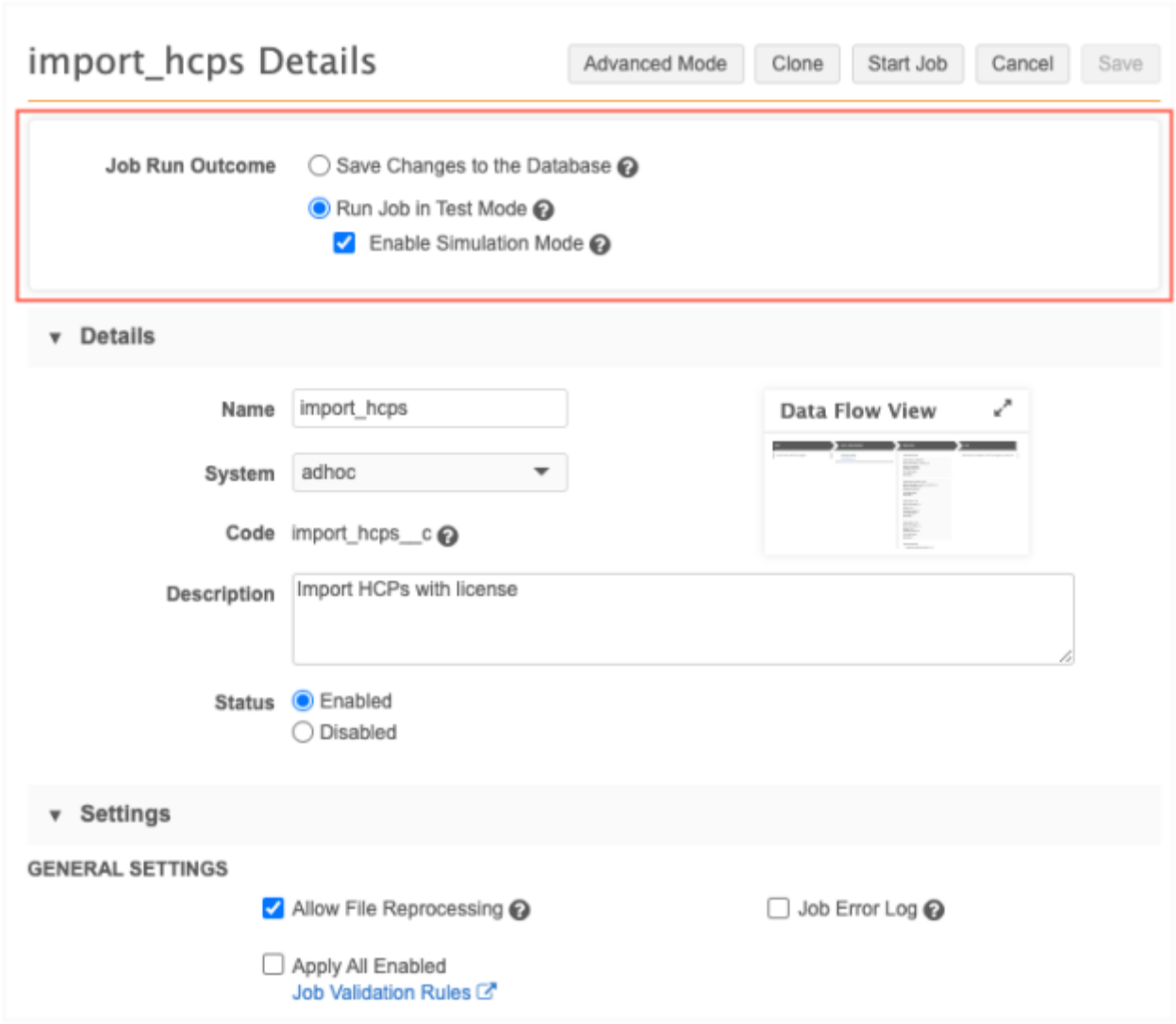
Simulated data can be reviewed in the following areas:

- **Job statistics** - Counts for added and updated records display in the **Job Summary Report** section on the Job Details page.
  When the job is run in **Test Mode**, statistics are not available for the **Job Summary Report**.
- **Files** - A .zip file is created in the Network file system. It contains a .csv file for each object impacted by the data. For example, if the feed contains updates for HCPs, a .csv file is created for the HCP object and for each of its sub-objects and relationships. The files for all simulation jobs are archived in the FTP folder.
- **Reporting tables** - The simulation output is available as reporting tables. Use the tables to compare the simulated data with production data. Only the tables for the last run simulation job are available. Each time a simulation job starts, the previous simulation tables are replaced.

If you identify issues, you can fix them and prevent unintended data updates from being loaded into your Network instance. When the job outcome is working as expected, run the job again using the **Save Changes to the Database** option to commit the changes to the production database.

Running a job in simulation mode

Source subscription configurations now include a section called **Job Run Outcome** at the top of the page.



