

Spectrum™ Technology Platform

Version 12.0

Metadata Insights Guide



Table of Contents

1 - Getting Started

| | |
|--|---|
| Understanding Your Organization's Data | |
| Assets | 4 |
| A First Look at Metadata Insights | 5 |
| Connecting to Data | 8 |

2 - Modeling

| | |
|----------------|----|
| Logical Model | 61 |
| Physical Model | 63 |
| Mapping | 71 |
| Model Store | 86 |

3 - Profiling

| | |
|--------------------------|-----|
| Creating a Profile | 99 |
| Running a Profile | 103 |
| Scheduling a Profile | 104 |
| Viewing Analysis History | 106 |
| Data Profiling Result | 107 |

4 - Lineage and Impact Analysis

| | |
|--|-----|
| Lineage and Impact Analysis Technology | |
| Preview | 112 |
| Viewing Lineage and Impact Analysis | 112 |
| Icons in Lineage and Impact Analysis | 115 |
| Uses | 119 |

1 - Getting Started

In this section

| | |
|---|---|
| Understanding Your Organization's Data Assets | 4 |
| A First Look at Metadata Insights | 5 |
| Connecting to Data | 8 |

Understanding Your Organization's Data Assets

If your organization is like most, you have a large number of data assets containing everything from customer contact information and purchase history, to financial data, to transaction records, and more. These systems can run on different platforms, sometimes managed by different departments with different security controls. There is potentially a wealth of data available to you to answer business questions, but it is a challenge to figure out which systems contain the data you need, which systems to trust, and how they're all connected.

Metadata Insights provides the visibility you need to identify the most trusted sources of data to use to satisfy a business request.

1. Start by connecting the physical data assets in your organization to Spectrum™ Technology Platform. See [Introduction to Data Sources](#) on page 8.
2. Then, define a physical data model to represent your data assets in Metadata Insights. Going through this process will help you understand how your data assets are structured, such as the tables and columns in each database, and the relationships between tables. See [Adding a Physical Data Model](#) on page 63.
3. With an understanding of the physical data assets available to you, you will want to make sure that the underlying data is of good quality. Use profiling to scan your data assets, identify the types of data contained in them (such as names, email addresses, and currency), and identify incomplete and malformed data. See [Creating a Profile](#) on page 99.

Tip: Using the reports from profiling, you can create Spectrum™ Technology Platform flows to improve data quality. If you have not licensed one of the data quality modules for Spectrum™ Technology Platform, contact your Pitney Bowes Account Executive.

4. With a physical data model created and a clear understanding of the state of your data through profiling, you can create logical models to represent the business entities that your business wants to understand, such as customers, vendors, or products. In this process you select the sources for the data you want to use to populate each entity, such as customer addresses and purchase history. See [Creating a Logical Model](#) on page 61.
5. To maintain your data assets you need to understand how they're all connected and how data flows from source to destination. Use the lineage and impact analysis feature of Metadata Insights to view the dependencies between data sources, destinations, and the processes that use the data. With this information, you can make informed decisions about the impact of a change to data sources, troubleshoot unexpected results, and understand how Spectrum™ Technology Platform entities like flows, subflows, and Spectrum databases affect each other. For more information, see [Viewing Lineage and Impact Analysis](#) on page 112.

A First Look at Metadata Insights

Metadata Insights gives you the control you need to deliver accurate and timely data-driven insights to your business. Use Metadata Insights to develop data models, view the flow of data from source to business application, and assess the quality of your data through profiling. With this insight, you can identify the data resources to use to answer particular business questions, adapt and optimize processes to improve the usefulness and consistency of data across your business, and troubleshoot data issues.

To access Metadata Insights, open a web browser and go to:

```
http://server:port/metadata-insights
```

