

Spectrum™ Technology Platform

Version 2019.1.0

Master Data Management Guide



Notices

© 2019 Pitney Bowes. All rights reserved. MapInfo and Group 1 Software are trademarks of Pitney Bowes Software Inc. All other marks and trademarks are property of their respective holders.

USPS® Notices

Pitney Bowes Inc. holds a non-exclusive license to publish and sell ZIP + 4® databases on optical and magnetic media. These trademarks are owned by the United States Postal Service: CASS, CASS Certified, DPV, eLOT, FASTforward, First-Class Mail, Intelligent Mail, LACS^{Link}, NCOA^{Link}, PAVE, PLANET Code, Postal Service, POSTNET, Post Office, RDI, Suite^{Link}, United States Postal Service, Standard Mail, United States Post Office, USPS, ZIP Code, and ZIP + 4. This list is not exhaustive of the trademarks belonging to the Postal Service.

Pitney Bowes Inc. is a non-exclusive licensee of USPS® for NCOA^{Link}® processing.

Prices for Pitney Bowes products, options, and services are not established, controlled, or approved by USPS® or United States Government. When utilizing RDI™ data to determine parcel-shipping costs, the business decision on which parcel delivery company to use is not made by the USPS® or United States Government.

Data Provider and Related Notices

Data Products contained on this media and used within Pitney Bowes Software applications are protected by various trademarks and by one or more of these copyrights:

- © Copyright United States Postal Service. All rights reserved.
 - © 2014 TomTom. All rights reserved. TomTom and the TomTom logo are registered trademarks of TomTom N.V.
 - © 2016 HERE Fuente: INEGI (Instituto Nacional de Estadística y Geografía) - Based upon electronic data © National Land Survey Sweden.
 - © Copyright United States Census Bureau
 - © Copyright Nova Marketing Group, Inc.
- Portions of this program are © Copyright 1993-2019 by Nova Marketing Group Inc. All Rights Reserved
- © Copyright Second Decimal, LLC
 - © Copyright Canada Post Corporation - Data is from a compilation in which Canada Post Corporation is the copyright owner.
 - © 2007 Claritas, Inc.

The Geocode Address World data set contains data licensed from the GeoNames Project (www.geonames.org) provided under the Creative Commons Attribution License ("Attribution License") located at <http://creativecommons.org/licenses/by/3.0/legalcode>. Your use of the GeoNames data (described in the Spectrum™ Technology Platform User Manual) is governed by the terms of the Attribution License, and any conflict between your agreement with Pitney Bowes and the Attribution License will be resolved in favor of the Attribution License solely as it relates to your use of the GeoNames data.

Table of Contents

1 - Introduction to Master Data Management

2 - Persistent Storage

Creating Models	7
Querying Models	8
Updating Models	9
Deleting Models and Model Data	9

3 - Visual Modeling

Creating and Updating Models with Visual Modeling	11
---	----

4 - DataStewardship

Data Security	14
Monitoring Data	14
Splitting and Merging Entities	15
Editing Entities and Relationships	15

5 - Data Visualization

Styling Data	18
Data Layouts	18
Data Charts	19

6 - Analytics

Centrality Algorithms	21
Job Details	21

7 - Data Hub Module

Introduction	24
Data Hub Settings	24
Data Hub Visualization	26
Data Hub Browser	66
Data Hub Stages	84
Relationship Analysis Client	132
Manage Models	227
Data Hub Command Line Reference	233

1 - Introduction to Master Data Management

The Master Data Management Data Hub Module identifies and measures relationships between people, groups of people, or almost any other type of entity. It can use your data to identify hidden, latent, or non-obvious connections among people, groups, and organizations. It can also identify shifting patterns of connectedness, distance, centrality, influence status, and importance. With this information, you can predict customer influence and impact and categorize entities as leaders, followers, influencers, outliers, or late adopters. You can also predict churn and offer more personalized customer service by anticipating customer needs and optimizing offerings. You can even use it to detect and prevent fraud, crime, and terrorism.

[In this section](#)

2 - Persistent Storage

The Master Data Management Data Hub Module supports the practice of persistent storage—or Create, Read, Update, Delete (CRUD). It is a complex tool that provides multiple methods of achieving the same goal, depending on your data and how you want to build and use your models.

[In this section](#)

Creating Models	7
Querying Models	8
Updating Models	9
Deleting Models and Model Data	9

Creating Models

There are several different methods to create Data Hub models.

- You can use the Write to Hub stage or the Import to Hub stage to create a model from entities and relationships defined by incoming data.
- You can use Metadata Insights to create a Data Hub model from an existing logical model.
- You can use Metadata Insights to create a Data Hub model and add entity and relationship types and configure their properties.
- You can use Visual Modeling in the Relationship Analysis Client Toolbox to manually add entity and relationship types in a Data Hub model and configure their properties.

A model structure can be populated by the Write to Hub stage. You may also choose to populate a model from within Relationship Analysis Client.

Write to Hub

The Write to Hub stage uses your input data to define entities and relationships that can be queried against in a Read From Hub stage or a Query Hub stage, or that can be visualized in the Relationship Analysis Client to identify relationships and trends that otherwise may be hard to find.

In addition to defining entities and relationships, you can define processing options and determine how you want data to be written to the hub.

Import to Hub

The Import to Hub stage, like the Write to Hub stage, uses your input data to create a model that can be queried against or visualized in the Relationship Analysis Client.

The Import to Hub stage differs from Write to Hub in that it requires two incoming channels of data: one for entities and one for relationships. You do not need to manually define those elements as you do with Write to Hub.

Visual Modeling

You can use visual modeling in the Relationship Analysis Client to create models by manually adding entities and relationships from the Toolbox. You first need to establish entity types and at least one

type of relationship. After that you can either use the same entity and relationship types to populate your model, or you can continue to create additional entity and relationship types.

Another way to create and populate your model in the Relationship Analysis Client is to use metadata. To do this, add information types, in the form of entities, and then relationships between entities. Populate the model manually from within the Relationship Analysis Client by adding properties for the entities and relationships. Alternatively, you can populate the model in the Write to Hub stage by mapping input fields to entity types and defining relationships.

You can now also visually edit a model in Metadata Insights. To do this, click **Model** on the Metadata Insights toolbar and then click the **Data Hub Model** tab. There you can visually edit entities, relationships, properties, and indexes in a new or existing Data Hub model.

Models created with visual modeling can be queried against just like models created in Write to Hub or Import to Hub.

Querying Models

The Master Data Management Data Hub Module provides Query Hub and Read from Hub to query and extract data from a model.

Query Hub

The Query Hub stage is an intermediate that uses incoming data to extract specific entities and relationships from a model.

Read from Hub

The Read From Hub stage uses a saved query or new query to read data from an existing model, then returns data as fields in the dataflow output for use with other stages or processes.

Query Timeout

Data Hub queries terminate execution if they take longer than 60 seconds to complete. This default behavior can be modified or disabled.

Note: The Spectrum server must be restarted for these changes to take effect.

To modify the time that queries run before timing out and terminating, open the `SpectrumProgramFiles/server/modules/hub/hub.properties` file and set the `hub.query.timeout` property to the new value in milliseconds. For example, to set the new time to 120 seconds, enter "120000" (120 x 1000).

To disable the timeout feature, open the `SpectrumProgramFiles/server/modules/hub/neo4j.properties` file and set the `execution_guard_enabled` property to "false".



Updating Models

Once a model has been created, you can update it by regenerating it from Write to Hub or Import to Hub.

You can use the **Model** page in Metadata Insights to manually add entities and relationships to a Data Hub model. You can also use Metadata Insights to add, delete, or modify properties that are associated with a model entities and relationships. Similar functionality is still at least partially available but deprecated in Relationship Analysis Client.

Deleting Models and Model Data

The Data Hub Module includes tools that enable you to delete data from a model or delete models themselves. To delete data from a model, you can remove that data from your input files and regenerate the model. Another approach would be to manually delete entities and relationships from within the Relationship Analysis Client. To do this, select one or more elements from your model and press Delete. You can also right-click the selection and click **Delete** on the shortcut menu.

You can use Metadata Insights to delete a model. On Metadata Insights menu bar, click **Model**, then click the **Data Hub Model** tab. Check the check box next to the model you want to delete, then click the Delete model  button. Alternatively, use the **Model Management** tool in the Relationship Analysis Client. First click Manage  on the toolbar to open the **Model Management** dialog box. Then, on the **Models** tab, click the model you want to delete and click **Remove**.

3 - Visual Modeling

In this section

Creating and Updating Models with Visual Modeling

11

Creating and Updating Models with Visual Modeling

The Master Data Management Data Hub Module provides multiple methods of creating models. One way is to use the **Write to Hub** stage or the **Import to Hub** stage to create a model from entities and relationships that have been defined from incoming data.

Another way to create models is to perform visual modeling in the Relationship Analysis Client by manually adding entities and relationships from the Toolbox. To start, you likely would need to establish at least two types of entities and at least one type of relationship, but after that you could either use the same types of entities and relationships to populate your model, or you could create additional types of entities and relationships. For instance, if you were creating a model of insurance data that was comprised of patient information, you might need an entity type of "Patient" for patient names and an entity type of "Claim" for the illness or injury that instigated the claim. However, if your insurance model was comprised of patients and doctors, you would need to add a third entity type of "Doctor". The relationship type between patients and doctors might be "visited", and the relationship type between patient and claim might be "claimed" or "reported". Alternatively, you could have a relationship type between "Doctor" and "Claim" of "diagnosed". Using this data you could build a model that identifies potential fraud data if you see a doctor repeatedly diagnosing—or a patient repeatedly claiming—a particular injury or illness.

Within the visual modeling method in the Relationship Analysis Client, another way you can create and populate your model is by using metadata. First you add information types, in the form of entities. From there you can add relationships between those entities and completely populate the model from within the Relationship Analysis Client, manually adding properties for the entities and relationships, or you can instead populate it by mapping input fields to entities and defining relationships from within the Write to Hub stage. Using the previous example, let's say you create entity types of "Patient", "Doctor", and "Claim" in the Relationship Analysis Client and then save your model with the name "Insurance Data". Then, you could go into Enterprise Designer and create a dataflow that has a Read from File stage going into a Write to Hub stage. You could configure the Read from File stage to view an input file of insurance data that includes the following fields (among others): "TIN", "SSN", and "Dx". In the Write to Hub stage, you would select "Insurance Data" in the "Model" field and then define entities by mapping input field "TIN" to type "Doctor", "SSN" to type "Patient", and "Dx" to type "Claim". After creating these entities, you would then define relationships between them. You could have "Patient_visited_Doctor" or "Doctor_treated_Claim" or "Patient_reported_Claim". After defining all the entities and relationships, you would run the job in Enterprise Designer, then go back to the Relationship Analysis Client, and you would see the "Insurance Data" model populated with the information that was in your input file, showing the relationships between the entities you just defined.

In addition to using the Relationship Analysis Client Toolbox to visually create models, you can use the Toolbox to add entities and relationships to an existing model. If you were looking at your "Insurance Data" model, you could create additional relationships between entities, or you could create new entities and add them to your model, linking them to other entities with existing or new relationships.

As you conduct visual modeling by creating a model or adding elements to your model, the Overview pane, the Model Details pane, the Model Components pane, and the Selection pane update accordingly, reflecting the properties in your model. This dynamic persistence enables you to use the tools of the Relationship Analysis Client in real time as you update your model.

For more information on visual modeling, see [Creating Models](#) on page 137 and [Editing Models](#) on page 218.

4 - DataStewardship

There are several ways in which the Master Data Management Data Hub Module provides for data stewardship. You can apply role-based security measures, designate who can execute algorithms, monitor your data for specific changes and automatically take action when certain conditions are met, and edit the components of your model.

In this section

Data Security	14
Monitoring Data	14
Splitting and Merging Entities	15
Editing Entities and Relationships	15

Data Security

Spectrum™ Technology Platform uses a role-based security model to control access to the system. In the Management Console you can use user-created roles to grant or deny a user permission for accessing various parts of the Data Hub Module. Access control allows you to override permissions defined in the [Secured Entity Types - Data Hub Module](#) settings based on a role or a user.

Example:

You create a role using the Secured Entities - Data Hub Module settings that allows users to view and modify all properties for a model. You then want to deny access to a specific property, such as "Social_Security_Number", for selected users. To deny access for that property, you would create an access control that overrides permissions defined in the role.

For more information on managing Data Hub security, see [Roles](#), [Secured Entity Types - Data Hub Module](#), and [Access Control](#) in the *Spectrum™ Technology Platform Administration Guide*.

Monitoring Data

The Relationship Analysis Client can monitor your models for data changes that meet certain conditions. When those conditions have been met, you can designate that an email is sent or a service is run. This kind of automated scrutiny enables you to keep an eye on your data without requiring the time or expense of manual observation. You can simply have an email notification be sent to you or someone else, or you can tell Spectrum™ Technology Platform to run a service of its own or a custom service that you have created.

You can set monitors for entities or relationships. Conditions for entity monitors are based on entity properties, which are evaluated against literals, or absolute values. For example, you could set a condition on a property called "Nationality" and have a service run when an entity's nationality is "Swedish". Conditions for relationship monitors are based on relationship properties as well as properties for the source and target entities attached to those relationships. As with entity monitors, these properties can be evaluated against literals, but they can also be evaluated against the properties themselves. For example, you could set a condition to look for when a relationship property is equal to that same property on the source or target entity. Likewise, you could set a condition to look for when a source entity's property is equal to that same property on a target entity.

To implement monitoring, you need to assign a name to your monitor, set the conditions under which you want to take action, and set up either an email notification or a service activation.

Note: Before creating an Email Action monitor, you must first set up Notifications in Management Console.

Click [here](#) for more information on monitoring data.

Splitting and Merging Entities

Another way Master Data Management helps you control your data is by providing the flexibility of splitting and merging entities in the Relationship Analysis Client.

If your model contains an entity that performs multiple functions within that model, you could split that entity into two entities: one entity for one function, and a second entity for the other function. One way this feature might be useful is in an organizational chart. Most entities are employees of someone in an organizational chart, but some of those employees are also managers. If you needed to maintain the integrity and structure of an organizational chart containing someone who fulfills both roles, you could simply split that one entity into two separate entities. One entity would fall into the employee role, while the other would fall into the manager role. In addition to splitting one entity into two, you can also determine which properties and relationships are tied to each specific entity—all, some, or none. Or, you can tie properties and relationships to both entities; they don't have to be tied to one or the other.

If your model contains two entities that represent the same subject, you can merge them and their respective data together into one entity. Let's say that your organizational chart shows one entity performing accounts receivable and another entity performing accounts payable. Then, the person performing accounts payable resigns, and due to budget constraints you are not able to replace her. Instead, the person conducting accounts receivable will absorb her duties. In this case, you could merge those two entities together, keeping some of the properties of the accounts receivable employee (her name, possibly her title) and adding some of the properties of the accounts payable employee (her primary responsibilities, or possibly her manager).

The benefit is that you can tweak your data to account for changes without having to recreate your model.

For more information, see [Splitting Entities](#) and [Merging Entities](#).

Editing Entities and Relationships

The Relationship Analysis Client is a flexible tool that enables you to manually add or delete entities or relationships from models. This can be useful if you don't want to recreate the model but have data in the form of an entity that you would like to add, or if your model contains entities or relationships that are no longer valid.

You can also add, edit, or delete properties and modify values for properties that are associated with entities and relationships. An example of this functionality might be that if your model uses data

that is looking to identify potential insurance fraud and one of the entity properties for a provider is his legal status for practicing medicine. Because there are ongoing investigations of this individual, you might need to change the value of his "Legal_Status" property to "prosecuted."

The pliability of real-time model editing saves you time and money by not having to alter your input data and recreate the model.

See [Editing Models](#) for more information on editing entities and relationships.

5 - Data Visualization

You can use Relationship Analysis Client to style entities and relationships, style the model, and use themes and criteria to apply styles and filters. You can use the Layouts feature to change the appearance of model elements and visually characterize entities and relationships. You can use charts to visualize your data.

In this section

Styling Data	18
Data Layouts	18
Data Charts	19

Styling Data

There are various features within the Relationship Analysis Client that enable you to draw attention to the information in your model that you find most important. You can determine the size, shape, and color of your model's entities, including selecting from a library of stock images that depict people, places, transportation, finance, and more. For relationships you can specify line width, color, and point decoration for both ends. You also can designate label settings for both entities and relationships.

The Gradient feature is a tool that enables you to visually change the style of a model using actual values in the data. You can represent model elements and highlight important elements based on their values by starting at one setting for the lowest value and another setting for the highest value. For instance, if your model uses insurance claim data and one of the entity properties is the number of claims filed by a provider, you could set pale yellow as the color for the entity with the lowest number of claims and orange as the color for the entity with the highest number of claims, and the entities for everything in between those two dates would fall somewhere between.

The Themes feature applies styles and filters to elements in your model based on conditions you set. You can predetermine aesthetic values for entities and relationships that meet the criteria set in your conditions. When parts of your model meet the criteria set, the theme will be applied and you will be able to visually identify those elements. You can even use filters to automatically hide from your model the elements that meet the set criteria.

For more information about styling the data in your model, see [Changing the Appearance of Model Data](#) on page 191.

Data Layouts

The Layouts feature within the Relationship Analysis Client enables you to change the appearance of model elements to show the entities and relationships visually. Certain data renders better in a particular model layout. The Relationship Analysis Client offers five layouts for your models. There are also options respective to each layout and options that are common to all layouts.

- Circular Layout—Emphasizes group and tree structures within a network.
- Hierarchic Layout—Highlights the main direction or flow within a directed model.
- Organic Layout—Based on the force-directed layout paradigm.
- Orthogonal Layout—Is a multipurpose layout provider for undirected models; produces compact drawings with no overlaps, few crossings, and few bends.
- Map Layout—Depicts locations of entities within your model using latitude and longitude.

For more information on these layouts and common options for layouts, see [Using Layouts](#).

Data Charts

The Charts feature of Relationship Analysis Client enables you to visualize data from a model in graphic format. You can select from four options the one that will best depict your data:

- **Data bar**—Presents numeric data from a model in bar chart format. Therefore, only properties with numeric formats are eligible for use with data bar charts. Data bar charts display each entry for the selected criteria from highest to lowest value. This represents the order of influence within the model, from most influential to least influential.
- **Treemap**—Presents data in a hierarchical manner, resulting in several individual squares and rectangles that together form a whole. The size and color of each rectangle reflects its value and influence within the model, and when you hover over any given block the data upon which you created the map will appear.
- **Heat map**—Presents data in table form using colors. A simple example of this is might be crime rate data over the years by month. Each row represents one of the twelve months, and each column represents a year in the time frame. Colors could range from very pale blue (lower crime rates) to dark blue (higher crime rates). You could use more than one spectrum of color, especially if you wanted your heat map to be more specific.
- **Timeline**—Presents data across a span of time. It can be used for entities, relationships, or both. You can plot events, actions, and so on over a course of time, and once the timeline is created you can zoom in or out to focus on more or fewer items.

For more information, see [Using Charts](#).

6 - Analytics

In this section

Centrality Algorithms	21
Job Details	21

Centrality Algorithms

Centrality algorithms measure the importance and significance, also known as the centrality, of individual entities and relationships in a model. Use these algorithms to distinguish leaders and followers, influencers and outliers, and similar characteristics of your data.

The Relationship Analysis Client provides four kinds of centrality measures to apply to your model:

Betweenness	Used to identify entities that control the information flow between different parts of the network.
Closeness	Used to identify entities that may have best access to other parts of the network and visibility of activities within the rest of the network.
Degree	Used to identify entities that have the most direct links to others.
Influence	Used to identify entities that have strong influence in the network due to their direct links to other highly active or well-connected entities.

There are three types of directions an algorithm can be run:

Incoming	The results will be based on relationships coming into the entity.
Outgoing	The results will be based on relationships going out of the entity.
Both	The results will be based on incoming and outgoing relationships.

There are additional measures that can take place with centrality algorithms. For instance, you can select, roughly, how precise the results should be. A lower precision will return more accurate results, but the algorithm will run more slowly. With the Closeness algorithm, you can select how the results should be returned. You can also designate that the relationship property should be used as weight or that low values should be considered more significant, such as if you were using data that ranks entities, and #1 was the best possible ranking.

Click [here](#) for more information on centrality algorithms.

Job Details

The Model Management feature in the Relationship Analysis Client provides important statistics regarding jobs that you have run when applying algorithms to your model. This information includes:

- Job ID
- Model name
- Algorithm used
- Status of the job

- Start time
- End time
- Any messages that may have been returned when the job was run

Larger models can take a long time to run; the Status column indicates if a job is still running.

7 - Data Hub Module

In this section

Introduction	24
Data Hub Settings	24
Data Hub Visualization	26
Data Hub Browser	66
Data Hub Stages	84
Relationship Analysis Client	132
Manage Models	227
Data Hub Command Line Reference	233

Introduction

The Data Hub module supports several different client interfaces.

Data Hub Visualization	A next generation relationship analysis tool. It presents data in a graphical form, showing the data structure and the relationships among the entities. Analyzing data visually enables users to gain insights they may miss using traditional table-based numeric analysis.
Data Hub Browser	Offers a discovery tool where you can search the contents of models by entering natural-language queries and browsing the results.
Data Hub Stages	Data Hub stages encapsulate Data Hub operations in dataflows. Dataflows are created in Enterprise Designer and executed with the Job Executer or Process Flow Executer client tools. These stages may also be incorporated in service dataflows, which can be accessed as web services or using the Spectrum™ Technology Platform API.
Relationship Analysis Client	Used to view data in a graphical manner that identifies and depicts structure and relationships between entities. You can also use the tool to manipulate entities and relationships based on their properties and values.
Command Line Interface	The Administration Utility Command Line Interface (CLI) provides Data Hub commands to manage models, view jobs and job status, perform backups, run centrality algorithms, and manage model schemas.

Data Hub Settings

Data Hub Settings enables users with Administration view and modify permissions to set preferences for audit logs and model backups.

Accessing Data Hub Settings

The **Data Hub Settings** page is located in the Management Console application.

1. On the Applications menu , click **Management Console**.
2. On the banner, click **Resources > Data Hub Settings**.

Data Hub Settings








The Data Hub settings are displayed in the Management Console application by clicking **Resources > Data Hub Settings**.

Audit model events	Check this check box to have the Data Hub Module maintain a log of when models are created, modified, or deleted.
Include metadata events	Check this check box to include metadata activity in the audit log. Audit model events must be enabled to include this activity.
Include read events	Check this check box to include data for when models are viewed in the audit log. Note that including this data can significantly impact storage limitations for the audit log. Audit model events must be enabled to include this activity.
Track history	Check this check box to enable the History feature in the Relationship Analysis Client. Doing so enables you to view changes made to entities and relationships.
Override default backup directory	<p>Check this check box to specify the path to an existing folder in which to save backups of Data Hub models. Each backed up model is saved to this location in a folder named <code>model.ModelName</code>. This setting overrides the default location for Data Hub model backups (<code>SpectrumFolder\server\modules\hub\db\backups</code>). The logon account for the Spectrum™ Technology Platform server must have write permission to the specified folder.</p> <p>Note: The default backup directory location (<code>SpectrumFolder/server/modules/hub/db/backups</code>) may be changed by uncommenting and editing the <code>hub.models.path.base</code> setting in the <code>SpectrumFolder/server/modules/hub/hub.properties</code> file.</p>
Backup page cache (MB)	You can increase or decrease the memory in megabytes (MB) used while backing up models. Increasing this value improves performance but uses additional RAM. Allowed values are between 8 and 8192.
Schedule backup	Check this check box to enable backups for your models and to designate the frequency and time at which backups should occur. Check Incremental to have the system use transaction logs to determine what has changed since the last backup and add those changes to an existing backup.
Log transactions	Check this check box to have the wrapper log include messages when transactions begin committing data to a model and when the transaction completes.
Log input data on exception	Check this check box to have the wrapper log include data from the input record that is being processed at the time an exception occurs. Leave this box unchecked to avoid potential security issues with sensitive information such as Social Security numbers, IDs, account numbers, and so on.

- Log virtual queries** Check this check box to log virtual queries.
- Query timeout** Check this check box to specify in seconds how long the server should wait for a query to complete. Allowed values are between 1 and 100000.

Data Hub Visualization

Data Hub Visualization enables you to view and manipulate data from your models in a graphical user interface. The interface has six different workspaces opened by buttons on the vertical workspaces bar:

- Dashboard on page 28**  This workspace provides counts and other details for entities, relationships, properties, and connections. It also identifies multiple top-10 rankings in your data and provides a news feed function that displays updates to watched items.
- Canvas on page 30**  In this workspace, you can manually select, load and unload entities and relationships, or run a query to look for specific data. You can also apply various layouts to your models to review them in different formats and identify additional links.
- Map on page 44**  This workspace plots locations using geographic and address information in your data, which you can use to assess scope and density of activities. You can also run a query to plot items with latitude/longitude data on the map.
- Table on page 45**  This workspace presents data in a tabular format, broken out into entities and relationships. You can modify the model by adding, editing, and deleting entities or relationships. Additionally, you can remove entities and relationships from the table while leaving the model.
- Visual Query Builder on page 46**  This workspace allows you to build simple and complex query functions on existing models in a visual manner.
- Entity Browser on page 54**  Use the Entity Browser workspace to view one or more entity types in a Data Hub model. You can limit the view by applying criteria for values contained by properties of the selected entity types. The criteria are applied by a series or one or more statements connected by a logical And or Or.
- Metadata on page 61**  This workspace provides detailed counts for entities, relationships, entity types, relationship labels, and properties. It also enables you to change the appearance of your model's elements at the metadata level.

Accessing Data Hub Visualization


Perform the steps in this task to access Data Hub Visualization.

For you to perform this procedure, the Spectrum™ Technology Platform server must be running.

1. Open a web browser and navigate to: `http://servername:port/hub-visualization`. For example, if your server is named "myserver" and it uses the default HTTP port 8080, you would enter `http://myserver:8080/hub-visualization`.
2. When the sign in screen appears, enter the same user name and password that was used to launch Enterprise Designer when you created your models and click **Sign in**. This opens a blank Dashboard.
3. On the **Model:** menu, click a model or **All Models** to open a model and view the complete Dashboard for the model.

You can now use all of the features within Data Hub Visualization.

Settings

The Data Hub Visualization settings determine how data models are displayed on the canvas. In the **Data Hub Visualization** application, click the Settings button  on the toolbar to configure the Data Hub Visualization settings.

Auto Open Model

Check this option to open models automatically when you access the Canvas. Uncheck this option to open the Canvas to a blank screen, and then run a query to populate the screen.

Open limit

The maximum number of entities that can be loaded into Visualization at one time. The default is 5000. Click **Save and restore model items between sessions** to preserve which elements of the model are loaded on the Canvas. This setting applies when you switch to another model, sign out of and then sign in to Data Hub Visualization, or refresh the application.

Query limit

Specifies the maximum number of queries that can be requested of Visualization in one session. The default is 100 and the maximum value you can enter is 1000.

Selection fading

Check this option if you want non-selected entities to fade into the background rather than to disappear entirely. Set the **Opacity** level of the fading between 1 for no fading and 0 for invisible. The default is .2.

Auto zoom to selected items

Check this option if you want the canvas to zoom to the selected items rather than retain the focus on the entire model.

Layout timeout (seconds)

The number of seconds that Visualization waits for a layout to load before timing out. The default is 10 seconds.

Dashboard widget timeout (seconds)

The number of seconds that Visualization waits for a dashboard widget to load before timing out. The default is 60 seconds.

Normalize centrality measurements (except degree)

Check this option to use the average length of the shortest path between an entity and all other entities in the model. With normalizing you can compare entities from models of different sizes.


Dashboard

The Dashboard depicts Entities and Relationships widgets that show a breakdown of entities and relationships by category (for instance, entities such as "People," "Places," and "Events," and relationships such as "Visited," "Attended," and "Resided"). It also provides specific counts for the category when you hover over a particular section of the pie.

Additionally, the Dashboard provides three top-10 widgets, which are lists of entities or relationships in a certain order. For example, the Top 10 Connected Entities provides a list of ten entities in the selected model containing the highest number of connections.






Each of the widgets provided also includes a set of tools that you can use to further use the data in the widget.

Widgets

The Dashboard provides the following widgets. Note that at any given time you can have just one entity widget and one relationship widget on the Dashboard. Once you have created your list, you can select one or more entities and perform additional functions using the tools in title bar of the pane: .

All Entities	Depicts a model's entities and properties in a pie chart.
All Relationships	Depicts a model's relationships and properties in a pie chart.
Relationship Counts	Is based on the number of connections an entity has incoming, outgoing, or both. You can retain the default of all entity types or create a top-10 list by entity type, such as the top-10 people, the top-10 places, or the top-10 events within a model. Further, if you do select an entity type, you can also select a relationship label from which to build the list. You can also use this widget to create the same lists for the bottom 10 entities.
Property Value of Connected Items	Shows the top 10 of a selected entity type based on an average property value of connected relationships or entity types.
Property Value	Shows the top 10 of a selected entity type based on selected property value.

Widget Tools



Tools      in widget panel title bar provide additional functionality. Available tools depend on the widget and whether entities are selected in a widget.


Entity profile report



Select a single entity in a widget and click this button to create an **Entity Profile Report**, which displays the following information:


- The total number of connections as well as the number of incoming and outgoing connections as a degree
- Each relationship connected to the entity, which can be set to only incoming, only outgoing, or both.
- The incoming or outgoing entity attached to each relationship, along with its degree of connectedness; click the report icon to see complete information for connected entities
- A list of the entity's properties when you click "More"

Note: The URL of these pages is also unique so you can bookmark or share these pages,

Open to canvas  Select one or more entities and click this button to load them on to the Canvas. You can select entities and click the Details button  to display details or double-click any entity to display its relationships. You use these and any other feature of the Canvas.

Open to map  Select one or more entities containing latitude/longitude information and click this button to place the entities on a map. You then use any feature of the Map tool.

Change settings  Click this to change settings for a widget. Other the Remove widget  button is visible while you configure settings for a widget.

Remove widget  Delete currently widget panel from the **Widgets** application window.

Note: If you accidentally delete a widget, just click the **Add widget** button to locate and restore the panel that you deleted.

News Feed

The Dashboard News Feed feature enables you to track changes to the entities in your model. It provides instant updates for the entities you are following, including the addition, change, or removal of properties, a change in ranking, and so on. A plus sign (+) indicates an added change, and a minus sign (-) indicates a deleted change. News Feed also shows you when the change occurred and which user was responsible for the change. You can view this activity in summary or detail mode.




You can select a News Feed item and click one of the tools to take action. These tools are the same as those offered for widgets. If you open an event entity to the canvas or the map, its connections will display as well.

Click the Refresh button  to update the newsfeed.

Note: To use News Feed you must enable **Track history** on the Data Hub Settings page in Management Console.








Canvas

The Canvas is used to view data in a graphical manner that identifies and depicts structure and relationships between entities. This page includes the following features:





Toolbar	Provides ways to view a model, apply layouts, load, select, and filter items.
Query Tool	Provides a way to query model data and save queries.
Current Item Details	Shows the selected item's label, type, degree status, and properties.
Entity Profile Report ()	Lists information about an entity as well as its connections and properties.
Edit Item Properties ()	Edits the properties for entities and relationships.
Watch Entity ()	Selects the item for your news feed.

Toolbar

The Canvas toolbar at the top of the page provides the following options:





Undo/Redo ()	Undo or redo the last action.
Fit to bounds ()	Set the size of the elements to fit the bounds of the canvas.
Zoom to selected ()	Zoom in on the selected entities on the canvas.
Apply current layout ()	<p>Apply the currently selected layout to the elements on the canvas.</p> <ul style="list-style-type: none"> • Organic ()—Based on a force-directed layout paradigm; useful for models that use enterprise networking, system management, and WWW visualization. • Circular ()—Emphasizes group and tree structures within a network and analyzes the connectivity structure of the network and arranges the partitions as separate circles. Useful for models that depict social networks, network management, and eCommerce. • Radial ()—Places entities on circles around a common center (also known as the root), expanding outward, similar to rings in a tree. The center entity

is the first entity, the first ring contains the "children" entities of the first entity, the second ring contains the "children" entities of the entities from the first ring, and so on. Useful for models that depict social networks, web analysis, or organizations with a central authority and multiple levels of management.






- Hierarchical Edge Bundling —Bundles groups of edges together using edge compatibility measures. This layout reflects high-level patterns, reduces visual clutter, and more clearly displays relationships that are based on fewer connections. Useful for models that depict social networks.
- Hierarchical —Highlights the main direction or flow within a directed model. The entities of a model are placed in hierarchically arranged layers such that most of the relationships of the model show the same overall orientation (for example, top-to-bottom). Useful for models that depict workflows, database modeling, and process modeling.
- Orthogonal —Acts as a multipurpose layout provider for undirected models. It produces compact drawings with no overlaps, few crossings, and few bends. Useful for models that use database schema, system management, and software engineering.
- Tree —Presents data in a hierarchical manner, resulting in several individual squares and rectangles that together form a whole.
















Centrality

Apply the selected centrality algorithm to the open model. The size of the entities and their labels will adjust accordingly.

- Influence —Identifies entities that have strong influence in the network due to their direct links to other highly active or well-connected entities.
- Closeness —Identifies entities that may have best access to other parts of the network and visibility of activities within the rest of the network.
- Betweenness —Identifies entities that control the information flow between different parts of the network.
- Degree —Identifies entities that have the most direct links to others.

Centrality options

- Incoming —Results will be based on relationships coming into the entity.
- Incoming & Outgoing —Results will be based on incoming and outgoing relationships.
- Outgoing —Results will be based on relationships going out of the entity.
- Relationship weight property—Identifies which relationship property should be used as weight for centrality.
 - Low values —Low values for the property are more significant.
 - High values —High values for the property are more significant.


Property Gradients 	Adjust the size of entities, relationships, and their labels. <ul style="list-style-type: none"> • Entity gradient property—Identifies which numeric or temporal entity property should be used as weight for gradients. <ul style="list-style-type: none"> • Lower values are more significant —Low values for the entity property are more significant. • Higher values are more significant —High values for the entity property are more significant. • Relationship gradient property—Identifies which numeric or temporal relationship property should be used as weight for gradients. <ul style="list-style-type: none"> • Lower values are more significant —Low values for the relationship property are more significant. • Higher values are more significant —High values for the relationship property are more significant.
Restore Default Decorations 	Undo centrality and gradient changes and restore defaults.
Select all entities 	Select all entities on the canvas.
Select connected 	Select remaining connections for the selected entities.
Select incoming 	Select all entities that are coming into the selected entities on the canvas.
Select outgoing 	Select all entities that are going out of the selected entities on the canvas.
Load connected 	Load remaining connections for the selected entities within limits set by Load options Maximum Entities.
Load incoming 	Load all entities that are coming into the selected entities on the canvas
Load outgoing 	Load all entities that are going out of the selected entities on the canvas.
Load options 	<ul style="list-style-type: none"> • Fill in Relationships—Return additional relationships revealed in discovery. If the discovered entities were connected to additional entities already on the canvas (other than the any selected prior to using Discovery), those relationships will also be returned. • Relationships—Select which relationships you want to be returned on the canvas. • Maximum entities—Return no more than this number of entities onto the canvas. Default is 100.
Load shortest paths 	Select on the canvas the smallest number of entities and relationships required to connect the two currently selected entities.

Shortest path options

- **Relationships**—Select which relationships you want to be returned in the shortest path.
- **Maximum paths**—Return no more than this number of paths onto the canvas. Default is 20.
- **Maximum path length**—Return a path whose length is no greater than this number. Default is 20.
- **Timeout**—The number of seconds that can elapse before the request times out. Default is 10.

Filters 

Filter entities and relationships by type and property values. You can filter on entities, on relationships, or on both entities and relationships simultaneously.

The Clear Filters  button appears next to the Filters button whenever filters are applied to entities and relationships on the canvas. Click this button to clear the filters and display all of the entities and relationships currently on the canvas.

For more information, see [Filters Tool](#) on page 38.

Unload selected



Remove all currently selected entities.


Unload options

- **Unload selected**—Remove all currently selected entities.
- **Unload unselected**—Remove all entities that are currently not selected.
- **Unload all**—Remove all entities from the canvas.




Query Tool

The Canvas provides a query tool that allows you to search a model. This query function supports a natural-language inspired grammar.

The query tool is initially expanded on the workspace. You can click the Close query button  to collapse the query entry field or the Query button  to expand it. As you enter each step of the query, Data Hub Visualization dynamically returns type-ahead suggestions based on the model metadata. You can press Enter to run the query and load the results on the Canvas. If items specified by the query exist on the Canvas, they will be selected. Click any entity on the Canvas to view its **Current Item Details**.

After you create a query, you can click the Save query button  and provide a name for the query. This allows you to use a query repeatedly on the Query Hub stage, the Read from Hub stage, the Relationship Analysis Client, or the Data Hub Browser.

Note: Query names must start with letters or an underscore and must contain only letters, numbers, periods, and underscores. They cannot contain spaces.

To use a saved query, select it from the **Select a query** list on the toolbar, and click the Run query button . The Query inputs button  is active when a query has input fields. Click the Query inputs button  and to enter values for fields on the **Query Inputs** drop-down box.

Query Structure

Whether you are searching entities or relationships, the more specific you are in your query, the more focused the results become. Entered values can contain any Unicode character, excluding control characters. Values containing words that match metadata names should be wrapped in single or double quotes.

The asterisk (*) wildcard character following a relationship will return all entities matching the rest of the query. This is especially useful for more complex queries.

Entities

Entity Queries

The root of an entity query targets an entity index. Therefore, any properties you may want to query should be indexed. Use case-insensitive indexing to yield greater search results.

Entity Patterns

An entity query follows one of these patterns:

EntityType Property value

EntityType Property Operator value

EntityType Property value condition Property value

EntityType Property Operator value condition Property Operator value

EntityType value

Property value

Property Operator value

EntityType

value

Query Operators

The following operators are valid for entity queries. Additionally, you can use wildcard characters with the `Is Like` operator. An asterisk (*) represents any character sequence, and a question mark (?) represents any single character.

Operator	Property Type
<	Numeric or temporal
<=	Numeric or temporal
>	Numeric or temporal
>=	Numeric or temporal
Is Not	String, Numeric or temporal
Is Similar To	String
Is Like	String

Operator	Property Type
Contains	String
Does Not Contain	String
Starts With	String
Does Not Start With	String
Ends With	String
Does Not Start With	String
Matches <i>regular expression</i>	String

Conditions

Condition operators act on expressions that evaluate to True or False and on Boolean properties in entity queries.

Operator	Condition
&&	And
	Or
Is Not	Not

Entity Query Examples

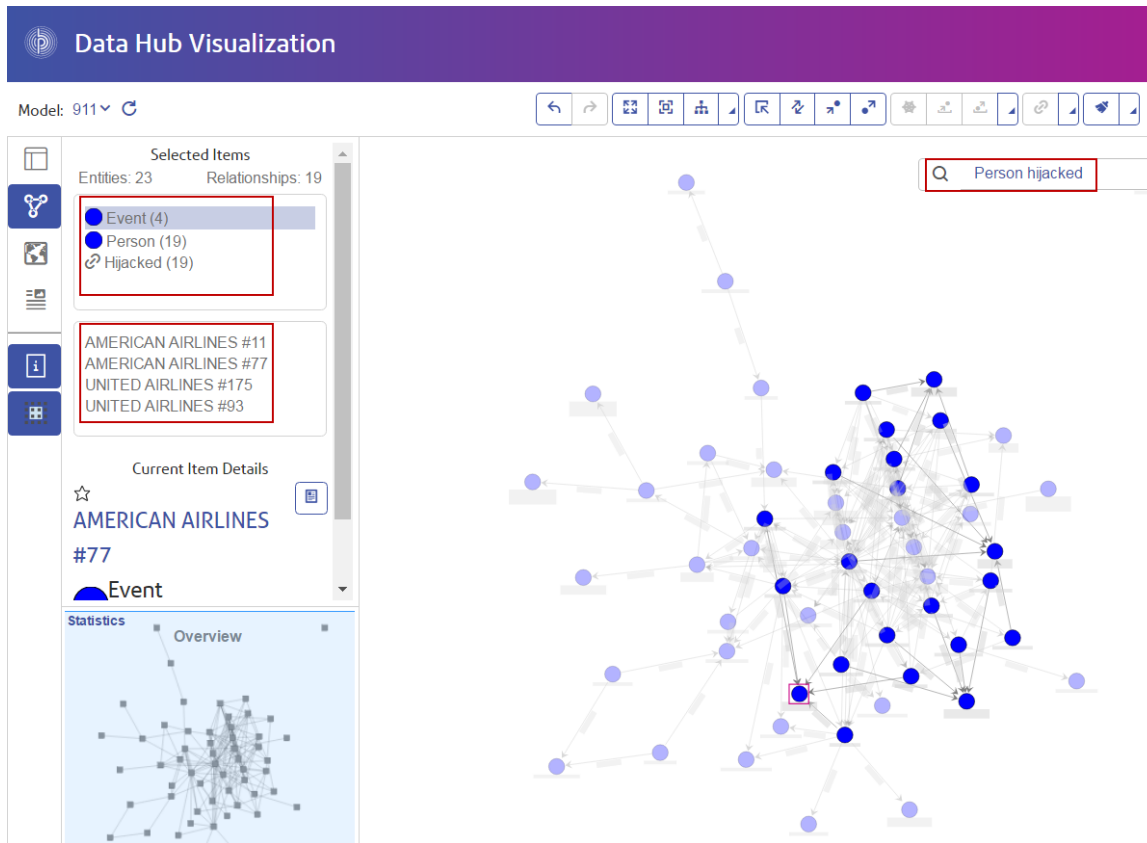
The following list includes possible entity queries into a database of terror information.

- Person Nationality Contains Yemen
- Person Name Starts With Ahm
- Person Name Starts With Osama && Name Ends With den
- Person Name Starts With Osa || Event Contains 911
- Person Osama bin Laden
- Osama bin Laden

Relationships

Relationship Queries

Relationship queries must include an entity prior to the relationship part of the query. For example, if you were looking at a model of September 11, 2001, terrorism data, you could create a query that says, "Nationality contains Saud hijacked". This query would return plane hijackers whose nationality is Saudi Arabia. More simply, if you entered a query that said, "person hijacked", the query would return all of the hijackers in the database. Incidentally, as shown below, it would also return the four planes as Event entities if your database included that information.



Relationship Patterns

The basic pattern for a relationship query follows these rules, which yield the same result: all connections to the selected relationship.

Entity Relationship

*Entity Relationship **

Relationship Query Examples

The following list shows possible entity and relationship queries into a model of terrorism data.

- Osama bin Laden Meeting
- Person Nationality Contains Yemen Attended
- Event Hijacked
- Place Lived
- Date Contains 2001 Hijacked

Browsing Results

As you enter each step of the query into the search bar, Data Hub Visualization will dynamically return type-ahead suggestions based on the model's metadata. After you enter the original query that yields results, you can then conduct subsequent, iterative browsing that is driven off of those initial results.

The following table depicts a query of terrorism data from beginning to end and are looking for a 9/11 hijacker whose name we know begins with "Ah" and ends with "i".

Criteria	Description
Person	<p>We know that the people in this database are represented by an entity type of "Person".</p> <pre>Person</pre>
Name	<p>We know that "Person" entities have properties called "Names".</p> <pre>Person Name</pre>
Starts With	<p>We know the first two letters of the hijacker's name.</p> <pre>Person Name Starts With</pre>
Ah	<p>We know those two letters are "Ah".</p> <pre>Person Name Starts With Ah</pre>
&&	<p>We need to use the "and" condition to search both first and last name information.</p> <pre>Person Name Starts With Ah &&</pre>
Name	<p>Again, we know that "Person" entities have properties called "Names."</p> <pre>Person Name Starts With Ah && Name</pre>
Ends With	<p>We know the last letter of the hijacker's name.</p> <pre>Person Name Starts With Ah && Name Ends With</pre>


Criteria	Description
i.	<p>We know that the letter is "i".</p> <div style="border: 1px solid gray; padding: 5px; background-color: #f0f0f0;"> <p>Person Name Starts With Ah && Name Ends With i</p> </div>

This image shows the query with all the steps in place. It returned four results, shown in the red box. Click a name to see its **Current Item Details**.

The screenshot displays the Data Hub Visualization interface. At the top, the title is "Data Hub Visualization" with a user profile for "admin". Below the title bar, there is a toolbar with various icons and a search bar containing the query "Person Name Starts With Ah && Name Ends With i". The main area is divided into three sections:

- Selected Items:** Shows 4 entities and 0 relationships. A list of names is displayed, with the first four items highlighted in a red box: Ahmed al-Darbi, Ahmed al-Ghamdi, Ahmed al-Haznawi, and Ahmed al-Nami.
- Current Item Details:** Shows details for "Ahmed al-Ghamdi", including his degree (14 - 4 - 10), an associate (Ahmed al-Ghamdi), and geographic coordinates (Latitude: 37.09024, Longitude: -95.71289).
- Network Graph:** A large, complex network graph with many nodes and edges. The node "Mohamed Atta" is prominently displayed in the center.

Filters Tool

The Canvas provides a Filters tool  on its toolbar that you can use to filter entities and relationships that appear on the canvas. The Filters tool enables you to filter entities by type, relationships by label, and both entities and relationships by property values.

To apply filters, you specify entity types or relationship labels that will be visible on the canvas. You may then optionally choose entity and relationship properties and specify allowed values for the property. The options for specifying allowed values for a property depend on the property type (Boolean, string, numeric, or temporal).


By default, orphaned entities are visible on the canvas. Optionally, you may choose to hide orphaned entities. Orphaned entities are those that have no relationships to other entities.

By default, entities with undefined property values (property not used, null or empty) are visible on the canvas. Optionally, when you filter by a property, you can choose to hide entities or relationships with an undefined property value.

Filtering Entities and Relationships on the Canvas

Use the Filters tool to filter entities and relationships visible on the Canvas. You can filter by entity type, by relationship label, and both entities and relationships by their property values.

As you configure entity and relationship filters with the Filters tool, the entities and relationships displayed on the canvas are hidden or displayed. As you make changes to the Filters options, an infobox pops up to indicate how many entities and relationships are hidden or displayed by each change.

1. On the Canvas toolbar, click the Filters button  to open the Filters tool.
2. Configure **Entity Filters**:
 - a) Optional: To hide entities orphaned by filter options, check the **Hide orphaned entities** check box.

Tip: Orphaned entities lack relationships to other entities.
 - b) In the entity types list, check the check boxes for entity types that you want to display or uncheck check boxes to hide entities. You can check or uncheck **Show** to check or uncheck all entity types in the list.
 - c) To filter entities by a property value, first select a property in the entity property list, then configure the options that appear to define allowed values for the entity property.




Entity property options depend on the property type (Boolean, string, numeric, or temporal). For more information, see [Property operators](#) on page 42.

Leave the entity property set to **--All--** if you do not want to filter entities by property values.
 - d) Optional: You can choose to hide entities that have an undefined property value (property not used, null or empty) by checking the **Hide entities with undefined values** check box.
3. Configure **Relationship Filters**.
 - a) In the relationship labels list, check the check boxes for relationship labels that you want to display or uncheck check boxes to hide relationships. You can check or uncheck **Show** to check or uncheck all relationship labels in the list.

Note: Only relationships labels between entities currently visible on the Canvas appear in this list. These are entities corresponding to the types that you checked in the entities list in step [2.b](#) on page 39.
 - b) To filter relationships by a property value, first select a property in the relationship property list, then configure options that appear to define allowed values for the relationship property.


Relationship property options depend on the property type (Boolean, string, numeric, or temporal). For more information, see [Property operators](#) on page 42.

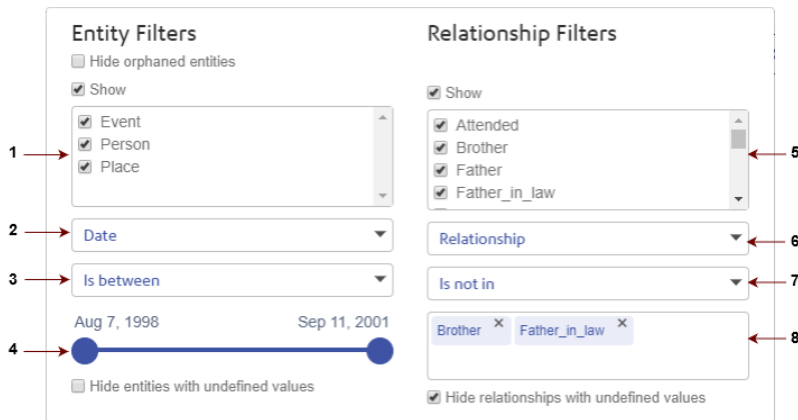
Leave the relationship property set to **--All--** if you do not want to filter relationships by property values.

- c) Optional: You can choose to hide relationships with an undefined property value (property not used, null or empty) by checking the **Hide relationships with undefined values** check box.
4. To close the Filters options, click the Filters button  on the toolbar.
- The Clear Filters button  appears next to the Filters button whenever filters are applied to entities or relationships on the canvas.
5. To clear filters and display all of the entities and relationships on the canvas, click the Clear Filters button .

Filters persist between sessions after you sign out. You can repeat this procedure at any time in a session to change or clear the filters.

Filter Options

This section describes entity and relationship filter options. You can click the Filters button  on the toolbar to configure both **Entity Filters** and **Relationship Filters**.




Entity filters elements

1. Entity types 2. Entity property 3. Property operator 4. Value bar or value box

Relationship filters elements

5. Relationship labels 6. Relationship property 7. Property operator 8. Value bar or value box

As you make changes to these options, an infobox displays how many entities and relationships were filtered by a change. To clear all existing filters, click the Clear Filters  button, which appears next to the Filters button whenever filters are applied to entities and relationships on the Canvas.

Hide orphaned entities

Check this option to hide entities when they do not have a visible connection to another entity.

Entity types or relationship labels

Check the check box next to entity types or relationship labels that you want to show on the canvas. Uncheck a check box next to any entity type or relationship label that you want to hide from the canvas. For either entity types or relationship labels, you can check or uncheck the check box next to **Show** to check or uncheck all types or labels in the list.

Entity or relationship property

In the entity or relationship property list, click the property on which you want to filter. Selecting a property in either list displays operators that depend on the property type (Boolean, string, numeric, or temporal).

- For a numeric or date property type, the property selection displays numeric operators and the value bar.
- For a Boolean property type, the property selection displays Boolean options.
- For other property types, the property selection displays string operators and the value box.

To not filter by property (allow all property values), click **--All--** in either list.

Property operators

Property operators display entities or relationships based on the value of a property. The the operators listed in this box depend on the property type:

- [Boolean Property Filter Options](#) on page 42
- [Numeric and Temporal Property Filter Options](#) on page 42
- [String Property Filter Options](#) on page 42

Value bar

This selection bar is displayed when you select a numeric or temporal property in the properties list box. Move sliders on the value or range bar to specify a value or range of values referred to by an operator. There is a single slider when an operator (such as **Equals**) refers to a single value or two sliders when an operator (such as **Is between**) refers to a range of values. You can move a slider only between property values that already exist in the model data.

Value box

This text box is displayed when you select a text property (any except numeric, temporal, or Boolean) in the entity or relationship property box. You will be prompted to enter values based on the selected property and operator:

When prompted to	Do this
Enter a value to filter on	Click in the box and type a string value.
Enter one or more values	Click in the box, type a string value, and press Enter to include it. To enter another value, click the empty space under previously entered values.
Select one or more values	Click in the box and select a value in the list from the model data. To select another value, click the empty space under previously selected values.

Hide entities with undefined values

Check this check box to hide entities with undefined values (not used, null or empty) for the selected property. This option is available only after you select a property under **Entity Filters**.

Hide relationships with undefined values Check this check box to hide relationships with undefined values (not used, null or empty) for the selected property. This option is available only after you select a property under **Relationship Filters**.

Boolean Property Filter Options

Selecting a Boolean property displays the following options.

Options

True	Displays entities or relationships when the selected property value evaluates as True.
False	Displays entities or relationships when the selected property value evaluates as False.

Numeric and Temporal Property Filter Options

The property operator box lists the operators shown here to filter values for a numeric or temporal property.

Operators

Equals	The property value must be equal to a particular value.
Does not equal	The property value must not be equal to a particular value.
Is between	The property value must be between two values.
Is not between	The property value must be less than the smaller or greater than the larger of two values.
Is greater than	The property value must be greater than a specified value.
Is greater than or equals	The property value must be greater than or equal to a specified value.
Is less than	The property value must be less than a specified value.
Is less than or equals	The property value must be less than or equal to a specified value.

String Property Filter Options

The property operator box lists the operators shown here to filter values for a text property (any property other than numeric, temporal, or Boolean).

Operators

Contains any	Property value must contain one or more of the string values in the list.
Does not contain any	Property value must not contain any of the string values in the list.
Contains all	Property value must contain all of the string values in the list.
Does not contain all	Property value must not contain all of the string values in the list.
Is in	Property value matches a string value in the list. In the value box you can select one or more existing property values to test against.
Is not in	Property value does not match any string value in the list. In the value box you can choose one or more existing property values to test against.





Starts with	Property value starts with a particular string value. In the value box you can type a string.
Does not start with	Property value does not start with a particular string value. In the value box you can type a string.
Matches regular expression	Property value matches a regular expression (regex). The regular expression must match the entire string in the property value.

Entity Profile Report

The Entity Profile Report shows all data associated with a particular entity in one place. This report has a unique URL and as such can be bookmarked or shared.

Information in an Entity Profile Report includes the following:

- The entity label (usually its name)
- The entity type (for example, "person" or "event")
- The total number of relationships connected to the entity
- A list of each property tied to the entity as well as the value for each of those properties. For instance, a property of "Latitude" may have a value of "23.424076."
- A list of each incoming relationship, with the source, relationship label, and target for that relationship. In this list, the target will always be the entity on whom the report is based. For instance, in a model used to identify insurance fraud, this list might have an entry with a source of someone's name, a relationship of "is_patient_of," and a target of "Dr. Smith."
- A list of each outgoing relationship, with the source, relationship label, and target for that relationship. In this list, the source will always be the entity on whom the report is based.

From this screen you can open to the Canvas () an entity and all of its connections or a relationship and all of the connections associated with that relationship. Likewise, you can open to the Map () an entity and all of its connections with latitude/longitude data. You can also undo or redo actions on this page ( .

Edit Item Properties

This feature enables you to add, modify, or delete the values of properties for a selected entity or relationship. If multiple items are selected, these changes will apply to the item that was selected last.

Note: You cannot add or delete properties themselves, only the data within those properties.

Watch Entity

Data Hub Visualization provides a News Feed feature that enables you to identify entities in your models whose activity you would like to track.

Although the News Feed appears on the Dashboard, you establish which entities you want to track on the Canvas.




1. With a model open, select an entity you would like to track. Information for that entity will appear in the **Current Item Details** pane.
2. Click Watch entity ☆ in the **Current Item Details** pane.
You can alternatively right-click selected entities and click **Watch selected entities** on the shortcut menu.
3. To unwatch an entity, click Stop watching entity ★ in the **Current Item Details** pane.
You can also right-click selected entities and click **Unwatch selected entities** on the shortcut menu.

Map

The Map workspace depicts geospatial data (entities with latitude and longitude properties) and places them on a map. The map automatically zooms in and out, based on the locations of the selected entities, to display all of the selected entities at one time. Click any entity on the map to view its **Current Item Details**.

Before you can view your model's latitude and longitude data on the map, you must first configure entity types on [the Metadata tab](#) of the Metadata page and identify which properties contain latitude and longitude data. Otherwise an error message will appear telling you the model has not been configured for mapping.

You can view the map in several ways:

- | | |
|---|---|
| Standard  | Depicts a customary roadmap showing selected entities. |
| Cluster  | Takes closely positioned pushpins and groups them into clusters to make it easier to navigate. |
| Heat  | A graphical representation of data where the individual values contained in a matrix are represented as colors. In a geographic mapping context, heat maps can provide visualization of the relative density of point data. |

You can also view your map from an aerial perspective and import real-time traffic data.