You can run a job using any of the three options:

- **Save Changes to the Database** - Commit the data updates to the database.
  This option replaces the **Apply Updates & Merge** option that was previously in the **General Settings** section.
- **Run Job in Test Mode** (Default) - Stop the job after the matching stage so no changes will be applied to the database. You can review the job statistics on the Job Details page.

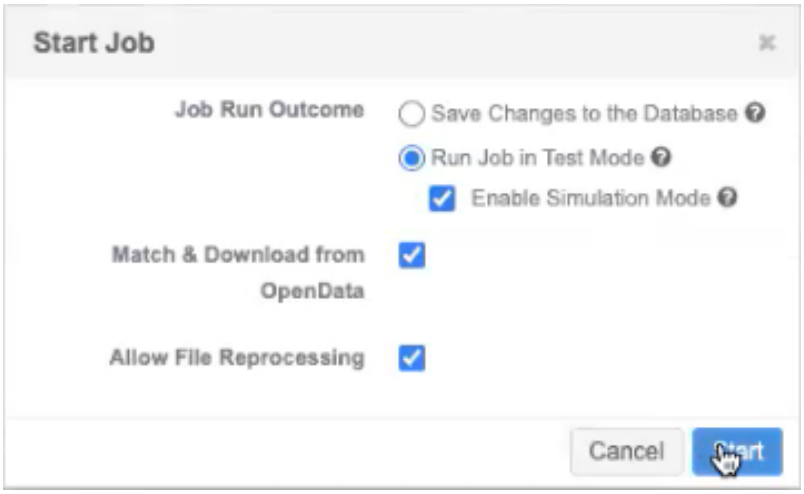
  This behavior is the same as when the **Apply Updates & Merge** option was not selected.

  - **Enable Simulation Mode** - Preview the changes to the production data. No changes are committed to the database.

Run a job

To ensure that the source subscription job does not create any data issues, test it in simulation mode. The data is loaded to a restricted area where you can preview the results using reporting.

1. Open a source subscription (**System Interfaces > Source Subscriptions**).
2. In the **Job Run Outcome** section, choose **Run Job in Test Mode** and **Enable Simulation Mode**.
3. Click **Start Job**.
4. In the **Start Job** confirmation dialog, the **Job Run Outcome** options display again in case you want to change them.



5. Click **Start**.

Tip: If you are running the job using the API, you can use the `mode` parameter to specify the job outcome. The parameter value will override the configuration outcome only for the job started by the API call.

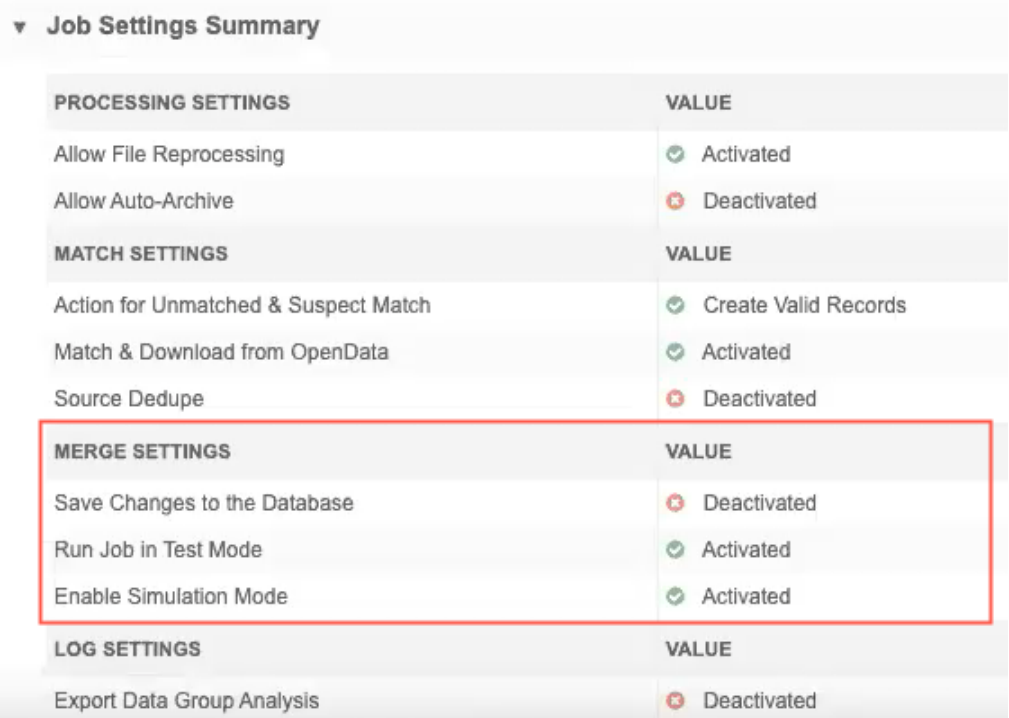
Job Details

After the job completes, review the statistics on the Job Details page and the simulated data output.

The following sections on the page are updated to support simulation mode.