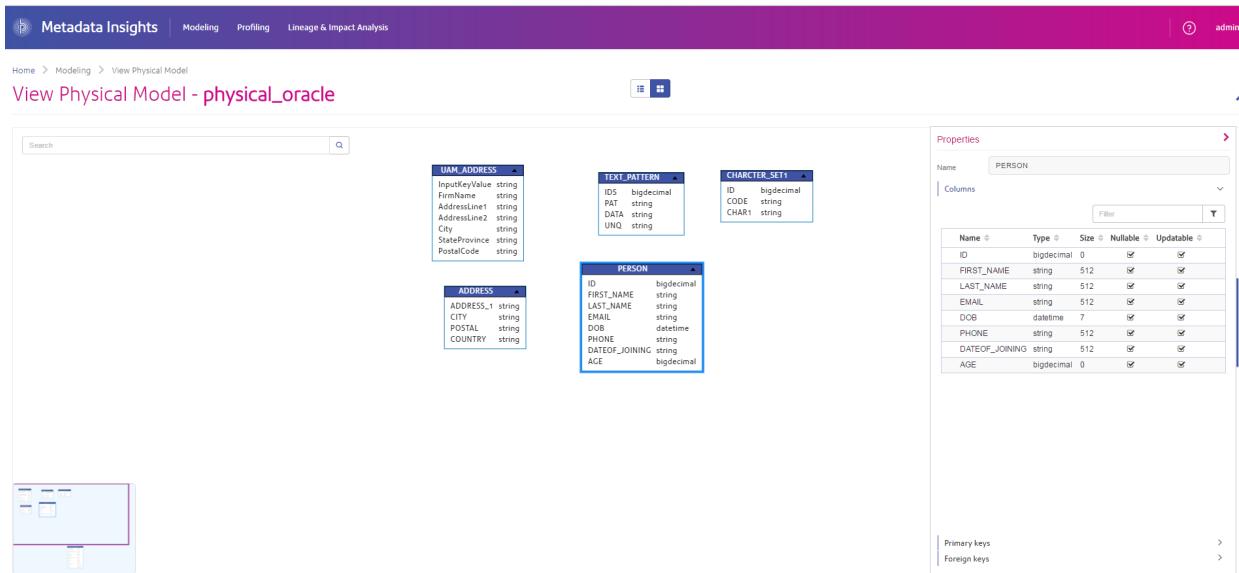
Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

Metadata Insights functions are divided into these areas: modeling, profiling, and lineage and impact analysis.

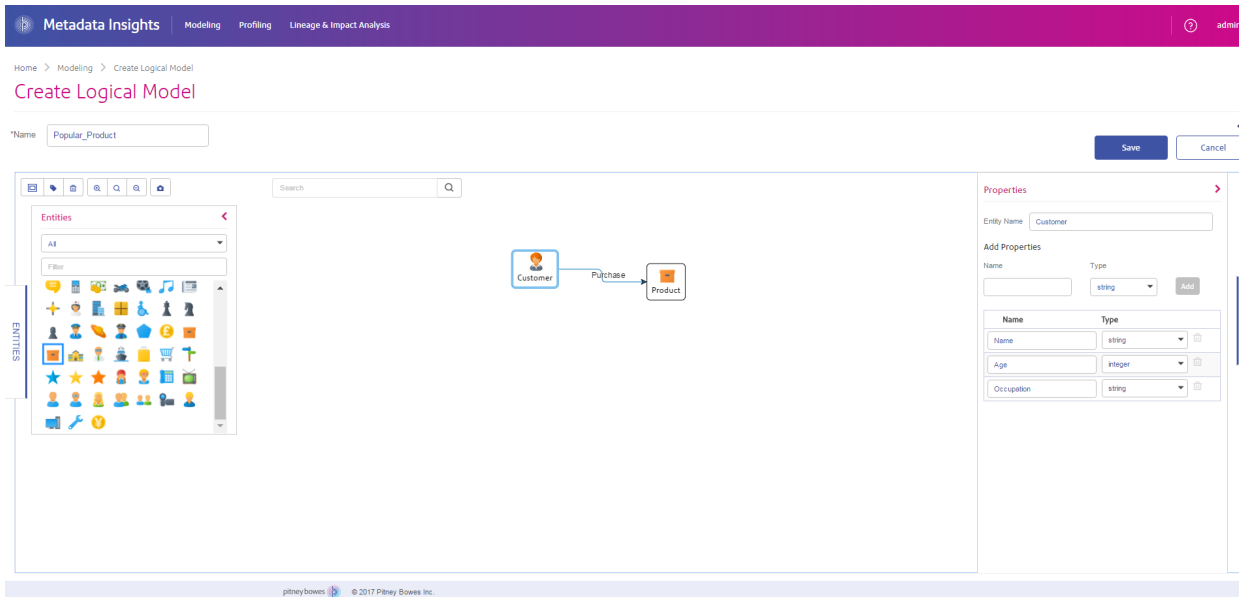
Modeling

The Modeling view is where you create physical and logical data models and deploy those into a model store, thus creating a layer of abstraction over the underlying data sources on the Spectrum™ Technology Platform server.