Query Tool


Map provides a query tool that allows you to search a model. This query function supports a natural-language inspired grammar.

The Map provides a query tool similar to that of the Canvas. It supports a natural-language inspired grammar and allows you to search a model and returns results on the map using geospatial data. It also provides a filter that enables you to narrow results based on entity type or relationship label, along with a traditional search field. When you enter your query, the results will load onto the Map. For more information, see [Canvas Query Tool](#).




Table

The Table presents model data in a tabular format, broken out into entities and relationships.


Entities Tab

The Entities tab () provides an entity's name, its Entity ID, and its properties. If you run a centrality algorithm on the Canvas, those values will appear as well. If you click an entity's name, its [Entity Profile Report](#) will appear.


The Entities tab provides the following options on the toolbar at the top of the page:

- Unload selected**  Removes all selected entities from the table but not the model.
- Add new entity**  Creates and adds an entirely new entity to the model, including its type, label, properties, relationships, and a unique key (optional). You can also query for connections.
- Edit selected entity**  Makes changes to the selected entity. You can add (or remove) property data, define relationships, and query for connections.



When you have completed your changes, you can do the following:

- Delete item**  Deletes all selected entities from the model, not just the table.
- Save Options** When you add or edit an entity, you have three Save options:
 - Save—Saves over the existing entity.
 - Save & Copy—Saves over the existing entity and copied as a new entity.
 - Save & New—Saves over the existing entity and then the editor is clear for a new entity to be added.

Relationships Tab

The Relationships tab () displays source and target entities and properties for each relationship. Click any column header to sort that column.

The Relationships tab provides the following options on the toolbar at the top of the page:

- Unload selected**  Removes all selected relationships from the table but not the model.
- Delete item**  Deletes all selected relationships from the model, not just the table.

Filter & Query Tool

The Table provides a query tool similar to that of the Canvas. It supports a natural-language inspired grammar and allows you to search a model and returns matching relationships or entities in a table format. It also provides a filter that enables you to narrow results based on entity type or relationship

label, along with a traditional search field. When you enter your query, the results will load onto the table. For more information, see [Canvas Query Tool](#).

Visual Query Builder

The Visual Query Builder is a workspace in Data Hub Visualization that performs simple and complex query functions on existing models in a visual manner.

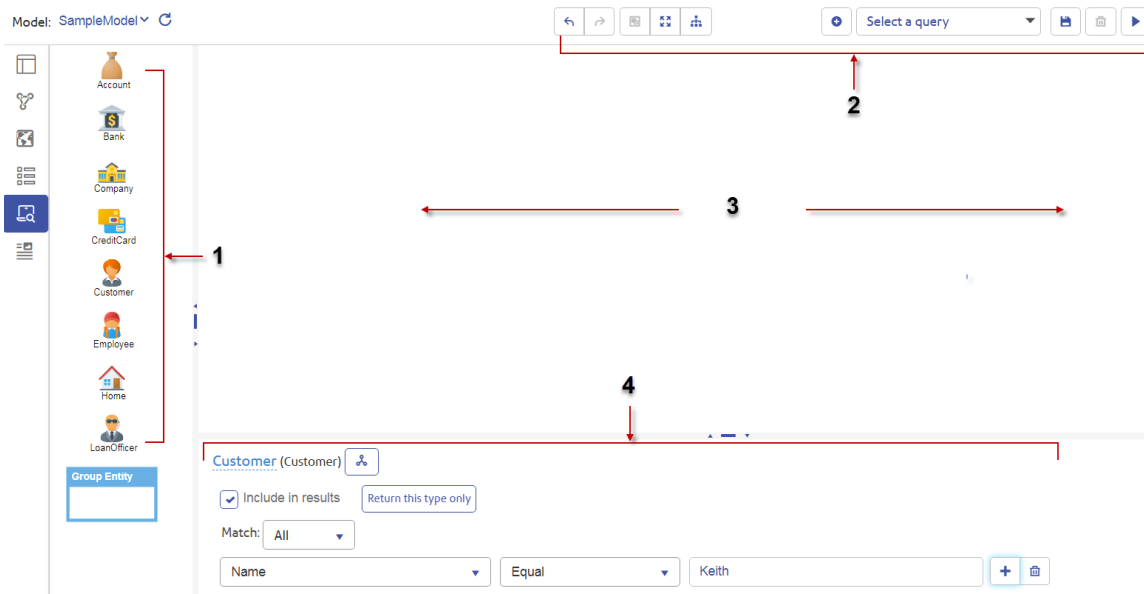
You begin by dragging a single entity type to the workspace. You can then use the Assistant to select from a list of relationship types and entity types that are connected to that initial entity type and build your query from there. You can also drag additional entity types to the workspace and draw connections between them to add relationship types to the query. To do this, you need be familiar with connections and types of relationships between entities in your data model. You can narrow the results of your query by applying properties to entity types or by applying relationship counts to relationship types. Both of these criteria act as constraints.

Although you build and run a query in Visual Query Builder, query results are loaded to the Data Hub Visualization Canvas. You view the results of a query by opening the Canvas workspace.

Layout

The Visual Query Builder is comprised of the following sections:

Entity Panel	Vertical list of icons on the left side of the screen that represent the entity types in your model.
Toolbar	Buttons along the top of the screen.
Expression Editor	Bottom portion of the screen where you can add expressions to entity types, which define constraints against property values, add counts to relationship types, which also act as constraints, view query summaries, view query validation, change labels, and access the Assistant.
Workspace	Central portion of the screen where you visually build queries.



1. Entity Panel 2. Toolbar 3. Workspace 4. Expression Editor

Toolbar


The Visual Query Toolbar contains the following buttons:

- Undo/Redo** (↶ ↷) Undo or redo the last action.
- Group selected entity types** (📁) Group two or more entity types with no connections.
- Fit content** (📐) Fit the content of the workspace within the viewable area.
- Apply layout** (📏) Arrange or rearrange the query content.
- New query** (🔄) Clear the canvas and start a new query definition.
- Select a query** (📄 Select a query) Choose from a list of queries defined for this model.
- Save query** (💾) Save the query for future use in Visual Query Builder with this model.
 - Note:** Queries saved in the Spectrum™ Technology Platform 12.0 SP2 will be available for use only in version 12.0 SP2.
- Delete query** (🗑️) Delete the query from the Spectrum™ Technology Platform server.
- Run query** (▶️) Execute the query.

Note: You must switch to the Visualization Canvas, Map, or Table to view the results of the query.


Using the Visual Query Builder

Queries are generally built using entity types joined by relationship types, and can be qualified by property values and relationship counts.

You can create the simplest query by dragging a single entity type onto the canvas. For example, if you were using data from a financial institution and your entity types included banks, customers, and employees, you could simply drag out any one of those entities and click the Run button  to perform a query that returns all entities of that type.

Using Add Connected Items

One of the most helpful tools within the Visual Query Builder is the **Add connected items** dialog box, which identifies all entity and relationship types that are connected to a selected entity type. This task shows how to create a simple query with two entity types:

1. Drag an entity type from the Entity Panel onto the workspace. This displays options in the Expression Editor for the entity.
2. You can add connected items using one of two methods:
 - Right-click the entity type and select **Add connected items**.
 - Select the entity type you just dragged out and click the Add connected items icon  in the Expression Editor.

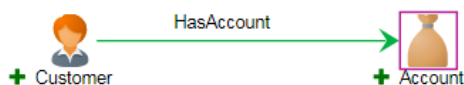
This opens the **Add connected items** dialog box.

3. Choose to filter by relationships to the entity or by entity types connected to the entity.

Click	To
Relationships	Select from relationship types that connect the selected entity, then select from connected entity types.
Entities	Select from connected entity types, then select from relationship types that connect the selected entity.

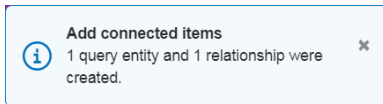
4. Click **Add**, then **Close** to add the specified entities to the workspace. To close the dialog box without making any changes, click anywhere on the workspace.

Selected entity types connected by selected relationship are added to the workspace types on the workspace:



The green plus signs to the left of each entity type signify that any entity type and relationship label matching this query will be returned in the results.

Data Visualization displays an infobox summarizing what was added to your query:



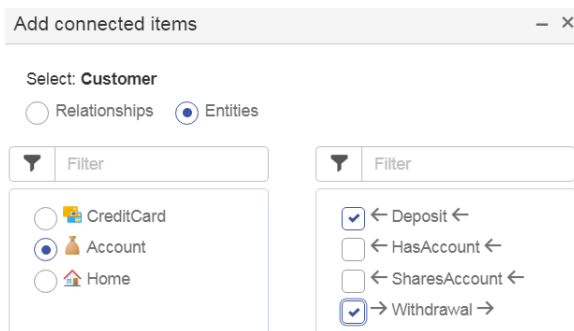
5. Optional: If you want the results to include just one entity type, select the entity type in the workspace and click **Return this type only** in the Expression Editor.
6. If you want to include some but not all of the entity types, select the ones you want included and check **Include in results**.
7. Run the query.
8. You can view these results on the Canvas, Map, Table workspace.

Using Multiple Relationships

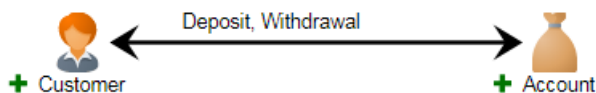
Just as two entities in real life can be connected in multiple ways, so can two entities in a database. For example, two people can be on a checking account together and on a mortgage together. Likewise, there are situations in real life where a relationship of one kind precludes a relationship of different kind. An example of this might be if someone has a relationship of "sibling" to one person, he or she can't also have a relationship of "married" to that same person.

Visual Query Builder has specific ways of selecting and displaying "and" and "or" relationships. When you use the **Add connected items** dialog box to select relationships, if you select more than one relationship at the same time, Visual Query Builder interprets this as an "or" relationship. That means it will look for entities that have *one relationship or the other*.

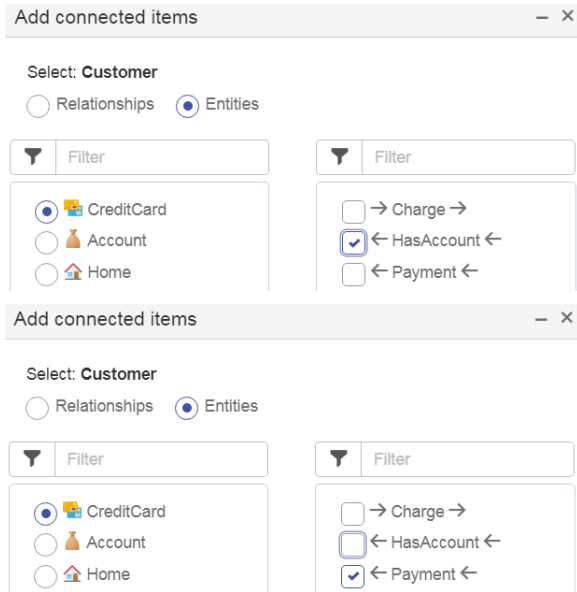
The image shows an example of an "or" relationship in the **Add connected items** dialog box:



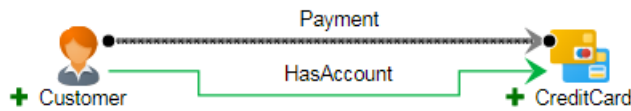
It adds entities who make deposits to or withdrawals from their account to the workspace.



Alternatively, if you select more than one relationship at different times, in separate steps, Visual Query Builder interprets this as an "and" relationship. Here is an example of an "and" relationship in the **Add connected items** dialog box.



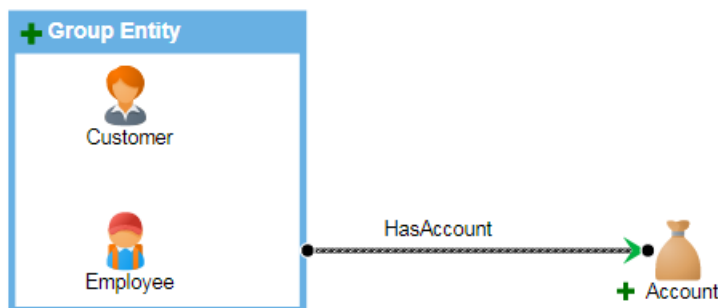
That means it will look for entities that have **both** relationships, not just one or the other. It would return entities who have credit card accounts **and** made payments on those accounts.



Using Group Entity

The Group Entity feature enables you to run queries with two or more entity types that are not connected but are grouped together to simplify the query process. The relationship label connected to the group can apply to any or all of the entity types in the group. The data for all matching entities and relationships will be returned in the results.

For example, let's say you are looking into account data for customers and employees. The query below would return data for any customers who have accounts as well as any employees who have accounts.



1. Select the entity types you want to group on the workspace.
2. Right-click and select **Group selected items**.

3. Continue building your query as usual.

Constructing the query in this manner allows you to run one query instead of two for the same results.

Adding Constraints Using Expressions

When entities have defined properties the values of those properties often vary by entity. You can use these to create more focused queries based on different values.

For example, you could have an entity type of "person" and properties for that entity type could include gender, age, or parental status. These properties would make it possible for you to run queries that look only for males, or for people in a certain age group, or for people who have at least one child. There are endless reasons for wanting to run a targeted query, and incorporating expressions that are based on properties into your query allows you to do that.

Example: Use an expression to add a constraint

Let us imagine a financial institution that wants to send a limit-increase offer to current credit card customers. It wants to omit customers who do not have a strong payment history. Provided data to support the query, Visual Query Builder makes it possible for them to exclude certain credit card customers from the query results by adding expressions—called constraints—to a query.

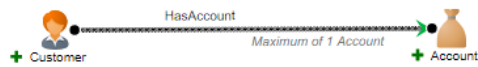
1. Drag a Customer entity type onto the workspace, then drag a CreditCard entity type onto the workspace.
2. Draw a connection between the two entity types and select the "HasAccount" relationship type.
3. Draw a second connection between the two entity types and select the "Payment" relationship type.
4. Click this relationship type between the two entity types.
5. In the Expression Editor, choose the "amount" property and the **Greater than** operator, enter "1" for the value, and click the plus sign at the end of the row.

When you run the query it will return customers who have credit cards and have made payments of more than \$1.

Note: Notice after running a query that a green check now appears next to the relationship type. This denotes that an expression has been tied to the relationship type in the query.

Adding Constraints Using Relationship Counts

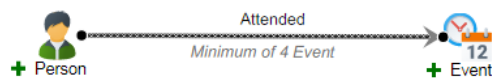
Just as queries can use expressions that incorporate property values to limit entity type results, they can also use a **Relationship count** to limit relationship type results. This field sets search criteria based on a quantity (exact, minimum, maximum, range, and so on) tied to the relationship. For example, if you are a financial institution, you might use this feature to identify customers with whom you could increase your business. You could extract customers who have only one account and then determine what other accounts would be appropriate to offer them. The query might look similar to this:



[HasAccount](#) (HasAccount)

Relationship count:

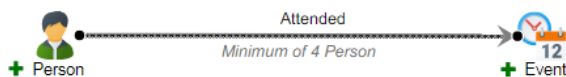
Additionally, you can set this criteria to apply to either end of the connection. Let's say that for the past two years your company has been offering customers free monthly meetings that provide information about retirement accounts. The first month the topic was Simple IRAs, the next month it was Roth IRAs, and so on. For marketing purposes you want to know which meetings had the highest attendance. You also want to know which customers are attending multiple events to determine if you should be advertising this offering to a certain segment of your customer base. This query will return all person and event entities where the person has attended at least four events.



[Attended](#) (Attended)

Relationship count:

This query will return all person and event entities where an event was attended by at least four persons.

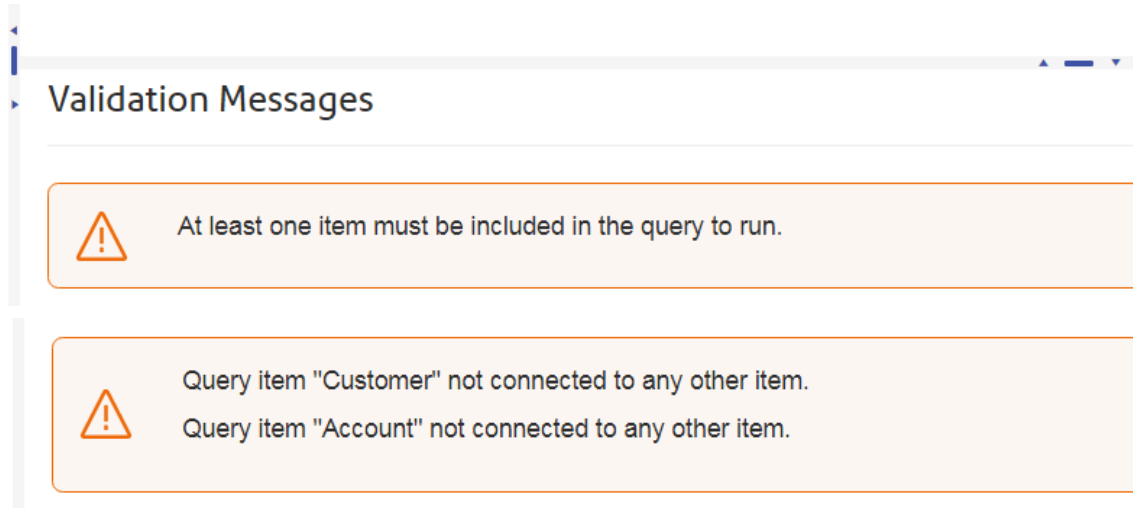


[Attended](#) (Attended)

Relationship count:

Viewing Validation Messages

If you have not yet placed entities on the workspace or you have not connected entities, you will see validation messages similar to the ones shown here in the same space where the Expression Editor normally appears.



If you are not sure which part of the query caused the error, hover over the error message to select the element causing the problem in the workspace.

Note: If Visualization returns a validation message while you are running a query it is possible that the query is not supported in the current version of Spectrum™ Technology Platform. The error message should provide appropriate information.

Viewing the Query Summary

The Query Summary appears in the same space as the Expression Editor when nothing on the workspace is selected. It provides an overview of the query, including any properties and relationship counts.

Query Conditions and Counts

Query Item				
▼	CreditCard → Payment → Customer			
	Match	Property Name	Operator	Value
	All	amount	Greater than	1

Entity Browser

The **Entity Browser** is a workspace in **Data Hub Visualization** that lists specified types of entities and their property values.

You begin by selecting entities that you want to view in the Entity Browser. You can then create entity property expressions to display entities that match any or all grouped expressions. Additional options allow you to initially order results by values of a property and to include or exclude properties in the results. After you run a query, you can reorder the results by any property displayed in the results table. The Entity Browser allows you to save and reuse queries and to export the results of a query to a file.

Entity Browser Layout

The Entity Browser defines queries to display entities. A query selects entities and includes expressions to display entities based on property values.

The Entity Browser has several different components that you can use to query models and select entities. Options order the results and specify entity properties returned by a query, while input fields define property value criteria in expressions. Saved queries can be used in the Canvas, Map, and Table workspaces.

The screenshot shows the Entity Browser interface with the following components highlighted by numbered callouts:






- 1. Entity types:** A dropdown menu showing 'Event' and 'Person'.
- 2. Property expressions:** A section with two groups of filters. The first group is 'That match ALL of the following' and the second is 'That match ANY of the following'. Each group contains two filters: 'Entity label' with 'Contains' and 'Starts with' operators, and 'Date' with 'Is less than' operator.
- 3. Saved queries:** A dropdown menu labeled 'Select a query'.
- 4. Order selection:** A dropdown menu labeled 'Order by' with 'Entity label' selected.
- 5. Properties selection:** A list of properties with checkboxes, including 'Date', 'Event', 'Latitude', 'Location', 'Longitude', 'Type', etc.
- 6. Input tab:** A tab labeled 'Input'.
- 7. Results table:** A table displaying the results of the query.

Entity	Date	Event	Latitude	Location	Longitude	Type
1998 United States embassy bombing Dar es Salaam	Aug 7, 1998	1998 United States embassy bo...	-6.822921	Dar es Salaam, Tanzania	39.26966	Bombing
1998 United States embassy bombing Nairobi	Aug 7, 1998	1998 United States embassy bo...	-1.2833333	Nairobi, Kenya	36.816666	Bombing
AMERICAN AIRLINES #11	Sep 11, 2001	AMERICAN AIRLINES #11	40.711494	WTC N., New York	-74.01222	Hijacking

Showing 4 entities.

1. Entity types 2. Property expressions 3. Saved queries 4. Order selection 5. Properties selection 6. Input tab 7. Results table

Figure 1: Entity Browser components

Entity types	Lists entity types returned by a query. The Run query  button executes a query in Entity Browser. The Export button  saves query results to a comma-separated values (CSV) file.
Property expressions	Match ALL or Match ANY operators group expressions in a query. The expressions specify entities returned by a query based on property values. An expression evaluates property values either based on a Literal value specified in the expression or based on a Field value that must be entered before running a query.
Saved queries	Lists and selects saved queries. The New query button  clears an existing query to create a new query. The Save query button  saves the query currently defined in the workspace. The Delete query button  deletes a query from the list. Saved queries are available in the Canvas , Map , and Table workspaces.
Order selection	This option orders entities by column values in ascending or descending order. You can also click a column heading on the results table to reorder entities by values in the column.
Properties selection	Checking or unchecking the check boxes for this option determine which entity properties are displayed by columns in the results table.
Input tab	This tab displays fields in query expressions. Field values must be entered before running a query. Values entered on this tab are saved as default values for fields when you save a query.
Results table	Specifies results of a query. The first column of the table lists entity labels while the remaining columns show property values for each entity. A cell is empty when an entity does not have a value for a property. The number of entities in the table is displayed at the bottom of the table.

Using the Entity Browser



Queries are built by selecting entity types, creating property expressions, specifying a default order, and selecting properties that you want to see in the results.


You can create the simplest query by selecting entity types. You can add property expressions to specify which entities are displayed based on property values. Expressions are grouped by the Match ALL or Match ANY operators. Value criteria in the expressions may be specified as literal values in the expressions themselves or by field values entered before running a query.




For any query, you can specify which properties are displayed in the results and how entities are ordered in the results. After you test a query, you can save it to reuse it or to run in the Canvas, Map, and Table workspaces. Values entered in fields before you save a query are saved as default values for the fields. After you execute a query in the Entity Browser, you can reorder the results by values in any column and export the results to a comma-separated values (CSV) file.

Creating queries in Entity Browser

The **Entity Browser** allows you to quickly browse filtered subsets of entities in your model and view their property values.

1. On the **Data Hub Visualization** workspaces bar, click the Entity Browser button . This opens the **Entity Browser** workspace.
2. On the toolbar, click a Data Hub model in the **Model** box to select a model.
3. In **Entity types**, click the drop-down button  to expand the list of entity types.

Tip: Enter any portion of an entity type name in the filter box  to narrow the lists to names that include a string.

4. Select the check box  for each entity type that you want to browse. You can check or uncheck the check box next to the filter box  to check or uncheck all entity types in the list.
5. Click the Run query button  to display the entities specified by a query. Entities are displayed in a table. You can click any column to reorder the table by values shown in the column. You can scroll down to view additional entities.

Completing this task displays entities for all of the selected entity types. By default, entities are ordered by the labels in the **Entity** column of the table.


You can now add property expressions to query entities by property values, change the default order or reorder entities, and select properties displayed in the query results table.


Adding expressions to a query

You can add expressions to the Entity Browser to only browse entities that meet specified criteria for their property values.

1. To browse entities by property values, first choose how to combine expressions.

If you choose	Entity property values must match
That match ALL of the following	All of the expressions, as in series of logical <i>and</i> statements.
That match ANY of the following	At least one of the expressions, as in a series of logical <i>or</i> statements.

You can click the Clear expressions button  on the initial Match ALL or Match ANY operator to clear all expressions under it.


2. To sequentially add expressions within the Match ALL or Match ANY operator, click the Add property expression button .
3. Select the property that you want to evaluate.


- Select an expression operator with which to evaluate the property.




For more information, see [Entity Browser property expression operators](#) on page 60.

- Select whether to evaluate the property based on a **Literal** value or a **Field** value:

Select	Then
Field	<p>Enter a name for the field, then on the Input tab enter a value for the field.</p> <p>The expression will use the field entry to evaluate the property.</p> <p>Note: When you save a query, values entered on the Input tab are saved as the default values for the fields.</p>
Literal	<p>Enter a literal value.</p> <p>The expression will use this value to evaluate the property.</p>

- To add an expression immediately after any existing expression, click the Insert property expression button  next to an expression, and repeat steps [3](#) on page 56 through [5](#) on page 57 to add the expression.

You can click the Delete expression button  to remove an expression along with any subordinate expressions. Alternatively, uncheck the check box next to an expression to exclude it without removing it. You can then always check the check box to include it again in a subsequent query.

- Optional: To add a subordinate expression, click the Insert subordinate expression button  next to an expression, then choose whether to Match ALL or Match ANY of the following expressions, and repeat steps [2](#) on page 56 through [6](#) on page 57.
The outcome of a parent expression depends on the evaluation of the subordinate expressions.
- Optional: To reorder expressions, drag the Drag expression handle  on an expression to a different position.
- Before running the query, enter values in any fields displayed on the **Input** tab.
- Click the Run query button  to display the filtered list of entities.

Completing this task displays entities in the **Entity Browser** that fulfill the combined expressions.

Changing the default order to display entities in the Entity Browser




You can change the default order to display entities in the **Entity Browser** in the **Data Hub Visualization** application.

Choose to sort in ascending or descending order by values in any column. Sort by text (A to Z or Z to A), numbers (smallest to largest or largest to smallest), and dates and times (oldest to newest and newest to oldest) in a column.

This procedure sorts entity data before it is loaded from the server. Ordering data on the server before loading it into the **Entity Browser** can be time-consuming for large models, so you may choose to avoid using a default sort to load entity data more quickly.

1. On the **Options** tab, click in the **Order by** box to open the list of possible sorts and click one of the following:

Click	To
Property name	Order by values in the corresponding column of the table.
Entity label	Order entities by entity labels displayed in the Entity column of the table. This is default behavior if you do not make any selection in this list.
None	Avoid sorting entities on the server. Select this option to load entity data more rapidly for larger models.


2. Click the Ascending button  or the Descending button  to order by ascending or descending order of values in a selected column.
3. Click the Run query button  to display entities in the specified order.

After running a query, you may click any column heading to sort entities in the table by a properties values in either ascending or descending order.

Select properties to display in the Entity Browser

You can choose which entity properties are displayed in columns of the **Entity Browser** in the **Data Hub Visualization** application.

By default, the **Entity Browser** displays columns for all properties of all entity types currently displayed in the browser. This procedure enables you to hide columns and simplify the table.


1. On the **Options** tab, check or uncheck check boxes in the **Included properties** box to add or remove entity properties displayed as columns in the **Entity Browser**.
You can check or uncheck the **Include properties** check box itself to check or uncheck all listed properties.
2. Click the Run query button .

Completing this procedure reloads data in the table with only the columns corresponding to the selected properties. The default sort specified in the **Order by** box will not be applied if it is selected to sort on a hidden column.

Export entity data from Entity Browser

You can export entities and property values returned by a query to a list of comma separated values saved in a CSV file. This procedure exports all entity data returned by a query up to one million rows.

Before you perform this procedure, run a query to display entities and their property values in the entity browser table.

1. Next to the **Entity types** list, click the Export button .
2. Depending on the browser you are using, click the option to **Save** the file.
This creates a file named `model_name-results.csv` located in the default download folder used by the browser.


After you complete this procedure, you can import the data into other applications and open the output file in a text editor or spreadsheet application, such as Microsoft Excel.


Save a query in Entity Browser

This procedure describes how to save queries in the **Entity Browser** workspace in the **Data Hub Visualization** application. Queries saved in **Entity Browser** may be run in the **Canvas, Map, Table,** and **Entity Browser** workspaces.

Before you perform this procedure, at least one entity type must be specified in the **Entity types** box.

A saved query retains everything except data in the results table.


1. Optional: Before you save a query, you can enter default values for any fields displayed on the **Input** tab.
2. On the toolbar, click the Save query button .
3. In the **Save Query** dialog box, type a **Name** for the query.
If you are editing a previously saved query, you can leave the previously saved name to save changes to the existing query. Type a new name to add a new query to the list of saved queries.
4. Click the **Save** button.
For a new query, this adds the name of the query to the **Select a query** box in the toolbar. For an existing query, you are prompted whether to **Save and Replace** the existing query. Click **Cancel** if you do not want to replace an existing query, and type a new name before you click **Save** again.

To clear a query from the **Entity Browser** workspace and create a new query, click the New query button  on the menu bar.

Delete a saved query in Entity Browser

This procedure describes how to permanently delete a query that was created in **Entity Browser**.

1. On the **Data Hub Visualization** workspaces bar, click the Entity Browser button .

2. In the **Select a query** box on the menu bar, select the query that you want to delete.
3. Click the Delete query button .

Create a new query in Entity Browser

This procedure describes how to clear an existing query from the **Entity Browser** workspace to create a new query.

- On the toolbar, click the New query button .
- You are prompted whether to continue if there are any unsaved changes to an existing query.

Entity Browser property expression operators

Use operators in **Entity Browser** property expressions to define conditions on properties. Available operators depend on the property type that you initially select for an expression.

- [String Property Operators](#) on page 60
- [Numeric or Temporal Property Operators](#) on page 60
- [Logical Property Operators](#) on page 61

String Property Operators

Exists	An entity must have the property.
Does not exist	An entity must not have the property.
Equals	The property value must match the specified string.
Does not equal	The property value must not match the specified string.
Is in	The property value must be in the specified string.
Is not in	The property value must not be in the specified string.
Contains	The property value must contain the specified string.
Does not contain	The property value must not contain the specified string.
Starts with	The property value must start with the specified string.
Does not start with	The property value must not start with the specified string.
Ends with	The property value must end with the specified string.
Does not end with	The property value must not end with the specified string.
Matches regular expression	The property value must match a regular expression (regex). The regular expression must match the entire string in the property value.

Numeric or Temporal Property Operators

Exists	An entity must have the property.
Does not exist	An entity must not have the property.
Equals	The property value must be equal to a particular value.
Does not equal	The property value must not be equal to a particular value.

Is between	The property value must be between two values.
Is not between	The property value must be less than the smaller or greater than the larger of two values.
Is greater than	The property value must be greater than a specified value.
Is greater than or equals	The property value must be greater than or equal to a specified value.
Is less than	The property value must be less than a specified value.
Is less than or equals	The property value must be less than or equal to a specified value.

Logical Property Operators



True	Displays entities or relationships when the property value evaluates as true.
False	Displays entities or relationships when the property value evaluates as false.

Metadata

The Metadata page provides detailed information about the entities and relationships in your model, including the following:

- Total number of entities
- Total number of relationships
- Number of each entity type
- Number of each relationship label
- Number and data type of each property type broken down by entities and relationships
- Index for entity properties



When you click an entity type or relationship label, additional information appears that details incoming and outgoing counts as well as the number of properties for that particular element.

By default, the Metadata page opens to the Model view  and shows metadata for all the elements in your model. However, if your model is not fully loaded, such as if you removed some of its elements on the Canvas or the Map, you can use the Canvas button  to view the metadata only for the remaining elements.

Additionally, the Metadata page enables you to change the look of entities and relationships by configuring metadata appearance.

Changing Metadata Appearance

Changing the appearance of metadata gives you further control over your data by making it more identifiable at a glance and by making certain entities and relationships stand out by using different graphics, shapes, colors, font size, and so forth.


1. In **Data Hub Visualization**, click the Metadata button  on the toolbar.
2. On the Metadata canvas, click the entity type or relationship label that you want to edit.
3. Click the Edit button .
4. Edit options for the selected entity or relationship.

To edit	See
Entity options	Entity Decoration on page 62
Relationship options	Relationship Decoration on page 63

=

5. Click **OK** to save your changes.

Entity Decoration

Entity decorations include style and label settings. These options are displayed by clicking an entity on the Metadata canvas, and then clicking the Edit button .

Style Settings

Entity graphic Allows you to change the default shape or graphic for a selected entity type by choosing either from the **Entity graphic** drop-down library or from a custom `.svg`, `.png`, or `.jpg` image from an external URI.

Check **Use custom image property** to override the default decoration with a custom image for a specific entity. If the property is not defined it will use the default decoration.

Note: `.svg` images are not supported in the Relationship Analysis Client.

Fill color If you selected an entity shape, allows you to change the interior color of the shape.

Border color If you selected an entity shape, allows you to change the border color of the shape.

Label Settings


Override default label Allows you to designate a different property as the label for the entity type. Choose from the drop-down list.

Wrap Forces labels exceeding maximum width to break onto multiple lines.

Trim Trims labels exceeding maximum width.

Font size	Sets the font size of the entity type's label.
Maximum width	Sets the maximum width allowable, in pixels, of the entity type label.
Color	Sets the color of the entity type label.

Relationship Decoration

Relationship decorations include line and label settings. These options are displayed by clicking a relationship on the Metadata canvas, and then clicking the Edit button .

Line Settings




Line type	Sets the appearance of the relationship label's line.
Thickness	Sets the width of the relationship label's line.
Color	Sets the color of the relationship label's line.

Label Settings

Font size	Sets the size of the relationship label.
Wrap	Forces labels exceeding maximum width to break onto multiple lines.
Trim	Trims labels exceeding maximum width.
Color	Sets the color of the relationship label.
Maximum width	Sets the maximum width allowable, in pixels, of the relationship label.
Distance from relationship	Sets how close to or far from the labels are to the relationships.

Change Property Order

Follow these steps to change the order in which properties appear in Data Hub Visualization and Data Hub Browser.

1. In **Data Hub Visualization**, click the Metadata button  on the workspaces bar.
2. On the Metadata canvas, click an entity type or relationship label.
3. Click the Reorder properties  button.
4. Drag properties up and down within the **Properties** list to set the preferred order.
5. Click the Save button  when you finish changing the order.


Properties



Drag properties to change display order.

Name	Count
↕ Latitude	27
↕ Associate	28
↕ Longitude	27
↕ MembershipDate	16
↕ Name	32
↕ Nationality	30
↕ Resident	27
↕ Role	30

Note: The Save button is disabled if you are on the Canvas view and therefore viewing only partial metadata.

Using Geospatial Data

If your model contains geospatial data that you wish to use in the Map workspace , you must first configure entity types so the map tool can identify the data as geospatial.

1. In **Data Hub Visualization**, click the Metadata button  on the workspaces bar.
2. On the Metadata canvas, click the entity type that you want to edit.
3. Click the Edit button .
4. Click the **Map** tab.
5. Check the **Use Geospatial data** check box.
6. For **Latitude property**, select the field containing latitude data.
7. For **Longitude property**, select the field containing longitude data.
8. Click **OK**.

Opening Models Using URLs

You can open models in any of the four Data Hub Visualization areas by manually coding the URL.

1. Be sure the Spectrum™ Technology Platform server is running.
2. Open a web browser.
3. Type the following and press **Enter**:
`http://servername:port/hub-visualization/#/modelname/view/canvas` .
 o view a model in any of the other areas, simply replace "canvas" with "dashboard", "map," or "metadata."

For example, if your server is named "myserver", it uses default HTTP port 8080, and you are opening a model called "Fraud" containing insurance claim data, you would enter one of the following:

- `http://myserver:8080/hub-visualization/#/Fraud/view/canvas`
- `http://myserver:8080/hub-visualization/#/Fraud/view/dashboard`
- `http://myserver:8080/hub-visualization/#/Fraud/view/map`
- `http://myserver:8080/hub-visualization/#/Fraud/view/metadata`

Querying Models Using the Parameters Feature

The parameters feature makes it possible for you to query a model within Data Hub Visualization based on information provided in the URL. By entering the name of the model with one of the query parameters followed by a value, you can bypass the traditional steps of querying a model.

This feature is useful in situations where you want to share all or specific parts of a model but don't want the recipient to have to complete the steps it would otherwise take to access the information. You can provide the URL—the address of the server, the model name, plus these parameters—and the recipient will go directly to the data you want to share.

You can use the parameters feature to run saved queries or natural language queries; this functionality is available in the Data Hub Visualization Canvas or Map.

Note: When a query is entered in a browser address bar it becomes **URL encoded**. As you change a model or enter a query, the URL in the address bar will update to reflect your actions. This enables you to open a model, run a query, and copy from the address bar.

Saved Queries

Saved queries are performed using the `?query` parameter and the optional `input` parameter.

1. Be sure the Spectrum™ Technology Platform server is running.
2. Open a web browser.
3. Enter the following in the address bar of the browser:

```
http://servename:port/hub-visualization/#/modelname/view/canvas?query=queryname
```

For example, if your server is named "myserver", it uses default HTTP port 8080, the model containing insurance claim data for the southwestern United States is called "Fraud", and you have a saved query called "California", you would enter the following:

```
http://myserver:8080/hub-visualization/#/Fraud/view/canvas?query=California
```

Based on how the query was configured, let's say that it returns all doctor entities with a ZIP Code that begins with "9". The specific ZIP Code is included in the query as an optional input value based on a field; in this case, the query returns entities whose values in the "ZIP" field start with "9".

You can pass this data using the `input` parameter along with a JSON array of name/value pairs in the following format:

```
input=[{"name": fieldname, "value": searchfieldname}]
```

For example, if you wanted to limit the query to Hermosa Beach, a suburb of Los Angeles, you would enter the following:

```
http://myserver:8080/hub-visualization/#/Fraud/view/canvas?query=California&input=[{"name": "ZIP", "value": "<"90254"}]
```

Note:

If you enter a query that includes dates or times, they must be included in the native format:

- Date: *YYYY-MM-DD*
- Time: *HH-MM-SS*
- DateTime: *YYYY-MM-DDTHH:MM:SS*

Natural Language Queries

Natural language queries are performed using the `?search` parameter.

1. Be sure the Spectrum™ Technology Platform server is running.
2. Open a web browser.
3. Enter the following in the address bar:

```
http://servername:port/hub-visualization/#/modelname/view/canvas?search=searchstring
```

For example, if your server is named "myserver", it uses default HTTP port 8080, you are opening a model called "Fraud" containing insurance claim data, and you want to view a list of hospitals in your model, you would enter the following:

```
http://myserver:8080/hub-visualization/#/Fraud/view/canvas?search=hospital
```

Data Hub Browser

What is Data Hub Browser?

Data Hub Browser is a discovery tool where you can search the contents of a model by browsing the results of a natural-language inspired query based on the model's metadata.

The Data Hub Browser provides information about each model and its elements, including counts and other details for entities, relationships, properties, and connections. You can save queries you

create in the Data Hub Browser and use them in the Query Hub stage, the Read from Hub stage, or the Relationship Analysis Client.

Accessing Data Hub Browser

Perform the steps in this task to access the Data Hub Browser.

For you to perform this procedure, the Spectrum™ Technology Platform server must be running.

1. Open a web browser and browse to: `http://servername:port/hub-browser`.
For example, if your server is named "myserver" and it uses default HTTP port 8080, you would browse to: `http://myserver:8080/hub-browser`.
2. When the sign in screen appears, enter the same user name and password that was used to launch Enterprise Designer when you created your models, and click **Sign in**.
This displays the **Models** tab.
3. In the **Model** box, click to select a model.

You can now use all of the features within the Data Hub Browser.

The Models Tab

The **Models** tab of the Data Hub Browser provides counts for entities and relationships in the model once you have made a selection from the **Model** drop-down. It also shows a depiction of the entity types in the model and how those entities are connected. If you hover over one of the entity types, the color of the connections attached to it will turn green or red. Green connections are those that come into the entity type, and red connections are those that go out of the entity type.



The Details Tab

The **Details** tab of the Data Hub Browser provides more specific information about the model and its metadata, including entity types, relationship labels, properties, and index types of properties.

Entities

When you click **Details > Entities** on the menu bar, you will see a list of entity types for your model, along with counts for each of the types. For example, the image below indicates that the model contains 10 entities that are of the Event type.

Event	10	>
Person	33	>
Place	12	>

If you click the row for any of the entity types, you will see the number of relationships coming into that entity type, the number of relationships going out of that entity type, and the number of properties associated with that entity type. The image below indicates that the Event entity type has 42 incoming entities, 0 outgoing entities, and a total of 90 properties associated with it.

Event	10 >	Incoming	42 >
Person	33 >	Outgoing	0 >
Place	12 >	Properties	45 >

If you click the **Incoming** or **Outgoing** row, you will see a list of connections made up of source entity types, relationships, and target entity types, along with the number of times those types of connections occur in the model. For example, the image below shows that there are 4 incoming occurrences where a Person was the Organizer of an Event.

Event	10 >	Incoming	42 >	Source Entity	Relationship	Target Entity	Count
Person	33 >	Outgoing	0 >	Person	Attended	Event	19
Place	12 >	Properties	45 >	Person	Hijacked	Event	19
				Person	Organizer	Event	4

If you click the **Properties** row, you will see a complete list of properties associated with that entity type. Also shown is the data type for each property as well as the index type. The Type can be any supported data type, and Index Type is either "Exact," "Case-Insensitive," or "None." The image below indicates that there are 10 Date properties for the Event entity type, and that property is of a string type with an exact index type.

Event	10 >	Incoming	42 >	Name	Type	Index Type	Count
Person	33 >	Outgoing	0 >	Date	string	EXACT	5
Place	12 >	Properties	45 >	Event	string	EXACT	10
				Latitude	string	EXACT	0
				Location	string	EXACT	0
				Longitude	string	EXACT	0
				Type	string	EXACT	0
				_stp_id	string	EXACT	10
				_stp_label	string	EXACT	10
				_stp_type	string	EXACT	10

Relationships

When you click **Details > Relationships** on the menu bar, you will see a list of relationship labels for your model, along with counts for each of the labels. For example, the image below indicates that the model contains 56 relationships with the Attended label.

Attended	56	>
Brother	2	>
Father	2	>
Friend	13	>
Hijacked	19	>
Lived	3	>
Meeting	15	>
Organizer	4	>
Roommate	20	>
Supervisor	21	>
Traveled	21	>
Wife	1	>

If you click the row for any of the relationship labels, you will see the number of connections using that label. You will also see the number of properties associated with that relationship label. The image below indicates that the Attended relationship label has 56 connections and a total of 150 properties associated with it.

Attended	56	>	Connections	56	>
Brother	2	>	Properties	150	>
Father	2	>			
Friend	13	>			
Hijacked	19	>			
Lived	3	>			
Meeting	15	>			
Organizer	4	>			
Roommate	20	>			
Supervisor	21	>			
Traveled	21	>			
Wife	1	>			

If you click the **Connections** row, you will see a list of connections made up of source entity types, relationships, and target entity types, along with the number of times those types of connections occur in the model. For example, the image below shows that there are 19 occurrences where a Person Attended an Event.

Attended 56 >	Connections 56 >	Source Entity	Relationship	Target Entity	Count
		Person	Attended	Event	19
Brother 2 >	Properties 150 >	Person	Attended	Place	37
Father 2 >					
Friend 13 >					
Hijacked 19 >					
Lived 3 >					
Meeting 15 >					
Organizer 4 >					
Roomate 20 >					
Supervisor 21 >					
Traveled 21 >					
Wife 1 >					

If you click the **Properties** row, you will see a complete list of properties associated with that relationship label. Also shown is the data type for each property as well as the number of times that property occurs in the model. The Type can be any supported data type. The image below indicates that there are 38 Date properties for the Attended relationship label, and that property is of a string type.

Attended 56 >	Connections 56 >	Name	Type	Count
		Activity	string	56
Brother 2 >	Properties 150 >	Date	string	38
Father 2 >		Place	string	0
Friend 13 >		Rank	string	56
Hijacked 19 >				
Lived 3 >				
Meeting 15 >				
Organizer 4 >				
Roomate 20 >				
Supervisor 21 >				
Traveled 21 >				
Wife 1 >				

The Discovery Tab

The **Discovery** tab of the Data Hub Browser contains the query tool that allows you to search a model's data. This query function supports a natural-language inspired grammar, which is converted into Gremlin graph language and then submitted to the server for results.

As you enter each step of the query into the search bar, the Data Hub Browser dynamically returns type-ahead suggestions based on the model's metadata. The first step of your query will return

some actual model results along with the suggestions; however, subsequent steps will return only the type-ahead suggestions that are based on the model's metadata.

After you enter the original query that yields results, you can then conduct subsequent, iterative browsing that is driven off of those initial results.

After entering your query, click **Save** to save the query and use it in the Query Hub stage, the Read from Hub stage, the Relationship Analysis Client, or the Visualization Client.

Query Structure

Whether you are searching entities or relationships, the more specific you are in your query, the more focused the results become. Entered values can contain any Unicode character, excluding control characters. Values containing words that match metadata names should be wrapped in single or double quotes.

The asterisk (*) wildcard character following a relationship will return all entities matching the rest of the query. This is especially useful for more complex queries.

Entities

Entity Queries

The root of an entity query targets an entity index. Therefore, any properties you may want to query should be indexed. Use case-insensitive indexing to yield greater search results.

Entity Patterns

An entity query follows one of these patterns:

EntityType Property value

EntityType Property Operator value

EntityType Property value condition Property value

EntityType Property Operator value condition Property Operator value

EntityType value

Property value

Property Operator value

EntityType

value

Query Operators

The following operators are valid for entity queries. Additionally, you can use wildcard characters with the `Is Like` operator. An asterisk (*) represents any character sequence, and a question mark (?) represents any single character.

Operator	Property Type
<	Numeric or temporal

Operator	Property Type
<=	Numeric or temporal
>	Numeric or temporal
>=	Numeric or temporal
Is Not	String, Numeric or temporal
Is Similar To	String
Is Like	String
Contains	String
Does Not Contain	String
Starts With	String
Does Not Start With	String
Ends With	String
Does Not Start With	String
Matches <i>regular expression</i>	String

Conditions

Condition operators act on expressions that evaluate to True or False and on Boolean properties in entity queries.

Operator	Condition
&&	And
	Or
Is Not	Not

Entity Query Examples

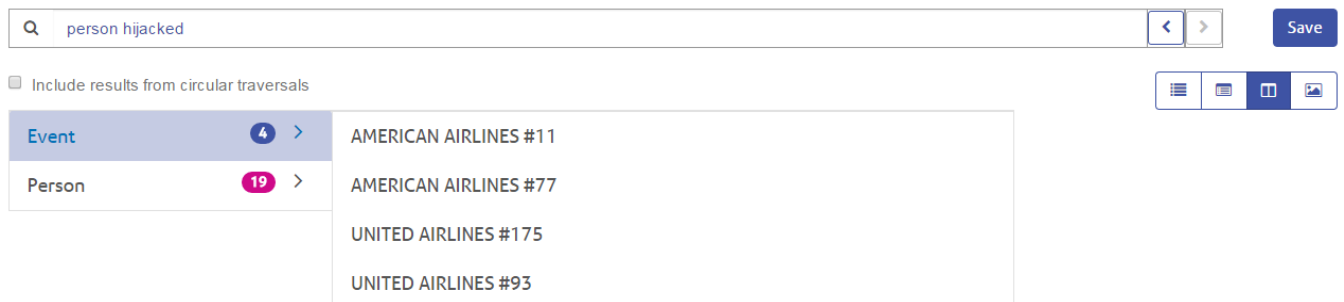
The following list includes possible entity queries into a database of terror information.

- Person Nationality Contains Yemen
- Person Name Starts With Ahm
- Person Name Starts With Osama && Name Ends With den
- Person Name Starts With Osa || Event Contains 911
- Person Osama bin Laden
- Osama bin Laden

Relationships

Relationship Queries

Relationship queries must include an entity prior to the relationship part of the query. For example, if you were looking at a model of September 11, 2001, terrorism data, you could create a query that says, "Nationality contains Saud hijacked". This query would return plane hijackers whose nationality is Saudi Arabia. More simply, if you entered a query that said, "person hijacked", the query would return all of the hijackers in the database. Incidentally, as shown below, it would also return the four planes as Event entities if your database included that information.



Relationship Patterns

The basic pattern for a relationship query follows these rules, which yield the same result: all connections to the selected relationship.

<Entity> <Relationship>

<Entity> <Relationship> *

Relationship Query Examples

The following list shows possible entity and relationship queries into a model of terrorism data.

- Osama bin Laden Meeting
- Person Nationality Contains Yemen Attended
- Event Hijacked
- Place Lived
- Date Contains 2001 Hijacked

Browsing Results

As you enter each step of the query into the search bar, the Data Hub Browser will dynamically return type-ahead suggestions based on the model's metadata. After you enter the original query that yields results, you can then conduct subsequent, iterative browsing that is driven off of those initial results. Within the results, you can click the search button to return all information related to that entity.

Note: If you want to change the label that is being returned for a particular entity type, you can change the property that is being used to populate that label within the model's metadata. See [Changing the Labels of Entity Types](#) on page 194.

The steps below depict a query of terrorism data from beginning to end and are looking for a 9/11 hijacker whose name we know begins with "Ah" and ends with "i".

1. **Person**—We know that the people in this database are represented by an entity type of "Person".

```
Person
```

2. **Name**—We know that "Person" entities have properties called "Names".

```
Person Name
```

3. **Starts With**—We know the first two letters of the hijacker's name.

```
Person Name starts with
```

4. **Ah**—We know those two letters are "Ah".

```
Person Name starts with Ah
```

5. **&&**—We need to use the "and" condition to search both first and last name information.

```
Person Name starts with Ah &&
```

6. **Name**—Again, we know that "Person" entities have properties called "Names."

```
Person Name starts with Ah && name
```

7. **Ends With**—We know the last letter of the hijacker's name.

```
Person Name starts with Ah && name ends with
```

8. **i**—We know that letter is "i".

```
Person Name starts with Ah && name ends with i
```

9. **Include results from circular traversals**—We leave this box unchecked to leave out elements that occur more than once in each traversal. For more information, see [Circular Traversals](#) on page 83.


The first image shows the query with all the steps in place. The query returns four results, displayed here in the bulleted list below the search bar. The toggle buttons in the red box provide different methods for viewing the results. Succeeding topics describe the different views.

Q Person Name Starts With Ah && name ends with i

Include results from circular traversals

- Person:Ahmed al-Darbi
- Person:Ahmed al-Ghamdi
- Person:Ahmed al-Haznawi
- Person:Ahmed al-Nami

After reviewing the results, we recognize the name Ahmed al-Nami. We hover over the name and click the magnifying glass that appears next to his name.

- Person:Ahmed al-Nami 

This displays the Entity Profile for Ahmed al-Nami shown below. It includes five relationships on which we can drill down to view details. We click the "Attended" relationship and find that he attended four events. We expand the Kandahar meeting and view the details of that event as well as data for Mr. Al-Nami.

Q Person Ahmed al-Nami

Include results from circular traversals

Person Ahmed al-Nami

Associate	Ahmed al-Nami	Name	Ahmed al-Nami
Latitude	37.09024	Nationality	Saudi
Longitude	-95.71289	Resident	Delray Beach, Florida
MembershipDate	Jan 1, 2000	Role	Hijacker

Transfer_Money	1 >	Al Farouq Training Camp	>
Provided_Money	1 >	Al Matar Training Camp	>
Friend	3 >	Al-Qaeda Training Camp	>
Attended	4 >	Kandahar Meeting Summer 2000	Q >
Roommate	1 >		
Traveled	1 >		
Hijacked	1 >		

Person Ahmed al-Nami

Associate	Ahmed al-Nami
Latitude	37.09024
Longitude	-95.71289
MembershipDate	Jan 1, 2000
Name	Ahmed al-Nami

View more...

Attended

Activity	Attended
Rank	5

Event Kandahar Meeting Summer 2000

Date	Jul 1, 2000
Event	Kandahar Meeting Summer 2000
Latitude	31.6031
Location	Kandahar, Afghanistan
Longitude	65.7615

View more...

Tree View

The Tree View presents query results in a hierarchical manner. The steps in this example still use terrorism data, but this query looks for people who organized the events leading up to and including September 11, 2001; three names and events were returned. The results are collapsible and expandable by clicking the arrow next to an element.

Q Person Organizer Event

Include results from circular traversals

- ▼ Person:Fahd al-Quso
 - Organizer ⇒ Event:USS Cole bombing
- ▼ Person:Mohammed Atef
 - Organizer ⇒ Event:1998 United States embassy bombing Dar es Salaam
 - Organizer ⇒ Event:1998 United States embassy bombing Nairobi
- ▼ Person:Walid Ba 'Attash
 - Organizer ⇒ Event:USS Cole bombing

Clicking the magnifying glass next to a name will return that person's information. The toggle buttons in the red box depict additional methods of viewing the results. As shown below, the "i" button shows the Entity Profile.

Q Person Mohammed Atef ↶ ↷ 📄

Include results from circular traversals ! ⚙️

Person Mohammed Atef

Associate	Mohammed Atef
MembershipDate	Jan 1, 1998
Name	Mohammed Atef
Nationality	Egypt
Role	Operation Chief
Supervisor	1 >
Organizer	2 >
Attended	1 >

The chart button accesses the Entity Chart, which presents entity information in a graphic manner. You can see on the top-right portion of the screen that green lines represent incoming relationships and red lines represent outgoing relationships. To view these relationships, you must hover over one of the names or events in the chart. For our example we hovered over Mohammed Atef, and he has three outgoing relationships and one incoming relationship.

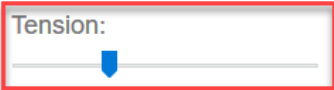
On the top-left portion of the screen, you can see in the red box a **Tension** slider control. This enables you to designate how tight or loose the connections are between entities.

You can also rotate the image. If you hover the cursor between two of the names, you will see it turn into a four-sided arrow. Drag your the cursor in any direction to see the image rotate.

Q Person Mohammed Atef ↶ ↷ 📄

Include results from circular traversals ! ⚙️

Tension:



■ Incoming

■ Outgoing

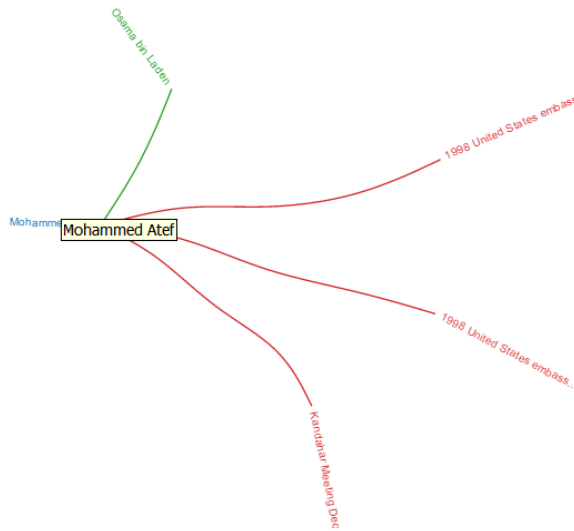


Table View

The Table View presents query results in a grid format. You can sort the data by clicking on a column heading ("Person", "Organizer", etc.); otherwise, the data is in the order that the system retrieved it.

Note: If the data in a given column would result in multiple headings, the heading will say "Entity."

↩ ↪ 📄

Include results from circular traversals

👤 📄 🗃️ 🌐

⌵

Person ▾	Organizer ▾	Event ▾
Fahd al-Quso	Organizer	USS Cole bombing
Mohammed Atef	Organizer	1998 United States embassy bombing Dar es Salaam
Mohammed Atef	Organizer	1998 United States embassy bombing Nairobi
Walid Ba 'Attash	Organizer	USS Cole bombing

To narrow the results of your query, type a search string in the **Filter** box. All columns will be searched, and only the results that contain the string you entered will be returned.

Column View

The Column View presents query results in a series of iterative columns. As you select from the results in one column, additional results appear in the subsequent column.

Q Person Organizer Event ↶ ↷ 📄

Include results from circular traversals 👤 🗃️ 🗃️ 🌐

Person	3 >
Event	3 >

By clicking anywhere in the "Person" row, we see the three entities in this model who have a role of "Organizer."