Job Settings Summary

The **Merge Settings** heading identifies the **Job Run Outcome** setting values that were applied to the job.



Job Settings Summary

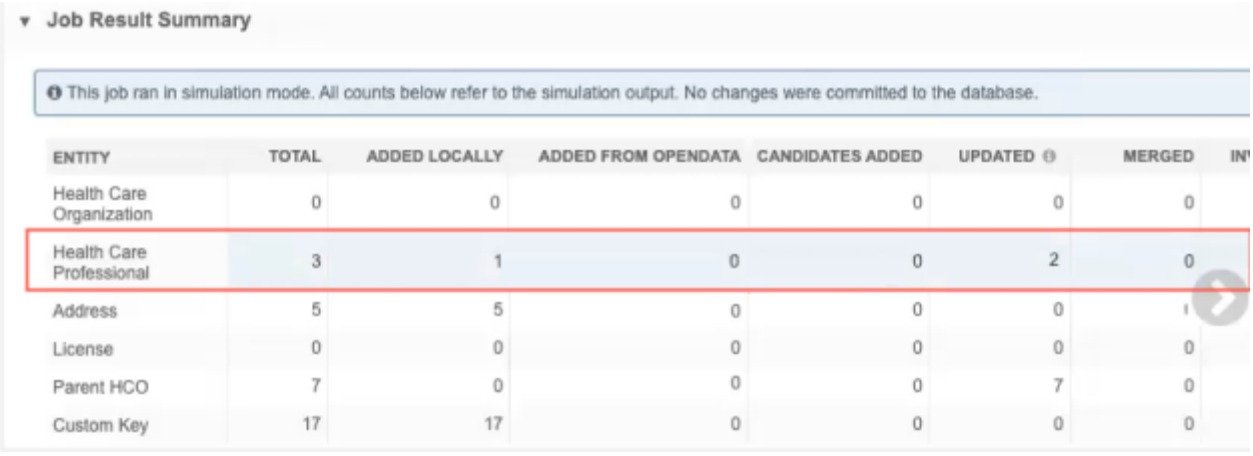
PROCESSING SETTINGS	VALUE
Allow File Reprocessing	Activated
Allow Auto-Archive	Deactivated
MATCH SETTINGS	**VALUE**
Action for Unmatched & Suspect Match	Create Valid Records
Match & Download from OpenData	Activated
Source Dedupe	Deactivated
MERGE SETTINGS	**VALUE**
Save Changes to the Database	Deactivated
Run Job in Test Mode	Activated
Enable Simulation Mode	Activated
LOG SETTINGS	**VALUE**
Export Data Group Analysis	Deactivated

Job Result Summary

This section is populated with the simulated statistics of the job after the merge stage. A banner displays to assure you that the data is simulated and that no changes were committed to the database.

Example

In this job, if the data had been committed to the database, one HCP would have been added as a local record, and two HCPs would have been updated.



Job Result Summary

This job ran in simulation mode. All counts below refer to the simulation output. No changes were committed to the database.

ENTITY	TOTAL	ADDED LOCALLY	ADDED FROM OPENDATA	CANDIDATES ADDED	UPDATED	MERGED	IN
Health Care Organization	0	0	0	0	0	0	
Health Care Professional	3	1	0	0	2	0	
Address	5	5	0	0	0		
License	0	0	0	0	0	0	
Parent HCO	7	0	0	0	7	0	
Custom Key	17	17	0	0	0	0	

Note: When a job is set to **Test Mode**, it stops the job after the match stage so, job statistics display only in the **Data Load Summary**, **Processed Data Summary**, and **Match Summary**. The **Job Result Summary** section is not updated on the Job Details page.

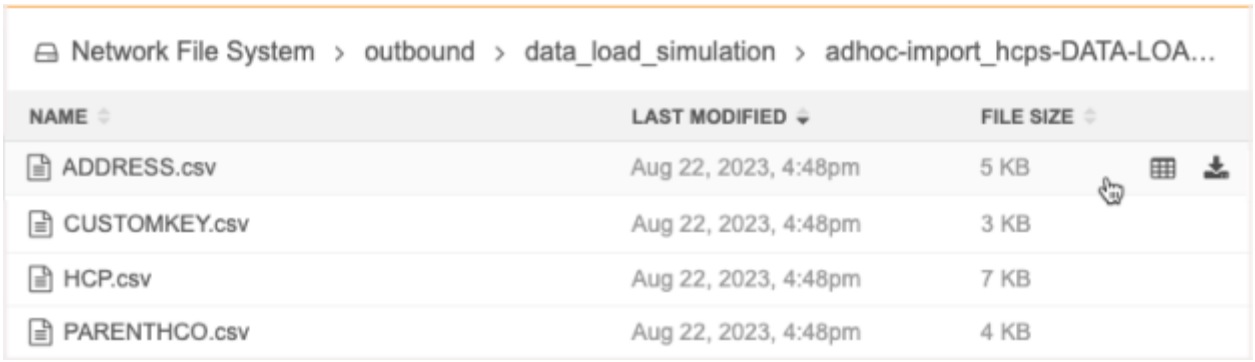
View simulated output files

In Network Explorer, a new folder in the **outbound** directory, **data_load_simulation**, contains the output files for all simulation jobs.