A *physical model* organizes your organization's data assets in a meaningful way. A physical model makes it possible to pull data from individual tables, columns, and views to create a single resource that you can then use to supply data to logical models or to perform profiling.



A *logical model* defines the objects that your business is interested in and the attributes of those objects, as well as how objects are related to each other. For example, a logical model for a customer might contain attributes for name and date of birth. It might also have a relationship to a home address object, which contains attributes for address lines, city, and postal code. Once you have defined the attributes of the objects your business is interested in, you can map physical data sources to the logical model's attributes, thereby identifying the specific data asset that will be used to populate that attribute.

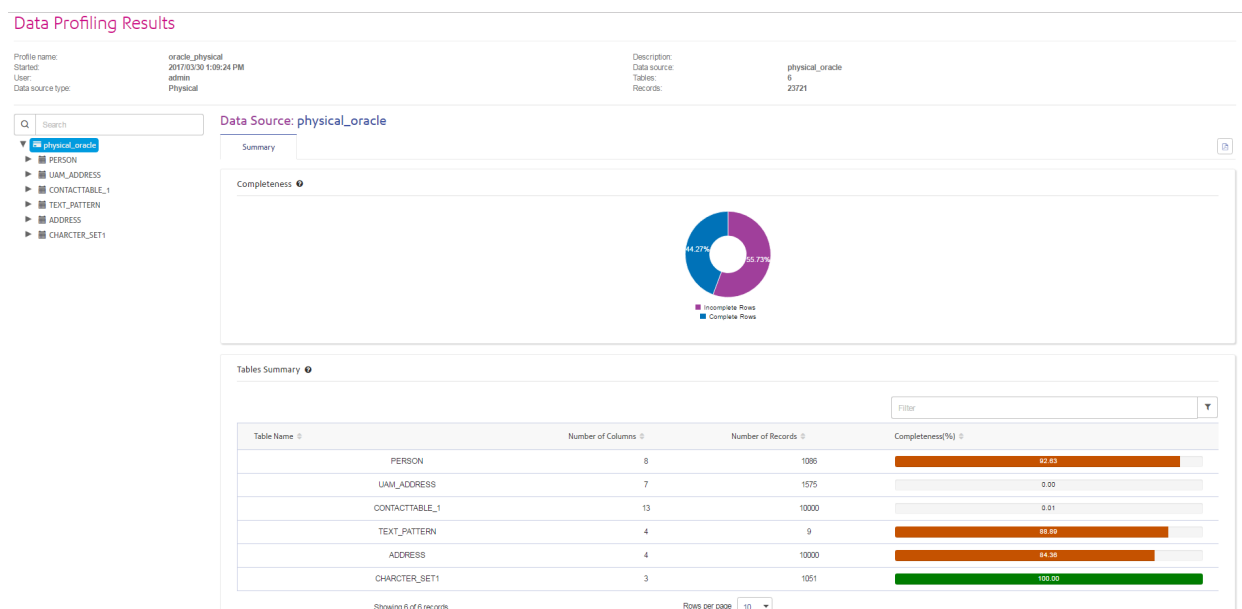


Profiling

Making informed business decisions requires quality data. So, it is important for you to have confidence in the completeness, correctness, and validity of your data. Incomplete records, malformed fields, and a lack of context can result in misleading or inaccurate data being delivered to your business users, which can result in flawed decisions.

Data profiling can help you be confident in your data. Profiling scans your data and generates reports that identify problems related to correctness, completeness, and validity. With these reports, you can take actions to fix incorrect or malformed data.

Metadata Insights provides profiling tools to run profiling on your data assets, as well as the data feeding into the logical and physical models defined in Metadata Insights. Using this information, you can determine the reliability of your data, design data quality rules, and perform standardization and normalization routines to fix data quality issues.