Q Person Organizer Event ↶ ↷ 📄

Include results from circular traversals 👤 🗃️ 🗃️ 🌐

Person	3 >	Fahd al-Quso
Event	3 >	Mohammed Atef
		Walid Ba 'Attash

Alternatively, clicking anywhere in the "Event" row will show the three events that the organizers planned.

Q Person Organizer Event ↶ ↷ 📄

Include results from circular traversals 👤 🗃️ 🗃️ 🌐

Person	3 >	1998 United States embassy bombing Dar es Salaam
Event	3 >	1998 United States embassy bombing Nairobi
		USS Cole bombing

As with the Tree View, we click the magnifying glass next to a name or an event to return its information. First a name:

Q Person Mohammed Atef ↶ ↷ 📄

Include results from circular traversals

! 🌐

Person **Mohammed Atef**

Associate	Mohammed Atef
MembershipDate	Jan 1, 1998
Name	Mohammed Atef
Nationality	Egypt
Role	Operation Chief

Supervisor	1 >
Organizer	2 >
Attended	1 >

Then an event:

Q Event USS Cole bombing ↶ ↷ 📄

Include results from circular traversals

! 🌐

Event **USS Cole bombing**

Date	Oct 12, 2000	Location	Aden, Yemen
Event	USS Cole bombing	Longitude	45.033333
Latitude	12.8	Type	Bombing

Organizer	2 >
-----------	-----

Also as with the Tree View, you can also access an Entity Chart by clicking the chart button.

Q Event USS Cole bombing ↶ ↷ 📄

Include results from circular traversals

! 🌐

Tension:



■ Incoming
■ Outgoing

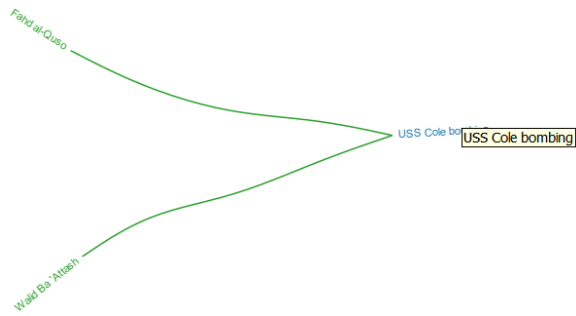


Chart View

Similar to the chart in the Tree View and Column View, the Chart View presents query results in a more graphic manner than the other views. We use the same query, which looks for people who organized the events leading up to and including September 11, 2001; three names and three events were returned.

You can see on the top-right portion of the screen that green lines represent incoming relationships and red lines represent outgoing relationships. To view these relationships, you must hover over one of the names or events in the chart. For our example we hovered over Mohammed Atef, and in these results he has two outgoing relationships, as depicted by the two red lines.

On the top-left portion of the screen, you can see in the red box a sliding Tension control. This enables you to designate how tight or loose the connections are between entities.

You can also rotate the image. If you hover the cursor between two of the names, you will see it turn into a four-sided arrow. Drag your the cursor in any direction to see the image rotate.

Circular Traversals

The **Include results from circular traversals** option on the **Discovery** tab enables you to include elements that occur more than once in each traversal. If you do not check this option, those elements are returned just once in each traversal.

For example, continuing with the same model, say you initially want to return data for any meetings that Mohamed Atta attended but once you have those results, you want to see all attendees of a particular meeting. Your initial query might look like this:

The screenshot shows a search bar with the text "Mohamed Atta Attended Event". Below the search bar, there is a checkbox labeled "Include results from circular traversals" which is unchecked. To the right of the checkbox are four icons: a group of people, a grid, a 3x3 grid, and a globe. Below these icons, a dropdown menu is open, showing a tree structure: "Person: Mohamed Atta" (expanded) with a sub-item "Attended => Event: Kandahar Meeting Dec. 1999".

The results show that he attended a meeting in Kandahar in December 1999. Now you want to know who else attended that meeting. Your second query would look like this:

The screenshot shows a search bar with the text "Mohamed Atta Attended Event Attended Person". Below the search bar, there is a checkbox labeled "Include results from circular traversals" which is unchecked. To the right of the checkbox are four icons: a group of people, a grid, a 3x3 grid, and a globe. Below these icons, a dropdown menu is open, showing a tree structure: "Person: Mohamed Atta" (expanded) with a sub-item "Attended => Event: Kandahar Meeting Dec. 1999" (expanded), which has three sub-items: "Attended => Person: Mohammed Atef", "Attended => Person: Ramzi Binalshibh", and "Attended => Person: Ziad Jarrah".

Note that if you do not check the **Include results from circular traversals** check box, Mohamed Atta's relationship will not be returned because that relationship (Person: Mohamed Atta->Attended->Meeting: Kandahar) was already traversed in the first step of the query. However, if you check the **Include results from circular traversals** check box, Mohamed Atta's relationship will be returned in addition to those for the other three attendees:

The screenshot shows a search bar with the text "Mohamed Atta Attended Event Attended Person". Below the search bar, there is a checkbox labeled "Include results from circular traversals" which is checked. To the right of the checkbox are four icons: a group of people, a grid, a 3x3 grid, and a globe. Below these icons, a dropdown menu is open, showing a tree structure: "Person: Mohamed Atta" (expanded) with a sub-item "Attended => Event: Kandahar Meeting Dec. 1999" (expanded), which has four sub-items: "Attended => Person: Mohamed Atta", "Attended => Person: Mohammed Atef", "Attended => Person: Ramzi Binalshibh", and "Attended => Person: Ziad Jarrah".

Data Hub Stages

Data Hub stages encapsulate Data Hub operations in dataflows. Dataflows are created in Enterprise Designer and executed with the Job Executor or Process Flow Executor client tools. These stages may also be incorporate in service dataflows, which can be accessed as web services or using the Spectrum™ Technology Platform API. For more information, see the *Data Flow Designer Guide*, which you can access on the Enterprise Designer **Help** menu.

Stage	Description
Delete from Hub on page 85	The Delete from Hub stage deletes entities and relationships from a model.
Import to Hub on page 87	The Import to Hub stage defines a new model or populate an existing model from two incoming channels of data. It includes an optional outgoing error port that collects records not successfully processed by the dataflow.
Merge Entities on page 91	The Merge Entities stage is used to merge two or more entities into one.
Read from Hub on page 93	The Read From Hub stage uses a saved query or new query to read data from an existing model.
Query Hub on page 104	The Query Hub stage is an intermediate stage that uses incoming data rows to define queries that extract specific entities and relationships from a model.
Split Entity on page 114	The Split Entity stage splits one entity into two or more new entities.
Write to Hub on page 116	Write to Hub is a sink stage allows you to intuitively create a model using input data to define entities, relationships, and properties. Upon execution, Write to Hub loads the data into the hub.

Delete from Hub

The Delete from Hub stage deletes entities and relationships from a model.

A dataflow that uses a Delete from Hub stage requires an input stage that contains data from or queries the same model whose elements you are deleting. It also has two optional output ports: one contains data for the deleted entities and relationships and the other contains data for the records that were not deleted.

To configure a Delete from Hub stage, you need to select the model you want to modify and then complete the Options tab and possibly the Runtime tab, depending on which write mode you want to use.

Input

The Delete from Hub stage requires that your dataflow include an input stage that contains data from or queries the same model whose elements you are deleting. This requirement could be met by a Read from Hub stage, a Query Hub stage, a source stage of some kind, or even a control stage that combines multiple other input stages.

Options

The Options Tab

The Options tab enables you to configure the elements that will be removed from your model.

1. Select the name of the model whose elements you want to remove in the **Model** field.
2. Check **Delete entities** if you want to remove entities from the model.
 - Click the **ID** drop-down and select the field that contains the entities you want to remove from your model. This is often, but not always, the `_stp_id` field if the ID and type are together, or the `_stp_label` field if the ID and type are separate.
 - If the input data stores the entity type with the ID, such as with `_stp_id`, check the **In ID field** box.
 - If the input data stores the entity type separately, such as with `_stp_label`, check the **In separate field** box and select the appropriate field in the **Type** drop-down. This is often, but not always, the `_stp_type` field.
 - To select a specific entity type, check the **Literal** box and select the appropriate field in the **Type** drop-down.
3. Check **Delete relationships** if you want to remove relationships from the model.
 - Select the field that contains the source entities in the **Source ID** field. This is often `_stp_id` or `_stp_label`.
 - If the input data stores the source entity type with the ID, such as with `_stp_id`, check the **In ID field** box.

- If the input data stores the source entity type separately, such as with `_stp_label`, check the **In separate field** box and select the appropriate field in the **Type** drop-down. This is often, but not always, the `_stp_type` field.
 - To select a specific source entity type, check the **Literal** box and select the appropriate field in the **Type** drop-down.
 - Select the field that contains the relationship between the source and target entities in the **Label** drop-down.
 - If there are multiple relationships with the same label between the source and target entities, click the **Unique ID in separate field** box and select the name of the **Unique ID**.
 - Select the field that contains the target entities in the **Target ID** field. This is often `_stp_id` or `_stp_label`.
 - If the input data stores the target entity type with the ID, such as with `_stp_id`, check the **In ID field** box.
 - If the input data stores the target entity type separately, such as with `_stp_label`, check the **In separate field** box and select the appropriate field in the **Type** drop-down. This is often, but not always, the `_stp_type` field.
 - To select a specific target entity type, check the **Literal** box and select the appropriate field in the **Type** drop-down.
4. Click **OK**.

The Runtime Tab

The Runtime tab allows you to control processing options.

1. Click **Concurrent writes** if you want to allow the model to be written to by multiple Data Hub stages at the same time. Click **Exclusive lock** (default) if you do not want to allow the model to be written to by multiple Data Hub stages. When this mode is checked, properties can be updated after they are created.
2. Click **OK**.

Output

The Delete from Hub stage has two optional outgoing ports to which you can attach various sink stages. One sink captures data for the successfully deleted entities and relationships, while the other collects any records that the dataflow did not process correctly. This is called the Error Port, and records that pass through this port into the sink are considered malformed.

Capturing malformed records can help you identify the problem with those records. When you attach a sink to the Error Port, the resulting output file will contain all the fields from the malformed records. It will also contain a Reason field that specifies why the record failed.

Import to Hub

The Import to Hub stage defines a new model or populate an existing model from two incoming channels of data. It includes an optional outgoing error port that collects records not successfully processed by the dataflow.

The Import to Hub stage is used to create a complex network of relationships called a hub, which can be displayed in a model. It can also be used to populate an existing model. Once a model is created, it can be queried against in a Read From Hub stage or a Query Hub stage, and it can be visualized in Metadata Insights or the Relationship Analysis Client.

A dataflow that uses an Import to Hub stage requires two incoming channels of data: one for entities and one for relationships. You'll then need to complete the Entities tab and the Relationships tab in the Import to Hub stage to complete your dataflow and create or update your model.

Input

The Import to Hub stage requires that your dataflow contain two channels: one that provides data for entities going into the Entity Port (the top port) and one that provides data for relationships going into the Relationship port (the bottom port). This requirement could be met by two source stages (each containing one input file), or it could come from multiple source stages that feed into Record Combiners and ultimately become two streams, or it could come from one source file whose data goes through a Conditional Router or a Splitter that outputs the data into two streams. It doesn't matter which method you use as long as the end result is a channel of entity data and a channel of relationship data that go into the Import to Hub stage.

Entity Data

Data going into the Entity Port needs to include both type and ID information for your entities. You can have a Type field ("Person") and an ID field ("Bob"), or you can have just an ID field that combines both type and ID information, separated by a colon ("Person:Bob"). For instance, your file could look something like the comma-delimited data below. The Type field tells us that the entities are people and places, and the ID field provides the names of the people and places.

```
Type, ID
Person, Robert
Person, James
Person, John
Place, London
Place, New York
Place, Los Angeles
Place, San Francisco
```

Alternatively, your input file could contain a single field that combines both type and ID:

```

ID
Person:Robert
Person:James
Person:John
Place:London
Place:New York
Place:Los Angeles
Place:San Francisco

```

Note: The fields that contain type and ID data do not actually need to be named "Type" and "ID"; any field name is acceptable.

Relationship Data

Data going into the Relationship Port needs to include fields that identify source types, source IDs, target types, target IDs, and labels that identify the relationships between the sources and targets. Note that all source and target entity information must reference entities that are provided on the Entity Port. Your relationship data may also include properties about those relationships. For instance, your file could look something like the data below. In this case, the `SourceType` field tells us that all sources are people, and the `TargetType` field tells us that the targets are people and places. The `SourceID` field provides names of all the sources, and the `TargetID` field provides names of the people and places. The `Label` field identifies the relationships, in this case "Works_With", "Works_At", or "Lives_At".

```

SourceType, SourceID, Label, TargetType, TargetID
Person, Robert, Works_With, Person, James
Person, Robert, Works_With, Person, John
Person, James, Works_With, Person, Robert
Person, James, Works_With, Person, John
Person, John, Works_With, Person, Robert
Person, John, Works_With, Person, James
Person, Robert, Works_At, Place, London
Person, James, Works_At, Place, London
Person, John, Works_At, Place, London
Person, Robert, Lives_At, Place, New York
Person, James, Lives_At, Place, Los Angeles
Person, John, Lives_At, Place, San Francisco

```

Options

The Entities Tab

The Entities tab enables you to configure the entities that will be included in your model. These entities represent objects or events, which may have properties associated with them, and these properties can be stored in your model as well if you choose to include it. Entities are linked to each other via relationships, which you will establish on the Relationships tab.

1. Enter the name of your model in the **Model** field.
2. Click the **ID** drop-down and select the field whose data you want to use to generate the entities for your model.

3. If the entity's type is contained in its own field, check the **Type in separate field** box and select the appropriate field in the **Type** drop-down.
4. The **Internal Index** grid includes a list of fields that are generated by the Import to Hub stage. The `_stp_id` field is always indexed; the `_stp_label` and `_stp_type` fields are optional. These fields can be indexed with or without case sensitivity.
5. The **Field Name** grid includes all the fields from your entity input file. Select the fields whose data you want included in the model by clicking the **Include** box for those fields.
6. Select which fields you want to be indexed in your model by clicking the **Index** box for those fields. Selecting which fields to index, rather than indexing all fields in your model, results in faster performance when writing to a model. However, if you later attempt to query fields in your model that were not indexed, the response time will be slower. For example, the **Specify starting entity** option in the query tool for the Relationship Analysis Client works only on indexed properties. You can query non-indexed properties using conditions, but the performance will be slower.
7. In the **Index Type** column, you can choose whether the data should be indexed with or without case sensitivity. Selecting Case Insensitivity typically results in greater response to a search. You cannot change the Index Type for an existing property unless it has zero counts within the model. In other words, if your model contains a property but none of the records that make up the model uses that property, you can change the index type. If one or more records uses the property, you cannot change the index type.

Note: The `_stp_id`, `_stp_type`, and `_stp_label` properties are internal properties and will always appear in the list of indexed fields. You can deselect `_stp_type` and `_stp_label`, but `_stp_id` must be indexed; however, you are able to designate whether its index type should be exact or with case insensitivity.

8. Click the **Relationships** tab to continue creating your model.

The Relationships Tab

After determining the entities for your model, you need to establish the relationships between source and target entities on the Relationships tab. These relationships represent the connection between two entities (for example, John Smith is a customer of ABC Enterprises, Inc.). As with entities, relationships may also contain properties, which you may or may not choose to include in your model.

1. Select the field that contains the source entity ID in the **Source ID** field.
2. If the source entity type is contained in its own field, check the **Type in separate field** box and select the appropriate field in the **Type** drop-down.
3. Select the field that contains the relationship between the source and target in the **Label** drop-down.
4. If you want to allow a relationship to be created more than once between a source and target entity, click the **Allow more than one relationship based on unique ID** box and select the field on which to base the relationship in the **Unique ID** drop-down.
5. Select the field that contains the target entity ID in the **Target ID** field.

6. If the target entity type is contained in its own field, check the **Type in separate field** box and select the appropriate field in the **Type** drop-down.
7. The **Field Name** grid includes all the fields from your entity input file. Select the fields whose data you want included in the model by clicking the **Include** box for those fields.
8. Click **OK**.

The Options Tab

The **Update existing model** field designates whether to update, append, remove, or retain data when an Import to Hub job is run and the model already exists. If this box is unchecked and you run a job that writes new data to the model, the existing data will first be deleted; if the new job results in blanks for a property, that property's existing data will be removed.

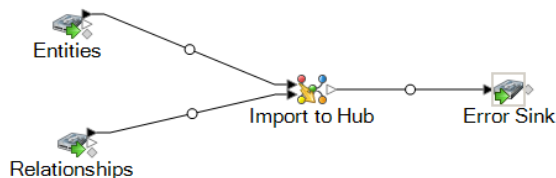
Note: Import to Hub has no error recovery capability; if you use this option, be sure to back up your existing model before updating the model with this option selected.

If this box is checked, you must specify how Import to Hub should handle existing data for both entities and relationships:

- Always update properties—Overwrite existing data with input data regardless of contents. If the input properties are empty or if properties are in the existing data but not the input data, those properties will be removed from the record.
- Update properties unless all input is null—Overwrite existing data with input data unless **all** of the input properties are empty, in which case the record will be written to the error port.
- Never overwrite non-empty properties—Do not overwrite or remove existing data with input data unless the existing properties are empty. Input data will be appended to the record.
- Never overwrite properties with empty input data—Overwrite existing data with input data but do not remove existing properties when input properties are empty.

Output

The Import to Hub stage has an optional outgoing port to which you can attach a sink stage that collects any records that the dataflow did not process correctly. This is called the Error Port, and records that pass through this port into the sink are considered malformed.



Capturing malformed records can help you identify the problem with those records. When you attach a sink to the Error Port, the resulting output file will contain a superset of the fields from both input files. It will also contain a Reason field that specifies why the record failed. So, for example, if your entities input file contains Type, ID, and Location fields, and your relationships input file contains Type, ID, and Label fields, your output file would contain Reason, Type, ID, Location, and Label fields.

Causes for record failure include, but are not limited to, the following:

- In the relationship configuration, the source entity equals the target entity.
- Relationships reference an entity that has not been defined.
- Duplicate entities or relationships exist.
- Input fields are out of order.
- Type, ID, or label fields are empty.

Merge Entities

The Merge Entities stage is used to merge two or more entities into one.

A dataflow that uses a Merge Entities stage requires one or more inputs that identify the entities you are merging. It also has two optional output ports: one contains data for the merged entity and its properties and the other contains data for the records that could not be merged.

To configure a Merge Entities stage, you need to select the model you want to modify and then complete the Options tab and possibly the Runtime tab, depending on which write mode you want to use.

Input

The Merge Entities stage requires that your dataflow include an input stage that contains data from or queries the same model whose elements you are merging. This requirement could be met by a Read from Hub stage, a Query Hub stage, a source stage of some kind, or even a control stage that combines multiple other input stages.

Options

The Options Tab

The Options tab enables you to configure the elements that will be merged in your model.

1. Select the model whose entities you want to merge in the **Model** drop-down box.
2. Select or enter the type of entities you are merging in the **Type** drop-down.
3. Select **Merge multiple sources on a single record** if you want to spread the source information across multiple fields in a single record. Select **Merge a single source on multiple records** if you want the source information to be in a single field across multiple records.
4. If you are merging multiple sources on a single record:
 - a. Select the field that contains the source entities in the **ID** field. This is often, but not always, the `_stp_id` field if the ID and type are together, or the `_stp_label` field if the ID and type are separate.
 - b. If the input data stores the entity type with the ID, such as with `_stp_id`, check the **In ID field** box.

- c. If the input data stores the entity type separately, such as with `_stp_label`, check the **In separate field** box and select the appropriate field in the **Type** drop-down. This is often, but not always, the `_stp_type` field.
- d. To select a specific entity type, check the **Literal** box and select the appropriate field in the **Type** drop-down.
- e. Repeat steps a and b to define at least one more source ID for the merge. You can add as many sources as you like. You can also delete any source by clicking the red "X" that is to the right of the source ID.

Note: By default, the first source will be considered the master source, but you can designate any of the sources to be the master. Properties for the master source will not be overwritten by properties for the other sources.

5. If you are merging a single source on multiple records:
 - a. Select the appropriate field in the **Group by** drop-down.
 - b. Select the field that contains the source entities in the **ID** field.
 - c. If the input data stores the entity type with the ID, such as with `_stp_id`, check the **In ID field** box.
 - d. If the input data stores the entity type separately, such as with `_stp_label`, check the **In separate field** box and select the appropriate field in the **Type** drop-down. This is often, but not always, the `_stp_type` field.
 - e. To select a specific entity type, check the **Literal** box and select the appropriate field in the **Type** drop-down.

Note: The entity in the first record of each group is considered the master. Properties for the master source will not be overwritten by properties of other records.

6. Click **OK**.

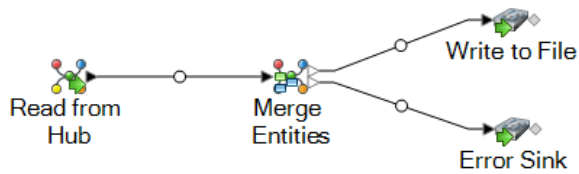
The Runtime Tab

The Runtime tab allows you to control processing options.

1. Click **Concurrent writes** if you want to allow the model to be written to by multiple Data Hub stages at the same time. Click **Exclusive lock** (default) if you do not want to allow the model to be written to by multiple Data Hub stages. When this mode is checked, properties can be updated after they are created.
2. Click **OK**.

Output

The Merge Entities stage has two optional outgoing ports to which you can attach various sink stages. One sink captures data for the successfully merged entity and its properties, while the other is used to collect data for the records that could not be merged. This is called the Error Port, and records that pass through this port into the sink are considered malformed.



Capturing malformed records can help you identify the problem with those records. When you attach a sink to the Error Port, the resulting output file will contain a superset of the fields from all records. It will also contain a Reason field that specifies why the record failed. So, for example, if one record contains Type, ID, and Location fields, and a second record contains Type, ID, and Label fields, your output file would contain Reason, Type, ID, Location, and Label fields.

Causes for record failure include, but are not limited to, the following:

- The ID field value is empty.
- The ID entered does not return any entities from the selected model.
- The ID and Type combination entered does not return any entities from the selected model.
- The ID and Type combination does not use a valid format (the required format is TYPE_VALUE:ID_VALUE).

Read from Hub

The Read From Hub stage uses a saved query or new query to read data from an existing model.

A source stage that uses a saved or new query to read the data inside an existing model. It then returns that data as fields in the dataflow output stage and makes it available for use with other stages or processes.

The Query Tab

The Query tab allows you to provide a query that returns data from the model you select. The model data is returned as data rows in the dataflow output. You can use the query builder provided, or you can create a custom query.

You can also select from or modify existing queries that are listed in the Query drop-down. If you modify existing queries and want to save those changes but also retain the original query, be sure to save the modified query under a new name; otherwise, you will overwrite the existing query. If you apply a different query to the model or cancel out of the application, changes to the query will be lost.

Follow these instructions to use the query builder:

1. Choose whether you want to **Select elements** included in the query results, **Add elements** included in the query results, or **Show result** by highlighting the results on the canvas.
2. Check the **Include results from partial traversals** box to include the results from each step in the query. Leave the box unchecked to include only the results that meet the requirements of the last step. For example, let's say that you are looking at a model that depicts world-wide terrorist activity for the events leading up to September 11, 2001, and you want to return data

for any meetings that both Osama bin Laden and Mohamed Atta attended. Your query might include the following steps:

- An initial step that includes an exact search type for entities with an `_stp_id` property that has a literal value of "Person:Osama bin Laden"
- An Entity to Relationship step of connected with a relationship label of "Meeting"
- A Relationship to Entity step of connected with a condition of an `_stp_id` property that contains "Atta"

If you leave the **Include results from partial traversals** box unchecked, the query would return a single meeting between Osama bin Laden and Mohamed Atta. If you check the box, the query would return all meetings between Osama bin Laden and anyone else in the model. The additional records would be returned because the second step looks for meetings attended by Osama bin Laden and checking the box returns results from each step in the query.

3. Leave the **Include results from circular traversals** box checked to include elements that occur more than once in each traversal. Uncheck the box to include those elements just once in each traversal. For example, let's say you are using the same model mentioned in step 1, and you initially want to return data for any meetings that Mohamed Atta attended but once you have those results, you want to see all attendees of a particular meeting. Your query might include the following steps:

- An initial step that includes an exact search type for entities with an `_stp_id` property that has a literal value of Person:Mohamed Atta, which will return his entity
- An Entity to Relationship step of connected with a relationship label of "Attended", which will return all events that he attended, including a meeting in Kandahar
- A Relationship to Entity step of connected with a condition of an Event property that contains "Kandahar", which will return just the Kandahar meeting
- An Entity to Relationship step of connected with a relationship label of "Attended", which will return relationships that connect to three other entities who attended the meeting in Kandahar and may or may not return the (already traversed) relationship that connects to Mohamed Atta, depending on whether you use this option.

If you leave the **Include results from circular traversals** box checked, Mohamed Atta's relationship will be returned in addition to those for the other three attendees. If you uncheck the box, Mohamed Atta's relationship will not be returned because that relationship (Person:Mohamed Atta->Attended->Meeting:Kandahar) was already traversed in the first step of the query.

4. Leave the **Limit results to** box checked and enter a number to specify the total maximum number of entities and relationships to return from the query. The default is 5000. The number entered here applies to unique elements, so if the same element appears in multiple results, they will count as one result. To avoid this scenario, use the dedup function discussed at the end of this topic; it will remove duplicate results in the output. If your root step returns a list and you are querying a large model, we strongly suggest entering a limit in this field to prevent the server from becoming unresponsive.

Note: Limits can be set here or as Query Result Limits set in Relationship Analysis Client [General Settings](#); if the limits are not the same, the lower limit will prevail.

5. Complete the Selection tab.

- Click **All entities**, **All relationships**, or **Specify starting entities** to identify what you want to query against. The Specify starting entities selection allows you to determine at what point in the model you want to begin your search. For instance, if you are looking at a model that depicts world-wide terrorist activity during specific years, you might have country names for entities. Rather than query against the entire model, you might want to look at activity just in Afghanistan. In this case you could select "All" as the **Search type** and "Country" as the **Property name**, leave **Literal** selected, and enter "Afghanistan" as the **Property value**.

Note: The value in the **Property value** field is case sensitive if that field was indexed with the Exact type selected (versus Case insensitive). For more information on selecting types when indexing fields, see [The Entities Tab](#) on page 117.

You could also click **Field** and select "Location" as the Property value, for example, rather than entering a specific value. If you select Field, an **Input Data** grid containing the name of the field you just selected will appear under the query name along with a cell where you can enter the default value. If you reuse this query elsewhere, you can use the default value you provided in this step, or you can override the default at that time.

- If you clicked **All entities** or **Specify starting entities**, select the **Entity types** for your query. You can choose to query selected types or all types. Click **Select None** to deselect any selected types. In addition to returning a more focused set of results, selecting entity types will affect other factors such as which properties and fields are available in the first step of the query, which directions, entity types, and relationship labels are available in subsequent steps of the query, and so on.
- If you clicked **Specify starting entities**, select the **Search type**:

Exact	Searches the index for data that matches exactly what you enter on the Selection tab, including casing. As with property values, the value here is case sensitive if that field was indexed with the Exact type selected; if case sensitivity was used and you search for "texas" while your data includes entries of "Texas", they will not be returned.
Starts with	Searches the index for data that contains text beginning with what you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "tech" or "tec" would be considered a match for a property value containing "Technical", "Technology", "Technologies", or "Technician".

Ends with	Searches the index for data that contains text ending with what you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "Emirates" or "tes" would be considered a match for a property value containing "United Arab Emirates".
Contains	Searches the index for data that contains the text that you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "Pitney" and "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc."
Any	Searches the index for data that contains any of the text that you enter on the Selection tab. For example, a literal property value of "Austin Tex" would be considered a match for a property value containing "Texarkana" or "Stephen F. Austin University".
All	Searches the index for data that contains all of the text that you enter on the Selection tab. For example, a literal property value of "Allstate claim 2013" would be considered a match for a property value containing "filed claim with Allstate June 2013", as would literal property values of "all state" or "all 13".
Between	<p>Searches the index for data that falls within a range that you specify on the Selection tab. When you use this search type, you must select a Property name that contains date, time, date/time, or numeric data. All numeric data types are supported with the exception of BigDecimal. For example, the following specifications would return all entities with StartDate values occurring in the year 2000:</p> <ul style="list-style-type: none">• A Property name of "StartDate" that is a Date type• A literal Start value of "1/1/2000"• A literal End value of "12/31/2000"

Fuzzy

Searches the index for the text you enter on the Selection tab but allows for some differentiation (missing letters, extra letters, or substitutions of letters). The amount of differentiation that is acceptable to still be considered a match depends on what you enter in the **Metric** field. This figure must be greater than zero and less than one; in other words, it must range from ".1" to ".9". For example, if you search for "Barton" and enter ".9" as the metric, the search will return records with "Carton" (replaces B with C), "Bartons" (adds s), and "Baton" (removes r), because all of these words are one character different from the search word "Barton".

Wildcard

Searches the index for the text you enter on the Selection tab but allows for a single wildcard character or a wildcard character sequence. Supported wildcard characters include the question mark (?), which matches any single character, and the asterisk (*), which matches any character sequence (including blanks). For example, if you search cities in Texas for "Aus*", the search will return records with "Austin", "Austonio" and "Austwell". If you conduct a similar search for "Aust??", only "Austin" will be returned because each question mark represents a single character and the other two city names have more characters in their name.

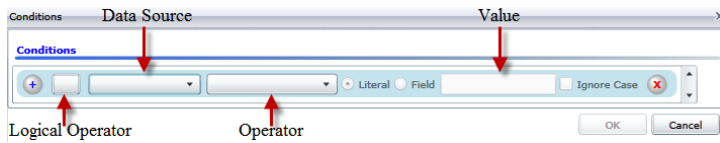
Note: A query that includes an asterisk wildcard as the first character in the search string may result in a lengthy response time.

- If you clicked **Specify starting entities**, select the **Property name** from the drop-down list. This list contains all properties associated with the entities and relationships that make up the model.

Note: You can only query properties that have been indexed; non-indexed properties will not appear in the Property name drop-down.

- If you clicked **Specify starting entities**, and selected a search type other than "Between", select the **Property value**. You can click **Literal** and enter a text string to be used in the search. Alternatively, you can click **Field** and select the field whose data should be searched; if you choose this option you will also need to enter a value in the Input Data grid.
- If you clicked **Specify starting entities**, and selected the "Between" search type, select the **Start value** and **End value** to enter the range. You can click **Literal** and enter a value to be used in the search. Alternatively, you can click **Field** and select the field whose data should be searched; if you choose this option you will also need to enter a value in the Input Data grid.

6. Complete the **Conditions** tab if you want to place additional constraints on the query. The Conditions tab has four entry fields:



- If you are creating the first condition, the **Logical operator** field will remain empty. If you are creating a subsequent condition, specify whether this condition should be used in conjunction with previous conditions ("And") or if it should be used instead of previous conditions ("Or").
- Select the property on which the condition will be based in **Data source** field.
- Select an operator for the condition that is appropriate for the data type in the **Operator** field:

Equals	Searches model elements for properties with values that match exactly what you enter in the Value field. This can be a numeric value or a text value.
Not Equals	Searches model elements for properties with values that have any value other than what you enter in the Value field. This can be a numeric value or a text value.
Exists	Searches the model elements for the existence of the property that you select in the Data Source field.
Does not Exist	Searches the model for elements that do not contain properties that you select in the Data Source field
Is Blank	Searches model elements for properties that contain no data. If a property value is blank, that element will be returned. This can be a numeric value or a text value.
Is Not Blank	Searches model elements for properties that contain any data. If a property value is not blank, that element will be returned. This can be a numeric value or a text value.
Greater Than	Searches model elements for properties whose values are greater than the value you specify. This can be a numeric, date, date/time, or time value.
Greater Than or Equals	Searches model elements for properties with numeric values that are greater than or equal to the value you specify. This can be a numeric, date, date/time, or time value.
Less Than	Searches model elements for properties with numeric values that are less than the value you specify. This can be a numeric, date, date/time, or time value.

Less Than or Equals	Searches model elements for properties with numeric values that are less than or equal to the value you specify. This can be a numeric, date, date/time, or time value.
Contains	Searches model elements for properties with values that contain what you enter in the Data Source field. The search does not need to be a complete word. For example, a literal property value of "Pitney" or "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc." This can be a numeric value or a text value.
Does not Contain	Searches model elements for properties that do not contain what you enter in the Data Source field. The search does not need to be a complete word. For example, a literal property value of "Pitney" or "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc.," so "Pitney Bowes Software" would not be returned. This can be a numeric value or a text value.
Starts With	Searches model elements for properties whose values start with the text you enter in the Data Source field. For example, if you enter "Van" for the LastName field it would return results with "Van Buren", "Vandenburg", or "Van Dyck".
Does Not Start With	Searches model elements for properties whose values do not start with the text you enter in the Data Source field. For example, if you enter "Van" for the LastName field it would not return results with "Van Buren", "Vandenburg", or "Van Dyck" but would return results with "Eddie Van Halen".
Ends With	Searches model elements for properties whose values end with the text you enter in the Data Source field. For example, if you filter for records that end with "burg" in the City field, it would return results with "Gettysburg", "Fredricksburg", and "Blacksburg".
Does Not End With	Searches model elements for properties whose values do not end with the text you enter in the Data Source field. For example, if you filter for records that end with "burg" in the City field it would not return results with "Gettysburg", "Fredricksburg", and "Blacksburg" but would return results with "Burgess".
Match Regular Expression	Searches the model elements for properties having a regular expression match for what you enter in the Data Source field. Regular expression matches identify strings of text of interest, such as particular characters, words, or patterns of characters. The value field should contain a valid regular expression pattern.
Is Roughly Similar To	Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing

letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with search metric of .5.

Is Similar To Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with search metric of .6.

Is Very Similar To Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with Search Metric of .7.

d. In the drop-down box following the list of operators:

- Select **Literal** and enter a text string the fourth box (called the **Value** field) to be used in the query.
- Select **Field** and then select the field whose data should be searched in the **Value** field.
- Select a previous step (such as "**Root**" or "**Step1**") and then a property in the **Value** field to compare property values for the current step against values returned in a previous step. (Note that if you named the output on the Output tab of previous steps, those names will appear in the drop-down rather than "Root" or "Step1".) In this case, the properties shown in the Value field are based on properties for the previous step. For example, if you knew the name of one person (Mohamed Atta) who attended a particular event (a meeting in Kandahar) but wanted to know the names of the other attendees, you could create the following query that includes a property value comparison:

- A root step that looks for an entity type Person with an `_stp_id` of that contains "Mohamed"
- An Entity to Relationship step with a relationship label of "Attended"
- A Relationship to Entity step with a condition that includes an Event that contains "Kandahar"
- An Entity to Relationship step with a relationship label of "Attended" plus a condition that this step's `_stp_id` does not contain the same `_stp_id` value that was found in the root step.

This query will find that Mohamed Atta attended an event in Kandahar and that it was also attended by three other people whose `_stp_id` value is not "Mohamed".

- e. Click **Ignore Case** if the query results can be either upper or lower cased.
- f. Repeat steps a through e to add additional conditions.
- g. Click **OK**.

If, for example, you want to target terrorist activity in Afghanistan between 2001 and 2010, you would create two conditions. First, you would select "date" for the **Property name**, then "Greater Than or Equals", leave **Literal** selected, and then enter "2001". You would follow this with a second condition set to "And" that also uses "date," then "Less Than or Equals", then "2010". Alternatively, you could click **Field** and select "Date" rather than entering a specific value. Add, delete, or change the order of conditions by using the icons on either side of the conditions. Click **Ignore Case** if the query results can be either upper or lower cased.

7. Complete the **Output** tab to define how you want your output to appear.

- Click the **Include in results** box if you want the results from this step to be included in the output.

Note: This box must be checked for the last step in any series; therefore, if there is only one step you cannot uncheck this box.

- Click **Specify name** and enter text in the **Name** field to provide a name for this step in the output. Click **List** to use this entry as the name and type of the field in hierarchical output; leave it unchecked to have this entry added as a prefix for all output fields. Using the example from step 3, you might call this step "Afghanistan". Output fields from this step may be named "Afghanistan.Latitude" or "Afghanistan.Date".
- Click **Use type name** to use the field type as the name for this step in the output. Entities will use entity types and relationships will use relationship labels. Continuing with the same example, output fields with this selection may be named "Person.Latitude" or "Person.Date". If you select this option and enter a name in the **Name** field, that name will also be added as a prefix for all output fields in addition to the field type. Continuing with the same example, output fields with this selection may be named "Afghanistan.Person.Latitude" or "Afghanistan.Person.Date".

8. Specify the steps you want the query to take by selecting the appropriate option in the **Add Operations** drop-down. You can complete this step for the Flow, Conditions, or Output tab. Note that your options vary by whether the root element is an entity or a relationship.

- If you choose **Entity to Entity** (valid for All entities and Specify starting entities), you can then refine your search to return data based on relationship labels between two entities (Connected), before entities (Predecessors), or after entities (Successors). For example, if you are querying a model of family members, and you choose a Relationship label of "Father," a Connected query will return all entities that have a Father label between them (in other words, fathers, sons, and daughters). A Predecessors query will return all entities who are a source entity of a Father relationship connected to another entity (in other words, fathers). A Successors query will return all entities who are the target entity of a Father relationship connected to another entity (in other words, sons and daughters).

As in the root step of your query, you can also select Entity types for this step of the query. You can choose to query selected types or all types. Click Select None to deselect any selected types.

- If you choose **Entity to Relationship** (valid for All entities and Specify starting entities), your options are very similar to those for Entity to Entity. You can refine your search to return data based on relationship labels that attach two entities (Connected), occur before entities (Predecessors), or occur after entities (Successors). You can also add conditions to and define output for the query.
- If you choose **Relationship to Entity** (valid for All relationships), you can refine your search to return data based on conditions you set. You can return data when a condition is in place for an entity that is connected to another entity (Connected), an entity that is a source to a relationship (Predecessors), and for when an entity is a target of a relationship (Successors).

As in the root step of your query, you can also select Entity types for this step of the query. You can choose to query selected types or all types. Click Select None to deselect any selected types.

Regardless of the type of operation you add, you can create Conditions for that operation. You can also define how you want the output from this step to appear. You will notice that steps subsequent to the root step are given a path for output. The path and the step name define the hierarchy of your output data. If you checked the List box in the root step, this path will default to being part of the path in the step before it; however, you can remove the name of the root step. For example, if you named the root step "Locations" and clicked the List box, the first step would by default show "/Locations" in the **Path** field. (Alternatively, you could remove "Locations" and leave just the slash to have this step originate at the root.) If you called the first step "CountryName", the second step would by default show "/Locations/CountryName" in the **Path** field and the Locations field output would contain a list of CountryName results. Click the **Include in results** box if you want the results from this step to be included in the output. Click **Dedup** if you want the query to remove duplicate results from the output.

9. Click **OK**.

The Fields Tab

The Fields tab allows you to designate fields to be returned in the output of your dataflow.

Note: The **Query** tab must be completed before the **Fields** tab is completed unless you are creating a custom query.

If you are using the query builder and have a field with a Dynamic Model Fields type, you can change the names of the fields that are returned for that step by selecting the step and clicking **Modify**. This will access the **Modify Field** dialog box, which contains a grid that is populated with fields and properties that map to the results of the step you selected. Similarly, the structure of this data is determined by whether you chose list output as well as the paths and names of the steps you entered on the Query tab.

If you are building a custom query, you can add, modify, or remove fields from your query.

Follow these steps to add fields to a custom query.

1. Click **Add** to open the **Add Field** dialog box.
2. Select the type of output field you want to add from the **Field type** drop-down box. The following data types are supported:

bigdecimal A numeric data type that supports 38 decimal points of precision. Use this data type for data that will be used in mathematical calculations requiring a high degree of precision, especially those involving financial data. The bigdecimal data type supports more precise calculations than the double data type.

boolean A logical type with two values: true and false.

date	A data type that contains a month, day, and year. For example, 2012-01-30 or January 30, 2012. You can specify a default date format in Management Console.
datetime	A data type that contains a month, day, year, and hours, minutes, and seconds. For example, 2012/01/30 6:15 PM.
double	A numeric data type that contains both negative and positive double precision numbers between 2^{-1074} and $(2-2^{-52}) \times 2^{1023}$. In E notation, the range of values is 4.9E-324 to 1.7976931348623157E308. For information on E notation, see en.wikipedia.org/wiki/Scientific_notation#E_notation .
float	A numeric data type that contains both negative and positive single precision numbers between 2^{-149} and $(2-2^{-23}) \times 2^{127}$. In E notation, the range of values is 1.4E-45 to 3.4028235E38. For information on E notation, see en.wikipedia.org/wiki/Scientific_notation#E_notation .
integer	A numeric data type that contains both negative and positive whole numbers between -2^{31} (-2,147,483,648) and $2^{31}-1$ (2,147,483,647).
long	A numeric data type that contains both negative and positive whole numbers between -2^{63} (-9,223,372,036,854,775,808) and $2^{63}-1$ (9,223,372,036,854,775,807).
string	A sequence of characters.
time	A data type that contains the time of day. For example, 21:15:59 or 9:15:59 PM.

If you want to add a field from your model, select **Dynamic Model Fields** and the tab will be populated with fields and properties from your model. The structure of this data is determined by whether you choose list output as well as the paths and names entered on **Add Field** dialog box.

You can also add a new, user-defined, custom data type if necessary. Among other types of data, a new type can be a list of any defined data type (though you could select most types from the drop-down options and click the List check box).

3. Enter the **Field name** for the field you want to query in the model.
4. Click **List** to return output in a hierarchical format; leave it unchecked to add prefixes to output fields.
5. Click **Cancel** when you're done adding fields.
6. Click **OK**.

Output

The Read From Hub stage requires that your dataflow contain an output stage that has defined, at minimum, the field or fields that you are querying. Otherwise, that data will not appear in your output. For example, if your Read From Hub stage queries the `_stf_id` field in your hub, your output stage must contain a field that captures that data.

Query Hub

The Query Hub stage is an intermediate stage that uses incoming data rows to define queries that extract specific entities and relationships from a model.

For example, Query Hub can be used as part of a service to understand a customer's influence score within the network or determine if a customer record already exists in the hub.

The Query Tab

The Query tab allows you to provide a query that returns data from the model you select. The model data is returned as data rows in the dataflow output. You can use the query builder provided, or you can create a custom query.

You can also select from or modify existing queries that are listed in the Query drop-down. If you modify existing queries and want to save those changes but also retain the original query, be sure to save the modified query under a new name; otherwise, you will overwrite the existing query. If you apply a different query to the model or cancel out of the application, changes to the query will be lost.

Follow these instructions to use the query builder:

1. Choose whether you want to **Select elements** included in the query results, **Add elements** included in the query results, or **Show result** by highlighting the results on the canvas.
2. Check the **Include results from partial traversals** box to include the results from each step in the query. Leave the box unchecked to include only the results that meet the requirements of the last step. For example, let's say that you are looking at a model that depicts world-wide terrorist activity for the events leading up to September 11, 2001, and you want to return data for any meetings that both Osama bin Laden and Mohamed Atta attended. Your query might include the following steps:
 - An initial step that includes an exact search type for entities with an `_stp_id` property that has a literal value of "Person:Osama bin Laden"
 - An Entity to Relationship step of connected with a relationship label of "Meeting"
 - A Relationship to Entity step of connected with a condition of an `_stp_id` property that contains "Atta"If you leave the **Include results from partial traversals** box unchecked, the query would return a single meeting between Osama bin Laden and Mohamed Atta. If you check the box, the query would return all meetings between Osama bin Laden and anyone else in the model. The additional records would be returned because the second step looks for meetings attended by Osama bin Laden and checking the box returns results from each step in the query.
3. Leave the **Include results from circular traversals** box checked to include elements that occur more than once in each traversal. Uncheck the box to include those elements just once in each traversal. For example, let's say you are using the same model mentioned in step 1, and you initially want to return data for any meetings that Mohamed Atta attended but once you have those results, you want to see all attendees of a particular meeting. Your query might include the following steps:

- An initial step that includes an exact search type for entities with an `_stp_id` property that has a literal value of `Person:Mohamed Atta`, which will return his entity
- An Entity to Relationship step of connected with a relationship label of "Attended", which will return all events that he attended, including a meeting in Kandahar
- A Relationship to Entity step of connected with a condition of an Event property that contains "Kandahar", which will return just the Kandahar meeting
- An Entity to Relationship step of connected with a relationship label of "Attended", which will return relationships that connect to three other entities who attended the meeting in Kandahar and may or may not return the (already traversed) relationship that connects to Mohamed Atta, depending on whether you use this option.

If you leave the **Include results from circular traversals** box checked, Mohamed Atta's relationship will be returned in addition to those for the other three attendees. If you uncheck the box, Mohamed Atta's relationship will not be returned because that relationship (Person:Mohamed Atta->Attended->Meeting:Kandahar) was already traversed in the first step of the query.

4. Leave the **Limit results to** box checked and enter a number to specify the total maximum number of entities and relationships to return from the query. The default is 5000. The number entered here applies to unique elements, so if the same element appears in multiple results, they will count as one result. To avoid this scenario, use the dedup function discussed at the end of this topic; it will remove duplicate results in the output. If your root step returns a list and you are querying a large model, we strongly suggest entering a limit in this field to prevent the server from becoming unresponsive.

Note: Limits can be set here or as Query Result Limits set in Relationship Analysis Client [General Settings](#); if the limits are not the same, the lower limit will prevail.

5. Complete the Selection tab.

- Click **All entities**, **All relationships**, or **Specify starting entities** to identify what you want to query against. The Specify starting entities selection allows you to determine at what point in the model you want to begin your search. For instance, if you are looking at a model that depicts world-wide terrorist activity during specific years, you might have country names for entities. Rather than query against the entire model, you might want to look at activity just in Afghanistan. In this case you could select "All" as the **Search type** and "Country" as the **Property name**, leave **Literal** selected, and enter "Afghanistan" as the **Property value**.

Note: The value in the **Property value** field is case sensitive if that field was indexed with the Exact type selected (versus Case insensitive). For more information on selecting types when indexing fields, see [The Entities Tab](#) on page 117.

You could also click **Field** and select "Location" as the Property value, for example, rather than entering a specific value. If you select Field, an **Input Data** grid containing the name of the field you just selected will appear under the query name along with a cell where you can enter the default value. If you reuse this query elsewhere, you can use the default value you provided in this step, or you can override the default at that time.

- If you clicked **All entities** or **Specify starting entities**, select the **Entity types** for your query. You can choose to query selected types or all types. Click **Select None** to deselect any selected types. In addition to returning a more focused set of results, selecting entity types will affect other factors such as which properties and fields are available in the first step of the query, which directions, entity types, and relationship labels are available in subsequent steps of the query, and so on.
- If you clicked **Specify starting entities**, select the **Search type**:

Exact	Searches the index for data that matches exactly what you enter on the Selection tab, including casing. As with property values, the value here is case sensitive if that field was indexed with the Exact type selected; if case sensitivity was used and you search for "texas" while your data includes entries of "Texas", they will not be returned.
Starts with	Searches the index for data that contains text beginning with what you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "tech" or "tec" would be considered a match for a property value containing "Technical", "Technology", "Technologies", or "Technician".
Ends with	Searches the index for data that contains text ending with what you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "Emirates" or "tes" would be considered a match for a property value containing "United Arab Emirates".
Contains	Searches the index for data that contains the text that you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "Pitney" and "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc."
Any	Searches the index for data that contains any of the text that you enter on the Selection tab. For example, a literal property value of "Austin Tex" would be considered a match for a property value containing "Texarkana" or "Stephen F. Austin University".

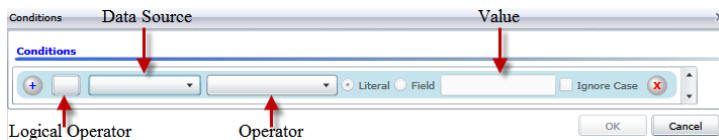
All	Searches the index for data that contains all of the text that you enter on the Selection tab. For example, a literal property value of "Allstate claim 2013" would be considered a match for a property value containing "filed claim with Allstate June 2013", as would literal property values of "all state" or "all 13".
Between	<p>Searches the index for data that falls within a range that you specify on the Selection tab. When you use this search type, you must select a Property name that contains date, time, date/time, or numeric data. All numeric data types are supported with the exception of BigDecimal. For example, the following specifications would return all entities with StartDate values occurring in the year 2000:</p> <ul style="list-style-type: none">• A Property name of "StartDate" that is a Date type• A literal Start value of "1/1/2000"• A literal End value of "12/31/2000"
Fuzzy	Searches the index for the text you enter on the Selection tab but allows for some differentiation (missing letters, extra letters, or substitutions of letters). The amount of differentiation that is acceptable to still be considered a match depends on what you enter in the Metric field. This figure must be greater than zero and less than one; in other words, it must range from ".1" to ".9". For example, if you search for "Barton" and enter ".9" as the metric, the search will return records with "Carton" (replaces B with C), "Bartons" (adds s), and "Baton" (removes r), because all of these words are one character different from the search word "Barton".
Wildcard	<p>Searches the index for the text you enter on the Selection tab but allows for a single wildcard character or a wildcard character sequence. Supported wildcard characters include the question mark (?), which matches any single character, and the asterisk (*), which matches any character sequence (including blanks). For example, if you search cities in Texas for "Aus*", the search will return records with "Austin", "Austonio" and "Austwell". If you conduct a similar search for "Aust??", only "Austin" will be returned because each question mark represents a single character and the other two city names have more characters in their name.</p> <p>Note: A query that includes an asterisk wildcard as the first character in the search string may result in a lengthy response time.</p>

- If you clicked **Specify starting entities**, select the **Property name** from the drop-down list. This list contains all properties associated with the entities and relationships that make up the model.

Note: You can only query properties that have been indexed; non-indexed properties will not appear in the Property name drop-down.

- If you clicked **Specify starting entities**, and selected a search type other than "Between", select the **Property value**. You can click **Literal** and enter a text string to be used in the search. Alternatively, you can click **Field** and select the field whose data should be searched; if you choose this option you will also need to enter a value in the Input Data grid.
- If you clicked **Specify starting entities**, and selected the "Between" search type, select the **Start value** and **End value** to enter the range. You can click **Literal** and enter a value to be used in the search. Alternatively, you can click **Field** and select the field whose data should be searched; if you choose this option you will also need to enter a value in the Input Data grid.

6. Complete the **Conditions** tab if you want to place additional constraints on the query. The Conditions tab has four entry fields:



- a. If you are creating the first condition, the **Logical operator** field will remain empty. If you are creating a subsequent condition, specify whether this condition should be used in conjunction with previous conditions ("And") or if it should be used instead of previous conditions ("Or").
- b. Select the property on which the condition will be based in **Data source** field.
- c. Select an operator for the condition that is appropriate for the data type in the **Operator** field:

Equals	Searches model elements for properties with values that match exactly what you enter in the Value field. This can be a numeric value or a text value.
Not Equals	Searches model elements for properties with values that have any value other than what you enter in the Value field. This can be a numeric value or a text value.
Exists	Searches the model elements for the existence of the property that you select in the Data Source field.
Does not Exist	Searches the model for elements that do not contain properties that you select in the Data Source field
Is Blank	Searches model elements for properties that contain no data. If a property value is blank, that element will be returned. This can be a numeric value or a text value.

Is Not Blank	Searches model elements for properties that contain any data. If a property value is not blank, that element will be returned. This can be a numeric value or a text value.
Greater Than	Searches model elements for properties whose values are greater than the value you specify. This can be a numeric, date, date/time, or time value.
Greater Than or Equals	Searches model elements for properties with numeric values that are greater than or equal to the value you specify. This can be a numeric, date, date/time, or time value.
Less Than	Searches model elements for properties with numeric values that are less than the value you specify. This can be a numeric, date, date/time, or time value.
Less Than or Equals	Searches model elements for properties with numeric values that are less than or equal to the value you specify. This can be a numeric, date, date/time, or time value.
Contains	Searches model elements for properties with values that contain what you enter in the Data Source field. The search does not need to be a complete word. For example, a literal property value of "Pitney" or "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc." This can be a numeric value or a text value.
Does not Contain	Searches model elements for properties that do not contain what you enter in the Data Source field. The search does not need to be a complete word. For example, a literal property value of "Pitney" or "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc.," so "Pitney Bowes Software" would not be returned. This can be a numeric value or a text value.
Starts With	Searches model elements for properties whose values start with the text you enter in the Data Source field. For example, if you enter "Van" for the LastName field it would return results with "Van Buren", "Vandenburg", or "Van Dyck".
Does Not Start With	Searches model elements for properties whose values do not start with the text you enter in the Data Source field. For example, if you enter "Van" for the LastName field it would not return results with "Van Buren", "Vandenburg", or "Van Dyck" but would return results with "Eddie Van Halen".
Ends With	Searches model elements for properties whose values end with the text you enter in the Data Source field. For example, if you filter for records that end with "burg" in the City field, it would return results with "Gettysburg", "Fredricksburg", and "Blacksburg".

Does Not End With	Searches model elements for properties whose values do not end with the text you enter in the Data Source field. For example, if you filter for records that end with "burg" in the City field it would not return results with "Gettysburg", "Fredricksburg", and "Blacksburg" but would return results with "Burgess".
Match Regular Expression	Searches the model elements for properties having a regular expression match for what you enter in the Data Source field. Regular expression matches identify strings of text of interest, such as particular characters, words, or patterns of characters. The value field should contain a valid regular expression pattern.
Is Roughly Similar To	Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with search metric of .5.
Is Similar To	Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with search metric of .6.
Is Very Similar To	Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with Search Metric of .7.

d. In the drop-down box following the list of operators:

- Select **Literal** and enter a text string the fourth box (called the **Value** field) to be used in the query.
- Select **Field** and then select the field whose data should be searched in the **Value** field.
- Select a previous step (such as "**Root**" or "**Step1**") and then a property in the **Value** field to compare property values for the current step against values returned in a previous step. (Note that if you named the output on the Output tab of previous steps, those names will appear in the drop-down rather than "Root" or "Step1".) In this case, the properties shown in the Value field are based on properties for the previous step. For example, if you knew the name of one person (Mohamed Atta) who attended a particular event (a meeting in Kandahar) but wanted to know the names of the other attendees, you could create the following query that includes a property value comparison:
 - A root step that looks for an entity type Person with an `_stp_id` of that contains "Mohamed"
 - An Entity to Relationship step with a relationship label of "Attended"
 - A Relationship to Entity step with a condition that includes an Event that contains "Kandahar"
 - An Entity to Relationship step with a relationship label of "Attended" plus a condition that this step's `_stp_id` does not contain the same `_stp_id` value that was found in the root step.

This query will find that Mohamed Atta attended an event in Kandahar and that it was also attended by three other people whose `_stp_id` value is not "Mohamed".

- e. Click **Ignore Case** if the query results can be either upper or lower cased.
- f. Repeat steps a through e to add additional conditions.
- g. Click **OK**.

If, for example, you want to target terrorist activity in Afghanistan between 2001 and 2010, you would create two conditions. First, you would select "date" for the **Property name**, then "Greater Than or Equals", leave Literal selected, and then enter "2001". You would follow this with a second condition set to "And" that also uses "date," then "Less Than or Equals", then "2010". Alternatively, you could click **Field** and select "Date" rather than entering a specific value. Add, delete, or change the order of conditions by using the icons on either side of the conditions. Click **Ignore Case** if the query results can be either upper or lower cased.

7. Complete the **Output** tab to define how you want your output to appear.

- Click the **Include in results** box if you want the results from this step to be included in the output.

Note: This box must be checked for the last step in any series; therefore, if there is only one step you cannot uncheck this box.