Access to these files is not restricted; if you have access to the folder in the File Explorer, you can view the output files.

Each simulation job creates a .zip file with the following naming convention: <*system*>-<*subscription_name*>-data-load-simulation-<*timestamp*>-job<*ID*>.zip.

The .zip file contains a .csv file for each entity that was updated or that has new records. The complete entity is included in the output, for example, if HCPs were updated or new HCPs were added (either in the data feed or they were matched and downloaded from OpenData), all of the sub-object and relationship data is added to the simulation output.



Note: If the simulation job does not create records or change any records, an empty .zip file is created.

Review the files

Reviewing the files can help you to see what might have gone wrong during the data load.

1. In the **data_load_simulation** directory, click the **Download** icon the to save the .zip file locally.
2. Extract the .csv files. A file is created for each entity that has updates or has new records. If an entity was not touched during the job, a file is not created.
3. Open a .csv file to review the simulated data.

The field values can help you to quickly identify the changes that occurred. For example, in the Address file, you can spot addresses that were inactivated or that were copied (synced) from a parent address through Network Address Inheritance.

address_line_1__v	address_line_2__v	address_line_3__v	address_ordinal__v	address_status__v	address_type__v	address_verification_status__v	administrative_area__v
Bergmannstr. 5-7	Kreuzberg	10961 Berlin	2	I	P		DE-BE
Engeldamm 35	Mitte	10179 Berlin	1	A	P		DE-BE
Kochstraße 8	Mitte	10969 Berlin	1	A	P		DE-BE
Lindenalle 14	Mitte	10177 Berlin	2	A	P		DE-BE

Tip: You can also view the .csv files as smart tables in Network.

Considerations for simulated output

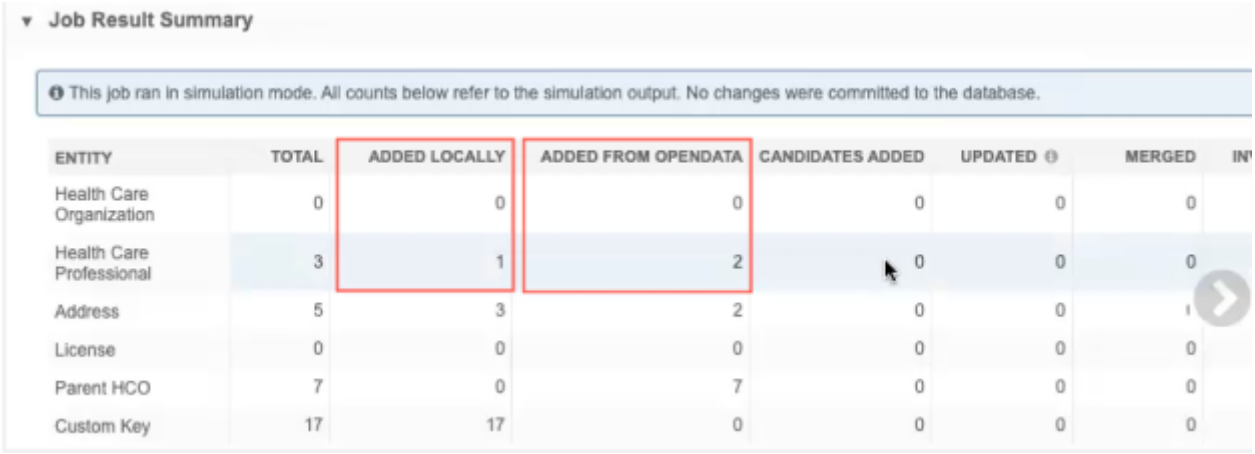
Records matched and downloaded from OpenData

If the **Match & Download from OpenData** setting was on in the source subscription configuration, records that matched will display in the simulated output.

- **Job details** - Records in the source file that are matched with OpenData records display in the **Job Result Summary**.

  Example

  if the data would have been committed to the database, one HCP record would have been created and two HCP records would have been downloaded from OpenData.



- **Simulated output files**

  For records that matched with OpenData records and would have been downloaded, the simulated files contain the data from OpenData as well as the data included in the source file.

Simulated alternate keys

If new records are created with an alternate key, the key counter is not incremented in simulation mode. Placeholder characters are used to reflect the key format.

Examples

In the simulated output, the alternate keys reflect the key format only; an alternate key is not generated.