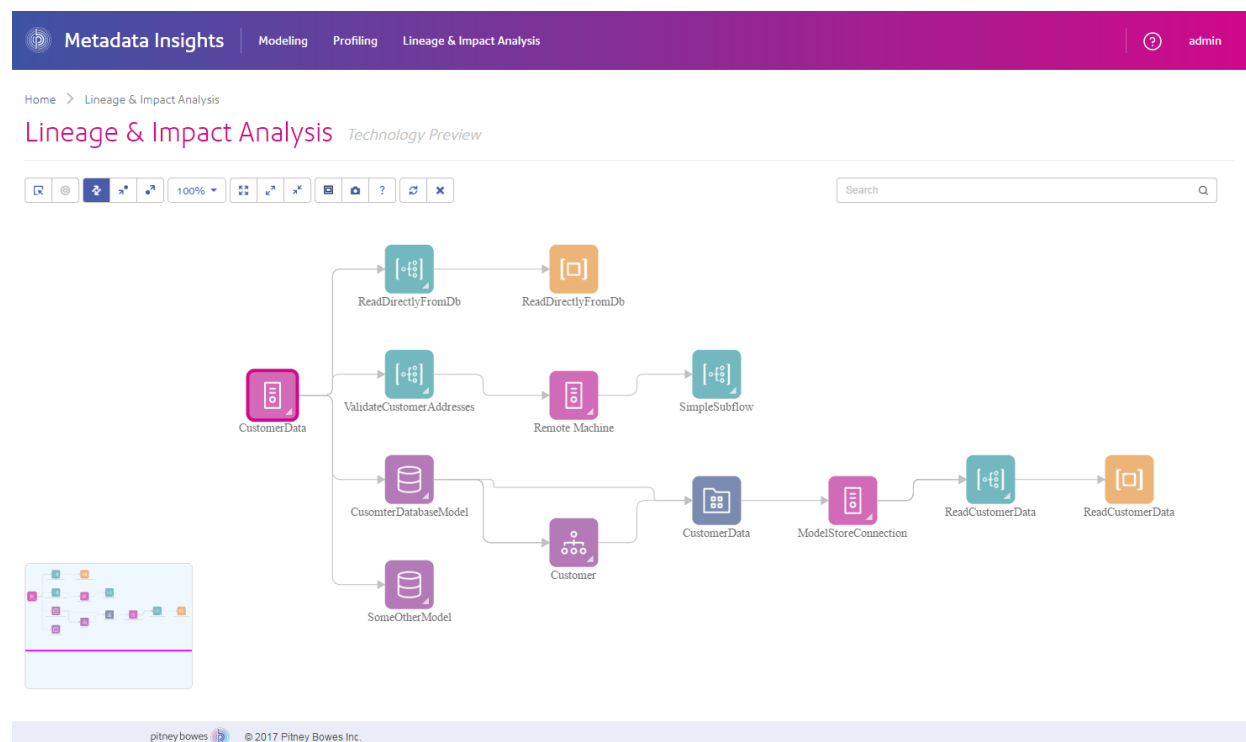
Lineage and Impact Analysis

The Lineage and Impact Analysis view shows how data flows from data sources to data destinations and through Spectrum™ Technology Platform flows. Lineage and impact analysis are similar concepts that describe different ways of tracing the flow of data.

Lineage shows where data comes from. You can use it to trace the path of data back to its source, showing all the systems that process and store the data along the way, such as Spectrum™ Technology Platform flows, databases, and files.

Impact analysis shows where data goes and the systems that depend on data from a selected data resource. You can use it to view the flows, databases, and files that use a data resource directly or indirectly. Looking at impact analysis is useful if you want to understand how a modification to a database, file, or flow will affect the processes and systems that use the data.

Metadata Insights can show lineage and impact analysis in a single diagram that shows the complete flow of data from source to destination. You can also choose to view lineage only or impact only. By viewing data lineage and impact analysis together you can pinpoint issues in your data processes and plan for upgrades and modifications to your data processes.



Connecting to Data

Introduction to Data Sources

A data source is a database, file server, cloud service, or other source of data that you want to process through Spectrum™ Technology Platform. Spectrum™ Technology Platform can connect to several types of data sources, including:

- Databases
- File servers
- Hadoop
- Cloud services from providers such as Amazon and Google

To connect Spectrum™ Technology Platform to a data source, use Management Console to define the connection. For example, if you want to read data from an XML file into a dataflow, and the XML file is located on a remote file server, you would have to define a connection to the file server before you can define the input XML file in a dataflow. Similarly, if you want to write dataflow output to a database, you must first define the database as an external resource.

Note: If you want to read from or write to data located in a file on the Spectrum™ Technology Platform server itself there is no need to define a connection.

Compression Support for Cloud File Servers

The file servers Amazon S3, Google cloud storage, and MS Azure Blobstore support the compressed formats `gzip (.gz)` and `zip (.zip)`.