- Click **Specify name** and enter text in the **Name** field to provide a name for this step in the output. Click **List** to use this entry as the name and type of the field in hierarchical output; leave it unchecked to have this entry added as a prefix for all output fields. Using the example from step 3, you might call this step "Afghanistan". Output fields from this step may be named "Afghanistan.Latitude" or "Afghanistan.Date".
- Click **Use type name** to use the field type as the name for this step in the output. Entities will use entity types and relationships will use relationship labels. Continuing with the same example, output fields with this selection may be named "Person.Latitude" or "Person.Date". If you select this option and enter a name in the **Name** field, that name will also be added as a prefix for all output fields in addition to the field type. Continuing with the same example, output fields with this selection may be named "Afghanistan.Person.Latitude" or "Afghanistan.Person.Date".

8. Specify the steps you want the query to take by selecting the appropriate option in the **Add Operations** drop-down. You can complete this step for the Flow, Conditions, or Output tab. Note that your options vary by whether the root element is an entity or a relationship.

- If you choose **Entity to Entity** (valid for All entities and Specify starting entities), you can then refine your search to return data based on relationship labels between two entities (Connected), before entities (Predecessors), or after entities (Successors). For example, if you are querying a model of family members, and you choose a Relationship label of "Father," a Connected query will return all entities that have a Father label between them (in other words, fathers, sons, and daughters). A Predecessors query will return all entities who are a source entity of a Father relationship connected to another entity (in other words, fathers). A Successors query will return all entities who are the target entity of a Father relationship connected to another entity (in other words, sons and daughters).

As in the root step of your query, you can also select Entity types for this step of the query. You can choose to query selected types or all types. Click Select None to deselect any selected types.

- If you choose **Entity to Relationship** (valid for All entities and Specify starting entities), your options are very similar to those for Entity to Entity. You can refine your search to return data based on relationship labels that attach two entities (Connected), occur before entities (Predecessors), or occur after entities (Successors). You can also add conditions to and define output for the query.
- If you choose **Relationship to Entity** (valid for All relationships), you can refine your search to return data based on conditions you set. You can return data when a condition is in place for an entity that is connected to another entity (Connected), an entity that is a source to a relationship (Predecessors), and for when an entity is a target of a relationship (Successors). As in the root step of your query, you can also select Entity types for this step of the query. You can choose to query selected types or all types. Click Select None to deselect any selected types.

Regardless of the type of operation you add, you can create Conditions for that operation. You can also define how you want the output from this step to appear. You will notice that steps subsequent to the root step are given a path for output. The path and the step name define the hierarchy of your output data. If you checked the List box in the root step, this path will default to being part of the path in the step before it; however, you can remove the name of the root step. For example, if you named the root step "Locations" and clicked the List box, the first step would by default show "/Locations" in the **Path** field. (Alternatively, you could remove "Locations" and leave just the slash to have this step originate at the root.) If you called the first step "CountryName", the second step would by default show "/Locations/CountryName" in the **Path** field and the Locations field output would contain a list of CountryName results. Click the **Include in results** box if you want the results from this step to be included in the output. Click **Dedup** if you want the query to remove duplicate results from the output.

9. Click **OK**.

The Fields Tab

The **Fields** tab allows you to designate fields to be returned and put into the output of your dataflow. If you are building a custom script, you may add, modify, or remove fields. If you are using the query builder the fields are populated for you, but you can change the name and properties of the entities and relationships that were auto-populated.

1. Create your query on the **Query** tab.

Note: This step can be completed before or after the **Fields** tab is complete.

2. Click the **Fields** tab.
3. Click **Add** to open the **Add Input Field** dialog box.
4. Use the **Fields** drop-down and the **Add** button to select the fields you want to query in the model. The fields available for selection depend on the fields assigned in the dataflow input

stage. Spectrum supports both simple and complex data types; you can use fields deep within the hierarchical structure of your input file in your query.

5. Click **Close** when you're done adding input fields.
6. Click **Add** to open the **Add Output Field** dialog box.
7. Select the type of output field you want to add from the **Type** drop-down box. The following data types are supported:

Data Type	Description
boolean	A logical type with two values: true and false.
double	A numeric data type that contains both negative and positive double precision numbers between 2^{-1074} and $(2-2^{-52}) \times 2^{1023}$. In E notation, the range of values is 4.9E-324 to 1.7976931348623157E308. For information on E notation, see: http://en.wikipedia.org/wiki/Scientific_notation#E_notation
relationships	The links between entities; the factor they share with other entities. If you choose this type of output field, you can change field names and determine which relationships to include upon output. You will need to include in your Gremlin script a command to retrieve a list of relationships and assign them to that data type, as shown in this example: <div style="background-color: #f0f0f0; padding: 5px; margin: 10px 0;"> <pre>data["Variants"]=g.idx('entities')[['Name':name]].bothE</pre> </div> <p>Note: If you create an output field this way, Query Hub will auto-populate the schema for you.</p>
float	A numeric data type that contains both negative and positive single precision numbers between 2^{-149} and $(2-2^{-23}) \times 2^{127}$. In E notation, the range of values is 1.4E-45 to 3.4028235E38. For information on E notation, see: http://en.wikipedia.org/wiki/Scientific_notation#E_notation
integer	A numeric data type that contains both negative and positive whole numbers between -2^{31} (-2,147,483,648) and $2^{31}-1$ (2,147,483,647).
long	A numeric data type that contains both negative and positive whole numbers between -2^{63} (-9,223,372,036,854,775,808) and $2^{63}-1$ (9,223,372,036,854,775,807)

Data Type	Description
entities	<p>The individual entities that have the relationships you are identifying. If you choose this type of output field, you can change field names and determine which entities to include upon output. You will need to include in your Gremlin script a command to retrieve a list of entities (or vertices) and assign them to that data type, as shown in this example:</p> <pre>data["Variants"]=g.idx('entities')[['Name':name]].both</pre> <p>Note: If you create an output field this way, Query Hub will auto-populate the schema for you.</p>
string	A sequence of characters.

You can also add a new, user-defined, custom data type if necessary. Among other types of data, a new type can be a list of any defined data type (though for string, double, integer, long, float, or Boolean types you would select that type from the drop-down options and click the List check box). You can also select an output field based on entities and relationships in your model. If you create an output field this way, Query Hub will auto-populate the schema for you. Additionally, you would need to retrieve the entities and/or relationships from Gremlin and then iterate over them, creating a loop. This would result in you building up the fields' child rows and assigning data as you go.

8. Specify a name for the output field you want to add in the **Name** field.
9. Click the **List** check box if you created a custom data type in the form of a list.
10. When you are done adding output fields, click **Close**.
11. Click **OK**.

Input/Output Requirements

The Query Graph stage requires that the input stage of your dataflow has defined the input fields that are accessed using data (“input”). Also, any input fields or output fields accessed using the data command need to be defined in the Fields tab in the input and output stages. Otherwise, they will not appear as input and output fields in other stages in your dataflow.

Split Entity

The Split Entity stage splits one entity into two or more new entities.

A dataflow that uses a Split Entity stage requires an input that identifies the entity you are splitting. It also has two optional output ports: one contains data for the split entities and the other contains data for the records that were not split.

To configure a Split Entity stage, you need to select the model you want to modify and then complete the Options tab and possibly the Runtime tab, depending on which write mode you want to use.

Input

The Split Entity stage requires that your dataflow include an input stage that contains data from or queries the same model whose element you are splitting. This requirement could be met by a Read from Hub stage, a Query Hub stage, a source stage of some kind, or even a control stage that combines multiple other input stages.

Options

The Options Tab

The Options tab enables you to configure the element that will be split in your model.

1. Select the model whose entities you want to split in the **Model** drop-down box.
2. Select the field that contains the entities you want to split in the Source **ID** field. This is often, but not always, the `_stp_id` field if the ID and type are together, or the `_stp_label` field if the ID and type are separate.
3. Select the type of entities you want to split in the Source **Type** drop-down.
4. Select or enter the type for the newly created entities in the Target **Type** drop-down.
5. Check **Replace label with data from input field** and select the appropriate field if you want to change the label of the newly created entities.
6. Select the relationships you want to be included with the split entities.
7. Repeat steps 4 through 6 if you want to create additional entities from the source entities.
8. Click **OK**.

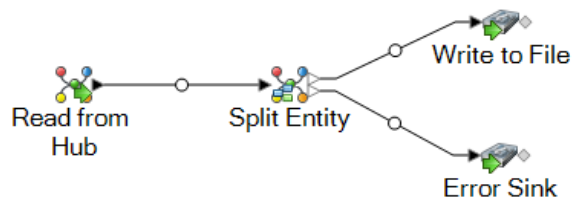
The Runtime Tab

The Runtime tab allows you to control processing options.

1. Click **Concurrent writes** if you want to allow the model to be written to by multiple Data Hub stages at the same time. Click **Exclusive lock** (default) if you do not want to allow the model to be written to by multiple Data Hub stages. When this mode is checked, properties can be updated after they are created.
2. Check the **Remove orphans** box to have entities with no relationships after the split removed from the model.
3. Click **OK**.

Output

The Split Entity stage has two optional outgoing ports to which you can attach various sink stages. One sink captures data for the successfully split entities and their properties, while the other is used to collect data for the records that were not split. This is called the Error Port, and records that pass through this port into the sink are considered malformed.



Capturing malformed records can help you identify the problem with those records. When you attach a sink to the Error Port, the resulting output file will contain a superset of the fields from both input files. It will also contain a Reason field that specifies why the record failed. So, for example, if your entities input file contains Type, ID, and Location fields, and your relationships input file contains Type, ID, and Label fields, your output file would contain Reason, Type, ID, Location, and Label fields.

Causes for record failure include, but are not limited to, the following:

- The ID field value is empty.
- The ID entered does not return any entities from the selected model.
- The ID and Type combination entered does not return any entities from the selected model.
- The ID and Type combination does not use a valid format (the required format is TYPE_VALUE:ID_VALUE).

Write to Hub

Write to Hub is a sink stage allows you to intuitively create a model using input data to define entities, relationships, and properties. Upon execution, Write to Hub loads the data into the hub.

The Write to Hub stage can be used to link data together, resulting in a complex network of relationships called a hub, which can be displayed in a model. Once that model is created, it can be queried against in a Read From Hub stage or a Query Hub stage, or it can be visualized in Data Hub Visualization workspaces or in the Relationship Analysis Client to identify relationships and trends that may otherwise be hard to find.

To configure a Write to Hub stage, you need to complete the Entities tab and the Relationships tab. You can also use the Options tab to set processing preferences and determine how you want data to be written to the hub.

For more information, see [Sample Write to Hub Dataflow](#) on page 122 for examples of configuring a Write to Hub dataflow from start to finish using a flat file and an XML file.

Input

The Write to Hub stage requires that your dataflow contain an input stage with defined fields that you can use to create a model.

The Entities Tab

The Entities tab allows you to configure how entities are created and updated. These entities can represent objects or events and will be stored in your model. You can use the Relationships tab to link them to other entities and create relationships. To create entities for your model, you will need to complete the following information:

1. Enter the name of your model in the **Model** field.
2. Click **Add...** to create a new entity. The **Add Entity** dialog box will appear.
3. Select the field name to be used to generate the Entity ID in the **Input field** field.
4. If you want the entity to have a name other than what is automatically provided, change the contents of the **Type** field to the desired name.
5. On the **Properties** tab, you define which properties you want to be included with the entity type you are creating.

For example, if you are creating an entity type that represents places, you might want to choose latitude and longitude as properties. You can select **Input** or **Metadata**, depending on what you want to use to define the properties. If your model does not contain metadata, that option will be disabled.

Option	Description
Input	<p>The grid is populated with input fields, which are shown in the Field column; there is one row for each input field. Select the fields you want to be included as properties for the entity type by checking the Include box for that field. The Name column represents the name you want to property to have in the model; it defaults to the input field name, but you can select from any property in the Name drop-down or manually enter a name.</p> <p>Note: Spectrum supports both simple and complex data types; you can use fields deep within the hierarchical structure of your input file as an entity. If you are using hierarchical data, you will also see a Filter control that allows you to filter out data on the Property list based on the path of the field. Likewise, you will see a control that allows you to hide non-primitive fields.</p>
Metadata	<p>The grid is populated with properties from the metadata, which are shown in the Name column; there is one row for each property. The properties shown are determined by whether you are defining a known entity type. If it is a known type, the Name column will include properties specific to that type; if it is not a known type, the Name column will include a list of all properties in the model. Select the properties</p>

Option	Description
--------	-------------

you want to populate for the entity type by checking the **Include** box for that property. The **Field** column contains names of input fields you can map properties to and whose data can be used to populate properties. If an input field matches the property name, it will automatically be mapped and the Include box will be checked.

- Click the **Updates** tab and select how you want Write to Hub to manage updates. After an entity is created it can be updated over time when data with the same ID is input into the Write to Hub stage.

Option	Description
Always update properties	Properties are always updated with the most recent information. This includes updating with null or empty strings.
Update properties unless all input is null	Properties are always updated unless all input fields associated with the selected properties are null.
Never overwrite properties with empty data	Properties are always updated unless the input is a null or empty string.
Never overwrite non-empty properties	Properties are never updated once populated with non-empty data.

- Repeat steps [4](#) on page 117 through [6](#) on page 118 to add additional entities. When you are done adding entities, click **Close**.
- Select which fields you want to be indexed in your model by clicking the **Indexes...** button and checking the box for those fields.

In the **Type** column, you can choose whether the data should be indexed exactly as-is, with case sensitivity, or if it should be indexed without case sensitivity, which typically results in greater response to a search.

Note: The `_stp_id`, `_stp_type`, and `_stp_label` properties are internal properties and will always appear in the list of indexed fields. You can deselect `_stp_type` and `_stp_label`, but `_stp_id` must be indexed; however, you are able to designate whether its index type should be exact or with case insensitivity.

Selecting which fields to index, rather than indexing all fields in your model, results in faster performance when writing to a model. However, if you later attempt to query fields in your model that were not indexed, the response time will be slower.

For example, the **Specify starting entity** option in the query tool for the Relationship Analysis Client works only on indexed properties. You can query non-indexed properties using conditions, but the performance will be slower.

The Relationships Tab

Provide information described in this procedure to create relationships for your model.

The **Relationships** tab allows you to configure how relationships are created between source and target entities. These relationships represent the connection between two entities (for example, John knows Mary).

1. Click **Add**
2. Select the entity to act as a source for your relationship in the **Source** field.
3. Click the appropriate label type for your relationship: **String** or **Field**. If string, enter the string in the text box underneath. If field, select the field you want to use to generate the label for this relationship in the drop-down box.
4. Optional: To allow a relationship to be created more than once between a source and target entity, click the **Allow more than one relationship based on unique ID** box and select the field on which to base the relationship in the drop-down box.
5. Select the entity to act as a target for your relationship in the **Target** field.
6. On the **Properties** tab, define which properties you want to be included with the relationship you are creating.

For example, if you are creating a relationship called "treated" between an entity type of "doctor" and an entity type of "patient," you might want to choose date and diagnosis as properties. You can select **Input** or **Metadata**, depending on what you want to use to define the properties. If your model does not contain metadata, that option will be disabled.

Option	Description
Input	The grid is populated with input fields, which are shown in the Field column; there is one row for each input field. Select the fields you want to be included as properties for the relationship by checking the Include box for that field. The Name column represents the name you want to property to have in the model; it defaults to the input field name, but you can select from any property in the Name drop-down or manually enter a name.
Metadata	The grid is populated with properties from the metadata, which are shown in the Name column; there is one row for each property. The properties shown are determined by whether you are defining a known relationship. If it is a known relationship, the Name column will include properties specific to that relationship; if it is not known, the Name column will include a list of all properties in the model. Select the properties you want to populate for the relationship by checking the Include box for that property. The Field column contains names of input fields you can map properties to and whose data can be used to populate properties. If an

Option	Description
--------	-------------

	input field matches the property name, it will automatically be mapped and the Include box will be checked.
--	---

7. Optional: Click **Add** to add a new condition.

On the **Conditions** tab, specify conditions that control when a relationship is created between a source and target entity.

- If you are creating the first condition, the **Logical operator** field will be grayed out. If you are creating a subsequent condition, specify whether this condition should be used in conjunction with previous conditions or if it should be used instead of previous conditions.
- Select the element on which the condition will be based in the **Data source** field.
- Select the field that the condition will be based in the **Field name** field.
- Select the operator for the condition in the **Operator** field.
- Enter the value for the condition in the **Value** field and click **Add**.
- Repeat steps [7.a](#) on page 120 through [7.e](#) on page 120 to add additional conditions.
- When you are done adding conditions, click **Close**.

Completing this step displays the **Add Condition** dialog.

8. Click the **Updates** tab.

9. Select the appropriate action for updating and overwriting properties.

After an relationship is created it can be updated over time when data with the same source and target ID is input into the Write to Graph stage. This selection determines how Write to Graph manages updates.

Option	Description
Always update properties	Properties are always updated with the most recent information. This includes updating with null or empty strings.
Update properties unless all input is null	Properties are always updated unless all input fields associated with the selected properties are null.
Never overwrite properties with empty data	Properties are always updated unless the input is a null or empty string.
Never overwrite non-empty properties	Properties are never updated once populated with non-empty data.

10. Click **Add**.

11. Repeat steps [1](#) on page 119 through [10](#) on page 120 to add additional relationships.

12. When you are done adding relationships, click **Close**.

The Options Tab

Write mode

- Click **Initial load** if you are loading the model for the first time. The model will be locked and unable to be written to by other Write to Hub stages. When this mode is checked, the only available option on the Updates tab in the Add Entity dialog box will be **Never overwrite non-empty properties**. Therefore, if you have multiple input files in your dataflow, they can all create properties, but none of them can update existing properties with new values. This mode provides better performance when initially loading a model. Existing data, if there is any, will be cleared prior to writing.
- Click **Concurrent writes** if you want to allow the model to be written to by multiple Write to Hub stages at the same time. When this mode is checked, the **Clear model before processing** option is disabled and the model is created prior to running the job. If the model does not already exist, it will be created when the stage is closed.
- Click **Exclusive lock** (default) if you do not want to allow the model to be written to by multiple Write to Hub stages. When this mode is checked, properties can be updated after they are created.

Clear model before processing

Check this check box if you wish to remove all existing entities and relationships before processing new data. If this is not selected, new information will be used to update any existing entities and relationships.

Note: Using this option does not alter security settings for the Data Hub module. The model will be recreated, but the security settings will remain the same.

Remove orphaned entities after processing

Check this check box if you wish to remove entities that have no relationships.

Setting Exclusive Lock Timeout Duration

If you select the **Exclusive lock** write mode on the Write to Hub Options tab, or if you apply a centrality algorithm (degree, betweenness, closeness, or influence) to a model, that model will be locked while those processes are running and any action you attempt on that model that requires write access will result in a timeout until those processes are complete.

You can specify how long a process should wait before timing out by modifying the neo4j properties file. The default is 10 seconds, or 10,000 milliseconds.

1. Open the file `<Spectrum™ Technology Platform install path>/server/modules/hub/db/neo4j.properties`.
2. Navigate to the `ha.wait.for.exclusive.lock.timeout` entry.

- Enter the duration, in milliseconds, that you want to wait for the lock to be released or the process to time out. A value of 0 milliseconds will cause an immediate timeout, and leaving this property blank will cause the server to wait indefinitely.

Note: If a model is currently in use with the **Concurrent writes** write mode selected, and a subsequent process is attempted with the **Exclusive lock** write mode selected, the model will be locked and the latter process will time out according to the settings made here.

Output

The Write to Hub stage has one optional outgoing port that collects any records that the dataflow did not process correctly. This is called the Error Port, and records that pass through this port into the sink are considered malformed.

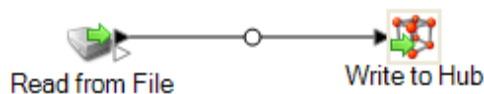
Sample Write to Hub Dataflow

This section describes how to configure a simple dataflow that includes a Write to Hub stage.

The first example uses a flat file for input, and the second example uses an XML file for input; both files include names of employees and their managers, along with other information described in more detail in the following sections. The end result is the same for both dataflows: a model that depicts the reporting structure of a small organization.

Flat Sample

The Write to Hub dataflow that uses a flat file for input looks like this:



Configuring Read from File

The Read from File stage uses a comma-delimited file that includes records with the following fields:

- Employee ID
- Name
- Title
- Manager ID

The input file itself looks like this:

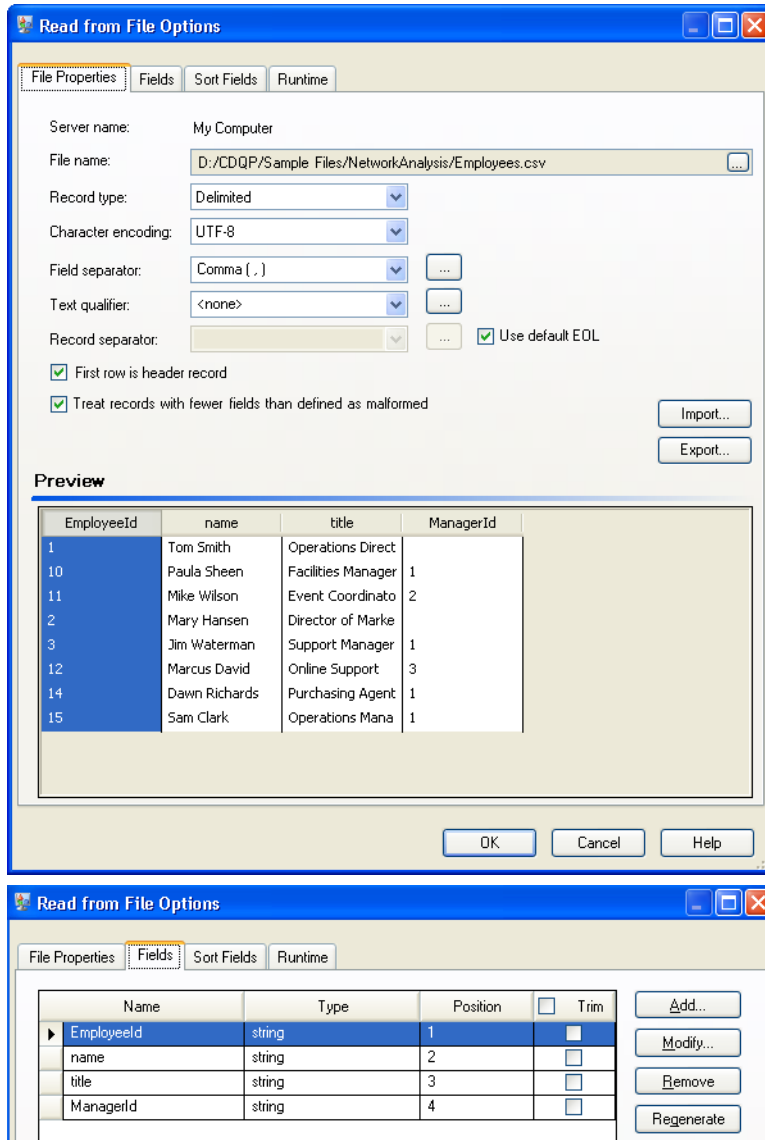
```

1 EmployeeId,name,title,ManagerId
2 1, Tom Smith, Operations Director,
3 10, Paula Sheen, Facilities Manager, 1
4 11, Mike Wilson, Event Coordinator, 2
5 2, Mary Hansen, Director of Marketing,
6 3, Jim Waterman, Support Manager, 1
7 12, Marcus David, Online Support, 3
8 14, Dawn Richards, Purchasing Agent, 1
9 15, Sam Clark, Operations Manager, 1

```

Notice that two employees do not have manager IDs. These employees (Tom Smith and Mary Hansen) are both directors and therefore have no manager in this exercise. All other employees have a number in the ManagerID field that refers to the employee who is their manager. For example, Paula Sheen's record has "1" in the ManagerID field, indicating that Tom Smith is her manager.

The Read from File stage appears as follows when it is configured to work with this input file:

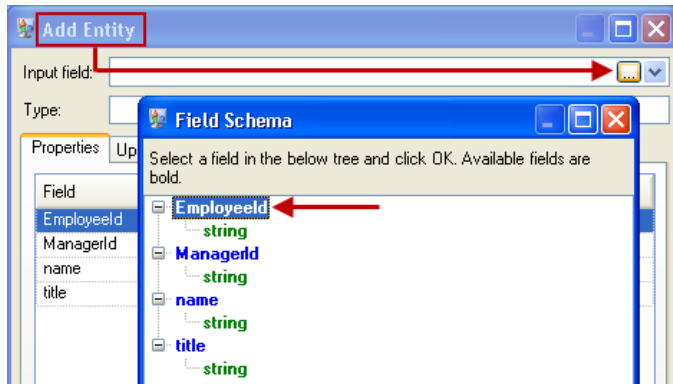


Configuring Write to Hub

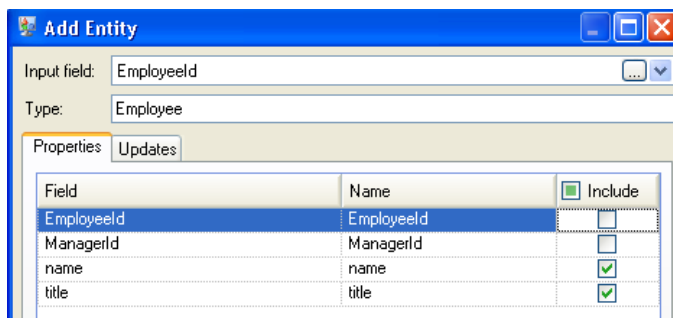
Next we configure the Write to Hub stage. After naming the model "Employees" we configure the stage to include the entities and relationships that will comprise the model.

Because we are creating a model that is similar to an organization chart, our entities are employees who are assigned numeric IDs. The first thing we do on the **Add Entity** dialog box is click the browse

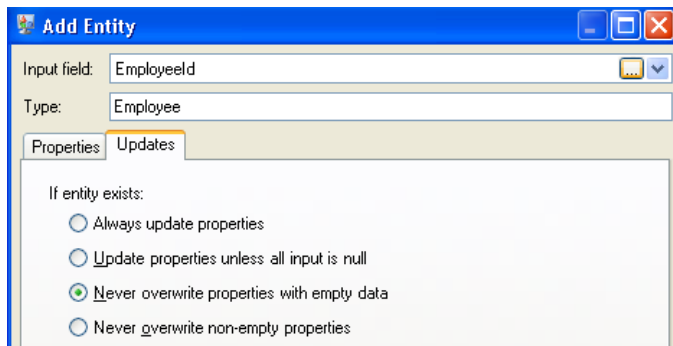
button to access the **Field Schema** dialog box, and then select "EmployeeId" from the list of available fields. This is the first group of entities in our model.



Next, we set the **Type** field to "Employee" and check the boxes for "name" and "title" because we want the information from those fields to be brought in as properties for the EmployeeID entities in the model.

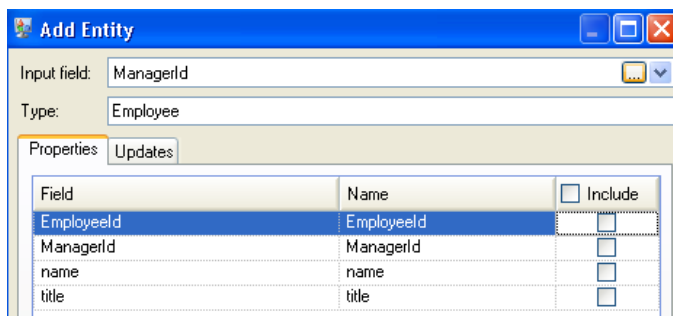


After setting properties for the Employee entity, we configure the processing options. The Updates tab enables you to specify whether properties can be updated in the model once they are in place and if they should overwrite existing data. For instance, in our example, Mary Hansen would be encountered twice because on record 4, she is referred to as an employee, but on record 3, she is referred to as a manager. When Write to Hub processes Mary for the second time, it could potentially overwrite or remove data that was populated as a result of the first time it processed Mary. By selecting **Never overwrite properties with empty data** (which is the default), any updates that occur will create new properties and overwrite existing properties, but they will not blank out properties that were set by the first encounter but missing in the second encounter. This also ensures that the order in which these records are read has no impact on the model.

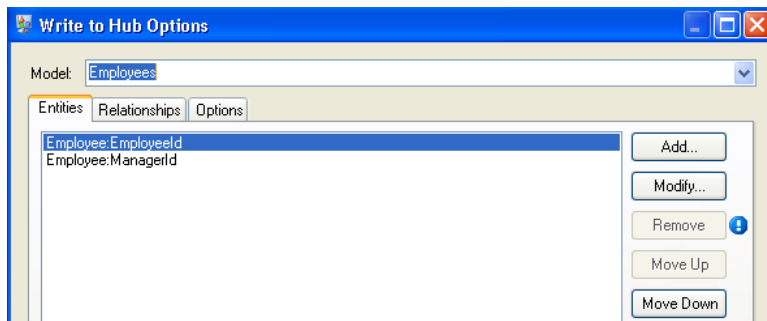


If we selected **Always update properties**, data would always be overwritten and only the last set of property data would be reflected in the model. If we selected **Update properties unless all input is null**, data would always be overwritten unless every field in the new record were blank. Finally, if we selected **Never overwrite non-empty properties**, the first set of data for any given field would be retained, unless that field were blank. In that case, the first set of non-blank data would be retained.

We repeat these steps to add "ManagerId" as the second group of entities in our model. Although ManagerID and EmployeeID are different fields in the input file, both entities' types are set to "Employee." If we set ManagerID to a different type, the model would contain two entities for mid-level managers. For example, Jim Waterman would have an entity as an employee and an entity as a manager. With both entities being set to "Employee" as the type, mid-level managers such as Jim will have just one entity in the model. That entity will have other entities coming into it (from employees) and another entity going out of it (to their respective manager). Note that we do not add properties to the ManagerID entities because the values in those fields (name, title) apply to the employees, not the managers. Also, we accept the **Never overwrite properties with empty data** default selection on the Updates tab.

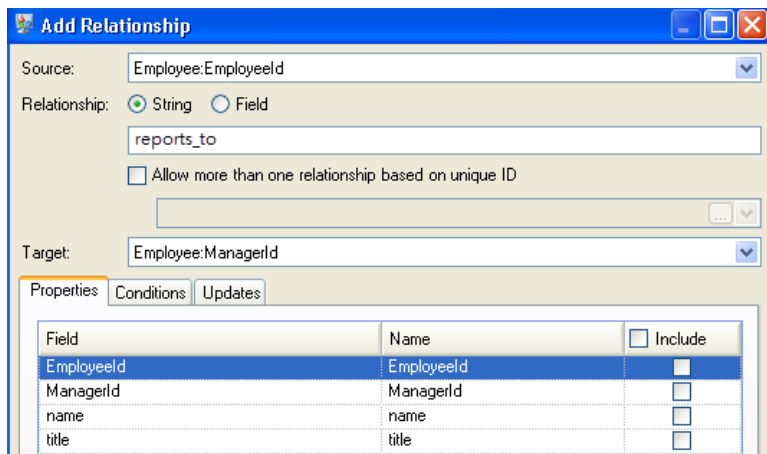


The completed Entities tab for this example appears as follows:

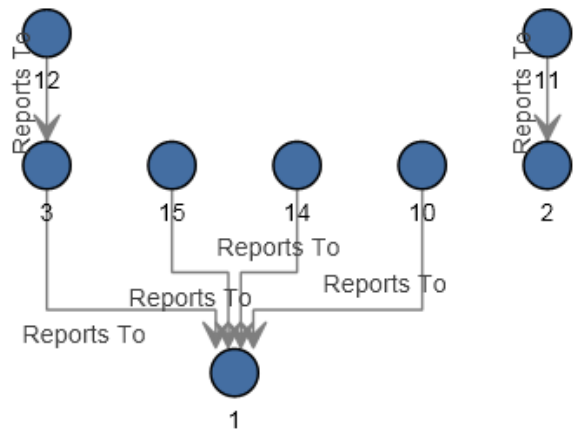


Now we configure the Relationships tab. The first thing we do on the **Add Relationship** dialog box is select the source of the relationship from the list of entities created on the Entities tab. The relationship between our entities reflects the reporting structure (employee to manager); therefore, we select the "Employee:EmployeeID" entity as the source. Next, we select "String" as name of the relationship, and we enter the text "reports_to." After that, we select the target of the relationship from the list of the entities created on the Entities tab; for our example, we select "Employee:ManagerID." If we were using a "manages" relationship instead of a "reports_to" relationship, we would reverse the selections in the source and target fields.

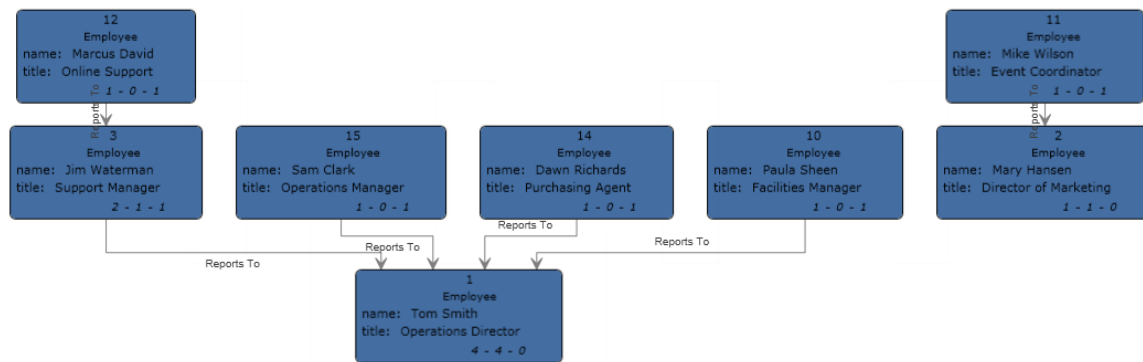
The completed Relationships tab for this example appears as follows:



The configuration of this dataflow is complete and results in the following model, as depicted in the Relationship Analysis Client. This example uses the **Hierarchic** layout with default settings for entities.

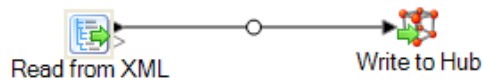


Another way to view this same data is with **Panel** style, as shown below. The benefit of using Panel style is that you can see the properties associated with each entity.



XML Sample

The Write to Hub dataflow that uses an XML file for input looks like this:



Configuring Read from XML

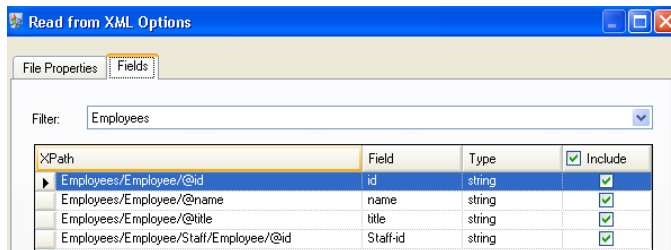
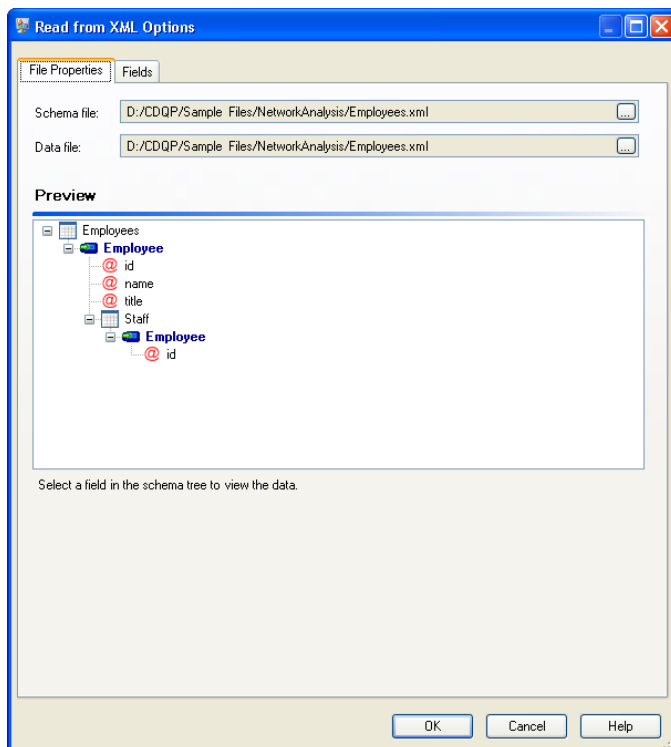
The Read from XML stage uses a hierarchical file that contains the following:

```

<Employees>
  <Employee id="1" name="Tom Smith" title="Operations Director">
    <Staff>
      <Employee id="3"/>
      <Employee id="10"/>
      <Employee id="14"/>
      <Employee id="15"/>
    </Staff>
  </Employee>
  <Employee id="2" name="Mary Hansen" title="Director of Marketing">
    <Staff>
      <Employee id="11"/>
    </Staff>
  </Employee>
  <Employee id="3" name="Jim Waterman" title="Support Manager">
    <Staff>
      <Employee id="12"/>
    </Staff>
  </Employee>
  <Employee id="10" name="Paula Sheen" title="Facilities Manager"/>
  <Employee id="11" name="Mike Wilson" title="Event Coordinator"/>
  <Employee id="12" name="Marcus David" title="Online Support"/>
  <Employee id="14" name="Dawn Richards" title="Purchasing Agent"/>
  <Employee id="15" name="Sam Clark" title="Operations Manager"/>
</Employees>

```

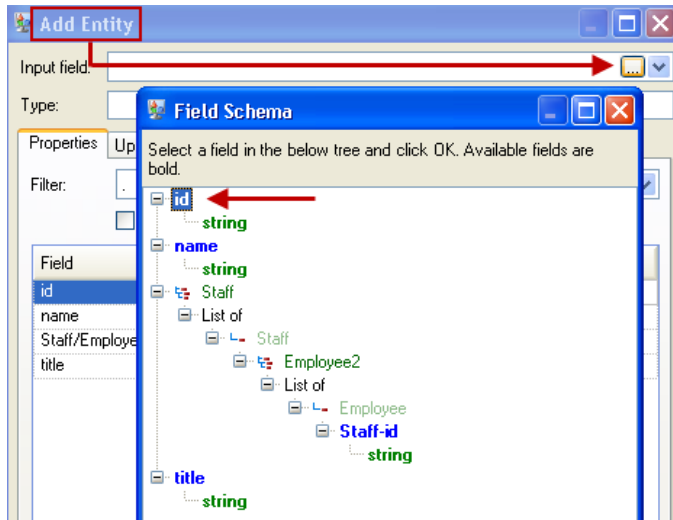
The Read from XML stage appears as follows when it is configured to work with this input file:



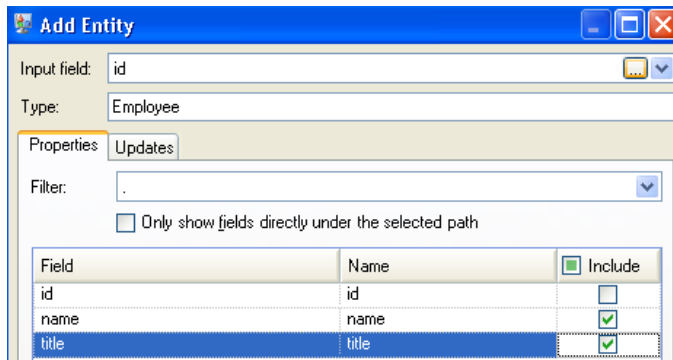
Configuring Write to Hub

Next we configure the Write to Hub stage. After naming the model "Employees" we configure the stage to include the entities and relationships that will comprise the model.

Because we are creating a model that is similar to an organization chart, our entities are employees who are assigned numeric IDs. The first thing we do on the **Add Entity** dialog box is click the browse button to access the **Field Schema** dialog box, and then select "id." This is the first group of entities in our model.

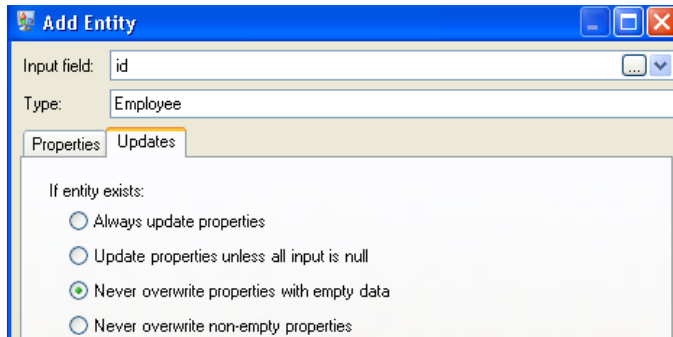


Next, we set the **Type** field to "Employee" and check the boxes for "name" and "title" because we want the information from those fields to be brought in as properties for the ID entities in the model.



After setting properties for the ID entity, we configure the processing options. The Updates tab enables you to specify whether properties can be updated in the model once they are in place and if they should overwrite existing data. For instance, in our example, Mary Hansen would be encountered twice because for ID 2, she is an employee, but for ID 11, she is a manager. When Write to Hub processes Mary for the second time, it could potentially overwrite or remove data that was populated as a result of the first time it processed Mary. By selecting **Never overwrite properties with empty data** (which is the default), any updates that occur will create new properties and overwrite existing properties, but they will not blank out properties that were set by the first encounter

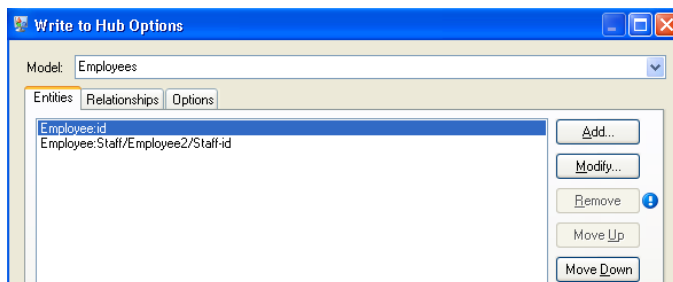
but missing in the second encounter. This also ensures that the order in which these records are read has no impact on the model.



If we selected **Always update properties**, data would always be overwritten and only the last set of property data would be reflected in the model. If we selected **Update properties unless all input is null**, data would always be overwritten unless every field in the new record were blank. Finally, if we selected **Never overwrite non-empty properties**, the first set of data for any given field would be retained, unless that field were blank. In that case, the first set of non-blank data would be retained.

We repeat these steps to add "ManagerId" as the second group of entities in our model. Although ManagerID and EmployeeID are different fields in the input file, both entities' types are set to "Employee." If we set ManagerID to a different type, the model would contain two entities for mid-level managers. For example, Jim Waterman would have an entity as an employee and an entity as a manager. With both entities being set to "Employee" as the type, mid-level managers such as Jim will have just one entity in the model. That entity will have other entities coming into it (from employees) and another entity going out of it (to their respective manager). Note that we do not add properties to the ManagerID entities because the values in those fields (name, title) apply to the employees, not the managers. Also, we accept the **Never overwrite properties with empty data** default selection on the Updates tab.

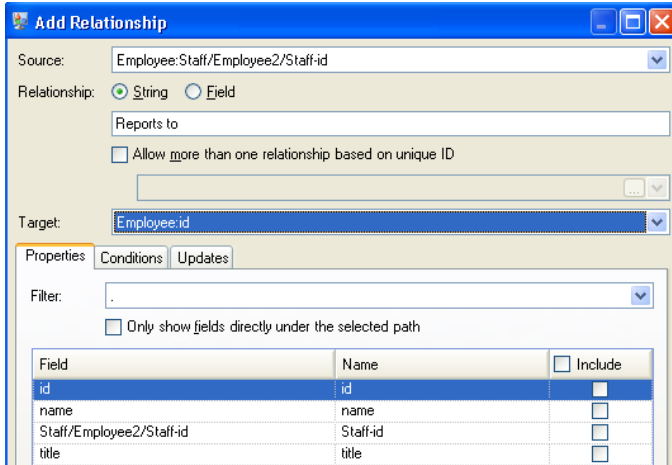
The completed Entities tab for this example appears as follows:



Now we configure the Relationships tab. The first thing we do on the **Add Relationship** dialog box is select the source of the relationship from the list of entities created on the Entities tab. The relationship between our entities reflects the reporting structure (employee to manager); therefore, we select the "Employee:Staff/Employee/Staff-id" entity as the source. Next, we select "String" as name of the relationship, and we enter the text "Reports to." After that, we select the target of the relationship from the list of the entities created on the Entities tab; for our example, we select

"Employee:id." If we were using a "manages" relationship instead of a "reports to" relationship, we would reverse the selections in the source and target fields.

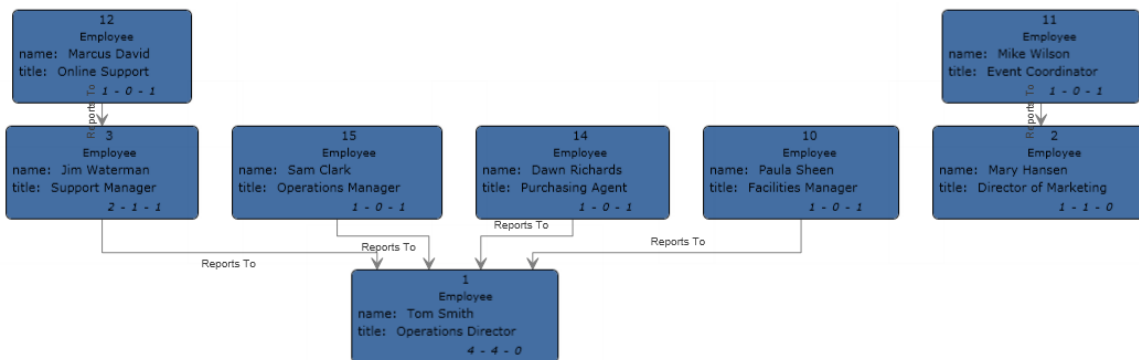
The completed Relationships tab for this example appears as follows:



The configuration of this dataflow is complete and results in the following model, as depicted in the Relationship Analysis Client:



As with the flat sample, this model can also be viewed in **Panel** style, as shown below.



Relationship Analysis Client

The Relationship Analysis Client component of the Data Hub Module is used to view data in a graphical manner that identifies and depicts structure and relationships between entities.

Relationship Analysis Client is web browser tool that provides a visual interface for the following:

- Viewing relationships and hierarchies within the hub
- Viewing viewing hidden or non-obvious relationships
- Creating what-if scenarios
- Performing temporal or geospatial analysis
- Creating rules-driven event triggers
- Running centrality algorithms to determine influence score either against the entire network or against the data being visualized within the client.

You can view and manipulate models that were created in the Write to Hub stage, or you can create new models in the Relationship Analysis Client itself. You can also use the tool to manipulate entities and relationships based on their properties and values.

- For an overview of the Relationship Analysis Client canvas, see [The Relationship Analysis Client Canvas](#) on page 133.
- See [Managing Models](#) on page 139 to read about managing models, including:
 - An overview of data for each model
 - Information on adding centrality measures to a model as well as a list of details that are provided on the Jobs tab after you apply those measures
 - Information on defining monitors to watch for data changes that meet certain conditions
- See [Viewing Models](#) on page 158 to read about all the ways you can view and manipulate the contents of a model, including:
 - The functions of the Model Detail and Query tabs in the Open Model window
 - The buttons on the Canvas Options ribbon
 - Using the Selection and Query tools
 - Using filters
- See [Changing the Appearance of Model Data](#) on page 191 to read about how to change the appearance of elements in a model, including:
 - Viewing a model in multiple ways simultaneously
 - Determining the appearance of entities and relationships
 - Configuring gradient settings to visually depict a range of values
 - Applying Panel Style to a model's entities and show property data
 - Configuring and applying themes to models, which potentially saves you time if you have multiple models that you want to look the same

- Applying layouts that change the visual structure of models based on what you want to emphasize (for example, group structure, hierarchy, locations, and so on)
- See [Analyzing Model Data](#) on page 213 to read about how to analyze data in a model, including:
 - Using the Discovery feature to load specific information based on conditions that you set, then add back in elements that were excluded in the original load.
 - Running centrality algorithms on models
 - Using groups to reveal latent relationships
 - Adding and deleting entities and relationships as well as the properties tied to those elements.

Accessing the Relationship Analysis Client

To access the Relationship Analysis Client:

1. Be sure the Spectrum™ Technology Platform server is running.
2. Open a web browser and go to: `http://<servername>:<port>/hub/`. For example, if your server is named "myserver" and it uses default HTTP port 8080, you would go to:
`http://myserver:8080/hub/`.
3. When the login screen appears, enter the same user name and password that was used to launch Enterprise Designer when you created the model. Press **OK**.
4. Click the **Open** button on the Home tab and select a model from the list. Press **OK**.

You can now use all of the features within the Relationship Analysis Client.






The Relationship Analysis Client Canvas

The initial Relationship Analysis Client canvas consists of the following elements:

- **Canvas Buttons**
- **Canvas Menu**
- **The Overview Pane**
- **The Model Details Pane**
- **The Primary Pane**
- **The Model Components Pane**
- **The Selection Pane**
- **The Query Pane**

Canvas Buttons

The following buttons are always present on the Relationship Analysis Client canvas:

Icon	Description
	Enables you to undo your last action. You can set the number of changes to undo in the Default Options dialog box. The default is 5.
	Enables you to redo the last action that was undone. You can set the number of actions to redo in the Default Options dialog box. The default is 5.
	Enables you to set default options .
	Opens the Event Log. Debug mode provides additional details for what is taking place within the system.
	Allows you to do two things: <ul style="list-style-type: none"> • Move the button bar below the ribbon • Minimize or expand the ribbon

Canvas Menus

The following menus are available on the Relationship Analysis Client canvas.

Note: Press Ctrl-Q-Tab to activate hot keys for menu items.

Menu Item	Description
Home	Provides options for managing models, viewing canvas contents, and selecting model elements.
View	Provides the ability to activate tools and options for filtering by selection and filtering by type.
Data	Provides options for manipulating and analyzing a model.
Styling	Enables you to configure style settings for entities and relationships.
Layouts	Enables you to select from multiple layouts and set options for those layouts.

Overview Pane

The Overview Pane allows you to determine the part of the model on which the primary pane should focus. It provides a complete view of the model with a shaded rectangular area representing the current view port, which represents a subset of the model. Click your cursor near the area of the

model where you want to focus and see the model move in the primary pane. Click the center of your model to have the focus on the entire model.

Note: This pane is on by default but can be toggled off and on using the Overview button on the Tools ribbon of the View tab.

Toolbox Pane

The Toolbox Pane contains icons for use when adding entities or relationships to a new or existing model or when creating models using metadata.

Note: This pane is on by default but can be toggled off and on using the Toolbox button on the Tools ribbon of the View tab.

Model Details Pane

The Model Details Pane shows analysis details about your model. It will show information for the following characteristics:

- **Name**—The name of the model whose data you are viewing.
- **Entities**—Number of entities
- **Relationships**—Number of relationships
- **Is Acyclic**—Formed by a collection of entities and directed relationships, each relationship connecting one entity to another, such that there is no way to start at some entity v and follow a sequence of relationships that eventually loops back to v again.
- **Is Biconnected**—Connected and inseparable, such that if any entity were to be removed, the model would remain connected.
- **Is Bipartite**—Entities can be divided into two disjoint sets U and V such that every relationship connects an entity in U to one in V ; that is, U and V are independent sets. Equivalently, a bipartite model is a model that does not contain any odd-length cycles.
- **Is Connected**—A path exists from one entity in the model to any other entity in the model.
- **Is Cyclic**—Consists of a single cycle, or in other words, some number of entities connected in a closed chain.
- **Is Forest**—An undirected model, all of whose connected components are trees; in other words, the model consists of a disjointed union of trees. Equivalently, a forest is an undirected cycle-free model.
- **Is Multiple Relationship Free**—Is free of two or more relationships that are incident to the same two entities.
- **Is Planar**—Can be drawn in such a way that no relationships are intersecting.
- **Is Rooted Tree**—One entity has been designated the root, in which case the relationships have a natural orientation, towards or away from the root.
- **Is Strongly Connected**—A path exists from each entity in the model to every other entity. In particular, this means paths in each direction; a path from a to b and also a path from b to a .
- **Is Tree**—An undirected model in which any two entities are connected by exactly one simple path.

Note: This pane is on by default but can be toggled off and on using the Model Details button on the Tools ribbon of the View tab.

Primary Pane

The Primary Pane displays the part of the model you chose to focus on in the Overview Pane. You can manually select entities, relationships, and labels in this pane, or you can use the Selection Pane to identify qualifying properties in your data and have the appropriate model elements selected by the Selection tool. If you double-click an element in the model, it will automatically zoom in to and center that element. To zoom in on multiple selected elements, press Ctrl and double-click anywhere in the pane.

Model Components Pane

The Model Components Pane displays model data in grid format. Entity data and relationship data are shown on two tabs. However, the Entities grid also includes data for predecessors and successors to each entity. You can determine which are predecessors and which are successors by the direction the green arrow is pointing.

Let's say you are looking at a model with insurance data, attempting to identify potential fraud. You might have entities representing individuals who sought treatment connected to entities representing doctors or clinics those individuals visited. In the Entities tab of the Model Components pane, if you clicked the plus sign next to an entity to expand the data, you would probably see the doctors and/or clinics that person visited. And if you clicked the plus sign next to one of the doctor or clinic entities, you would see all of the metadata associated with that entity.

In the Relationships grid you will see data grouped by the relationships between entities. Using the insurance fraud example, you might see in the first column of that grid "visited" to represent that the person visited a doctor or clinic.

Note: This pane is on by default but can be toggled off and on using the Model Components button on the Tools ribbon of the View tab.

Selection Pane

The Selection Pane allows you to use the Selection Tool to identify qualifying properties in your data and select model elements (also known as items) that contain those properties. It also shows the properties and relationships for selected elements. If multiple elements are selected, it will show the properties and relationships for the element that was selected last. For more information, see [Using the Selection Pane](#).

Note: This pane is on by default but can be toggled off and on using the Selection button on the Tools ribbon of the View tab.

Query Pane

The Query Pane allows you to use the Query Tool to perform a query. For more information on using the Query Tool, see [Using the Query Tool](#) on page 167.

Note: This pane is on by default but can be toggled off and on using the Map button on the Tools ribbon of the Query tab.

Creating Models

This feature enables you to create models from within the Relationship Analysis Client by manually adding entities and relationships from the Toolbox. It also allows you to add properties associated with those entities and relationships.

As you add elements to your model, the Overview pane, the Model Details pane, the Model Components pane, and the Selection pane update accordingly, reflecting the properties in your model.

Note: Metadata Insights is now the preferred method to create and edit Data Hub models. To add virtual metadata, you can now only use Metadata Insights. For more information, see [Modeling](#) in the *Metadata Insights Guide*.

Adding Entities to a New Model From the Toolbox

This feature allows you to add entities to a new model you are creating using the Relationship Analysis Client Toolbox.

1. From the **Model** ribbon on the **Home** tab, click **New**.
2. If you want your new model to use an existing model's metadata, check the **Copy metadata from model** box and select the model whose metadata you want to use.
3. Click **OK**.
4. From the **Entities** tab in the Toolbox, drag the **New** icon onto the model canvas. The **Add Entity** dialog box appears.

Note: If this is not the first entity you are adding to the model, you may have multiple entity icons to choose from in the Toolbox.

5. Enter a **Label** for the new entity. This label is the name of the entity you are creating, such as "BobSmith" or "AlliedInsurance".
6. Enter the **Type**. This is the type of entity you are creating, such as "Person" or "FirmName".

Note: If this is not the first entity you are adding to the model, you can pick from an existing type or enter a new type.

7. Add **Properties** to the entity by completing the **Name**, **Type**, and **Value** fields. For example, if you were building a model of insurance data, you might include ZIP Code information for providers. In that case, you would enter "ZIP" as the name of the property, followed by "String" as the type, and "78232" as the value.

Note: The Type field will automatically populate when you enter a name, but you can select another type from the drop-down box if necessary.