- **Alpha-numeric keys**: The uppercase letter `X` is used as a placeholder: If the alternate key format is USZJF-RQK-NKB, the simulated output displays `XXXXX-XXX-XXX`.
- **Numeric keys** - The number `9` is used as a placeholder. If the alternate key format is 00000-0006, the simulated output displays `99999-9999`

Report on simulated data

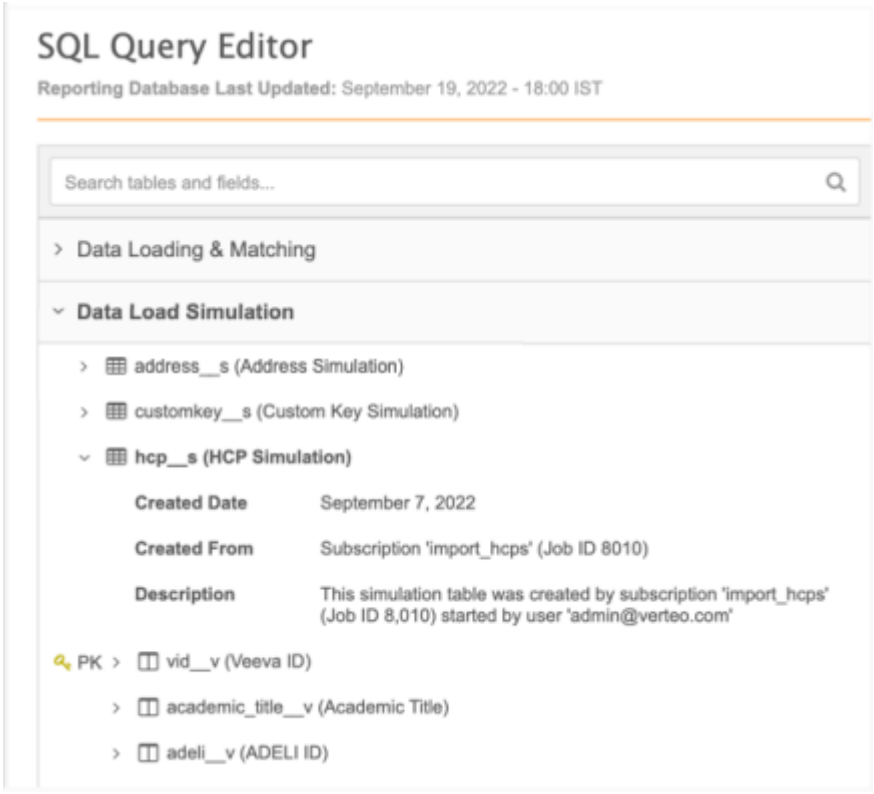
Reporting tables are created for the simulated data in the SQL Query Editor (**Reports**). Use the tables to generate reports so you can compare the simulated data to the production data in order to understand which changes the job would have made to your data.

If you have access to the SQL Query Editor, you can view the simulated tables and data. There are no access restrictions; for example, data visibility profiles, field restrictions, and dynamic access control do not apply.

To view the tables:

- In the SQL Query Editor, expand the **Data Load Simulation** category. The simulated files are created as tables with the `__s` suffix.

Note: Only the tables from the last run simulation job display; the tables are replaced each time a simulation job is run. Reporting tables are not created if a simulation job does not create any new or changed records; the tables from the last job that produced changes are retained.



- Expand a table to view the metadata and the fields. The metadata includes the subscription name, the job ID, and the user that created the simulation output.

Important: The tables include all data model fields for each entity so they have the same structure as production tables. This makes it easier to write the queries and compare the data.

Example queries

Create SQL queries to compare the simulated data with the production data.

Tip: Use left joins to join a simulation table with the production table because you might have new records that do not exist in the production table yet.