The Spectrum™ Technology Platform handles the compression and decompression of the files written to and read from the file servers.

Note: You can use the same file server to handle both normal reads and writes of files as well as compression and decompression of files.

Reading a Compressed Format File

While reading a file from the server, its compression format is derived from the metadata key property `Content-Encoding` received from the server.

Writing a Compressed Format File

While writing a file to a server, mention the required compression format: `.gz` or `.zip`. The file is compressed based on the specified compression extension.

The metadata key property `Content-Encoding` is also set according to the selected compression format. This property value is passed to the cloud file server while writing the file to it.

Supported Entities and Operations

Spectrum™ Technology Platform supports these entities and operations for each connection type:

Connection Type Table Properties

Marketo

The entities are of the following types:

1. Entity
2. Entity Update

Note: Entity Update is a virtual table used for Update on Lead entity. For example, **Merge_Leads** should be used for merging different Marketo Leads.

Connection Type Table Properties

Netsuite

The entities are of the types:

- Standard Records
- Custom Records
- Joins
- Saved Searches

On viewing the schema of the created physical model, for each entity type, the resultant schema is displayed.

For example, for a Saved Search record, the schema of the search result is displayed. For a Join record, the schema of the join's result is displayed.

Note: In a NetSuite connection table, the primary key column is `internalId`.

SAP NetWeaver

The entity columns are of two types:

- **Native:** Columns with native data types are displayed with their respective data types.
- **Custom-defined:** Columns with custom-defined data types are displayed with a blank data type.

To deploy a model store derived from a SAP connection, ensure its logical and physical models include only such entities whose columns are of native data types. If the physical and logical models have entities of custom-defined data types, the model store cannot be deployed.

SAP OData Services supports a limited sets of operators listed here:

<http://www.odata.org/documentation/odata-version-2-0/uri-conventions/>. Example: `IsNull` or `Is not Null` operators are not supported.

Siebel

The business components are displayed in the format `Business Object.Business Component`.

Connection Type Table Properties

| | | | |
|--------|-----------------------------|---|--|
| Splunk | Supported Operations | LIKE, ORDER BY, LIMIT, IN, BETWEEN, !=, <=, >=, <, >, multiple AND/OR operators | |
| | Supported Functions | String Functions | upper, lower, length, len, ltrim, rtrim, substring, max, min |
| | | Mathematical Functions | abs, ceil, exp, floor, sqrt, round |

Note: For all other query operations, use the Splunk `search` column as explained below.

Metadata Insights provides a column `search` in the Splunk table using which you can look up the required data in the Splunk connection.

While executing a `select` query on the `SplunkTable`, use the `search` column in the `where` clause in any of the below scenarios:

1. To include such search criteria which cannot be specified using ANSI SQL syntax.
2. To include such Splunk-specific search criteria which cannot be included as part of the main SQL query.

For example, the below query looks for such a `_raw` value which contains the key `opp` with the value `ACC`.

```
select "_raw" from SplunkTable where "search"='search opp=ACC'
```

Connection Type Table Properties

| | |
|----------------|--|
| SuccessFactors | <p>The entities are of two types:</p> <ol style="list-style-type: none"> Entity: Represents a table representing a business entity. Join: Represents a mapping between any two <i>Entity</i> type tables: a parent table and any of its child tables. <p>Note: Links are not present between tables in the physical model schema derived from a SuccessFactors connection. This is because foreign keys are not present in SuccessFactors tables, and joins between tables are indicated by <i>Join</i> type tables in the Spectrum™ Technology Platform.</p> |
|----------------|--|

The features of Join tables are:

- The name of a *Join* table indicates the two *Entity* tables which have been mapped together.
- Each record of a *Join* table contains the primary key from the parent entity and the columns of the respective child entity. Thus the particular parent entity's primary key is mapped to the child entity's details.

For example, *User#HR* is a *Join* table in which *User* is the parent entity and *Hr* is the child entity. This join represents all the users and their respective HR representatives. The join table *User#HR*, therefore, has the parent table *User*'s primary key *UserId*, which is mapped to the columns of the child table *HR*, like *hr_userId*, *hr_username*, *hr_email*, and so on.

- In case of *Join* tables, the *insert* and *update* functions work like the *upsert* function. This is because *Join* tables are not actual entities in SuccessFactors, but are mappings between entities and their navigation properties or child tables.