8. Click **Add**, then **OK**. The new entity appears.
9. Repeat steps 3-7 to add your remaining entities.
10. Click **Save** and enter the name of your new model in the **Save As** dialog box.

Adding Relationships to a New Model From the Toolbox

This feature allows you to add relationships between entities in a new model you are creating using the Relationship Analysis Client Toolbox.

1. With a new model open and two or more entities on the canvas, click the **Relationships** tab in the Toolbox. The **<New>** icon will be selected by default.

Note: If this is not the first relationship you are adding to the model, you may have multiple relationship icons to choose from in the Toolbox.

2. On the canvas, click and drag from the source entity to the target entity and release the mouse. If you chose an existing type of relationship in Step 1, the relationship is now established. If you chose a new relationship in Step 1, the **Add Relationship** dialog box appears.
3. Enter a **Label** for the new relationship. This label describes the relationship between the two entities you are connecting. For example, if you were building a model of insurance data, the label might be "Visited" (between a patient entity and provider entity) or "Billed" (between a provider entity and an insurance company entity).
4. If the two entities have multiple relationships with the same label, you must check **Use unique identifier value** and enter a value that is specific to this relationship. Two entities cannot have multiple relationships with the same label without a unique identifier.
5. Add **Properties** to the relationship by completing the **Name**, **Type**, and **Value** fields. For example, if you were building a model of insurance data, you might include ZIP Code information for providers. In that case, you would enter "ZIP" as the name of the property, followed by "String" as the type, and "78232" as the value.

Note: The Type field will automatically populate when you enter a name, but you can select another type from the drop-down box if necessary.

6. Click **Add**, then **OK**. The relationship appears.

Managing Models

The Model Management feature provides an overview of data for each model without opening that model. It enables you to view model information, add and update centrality measures using algorithms, define monitors, and remove models. It also provides detailed information for the jobs run in the Model Algorithms tab.

- [Viewing Model Information](#)
- [Applying Centrality Measures to a Model](#)
- [Monitoring Models](#)
- [Querying Models](#)
- [Removing Models](#)
- [Viewing Job Information](#)

Follow these steps to access model management:

1. With the Home tab selected, click the **Manage** button.
2. Select a model from the list in the **Models** dialog box by clicking it.

Models Tab

The Models tab shows the number of entities and relationships for each model. If you click anywhere in a particular model's row (for example, in the Name, Entities, or Relationships columns), that model will open. If you click the column header, the model data will sort in ascending order for that column. If you click the column header twice, the model data will sort in descending order for that column.

Model Detail Tab

The Model Detail tab shows the number of entities and the number of relationships for a selected model. If you click the column header, the model data will sort in ascending order for that column. If you click the column header twice, the model data will sort in descending order for that column.

This tab also provides model metadata in grid format. Entity data and relationship data are shown on two tabs. The Entities grid includes data for predecessors and successors to each entity as well as the properties associated with each entity. The Relationships grid includes the types of relationships that exist in the model and the properties associated with each relationship.

Let's say you are looking at a model with insurance data, attempting to identify potential fraud. You might have entities representing individuals who sought treatment connected to entities representing doctors or clinics those individuals visited. In the From Relationships section of the **Entities** tab, if you clicked the plus sign next to an entity to expand his or her data, you might see "Person" in the From column, "Visited" in the Type column, and "Doctor" in the To column, followed by the number of times someone visited a doctor.

In the Relationships grid you will see data grouped by the relationships between entities. Using the insurance fraud example, you might see "Visited" in that grid, and if you clicked the plus sign next

to that relationship to expand its data, you would probably see "Claimant" in the "From" column, "Visited" in the "Type" column, "Provider" in the "To" column, followed by the number of instances of that relationship in the model in the "Count" column.

The Entities Tab

The **Entities** tab includes a list of each entity in your model along with the number of times that entity occurs in the model.

If you click the plus sign next to an entity, the information described below will appear for that entity. If you click the plus sign next to "All," the information described below will appear for all entities.

Entities		Relationships	
Entity Type	Count		
All	55		
Person	33		
From Relationships			
From	Type	To	Count
Person	Roomate	Person	20
Person	Father-in-law	Person	1
Person	Transfer Money	Person	5
Person	Provided Money	Person	6
Person	Supervisor	Person	21
Person	Hijacked	Event	19
Person	Meeting	Person	15
Person	Wife	Person	1
Person	Friend	Person	13
Person	Organizer	Event	4
Person	Lived	Place	3

- The **From Relationships** tab shows information for relationships in your model where the entity type you have selected is in a "from," or source, position. In this tab, the **From** column will always be the entity whose data you are viewing. Information is then organized by **Type**, which depicts the relationship between the From and To entities. In the example above the type represents the relationship between two persons. The **To** column includes entity types that are in the "to," or target, position in the relationship. In the example above, we are looking at Person entities coming into other Person entities. This tab also provides the **Count** for each relationship type, indicating how many occurrences there are of that relationship type for the entity whose data you are viewing. In our example, there are six incidents of persons who provided money to another person and two incidents of a person being a father to another person.
- The **To Relationships** tab shows information for relationships in your model where the entity type you have selected is in the "to," or target, position. The **From** column includes entity types that are in the "from," or source, position in the relationship. As with the previous tab, information is

then organized by **Type**. In this tab, the **To** column will always be the entity whose data you are viewing. The **Count** column is the same as described for the previous tab.

- The **Properties** tab shows information for all the properties associated with the entity whose data you are viewing. This tab also shows the **Type** of each property as well as the **Count**, which represents the number of occurrences there are of each property for the entity whose data you are viewing. Any given entity may not include every single property listed, but at least one entity will contain each property listed.

The Relationships Tab

The **Relationships** tab includes a list of each entity in your model along with the number of times that entity occurs in the model.

If you click the plus sign next to a relationship, the information described below will appear for that relationship. If you click the plus sign next to "All," the information described below will appear for all relationships.

Entities		Relationships	
Relationship Label	Count		
+	All	192	
-	Attended	56	

Relationships		Properties		
From	Type	To	Count	
>	Person	Attended	Event	19
	Person	Attended	Place	37

- The **Relationships** tab shows information for the selected relationship in your model. In this tab, the **From** column represents the source entity of the relationship. Information is then organized by **Type**, which depicts the relationship between the From and To entities. In the example above the type represents the relationship between two persons. The **To** column includes entity types that are in the "to," or receiving, position in the relationship. In the example above, we are looking at Person entities coming into Event and Place entities. This tab also provides the **Count** for each relationship, indicating how many occurrences there are of that relationship type for the entity whose data you are viewing. In our example, there are 19 incidents of persons who an event and 37 incidents of a person attending a place.
- The **Properties** tab shows information for all the properties associated with the relationship whose data you are viewing. This tab also shows the **Type** of each property as well as the **Count**, which represents the number of occurrences there are of each property for the relationship whose data you are viewing. Any given relationship may not include every single property listed, but at least one relationship will contain each property listed.

Model Audit Events Tab

The Model Audit Events tab enables you to view the activity that has taken place for a model. This feature returns create, read, update, and delete events for the model and its metadata. Depending

on which button you select, you can return information for the model itself or for the properties of entities and relationships within the model. The table below details the information returned for each option.

Model events	<p>Displays create, read, update, and delete activity for the model itself, including the following:</p> <ul style="list-style-type: none"> • Timestamp—When the activity took place • User—User name of the person who generated the activity • Object Type—The type of object that had the activity; in this case, model • Operation—The activity that occurred: create, read, update, or delete • Object ID—The name of the model
Metadata events	<p>Displays create and delete activity for entity types and relationship labels, as well as add and delete activity for properties of entity types and relationship labels, including the following:</p> <ul style="list-style-type: none"> • Timestamp—When the activity took place • User—User name of the person who generated the activity • Object Type—The type of object that had the activity; in this case, metadata • Operation—The activity that occurred: create, read, update, or delete • Object ID—A combination of the model name, the parent element type, and the property name (for example, "Fraud[Entity].Date") • Parent Type—The element type that contained the property: entity or relationship • Parent—A combination of the model name, the parent element type, and either the <code>_stp_type</code> (entities) or <code>_stp_id</code> (relationships) of the parent element (for example, "Fraud[Entity].Person" or "Fraud[Relationship].Treated") • Metadata Type—The type of metadata that had the activity: entity, relationship, or property • Property Type—The type of property that had the activity, when applicable: string, double, date, etc.

When you leave the Model Audit Events tab, the Relationship Analysis Client will retain the last settings you made, including the level of data, the column size, the column positions, filtering, grouping, and sorting.

Model Modifications Tab

The Model Modifications tab enables you to alter a model either by applying centrality using algorithms or by making schema changes.

Algorithms

The Algorithms function allows you to apply centrality to a model. Centrality is a way of measuring the importance and significance of individual entities and relationships. When you run centrality algorithms, the higher the value, the more important the element.

1. Click the **Algorithm** drop-down to select the kind of centrality measure you want to apply to your model.

- **Betweenness**—Used to identify entities that control the information flow between different parts of the network.
 - **Closeness**—Used to identify entities that may have best access to other parts of the network and visibility of activities within the rest of the network.
 - **Degree**—Used to identify entities that have the most direct links to others.
 - **Influence**—Used to identify entities that have strong influence in the network due to their direct links to other highly active or well-connected entities.
2. Select the direction in which you want to apply the algorithm:
 - **Incoming**—The results will be based on relationships coming into the entity.
 - **Outgoing**—The results will be based on relationships going out of the entity.
 - **Both**—The results will be based on incoming and outgoing relationships.
 3. If you are using an Influence algorithm, slide the **Precision** scale to determine how precise the results should be. A lower precision will return more accurate results, but the algorithm will run more slowly.
 4. If you are using a Closeness algorithm, click the appropriate button for the way in which you want results to be returned:
 - **Standard**—Results are based on the number of attachments, or relationships, an entity has as well as the reverse of the sum of shortest paths to each entity.
 - **Dangalchev**—Results are based not only on the number of entities linked to another entity but also the number of relationships in each of the linked entities.
 - **Opsahl**—Results are based on the sum of reversed shortest paths to each entity.
 5. Click the **Use relationship property as weight** if you want to measure how unfavorable a relationship is, and select the relationship property you want to use from the **Property** drop-down. In this case, a higher value indicates a negative association.
 6. Click the **Low values are more significant** box if you are using a relationship property as weight and that property is one where a lower value is considered better than a higher value. For example, if the property is some sort of ranking system, typically 1, or 1st, is the best value. Another example is if the property is distance, and you are trying to determine the shortest route: 5 miles is considered better than 10 miles.
 7. Click the **Override default output property name** if you want the output property name to be something other than the algorithm you selected. Then enter the new name in the **Property** field.
 8. Click **Run**.
 9. Click the **Jobs** tab to view job details after running the algorithm. It will provide the job ID, the model name, the algorithm used, the status of the job, the start time, and the end time. Be aware that the bigger the model, the longer it takes the algorithm to run; watch the Status column to determine if a job is still running.

Note:

Results are not stored over time. If you close Relationship Analysis and reopen it, the information on the Jobs tab is cleared.

Schema Changes

The Schema Change tool allows you to modify a populated model's metadata. Operations performed here can affect elements across the entire model. For example, if you delete an entity type, every instance of that entity type in the model will be deleted and could result in a loss of data.

When you make a schema change, a job will run in the background while the model is updated. You can view the details and status of that job by clicking the Model Management Jobs tab.

For **Modify Index Type**:

1. Select the **Property** whose type you are changing.
2. Select the new **Index type**.
 - None—The data will not be stored in the index.
 - Exact—The data will be indexed as-is, with case sensitivity.
 - Case Insensitive—The data will be indexed without case sensitivity.
3. Click **Run**.

For **Rename Entity Type**:

1. Select the **Entity type** you want to rename.
2. Enter the **New name** of the entity type. Names must start with a letter or an underscore and can only contain letters, numbers, underscores, or periods.
3. Click **Run**.

For **Rename Entity Type Property**:

1. Select the **Entity type** whose property you want to rename.
2. Select the **Property** you want to rename.
3. Enter the **New name** of the property. Names must start with a letter or an underscore and can only contain letters, numbers, underscores, or periods.
4. Click **Run**.

For **Rename Relationship Label**:

1. Select the **Relationship Label** you want to rename.
2. Enter the **New name** of the relationship label. Names must start with a letter or an underscore and can only contain letters, numbers, underscores, or periods.
3. Click **Run**.

For **Rename Relationship Label Property**:

1. Select the **Relationship label** whose property you want to rename.
2. Select the **Property** you want to rename.
3. Enter the **New name** of the property. Names must start with a letter or an underscore and can only contain letters, numbers, underscores, or periods.
4. Click **Run**.

For **Delete Entity Type**:

1. Select the **Entity type** you want to delete.
2. Click **Run**.

For **Delete Entity Type Property**:

1. Select the **Entity type** whose property you want to delete.
2. Select the **Property** you want to delete.
3. Click **Run**.

For **Delete Relationship Label**:

1. Select the **Relationship label** you want to delete.
2. Click **Run**.

For **Delete Relationship Label Property**:

1. Select the **Relationship label** whose property you want to delete.
2. Select the **Property** you want to delete.
3. Click **Run**.

Model Monitors Tab

The Relationship Analysis Client tool allows you to monitor your models for data changes that meet certain conditions. You can set monitors for entities or relationships.

Conditions for entity monitors are based on entity properties, which are evaluated against literals, or absolute values. For example, you could set a condition on a property called "Nationality" and have a service run when an entity's nationality is "Canadian".

Conditions for relationship monitors are based on relationship properties as well as properties for the source and target entities attached to those relationships. As with entity monitors, these properties can be evaluated against literals, but they can also be evaluated against the properties themselves. For example, you could set a condition to look for when a relationship property is equal to that same property on the source or target entity. Likewise, you could set a condition to look for when a source entity's property is equal to that same property on a target entity.

To implement monitoring, you need to assign a name to your monitor, set the conditions under which you want to take action, and set up either an email notification or a service activation.

Note: Before creating an Email Action monitor, you must first set up **Notifications** in Management Console.

Follow these steps to add a monitor:

1. Click **Add** and select whether you want to create an entity monitor or a relationship monitor.
2. Enter a **Name** for the monitor you are creating.
3. Deselect the **Enabled** check box if you do not want to activate the monitor at this time. Otherwise leave it selected. Note that you can enable and disable monitors at any time, not just when they are created or modified.

4. Click **Add Conditions** to build the scenarios under which you want to take action
 - a) If you are creating a relationship monitor, select the source type (Property, Source Entity, or Target Entity).
 - b) Select the property whose data you are monitoring.
 - c) Select the operand you are using to create the condition.
 - d) Enter the value at which an email or service should be activated. Note that if the property you are monitoring is a date, time, or datetime type, and you have selected "Increased By" or "Decreased By" as the operand, the value in this field must represent a time span using the following variables, either alone or combined with a space in between: "y" for years, "w" for weeks, "d" for days, "h" for hours, "m" for minutes, and "s" for seconds. For example, if you had a date field with a value of 10/05/13 and you wanted to set a condition to be activated when the date reached 12/19/13, you would enter "2m 2w". Or, if you had a time field with a value of 3:00:00 PM and you wanted to set a condition to be activated when the time reached 11:00:00 PM, you would enter "8h". Alternatively, if you had a datetime field with a value of 04/18/12 10:30:00 AM and you wanted to set a condition to be activated when the date reached 10/28/13 and the time reached 4:00:00 PM, you would enter "1y 6m 2w 3d 5h 30m".
 - e) If you do not want the monitor to consider case when evaluating model data, click the **Ignore Case** check box.
5. Repeat Step 5 to add more conditions. Note that you can move the conditions up and down in order of priority by selecting the condition and using the up and down arrows.
6. Under **Actions**, click **Add** and select the type of action you want to take.
7. Complete the steps for **Email Actions** or **Service Actions**.
8. If you are creating a service action that uses saved values, check **Execute actions after transaction completes**. Check **Execute actions asynchronously** if the action triggering the monitor is not tied to it, allowing the calling process to complete and return and the monitor to run in an asynchronous, parallel manner.

Creating Email Actions

1. Enter a valid email address in the **To:** field.
2. Enter what you would like the subject line of the Email Action to be in the **Subject:** field. Alternatively, you can right-click to use the **Cut**, **Copy**, and **Paste** functions or select **Insert** to select from a list of tokens.

Note: Pressing **\$** will also activate the list of tokens.

3. Enter the text you would like to appear in the body of the Email Action. You can manually enter a message of your choice or use the list of tokens to build a message. Using tokens allows you to format the email using variables that are replaced with actual values at runtime. As you add tokens, you will see the code update dynamically in the **Message** box of the Email Action.
4. Click **OK** twice. Your email monitor is now created.

Creating Service Actions

1. Select the service you want to activate when the monitor finds the data and conditions you are looking for. Both Spectrum and custom services are available in the drop-down list. The fields in that service will appear in the **Field Mappings** table.
2. Map fields from your model to input fields in the service.
 - The **Source Type** defaults to "Entity Property" or "Relationship Property," depending on whether the service action is for an entity or a relationship; you can also select "Literal" to tell the monitor to look for specific text. If the service action is for a relationship, you can also select from "Source Entity" or "Target Entity" as the source type.
 - Select the appropriate property in the **Source** column drop-down to map it to the field shown in the **Field** column. If you selected "Literal" as the Source Type, enter the exact text you want the monitor to look for.
3. Click **Options** to override the default options for the service you are activating. You can sort the fields by Name or Value.

Note: Only exposed options will appear in this table.

4. Click **OK** twice. Your service monitor is now created.

Model Queries Tab

The Model Queries tab in Model Management enables you to create queries that return data from a selected model. After you create a model query, you can run it by opening the model from the Query tab on the Open Model dialog box. You can also run a query that was created in Model Management from the Query pane on the Relationship Analysis Client canvas. It will appear in the Query drop-down.

Creating Queries

When creating a query, you can use the query builder provided, or you can write a custom query.

You can also select from or modify existing queries that are listed in the Query drop-down. When you modify existing queries, you must either save the changes and overwrite the existing query (using the same name) or save the changes and give the modified query a new name. Otherwise, the changes will be temporarily saved as a custom query and will be accessible only from Open Model and not from Model Management. The original query will not be modified, and the new, custom query will be available only as long as the application is open or until you use a different query for that model. If you apply a different query to the model or close the application, changes to the query will be lost.

Follow these instructions to use the query builder:

1. Choose whether you want to **Select elements** included in the query results, **Add elements** included in the query results, or **Show result** by highlighting the results on the canvas.
2. Check the **Include results from partial traversals** box to include the results from each step in the query. Leave the box unchecked to include only the results that meet the requirements of the last step. For example, let's say that you are looking at a model that depicts world-wide terrorist activity for the events leading up to September 11, 2001, and you want to return data

for any meetings that both Osama bin Laden and Mohamed Atta attended. Your query might include the following steps:

- An initial step that includes an exact search type for entities with an `_stp_id` property that has a literal value of "Person:Osama bin Laden"
- An Entity to Relationship step of connected with a relationship label of "Meeting"
- A Relationship to Entity step of connected with a condition of an `_stp_id` property that contains "Atta"

If you leave the **Include results from partial traversals** box unchecked, the query would return a single meeting between Osama bin Laden and Mohamed Atta. If you check the box, the query would return all meetings between Osama bin Laden and anyone else in the model. The additional records would be returned because the second step looks for meetings attended by Osama bin Laden and checking the box returns results from each step in the query.

3. Leave the **Include results from circular traversals** box checked to include elements that occur more than once in each traversal. Uncheck the box to include those elements just once in each traversal. For example, let's say you are using the same model mentioned in step 1, and you initially want to return data for any meetings that Mohamed Atta attended but once you have those results, you want to see all attendees of a particular meeting. Your query might include the following steps:

- An initial step that includes an exact search type for entities with an `_stp_id` property that has a literal value of Person:Mohamed Atta, which will return his entity
- An Entity to Relationship step of connected with a relationship label of "Attended", which will return all events that he attended, including a meeting in Kandahar
- A Relationship to Entity step of connected with a condition of an Event property that contains "Kandahar", which will return just the Kandahar meeting
- An Entity to Relationship step of connected with a relationship label of "Attended", which will return relationships that connect to three other entities who attended the meeting in Kandahar and may or may not return the (already traversed) relationship that connects to Mohamed Atta, depending on whether you use this option.

If you leave the **Include results from circular traversals** box checked, Mohamed Atta's relationship will be returned in addition to those for the other three attendees. If you uncheck the box, Mohamed Atta's relationship will not be returned because that relationship (Person:Mohamed Atta->Attended->Meeting:Kandahar) was already traversed in the first step of the query.

4. Leave the **Limit results to** box checked and enter a number to specify the total maximum number of entities and relationships to return from the query. The default is 5000. The number entered here applies to unique elements, so if the same element appears in multiple results, they will count as one result. To avoid this scenario, use the dedup function discussed at the end of this topic; it will remove duplicate results in the output. If your root step returns a list and you are querying a large model, we strongly suggest entering a limit in this field to prevent the server from becoming unresponsive.

Note: Limits can be set here or as Query Result Limits set in Relationship Analysis Client [General Settings](#); if the limits are not the same, the lower limit will prevail.

5. Complete the Selection tab.

- Click **All entities**, **All relationships**, or **Specify starting entities** to identify what you want to query against. The Specify starting entities selection allows you to determine at what point in the model you want to begin your search. For instance, if you are looking at a model that depicts world-wide terrorist activity during specific years, you might have country names for entities. Rather than query against the entire model, you might want to look at activity just in Afghanistan. In this case you could select "All" as the **Search type** and "Country" as the **Property name**, leave **Literal** selected, and enter "Afghanistan" as the **Property value**.

Note: The value in the **Property value** field is case sensitive if that field was indexed with the Exact type selected (versus Case insensitive). For more information on selecting types when indexing fields, see [The Entities Tab](#) on page 117.

You could also click **Field** and select "Location" as the Property value, for example, rather than entering a specific value. If you select Field, an **Input Data** grid containing the name of the field you just selected will appear under the query name along with a cell where you can enter the default value. If you reuse this query elsewhere, you can use the default value you provided in this step, or you can override the default at that time.

- If you clicked **All entities** or **Specify starting entities**, select the **Entity types** for your query. You can choose to query selected types or all types. Click **Select None** to deselect any selected types. In addition to returning a more focused set of results, selecting entity types will affect other factors such as which properties and fields are available in the first step of the query, which directions, entity types, and relationship labels are available in subsequent steps of the query, and so on.
- If you clicked **Specify starting entities**, select the **Search type**:

Exact	Searches the index for data that matches exactly what you enter on the Selection tab, including casing. As with property values, the value here is case sensitive if that field was indexed with the Exact type selected; if case sensitivity was used and you search for "texas" while your data includes entries of "Texas", they will not be returned.
Starts with	Searches the index for data that contains text beginning with what you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "tech" or "tec" would be considered a match for a property value containing "Technical", "Technology", "Technologies", or "Technician".

Ends with	Searches the index for data that contains text ending with what you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "Emirates" or "tes" would be considered a match for a property value containing "United Arab Emirates".
Contains	Searches the index for data that contains the text that you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "Pitney" and "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc."
Any	Searches the index for data that contains any of the text that you enter on the Selection tab. For example, a literal property value of "Austin Tex" would be considered a match for a property value containing "Texarkana" or "Stephen F. Austin University".
All	Searches the index for data that contains all of the text that you enter on the Selection tab. For example, a literal property value of "Allstate claim 2013" would be considered a match for a property value containing "filed claim with Allstate June 2013", as would literal property values of "all state" or "all 13".
Between	<p>Searches the index for data that falls within a range that you specify on the Selection tab. When you use this search type, you must select a Property name that contains date, time, date/time, or numeric data. All numeric data types are supported with the exception of BigDecimal. For example, the following specifications would return all entities with StartDate values occurring in the year 2000:</p> <ul style="list-style-type: none">• A Property name of "StartDate" that is a Date type• A literal Start value of "1/1/2000"• A literal End value of "12/31/2000"

Fuzzy

Searches the index for the text you enter on the Selection tab but allows for some differentiation (missing letters, extra letters, or substitutions of letters). The amount of differentiation that is acceptable to still be considered a match depends on what you enter in the **Metric** field. This figure must be greater than zero and less than one; in other words, it must range from ".1" to ".9". For example, if you search for "Barton" and enter ".9" as the metric, the search will return records with "Carton" (replaces B with C), "Bartons" (adds s), and "Baton" (removes r), because all of these words are one character different from the search word "Barton".

Wildcard

Searches the index for the text you enter on the Selection tab but allows for a single wildcard character or a wildcard character sequence. Supported wildcard characters include the question mark (?), which matches any single character, and the asterisk (*), which matches any character sequence (including blanks). For example, if you search cities in Texas for "Aus*", the search will return records with "Austin", "Austonio" and "Austwell". If you conduct a similar search for "Aust??", only "Austin" will be returned because each question mark represents a single character and the other two city names have more characters in their name.

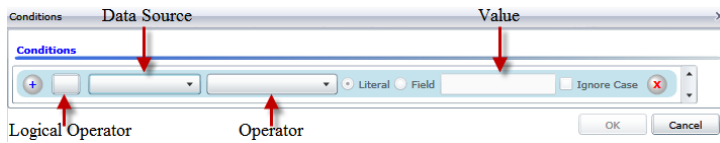
Note: A query that includes an asterisk wildcard as the first character in the search string may result in a lengthy response time.

- If you clicked **Specify starting entities**, select the **Property name** from the drop-down list. This list contains all properties associated with the entities and relationships that make up the model.

Note: You can only query properties that have been indexed; non-indexed properties will not appear in the Property name drop-down.

- If you clicked **Specify starting entities**, and selected a search type other than "Between", select the **Property value**. You can click **Literal** and enter a text string to be used in the search. Alternatively, you can click **Field** and select the field whose data should be searched; if you choose this option you will also need to enter a value in the Input Data grid.
- If you clicked **Specify starting entities**, and selected the "Between" search type, select the **Start value** and **End value** to enter the range. You can click **Literal** and enter a value to be used in the search. Alternatively, you can click **Field** and select the field whose data should be searched; if you choose this option you will also need to enter a value in the Input Data grid.

6. Complete the **Conditions** tab if you want to place additional constraints on the query. The Conditions tab has four entry fields:



- If you are creating the first condition, the **Logical operator** field will remain empty. If you are creating a subsequent condition, specify whether this condition should be used in conjunction with previous conditions ("And") or if it should be used instead of previous conditions ("Or").
- Select the property on which the condition will be based in **Data source** field.
- Select an operator for the condition that is appropriate for the data type in the **Operator** field:

Equals	Searches model elements for properties with values that match exactly what you enter in the Value field. This can be a numeric value or a text value.
Not Equals	Searches model elements for properties with values that have any value other than what you enter in the Value field. This can be a numeric value or a text value.
Exists	Searches the model elements for the existence of the property that you select in the Data Source field.
Does not Exist	Searches the model for elements that do not contain properties that you select in the Data Source field
Is Blank	Searches model elements for properties that contain no data. If a property value is blank, that element will be returned. This can be a numeric value or a text value.
Is Not Blank	Searches model elements for properties that contain any data. If a property value is not blank, that element will be returned. This can be a numeric value or a text value.
Greater Than	Searches model elements for properties whose values are greater than the value you specify. This can be a numeric, date, date/time, or time value.
Greater Than or Equals	Searches model elements for properties with numeric values that are greater than or equal to the value you specify. This can be a numeric, date, date/time, or time value.
Less Than	Searches model elements for properties with numeric values that are less than the value you specify. This can be a numeric, date, date/time, or time value.

Less Than or Equals	Searches model elements for properties with numeric values that are less than or equal to the value you specify. This can be a numeric, date, date/time, or time value.
Contains	Searches model elements for properties with values that contain what you enter in the Data Source field. The search does not need to be a complete word. For example, a literal property value of "Pitney" or "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc." This can be a numeric value or a text value.
Does not Contain	Searches model elements for properties that do not contain what you enter in the Data Source field. The search does not need to be a complete word. For example, a literal property value of "Pitney" or "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc.," so "Pitney Bowes Software" would not be returned. This can be a numeric value or a text value.
Starts With	Searches model elements for properties whose values start with the text you enter in the Data Source field. For example, if you enter "Van" for the LastName field it would return results with "Van Buren", "Vandenburg", or "Van Dyck".
Does Not Start With	Searches model elements for properties whose values do not start with the text you enter in the Data Source field. For example, if you enter "Van" for the LastName field it would not return results with "Van Buren", "Vandenburg", or "Van Dyck" but would return results with "Eddie Van Halen".
Ends With	Searches model elements for properties whose values end with the text you enter in the Data Source field. For example, if you filter for records that end with "burg" in the City field, it would return results with "Gettysburg", "Fredricksburg", and "Blacksburg".
Does Not End With	Searches model elements for properties whose values do not end with the text you enter in the Data Source field. For example, if you filter for records that end with "burg" in the City field it would not return results with "Gettysburg", "Fredricksburg", and "Blacksburg" but would return results with "Burgess".
Match Regular Expression	Searches the model elements for properties having a regular expression match for what you enter in the Data Source field. Regular expression matches identify strings of text of interest, such as particular characters, words, or patterns of characters. The value field should contain a valid regular expression pattern.
Is Roughly Similar To	Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing

letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with search metric of .5.

Is Similar To Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with search metric of .6.

Is Very Similar To Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with Search Metric of .7.

d. In the drop-down box following the list of operators:

- Select **Literal** and enter a text string the fourth box (called the **Value** field) to be used in the query.
- Select **Field** and then select the field whose data should be searched in the **Value** field.
- Select a previous step (such as "**Root**" or "**Step1**") and then a property in the **Value** field to compare property values for the current step against values returned in a previous step. (Note that if you named the output on the Output tab of previous steps, those names will appear in the drop-down rather than "Root" or "Step1".) In this case, the properties shown in the Value field are based on properties for the previous step. For example, if you knew the name of one person (Mohamed Atta) who attended a particular event (a meeting in Kandahar) but wanted to know the names of the other attendees, you could create the following query that includes a property value comparison:

- A root step that looks for an entity type Person with an `_stp_id` of that contains "Mohamed"
- An Entity to Relationship step with a relationship label of "Attended"
- A Relationship to Entity step with a condition that includes an Event that contains "Kandahar"
- An Entity to Relationship step with a relationship label of "Attended" plus a condition that this step's `_stp_id` does not contain the same `_stp_id` value that was found in the root step.

This query will find that Mohamed Atta attended an event in Kandahar and that it was also attended by three other people whose `_stp_id` value is not "Mohamed".

- e. Click **Ignore Case** if the query results can be either upper or lower cased.
- f. Repeat steps a through e to add additional conditions.
- g. Click **OK**.

If, for example, you want to target terrorist activity in Afghanistan between 2001 and 2010, you would create two conditions. First, you would select "date" for the **Property name**, then "Greater Than or Equals", leave **Literal** selected, and then enter "2001". You would follow this with a second condition set to "And" that also uses "date," then "Less Than or Equals", then "2010". Alternatively, you could click **Field** and select "Date" rather than entering a specific value. Add, delete, or change the order of conditions by using the icons on either side of the conditions. Click **Ignore Case** if the query results can be either upper or lower cased.

7. Complete the **Output** tab to define how you want your output to appear.

- Click the **Include in results** box if you want the results from this step to be included in the output.

Note: This box must be checked for the last step in any series; therefore, if there is only one step you cannot uncheck this box.

- Click **Specify name** and enter text in the **Name** field to provide a name for this step in the output. Click **List** to use this entry as the name and type of the field in hierarchical output; leave it unchecked to have this entry added as a prefix for all output fields. Using the example from step 3, you might call this step "Afghanistan". Output fields from this step may be named "Afghanistan.Latitude" or "Afghanistan.Date".
- Click **Use type name** to use the field type as the name for this step in the output. Entities will use entity types and relationships will use relationship labels. Continuing with the same example, output fields with this selection may be named "Person.Latitude" or "Person.Date". If you select this option and enter a name in the **Name** field, that name will also be added as a prefix for all output fields in addition to the field type. Continuing with the same example, output fields with this selection may be named "Afghanistan.Person.Latitude" or "Afghanistan.Person.Date".

8. Specify the steps you want the query to take by selecting the appropriate option in the **Add Operations** drop-down. You can complete this step for the Flow, Conditions, or Output tab. Note that your options vary by whether the root element is an entity or a relationship.

- If you choose **Entity to Entity** (valid for All entities and Specify starting entities), you can then refine your search to return data based on relationship labels between two entities (Connected), before entities (Predecessors), or after entities (Successors). For example, if you are querying a model of family members, and you choose a Relationship label of "Father," a Connected query will return all entities that have a Father label between them (in other words, fathers, sons, and daughters). A Predecessors query will return all entities who are a source entity of a Father relationship connected to another entity (in other words, fathers). A Successors query will return all entities who are the target entity of a Father relationship connected to another entity (in other words, sons and daughters).

As in the root step of your query, you can also select Entity types for this step of the query. You can choose to query selected types or all types. Click Select None to deselect any selected types.

- If you choose **Entity to Relationship** (valid for All entities and Specify starting entities), your options are very similar to those for Entity to Entity. You can refine your search to return data based on relationship labels that attach two entities (Connected), occur before entities (Predecessors), or occur after entities (Successors). You can also add conditions to and define output for the query.
- If you choose **Relationship to Entity** (valid for All relationships), you can refine your search to return data based on conditions you set. You can return data when a condition is in place for an entity that is connected to another entity (Connected), an entity that is a source to a relationship (Predecessors), and for when an entity is a target of a relationship (Successors).

As in the root step of your query, you can also select Entity types for this step of the query. You can choose to query selected types or all types. Click Select None to deselect any selected types.

Regardless of the type of operation you add, you can create Conditions for that operation. You can also define how you want the output from this step to appear. You will notice that steps subsequent to the root step are given a path for output. The path and the step name define the hierarchy of your output data. If you checked the List box in the root step, this path will default to being part of the path in the step before it; however, you can remove the name of the root step. For example, if you named the root step "Locations" and clicked the List box, the first step would by default show "/Locations" in the **Path** field. (Alternatively, you could remove "Locations" and leave just the slash to have this step originate at the root.) If you called the first step "CountryName", the second step would by default show "/Locations/CountryName" in the **Path** field and the Locations field output would contain a list of CountryName results. Click the **Include in results** box if you want the results from this step to be included in the output. Click **Dedup** if you want the query to remove duplicate results from the output.

9. Click **OK**.

Modifying Queries

Once a query is created, you can modify it from the Model Queries tab on the Model Management dialog box. Simply select the query and click **Modify**. Then edit your query accordingly.

You can also modify queries on the Query Pane, on the Query tab in the Open Model dialog box, in Read from Hub, and in Query Hub. When you modify queries from these locations, you must either save the changes and overwrite the existing query (using the same name) or save the changes and give the modified query a new name. Otherwise, the changes will be temporarily saved as a custom query and will be accessible only from Open Model and not from Model Management. The original query will not be modified, and the new, custom query will be available only as long as the Relationship Analysis Client is open or until you use a different query for that model. If you apply a different query to the model or close the Relationship Analysis Client, the changes to the query will be lost.

Copying Queries

Once a query is created, you can copy it from the Model Queries tab on the Model Management dialog box. Select the query and click **Copy**. The Modify Query dialog box will open, containing all the settings from the query you copied. You can then change the query in any way, adding, editing, or removing steps and conditions.

Removing Queries

In addition to modifying a query on the Model Queries tab, you can also remove a query by selecting it and clicking **Remove**.

Copying Models

Follow these steps to copy a model:

1. With Relationship Analysis Client open, click the **Manage** button.
2. Select a model from the list in the **Model Management** dialog box by clicking it.
3. Click the **Copy** button.
4. Enter the name of the copied model.

5. Check **Copy monitors** if you want all monitors for the original model to be included in the copied model.
6. Check **Copy queries** if you want all saved queries for the original model to be included in the copied model.
7. Click **OK**.

Clearing Models

Follow these steps to remove the contents of a model.

Note: Metadata will be retained even when you remove the contents.

1. With Relationship Analysis Client open, click the **Manage** button.
2. Select a model from the list in the **Model Management** dialog box by clicking it.
3. Click the **Clear** button.
4. Click **Yes** to remove the contents of the model.

Removing Models

Follow these steps to access this information:

1. With Relationship Analysis Client open, click the **Manage** button.
2. Select a model from the list in the **Model Management** dialog box by clicking it.
3. Click the **Remove** button.
4. Click **Yes** to remove the model.

Jobs Tab

The Jobs tab provides details for algorithms you run on the Models tab. It provides the following information:

- Job ID
- Name of the model against which the algorithm was run
- Name of the algorithm that was run
- Status of the job
- Start time and end time
- Any messages that may have been returned when the job was run

Backups Tab

The Backups tab shows which models have been backed up and enables you to restore or remove those backups from the server.

Viewing Models

The Relationship Analysis Client offers several methods to view model data:

- **Opening a Model**
- **Canvas Options**
- **Using the Selection Tool**
- **Using the Query Tool**
- **Using Filters**

Opening a Model

The Open Model dialog box of the Relationship Analysis Client tool provides an overview of data for each model even before that model is open. It also enables you to write a query that returns data from the model you select.

Follow these steps to access this information:

1. With Relationship Analysis Client open, click the **Open** button.
2. Select a model from the list by clicking it.
3. Remain on the **Models** tab or click one of the other two tabs.

Models Tab

The Models tab shows the number of entities and relationships for each model. If you click anywhere in a particular model's row (for example, in the Name, Entities, or Relationships columns), that model will open. If you click the column header, the model data will sort in ascending order for that column. If you click the column header twice, the model data will sort in descending order for that column.

Query Tab

The Query tab allows you to create a query that returns data from the model you select. You can use the query builder provided, or you can create a manual query.

You can also select from or modify existing queries that are listed in the Query drop-down. When you modify existing queries, you must either save the changes and overwrite the existing query (using the same name) or save the changes and give the modified query a new name. Otherwise, the changes will be temporarily saved as a custom query and will be accessible only from Open Model and not from Model Management. The original query will not be modified, and the new, custom query will be available only as long as the application is open or until you use a different query for that model. If you apply a different query to the model or close the application, changes to the query will be lost.

Follow these instructions to use the query builder:

1. Choose whether you want to **Select elements** included in the query results, **Add elements** included in the query results, or **Show result** by highlighting the results on the canvas.

2. Check the **Include results from partial traversals** box to include the results from each step in the query. Leave the box unchecked to include only the results that meet the requirements of the last step. For example, let's say that you are looking at a model that depicts world-wide terrorist activity for the events leading up to September 11, 2001, and you want to return data for any meetings that both Osama bin Laden and Mohamed Atta attended. Your query might include the following steps:

- An initial step that includes an exact search type for entities with an `_stp_id` property that has a literal value of "Person:Osama bin Laden"
- An Entity to Relationship step of connected with a relationship label of "Meeting"
- A Relationship to Entity step of connected with a condition of an `_stp_id` property that contains "Atta"

If you leave the **Include results from partial traversals** box unchecked, the query would return a single meeting between Osama bin Laden and Mohamed Atta. If you check the box, the query would return all meetings between Osama bin Laden and anyone else in the model. The additional records would be returned because the second step looks for meetings attended by Osama bin Laden and checking the box returns results from each step in the query.

3. Leave the **Include results from circular traversals** box checked to include elements that occur more than once in each traversal. Uncheck the box to include those elements just once in each traversal. For example, let's say you are using the same model mentioned in step 1, and you initially want to return data for any meetings that Mohamed Atta attended but once you have those results, you want to see all attendees of a particular meeting. Your query might include the following steps:

- An initial step that includes an exact search type for entities with an `_stp_id` property that has a literal value of Person:Mohamed Atta, which will return his entity
- An Entity to Relationship step of connected with a relationship label of "Attended", which will return all events that he attended, including a meeting in Kandahar
- A Relationship to Entity step of connected with a condition of an Event property that contains "Kandahar", which will return just the Kandahar meeting
- An Entity to Relationship step of connected with a relationship label of "Attended", which will return relationships that connect to three other entities who attended the meeting in Kandahar and may or may not return the (already traversed) relationship that connects to Mohamed Atta, depending on whether you use this option.

If you leave the **Include results from circular traversals** box checked, Mohamed Atta's relationship will be returned in addition to those for the other three attendees. If you uncheck the box, Mohamed Atta's relationship will not be returned because that relationship (Person:Mohamed Atta->Attended->Meeting:Kandahar) was already traversed in the first step of the query.

4. Leave the **Limit results to** box checked and enter a number to specify the total maximum number of entities and relationships to return from the query. The default is 5000. The number entered here applies to unique elements, so if the same element appears in multiple results, they will count as one result. To avoid this scenario, use the dedup function discussed at the end of this topic; it will remove duplicate results in the output. If your root step returns a list and

you are querying a large model, we strongly suggest entering a limit in this field to prevent the server from becoming unresponsive.

Note: Limits can be set here or as Query Result Limits set in Relationship Analysis Client [General Settings](#); if the limits are not the same, the lower limit will prevail.

5. Complete the Selection tab.

- Click **All entities**, **All relationships**, or **Specify starting entities** to identify what you want to query against. The Specify starting entities selection allows you to determine at what point in the model you want to begin your search. For instance, if you are looking at a model that depicts world-wide terrorist activity during specific years, you might have country names for entities. Rather than query against the entire model, you might want to look at activity just in Afghanistan. In this case you could select "All" as the **Search type** and "Country" as the **Property name**, leave **Literal** selected, and enter "Afghanistan" as the **Property value**.

Note: The value in the **Property value** field is case sensitive if that field was indexed with the Exact type selected (versus Case insensitive). For more information on selecting types when indexing fields, see [The Entities Tab](#) on page 117.

You could also click **Field** and select "Location" as the Property value, for example, rather than entering a specific value. If you select Field, an **Input Data** grid containing the name of the field you just selected will appear under the query name along with a cell where you can enter the default value. If you reuse this query elsewhere, you can use the default value you provided in this step, or you can override the default at that time.

- If you clicked **All entities** or **Specify starting entities**, select the **Entity types** for your query. You can choose to query selected types or all types. Click **Select None** to deselect any selected types. In addition to returning a more focused set of results, selecting entity types will affect other factors such as which properties and fields are available in the first step of the query, which directions, entity types, and relationship labels are available in subsequent steps of the query, and so on.
- If you clicked **Specify starting entities**, select the **Search type**:

Exact

Searches the index for data that matches exactly what you enter on the Selection tab, including casing. As with property values, the value here is case sensitive if that field was indexed with the Exact type selected; if case sensitivity was used and you search for "texas" while your data includes entries of "Texas", they will not be returned.

Starts with	Searches the index for data that contains text beginning with what you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "tech" or "tec" would be considered a match for a property value containing "Technical", "Technology", "Technologies", or "Technician".
Ends with	Searches the index for data that contains text ending with what you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "Emirates" or "tes" would be considered a match for a property value containing "United Arab Emirates".
Contains	Searches the index for data that contains the text that you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "Pitney" and "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc."
Any	Searches the index for data that contains any of the text that you enter on the Selection tab. For example, a literal property value of "Austin Tex" would be considered a match for a property value containing "Texarkana" or "Stephen F. Austin University".
All	Searches the index for data that contains all of the text that you enter on the Selection tab. For example, a literal property value of "Allstate claim 2013" would be considered a match for a property value containing "filed claim with Allstate June 2013", as would literal property values of "all state" or "all 13".
Between	Searches the index for data that falls within a range that you specify on the Selection tab. When you use this search type, you must select a Property name that contains date, time, date/time, or numeric data. All numeric data types are supported with the exception of BigDecimal. For example, the following specifications would return all entities with StartDate values occurring in the year 2000: <ul style="list-style-type: none">• A Property name of "StartDate" that is a Date type• A literal Start value of "1/1/2000"• A literal End value of "12/31/2000"

Fuzzy

Searches the index for the text you enter on the Selection tab but allows for some differentiation (missing letters, extra letters, or substitutions of letters). The amount of differentiation that is acceptable to still be considered a match depends on what you enter in the **Metric** field. This figure must be greater than zero and less than one; in other words, it must range from ".1" to ".9". For example, if you search for "Barton" and enter ".9" as the metric, the search will return records with "Carton" (replaces B with C), "Bartons" (adds s), and "Baton" (removes r), because all of these words are one character different from the search word "Barton".

Wildcard

Searches the index for the text you enter on the Selection tab but allows for a single wildcard character or a wildcard character sequence. Supported wildcard characters include the question mark (?), which matches any single character, and the asterisk (*), which matches any character sequence (including blanks). For example, if you search cities in Texas for "Aus*", the search will return records with "Austin", "Austonio" and "Austwell". If you conduct a similar search for "Aust??", only "Austin" will be returned because each question mark represents a single character and the other two city names have more characters in their name.

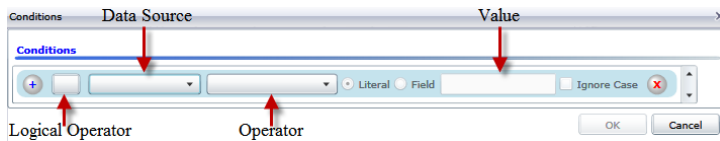
Note: A query that includes an asterisk wildcard as the first character in the search string may result in a lengthy response time.

- If you clicked **Specify starting entities**, select the **Property name** from the drop-down list. This list contains all properties associated with the entities and relationships that make up the model.

Note: You can only query properties that have been indexed; non-indexed properties will not appear in the Property name drop-down.

- If you clicked **Specify starting entities**, and selected a search type other than "Between", select the **Property value**. You can click **Literal** and enter a text string to be used in the search. Alternatively, you can click **Field** and select the field whose data should be searched; if you choose this option you will also need to enter a value in the Input Data grid.
- If you clicked **Specify starting entities**, and selected the "Between" search type, select the **Start value** and **End value** to enter the range. You can click **Literal** and enter a value to be used in the search. Alternatively, you can click **Field** and select the field whose data should be searched; if you choose this option you will also need to enter a value in the Input Data grid.

6. Complete the **Conditions** tab if you want to place additional constraints on the query. The Conditions tab has four entry fields:



- If you are creating the first condition, the **Logical operator** field will remain empty. If you are creating a subsequent condition, specify whether this condition should be used in conjunction with previous conditions ("And") or if it should be used instead of previous conditions ("Or").
- Select the property on which the condition will be based in **Data source** field.
- Select an operator for the condition that is appropriate for the data type in the **Operator** field:

Equals	Searches model elements for properties with values that match exactly what you enter in the Value field. This can be a numeric value or a text value.
Not Equals	Searches model elements for properties with values that have any value other than what you enter in the Value field. This can be a numeric value or a text value.
Exists	Searches the model elements for the existence of the property that you select in the Data Source field.
Does not Exist	Searches the model for elements that do not contain properties that you select in the Data Source field
Is Blank	Searches model elements for properties that contain no data. If a property value is blank, that element will be returned. This can be a numeric value or a text value.
Is Not Blank	Searches model elements for properties that contain any data. If a property value is not blank, that element will be returned. This can be a numeric value or a text value.
Greater Than	Searches model elements for properties whose values are greater than the value you specify. This can be a numeric, date, date/time, or time value.
Greater Than or Equals	Searches model elements for properties with numeric values that are greater than or equal to the value you specify. This can be a numeric, date, date/time, or time value.
Less Than	Searches model elements for properties with numeric values that are less than the value you specify. This can be a numeric, date, date/time, or time value.

Less Than or Equals	Searches model elements for properties with numeric values that are less than or equal to the value you specify. This can be a numeric, date, date/time, or time value.
Contains	Searches model elements for properties with values that contain what you enter in the Data Source field. The search does not need to be a complete word. For example, a literal property value of "Pitney" or "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc." This can be a numeric value or a text value.
Does not Contain	Searches model elements for properties that do not contain what you enter in the Data Source field. The search does not need to be a complete word. For example, a literal property value of "Pitney" or "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc.," so "Pitney Bowes Software" would not be returned. This can be a numeric value or a text value.
Starts With	Searches model elements for properties whose values start with the text you enter in the Data Source field. For example, if you enter "Van" for the LastName field it would return results with "Van Buren", "Vandenburg", or "Van Dyck".
Does Not Start With	Searches model elements for properties whose values do not start with the text you enter in the Data Source field. For example, if you enter "Van" for the LastName field it would not return results with "Van Buren", "Vandenburg", or "Van Dyck" but would return results with "Eddie Van Halen".
Ends With	Searches model elements for properties whose values end with the text you enter in the Data Source field. For example, if you filter for records that end with "burg" in the City field, it would return results with "Gettysburg", "Fredricksburg", and "Blacksburg".
Does Not End With	Searches model elements for properties whose values do not end with the text you enter in the Data Source field. For example, if you filter for records that end with "burg" in the City field it would not return results with "Gettysburg", "Fredricksburg", and "Blacksburg" but would return results with "Burgess".
Match Regular Expression	Searches the model elements for properties having a regular expression match for what you enter in the Data Source field. Regular expression matches identify strings of text of interest, such as particular characters, words, or patterns of characters. The value field should contain a valid regular expression pattern.
Is Roughly Similar To	Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing

letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with search metric of .5.

Is Similar To Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with search metric of .6.

Is Very Similar To Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with Search Metric of .7.

d. In the drop-down box following the list of operators:

- Select **Literal** and enter a text string the fourth box (called the **Value** field) to be used in the query.
- Select **Field** and then select the field whose data should be searched in the **Value** field.
- Select a previous step (such as "**Root**" or "**Step1**") and then a property in the **Value** field to compare property values for the current step against values returned in a previous step. (Note that if you named the output on the Output tab of previous steps, those names will appear in the drop-down rather than "Root" or "Step1".) In this case, the properties shown in the Value field are based on properties for the previous step. For example, if you knew the name of one person (Mohamed Atta) who attended a particular event (a meeting in Kandahar) but wanted to know the names of the other attendees, you could create the following query that includes a property value comparison:

- A root step that looks for an entity type Person with an `_stp_id` of that contains "Mohamed"
- An Entity to Relationship step with a relationship label of "Attended"
- A Relationship to Entity step with a condition that includes an Event that contains "Kandahar"
- An Entity to Relationship step with a relationship label of "Attended" plus a condition that this step's `_stp_id` does not contain the same `_stp_id` value that was found in the root step.

This query will find that Mohamed Atta attended an event in Kandahar and that it was also attended by three other people whose `_stp_id` value is not "Mohamed".

- e. Click **Ignore Case** if the query results can be either upper or lower cased.
- f. Repeat steps a through e to add additional conditions.
- g. Click **OK**.

If, for example, you want to target terrorist activity in Afghanistan between 2001 and 2010, you would create two conditions. First, you would select "date" for the **Property name**, then "Greater Than or Equals", leave Literal selected, and then enter "2001". You would follow this with a second condition set to "And" that also uses "date," then "Less Than or Equals", then "2010". Alternatively, you could click **Field** and select "Date" rather than entering a specific value. Add, delete, or change the order of conditions by using the icons on either side of the conditions. Click **Ignore Case** if the query results can be either upper or lower cased.

7. Complete the **Output** tab to define how you want your output to appear.

- Click the **Include in results** box if you want the results from this step to be included in the output.

Note: This box must be checked for the last step in any series; therefore, if there is only one step you cannot uncheck this box.

- Click **Specify name** and enter text in the **Name** field to provide a name for this step in the output. Click **List** to use this entry as the name and type of the field in hierarchical output; leave it unchecked to have this entry added as a prefix for all output fields. Using the example from step 3, you might call this step "Afghanistan". Output fields from this step may be named "Afghanistan.Latitude" or "Afghanistan.Date".
- Click **Use type name** to use the field type as the name for this step in the output. Entities will use entity types and relationships will use relationship labels. Continuing with the same example, output fields with this selection may be named "Person.Latitude" or "Person.Date". If you select this option and enter a name in the **Name** field, that name will also be added as a prefix for all output fields in addition to the field type. Continuing with the same example, output fields with this selection may be named "Afghanistan.Person.Latitude" or "Afghanistan.Person.Date".

8. Specify the steps you want the query to take by selecting the appropriate option in the **Add Operations** drop-down. You can complete this step for the Flow, Conditions, or Output tab. Note that your options vary by whether the root element is an entity or a relationship.

- If you choose **Entity to Entity** (valid for All entities and Specify starting entities), you can then refine your search to return data based on relationship labels between two entities (Connected), before entities (Predecessors), or after entities (Successors). For example, if you are querying a model of family members, and you choose a Relationship label of "Father," a Connected query will return all entities that have a Father label between them (in other words, fathers, sons, and daughters). A Predecessors query will return all entities who are a source entity of a Father relationship connected to another entity (in other words, fathers). A Successors query will return all entities who are the target entity of a Father relationship connected to another entity (in other words, sons and daughters).

As in the root step of your query, you can also select Entity types for this step of the query. You can choose to query selected types or all types. Click Select None to deselect any selected types.

- If you choose **Entity to Relationship** (valid for All entities and Specify starting entities), your options are very similar to those for Entity to Entity. You can refine your search to return data based on relationship labels that attach two entities (Connected), occur before entities (Predecessors), or occur after entities (Successors). You can also add conditions to and define output for the query.
- If you choose **Relationship to Entity** (valid for All relationships), you can refine your search to return data based on conditions you set. You can return data when a condition is in place for an entity that is connected to another entity (Connected), an entity that is a source to a relationship (Predecessors), and for when an entity is a target of a relationship (Successors).

As in the root step of your query, you can also select Entity types for this step of the query. You can choose to query selected types or all types. Click Select None to deselect any selected types.

Regardless of the type of operation you add, you can create Conditions for that operation. You can also define how you want the output from this step to appear. You will notice that steps subsequent to the root step are given a path for output. The path and the step name define the hierarchy of your output data. If you checked the List box in the root step, this path will default to being part of the path in the step before it; however, you can remove the name of the root step. For example, if you named the root step "Locations" and clicked the List box, the first step would by default show "/Locations" in the **Path** field. (Alternatively, you could remove "Locations" and leave just the slash to have this step originate at the root.) If you called the first step "CountryName", the second step would by default show "/Locations/CountryName" in the **Path** field and the Locations field output would contain a list of CountryName results. Click the **Include in results** box if you want the results from this step to be included in the output. Click **Dedup** if you want the query to remove duplicate results from the output.

9. Click **OK**.

Refreshing a Model

The Refresh feature of the Relationship Analysis Client will re-open the model that is currently open to its original settings. If you have made any changes to the model, you will be prompted to save the changes before refreshing. This feature is also available when you are working with model metadata or the results of a query.

With a model or metadata open in the Relationship Analysis Client, click the Model tab and then the **Refresh** button.

Canvas Options

The Canvas Options for Relationship Analysis Client include the following:

- **Zoom In**—Narrows the focus to the central point on the current view. Use the Overview pane to move the focus to a specific area on the model.
- **Zoom Out**—Expands the focus to show more of the model. Works as an undo for zooming in, if applicable.
- **Toggle Auto Zoom**—Enables you to turn automatic zooming off and on. Auto zoom will zoom in on items as they are selected, fitting all selected items into the viewport.
- **Fit to Bounds**—Fits the entire model within the bounds of the model view.
- **Selection Fading**—Causes items that are not selected to fade into the background. When nothing is selected, all items appear in the foreground.

Using the Query Tool

The Query tool allows you to create a query that returns data for the open model. You can use the query builder provided, or you can write a custom query.

Before you create your query, you must decide how you want results to be returned. Your options vary, depending on whether you have loaded a full model or a partial model into the Relationship Analysis Client. A full model is one that has all entities and relationships present on the canvas. A partial model is one that has some but not all entities and relationships present on the canvas because you used a query, selection, or filter function to bring in just some of the elements of your model.

- **Select elements**—This option will cause the elements identified in the query to be selected on the canvas.
- **Add elements**—This option will cause elements identified in the query that were not already loaded to be added to the canvas.

Note: This option has no effect on a fully loaded model because all elements are already present.

- **Show result**—This option will cause the Relationship Analysis Client to display raw data returned from the query in the Result window.