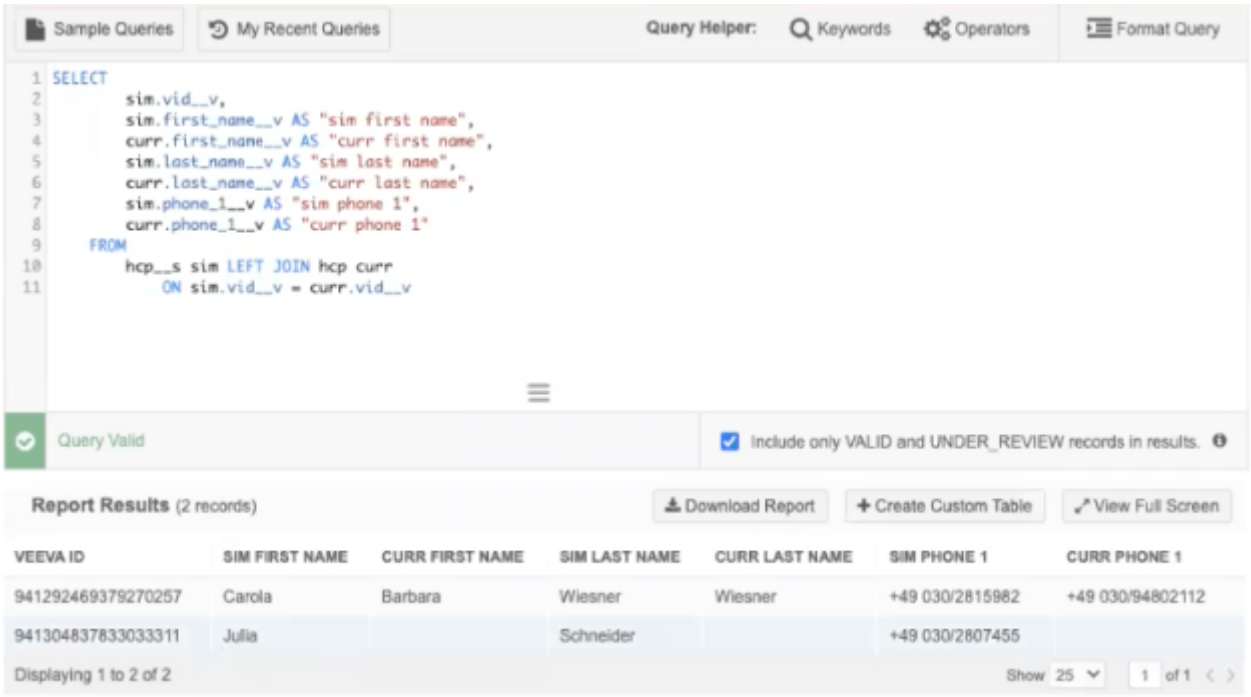
Example 1 - HCP query

This query compares specific fields between a simulated HCP table with the production data.

```
SELECT
         sim.vid__v,
         sim.first_name__v AS "sim first name",
         curr.first_name__v AS "curr first name",
         sim.last_name__v AS "sim last name",
         curr.last_name__v AS "curr last name",
         sim.phone_1__v AS "sim phone 1",
         curr.phone_1__v AS "curr phone 1"
     FROM
         hcp__s sim LEFT JOIN hcp curr
             ON sim.vid__v = curr.vid__v
```

Example results

In these results, you can see that the first row makes changes to an existing record and the second row creates a new record.



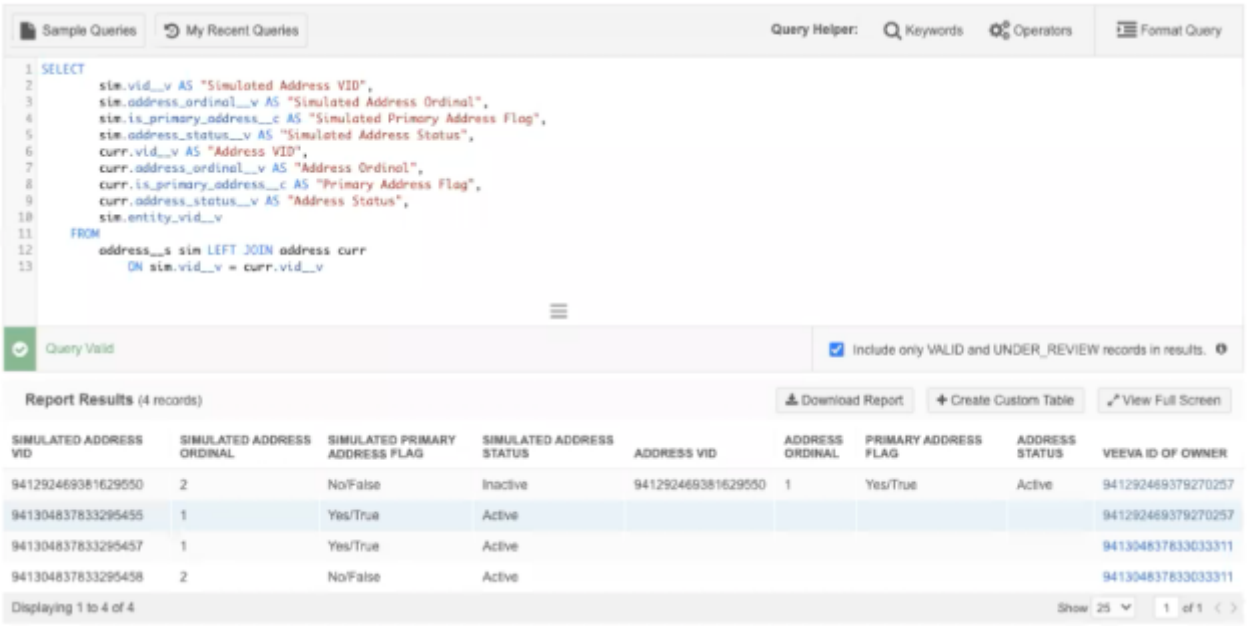
Example 2 - Address query

This query compares the data in the address simulation table to production data.

```
SELECT
         sim.vid__v AS "Simulated Address VID",
         sim.address_ordinal__v AS "Simulated Address Ordinal",
         sim.is_primary_address__c AS "Simulated Primary Address Flag",
         sim.address_status__v AS "Simulated Address Status",
         curr.vid__v AS "Address VID",
         curr.address_ordinal__v AS "Address Ordinal",
         curr.is_primary_address__c AS "Primary Address Flag",
         curr.address_status__v AS "Address Status",
         sim.entity_vid__v
     FROM
         address__s sim LEFT JOIN address curr
             ON sim.vid__v = curr.vid__v
```

Example results

The results can tell you when ordinals are recalculated because addresses are inactivated.