To *insert/update* any *Join* table, the parent entity is updated, while in child table a new record is inserted or the existing record is updated corresponding to the parent record.

Note: While updating, the mapping between a parent and a child is modified. It is also possible to modify the individual attributes of the child as required.

| | |
|----------|---|
| SugarCRM | <p>Supported operations:</p> <p><code>LIKE</code> (its operation is limited to picking up options starting with the specified value, such as in the statement <code>WHERE name LIKE 's%'</code>, which picks up all the names starting with the alphabet S.), <code>IS NULL</code>, <code>IS NOT NULL</code>, <code>IN</code>, <code>NOT IN</code>, <code>></code>, <code>>=</code>, <code><</code>, <code><=</code>, <code>=</code>, <code><></code>, <code>AND</code>, <code>OR</code></p> |
|----------|---|


Connection Types

Connecting to Amazon

Connecting to Amazon DynamoDB

In order for Spectrum™ Technology Platform to access data in Amazon DynamoDB you must define a connection to Amazon DynamoDB using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Amazon DynamoDB.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.


5. In the **Type** field, choose **Amazon DynamoDB**.
6. In the **Access Key ID** field, enter the 20-character alpha-numeric sequence provided to you to access your Amazon AWS account.
7. In the **Secret Access Key** field, enter the 40-character key needed to authenticate the connection.
8. In the **Region** field, select the region of the Amazon AWS account.
9. To test the connection, click **Test**.
10. Click **Save**.

Amazon DynamoDB Limitations

1. Hierarchical data types like lists, sets and maps are interpreted as String data types. This is because these data types are not supported.
2. Null values in a DynamoDB data source are interpreted as empty column values.

Connecting to Amazon S3

In order for Spectrum™ Technology Platform to access data in Amazon S3 you must define a connection to Amazon S3 using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Amazon S3.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Cloud**.
6. In the **Cloud service** field, choose **AmazonS3**.
7. In the **Bucket name** field, enter the bucket name as defined in your Amazon S3 cloud service. This is the bucket where Spectrum™ Technology Platform will read and write files.
8. Enter your access key and secret key assigned to you by Amazon..
9. In the **Storage Type**, field select the level of redundancy that you want to allow for data storage.

Standard

The default level of redundancy provided by Amazon S3.

Reduced redundancy Stores non-critical and easily-reproducible data at lower levels of redundancy. This provides fairly reliable storage at a lower cost.

10. In the **Encryption** section, select the encryption method for the data. You can select server side encryption, client side encryption, or both.

Server side key The data is encrypted and decrypted at the server side. Your data is transmitted in plain text to the Amazon cloud service where it is encrypted and stored. On retrieval, the data is decrypted by the Amazon cloud service then transmitted in plain text to your system.

You have two options for specifying the key:

- **AWS managed:** The key is automatically generated by the Amazon S3 cloud service.
- **Customer provided:** Enter the key to be used by the Amazon S3 cloud service to encrypt and decrypt the data on the server side.

Client side key The data is encrypted and decrypted at the client side. The data is encrypted locally on your client system then transmitted to the Amazon S3 cloud storage. On retrieval, the data is transmitted back in an encrypted format to your system and is decrypted on the client system.

Client side key: Enter the key to be used by your client system to encrypt and decrypt the data.

If you select both **Server side key** and **Client side key**, encryption and decryption is performed at both server and client sides. Data is first encrypted with your client side key and transmitted in an encrypted format to Amazon, where it is again encrypted with the server side key and stored. On retrieval, Amazon first decrypts the data with the server side key, transmitting the data in an encrypted format to your system, where it is finally decrypted with the client side key.

Note: To use the encryption feature of Amazon S3 cloud, you need to install the Amazon S3 Security JAR files. For more information, see [Using Amazon S3 Cloud Encryption](#) on page 15.

For more information about Amazon S3 encryption features, see:

docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html

11. If you want to set access permissions, in the **Permissions** section, click .

The three kinds of Grantees are:

| | |
|---------------------------|---|
| Everyone | Every one else other than Authenticated Users and Log Delivery group. |
| AuthenticatedUsers | For users who are logged into Amazon. |
| LogDelivery | For users who write activity logs in a user-specified Bucket, if Bucket Logging is enabled. |

For each Grantee, select the desired permissions:

| | |
|----------------------|---|
| Open/Download | Allow the user to download the file. |
| View | Allow the user to view the current permissions on the file. |
| Edit | Allow the user to modify and set the permissions on the file. |

12 To test the connection, click **Test**.

13 Click **Save**.

Using Amazon S3 Cloud Encryption

To use the encryption security feature of the Amazon S3 cloud service, you need to download security JAR files and place them on the Spectrum™ Technology Platform server. Using encryption is optional.