Follow these instructions to use the query builder:

1. Choose whether you want to **Select elements** included in the query results, **Add elements** included in the query results, or **Show result** by highlighting the results on the canvas.
2. Check the **Include results from partial traversals** box to include the results from each step in the query. Leave the box unchecked to include only the results that meet the requirements of the last step. For example, let's say that you are looking at a model that depicts world-wide terrorist activity for the events leading up to September 11, 2001, and you want to return data for any meetings that both Osama bin Laden and Mohamed Atta attended. Your query might include the following steps:
 - An initial step that includes an exact search type for entities with an `_stp_id` property that has a literal value of "Person:Osama bin Laden"
 - An Entity to Relationship step of connected with a relationship label of "Meeting"
 - A Relationship to Entity step of connected with a condition of an `_stp_id` property that contains "Atta"

If you leave the **Include results from partial traversals** box unchecked, the query would return a single meeting between Osama bin Laden and Mohamed Atta. If you check the box, the query would return all meetings between Osama bin Laden and anyone else in the model. The additional records would be returned because the second step looks for meetings attended by Osama bin Laden and checking the box returns results from each step in the query.

3. Leave the **Include results from circular traversals** box checked to include elements that occur more than once in each traversal. Uncheck the box to include those elements just once in each traversal. For example, let's say you are using the same model mentioned in step 1, and you initially want to return data for any meetings that Mohamed Atta attended but once you have those results, you want to see all attendees of a particular meeting. Your query might include the following steps:

- An initial step that includes an exact search type for entities with an `_stp_id` property that has a literal value of `Person:Mohamed Atta`, which will return his entity
- An Entity to Relationship step of connected with a relationship label of "Attended", which will return all events that he attended, including a meeting in Kandahar
- A Relationship to Entity step of connected with a condition of an Event property that contains "Kandahar", which will return just the Kandahar meeting
- An Entity to Relationship step of connected with a relationship label of "Attended", which will return relationships that connect to three other entities who attended the meeting in Kandahar and may or may not return the (already traversed) relationship that connects to Mohamed Atta, depending on whether you use this option.

If you leave the **Include results from circular traversals** box checked, Mohamed Atta's relationship will be returned in addition to those for the other three attendees. If you uncheck the box, Mohamed Atta's relationship will not be returned because that relationship (Person:Mohamed Atta->Attended->Meeting:Kandahar) was already traversed in the first step of the query.

4. Leave the **Limit results to** box checked and enter a number to specify the total maximum number of entities and relationships to return from the query. The default is 5000. The number entered here applies to unique elements, so if the same element appears in multiple results, they will count as one result. To avoid this scenario, use the dedup function discussed at the end of this topic; it will remove duplicate results in the output. If your root step returns a list and you are querying a large model, we strongly suggest entering a limit in this field to prevent the server from becoming unresponsive.

Note: Limits can be set here or as Query Result Limits set in Relationship Analysis Client [General Settings](#); if the limits are not the same, the lower limit will prevail.

5. Complete the Selection tab.
 - Click **All entities**, **All relationships**, or **Specify starting entities** to identify what you want to query against. The Specify starting entities selection allows you to determine at what point in the model you want to begin your search. For instance, if you are looking at a model that depicts world-wide terrorist activity during specific years, you might have country names for entities. Rather than query against the entire model, you might want to look at activity just in Afghanistan. In this case you could select "All" as the **Search type** and "Country" as the **Property name**, leave **Literal** selected, and enter "Afghanistan" as the **Property value**.

Note: The value in the **Property value** field is case sensitive if that field was indexed with the Exact type selected (versus Case insensitive). For more information on selecting types when indexing fields, see [The Entities Tab](#) on page 117.

You could also click **Field** and select "Location" as the Property value, for example, rather than entering a specific value. If you select Field, an **Input Data** grid containing the name of the field you just selected will appear under the query name along with a cell where you can enter the default value. If you reuse this query elsewhere, you can use the default value you provided in this step, or you can override the default at that time.

- If you clicked **All entities** or **Specify starting entities**, select the **Entity types** for your query. You can choose to query selected types or all types. Click **Select None** to deselect any selected types. In addition to returning a more focused set of results, selecting entity types will affect other factors such as which properties and fields are available in the first step of the query, which directions, entity types, and relationship labels are available in subsequent steps of the query, and so on.
- If you clicked **Specify starting entities**, select the **Search type**:

Exact	Searches the index for data that matches exactly what you enter on the Selection tab, including casing. As with property values, the value here is case sensitive if that field was indexed with the Exact type selected; if case sensitivity was used and you search for "texas" while your data includes entries of "Texas", they will not be returned.
Starts with	Searches the index for data that contains text beginning with what you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "tech" or "tec" would be considered a match for a property value containing "Technical", "Technology", "Technologies", or "Technician".
Ends with	Searches the index for data that contains text ending with what you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "Emirates" or "tes" would be considered a match for a property value containing "United Arab Emirates".
Contains	Searches the index for data that contains the text that you enter on the Selection tab. The search does not need to be a complete word. For example, a literal property value of "Pitney" and "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc."
Any	Searches the index for data that contains any of the text that you enter on the Selection tab. For example, a literal property value of "Austin Tex" would be considered a match for a property value containing "Texarkana" or "Stephen F. Austin University".

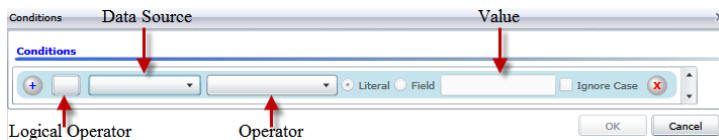
All	Searches the index for data that contains all of the text that you enter on the Selection tab. For example, a literal property value of "Allstate claim 2013" would be considered a match for a property value containing "filed claim with Allstate June 2013", as would literal property values of "all state" or "all 13".
Between	<p>Searches the index for data that falls within a range that you specify on the Selection tab. When you use this search type, you must select a Property name that contains date, time, date/time, or numeric data. All numeric data types are supported with the exception of BigDecimal. For example, the following specifications would return all entities with StartDate values occurring in the year 2000:</p> <ul style="list-style-type: none">• A Property name of "StartDate" that is a Date type• A literal Start value of "1/1/2000"• A literal End value of "12/31/2000"
Fuzzy	Searches the index for the text you enter on the Selection tab but allows for some differentiation (missing letters, extra letters, or substitutions of letters). The amount of differentiation that is acceptable to still be considered a match depends on what you enter in the Metric field. This figure must be greater than zero and less than one; in other words, it must range from ".1" to ".9". For example, if you search for "Barton" and enter ".9" as the metric, the search will return records with "Carton" (replaces B with C), "Bartons" (adds s), and "Baton" (removes r), because all of these words are one character different from the search word "Barton".
Wildcard	<p>Searches the index for the text you enter on the Selection tab but allows for a single wildcard character or a wildcard character sequence. Supported wildcard characters include the question mark (?), which matches any single character, and the asterisk (*), which matches any character sequence (including blanks). For example, if you search cities in Texas for "Aus*", the search will return records with "Austin", "Austonio" and "Austwell". If you conduct a similar search for "Aust??", only "Austin" will be returned because each question mark represents a single character and the other two city names have more characters in their name.</p> <p>Note: A query that includes an asterisk wildcard as the first character in the search string may result in a lengthy response time.</p>

- If you clicked **Specify starting entities**, select the **Property name** from the drop-down list. This list contains all properties associated with the entities and relationships that make up the model.

Note: You can only query properties that have been indexed; non-indexed properties will not appear in the Property name drop-down.

- If you clicked **Specify starting entities**, and selected a search type other than "Between", select the **Property value**. You can click **Literal** and enter a text string to be used in the search. Alternatively, you can click **Field** and select the field whose data should be searched; if you choose this option you will also need to enter a value in the Input Data grid.
- If you clicked **Specify starting entities**, and selected the "Between" search type, select the **Start value** and **End value** to enter the range. You can click **Literal** and enter a value to be used in the search. Alternatively, you can click **Field** and select the field whose data should be searched; if you choose this option you will also need to enter a value in the Input Data grid.

6. Complete the **Conditions** tab if you want to place additional constraints on the query. The Conditions tab has four entry fields:



- If you are creating the first condition, the **Logical operator** field will remain empty. If you are creating a subsequent condition, specify whether this condition should be used in conjunction with previous conditions ("And") or if it should be used instead of previous conditions ("Or").
- Select the property on which the condition will be based in **Data source** field.
- Select an operator for the condition that is appropriate for the data type in the **Operator** field:

Equals	Searches model elements for properties with values that match exactly what you enter in the Value field. This can be a numeric value or a text value.
Not Equals	Searches model elements for properties with values that have any value other than what you enter in the Value field. This can be a numeric value or a text value.
Exists	Searches the model elements for the existence of the property that you select in the Data Source field.
Does not Exist	Searches the model for elements that do not contain properties that you select in the Data Source field
Is Blank	Searches model elements for properties that contain no data. If a property value is blank, that element will be returned. This can be a numeric value or a text value.

Is Not Blank	Searches model elements for properties that contain any data. If a property value is not blank, that element will be returned. This can be a numeric value or a text value.
Greater Than	Searches model elements for properties whose values are greater than the value you specify. This can be a numeric, date, date/time, or time value.
Greater Than or Equals	Searches model elements for properties with numeric values that are greater than or equal to the value you specify. This can be a numeric, date, date/time, or time value.
Less Than	Searches model elements for properties with numeric values that are less than the value you specify. This can be a numeric, date, date/time, or time value.
Less Than or Equals	Searches model elements for properties with numeric values that are less than or equal to the value you specify. This can be a numeric, date, date/time, or time value.
Contains	Searches model elements for properties with values that contain what you enter in the Data Source field. The search does not need to be a complete word. For example, a literal property value of "Pitney" or "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc." This can be a numeric value or a text value.
Does not Contain	Searches model elements for properties that do not contain what you enter in the Data Source field. The search does not need to be a complete word. For example, a literal property value of "Pitney" or "Pitney Bowes" would be considered a match for a property value containing "Pitney Bowes Software Inc.," so "Pitney Bowes Software" would not be returned. This can be a numeric value or a text value.
Starts With	Searches model elements for properties whose values start with the text you enter in the Data Source field. For example, if you enter "Van" for the LastName field it would return results with "Van Buren", "Vandenburg", or "Van Dyck".
Does Not Start With	Searches model elements for properties whose values do not start with the text you enter in the Data Source field. For example, if you enter "Van" for the LastName field it would not return results with "Van Buren", "Vandenburg", or "Van Dyck" but would return results with "Eddie Van Halen".
Ends With	Searches model elements for properties whose values end with the text you enter in the Data Source field. For example, if you filter for records that end with "burg" in the City field, it would return results with "Gettysburg", "Fredricksburg", and "Blacksburg".

Does Not End With	Searches model elements for properties whose values do not end with the text you enter in the Data Source field. For example, if you filter for records that end with "burg" in the City field it would not return results with "Gettysburg", "Fredricksburg", and "Blacksburg" but would return results with "Burgess".
Match Regular Expression	Searches the model elements for properties having a regular expression match for what you enter in the Data Source field. Regular expression matches identify strings of text of interest, such as particular characters, words, or patterns of characters. The value field should contain a valid regular expression pattern.
Is Roughly Similar To	Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with search metric of .5.
Is Similar To	Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with search metric of .6.
Is Very Similar To	Searches model elements for properties with values close to what you enter in the Data Source field but allows for some differentiation (missing letters, extra letters, or substitutions of letters). This operator is equivalent to the Fuzzy search type with Search Metric of .7.

d. In the drop-down box following the list of operators:

- Select **Literal** and enter a text string the fourth box (called the **Value** field) to be used in the query.
- Select **Field** and then select the field whose data should be searched in the **Value** field.
- Select a previous step (such as "**Root**" or "**Step1**") and then a property in the **Value** field to compare property values for the current step against values returned in a previous step. (Note that if you named the output on the Output tab of previous steps, those names will appear in the drop-down rather than "Root" or "Step1".) In this case, the properties shown in the Value field are based on properties for the previous step. For example, if you knew the name of one person (Mohamed Atta) who attended a particular event (a meeting in Kandahar) but wanted to know the names of the other attendees, you could create the following query that includes a property value comparison:
 - A root step that looks for an entity type Person with an `_stp_id` of that contains "Mohamed"
 - An Entity to Relationship step with a relationship label of "Attended"
 - A Relationship to Entity step with a condition that includes an Event that contains "Kandahar"
 - An Entity to Relationship step with a relationship label of "Attended" plus a condition that this step's `_stp_id` does not contain the same `_stp_id` value that was found in the root step.

This query will find that Mohamed Atta attended an event in Kandahar and that it was also attended by three other people whose `_stp_id` value is not "Mohamed".

- e. Click **Ignore Case** if the query results can be either upper or lower cased.
- f. Repeat steps a through e to add additional conditions.
- g. Click **OK**.

If, for example, you want to target terrorist activity in Afghanistan between 2001 and 2010, you would create two conditions. First, you would select "date" for the **Property name**, then "Greater Than or Equals", leave Literal selected, and then enter "2001". You would follow this with a second condition set to "And" that also uses "date," then "Less Than or Equals", then "2010". Alternatively, you could click **Field** and select "Date" rather than entering a specific value. Add, delete, or change the order of conditions by using the icons on either side of the conditions. Click **Ignore Case** if the query results can be either upper or lower cased.

7. Complete the **Output** tab to define how you want your output to appear.

- Click the **Include in results** box if you want the results from this step to be included in the output.

Note: This box must be checked for the last step in any series; therefore, if there is only one step you cannot uncheck this box.

- Click **Specify name** and enter text in the **Name** field to provide a name for this step in the output. Click **List** to use this entry as the name and type of the field in hierarchical output; leave it unchecked to have this entry added as a prefix for all output fields. Using the example from step 3, you might call this step "Afghanistan". Output fields from this step may be named "Afghanistan.Latitude" or "Afghanistan.Date".
- Click **Use type name** to use the field type as the name for this step in the output. Entities will use entity types and relationships will use relationship labels. Continuing with the same example, output fields with this selection may be named "Person.Latitude" or "Person.Date". If you select this option and enter a name in the **Name** field, that name will also be added as a prefix for all output fields in addition to the field type. Continuing with the same example, output fields with this selection may be named "Afghanistan.Person.Latitude" or "Afghanistan.Person.Date".

8. Specify the steps you want the query to take by selecting the appropriate option in the **Add Operations** drop-down. You can complete this step for the Flow, Conditions, or Output tab. Note that your options vary by whether the root element is an entity or a relationship.

- If you choose **Entity to Entity** (valid for All entities and Specify starting entities), you can then refine your search to return data based on relationship labels between two entities (Connected), before entities (Predecessors), or after entities (Successors). For example, if you are querying a model of family members, and you choose a Relationship label of "Father," a Connected query will return all entities that have a Father label between them (in other words, fathers, sons, and daughters). A Predecessors query will return all entities who are a source entity of a Father relationship connected to another entity (in other words, fathers). A Successors query will return all entities who are the target entity of a Father relationship connected to another entity (in other words, sons and daughters).

As in the root step of your query, you can also select Entity types for this step of the query. You can choose to query selected types or all types. Click Select None to deselect any selected types.

- If you choose **Entity to Relationship** (valid for All entities and Specify starting entities), your options are very similar to those for Entity to Entity. You can refine your search to return data based on relationship labels that attach two entities (Connected), occur before entities (Predecessors), or occur after entities (Successors). You can also add conditions to and define output for the query.
- If you choose **Relationship to Entity** (valid for All relationships), you can refine your search to return data based on conditions you set. You can return data when a condition is in place for an entity that is connected to another entity (Connected), an entity that is a source to a relationship (Predecessors), and for when an entity is a target of a relationship (Successors). As in the root step of your query, you can also select Entity types for this step of the query. You can choose to query selected types or all types. Click Select None to deselect any selected types.

Regardless of the type of operation you add, you can create Conditions for that operation. You can also define how you want the output from this step to appear. You will notice that steps subsequent to the root step are given a path for output. The path and the step name define the hierarchy of your output data. If you checked the List box in the root step, this path will default to being part of the path in the step before it; however, you can remove the name of the root step. For example, if you named the root step "Locations" and clicked the List box, the first step would by default show "/Locations" in the **Path** field. (Alternatively, you could remove "Locations" and leave just the slash to have this step originate at the root.) If you called the first step "CountryName", the second step would by default show "/Locations/CountryName" in the **Path** field and the Locations field output would contain a list of CountryName results. Click the **Include in results** box if you want the results from this step to be included in the output. Click **Dedup** if you want the query to remove duplicate results from the output.

9. Click **OK**.

Using the Selection Tools

The Selection Tools enable you to select elements in a model based on certain criteria. Using the Selection ribbon, you can select types of elements (entities or relationships); you can also select entities based on their connection to the current (or last selected) element (all connected, predecessors, or successors). The Selection pane provides a search function for locating elements within a model.

Using the Selection Ribbon

The Selection Ribbon enables you to quickly select elements or groups of elements in a model.

- **Select All**—Selects all elements on the canvas.
- **Select All Entities**—Selects all entities on the canvas.
- **Select All Relationships**—Selects all relationships on the canvas.

- **Keep Selection**—Keeps the highlighted element selected when performing other actions. For example, if you have an entity selected and click Keep Selection followed by Select Connected, that one entity plus all elements selected to that entity would be selected.
- **Current Item**—Causes actions to be taken only for the selected element. For example, if you have an entity selected and click Current Item followed by Select Predecessors, all elements coming into that item will be selected. If you then click Select Successors, only successors to the initial entity will be returned. Successors to the entities brought in when you clicked Select Predecessors will not be returned.
- **Select Connected**—Selects all elements connected to the selected element.
- **Select Predecessors**—Selects all elements coming into the selected element.
- **Select Successors**—Selects all elements going out of the selected element.

You can also access these options by right-clicking in the Primary Pane.


Using the Selection Pane

The Selection Pane enables you to designate qualifying properties in your data and select model elements that contain those properties. For example, if you are looking at data involving the United States of America, you might want to see which states have a population greater than 10,000,000.

Use the Search tab to select elements in your model based on a simple Where statement. Use the Advanced tab to select elements in your model that are connected in some manner to currently selected elements.

Note: This pane is on by default but can be toggled off and on using the Selection button on the Tools ribbon of the View tab.

The Search Tab

1. Select either **Entities** or **Relationships** as the model element whose data you're searching.
2. Complete the **Where** step by selecting the property whose data you're searching, the operator you want to use in your selection, and the value that determines whether an item is returned. Using the population example, your Where step could be State Population, Greater Than, and 10,000,000. This would return data and select entities for states whose population is greater than 10,000,000.
3. Click the Run button . The items whose data met the requirements of the selection task are shown in the **Selected items** list box; by default, the last item in the list will be selected. (Likewise, all data is shown for each selected entity in the Model Components pane.)
4. Click any of the items in the **Selected items** list box to see its properties, relationships, and history.
5. Click the **Relationships** tab to see a list of items that are linked to the item you clicked in step 4. Again, using the population example, this selection would return Texas, among other states. If you click Texas in the Selected items list box and then clicked the Relationships tab, you might see that Texas has relationships as a border state with Arkansas, Louisiana, New Mexico, and Oklahoma. In this view, you can sort by relationship or by entity.
6. Click the **History** tab to view changes that have been made to the entity or relationship you selected in step 4. Alternatively, you can access history data by clicking an entity or relationship

on the canvas. If you selected an entity, it will provide detailed information for changes to that entity as well as high-level information for changes to the relationships tied to that entity. If you selected a relationship, it will provide detailed information for changes to that relationship.

Note: This information will be returned only if you have activated the History function.

To do so:

1. Open **Data Hub Settings** on the Resources tab of Management console.
2. Check the **Track history** box to activate history.

The History feature groups data by version number, in reverse chronological order, and provides a date and time stamp for each version. It also provides the user name of the person who made the changes. Additionally, the History tab displays the following details about any given change:

- **Operation**—The action that took place: added, modified, or deleted. Note that if you delete an entity, it will be shown only if it is re-added to the model. However, if you delete relationships or properties, those deletions will appear in the history.
- **Type**—The component that changed: property or relationship
- **Name**—The name of what changed. For properties, this is the ID of the property that changed. For entities whose relationship changed, this is the source and target IDs plus the relationship name. For example, if you were using insurance fraud data and you changed something about a relationship and then viewed the history of one of the entities tied to that relationship, you might see something like this: "Provider:Dr. John Smith=>Treated=>Patient:Jane Doe".
- **Value**—The new value that was entered (if any)
- **Previous Value**—The value prior to the change (if any)

You can perform a subsequent selection task that yields results opposite those returned in your first selection task. For instance, using the population example, a subsequent inverse selection task would result in all states whose population is **less than** 10,000,000 being selected. You can even retain the selections returned in your first selection task. After performing an initial selection task, check the **Keep current selection** box if you want to retain the selections returned in your first selection task. If you do not want to retain the original selections, leave this box blank. Check the **Select inverse** box and click the **Run** button:

You could also use the inverse function combined with the Filter tool to select and remove entities and relationships you do not want, thus allowing you to isolate entities and relationships you do want. Say you are using a model comprised of state data and you want to know which states have more than four electoral votes. You could create a selection that uses entities where Votes in Electoral College equals 4, click the Select inverse box, and click the Apply button. This will cause all states who have more or less than four votes in the electoral college to be selected. From there, you could click the **Filters** tab, click **Entities** in the Selection box, and only entities whose states have 4 electoral college votes would remain.

The Advanced Tab

When you have selected one or more elements of your model you can use these advanced functions to select additional elements. Choose to return elements who are connected to, precede, or succeed currently selected entities. If you would like to select more than just first-degree connections, specify

how many layers of connections to return in the **Depth** field. A depth of 1 would return only the entities that are connected to the selected entity. A depth of 2 would return those entities plus the entities connected to them, and so on.

You can choose to keep the current element selected when performing other actions. For example, if you have an entity selected and click **Keep current selection** followed by **Connected**, that one entity plus all relationships would be selected.

You can also cause actions to be taken only for the currently selected element. For example, if you have an entity selected and click **Current item only** followed by **Predecessors**, all elements coming into that item will be selected. If you then click **Successors**, only successors to the initial entity will be returned. Successors to the entities brought in when you clicked Select Predecessors will not be returned.

Another option on the Advanced tab is to return all items connected to currently selected entities by the **Shortest path**. If you check the **Directed** box, the path will go just in one direction. If you leave it unchecked, the path may go in multiple directions. In the **Weight** drop-down, you can select the property the Selection tool should look at when determining shortest paths.

After making your selections, click the Run button:



The items whose data met the requirements of the selection task are shown in the Selected items list box. (Likewise, all data is shown for each selected entity in the Model Components pane.) Click any of the items in the Selected items list box to see all of that item's data in the Properties tab. Click the Relationships tab to see a list of items that are linked to the item you clicked in step 4. Again, using the population example, this selection would return Texas, among other states. If you click Texas in the Selected items list box and then clicked the Relationships tab, you might see that Texas has relationships as a border state with Arkansas, Louisiana, New Mexico, and Oklahoma. In this view, you can sort by relationship or by entity.

Using the Current Item Pane

The Current Item Pane enables you to view the properties and relationships associated with any given entity or relationship by selecting it either on the canvas or from search results. If the History function is enabled, you can also view the history of a selected entity or relationship by clicking the History tab. The History function is turned off by default; click [here](#) to learn more.

Note: This pane is on by default but can be toggled off and on using the Current Item button on the Tools ribbon of the View tab.

Using the Map Tool

The Map tool shows geographic points on a map, represented by push pins, which correspond to selected entities within a model. Using latitudes and longitudes, the map is updated dynamically as you select entities on the canvas. Likewise, the map automatically zooms in and out, based on the locations of the selected entities, to display all of the selected entities at one time. When you click

a push pin, any connections to other push pins will be drawn on the map. This pane is on by default but can be toggled off and on using the Map button on the Tools ribbon of the View tab.

If your model contains Latitude and Longitude fields, the Map tool will detect them and automatically use that data to identify the locations of the selected entities. If your model does not contain those fields, use the **Options** drop-down to point the Map tool to the fields that contain latitude and longitude data.

The Map tool provides the following features on the Map pane:

- Zoom control—Enables you to set one of the following zoom levels for the map: neighborhood, city, region, state, or world. You can also use the scroll wheel on your mouse to zoom in and out.
- Pan control—Enables you to pan across the map in four directions by clicking one of the arrows. Alternatively, you can move the map itself by holding down the mouse button and dragging in any direction. You can also click anywhere in the map to shift the focus to that area.
- A selection of view options that vary depending on which provider you select. These are the default options:
 - Road View—Displays the map in a traditional view, using vector imagery of roads, buildings, and geography.
 - Aerial View—Displays the map using satellite imagery and highlights roads and major landmarks among the satellite images. Also provides the option of showing labels on the map, including the names of streets and the names of companies that reside in the building that is labeled.
 - Bird's Eye View—Displays the map using imagery captured from low-flying aircraft. While aerial views show a top-down perspective, bird's-eye images are taken at an oblique 45-degree angle, which enables the showing of sides and roofs of buildings. Details such as signs, advertisements and pedestrians are clearly visible.

The Map tool provides the following on the Options pane:

- Latitude: Select the property that contains latitude data.
- Longitude: Select the property that contains longitude data.
- Provider: Select the provider whose map data you want to use.
- Distance unit: Select whether the units should be displayed in miles or kilometers.
- Show map controls: Check this box to show or hide the zoom and pan controls, the view-select command bar, and the latitude/longitude coordinates.

Using Charts

The Charts feature of Relationship Analysis Client enables you to visualize data from a model in graphic format. When you click the **View** tab, then click the **Charts** button on the **Tools** ribbon, you will be prompted to select from four options:

- **Data bar**
- **Treemap**
- **Heat map**
- **Timeline**

After selecting the type of chart you want to create, you will then need to select criteria on which the chart should be based and click the check mark icon. If the check mark icon appears to be faded, you have not fulfilled the minimum criteria required to create the chart. If no results are returned, that means none of the data in the model met the criteria you selected.

To delete a chart, right-click the chart's tab and select **Hide**.

Note: If more than one view is open for a given model, the last selected view is the one used for the chart display. If you make changes to values that are used in a chart, you must close the existing chart and relaunch the Charts feature to see the latest values.

Using Data Bar Charts

Data bar charts present numeric data from a model in bar chart format. Therefore, only properties with numeric formats are eligible for use with data bar charts. Data bar charts display each entry for the selected criteria from highest to lowest value. This represents the order of influence within the model, from most influential to least influential. For instance, if you were looking at a model of suspected insurance fraud data, and you selected entities, with `_stp_indegree` as the property name, you might see a data bar chart that showed a list of providers. The providers who saw the most patients (represented in the model as incoming relationships to the providers) would be at the top of the chart, and the providers who saw the least amount of patients would be at the bottom of the chart.

1. Select either **Entities** or **Relationships**.
2. Select from the **Property Name** field whether you want the data bar chart to include incoming counts, outgoing counts, or both.
3. Click the check mark icon.

You can create multiple data bar charts simply by selecting new criteria and clicking the check mark icon. All data bar charts will appear on the canvas as you create them.

Using Treemaps

Treemaps present data in a hierarchical manner, resulting in several individual squares and rectangles that together form a whole. The size and color of each rectangle reflects its value and influence within the model, and when you hover over any given block the data upon which you created the map will appear. For instance, if you were looking at a model of suspected insurance fraud data, you might select the following criteria:

- Entities
- Group by BusinessName
- Groups include all
- `_stp_degree` as the business names' value

With these settings, you might see a treemap of blocks representing each company against whom workers' compensation claims have been filed. In a model of 16 claims, a company with eight claims would have a block that comprises half of the treemap, a company with 4 claims would comprise a quarter of the map, and companies with 2 claims would comprise an eighth of the map. If you hovered

over the block for the first company, its name and "8" would appear; if you hovered over the block for the second company, its name and "4" would appear, and so on.

1. Select either **Entities** or **Relationships**.
2. Select a property in the **Group by** drop-down. You can also select <none> if you do not want the treemap to be based on a specific property.
3. Specify in the **Groups** drop-down which groups within that property you want to be included in the treemap. If you selected <none> in step 2 this field will not appear.
4. Select from the **Value** field whether you want the treemap to include incoming counts, outgoing counts, or both.
5. Click the check mark icon.

You can create an additional treemap by selecting Charts > Treemap and following steps 1-5 again. However, if you select new criteria from the existing treemap and click the check mark icon, a new treemap will appear in place of the existing treemap.

Using Heat Maps

Heat maps present data in table form using colors. A simple example of this is unemployment data over the years by month. Each row represents one of the twelve months, and each column represents a year in the time frame. Colors could range from very pale red (lower unemployment rates) to dark red (higher unemployment rates). You could use more than one spectrum of color, especially if you wanted your heat map to be more specific. Using the unemployment example above, you could select the range of reds for 1%-5% unemployment, and then add a range of blues for 6%-10% unemployment.

Note: Only properties with numeric formats are eligible for use with heat maps.

1. Select whether you want the heat map to use values **By property** or **By relationship**.
2. Select whether you want to use entity or relationship data in the heat map.
3. Select the field you want to use in the rows of the heat map.
4. Select the field you want to use in the columns of the heat map.
5. Select from the **Value** field whether you want the heat map to include incoming counts, outgoing counts, or both.
6. Click the check mark icon or continue with advanced settings.

Advanced settings are not required; they allow you to fine-tune the heat map configuration.

1. Select how you want the rows to be sorted. You can sort them ascending or descending, by field name or by value. Using the unemployment example above, you could sort the rows ascending (or descending) by month, or you could sort them ascending (or descending) by value.
2. Select how you want the columns to be sorted. You can sort them ascending or descending, by field name or by value. Using the unemployment example above, you could sort the columns ascending (or descending) by year, or you could sort them ascending (or descending) by value.
3. Specify the range of colors you want to use in your heat map in the **Low Value** and **High Value** fields.

4. Specify the maximum number of rows that can be on a single page. The minimum you can enter is 15 and the maximum you can enter is 50.
5. Specify the maximum number of columns that can be on a single page. The minimum you can enter is 5 and the maximum you can enter is 50.
6. Click the check mark icon.

You can create an additional heat map by selecting Charts > Heat map and following steps 1-6 again. However, if you select new criteria from the existing heat map and click the check mark icon, a new heat map will appear in place of the existing heat map.

Using Timelines

Timelines present data across a span of time. They can be used for entities, relationships, or both. You can plot events, actions, and so on over a course of time, and once the timeline is created you can zoom in or out to focus on more or fewer items.

1. Click either **Entities** or **Relationships**. You must complete information for one or both of these options to generate a timeline.
2. Select a field from which the **Start date** can be extracted. You must complete this field to generate a timeline.

Note: If any component that meets the criteria of the timeline does not have an associated start date, it will not be included in the chart.

3. Select a field from which the **End date** can be extracted. You can use the same field for the end date as you did for the start date; alternatively, you can use a different field.
4. If you did not select an end date, select a **Duration** for the timeline. This figure represents the number of days from the start date through which the timeline should take place. This field can only contain values if the **End date** field is empty.
5. Click the check mark icon or continue with advanced settings.

Advanced settings are not required; they allow you to fine-tune the timeline configuration.

1. Click **Group items** to group elements in the timeline by type or by component.
2. Select how you want the data to be grouped. Selecting **By type** will group entities based on values in the `_stp_type` field and relationships based on the value of the relationship label. Selecting **By components** will group items based on whether they are entities or relationships. Entities with durations are shown in blue bars, and instant-item entities are shown as blue dots. Relationships are shown in green.
3. Specify the **Date format** you want to use in your timeline. The default date is based on the format used in the start date value format.
4. Click the check mark icon.

You can create an additional timeline by selecting Charts > Timeline and following steps 1-5 again. However, if you select new criteria from the existing timeline and click the check mark icon, a new timeline will appear in place of the existing timeline.

Viewing Timeline Information

- Instant items on a timeline do not have a label, but if you hover over the item the label will appear as a ToolTip. Likewise, ToolTips for entities show their labels as well. However, ToolTips for relationships include the source, target, and relationship labels.
- When you select an item on the canvas, that item is also selected in the timeline. Conversely, when you select one or more items on the timeline, those items are selected on the canvas as well. You can also see detailed information for the selected items on the Model Components Pane and the Selection Pane.
- The vertical slider on the bottom of the timeline allows you to scroll back and forth across all dates in the timeline.
- You can adjust the level of zoom in the timeline either with the mouse wheel or by dragging the handles on either end of the slider.
- Double-click the slider selection to have the timeline include data for the entire range of dates.
- Double-click the handle on the left side of the slider to change the visible timeframe to begin at the start date. For example, if your date range was from 1990-2000, and your current view showed 1995-1997, clicking the left handle would change your view to start at 1990.
- Double-click the handle on the right side of the slider to change the visible timeframe to end at the end date. For example, if your date range was from 1990-2000, and your current view showed 1995-1997, clicking the right handle would change your view to end at 2000.

Using Filters

The Filters section of Relationship Analysis Client allows you to hide selected entities or relationships from your canvas. Selection and Type filters enable you to clear out extraneous elements—those whose properties don't meet your requirements or whose data doesn't factor into your current interest. Using these kinds of filters can reveal important connections in the elements left behind. Temporal filters enable you to capture a segment of data in a specific time frame. They can also help you compare the activity of your network from one time to another. Numeric filters enable you to capture data containing a property that falls within a range that you specify. For instance, if your data contains information about insurance claims, you might have a field called "NumberOfVisits" for the number of times a doctor has claimed treating a particular patient, with relationships in your model reflecting that relationship. If you are trying to isolate relationships with 10 or more visits, you could filter your data based on that field, and set the minimum value to 10. This would remove relationships that have a number lower than 10 in the NumberOfVisits field.

Note: When an entity is filtered, all connecting relationships will also be filtered. When a relationship is filtered, the connecting entities are **not** filtered.

Using Selection Filters

Follow the steps below to use Selection filters.

1. With a model open, click the **View** tab.
2. Select the entities or relationships you want to hide by clicking on them in the primary canvas or by running a **selection**.
3. In the **Filter Selection** group, click either **Entities** or **Relationships**, whichever is appropriate.

- Click **Selected** or **Non-Selected**, depending on whether you want the entities or relationships that are selected to disappear, or if you want the entities or relationships that are **not** selected to disappear.

If you wish to restore the entities or relationships you removed, click **Clear all** in the **Selection** group.

Note: This restore feature works only for entities or relationships removed using Selection filters.

Using Type Filters

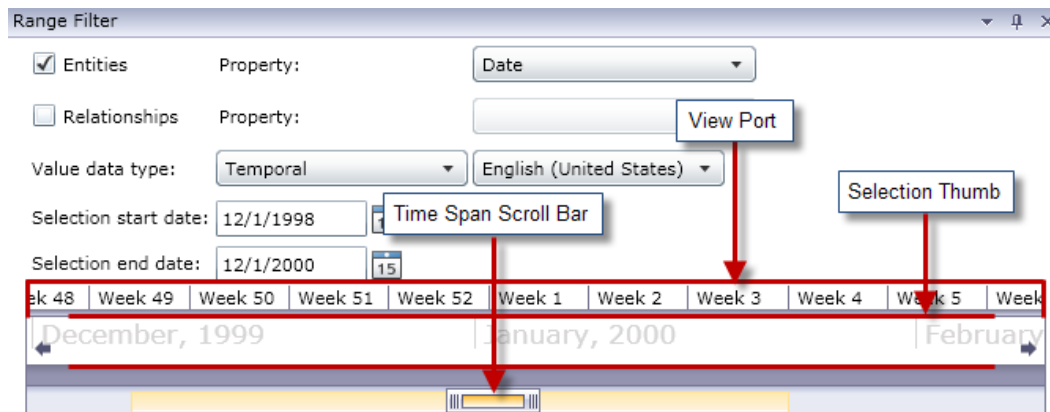
Follow the steps below to use Type filters.

- With a model open, click the **View** tab.
- In the **Filter By Type** group, click **Entities** or **Relationships**.
- Click the entity type (for example, person, location, event, and so on) or relationship property (for example, brother, Afghanistan, bombing, and so on). These types are specified in the Write to Hub stage and viewed in the `_stp_type` property of an element. The elements that meet the criteria of your filter will be hidden from view.

Using Temporal Filters

Perform this procedure to use Temporal filters.

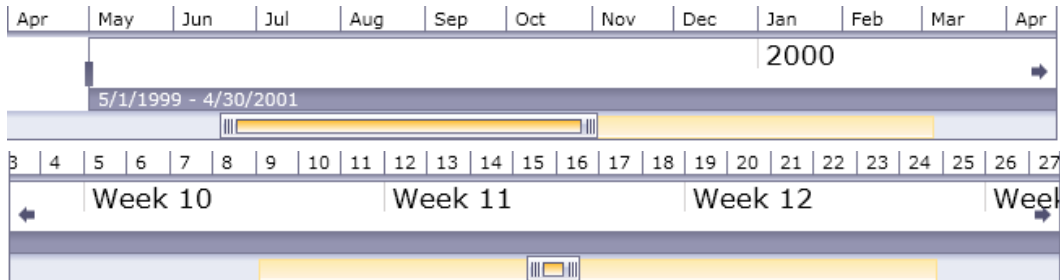
The image below identifies parts of a temporal filter. You should see this entire pane when you activate a temporal filter. If you do not see the Time Span Scroll Bar at the bottom of the pane, increase the height of the filter pane until the entire pane is visible.



The **View Port** is the visual representation of the span of dates you select. It includes the **Selection Thumb**, which enables you to change the dates in the filter by clicking and dragging the handles on either end of the thumb. Note that the Selection start date and the Selection end date fields update dynamically when you let go of the handles.

The View Port also includes the **Time Span Scroll Bar**, which enables you to zoom in (or out) on the selected dates by clicking and dragging the handles on either end of the bar. As shown in the two examples below, when you drag the handles closer to each other, you will see years turn into

months, months into weeks, and weeks into days. The reverse occurs when you drag the handles away from each other. The benefit of this tool is that as you zoom in or out with the Time Span Scroll Bar, you then have more flexibility and control over the time span the Selection Thumb covers; the thumb will update dynamically as you make changes with the scroll bar. If initially the thumb spanned a year or month, you could adjust the view with the scroll bar and then use the thumb to span a week or day.



1. With a model open, click the **View** tab.
2. In the **Tools** group, click **Range Filter**. The Range Filter pane will open underneath the Primary Pane. The pane should look similar to the image above. If it does not, you may need to increase the size of the pane to reveal the bottom portion.
3. Select **Entities** and/or **Relationships**, depending on which set of data you want to filter.
4. Select the **Entity property** and/or the **Relationship property** that contains the date information. **Note:** Entities or relationships that do not contain valid data in the property you select here will not be filtered and thus will remain visible in the model.
5. In the **Value data type** drop-down, select "Temporal" if it is not already selected by default. Then select the language in which your dates are formatted.
6. The **Selection start date** field will auto-populate after you complete step 4, but you can enter a different value here if you want to increase the minimum of the range. For example, if you are looking at insurance data and filtering on dates during which claims were filed, the earliest date in your data may be 01/02/2001. However, perhaps you don't want to look at claims from the first few years of data. You could, hypothetically, change the Selection start date to 01/02/2004. This would cause all data with dates 01/01/2004 and earlier to be removed from the model.
7. The **Selection end date** field will auto-populate after you complete step 4, but you can enter a different value here if you want to decrease the maximum of the range. For example, if you are looking at insurance data and filtering on dates during which claims were filed, the latest date in your data may be 12/31/2010. However, perhaps you don't want to look at claims from the last few years of data. You could, hypothetically, change the Selection end date to 12/31/2007. This would cause all data with dates 01/01/2008 and later to be removed from the model.
8. Use the **Selection Thumb** and the **Time Span Scroll Bar** to set a more specific range for which you want to return results. The model will dynamically update, removing entities and relationships, as you move the slider and let go of the handles.

Using Numeric Filters

Follow the steps below to use Numeric filters.

1. Click the **View** tab.
2. In the **Tools** ribbon, click **Range Filter**. The Range Filter pane will open.
3. Select **Entities** and/or **Relationships**, depending on which set of data you want to filter.
4. Select the **Entity property** and/or the **Relationship property** that contains the numeric information. **Note:** Entities or relationships that do not contain valid data in the property you select here will not be filtered and thus will remain visible in the model.
5. In the **Value data type** drop-down, select "Numeric".
6. The **Minimum value** field will auto-populate after you complete step 4, but you can enter a different value here if you want to increase the minimum of the range. For example, if you are looking at insurance data and filtering on claim numbers, the lowest claim number in your data may be 1001. However, perhaps you don't want to look at the first thousand claims. You could, hypothetically, change the Minimum value to 2001. This would cause all data with claim numbers of 2000 and lower to be removed from the model.
7. The **Maximum value** field will auto-populate after you complete step 4, but you can enter a different value here if you want to decrease the maximum of the range. Using the example in Step 6, the highest claim number in your data may be 9000, and perhaps you don't want to view the last thousand claims. You could, hypothetically, change the Maximum value to 8000. This would cause all data with claim numbers of 8001 and higher to be removed from the model.
8. Use the slider to set the range for which you want to return results. The model will dynamically update, removing entities and relationships, as you move the slider.

Using Reports

You can generate three reports from within the Relationship Analysis Client. When you click the **Data** tab, then click the **Reporting** button, you will be prompted to select from three options:

- Model Metadata
- Entity Profile
- Snapshot

If a report name is grayed out, you have not fulfilled the criteria required to create that report. For example, an entity must be selected prior to generating an Entity Profile Report.

Using the Entity Profile Report

The Entity Profile Report enables you to see all data associated with a particular entity in one place. Information in an Entity Profile Report includes the following:

- The entity's label (usually its name)
- The entity's type (for example, "person" or "event")
- The total number of relationships connected to the entity
- A list of each property tied to the entity as well as the value for each of those properties. For instance, a property of "Latitude" may have a value of "23.424076."
- A list of each incoming relationship, with the source, relationship label, and target for that relationship. In this list, the target will always be the entity on whom the report is based. For

instance, in a model used to identify insurance fraud, this list might have an entry with a source of someone's name, a relationship of "is_patient_of", and a target of "Dr. Smith."

- A list of each outgoing relationship, with the source, relationship label, and target for that relationship. In this list, the source will always be the entity on whom the report is based.
- A history of changes for that entity and its relationships.

Note: This information will be returned only if you have activated the History function. To do so,

1. Open the file `<Spectrum™ Technology Platform install path>/server/modules/hub/hub.properties`.
2. Uncomment and set the hub audit history line to "true" as follows:
`hub.audit_history=true`.
3. Restart the Spectrum Technology Platform server.

The history feature groups data by version number, in reverse chronological order, and provides a date and time stamp for each version. It also provides the user name of the person who made the changes. Additionally, it displays the following details about any given change:

- **Operation**—The action that took place: added, modified, or deleted. Note that if you delete an entity, it will be shown only if it is re-added to the model. However, if you delete relationships or properties, those deletions will appear in the history.
- **Type**—The component that changed: property or relationship
- **Name**—The name of what changed. For properties, this is the ID of the property that changed. For entities whose relationship changed, this is the source and target IDs plus the relationship name. For example, if you were using insurance fraud data and you changed something about a relationship and then viewed the history of one of the entities tied to that relationship, you might see something like this: "Provider:Dr. John Smith=>Treated=>Patient:Jane Doe".
- **Value**—The new value that was entered (if any)
- **Previous Value**—The value prior to the change (if any)

To generate an Entity Profile Report, simply select an entity within a model, then right-click that entity and select **Entity Profile Report** from the context menu. The report will appear on top of the model; use the arrows to scroll through the pages. You can choose to have the report dockable or floating by right-clicking the report's tab and selecting the appropriate option. You can also print the report or save it in PDF format.

To close the report, right-click the report's tab and select **Hide**.

Note: If you select this option the report will be deleted; you will need to regenerate it to view the report again.

Using the Model Metadata Report

The Model Metadata Report provides counts for model, entity, and relationship metadata associated with a model. By using Advanced Settings, you can determine how much or how little of that data you want the report to return; the default settings return all data.

Note: You must select at least one of the options in Advanced Settings; otherwise, the report will simply say, `No data to display`.

Assuming you have selected at least one option in Advanced Settings, information in a Model Metadata Report always includes the following:

- The model's name
- The total number of entities in the model
- The total number of relationships in the model

Optional information in a Model Metadata Report includes the following:

Model Metadata

- **Screen capture**—A screen capture that depicts the model's entity types (for example, "Provider" or "Patient_Name") and relationship types (for example, "Treated" or "Filed_A_Claim_With")
- **Entity types**—Entity types and the counts for each type
- **Properties** (under Entity types)—Entity properties, the format of the property (for example, "string" or "Boolean"), and the counts for each property
- **Relationships**—Relationship types and the counts for each type
- **Properties** (under Relationships)—Relationship properties, the format of the property (for example, "string" or "Boolean"), and the counts for each property

Entity Metadata

The following information, broken down by each entity type:

- **Include entity type metadata**—The total number of entities for a given type (for example, a "Person" entity might have 5 Doctor entities and 3 Patient entities, giving you a count of 8 for the "Person" type)
- **Incoming relationships per type**—The number of incoming relationships for a given entity type
- **Outgoing relationships per type**—The number of outgoing relationships for a given entity type
- Source, relationship, and target connections for any given entity type's incoming and outgoing relationships, as well as counts for each (for example, your model may have 4 instances of "Joe Smith" [source] "Visited" [relationship] "Robert Johnson, M.D." [target])
- **Properties per type**—The total number of properties for a given entity type
- Property names and property types for a given entity type, as well as counts for each

Relationship Metadata

The following information, broken down by each relationship type:

- **Include relationship metadata**—The total number of relationships for a given type (for example, for a relationship of "Visited": 3 Person entities could have "Visited" Robert Johnson, M.D., and 4 other Person entities could have "Visited" Patricia Benson, M.D., giving you a count of 7 for the relationship type "Visited")
- **Connections per relationship**—Source, relationship, and target connections for a given type, as well as counts for each (using the previous example, 3 Persons [source] could have "Visited" [relationship] Robert Johnson, M.D. [target])

- **Properties per relationship**—The total number of properties for a given relationship type
- Property names and property types for a given relationship type, as well as counts for each

To generate a Model Metadata Report, have a model open and click **Model Metadata** from the Reporting ribbon on the Data tab. The report will appear on top of the model; use the arrows to scroll through the pages. If you want to customize your report and have it include some but not all of the optional information listed above, click **Show advanced settings** and select the options you want to include. The report will update dynamically as you select and deselect options.

You can choose to have the report dockable or floating by right-clicking the report's tab and selecting the appropriate option. You can also print the report or save it in PDF format.

To close the report, right-click the report's tab and select **Hide**.

Note: If you select this option the report will be deleted; you will need to regenerate it to view the report again.

Using the Snapshot Report

The Snapshot Report enables you to capture the current state of a model. This report is essentially a picture of the model at any given time; as such, its contents are not editable. However, there are several aesthetic modifications you can make to the image, such as resizing, rotating, flipping, cropping, adding colors, and sharpening. You can also add text to the report and define where on the page the text should be placed, based on horizontal and vertical positions. Additionally, you can save the report as a BMP or PNG file.

Back Up Data Hub Models

You can use the Relationship Analysis Client to back up Data Hub models in a visual interface.

1. Open the Relationship Analysis Client.
2. Click **Manage**.
3. Select the model you want to back up then click **Backup**.
4. Click **Yes** then **OK**.

After you back up models, you can view and manage backups on the **Backups** tab.

Restoring Models

You can use the restore feature in the Relationship Analysis Client to restore models that were backed up either on demand or by schedule.

Backups are stored in `<Spectrum™ Technology Platform install path>/server/modules/hub/db/backups`. This folder contains one folder for each backed up model. A backup folder contains all the data and metadata for its respective model. Queries, monitors, themes, and security entries are stored in the Spectrum configuration database and must be backed up as part of the Spectrum backup schedule.

1. With Relationship Analysis Client open, click the **Manage** button.
2. Click the **Backups** tab.
3. Select a model from the list in the **Model Management** dialog box by clicking it.
4. Click **Restore** tab, then **Yes** to confirm the restore.
5. Click **OK**.

Back Up Data Hub Models

You can use the Relationship Analysis Client to back up Data Hub models in a visual interface.

1. Open the Relationship Analysis Client.
2. Click **Manage**.
3. Select the model you want to back up then click **Backup**.
4. Click **Yes** then **OK**.

After you back up models, you can view and manage backups on the **Backups** tab.

Changing the Appearance of Model Data

The Relationship Analysis Client of the Data Hub Module has several tools you can use to alter the appearance of model data. This feature allows you to highlight or call out specific entities and relationships that you want focus on.

- [Using Multiple Views](#)
- [Using Default Options](#)
- [Changing the Labels of Entity Types](#) on page 194
- [Using Entity Settings](#)
- [Using Relationship Settings](#)
- [Using Gradient Settings](#)
- [Using Themes](#)
- [Using Layouts](#)

Using Multiple Views

In Relationship Analysis Client, you can create multiple views of the same model. Likewise, you can compare different parts of the same model using multiple views. This feature allows you to manipulate model data while preserving the integrity of the original model. You can compare models as you toggle among views to identify possible relationships, establish how many events of a certain type (such as flights to a particular country) took place on a particular date or time, consider the same data in varying layouts, and so on.

To create an additional view for your model, click **New View** in the **Graph** pane. An identical model will open in a new tab with the name "View 1." You can manipulate the data in this view just as you would your original model. After you make changes to the original model or another view for that

model, you will notice that as you bring each view to the forefront by clicking its tab, the surrounding panes will update accordingly with information for that view.

When you add and remove entities and relationships or modify relationship properties that are used as weights, centrality measurements for that model will change. By creating multiple views and altering the model in those subsequent views, you can rerun centrality algorithms to see how the changes affect the model. For example, if Osama bin Laden were the most central entity in a terrorism model and you removed him from that model, he obviously would no longer be the most central entity. This might uncover another primary entity that did not stand out in the previous view. Using multiple views allows you to make these changes and examine alternative scenarios while keeping you from altering the original model.

Click [here](#) for more information on centrality measures.

Using Default Options

Default options can be designated for the following:

- [General settings](#)
- [Entity settings](#)
- [Relationship Settings](#)

Using General Settings

This feature allows you to set default options for animation and undo/redo.

Animation

1. Click the **Animate layout transitions** box if you want the Relationship Analysis Client to show animation during transitions from one view to another.
2. If you clicked the box in Step 1, select the number of seconds you want the animation to appear in the **Duration** control. Default is 1 second and the maximum is 10 seconds.

Undo

1. Click the **Track changes for undo and redo** box if you want to use undo and redo functionality.
2. If you clicked the box in Step 1, select the number of instances you want the Relationship Analysis Client to track in the **Max changes** control. The default is 5 instances and the maximum is 25 instances.

Selected Items This option enables you to change how selected entities or relationships are represented. You can change their color or the thickness of their outline.

Current Item This option enables you to change how the current entity or relationship is represented. If you have multiple items selected, the last item is considered the current item. This item's information is displayed in the Selection pane of the Relationship Analysis Client. As with selected items, you can change the current item's color or the thickness of its outline.

Query result limits This option enables you to impose limits on the number of entities or relationships that can be loaded into the Relationship Analysis Client. The maximum for both fields is 1,000 to avoid overloading the Relationship Analysis Client tool.

Entities panel style option This option enables you to determine the maximum number of properties that should appear on the Panel Style tool when used to style entities. The default is 5 properties, and the maximum is 10.

Using Entities Decoration

This feature allows you to determine default shape and label settings for entities.

Entity Shape Settings

- **Shape**—Sets the default shape of entities.
- **Image**—Sets an image to represent entities.
- **Width**—Sets the default width of the entities.
- **Maintain aspect ratio**—Automatically adjusts the height based on the width, to maintain the aspect ratio of the entities.
- **Height**—Sets the default height of the entities.
- **Fill Color**—Sets the default interior color of the entities.
- **Shadow**—Adds the appearance of a shadow behind the entities.
- **Border Color**—Sets the default color of the entities' borders.
- **Thickness**—Sets the default width of the entities' borders.

Entity Label Settings

- **Show labels**—Activates labels on the model.
- **Font**—Sets the default font of the labels.
- **Font size**—Sets the default font size of the labels.
- **Location**—Sets the default location of the entity labels as inside or outside the entities.
- **Position**—Sets the default position of the entity labels relative to the entities. The label can appear on any side or corner of the entity.
- **Color**—Sets the default color of the labels.
- **Wrap label**—Forces labels exceeding maximum width to break onto multiple lines.
- **Trim label**—Trims labels exceeding maximum width.
- **Maximum width**—Sets the maximum width allowable for labels.

Using Relationships Decoration

This feature allows you to determine line and label settings for relationships.

Relationship Line Settings

- **Line color**—Sets the default color of the relationships.
- **Cross style**—Sets the default appearance for crossed lines.
- **Line type**—Sets the default appearance of the relationships.
- **Thickness**—Sets the default width of the relationships.
- **Source arrow**—Sets the default appearance of the starting point (source) of relationships.
- **Scale**—Sets the default size of the source arrow.

- **Target arrow**—Sets the default appearance of the ending point (target) of relationships.
- **Scale**—Sets the default size of the target arrow.

Relationship Label Settings

- **Show labels**—Activates relationship labels on the model.
- **Color**—Sets the default color of the relationship labels.
- **Font**—Sets the default font for the relationship labels.
- **Font size**—Sets the default size of the relationship label.
- **Distance from relationship**—Sets how close to or far from the labels are to the relationships.
- **Wrap label**—Forces labels exceeding maximum width to break onto multiple lines.
- **Trim label**—Trims labels exceeding maximum width.
- **Maximum width**—Sets the maximum width allowable for labels.

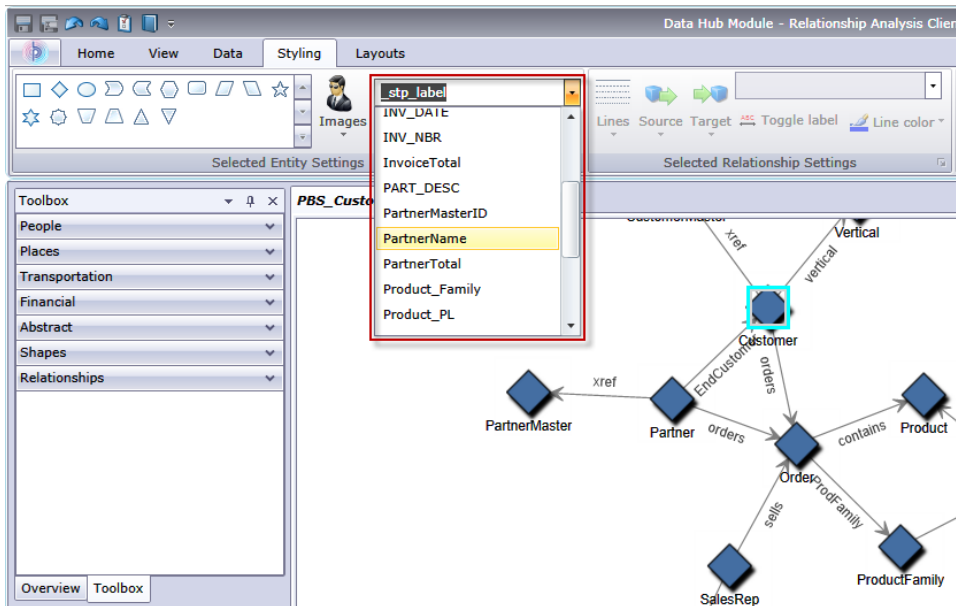
Changing the Labels of Entity Types

Perform this procedure to select a property that is associated with that entity type to populate the label.

This feature enables you to change the label for entity types by modifying their metadata. The default label is `_stp_label`, which is generated when you create the model and is based on the input field you associated with the entity type when you configured the model. However, there may be situations where you want to change the labels that are appearing in your model. For example, you could have a model of customer data containing an entity type of "Customer" that is populated by the "CustomerID" input field and has tied to it a property called "Name". With that configuration, all entities of the "Customer" type would show the customer's numeric ID as the label. You could change the label property to "Name", which would result in all Customer entity labels showing customer names instead of numeric IDs.