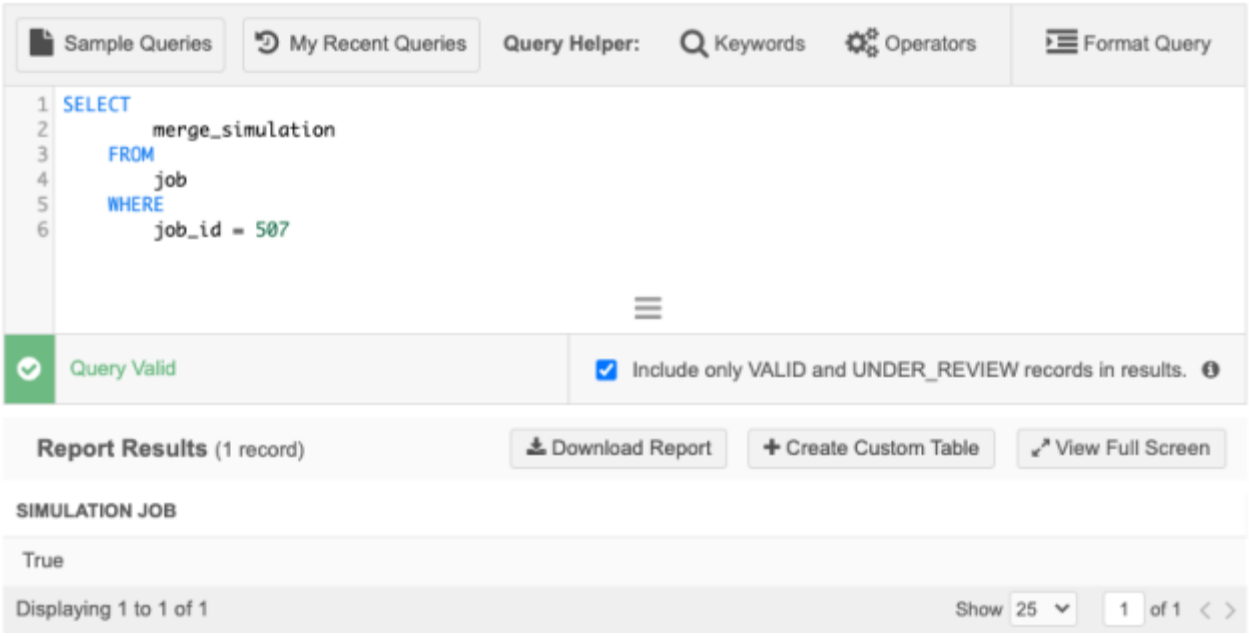


Identify jobs that ran in simulation mode

Use the `merge_simulation` column in the **job (Job Details)** table to identify source subscription jobs that ran in simulation mode.

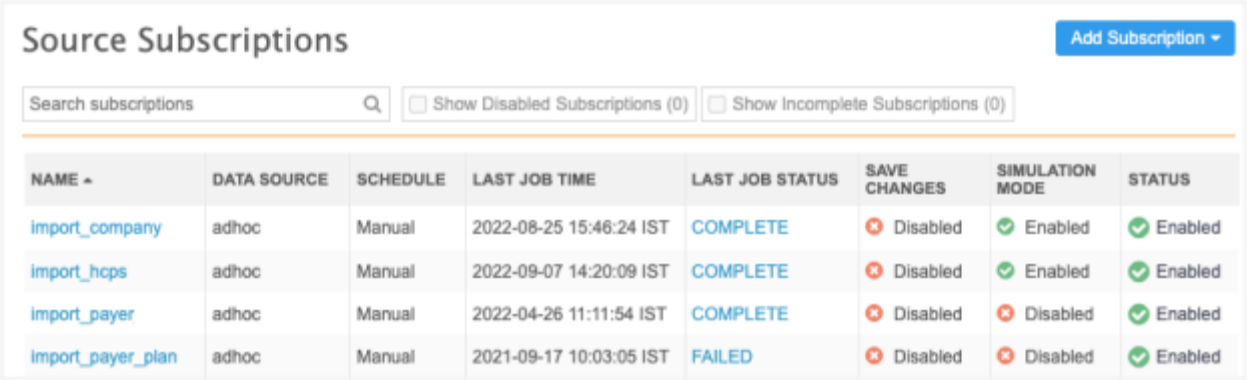
The column value is either True or False.

Example



Source Subscriptions view

The Source Subscriptions page (**System Interfaces**) identifies the current configuration for each subscription.



Save Changes

- Enabled - The subscription is configured to commit changes to the database.
- Disabled - The subscription is configured to run in test mode. No changes will be committed to the database.