1. Go to the download site.

For Windows and Linux platforms using Java 7, the JAR files can be downloaded from:

<http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html>

For AIX platforms using Java 7, the JAR files can be downloaded from:

<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk>

2. Download these two JAR files:

- local_policy.jar
- US_export_policy.jar

3. Place the JAR files in the location:

```
SpectrumFolder\Pitney Bowes\Spectrum\java64\jre\lib\security
```

4. Restart the server.

Connecting to Amazon SimpleDB

In order for Spectrum™ Technology Platform to access data in Amazon SimpleDB you must define a connection to Amazon SimpleDB using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Amazon SimpleDB.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.

2. Go to **Resources > Data Sources**.

3. Click the Add button .

4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Amazon SimpleDB**.
6. In the **Access Key ID** field, enter the 20-character alpha-numeric sequence provided to you to access your Amazon AWS account.
7. In the **Secret Access Key** field, enter the 40-character key needed to authenticate the connection.
8. To test the connection, click **Test**.
9. Click **Save**.

Amazon SimpleDB Limitations


In the Write to DB stage, the write mode **Update** is not available when writing to an Amazon SimpleDB table. The **Insert** option handles both insert and update operations. It differentiates between an insert and an update using the unique value of the `ItemName` column which is present in all Amazon SimpleDB tables.

This limitation is due to the fact that an update query requires a Primary Key for each record of the table to be updated, which is not supported by Amazon SimpleDB databases.

Connecting to Apache Cassandra

In order for Spectrum™ Technology Platform to access data in a Cassandra database, you must define a connection to the Cassandra database using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, the Cassandra database.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.


Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Apache Cassandra**.
6. In the **Host** field, enter the machine name or the IP on which the Apache Cassandra database is installed.
7. In the **Keyspace** field, enter the name of the keyspace of the data center you wish to access.
8. In the **Port** field, enter the port on which the Apache Cassandra database is configured.
9. Enter the user name and password to use to authenticate to the Cassandra database.
10. In the **Consistency Level** field, select how consistent data rows must be across replica nodes for a successful data transaction. This can be at least one, or all, or a combination of available nodes.

11. In the **Fetch Size**, enter the number of resultset rows you wish to fetch on each read transaction.
12. To test the connection, click **Test**.
13. Click **Save**.

Connecting to Azure Cloud

In order for Spectrum™ Technology Platform to access data in Microsoft Azure, you must define a connection to Microsoft Azure using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Microsoft Azure.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.


5. In the **Type** field, choose **Cloud**.
6. In the **Cloud service** field, choose **AzureBlobStorage**.
7. In the **Protocol** field select whether you want the connection between Azure and Spectrum™ Technology Platform to use HTTP or HTTPS.
8. In the **Account Name** field, enter the name of your Azure storage account.
9. In the **Access Key** field, enter the access key to your Azure account.
10. To test the cloud connection, click **Test**.
11. Click **Save**.

Connecting to a Flat File

Connecting to a Delimited Flat File

To add a new Delimited Flat File connection navigate to **Connections > Flat File**, and select the **Record Type** as **Delimited**. Enter the file's access details and content type details to allow the Data Federation Module to read it correctly.

Note: This connection is for use in the Metadata Insights module.

1. Go to **Connections > Flat File**.
2. By default, the screen opens in the create mode. Otherwise, click  to add a new Flat File connection.
3. Enter a **Connection Name** for the Flat File data connection.
4. Enter the **File Path** by clicking **Browse** and selecting the directory of the file.
5. Select the **Character Encoding** of the flat file from the drop-down.
6. Select the **Record Type** as **Delimited**.