1. With Relationship Analysis Client open, click the **Open** button.
2. Select a model from the list by clicking it.
3. Click the **Model Detail** tab.
4. Click **Open Metadata**.
5. Select the entity type you want to modify.
6. Click the **Styling** tab.
7. From the **Label property** drop-down, shown in the red box below, select the property whose data you want to populate the label for this entity type.
8. Save the model.

If you are using the Data Hub Browser, refresh the screen and you will see the updated labels in the model's metadata.



Changing the Appearance of Selected Entities

The Selected Entity Settings ribbon on the Styling tab allows you to determine the appearance of entities. This gives you further control over your data by allowing you to manually make certain entities stand out by using size, color, border thickness, and so on. For instance, if you are looking at data involving the United States of America, you could use the [Selection Tool](#) to identify and select states with population greater than 10,000,000. Then, you could change the entities of those states to be a different shape and color than the entities for states with population less than 10,000,000.

Settings made here will apply only to selected entities, whereas settings made in the Gradient section will apply to all entities.

With one or more entities selected, change any of the following settings.

Entity shape

Allows you to change the shape of selected entities.

Image

Allows you to change the image for the selected entities.

Property

Allows you to change the selected entity's current label to a different property.

Toggle label

Activates and deactivates labels for selected entities.

Fill color

Allows you to change the interior color of the selected entities.

Entity Settings

Access the following settings by clicking the arrow in the bottom-right corner of the Entity Settings area of the ribbon. Allows you to use shape and label settings to alter the appearance of entities.

Shape Settings

- **Shape**—Sets the shape of the entities.
- **Image**—Allows you to change the image for the selected entities.
- **Width**—Sets the width of the entities.
- **Maintain aspect ratio check box**—Automatically adjusts the height based on the width, to maintain the aspect ratio of the entities.
- **Height**—Sets the height of the entities.
- **Fill color**—Sets the interior color of the selected entities.
- **Shadow**—Adds the appearance of a shadow behind the selected entities.
- **Border color**—Sets the color of the selected entities' borders.
- **Thickness**—Sets the width of the selected entities' borders.

Label Settings

- **Show labels**—Activates labels on the model.

Note: Large models will load and change layouts more quickly if labels are turned off for both entities and relationships.

- **Font**—Sets the font of the labels.
- **Font size**—Sets the font size of the labels.
- **Label location**—Sets the location of the entity labels as inside or outside the entities.
- **Label position**—Sets the position of the entity labels relative to the entities. The label can appear on any side or corner of the entity.
- **Color drop-down box**—Sets the color of the labels.
- **Wrap label**—Forces labels exceeding maximum width to break onto multiple lines.
- **Trim label**—Trims labels exceeding maximum width.
- **Maximum width**—Sets the maximum width allowable for labels.

Click **Apply** to make shape and label settings changes to your model.

Changing the Appearance of Selected Relationships

The Selected Relationship Settings ribbon in the Styling tab allows you to determine the appearance of relationships. This gives you further control over your data by allowing you to manually make certain relationships stand out by using color, line type, arrow type, and so on. You can also turn labels off and on and determine how far from the entity the label should be. For instance, if you are looking at data involving the insurance claims, you could use the **Selection Tool** to identify and select relationships with a label of "Claimant." Then, you could change those relationships to be a different type and color than the relationships with different labels. You could also change the font and size of the selected relationships' labels.

Settings made here will apply only to selected relationships, whereas settings made in the Gradient section will apply to all relationships.

With one or more relationships selected, change any of the following settings.

Lines

Sets the appearance of selected relationships.

Source

Sets the appearance of the starting point (source) of selected relationships.

Target

Sets the appearance of the ending point (target) of selected relationships.

Property

Allows you to change the selected relationship's current label to a different property.

Toggle label

Activates and deactivates labels for selected relationships.

Line color

Sets the color of selected relationships.

Edge Settings

Access the settings below by clicking the arrow in the bottom-right corner of the Edge Settings area of the ribbon. Allows you to use line and label settings to alter the appearance of relationships.

Line Settings

- **Line color**—Sets the color of the relationships.
- **Line type**—Sets the appearance of the relationships.
- **Thickness**—Sets the width of the relationships.
- **Source arrow**—Sets the appearance of the starting point (source) of selected relationships.
- **Scale**—Sets the size of the source arrow.
- **Target arrow**—Sets the appearance of the ending point (target) of selected relationships.
- **Scale**—Sets the size of the target arrow.

Label Settings

- **Show labels check box**—Activates relationship labels on the model.

Note: Large models will load and change layouts more quickly if labels are turned off for both entities and relationships.

- **Color drop-down box**—Sets the color of the relationship labels.
- **Font**—Sets the font for the relationship labels.
- **Font size**—Sets the size of the relationship label.
- **Distance from relationship**—Sets how close to or far from the labels are to the relationships.

- **Wrap label**—Forces labels exceeding maximum width to break onto multiple lines.
- **Trim label**—Trims labels exceeding maximum width.
- **Maximum width**—Sets the maximum width allowable for labels.

Click **Apply** to make line and label settings changes to your model.

Using Gradient Settings

The Gradient ribbon on the Styling tab enables you to visually change the style of a model using actual values in the data. You can represent model elements and highlight important elements based on their values by starting at one setting for the lowest value and another setting for the highest value. You do this by determining the appearance of entities and relationships. For instance, if your model uses insurance claim data and one of the entity properties is the dollar amount of claims, you could set pink as the color for the entity with the lowest claim amount and red as the color for the entity with the highest claim amount, and the entities for everything in between those two dates would fall somewhere between pink and red.

Settings made here will apply to all entities or relationships, whereas settings made in the Entity Settings or Relationship Settings are apply only to selected entities or relationships.

For more information on gradient settings:

- [Using Gradient Settings with Entities](#)
- [Using Gradient Settings with Relationships](#)

Entity Gradient Settings

These settings allow you to use color and size to represent the range of differences among entities. For example, if your model uses cities as entities, and one of the fields in your model's database is population, you could show the color gradient from yellow (smallest city) to green (largest city). Likewise, you could set the shape as a star, and the smallest city would have a small star and the largest city would have a large star. (You determine the size of the largest and smallest stars, and everything else falls in between.)

Gradient Value

- **Gradient property**—Sets the field on which the gradient is based.
- **Override least/most significant values**—Overrides the default lowest and highest values used for gradient.
- **Least significant value**—Shows the value for the lowest gradient property.
- **Most significant value**—Shows the value for the highest gradient property.
- **Low values are more significant**—Specifies that low values are preferable to high values, such as with a scale of 1-5, when "1" is the highest rating.

Color Settings

- **Apply to entities color**—Activates the color settings options.
- **Use these colors**—Overrides the existing element's color.
- **Low value**—Sets the color for the entities with the lowest value.

- **High value**—Sets the color for the entities with the highest value.
- **Reduce color range**—
- **Scale entity color by**—Retains the entity's fill color but with varying hue, saturation or brightness (depending on each entity's gradient value).
- **Low %**—Sets the low percentage for the scale. For example, if this number is "20" and you chose "Saturation" in the previous field and red as the fill color, entities with the lowest gradient value would have a 20% red color.
- **High %**—Sets the high percentage for the scale. For example, if this number is "100" and you chose "Saturation" in the previous field and red as the fill color, entities with the highest gradient value would have a 100% red color.

Size Settings

- **Apply to entities size**—Activates the size settings options.
- **Use these sizes**—Allows you to select the sizes you want the gradient to use.
- **Low value**—Sets the size for the entities with the lowest value.
- **High value**—Sets the size for the entities with the highest value.
- **Scale entity size by**—Changes the entities' sizes, varying overall, in height, or in width (depending on each entity's gradient value).
- **Low %**—Sets the low percentage for the scale. For example, if this number is "20" and you chose "Width" in the previous field, entities with the lowest gradient value would have a width that is 20% of the highest number's width.
- **High %**—Sets the high percentage for the scale. For example, if this number is "100" and you chose "Width" in the previous field, entities with the highest gradient value would have a width that is 100%.
- **Scale label size**—Changes the size of the entity's label relative to the size of the entity.

Click **OK** to make shape and label settings changes to your model.

Relationship Gradient Settings

These settings allow you to use color and line settings to represent the range of differences among relationships. Using the Entity Gradient Settings example, your model with city information may have a field that includes each city's rank based on population. The city having the highest population, with a rank of "1", could have a color gradient of lavender, and the city having the lowest population, with a rank of 100 (presuming there were 100 cities in the model), could have a color gradient of dark purple. Likewise, you could set the thickness of the relationship to go from thinnest (lowest-ranked city) to thickest (highest-ranked city). You determine the size of the thinnest and thickest lines, and everything else falls in between.

Note: You would need to check the **Low values are more significant** box for this example.

Gradient Value

- **Gradient property**—Sets the field on which the gradient is based.
- **Override least/most significant values**—Overrides the default lowest and highest values used for gradient.

- **Least significant value**—Shows the value for the lowest gradient property.
- **Most significant value**—Shows the value for the highest gradient property.
- **Low values are more significant**—Specifies that low values are preferable to high values, such as with a scale of 1-5, when "1" is the highest rating.

Color Settings

- **Apply to relationships color**—Activates the color settings options.
- **Use these colors** —Allows you to select the colors you want the gradient to use.
- **Low value**—Sets the color for the relationships with the lowest value.
- **High value**—Sets the color for the relationships with the highest value.
- **Scale relationship color by** —Retains the relationship's color but with varying hue, saturation or brightness (depending on each entity's gradient value).
- **Low %**—Sets the low percentage for the scale. For example, if this number is "20" and you chose "Saturation" in the previous field and red as the fill color, relationships with the lowest gradient value would have a 20% red color.
- **High %**—Sets the high percentage for the scale. For example, if this number is "100" and you chose "Saturation" in the previous field and red as the fill color, relationships with the highest gradient value would have a 100% red color.

Line Settings

- **Apply to relationships thickness**—Activates the line settings options.
- **Use these sizes**—Allows you to select the sizes you want the gradient to use.
- **Low value**—Sets the size for the relationships with the lowest value.
- **High value**—Sets the size for the relationships with the highest value.
- **Scale relationship thickness by**—Changes the relationships' sizes (depending on each entity's gradient value).
- **Low %**—Sets the low percentage for the scale. For example, if this number is "20", relationships with the lowest gradient value would have a size that is 20% of the highest number's size.
- **High %**—Sets the high percentage for the scale. For example, if this number is "100", relationships with the highest gradient value would have a size that is 100%.
- **Scale label size**—Changes the size of the relationship's label relative to the size of the relationship.

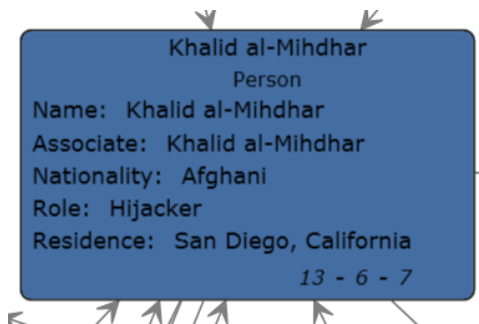
Click **OK** to make color and line settings changes to your model.

Using Panel Style

The Panel Style ribbon on the Styling tab enables you to view property data for each entity within the canvas, providing more model data at a time than can be shown in the Model Components or Selection panes.

The panel will display a **title**, which is the entity's identification and is fulfilled by the `_stp_id` field in the model. It can contain up to 300 characters. The panel also contains data for up to 10 of the most-used **properties** for its type; if entities contain more than 10 properties, the panel will display data for the 10 that are used most frequently. See [Entities panel style option](#) for more information on configuring the number of properties on panels. Property data can contain up to 200 characters

each. The panel also provides the number of **connections** the entity has; it displays the total number of connections, the number of entities coming in (predecessors), and the number of entities going out (successors).



In the image above, the **title** of the panel, or the entity's identification, is Khalid al-Mihdhar, and it tells us he is a person. The Name, Associate, Nationality, Role, and Residence fields are **properties** associated with the entity Khalid al-Mihdhar. The number 13 at the bottom of the panel represents the total number of **connections** coming into and going out of this entity. You can see by the incoming arrows and the number "6" on the bottom of the panel that there are six predecessors to this entity. Likewise, there are seven successors. The number of predecessors plus the number of successors will always add up to the total number of connections.

The Panel Style view updates dynamically as you edit the values of properties and add entities manually or by using the Discovery feature.

You can activate Panel Style simply by clicking the **Entities** button on the **Panel Style** ribbon in the **Styling** tab. You can turn off Panel Style by clicking that button again.

If you remove entities that are connected to other entities and you are using Panel Style, you will see a small plus sign in the bottom-right corner of panels for selected entities who have missing connections, as indicated by the red arrow in the following image. This will also occur if you open a partial model that is missing entities that were previously connected to those entities returned in the partial model. If you wish to restore the entities that were previously connected to the selected entity, click the plus sign. The connected entities will reload into the model.

Using Themes

The Themes ribbon on the Styling tab enables you to apply styles and filters to elements in your model based on conditions you set. You can predetermine aesthetic values for entities and relationships that meet the criteria set in your conditions, and you can use filters to automatically hide those elements from your model.

1. Click the **Define** button on the **Themes** ribbon in the **Styling** tab.
2. Click **Add**.
3. Enter a name for the theme and click **Add**.
4. Select from the list of styles or filters that you want to include in your theme. You can choose from **entity decorations**, **relationship decorations**, **gradients**, or **filters**.
5. Click **Add Conditions**.

6. Select the property for which you want to create a condition and apply a theme. For example, you could select a property called "Date" and complete the expression to say "Equals" and "09/11/2001".
7. Edit the options for your style or filter to configure the theme.
8. Click **OK** twice.

Once a theme is defined, you can click **Define > Apply** or select it from the Themes drop-down to apply it to your model. You can also modify and delete themes by clicking **Define**, selecting a theme, and then clicking **Modify** or **Delete**.

If you have a theme that contains characteristics you would like to include in another theme, you can make a copy of the first theme, rename it, and alter it accordingly. For example, you could have a model with entities that represent states, and one of the fields in your model's database is population. You could create "Theme 10" for states with a population of more than 10,000,000 and set those entities' fill color as green (among other settings, such as the shape of the entity or an image used in place of a shape). If you wanted to call out states with a population of at least 5 million but no more than 10 million, you could copy Theme 10, save it as "Theme 5", and change the Fill Color to blue. That way, entities representing large states would look different from entities for the remaining states in your model, and they would look similar to each other, but just different enough that you can see which have a population of 5 million to 10 million and which have a population of 10 million or more.

To copy a theme:

1. Click **Define**.
2. Select a theme from the list and click **Copy**.
3. Enter a name for the new theme in the **Name** field.
4. Click **Add** to apply additional settings to the theme, or click **Modify** to alter the existing settings for the theme.

Using Layouts

The Layouts feature within the Relationship Analysis Client enables you to change the appearance of model elements to show the entities and relationships visually. Certain data renders better in a particular model layout. The Relationship Analysis Client offers five layouts for your models. There are also options respective to each layout and options that are common to all layouts. Click the links below to see more information about each layout and the type of data that is most appropriate for that layout.

- [Common Options](#)
- [Circular Layout](#)
- [Hierarchic Layout](#)
- [Organic Layout](#)
- [Orthogonal Layout](#)
- [Map Layout](#)

Common Options

These options are common to all layouts.

- **Default**—Routes relationships using the shortest straight lines.
- **Organic**—Routes relationships organically to ensure that they do not overlap entities and that they keep a specifiable minimal distance to the entities. This algorithm will work correctly only if there is enough room between each pair of entities in the model.
- **Orthogonal**—Routes a diagram's relationships using vertical and horizontal line segments only. The positions of the diagram's entities will remain fixed. Usually, the routed relationships will not cut through any entities or overlap any other relationships. Some relationships may be re-drawn after the user moves entities or, subsequently, added relationships may be drawn to fit the existing diagram.
- **Consider entity labels**—Guarantees that entity labels will not overlap entities in the model.
- **Use existing layout as a sketch**—Interprets the initial model layout as a sketch for the desired outcome of the layout process. The algorithm tries to identify the entities that should lie on an entity partition's circle boundary and maintain their cyclic order around the center of the entity partition.
- **Selected elements only**—Applies settings only to selected elements and not all elements.
- **Multi-page layout**— Displays the contents of a model in logical clusters across multiple pages, making them easier to view.

Click the **Multi-page layout** button to activate this feature and let the Relationship Analysis Client distribute the contents accordingly. Alternatively, if you want the results to appear within a specific number of pages, enter the minimum and/or maximum number of pages onto which the contents should be distributed and then click the **Multi-page layout** button. The smaller the value in the **Max** field, the more quickly the results will be returned.

Note: The Relationship Analysis Client may return more pages than you indicate as the maximum if the model contains a higher number of unconnected segments. For example, if you set the maximum number of pages to 5 but 7 unconnected segments are identified when you click the button and the algorithm is run, 7 pages will be returned. Use the arrow keys to scroll through the pages.

Circular Layout

A circular layout emphasizes group and tree structures within a network. It analyzes the connectivity structure of the network and arranges the partitions as separate circles.

A circular layout is useful for models that depict social networks, network management, and e-commerce.

Layout Style

- **Compact** (default)—Each partition will represent a so-called biconnected component of the model. This component consists of entities that are reachable by two relationship-disjoint paths. Entities that belong to more than one biconnected component will be assigned exclusively to one partition.

- **Isolated**—Entity partitions will be formed similarly to those in Bcc Compact with the exception that all entities belonging to more than one biconnected component will be assigned an isolated partition.
- **Groups**—Entity partitions that form circles will be provided by the user. To specify the partitions, a data provider holding such supplemental layout data must be bound to the model.
- **Cycle**—All entities will be arranged on a single circle.

Partition Style

- **Cycle**—All entities from a given partition are placed on a circle.
- **Disk**—Entities from a given partition that connect to entities from other partitions are placed on the disk's boundary while all other entities are placed on the inside.
- **Organic**—Used to arrange the entities from a partition. Entities that connect to entities from other partitions can be placed both on the inside or on the boundary of the disk.

Layout Options

The Circular layout has general options, partition options, and options related to how the model tree is arranged. Access the following settings by clicking the arrow in the bottom-right corner of the Circular Layout Options ribbon. After changing settings, click Apply to immediately view the changes on the canvas in the background or move on to another tab. Click **OK** to return to the canvas. If you make changes and later wish to restore default settings, click **Revert**.

General Options

See [Layout Style](#) for more information on the drop-down options.

- **Place children on common radius**—Enables tighter drawings by using adapted radii for child entities. By default, the entities in tree-like structures connecting to a circle are always placed on a common radius according to their distance from the root entity. Tighter drawings can be achieved by abandoning the common radius and instead placing these child entities using radii adapted to the size of an actual child entity and the sizes of entities in its vicinity.

Partition Options

- See [Partition Style](#) for more information on the drop-down options.
- **Automatically set radius/Minimum entity distance**—Determines the radius of each circle/disk in the layout automatically.
- **Manually set radius/Radius**—Determines the fixed radius for all circles/disks in the resulting layout.

Tree Arrangement Options

- **Preferred child wedge**—Determines the angular range of the sector that will be reserved for the children of a root entity. The possible angular range lies between 1 and 359. The smaller the chosen value, the greater the impression that the entities drive away from their root entities and the center of the model. Generally speaking, the compactness of the layout will decrease with smaller values. Very small values will lead to layouts that consume a lot of space.
- **Minimum relationship length**—Determines the minimal length of an relationship that connects two entities that lie on separate circles. The smaller the chosen value, the more compact the resulting layout.

- **Maximum deviation angle**—Determines the allowed angular deviation from the optimal relationship direction. The bigger the chosen value, the more compact the resulting layout. If a value smaller than 90 degrees is chosen, then the tree relationships might cross through the circular arranged components.
- **Compactness**—Influences the length of the tree relationships as it is computed by the layout. The smaller the compactness factor, the shorter the tree-relationships and the more compact the overall layout. The bigger the compactness factor the more difficult, and hence slower, the layout computation.
- **Minimum entity distance**—Determines the minimal distance between the borders of two adjacent entities on the circle/the disk's boundary. The smaller the distance, the more compact the resulting layout.
- **Allow overlaps**—Further increases compactness of the resulting layout, but potentially introduces slight entity overlaps.

Radial Layout

A radial layout places entities on circles around a common center (also known as the root), expanding outward, similar to rings in a tree. The center entity is the first entity, the first ring contains the "children" entities of the first entity, the second ring contains the "children" entities of the entities from the first ring, and so on. As such, "children" entities are on higher layers, or larger rings, than their "parent" entities. Center entity policies determine which entities are placed at the center or on the innermost ring, and layering strategies determine how the entities are distributed over the rings.

A radial layout is useful for models that depict social networks, web analysis, or organizations with a central authority and multiple levels of management.

Center Entities Policy

- **Weighted Centrality** (default)—The entity with the greatest number of all paths interconnecting all entities with each other will be the center entity.
- **Centrality**—The entity with the highest centrality will be the center entity.
- **Directed**—All entities with zero incoming relationships will be the center entities.
- **Selected Entities**—User-specified entities will be the center entities.

Layering Strategy

- **BFS** (default)—A breadth-first search (BFS) strategy where all relationships will span no more than one layer in the model. Relationships between entities that belong to the same layer are possible.
- **Hierarchical**—Assigns layers in such a way that there are no overlaps and the overall sum of the layer distances of all relationships in the layout is minimal.

Layout Options

The Radial layout has general options related to how the model tree is arranged. Access the following settings by clicking the arrow in the bottom-right corner of the Radial Layout Options ribbon. After changing settings, click **Apply** to immediately view the changes on the canvas in the background or move on to another tab. Click **OK** to return to the canvas. If you make changes and later wish to restore default settings, click **Revert**.

- **Center Entities Policy**—Determines which entities are placed at the center or on the innermost ring:
 - **Weighted Centrality** (default)—The entity with the highest weighted centrality will be the center entity.
 - **Centrality**—The entity with the highest centrality will be the center entity.
 - **Directed**—All entities with zero incoming relationships will be center entities.
 - **Selected Entities**—A user-specified entity will be the center entity.
- **Layering strategy**—Determines how the entities are distributed over the rings:
 - **BFS** (default)—A breadth-first search (BFS) strategy where all relationships will span no more than one layer in the model. Relationships between entities that belong to the same layer are possible.
 - **Hierarchical**—Assigns layers in such a way that there are no overlaps and the overall sum of the layer distances of all relationships in the layout is minimal.
- **Layer spacing**—Determines the spacing value for the circles. The radius of each circle will be a multiple of this value. Default is 25.
- **Minimal layer distance**—The minimal distance between two circles. Default is 100.
- **Minimum entity to entity distance**—The minimum distance between two neighboring entities on a circle. Default is 30.
- **Minimal bend angle**— The minimal kink angle in degrees between two adjacent relationship segments. Increasing this value reduces the number of bends so the relationship path gets less smooth. Valid values are between 0.0 and 90.0. The default is 5.
- **Maximal child sector angle**— The maximal sector angle in degrees around an entity where the children of the entity may be placed. The child sector of an entity has the entity's center as center. The sector points away from the center of the circle the entity is placed on. Default is 180.

Hierarchic Layout

A hierarchic layout highlights the main direction or flow within a directed model. The entities of a model are placed in hierarchically arranged layers such that most of the relationships of the model show the same overall orientation (for example, top-to-bottom). Additionally, the ordering of the entities within each layer is chosen in such a way that the number of relationship crossings is small.

A hierarchic layout is useful for models that depict workflows, database modeling, and process modeling.

Orientation

- **Top to Bottom** (default)—Layout will be oriented from top to bottom.
- **Bottom to Top**—Layout will be from bottom to top, which means that the layout will be flipped along the x axis.
- **Left to Right**—Layout will be from left to right, which means that the layout will be rotated by 90 degrees counterclockwise.
- **Right to Left**—Layout will be from right to left, which means that the layout will be rotated by 90 degrees clockwise.

Style

- **Orthogonal** (default)—Vertical and horizontal line 90-degree segments only.
- **Polyline**—A continuous line composed of one or more 45-degree line segments.

Layout Options

The Hierarchic layout has general options, relationship options, and layer options. Access the following settings by clicking the arrow in the bottom-right corner of the Hierarchic Layout Options ribbon. After changing settings, click **Apply** to immediately view the changes on the canvas in the background or move on to another tab. Click **OK** to return to the canvas. If you make changes and later wish to restore default settings, click **Revert**.

General Options

See [Orientation](#) for more information on the drop-down options.

- **Minimum distance**—The minimum distance between two entities in the same layer, an entity and a non-adjacent relationship in the same layer, two relationships in the same layer, or two layers.
- **Layout components separately**—Place unconnected sub-models vertically, or stacked. Otherwise they are placed horizontally.
- **Symmetric placement**—Return a more symmetric model, although it is likely to have more bends.

Relationship Options

- **Routing style**—Choose one of the following options:

Orthogonal (default)—Vertical and horizontal line segments only.

Polyline—A continuous line composed of one or more line segments.

- **Automatic relationship grouping**—Groups as many relationships as possible without changing the semantic of the model. Edges are grouped either at a common source entity or at a common target entity.
- **Backloop routing**—Forces relationships that connect to targets located above their source entities to exit at the bottom and enter at the top of their source and target, respectively, emphasizing the main direction of the diagram.
- **Port constraint optimization**—The Relationship Analysis Client tries to find a better source port constraint if the current constraint is null. A port constraint expresses at what position a relationship is allowed to connect to either its source or target entity. A weak port constraint limits the position of the port to a particular side of an entity. A strong port constraint fixes the position of the port completely to the position of the current port coordinates.
- **Minimum** —The minimum length of the first segment, the last segment, the relationship itself, or the distance between one relationship to another.
- **Slope** (for polyline routing style only)—Determines the minimum slope of the relationship's mid segments in a vertical layout.
- **Labeling**—Choose one of the following styles:
 - None** (default)—No labels will appear on selected relationships.

Generic—Finds label positions for the labels from a given model so that they, ideally, do not overlap with each other or with model elements. It does so without altering entities or relationships in any way.

Hierarchic (default)—Finds optimal placements for relationship labels such that there are no overlaps of labels with each other or with model elements.

- **Label model**(inactive when "None" is selected for Labeling)—choose one of the following styles:

As Is (default)—Retains labels as they are; they will not be altered.

Center—Places labels in the center of the relationships.

Side (default)—Places labels on the side of the relationships.

Free—Allows for unspecified label positions.

Layer Options

- **Assignment Policy**—Assigns the entities of a model to separate layers. Choose one of the following styles:

Optimal (default)—Assigns layers in such a way that the overall sum of the layer distances of all relationships in the layout is minimal.

Tight Tree Heuristic—Approximates the layering done by Hierarchical-Optimal.

BFS Layering (breadth-first search) (default)—Results in all relationships spanning at most one layer in the resulting drawing. Edges between entities that belong to the same layer are possible.

From Sketch—Tries to find a layering that is similar to the one in the input model. When this layering strategy is used, the layout algorithm may place entities in the same layer, even though they are connected by a relationship. These inner layer relationships are always routed in an orthogonal style.

Topmost—Assigns all entities without incoming relationships (in degree zero) to the topmost layer of the layout. The number of separate layers will be as small as possible.

- **Scale**(From Sketch assignment policies only)—Sets the size of the halo around an entity or the respective insets that are used to calculate layer overlaps.
- **Halo**(From Sketch assignment policies only)—Sets the scale of an entity's height.
- **Minimum size**—(From Sketch assignment policies only) Allows a group entity to accommodate the size of its label, provided it meets minimum size standards.
- **Maximum size**—(From Sketch assignment policies only) Allows a group entity to accommodate the size of its label, provided it meets maximum size standards.
- **Alignment within layer**—Determines how to align the entities within the layer. Choose one of the following styles:

Top of Entities —Aligns the layer's entities along the top.

Center of Entities —Aligns the layer's entities along the center.

Bottom of Entities (default)—Aligns the layer's entities along the bottom.

Organic Layout

An organic layout is based on the force-directed layout paradigm. When calculating a layout, the entities are considered to be physical objects with mutually repulsive forces, such as protons or electrons. The connections between entities also follow the physical analogy and are considered to be springs attached to the pair of entities. These springs produce repulsive or attractive forces between their end points if they are too short or too long. The layout algorithm simulates these physical forces and rearranges the positions of the entities in such a way that the sum of the forces emitted by the entities and the relationships reaches a (local) minimum. Resulting layouts often expose the inherent symmetric and clustered structure of a model; they show a well-balanced distribution of entities and have few relationship crossings.

An organic layout is useful for models that use enterprise networking, system management, and WWW visualization.

Output

Output determines whether the layout is fit into a particular shape or to a specified aspect ratio.

- **No Restriction**—Layout is not restricted to a particular output shape or aspect ratio.
- **Aspect Ratio**—Layout is restricted to an aspect ratio.
- **Elliptical**—Layout will be output into an elliptical shape.
- **Rectangular**—Layout will be output into a rectangular shape.

Layout Options

The Organic layout has general options and restriction options. Access the following settings by clicking the arrow in the bottom-right corner of the Organic Layout Options ribbon. After changing settings, click Apply to immediately view the changes on the canvas in the background or move on to another tab. Click **OK** to return to the canvas. If you make changes and later wish to restore default settings, click **Revert**.

General Options

- **Layout is deterministic**—Produces identical results for identical input models and identical settings.
- **Allow entity\relationship overlaps**—Guarantees the absence of entity and relationship overlaps even for very dense models.
- **Allow entity overlaps**—Guarantees the absence of entity overlaps even for very dense models.
- **Minimum entity distance**—Prevents entity (label) overlaps and keeps minimum distances between the entities.
- **Preferred relationship length**—Specifies the general preferred length of all relationships.
- **Compactness**—Produces significantly more or less compact layouts. Small values for property "Compactness" result in large area drawings with a relatively sparse entity distribution. Values greater than 0.5 result in artificially compacted layouts, and values next to 1.0 result in layouts with near to minimal space allocations. Obeyed only for medium or high quality settings..
- **Quality/Time ratio**—Uses small values to produce acceptable layouts for huge models (thousands of entities within seconds) and greater values to produce high-quality layouts, which may take longer (up to a few minutes for hundreds of entities).

Restriction Options

- **Output area**—Determines whether the layout is fit into a particular shape or to a specified aspect ratio. Choose one of the following styles:

No Restriction—Layout is not restricted to a particular output shape or aspect ratio.

Aspect Ratio—Layout is restricted to an aspect ratio.

Elliptical—Layout will be output into an elliptical shape.

Rectangular—Layout will be output into a rectangular shape.

- **Bounds**—(Only for Elliptical or Rectangular output area) Allows you to set the X and Y bounds as well as the width and height of the ellipsis or rectangle.
- **Aspect ratio**—(Only for Aspect Ratio output area) Allows you to specify the aspect ratio of your output.

Orthogonal Layout

An orthogonal layout is a multipurpose layout provider for undirected models. It produces compact drawings with no overlaps, few crossings, and few bends.

An orthogonal layout is useful for models that use database schema, system management, and software engineering.

The orthogonal layout has the options shown below.

- **Layout style**—choose one of the following styles:

Normal (default)—Node sizes will not be changed by this layout. The drawing will contain very few bends.

Box—Nodes are resized according to the number and position of their neighbors to reduce the overall number of bends.

Fixed Box—Similar to Box but retains original entity sizes.

Fixed Mixed—Similar to Mixed but retains original entity sizes.

Mixed—Resembles Box but resizes all entities to equal size. Introduces additional bends and routes the last relationship segment of these relationships non-orthogonally to their adjacent entities.

Normal Tree—Similar to Normal but larger subtrees are processed using a specialized tree layout algorithm, which is better suited for tree-like structures than the original orthogonal layout style.

Uniform—All entity sizes will be changed to equal size before the model is processed.

- **Reduce relationship crossings**—Reduces the overall number of relationship crossings. The cost for the reduction is increased execution time of the layout.
- **Length reduction**—Reduces the overall relationship lengths. The cost for the reduction is an increased execution time of the layout algorithm.

- **Consider entity labels**—Guarantees that entity labels will not overlap entities in the model.

Access the following settings by clicking the arrow in the bottom-right corner of the Organic Layout Options area of the ribbon. After changing settings, click **Apply** to apply the changes and move on to another tab. Or, click **OK** to return to the canvas. If you make changes you wish to undo and have not yet clicked Apply or OK, click **Revert**.

General Tab

- **Layout style**—See description above.
- **Grid spacing**—Defines the virtual grid spacing used by the layout algorithm. Each entity will be placed in such a way that its center point lies on a grid point. Edges will be routed in such a way that their segments lie on grid lines, if the terminal entities of the relationships allow to place the ports accordingly. Note that this option is only guaranteed to be obeyed for Normal layout style.
- **Reduce relationship crossings**—See description above.
- **Length reduction**—See description above.
- **Minimize bends**—Avoids helical arrangement of chains of entities.
- **Use randomization**—Increases overall layout quality with high probability. The cost for this layout optimization is increased execution time and non-deterministic results for subsequent layout invocations.
- **Use face maximization**—
- **Use existing layout as a sketch**—Interprets the initial model layout as a sketch of the resulting orthogonal layout. The layout algorithm tries to *orthogonalize* the given sketch without making too much modifications in respect to the original drawing
- **Consider entity labels**—See description above.
- **Edge labeling**—choose one of the following styles: None (default)—No labels will appear on selected relationships. Generic—Finds label positions for the labels from a given model so that they, ideally, do not overlap with each other or with model elements. It does so without altering entities or relationships in any way. Integrated (default)—used in conjunction with layout style "Normal." Taken into consideration when determining both entity placement and relationship path generation. With this strategy it is guaranteed that no relationship label will overlap other objects in the diagram.
- **Edge label mode**(inactive when "None" is selected for Labeling)—choose one of the following styles:
 - As Is** (default)—Retains labels as they are; they will not be altered.
 - Center**—Places labels in the center of the relationships.
 - Side** (default)—Places labels on the side of the relationships.
 - Free**—Allows for unspecified label positions.

Style

- **Box**—Nodes are resized according to the number and position of their neighbors to reduce the overall number of bends.
- **Fixed Box**—Similar to Box but retains original entity sizes.

- **Mixed**—Resembles Box but resizes all entities to equal size. Introduces additional bends and routes the last relationship segment of these relationships non-orthogonally to their adjacent entities.
- **Fixed Mixed**—Similar to Mixed but retains original entity sizes.
- **Normal** (default)—Node sizes will not be changed by this layout. The drawing will contain very few bends.
- **Normal Tree**—Similar to Normal but larger subtrees are processed using a specialized tree layout algorithm, which is better suited for tree-like structures than the original orthogonal layout style.
- **Uniform**—All entity sizes will be changed to equal size before the model is processed.

Layout Options

The Orthogonal layout has general options. Access the following settings by clicking the arrow in the bottom-right corner of the Orthogonal Layout Options ribbon. After changing settings, click **Apply** to immediately view the changes on the canvas in the background or move on to another tab. Click **OK** to return to the canvas. If you make changes and later wish to restore default settings, click **Revert**.

- **Layout style**—See description above.
- **Grid spacing**—Defines the virtual grid spacing used by the layout algorithm. Each entity will be placed in such a way that its center point lies on a grid point. Edges will be routed in such a way that their segments lie on grid lines, if the terminal entities of the relationships allow to place the ports accordingly. Note that this option is only guaranteed to be obeyed for Normal layout style.
- **Reduce relationship crossings**—Reduces the overall number of relationship crossings. The cost for the reduction is increased execution time of the layout.
- **Length reduction**—Reduces the overall relationship lengths. The cost for the reduction is an increased execution time of the layout algorithm.
- **Minimize bends**—Avoids helical arrangement of chains of entities.
- **Use randomization**—Increases overall layout quality with high probability. The cost for this layout optimization is increased execution time and non-deterministic results for subsequent layout invocations.
- **Use face maximization**—
- **Relationship labeling**—Choose one of the following styles:

None (default)—No labels will appear on selected relationships.

Generic—Finds label positions for the labels from a given model so that they, ideally, do not overlap with each other or with model elements. It does so without altering entities or relationships in any way.

Integrated (default)—used in conjunction with layout style "Normal." Taken into consideration when determining both entity placement and relationship path generation. With this strategy it is guaranteed that no relationship label will overlap other objects in the diagram.

- **Relationship label model** (inactive when "None" is selected for Labeling)—choose one of the following styles:

As Is (default)—Retains labels as they are; they will not be altered.

Center—Places labels in the center of the relationships.

Side (default)—Places labels on the side of the relationships.

Free—Allows for unspecified label positions.

Analyzing Model Data

The Data section of Relationship Analysis Client provides tools help you analyze the data in your model.

- [Using Discovery](#)
- [Using Bipartite Settings](#)
- [Identifying Cycles](#)
- [Using Groups](#)
- [Editing Models](#)

Using Discovery

The Discovery feature enables you to load the smallest number of elements based on conditions specified in a query. It also enables you to load additional elements that are most important for the analysis task. For example, you could be using a model that contains terrorism data, and you might only want to see activity that is tied to a particular suspect, or entity. After you view this data, you may decide to look at other entities this suspect is connected to. You could use Discovery to load all elements connected to this suspect, or you could load only the elements who succeed, or follow, this suspect.

After you have reloaded your model to include additional elements, you may want to remove some of those elements. Using the previous example, perhaps the suspect has young sons or daughters as connections. Because of their age, you do not consider them suspects. You can use the Unload function to return entities with a relationship of "Son" or "Daughter".

Using Discovery Options, you can determine whether or not to restore additional relationships after performing a Discovery function. You can also set the maximum number of entities to return using Discovery.

Returning Elements

If you open a partial model using the **Query** function on the Data tab, the **Discovery** feature enables you to query the model to return entities that are somehow connected to the existing, selected entity. You can return the following types of entities to your partial model:

- All entities that are connected to the selected entity
- Entities that precede the selected entity
- Entities that follow the selected entity

You can further filter these results by selecting a **Relationship Label** and then clicking **Connected**, **Predecessors**, or **Successors**. For example, if you are viewing a partial model of terrorism suspects, you could select one or more entities, then select a TraveledWith relationship label, and then click Connected. The model would reload with the original entities along with all entities who were connected to and traveled with the original entities.

To use Discovery:

1. Open a partial model using the **Query** function on the **Open** tab.
2. Select the entity whose immediate connections you want to identify.

Note: If you select multiple entities, Discovery will return results only for the last one selected.

3. Select a **Relationship Label** if you want to filter the connections.
4. Click **Connected**, **Predecessors**, or **Successors**. The model will reload, showing any additional entities that were connected to the original entities in the manner you selected. If the graph becomes fully loaded with all of its entities, Discovery will become disabled.

Note: The availability of these options depends on your model's contents. Sometimes Predecessors or Successors will be grayed out.

Unloading Elements

After you use Discovery to bring additional entities and relationships into the model, you can unload some of the data that is brought in by using the Unload feature. Using the same example above, after you brought in entities who traveled with the existing entities, you could then unload the entities who have a relationship label of "wife" because you might be looking only at male suspects.

1. Click the entities you want to remove from your model, or click the entities whose connected entities you want to remove. For example, you might want to remove all connections to a terrorism suspect if the only connections are family members who are not considered suspects.
2. Select a **Relationship Label** if you are unloading entities connected to selected entities and want to filter which ones are removed based on the relationships between them. Using our example, you might select "Wife" or "Child."

Note: You can only unload entities that have no connections other than to the entity you select in Step 1.

3. Click **Unload**.
4. Choose one of the following options.

Selected entities	Removes all entities currently selected.
-------------------	--

Unselected entities	Removes all entities not currently selected.
---------------------	---

Connected leaves	Removes all entities and relationships connected to the currently selected entities.
Predecessor leaves	Removes all entities and relationships that come into, or precede, the currently selected entities.
Successor leaves	Removes all entities and relationships that go out of, or succeed, the currently selected entities.

Note: The availability of these options depends on your model's contents. Sometimes you can choose only Selected entities or Unselected entities.

Options

By checking **Fill in relationships** in the Discovery Options box, you can return additional relationships revealed in Discovery. If the discovered entities were connected to additional entities already on the canvas (other than any you selected prior to using Discovery), those relationships will also be returned.

You can also determine the maximum number of entities or relationships to return. The default is 100.

Using Centrality

Centrality is a way of measuring the importance and significance of individual entities and relationships. When you run centrality algorithms, the higher the value, the more important the element.

- From the Centrality ribbon on the Data tab, select the kind of centrality measure you want to apply to your model.
 - **Betweenness**—Reflects the number of shortest paths between one entity and other entities.
 - **Closeness**—Reflects the length of geodesic distances between one entity and other entities.
 - **Degree**—Reflects the number of relationships on a entity.
 - **Influence**—Reflects the importance of an entity, based on its connections to high-scoring entities.
- Select the direction in which you want to apply the algorithm:
 - **Incoming**—The results will be based on relationships coming into the entity.
 - **Outgoing**—The results will be based on relationships going out of the entity.
 - **Both**—The results will be based on relationships both coming into and going out of the entity.
- If you are using a Closeness algorithm, click the appropriate button for the way in which you want results to be returned:
 - **Standard**—Results are based on the number of attachments, or relationships, an entity has as well as the reverse of the sum of shortest paths to each entity.

- **Dangalchev**—Results are based not only on the number of entities linked to another entity but also the number of relationships in each of the linked entities.
 - **Opsahl**—Results are based on the sum of reversed shortest paths to each entity.
4. If you are using an Influence algorithm, slide the **Precision** scale to determine how precise the results should be. A lower precision will return more accurate results, but the algorithm will run more slowly.
 5. Click the **Use relationship property as weight** if you want to measure how unfavorable a relationship is, and select the relationship property you want to use from the **Property** drop-down. In this case, a higher value indicates a negative association.
 6. Click the **Low values are more significant** box if you are using a relationship property as weight and that property is one where a lower value is considered better than a higher value. For example, if the property is some sort of ranking system, typically 1, or 1st, is the best value. Another example is if the property is distance, and you are trying to determine the shortest route: 5 miles is considered better than 10 miles.
 7. Click the **Override default output property name** if you want the output property name to be something other than the algorithm you selected. Then enter the new name in the **Property** field.
 8. Click **OK**.

Using Groups

The Groups feature lets you select biconnected elements, which are in a model that is connected and inseparable, such that if an arbitrary entity were removed, the model would remain connected. This shows you that there are at least two different paths to reach one entity from another entity.

You can determine whether your model is biconnected by clicking the **Model Details** button on the Tools ribbon and seeing if the "Is Biconnected" field is set to "True".

This feature also lets you set relationship betweenness clustering options. Click the **Betweenness clustering** button on the Groups ribbon of the Data tab. On this dialog box you can set minimum and maximum group count and designate which relationship property should be used as the weight. When you click **OK**, you will see the elements in your model that meet the requirements you set appear in a different color. Highlighting these elements can identify latent relationships in your data.

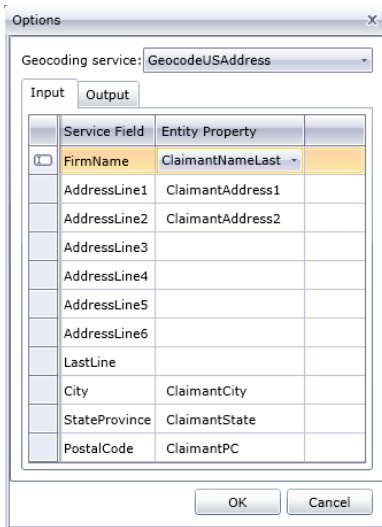
Using Geocoding

Perform this procedure to define geocoding options.

The Geocoding feature enables you to manually geocode the entities in your model by using a service that performs geocoding. If you have already defined geocoding options, simply click the top half of the **Geocode** button from the **Geocoding** ribbon on the **Data** tab to perform geocoding.

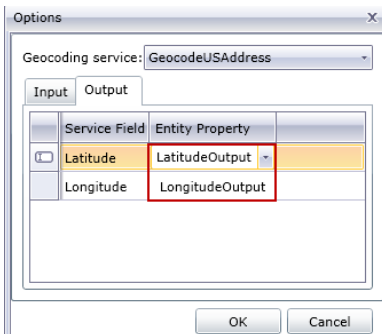
1. In the **Geocoding service** box, select a service that performs geocoding.

If the service you select contains latitude and longitude fields, the Geocoding feature will automatically detect those fields and populate the **Entity Property** column. If the service performs geocoding but the latitude and longitude data is in other fields, you will need to map those fields in the **Entity Property** column:



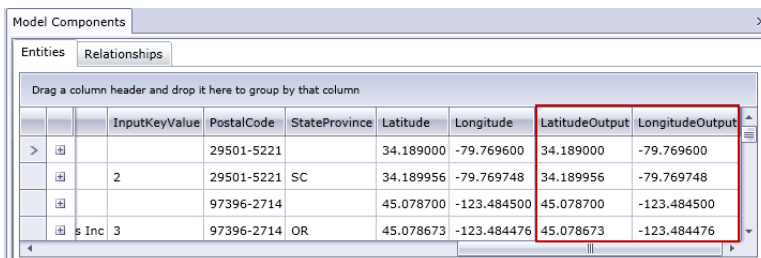
If you select a service that does not perform geocoding, an error message displays *Service does not appear to geocode.*

2. To change the name of the latitude and longitude fields when geocoding is performed, click the **Output** tab and change the names in the **Entity Property** column.



3. Click **OK**.

The Relationship Analysis Client will perform geocoding and add latitude and longitude data to your model.



Editing Models

This feature enables you to manually add or delete entities or relationships from models. It also allows you to manually add, edit, or remove properties and modify values for properties associated with entities and relationships. An example of this functionality might be that if your model uses data for the United States of America and one of the entity properties for a state is population, you could change its value if for some reason the population has changed since your model was created.

After you edit your model you can simply click **Save** to apply the changes to the existing model, or you can click **Save As** to save the edited model under a new name.

Adding Entities

This feature allows you to manually add entities to an existing model. This can be useful if you don't want to recreate the model but have data in the form of an entity that you would like to add to a model.

1. With an existing model open, click to open **Entities** in the Toolbox.
2. Drag the appropriate icon onto the model canvas. Choose from the existing types of entities or choose **<New>** if the entity is a type that is new to your model.
3. Enter a **Label** for the new entity.
4. If you are adding a new type of entity, enter the **Type**. You can pick from an existing type or enter a new type.

Note: If you added an existing type of entity, this field will automatically populate and cannot be changed.

5. Add **Properties** to the entity by completing the **Name**, **Type**, and **Value** fields.

Note: The Type field will automatically populate when you enter a name, but you can select another type from the list if necessary.

6. Click **Add**, then **OK**. The new entity appears.

If you save your model after adding the entity, it will be included in the counts shown when you open or manage the model.

Adding Relationships

This feature allows you to manually add relationships to an existing model. This can be useful if you have identified new relationships between existing entities or if you have added new entities to your model that you now need to tie to other entities.

1. With an existing model open, click **Relationships** in the Toolbox.
2. Click the appropriate icon. Choose from the existing types of relationships or choose **<New>** if the relationship is a type that is new to your model.
3. On the canvas, click and drag from the source entity to the target entity and release the mouse. If you chose an existing type of relationship in Step 2, the relationship is now established. If you chose a new relationship in Step 2, the **Add Relationship** dialog box appears.
4. Enter a **Label** for the new relationship.

- If the two entities have multiple relationships with the same label, you must check **Use unique identifier value** and enter a value that is specific to this relationship. Two entities cannot have multiple relationships with the same label without a unique identifier.
- Add **Properties** to the relationship by completing the **Name**, **Type**, and **Value** fields.

Note: The Type field will automatically populate when you enter a name, but you can select another type from the list if necessary.

- Click **Add**, then **OK**. The relationship appears.

If you save your model after adding the relationship, it will be included in the counts shown when you open or manage the model.

Moving Relationships

This feature allows you to manually move relationships between entities. This can be useful if you need to change the source or target entity for a given relationship.

Note: While both the source and the target for the relationship can be moved, you cannot move them simultaneously; you must move one end (for example, source entity) first and then the other end (for example, target entity) to move the entire relationship.

- With an existing model open, click the relationship you would like to move. Notice the black square that appears in the center of both the source and target entity tied to that relationship.
- Click the black square that is in the entity you are removing from the relationship, drag it to the new entity, and release the mouse. When you drag the relationship to the new entity, it will become highlighted if the move is valid. If the entity does not become highlighted, you cannot move the relationship to that entity.

Deleting Entities and Relationships

This feature enables you to manually delete entities and relationships in an existing model. This can be useful if you don't want to recreate the model but have entities or relationships you would like to remove from your model.

- With an existing model open, click the **Edit model** button on the **Editing** ribbon.
- Click the entity or relationship you want to remove and press **Delete**. The model updates accordingly.
- Save the model. Run the Centrality algorithm to see differences in the model after these changes.

Note: When you delete an entity, the relationships attached to it are deleted as well. However, when you delete a relationship, the entities it connects are **not** deleted.

Splitting Entities

This feature enables you to split one entity into two entities. If your model contains entities that perform multiple functions within that model, you could create one entity for one function, and a second entity for the other function. Using the organization chart example from [Write to Hub](#), you might want to create two entities out of the entity for Jim Waterman, who has employee ID 3. Jim

is a manager to Marcus David (employee ID 12), but he is also an employee of Tom Smith (employee ID 1). Therefore, in that model, you might want a manager entity and an employee entity.

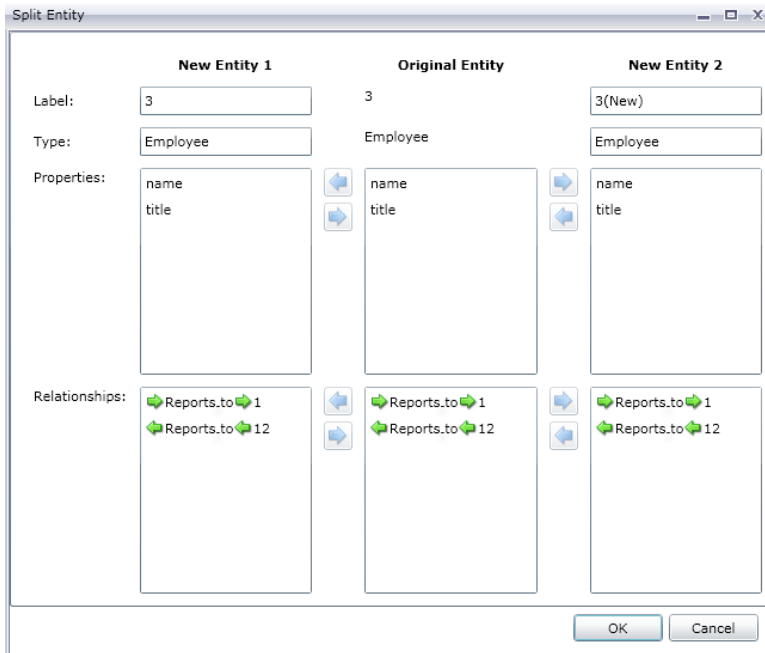
In addition to splitting one entity into two, you can also determine which properties and relationships are tied to each specific entity—all, some, or none. Or, you can tie properties and relationships to both entities; they don't have to be tied to one or the other.

1. With an existing model open, click the entity you want to split.
2. Click the **Split** button on the **Editing** ribbon or right-click and select **Split entity**. The **Split Entity** dialog box will appear, with the Original Entity in the middle, New Entity 1 on the left, and New Entity 2 on the right. By default, all properties and relationships will appear for all three entities.
3. If you want to change the labels for the new entities, enter new names in the **Label** field. In our example, we will keep the label as "3" because using the employee ID is consistent with the rest of the entities in the model.
4. If you want to change the types of the new entities, use the **Type** field. In our example, we will keep the type for New Entity 1 as "Employee," but we will change New Entity 2 to "Manager."

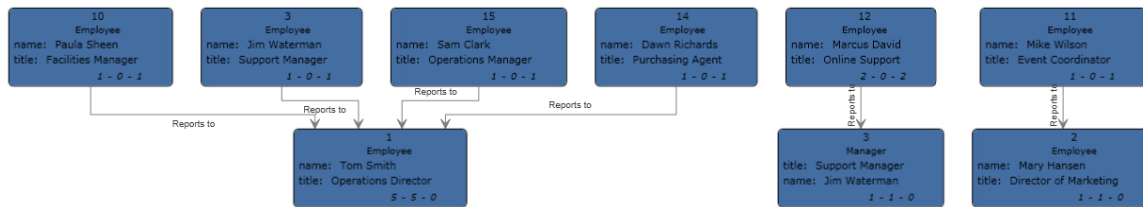
Note: Each entity must have a unique combination of label and type. In our example, we can have two entities with the label of "3" since the types are different, but we could not have two entities with the label of "3" and a type of "Employee."

5. Determine which properties you want tied to each entity by clicking a property and then using the blue arrows in between the entities in the **Properties** field. Our example will keep the "name" and "title" properties for both new entities because this information is useful as an employee and as a manager.
6. Determine which relationships you want tied to each entity by clicking a relationship and then using the blue arrows in between the entities in the **Relationships** field. In our example, we will keep the outgoing "Reports_To" relationship for New Entity 1 because that represents the relationship between Jim as an employee and his manager Tom. We will keep the incoming "Reports_To" relationship for New Entity 2 because that represents the relationship between Jim as a manager and his employee Marcus.

The completed Split Entity dialog box for our example looks like this:



The model for our example has been updated with the two new entities. You can see that New Entity 1 is shown as an employee to Tom Smith, and New Entity 2 is shown as a manager to Marcus David.



Merging Entities

This feature enables you to merge two entities into one. If your model contains two entities that represent the same subject, you can merge them and their respective data together into one entity. Using the end result from **Splitting Entities**, you might want to merge together the entities for Jim Waterman, who has two entities in that model: one as a manager to Marcus David, and one as an employee of Tom Smith.

In addition to merging two entities into one, you can also determine which properties and relationships will be associated with the new, single entity—all, some, or none.

1. With an existing model open, select two entities you want to merge. You can do this by clicking and dragging over the two entities, or you can click one entity and then press Ctrl and click the second entity.
2. Click the **Merge** button on the **Editing** ribbon or right-click and select **Merge entities**. The **Merge Entities** dialog box will appear, with Original Entity 1 in on the left, Original Entity 2 on the right, and New Entity in the middle.

3. If you want to change the label for the new entity, enter a new name in the **Label** field. In our example, we will keep the label as "3" because using the employee ID is consistent with the rest of the entities in the model.
4. If you want to change the type of the new entity, use the **Type** field. In our example, we will keep the type as "Employee."
5. Determine which properties you want tied to the new entity by clicking a property and then using the blue arrows in between the entities in the **Properties** field. Our example will keep the "name" and "title" properties for the new entity because this is consistent with the rest of the entities in the model. We can pull the data for those properties from either of the two entities because the data is the same ("Jim Waterman" and "Support Manager," respectively). Note that as you add properties to the new entity, they are prefaced with a [1] or [2], indicating the original entity that property came from.

Each property must have a unique name. Therefore, if you want to merge two properties with the same name but different data, you would need to modify the name of the property for one of the entities by clicking a property, clicking the green arrow to open the Advanced Property Mapping dialog box, and either accepting the default new name (which appends a number to the end of the property name, such as "name2") or entering a new name.

For example, if our organization chart model included start dates for each entity, Original Entity 1 (the employee role) might have a start date of 04/20/2000 and Original Entity 2 (the management role) might have a start date of 08/15/2004. If you wanted to include both start dates in the merged entity, you could click "start_date" for Original Entity 1, click the green arrow, and either accept "start_date2" or enter "original start date" in the **New Name** field.

6. Determine which relationships you want tied to each entity by clicking a relationship and then using the blue arrows in between the entities in the **Relationships** field. In our example, we will keep both "Reports_To" relationships because they represent the relationships for Jim as an employee and as a manager. Note that as you add relationships to the new entity, they are prefaced with "[1]" or "[2]", indicating the original entity that relationship came from.