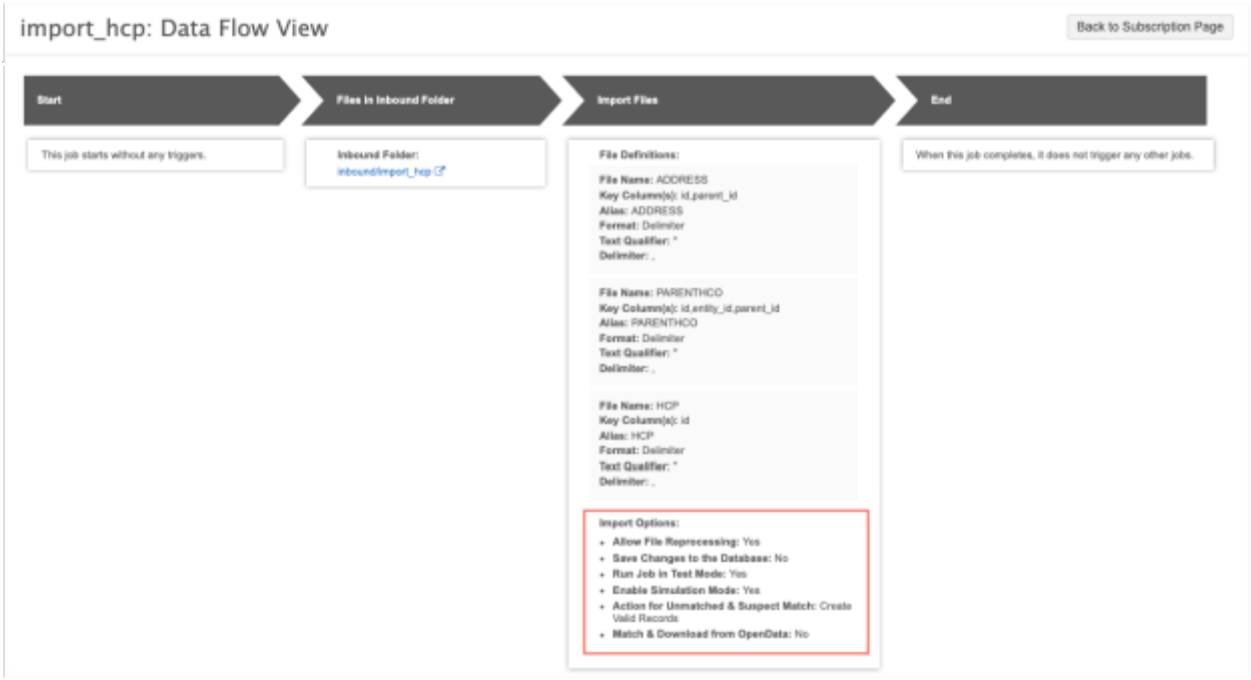
Simulation Mode

- Enabled - The subscription is configured to simulate the job output. No changes will be committed to the database.
- Disabled - The subscription is configured to run in test mode. No changes will be committed to the database.

Data flow view

After a source subscription is saved, the **Data Flow View** displays on the configuration page.

The setting values that are applied to the current job display in the **Import Files** stage.



Considerations for Veeva IDs

When incoming records are not matched to existing records in Network, new records are created for the simulated data. Veeva IDs that are generated for these new records are used only for the simulated output. If you run the subscription again to commit the data to the Network database, new Veeva IDs and are generated for the records.

Duplicate custom keys

If you load an entity with a custom key and that key already exists for a different entity in the production data, an error will display in the **Job Error Log**. You can fix the issue before you commit the data.

When different entities have the same custom key in an incoming data load, the duplicate keys cannot be detected in the simulated output. These types of duplicates can be detected, but only when the production database is updated.

Advanced properties

Some properties that are used for simulated jobs are read-only to ensure that the values do not change if they are copied and pasted between subscriptions.

The following properties are read-only:

- job.match.skipMerge
- job.immutable
- job.simulation

These properties cannot be changed in **Advanced Mode**.

Running jobs using the API

When jobs are triggered externally using the API, use the `mode` parameter to control the subscription job outcome. This is helpful because you can easily toggle between the different job modes without always having to update the subscription settings.

Note: Using this parameter controls the specific job that is started by the API call; it does not change the subscription configuration.

Supported endpoints

- `api/{version}/systems/{system_name}/{subscription_type}/{subscription_name}/job`
- `api/{version}/subscription/{subscription_name}/job`

Possible parameter values

These values can be used for the `mode` parameter.

- **default** - Run the subscription job using the current subscription configuration.

When one of the following parameter values are used, it overrides the configuration saved in the subscription for this job only.

- **save_changes** - The changes are saved to the database.
- **test_mode** - The job runs in test mode only.
- **simulation_mode** - The job runs in simulation mode.

Example request

```
api/{version}/systems/{system_name}/{subscription_type}/{subscription_name}/job?mode=simulation_mode
```