Unlike properties, each relationship does not need to have a unique name. Two relationships can be merged into one new relationship by clicking the relationship, then clicking the green arrow to open the Advanced Relationship Mapping dialog box, doing nothing with the **Label** and **Unique ID** fields, and pressing **OK**. If you do merge two relationships into one, you will see it prefaced with "[1,2]" to represent that the relationship came from both original entities.

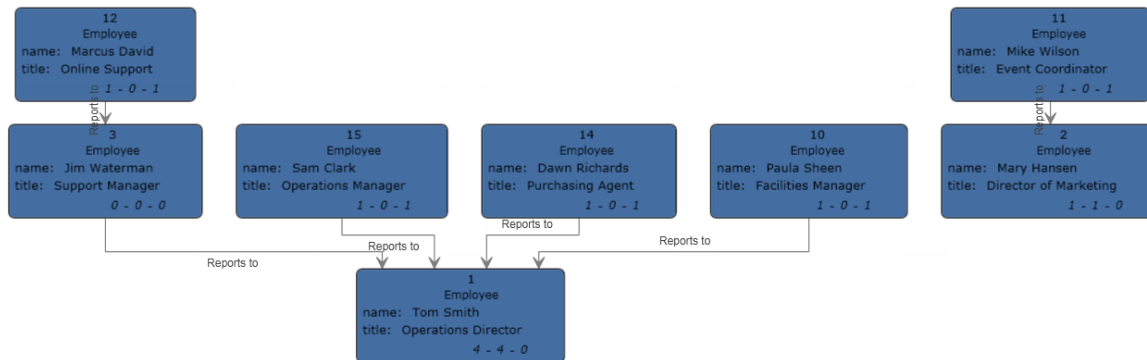
If you want to retain two relationships that have the same name, you would need to modify the name of the relationship for one of the entities by clicking the relationship, clicking the green arrow to open the Advanced Relationship Mapping dialog box, and entering something in the Unique ID field. After that, if the relationship has properties associated with it, you would also need to determine which properties should be carried over from the left and right relationships to the new relationship.

For example, let's say that our organization chart model used titles instead of names and employee numbers, and that one of the employees worked for two people with the title of "Director." To merge those two relationships labeled "Reports_to_Director" but keep them unique within the model, you would need to click one of the relationships, click the green arrow, and

enter a unique identifier of some sort in the **Unique ID** field. In our case, that would probably be the name of the director, and you would repeat this action for the other director so both relationships for the entity include the respective director's name.

The completed **Merge Entities** dialog box for our example looks like this:

The model for our example has been updated with just one entity for Jim Waterman. You can see that the Marcus David entity (Jim's employee) is coming into the Jim Waterman entity, and the Tom Smith entity (Jim's boss) is going out of the Jim Waterman entity.



Adding Properties

With an existing model open and at least one entity or relationship selected, click the **Entities**, **Relationships**, or **All** button on the **Editing** ribbon in the **Data** tab. The **Edit Properties** dialog box appears.

1. Enter the name of the new property in the **Name** field.
2. Select the type of the new property in the **Type** drop-down box. The following data types are supported:

Data Type	Description
Boolean	A logical type with two values: true and false.
Double	A numeric data type that contains both negative and positive double precision numbers between 2^{-1074} and $(2-2^{-52}) \times 2^{1023}$. In E notation, the range of values is 4.9E-324 to 1.7976931348623157E308. For information on E notation, see: http://en.wikipedia.org/wiki/Scientific_notation#E_notation
Float	A numeric data type that contains both negative and positive single precision numbers between 2^{-149} and $(2-2^{-23}) \times 2^{127}$. In E notation, the range of values is 1.4E-45 to 3.4028235E38. For information on E notation, see: http://en.wikipedia.org/wiki/Scientific_notation#E_notation
Integer	A numeric data type that contains both negative and positive whole numbers between -2^{31} (-2,147,483,648) and $2^{31}-1$ (2,147,483,647).
Long	A numeric data type that contains both negative and positive whole numbers between -2^{63} (-9,223,372,036,854,775,808) and $2^{63}-1$ (9,223,372,036,854,775,807)
String	A sequence of characters.

3. Add the value of the new property in the **Value** field.
4. Click **Add**.
5. Click **OK**.

Modifying Properties

1. With an existing model open and at least one entity or relationship selected, click the **Entities**, **Relationships**, or **All** button on the **Editing** ribbon in the **Data** tab. The **Edit Properties** dialog box appears.
2. Select the property whose values you wish to change from the **Name** field.
3. Enter new information in the **Value** field.
4. Click **OK**.
5. Save the model.

Removing Properties

This feature enables you to manually remove properties associated with entities and relationships.

1. With an existing model open and at least one entity or relationship selected, click the **Entities**, **Relationships**, or **All** button on the **Editing** ribbon in the **Data** tab. The **Edit Properties** dialog box appears.
2. Select the property you want to remove from the **Name** field and click **Remove**.

3. Click **OK**.
4. Save the model.

Using the Parameters Feature

The parameters feature makes it possible for you to open the Relationship Analysis Client to a specific model based on the information provided in the URL. By entering a parameter (in this case, the model parameter) in the URL followed by a value (in this case, the name of the model), you can bypass the traditional steps of opening a model. In addition to opening an entire model, you can likewise open a subset of a model by following the model parameter with a query that targets certain data within that model.

This feature is useful in situations where you want to share all or specific parts of a model but don't want the recipient to have to complete the steps it would otherwise take to access the information. You can provide the URL—the address of the server plus these parameters—and the recipient will go directly to the data you want to share.

Opening Models

Opening models using this feature is similar to opening them from within the Relationship Analysis Client, with the exception of manually entering the model name.

1. Be sure the Spectrum™ Technology Platform server is running.
2. Open a web browser.
3. Type the following and press **Enter**:

- `http://<servername>:<port>/hub/`
- `"?Model="`
- The name of the model

For example, if your server is named "myserver", it uses default HTTP port 8080, and you are opening a model called "Fraud" containing insurance claim data, you would enter the following:

```
http://myserver:8080/hub/?Model=Fraud.
```

You can now use all of the features within the Relationship Analysis Client.

Querying Models

You can use the parameters feature to run saved queries or gremlin queries.

Saved Queries

1. Be sure the Spectrum™ Technology Platform server is running.
2. Open a web browser.
3. Type the following and press **Enter**:

- `http://<servername>:<port>/hub/`
- `"?Model="`
- The name of the model
- `"&QueryName="`
- The name of the query

For example, if your server is named "myserver", it uses default HTTP port 8080, you are opening a model called "Fraud" containing insurance claim data for the southwestern United States, and you have a saved query called "California", you would enter the following:

```
http://myserver:8080/hub/?Model=Fraud&QueryName=California
```

Based on how the query was configured, it would return all doctor entities with a ZIP Code that begins with "9". The specific ZIP Code is included in the query as an optional input value based on a field; in this case, the query returns entities whose values in the "ZIP" field start with "9". You can pass this data using parameters in the following format: `&Data.<inputName>=<inputValue>`. For example, if you wanted to limit the query to Hermosa Beach, a suburb of Los Angeles, you would enter the following:

```
http://myserver:8080/hub/?Model=Fraud&QueryName=California&Data.ZIP=90254
```

Note that if you enter a query that includes dates or times, they must be included in the native format:

- Date: YYYY-MM-DD
- Time: HH-MM-SS
- DateTime: YYYY-MM-DDTHH:MM:SS

Gremlin Queries

1. Be sure the Spectrum™ Technology Platform server is running.
2. Open a web browser.
3. Type the following and press **Enter**:

- `http://<servername>:<port>/hub/`
- `"?Model="`
- The name of the model
- `"&Query="`
- The name of the query

For example, if your server is named "myserver", it uses default HTTP port 8080, you are opening a model called "Fraud" containing insurance claim data, and you want to open the entity with an internal ID of "1", you would enter the following:

```
http://myserver:8080/hub/?Model=Fraud&Query=g.v(1)
```

Manage Models

Model management features enable you to manage model backups, jobs, models, and metadata.

Many management features may be invoked from the [Administrative Utility command line interface \(CLI\)](#). [Relationship Analysis Client](#) and [Metadata Insights](#) support similar functionality in a visual interface. You may also use [Metadata Insights](#) in **Platform Client Tools** to copy a logical model or add a new Data Hub model and copy, edit, or delete existing Data Hub models.

Copy Models

You can copy the contents of a Data Hub model in Relationship Analysis Client, in Metadata Insights, or with the Administration Utility Command Line Interface (CLI).

Optionally, when using the Command Line Interface or Relationship Analysis Client, you can choose whether to copy model monitors and queries.

Method	See
Command Line Interface	hub model copy on page 247 For additional commands, see Data Hub Command Line Reference on page 233.
Metadata Insights	Copying a Data Hub Model in Metadata Insights
Relationship Analysis Client	Copying Models on page 156

Back Up Models

It is recommended to back up models. Periodic backups allow you quickly recover a model if it is corrupted or accidentally deleted from the server.

Backed up models are stored in `<Spectrum™ Technology Platform install path>/server/modules/hub/db/backups`. This folder contains one folder for each backed up model (`model.modelname`). A folder contains all the data and metadata for its respective model.

Queries and monitors are stored in the Spectrum configuration database and must be backed up as part of the Spectrum backup schedule.

You can use the Administrative Utility Command Line Interface (CLI) or the Relationship Analysis Client to back up Data Hub models. The CLI commands permit you to perform full or incremental backups of models and may be used in scripts that automate administrative tasks.

Method	See
Command Line Interface	<p>hub backup all on page 241 Backs up all Data Hub models.</p> <p>hub backup model on page 243 Backs up a specific Data Hub model.</p> <p>For additional Data Hub backup commands, see Data Hub Command Line Reference on page 233.</p>
Relationship Analysis Client	Back Up Data Hub Models on page 190

Enabling or Disabling Model Backups

Model backups are enabled by default, as this is the recommended configuration for a Data Hub. Ordinarily you do not need to configure this option.

The `online_backup_enabled` property in the `neo4j.properties` controls whether models on a Data Hub can be backed up.

1. Open `<Spectrum™ Technology Platform install path>/server/modules/hub/db/neo4j.properties`
2. If necessary, uncomment the `online_backup_enabled` property.
3. Set the `dbms.backup.enabled` property to enable or disable backups:

Setting	Description
true	Enables online backups. This is the default value for this setting.
false	Disables online backups.

4. Save the file.

Configure Port Numbers for Model Backups

This section describes how an administrator can change port numbers for both automatic assignments to models and manual assignment for a particular model.

Configure port assignments for model backups to avoid or eliminate conflicts with other port assignments on a server.

- [Automatically assign model backup ports for models](#) on page 229
- [Manually assign a model backup port for a model](#) on page 229

Automatically assign model backup ports for models

This procedure describes how to assign a range of port numbers for automatic assignment to models. Port numbers are assigned sequentially to models as they are created or accessed.

Specify a range of port numbers to use for backups of models. Port numbers are assigned to models starting with the base number and incremented sequentially by 1 as models are created or accessed. The range should not be less than the expected number of models.

1. Open the `<Spectrum™ Technology Platform install path>/server/modules/hub/db/neo4j.properties` file.
2. Set the `dbms.backup.address` property to a range of allowed port values.
The range should exceed the number of models on the server. If necessary, remove the `#` character to uncomment this setting.
For example, `dbms.backup.address=:6372-6627`
3. Save the file.

Manually assign a model backup port for a model

This procedure describes how to manually assign a port number to a model for online backup.

1. Copy the `neo4j.properties` file from `<Spectrum™ Technology Platform install path>/server/modules/hub/db` to `<Spectrum™ Technology Platform install path>/server/modules/hub/db/model.<model name>`.
2. Edit this model-specific properties file by setting the `dbms.backup.address` property to an explicit port number.
If necessary, remove the `#` character to uncomment this setting, and specify a single port number.
For example, `dbms.backup.address=:6373`
3. Save the file.

Note: Properties defined for a model in `hub/db/model.<model_name>/neo4j.properties` override property assignments defined in the default file `hub/db/neo4j.properties`.

Schedule Model Backups

After you have enabled the use of model backups, you can use Data Hub settings in Management Console to schedule regular backups of models.


Note: This method will back up *all* existing models. If you want to back up some models but not others, use the [on-demand method](#) for backing up.

1. In **Management Console**, click **Resources > Data Hub Settings**.
2. Check the **Schedule backup** check box.
3. Designate the frequency and time at which the backups should occur.
4. To perform incremental backups to an existing backup, check the **Incremental** check box.
5. Click **Save**.

Change the Default Backup Directory

Data Hub Settings provides an option to override the default location for Data Hub model backups.

1. In **Management Console**, click **Resources > Data Hub Settings**.
2. Check the **Override default backup directory** check box.
3. Enter the full path to an existing folder.

This initially specifies default location for backups. You can click the browse button  to locate a folder.

Note: The logon account for the Spectrum™ Technology Platform server must have write permission to the specified folder.

4. Click **Save**.

Backing Up Models in a Cluster

To ensure that a backed-up model in a cluster environment is accessible to all servers in the cluster, the model needs to be backed up to a shared location, such as a mapped drive, a network file system (NFS), or a storage area network (SAN).

Perform these steps on every server in the cluster.

1. Open `<Spectrum™ Technology Platform install path>/server/modules/hub/hub.properties`.
2. Set the **hub.backup.path.base** property to the shared location, such as `/sandrive/spectrum/datahub/backups`.
3. Save the file.

Restore Models

You can use the The Administrative Utility Command Line Interface (CLI) or the Relationship Analysis Client to restore Data Hub models from backups.

Method	See
Command Line Interface	hub backup restore on page 244 For additional Data Hub backup commands, see Data Hub Command Line Reference on page 233.
Relationship Analysis Client	Restoring Models on page 190

Clear Model Content

You can clear the contents of a Data Hub model in Relationship Analysis Client or with the Administration Utility Command Line Interface (CLI). This removes the contents of a Data Hub model but retains the model and its metadata in place.

Method	See
Command Line Interface	hub model clear on page 246 For additional Data Hub model commands, see Data Hub Command Line Reference on page 233.
Relationship Analysis Client	Clearing Models on page 157

Remove Models

You can remove Data Hub models in Relationship Analysis Client, in Metadata Insights, or with the Administration Utility Command Line Interface (CLI).

Method	See
Command Line Interface	hub model delete on page 248 For additional Data Hub model commands, see Data Hub Command Line Reference on page 233.
Metadata Insights	Deleting a Data Hub Model
Relationship Analysis Client	Removing Models on page 157

Edit Metadata

You can edit the Data Hub model in Relationship Analysis Client, in Metadata Insights, or with the Administration Utility Command Line Interface (CLI).

Method	See
Command Line Interface	<ul style="list-style-type: none"> • hub schema rename entityType on page 263 • hub schema rename entityProperty on page 262 • hub schema rename relationshipLabel on page 264 • hub schema rename relationshipProperty on page 265 • hub schema delete entityType on page 253 • hub schema delete entityProperty on page 253 • hub schema delete relationshipLabel on page 254 • hub schema delete relationshipProperty on page 255 • hub schema modify indexType on page 261 For other Data Hub schema commands in the CLI, see Data Hub Command Line Reference on page 233.
Metadata Insights	Editing a Data Hub Model in Metadata Insights

Method	See
Relationship Analysis Client	Schema Changes on page 144

View Jobs

You view jobs and job status in Relationship Analysis Client or with the Administration Utility Command Line Interface (CLI).

Method	See
Command Line Interface	<ul style="list-style-type: none"> • hub job list on page 245 • hub job status on page 246
Relationship Analysis Client	Jobs Tab on page 157

Data Hub Command Line Reference

The Administration Utility Command Line Interface (CLI) provides Data Hub commands to manage models, view jobs and job status, and manage model schemas and metadata

You can use CLI commands in a script to automate administrative tasks. You can also use execute CLI interactively in the Administration Utility command window. For more information, see [Administration Utility](#) in the Spectrum™ Technology Platform *Administration Guide*.

hub algorithm betweenness

Runs the betweenness algorithm on a model and saves the results for each entity to a model property.

Centrality algorithms measure the importance and significance of individual entities and relationships. When you run centrality algorithms, the value returned by an algorithm indicates importance of an element. The betweenness algorithm reflects the number of shortest paths between one entity and other entities. It is often used to find entities that serve as a bridge from one part of a graph to another.

Usage

```
hub algorithm betweenness --m model --d direction --wp weightProperty
--lv significantLowValues --op outputProperty --w waitForComplete
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the model.
No	--d <i>direction</i>	Specifies the direction to apply to the algorithm where <i>direction</i> is one of the following: in The results will be based on incoming relationships on the entity. out The results will be based on outgoing relationships on the entity. both The results will be based on both incoming and outgoing relationships on the entity. This is the default value.
No	--wp <i>weightProperty</i>	Specifies a relationship property to use to measure how unfavorable a relationship is. By default, a higher value indicates a negative association. The default setting is null.
No	--lv <i>significantLowValues</i>	If a relationship property is used as weight, this specifies whether a lower value is considered better than a higher value. true Specifies that a lower value is considered better than a higher value for a relationship property used as weight. For example, if the property is some sort of ranking system, 1 or 1st would be considered the best value. If the property is distance, and you are trying to determine the shortest route, 5 miles would be considered better than 10 miles. false Specifies that a higher value is considered better than a lower value for a relationship property used as weight. This is the default value.
No	--op <i>outputProperty</i>	Specifies the output property name to be something other than the algorithm name. The default is Betweenness.

Required Argument	Description
No	<p><code>--w <i>waitForComplete</i></code></p> <p>Specifies whether to wait for jobs to complete in a synchronous mode.</p> <p>true</p> <p>Specifies to wait for jobs to complete in a synchronous mode.</p> <p>false</p> <p>Specifies to not wait for jobs to complete in a synchronous mode. This is the default value.</p>

Example

The following will run the betweenness algorithm on the 911 model.

```
hub algorithm betweenness --m 911
```

hub algorithm closeness

Runs the closeness algorithm on a model and saves the results for each entity to a model property. Centrality algorithms measure the importance and significance of individual entities and relationships. When you run centrality algorithms, the value returned by an algorithm indicates importance of an element. The closeness centrality of an entity measures its average farness (inverse distance) to all other entities. Entities with a high closeness score have the shortest distances to all other nodes.

Usage

```
hub algorithm closeness --m model --d direction --m method --wp weightProperty --lv significantLowValues --op outputProperty --w waitForComplete
```

Required Argument	Description
Yes	<p><code>--m <i>model</i></code></p> <p>Specifies the model.</p>
No	<p><code>--d <i>direction</i></code></p> <p>Specifies the direction to apply to the algorithm where <i>direction</i> is one of the following:</p> <p>in</p> <p>The results will be based on incoming relationships on the entity.</p> <p>out</p> <p>The results will be based on outgoing relationships on the entity.</p> <p>both</p>

Required Argument	Description
	The results will be based on both incoming and outgoing relationships on the entity. This is the default value.
No <code>--me <i>method</i></code>	<p>Specifies the method in which results are returned:</p> <ul style="list-style-type: none"> • s—Standard. Results are based on the number of attachments, or relationships, an entity has as well as the reverse of the sum of shortest paths to each entity. This is the default value. • d—Dangalchev. Results are based not only on the number of entities linked to another entity but also the number of relationships in each of the linked entities. • o—Opsahl. Results are based on the sum of reversed shortest paths to each entity.
No <code>--wp <i>weightProperty</i></code>	Specifies a relationship property to use to measure how unfavorable a relationship is. By default, a higher value indicates a negative association. The default setting is null.
No <code>--lv <i>significantLowValues</i></code>	<p>If a relationship property is used as weight, this specifies whether a lower value is considered better than a higher value.</p> <p>true</p> <p>Specifies that a lower value is considered better than a higher value for a relationship property used as weight. For example, if the property is some sort of ranking system, 1 or 1st would be considered the best value. If the property is distance, and you are trying to determine the shortest route, 5 miles would be considered better than 10 miles.</p> <p>false</p> <p>Specifies that a higher value is considered better than a lower value for a relationship property used as weight. This is the default value.</p>
No <code>--op <i>outputProperty</i></code>	Specifies the output property name to be something other than the algorithm name. The default is Closeness.
No <code>--w <i>waitForComplete</i></code>	<p>Specifies whether to wait for jobs to complete in a synchronous mode.</p> <p>true</p>

Required Argument	Description
	Specifies to wait for jobs to complete in a synchronous mode.
false	Specifies to not wait for jobs to complete in a synchronous mode. This is the default value.

Example

The following will run the closeness algorithm on the 911 model.

```
hub algorithm closeness --m 911
```

hub algorithm degree

Runs the degree algorithm on a model and saves the results for each entity to a model property.

Centrality algorithms measure the importance and significance of individual entities and relationships. When you run centrality algorithms, the value returned by an algorithm indicates the importance of an element. The degree algorithm reflects the number of relationships on an entity. The Degree Centrality algorithm can help find important or popular entities in a graph.

Usage

```
hub algorithm degree --m model --d direction --wp weightProperty --lv significantLowValues --op outputProperty --w waitForComplete
```

Required Argument	Description
Yes --m <i>model</i>	Specifies the model.
No --d <i>direction</i>	Specifies the direction to apply to the algorithm where <i>direction</i> is one of the following: <ul style="list-style-type: none"> in <p>The results will be based on incoming relationships on the entity.</p> out <p>The results will be based on outgoing relationships on the entity.</p> both <p>The results will be based on both incoming and outgoing relationships on the entity. This is the default value.</p>

Required	Argument	Description
No	<code>--wp <i>weightProperty</i></code>	Specifies a relationship property to use to measure how unfavorable a relationship is. By default, a higher value indicates a negative association. The default setting is null.
No	<code>--lv <i>significantLowValues</i></code>	<p>If a relationship property is used as weight, this specifies whether a lower value is considered better than a higher value.</p> <p>true</p> <p>Specifies that a lower value is considered better than a higher value for a relationship property used as weight. For example, if the property is some sort of ranking system, 1 or 1st would be considered the best value. If the property is distance, and you are trying to determine the shortest route, 5 miles would be considered better than 10 miles.</p> <p>false</p> <p>Specifies that a higher value is considered better than a lower value for a relationship property used as weight. This is the default value.</p>
No	<code>--op <i>outputProperty</i></code>	Specifies the output property name to be something other than the algorithm name. The default is Degree.
No	<code>--w <i>waitForComplete</i></code>	<p>Specifies whether to wait for jobs to complete in a synchronous mode.</p> <p>true</p> <p>Specifies to wait for jobs to complete in a synchronous mode.</p> <p>false</p> <p>Specifies to not wait for jobs to complete in a synchronous mode. This is the default value.</p>

Example

The following will run the degree algorithm on the 911 model.

```
hub algorithm degree --m 911
```

hub algorithm influence

Runs the influence algorithm on a model and saves the results for each entity to a model property. The influence algorithm reflects the importance of an entity, based on its connections to high-scoring entities.

Centrality algorithms measure the importance and significance of individual entities and relationships. When you run centrality algorithms, the value returned by an algorithm indicates the importance of an element. The influence algorithm implements Eigenvector Centrality to measure the transitive influence or connectivity of entities. Relationships to high-scoring entities contribute more to the score of an entity than connections to low-scoring nodes. A high score means that an entity is connected to other entities that have high scores.

Usage

```
hub algorithm influence --m model --d direction --p precision --wp weightProperty
--lv significantLowValues --op outputProperty --w waitForComplete
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the model.
No	--d <i>direction</i>	Specifies the direction to apply to the algorithm where <i>direction</i> is one of the following: in The results will be based on incoming relationships on the entity. out The results will be based on outgoing relationships on the entity. both The results will be based on both incoming and outgoing relationships on the entity. This is the default value.
No	--p <i>precision</i>	Specifies how precise the results should be. A lower precision will return more accurate results, but the algorithm will run more slowly. This argument may be set between 0.00001 and 0.1. The default is 0.01.
No	--wp <i>weightProperty</i>	Specifies a relationship property to use to measure how unfavorable a relationship is. By default, a higher value indicates a negative association. The default setting is null.

Required Argument	Description
No <code>--lv <i>significantLowValues</i></code>	<p>If a relationship property is used as weight, this specifies whether a lower value is considered better than a higher value.</p> <p>true</p> <p>Specifies that a lower value is considered better than a higher value for a relationship property used as weight. For example, if the property is some sort of ranking system, 1 or 1st would be considered the best value. If the property is distance, and you are trying to determine the shortest route, 5 miles would be considered better than 10 miles.</p> <p>false</p> <p>Specifies that a higher value is considered better than a lower value for a relationship property used as weight. This is the default value.</p>
No <code>--op <i>outputProperty</i></code>	<p>Specifies the output property name to be something other than the algorithm name. The default is Influence.</p>
No <code>--w <i>waitForComplete</i></code>	<p>Specifies whether to wait for jobs to complete in a synchronous mode.</p> <p>true</p> <p>Specifies to wait for jobs to complete in a synchronous mode.</p> <p>false</p> <p>Specifies to not wait for jobs to complete in a synchronous mode. This is the default value.</p>

Example

The following command line runs the algorithm on the 911 model.

```
hub algorithm influence --m 911
```


hub backup all

Backs up all Data Hub models.

Use the `hub backup all` command to perform a full or incremental backup of all Data Hub models. An incremental backup adds changes that were made to a model since a previous backup. A model is added to the default backup directory for Data Hub models unless you specify a different location.

Usage

```
hub backup all --f fullBackup --p path
```

Required	Argument	Description
No	<code>--f <i>fullBackup</i></code>	<p>Performs a full or incremental backup of all models, where <i>fullBackup</i> is one of the following:</p> <p>true</p> <p>Performs a full backup of all models. Full backups replace any existing backups of the models. This is the default setting.</p> <p>false</p> <p>Performs an incremental backup of all models to an existing backup.</p>
No	<code>--p <i>path</i></code>	<p>Specifies the path and folder to which you want to save the backups. If you omit this option the backups are placed in the default backup directory.</p> <p>Note: The default backup directory location (<i>SpectrumFolder/server/modules/hub/db/backups</i>) may be changed by uncommenting and editing the <code>hub.models.path.base</code> setting in the <i>SpectrumFolder/server/modules/hub/hub.properties</i> file.</p>

Example

This example backs up all existing models to a folder called HubBackup on the local C drive.

```
hub backup all --f true --p C:\HubBackup
```

hub backup delete

Deletes a backup of a Data Hub model.

Use the `hub backup delete` command to delete a backup of a Data Hub model. A model is deleted from the default backup directory for Data Hub models unless you specify a different location.

Usage

```
hub backup delete --m model --p path
```

Required	Argument	Description
Yes	<code>--m <i>model</i></code>	Specifies the name of the model backup you want to delete.
No	<code>--p <i>path</i></code>	Specifies the path and folder to which the backup was saved. If you omit this option the command will delete the model backup from the default backup directory.

Note: The default backup directory location (`SpectrumFolder/server/modules/hub/db/backups`) may be changed by uncommenting and editing the `hub.models.path.base` setting in the `SpectrumFolder/server/modules/hub/hub.properties` file.

Example

This example deletes a backed-up model called PersonalBanking from the default backup folder.

```
hub backup delete --m PersonalBanking
```

hub backup list

Lists backups of Data Hub models.

Use the `hub backup list` command to return a list of all Data Hub models that have been backed up to a particular folder. The command lists models in the default backup directory for Data Hub models unless you specify a different location.

Usage

```
hub backup list --p path
```

Required	Argument	Description
No	<code>--p path</code>	Specifies the path and folder to which the backups were saved. If you omit this option the command will return a list of backed-up models in the default backup directory. Note: The default backup directory location (<i>SpectrumFolder/server/modules/hub/db/backups</i>) may be changed by uncommenting and editing the <code>hub.models.path.base</code> setting in the <i>SpectrumFolder/server/modules/hub/hub.properties</i> file.

Example

This example returns a list of the backed-up models from the default backup folder.

```
hub backup list
```

hub backup model

Backs up a specific Data Hub model.

Use the `hub backup model` command to perform a full or incremental backup of a specified Data Hub model. The incremental method adds changes that were made to a model since a previous update. A backup is located in the default backup folder for Data Hub models unless a different location is specified with the `path` (`--p`) option.

Usage

```
hub backup model --m model --f fullBackup --p path
```

Required	Argument	Description
Yes	<code>--m model</code>	Specifies the name of the model you want to backup.
No	<code>--f fullBackup</code>	Performs a full or incremental backup of the model, where <i>fullBackup</i> is one of the following: true Performs a full, initial backup of the model. This is the default setting. false Performs an incremental backup of the model to an existing backup.
No	<code>--p path</code>	Specifies the path and folder to which you want to save the backup. If you omit this option the backup is placed default backup directory.

Required Argument	Description
	Note: The default backup directory location (<i>SpectrumFolder/server/modules/hub/db/backups</i>) may be changed by uncommenting and editing the <code>hub.models.path.base</code> setting in the <i>SpectrumFolder/server/modules/hub/hub.properties</i> file.

Example

This example backs up a single model called ConsumerFraud to a folder called HubModelBackup in the `C:\DataHub` directory. If a model by that same name already exists, the restored model will be updated.

```
hub backup model --m ConsumerFraud --f false --p
C:\DataHub\HubModelBackups\model.ConsumerFraud
```

hub backup restore

Restores a Data Hub model from a backup.

Use the `hub backup restore` command to restore the backup of a Data Hub model. You may optionally choose whether to restore a model only when there is not already an existing model of the same name. The command restores a model from the default backup location if you do not specify a different location.

Usage

```
hub backup restore --m model --d deleteIfExists --p path
```

Required Argument	Description
Yes	<code>--m <i>model</i></code> Specifies the name of the model you want to restore.
No	<code>--d <i>deleteIfExists</i></code> Specifies whether to delete an existing model of the same name, where <i>deleteIfExists</i> is one of the following: true Deletes the existing model and restores the backed-up model. This is the default setting. false Leaves the existing model in place and does not restore the backed-up model.
No	<code>--p <i>path</i></code> Specifies the path and folder to which the backup was saved. If you omit this option the backup is restored from the default backup directory.

Required Argument	Description
	<p>Note: The default backup directory location (<i>SpectrumFolder/server/modules/hub/db/backups</i>) may be changed by uncommenting and editing the <code>hub.models.path.base</code> setting in the <i>SpectrumFolder/server/modules/hub/hub.properties</i> file.</p>

Example

This example restores a backed-up model called ConsumerFraud from the `C:\DataHub\HubModelBackup` folder. If a model by that same name already exists, the restored model will overwrite it.

```
hub backup restore --m ConsumerFraud --d true --p
C:\DataHub\HubModelBackup
```

hub job list

Returns a list of all Data Hub jobs.

Use the `hub job list` command to return a list of all Data Hub jobs with or without date and time specifications.

Usage

```
hub job list --f from datetime --t to datetime
```

Required Argument	Description
No <code>--f <i>from datetime</i></code>	<p>If you want to see the list for a specific date and time range, specify the starting date and time for the range, in the format 'MM-dd-yyyy HH:mm:ss'. For example, December 31, 2014 1:00 PM would be specified as '12-31-2014 13:00:00'.</p> <p>When you specify a date and time range, the list will include jobs that started execution on or after the date you specified in the <code>--f</code> argument and before the date you specify in the <code>--t</code> argument.</p> <p>If you omit this argument the list will include jobs that started execution on the current date.</p>
No <code>--t <i>to datetime</i></code>	<p>If you want to see the list for a specific date and time range, specify the ending date and time for the range, in the format 'MM-dd-yyyy HH:mm:ss'. For example, December 31, 2014 1:00 PM would be specified as '12-31-2014 13:00:00'.</p>

Required Argument	Description
	When you specify a date and time range, the list will include jobs that started execution on or after the date you specified in the <code>--f</code> argument and before the date you specify in the <code>--t</code> argument.
	If you omit this argument the list will include jobs that started execution on or after the date specified in the <code>--f</code> argument.

Example

This example lists all Data Hub jobs run on or after January 1, 2010 at 00:00:00.

```
hub job list --f '01-01-2010 00:00:00'
```

hub job status

Returns the status of a Data Hub job.

Use the `hub job status` command to return the status of a Data Hub job.

Usage

```
hub job status --id jobID
```

Required Argument	Description
Yes <code>--id <i>jobID</i></code>	Specifies the ID of the Data Hub job.

Example

This example returns the status of Data Hub job 24.

```
hub job status --id 24
```

hub model clear

Removes the contents of a Data Hub model.

Use the `hub model clear` command to remove the contents of a Data Hub model but leave it and its metadata in place.

Usage

```
hub backup clear --m model
```

Required	Argument	Description
Yes	<code>--m <i>model</i></code>	Specifies the name of the model whose contents you want to clear.

Example

This example clears a model called CustomerDB_032018.

```
hub model clear --m CustomerDB_032018
```

hub model copy

Copies the contents of a Data Hub model.

Use the `hub model copy` command to copy the contents of a Data Hub model, and optionally its monitors and its queries.

Usage

```
hub model copy --m model --nm newmodel --cm copymonitors --cq copyqueries
```

Required	Argument	Description
Yes	<code>--m <i>model</i></code>	Specifies the name of the model you want to copy.
Yes	<code>--nm <i>newmodel</i></code>	Specifies the name of the new model.
No	<code>--cm <i>copymonitors</i></code>	Specifies whether to copy any existing monitors from the old model into the new model, where <i>copymonitors</i> is one of the following: true Copies monitors. This is the default setting. false Does not copy monitors.
No	<code>--cq <i>copyqueries</i></code>	Specifies whether to copy any saved queries from the old model into the new model, where <i>copyqueries</i> is one of the following: true Copies queries. This is the default setting. false Does not copy queries.

Example

This example copies a model called CustomerBanking_DataType from the default backup folder and names the copy CustomerBanking_DataType_New. It also copies any monitors or queries associated with the old model into the new model.

```
hub model copy --m CustomerBanking_DataType --nm
CustomerBanking_DataType_New --cm true --cq true
```

hub model create security

Creates secured entities for a Data Hub model.

Use the `hub model create security` command to create secured entities for a Data Hub model. These provide override options in Management Console `System > Security > Access Control`.

Usage

```
hub model create security --m model
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model for which you want to create secured entities.

Example

This example creates secured entities for a model called Single_Account_Holders.

```
hub model create security --m Single_Account_Holders
```

hub model delete

Deletes a Data Hub model.

Use the `hub model delete` command to delete a specific Data Hub model.

Usage

```
hub model delete --m model
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model you want to delete.

Example

This example deletes a model called PersonalBanking.

```
hub model delete --m PersonalBanking
```

hub model export

Exports a Data Hub model.

Use the `hub model export` command to export a Data Hub model as a folder structure.

Usage

```
hub model export --m ModelName --p Path --xd TrueOrFalse
```

Required	Argument	Description
Yes	--m <i>ModelName</i>	Specifies the name of the model you want to export.
Yes	--p <i>Path</i>	Specifies the path where you want to save the export folder.
No	--xd <i>TrueOrFalse</i>	Specifies whether to exclude data in the export, where <i>TrueOrFalse</i> is one of the following: true Excludes data. This is the default setting. false Does not exclude data.

Example

This example exports in folder format a model called Fraud_2015 to the `HubModels` directory on the C drive. It also retains data in the export.

```
hub model export --m Fraud_2015 --p C:\HubModels --xd false
```

hub model import

Imports a Data Hub model.

Use the `hub model import` command to import a Data Hub model.

Usage

```
hub model import --m ModelName --p Path
```

Required	Argument	Description
Yes	<code>--m <i>ModelName</i></code>	Specifies the name of the model you want to export.
Yes	<code>--p <i>Path</i></code>	Specifies the path for the location of the model you are importing.

Example

This example imports a model called `Fraud_2015` from the `HubModels` folder on the C: drive.

```
hub model import --m Fraud_2015 --p C:\HubModels
```

hub model list

Lists Data Hub models.

Use the `hub model list` command to return a list of all Data Hub models as well as counts for entities and relationships for each model.

Usage

```
hub model list --c counts
```

Required	Argument	Description
No	<code>--c <i>counts</i></code>	Specifies whether to include counts for entities and relationships, where <i>counts</i> is one of the following: <ul style="list-style-type: none"> true Includes counts. This is the default setting. false Does not include counts.

Example

This example lists all Data Hub models and provides counts for entities and relationships for each model.

```
hub model list --c true
```

hub model reindex

Reindexes Data Hub models.

Use the `hub model reindex` command to reindex a single Data Hub model or all Data Hub models. The utility will return a status message for each model on a separate line, as shown below.

```
|  MODEL NAME  | STATUS | INDEX TYPE      | FAILURE MESSAGE IF APPLICABLE |
|-----+-----+-----+-----+
| Fraud_Index  | PASSED | lucene+native-2.0 |                               |
| Index_Insured | PASSED | lucene+native-2.0 |                               |
```

The utility will also return failure messages, if necessary:

```
|  MODEL NAME  | STATUS | INDEX TYPE      | FAILURE MESSAGE IF APPLICABLE |
|-----+-----+-----+-----+
| HUB_Index    | FAILED | null            | Model does not exist.         |
```

Usage

```
hub model reindex --m model --a all
```

Required	Argument	Description
No	--m <i>model</i>	Specifies the name of the model whose contents you want to reindex if you are reindexing a single model. Note: You must include either --m or --a, not both.
No	--a <i>all</i>	Specifies to reindex all models, where <i>all</i> is one of the following: true Reindexes all models. false Does not reindex all models. This is the default setting. Note: You must include either --a or --m, not both.

Example

This example reindexes all Data Hub models.

```
hub model reindex --a
```

hub schema copy

Copies model metadata.

Use the `hub schema copy` command to copy Data Hub model metadata, and optionally its monitors and queries.

Usage

```
hub schema copy --m model --nm newmodel --cm copymonitors --cq copyqueries
```

Required	Argument	Description
Yes	<code>--m <i>model</i></code>	Specifies the name of the model whose schema you want to copy.
Yes	<code>--nm <i>newmodel</i></code>	Specifies the name of the new model.
No	<code>--cm <i>copymonitors</i></code>	Specifies whether to copy any existing monitors from the old model into the new model, where <i>copymonitors</i> is one of the following: true Copies monitors. This is the default setting. false Does not copy monitors.
No	<code>--cq <i>copyqueries</i></code>	Specifies whether to copy any saved queries from the old model into the new model, where <i>copyqueries</i> is one of the following: true Copies queries. This is the default setting. false Does not copy queries.

Example

This example copies the schema from a model called `PersonalLending` from the default backup folder and names the copy `PersonalLending_New`. It also copies any monitors associated with the old model but does not copy any queries associated with the old model.

```
hub schema copy --m PersonalLending --nm PersonalLending_New
--cm true --cq false
```

hub schema delete entityProperty

Delete a model property.

Use the `hub schema delete entityProperty` command to delete a Data Hub model property.

Usage

```
hub schema delete entityProperty --m model --e entityType --p property --w waitForComplete
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model whose entity type property you want to delete.
No	--e <i>entityType</i>	Specifies the target entity type; includes all entity types if not specified.
Yes	--p <i>property</i>	Specifies the property you want to delete.
No	--w <i>waitForComplete</i>	Specifies whether to wait for jobs to complete in a synchronous mode, where <i>waitForComplete</i> is one of the following: true Waits for jobs to complete. false Does not wait for jobs to complete. This is the default setting.

Example

This example deletes the property HireDate from a model called Staff and an entity type of EmployeeName.

```
hub schema delete entityProperty --m Staff --e EmployeeName --p HireDate
```

hub schema delete entityType

Delete a model entity type.

Use the `hub schema delete entityType` command to delete a Data Hub model entity type. It optionally specifies whether to complete jobs in synchronous mode.

Usage

```
hub schema delete entityType --m model --e entityType --w waitForComplete
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model whose entity type you want to delete.
Yes	--e <i>entityType</i>	Specifies the type of entity to be deleted.
No	--w <i>waitForComplete</i>	Specifies whether to wait for jobs to complete in a synchronous mode, where <i>waitForComplete</i> is one of the following: true Waits for jobs to complete. false Does not wait for jobs to complete. This is the default setting.

Example

This example deletes the entity type Employee from a model called PersonalLending.

```
hub schema delete entityType --m PersonalLending --e Employee
```

hub schema delete relationshipLabel

Deletes a relationship label from a model.

Use the `hub schema delete relationshipLabel` command to delete a relationship label from a model. You may optionally choose wait for other jobs in synchronous mode.

Usage

```
hub schema delete relationshipLabel --m model --r relationshipLabel --s sourceEntityType --t targetEntityType --w waitForComplete
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model whose relationship label you want to delete.
Yes	--r <i>relationshipLabel</i>	Specifies the relationship label to be deleted.
No	--s <i>sourceEntityType</i>	Specifies the type of source entity.
No	--t <i>targetEntityType</i>	Specifies the type of target entity.

Required	Argument	Description
No	<code>--w <i>waitForComplete</i></code>	Specifies whether to wait for jobs to complete in a synchronous mode, where <i>waitForComplete</i> is one of the following: true Waits for jobs to complete. false Does not wait for jobs to complete. This is the default setting.

Example

This example deletes the relationshipLabel Hired from a model called Staff.

```
hub schema delete relationshipLabel --m Staff --r Hired
```

hub schema delete relationshipProperty

Deletes a relationship property from a model.

Use the `hub schema delete relationshipProperty` command to delete a Data Hub model relationship property.

Usage

```
hub schema delete relationshipProperty --m model --r relationshipLabel --p property --s sourceEntityType --t targetEntityType --w waitForComplete
```

Required	Argument	Description
Yes	<code>--m <i>model</i></code>	Specifies the name of the model whose entity type or relationship label property you want to delete.
No	<code>--r <i>relationshipLabel</i></code>	Specifies the target relationship label; includes all relationship labels if not specified.
Yes	<code>--p <i>property</i></code>	Specifies the property you want to delete.
No	<code>--s <i>sourceEntityType</i></code>	Specifies the type of source entity.
No	<code>--t <i>targetEntityType</i></code>	Specifies the type of target entity.
No	<code>--w <i>waitForComplete</i></code>	Specifies whether to wait for jobs to complete in a synchronous mode, where <i>waitForComplete</i> is one of the following: true Waits for jobs to complete. false

Required Argument	Description
	Does not wait for jobs to complete. This is the default setting.

Example

This example deletes the property HireDate from a model called Staff and a relationship label of Hired.

```
hub schema delete relationshipProperty --m Staff --r Hired --p HireDate
```

hub schema export

Exports a model.

Use the `hub schema export` command to export a Data Hub model, its metadata, its monitors, and its queries. If you do not specify a path to where you would like the model exported, the system export the model to the current working directory using the name you specify.

Usage

```
hub schema export --m model --f file --cm copymonitors --cq copyqueries
```

Required	Argument	Description
Yes	<code>--m <i>model</i></code>	Specifies the name of the model you want to export.
Yes	<code>--p <i>path</i></code>	(Deprecated) Specifies the path where you want to save the export folder. This path is relative to where you have installed the Spectrum™ Technology Platform server.
No	<code>--f <i>file</i></code>	Specifies the path where you want to save the export folder. This path is relative to where you have installed the Spectrum™ Technology Platform Administration Utility.
No	<code>--cm <i>copymonitors</i></code>	Specifies whether to export any existing monitors, where <i>copymonitors</i> is one of the following: true Exports monitors. This is the default setting. false Does not export monitors.
No	<code>--cq <i>copyqueries</i></code>	Specifies whether to export any saved queries, where <i>copyqueries</i> is one of the following: true Exports queries. This is the default setting.

Required	Argument	Description
	false	Does not export queries.

Example

This example exports the schema for a model called Fraud_2015 to the HubModels directory on the C drive. It does not export any monitors but does export any queries associated with the model.

```
hub schema export --m Fraud_2015 --f C:\HubModels --cm false
--cq true
```

hub schema import

Imports a model.

Use the `hub schema import` command to import a Data Hub model, its metadata, its monitors, and its queries. If you do not specify a path where you would like the model imported, the system will look for a file of the name you specify in the current working directory.

Usage

```
hub schema import --m model --f file
```

Required	Argument	Description
Yes	<code>--m <i>model</i></code>	Specifies the name of the model whose schema you want to import.
No	<code>--p <i>path</i></code>	(Deprecated) Specifies the path for the location of the model whose schema you are importing. This path is relative to where you have installed the Spectrum™ Technology Platform server.
No	<code>--f <i>file</i></code>	Specifies the path for the location of the model whose schema you are importing. This path is relative to where you have installed the Spectrum™ Technology Platform Administration Utility.

Example

This example imports the schema for a model called Fraud_2015 from the HubModels directory on the C drive.

```
hub schema import --m Fraud_2015 --f C:\HubModels
```

hub schema importLogicalModel

Imports a Metadata Insights Logical Model into Data Hub.

Use the `hub schema importLogicalModel` command to import a Metadata Insights Logical Model into Data Hub.

Usage

```
hub schema importLogicalModel --m model --n logicalModelName
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name you would like to give the model in Data Hub.
No	--n <i>logicalModelName</i>	Specifies the name of the Metadata Insights model whose schema you are importing.

Example

This example imports a Metadata Insights model called Insured and names it Insured2018.

```
hub schema importLogicalModel --m Insured2018 --n Insured
```

hub schema list all

List entity types, relationship labels and total counts for a model.

Use the basic `hub schema list all` command to return a list of entity types, relationship labels and total counts for a model. Add the `verbose` argument to include entity properties, relationship label connections, relationship properties, and indexed properties.

Usage

```
hub schema list all --m model --v verbose
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model whose schema information you want to list.
No	--v <i>verbose</i>	Specifies whether to include verbose output, where <i>verbose</i> is one of the following: true Includes verbose output. false

Required	Argument	Description
		Does not include verbose output. This is the default setting.

Example

This example lists all relationship properties for a model named PersonalBanking and does not include verbose output.

```
hub schema list all --m PersonalBanking
```

hub schema list entityProperties

Lists entity properties for a model.

Use the `hub schema list entityProperties` command to return a list of all entity properties for a model.

Usage

```
hub schema list entityProperties --m model --e entityType --i indexedOnly
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model whose entity properties you want to list.
No	--e <i>entityType</i>	Limits the results to the specified entity type.
No	--i <i>indexedOnly</i>	Specifies whether to limit the results to indexed properties only, where <i>indexedOnly</i> is one of the following: <ul style="list-style-type: none"> true <ul style="list-style-type: none"> Limits the results. This is the default setting. false <ul style="list-style-type: none"> Does not limit the results.

Example

This example lists all entity properties for a model named PersonalBanking, filters the results to include just the Customer type, and filters the results to indexed properties only.

```
hub schema list entityProperties --m PersonalBanking --e Customer
```

hub schema list entityTypees

Lists entity types for a model.

Use the `hub schema list entityTypees` command to return a list of all entity types for a model.

Usage

```
hub schema list entityTypees --m model
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model whose entity types you want to list.

Example

This example lists all entity types for a model named Fraud.

```
hub schema list entityTypees --m Fraud
```

hub schema list relationshipLabels

Lists relationship labels for a model.

Use the `hub schema list relationshipLabels` command to return a list of all relationship labels for a model.

Usage

```
hub schema list relationshipLabels --m model --s sourceEntityType --t targetEntityType --c showConnections
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model whose relationship labels you want to return.
No	--s <i>sourceEntityType</i>	Specifies the type of source entity.
No	--t <i>targetEntityType</i>	Specifies the type of target entity.
No	--c <i>showConnections</i>	Specifies whether to show source and target entity types, where <i>showConnections</i> is one of the following: true Shows connections. false

Required	Argument	Description
		Does not show connections. This is the default setting.

Example

This example returns a list of relationship labels with source and target entity types for a model called June2017 with a source entity type of Customer and a target entity type of AccountType.

```
hub schema list relationshipLabels --m June2017 --s Customer --t AccountType --c
```

hub schema list relationshipProperties

Lists entity properties for a model.

Use the `hub schema list relationshipProperties` command to return a list of all entity properties for a model.

Usage

```
hub schema list relationshipProperties --m model --r relationshipLabel
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model whose relationship properties you want to list.
No	--r <i>relationshipLabel</i>	Filters the results to the specified relationship label type.

Example

This example lists all relationship properties for a model named PrivateBanking and filters the results to include just the Current type.

```
hub schema list relationshipProperties --m PrivateBanking --r Current
```

hub schema modify indexType

Changes the index type of a model.

Use the `hub schema modify indexType` command to change the index type of a Data Hub model property.

Usage

```
hub schema modify indexType --m model --p property --i index --w waitForComplete
```

Required	Argument	Description
Yes	--m <i>model</i>	Specifies the name of the model whose entity type you want to modify.
Yes	--p <i>property</i>	Specifies the property you want to index.
Yes	--i <i>index</i>	Specifies the property index. NONE Removes the property index. EXACT Sets the property index to exact. CASE_INSENSITIVE Sets the property index to case insensitive.
No	--w <i>waitForComplete</i>	Specifies whether to wait for jobs to complete in a synchronous mode, where <i>waitForComplete</i> is one of the following: true Waits for jobs to complete. false Does not wait for jobs to complete. This is the default setting.

Example

This example changes the index type for a property called HireDate in a model called Staff to exact.

```
hub schema modify indexType --m Staff --p HireDate --i EXACT
```

hub schema rename entityProperty

Rename a model property.

Use the `hub schema rename entityProperty` command to rename a Data Hub model property.

Usage

```
hub schema rename entityProperty --m model --e entityType --p property  
--np newProperty --w waitForComplete
```

Required	Argument	Description
Yes	<code>--m <i>model</i></code>	Specifies the name of the model whose entity property you want to rename.
No	<code>--e <i>entityType</i></code>	Specifies the target entity type; includes all entity types if not specified.
Yes	<code>--p <i>property</i></code>	Specifies the property you want to rename.
Yes	<code>--np <i>newProperty</i></code>	Specifies the new property name.
No	<code>--w <i>waitForComplete</i></code>	Specifies whether to wait for jobs to complete in a synchronous mode, where <i>waitForComplete</i> is one of the following: true Waits for jobs to complete. false Does not wait for jobs to complete. This is the default setting.

Example

This example renames the property HireDate to Start Date in a model called Staff with a relationship label of Hired.

```
hub schema rename entityProperty --m Staff --r Hired --p
HireDate --np StartDate
```

hub schema rename entityType

Rename an entity type for a model.

Use the `hub schema rename entityType` command to rename a Data Hub model's entity type.

Usage

```
hub schema rename entityType --m model --e entityType --ne newEntityType --w
waitForComplete
```

Required	Argument	Description
Yes	<code>--m <i>model</i></code>	Specifies the name of the model whose entity type you want to rename.
Yes	<code>--e <i>entityType</i></code>	Specifies the type of entity to be renamed.
Yes	<code>--ne <i>newEntityType</i></code>	Specifies the new entity type.

Required	Argument	Description
No	<code>--w <i>waitForComplete</i></code>	Specifies whether to wait for jobs to complete in a synchronous mode, where <i>waitForComplete</i> is one of the following: true Waits for jobs to complete. false Does not wait for jobs to complete. This is the default setting.

Example

This example renames an entity type from Employee to Staff in a model called PersonalLending.

```
hub schema rename entityType --m PersonalLending --e Employee
--ne Staff
```

hub schema rename relationshipLabel

Rename a relationship label.

Use the `hub schema rename relationshipLabel` command to rename a Data Hub model relationship label.

Usage

```
hub schema rename relationshipLabel --m model --r relationshipLabel
--nr newRelationshipLabel --s sourceEntityType --t targetEntityType --w waitForComplete
```

Required	Argument	Description
Yes	<code>--m <i>model</i></code>	Specifies the name of the model whose relationship label you want to rename.
Yes	<code>--r <i>relationshipLabel</i></code>	Specifies the relationship label you want to rename.
Yes	<code>--nr <i>newRelationshipLabel</i></code>	Specifies the new relationship label name.
No	<code>--s <i>sourceEntityType</i></code>	Specifies the type of source entity.
No	<code>--t <i>targetEntityType</i></code>	Specifies the type of target entity.
No	<code>--w <i>waitForComplete</i></code>	Specifies whether to wait for jobs to complete in a synchronous mode, where <i>waitForComplete</i> is one of the following: true Waits for jobs to complete.

Required Argument	Description
	false
	Does not wait for jobs to complete. This is the default setting.

Example

This example renames the relationshipLabel Hired to Employed in a model called Staff.

```
hub schema rename relationshipLabel --m Staff --r Hired --nr
Employed
```

hub schema rename relationshipProperty

Rename a model property.

Use the `hub schema rename relationshipProperty` command to rename a Data Hub model's property.

Usage

```
hub schema rename relationshipProperty --m model --r relationshipLabel --p property
--np newProperty --s sourceEntityType --t targetEntityType --w waitForComplete
```

Required Argument	Description
Yes <code>--m <i>model</i></code>	Specifies the name of the model whose relationship property you want to rename.
No <code>--r <i>relationshipLabel</i></code>	Specifies the target relationship label.
Yes <code>--p <i>property</i></code>	Specifies the property you want to rename.
Yes <code>--np <i>newProperty</i></code>	Specifies the new property name.
No <code>--s <i>sourceEntityType</i></code>	Specifies the type of source entity.
No <code>--t <i>targetEntityType</i></code>	Specifies the type of target entity.
No <code>--w <i>waitForComplete</i></code>	Specifies whether to wait for jobs to complete in a synchronous mode, where <i>waitForComplete</i> is one of the following: true Waits for jobs to complete. false Does not wait for jobs to complete. This is the default setting.

Example

This example renames the relationship property HireDate to Start Date in a model called Staff with a relationship label of Hired.

```
hub schema rename relationshipProperty --m Staff --r Hired --p  
HireDate --np StartDate
```

Notices

© 2019 Pitney Bowes. All rights reserved. MapInfo and Group 1 Software are trademarks of Pitney Bowes Software Inc. All other marks and trademarks are property of their respective holders.

USPS® Notices

Pitney Bowes Inc. holds a non-exclusive license to publish and sell ZIP + 4® databases on optical and magnetic media. These trademarks are owned by the United States Postal Service: CASS, CASS Certified, DPV, eLOT, FASTforward, First-Class Mail, Intelligent Mail, LACS^{Link}, NCOA^{Link}, PAVE, PLANET Code, Postal Service, POSTNET, Post Office, RDI, Suite^{Link}, United States Postal Service, Standard Mail, United States Post Office, USPS, ZIP Code, and ZIP + 4. This list is not exhaustive of the trademarks belonging to the Postal Service.

Pitney Bowes Inc. is a non-exclusive licensee of USPS® for NCOA^{Link}® processing.

Prices for Pitney Bowes products, options, and services are not established, controlled, or approved by USPS® or United States Government. When utilizing RDI™ data to determine parcel-shipping costs, the business decision on which parcel delivery company to use is not made by the USPS® or United States Government.

Data Provider and Related Notices

Data Products contained on this media and used within Pitney Bowes Software applications are protected by various trademarks and by one or more of these copyrights:

- © Copyright United States Postal Service. All rights reserved.
 - © 2014 TomTom. All rights reserved. TomTom and the TomTom logo are registered trademarks of TomTom N.V.
 - © 2016 HERE Fuente: INEGI (Instituto Nacional de Estadística y Geografía) - Based upon electronic data © National Land Survey Sweden.
 - © Copyright United States Census Bureau
 - © Copyright Nova Marketing Group, Inc.
- Portions of this program are © Copyright 1993-2019 by Nova Marketing Group Inc. All Rights Reserved
- © Copyright Second Decimal, LLC
 - © Copyright Canada Post Corporation - Data is from a compilation in which Canada Post Corporation is the copyright owner.
 - © 2007 Claritas, Inc.

The Geocode Address World data set contains data licensed from the GeoNames Project (www.geonames.org) provided under the Creative Commons Attribution License ("Attribution License") located at <http://creativecommons.org/licenses/by/3.0/legalcode>. Your use of the GeoNames data (described in the Spectrum™ Technology Platform User Manual) is governed by the terms of the Attribution License, and any conflict between your agreement with Pitney Bowes and the Attribution License will be resolved in favor of the Attribution License solely as it relates to your use of the GeoNames data.



3001 Summer Street
Stamford CT 06926-0700
USA

www.pitneybowes.com