7. In **Field Delimiter**, select the expected separator between any two fields of a file record.
8. Select the **Text Qualifier (optional)**, if any, that encloses the field values of a file record.
9. In **Line Separator**, the value `Default` is selected, indicating that the expected line separator depends on whether Spectrum™ Technology Platform is running on a Unix or Windows system.
10. To specify whether the first row of the file is a header row, shift the **First Row is Header Row** slider to either **Yes** or **No**.
11. To specify whether the data type of the various fields in any record of the file should be automatically detected, shift the **Detect data type from file** slider to either **Yes** or **No**.
12. To skip malformed records during file parsing, shift the **Skip Malformed Records** slider to **On**.
13. Click **Test**.
A message confirms the successful test of the connection.
14. Click **Save**.
A message confirms the successful creation of the connection.

In order to view a sample record fetched using the created Delimited Flat File connection, click **Preview** in the header bar. File records will be fetched and the Fields sorted according to the details provided by you.

Connecting to a Fixed Width Flat File

To add a new Fixed Width Flat File connection navigate to **Connections > Flat File** and select the **Record Type** as **Fixed Width**. Enter the file's access details and content type details to allow the Data Federation Module to read it correctly.

Note: This connection is for use in the Metadata Insights module.

1. Go to **Connections > Flat File**.
2. By default, the screen opens in the create mode. Otherwise, click  to add a new Flat File connection.
3. Enter a **Connection Name** for the Flat File data connection.
4. Enter the **File Path** by clicking **Browse** and selecting the directory of the file.
5. Select the **Character Encoding** of the flat file from the drop-down.
6. Select the **Record Type** as **Fixed Width**.
7. In the **Record Length** field, enter the total number of characters in a file record.
Repeat Step 8 to Step 13 to enter details of all fields expected in a file record.
8. Click **Add Field** to add a row for a field in a file record.
9. In the **Name** column, enter the name for the field value.
10. In the **Type** column, select the data type of the field value.
11. In the **Start Position** column, enter the position in the file record at which of the field value begins.
For the first field in a file record, the **Start Position** counting begins from 1.
12. In the **Length** field, enter the total number of characters the field covers, including the character at the **Start Position**.

The sum of the **Start Position** and **Length** values for any field should be less than or equal to the **Record Length**

If the File Record is:

```
01234Rob Smith29PitneyBowes
```

Record Length = 27

For the field 'Name':

Start Position = 6

Length = 9

```
Name = Rob Smith
```

13. Check the **Trim** checkbox if you wish to trim any white spaces at the beginning and/or end of a field value.
14. Click **Test**.
A message confirms the successful test of the connection.
15. Click **Save**.
A message confirms the successful creation of the connection.

In order to view a sample record fetched using the created Fixed Width Flat File connection, click **Preview** in the header bar. File records will be fetched and the Fields sorted according to the details provided by you.

Date Time Formats in a File Connection

While reading date and time values from files using a File Connection in the Metadata Insights Module, the values need to adhere to certain specific date-time formats.

Accepted Date Time Formats

- Date: "yyyy-mm-dd"
- Datetime: "yyyy-mm-dd HH:mm:ss"
- Time: "HH:mm:ss"

These are as per standard date-time notations.

Delimited Files

If the **Detect type** feature is turned on while configuring the Delimited File Connection, then the date and time values in the file records, which adhere to the above formats, are automatically detected as Date type.

If a date-time value does not adhere to one of the accepted formats, the value is read as a String type value instead of a Date type value.

Fixed Width Files

For Fixed Width files, date type values are configured while creating the Fixed Width File Connection. Hence these values are read as Date type values, irrespective of whether they adhere to the accepted formats or not.

If the date-time value in a Fixed Width file does not adhere to the accepted formats, it needs to be handled using **Transformations** at the logical model creation stage by applying the below *Conversion* category function to the value:

```
parsedate(String date, String format)
```

In this, the *date* is the value received from the file, while the *format* is the date-time format in which the value is received from the file. This helps to parse the date-time value correctly.

For example, if the *date* = 23-Feb-2008, then the *format* = dd-MMM-yyyy.

Resulting Value Formats

While previewing data in a model store:


- If the value has been read as a date/time value, it is reflected in one of the accepted date/time formats in the preview.
- If the value has been read as a String value, it is reflected as it is in the preview.

Connecting to an FTP Server

In order for Spectrum™ Technology Platform to access files on an FTP server you must define a connection to the FTP server using Management Console. Once you do this, you can create dataflows in Enterprise Designer that can read data from, and write data to, files on the FTP server.

Before connecting to an FTP server, verify that the timeout settings on your FTP server are appropriate for the jobs that will use this connection. Depending on the design of a job, there may be periods of time when the connection is idle, which may cause the connection to time out. For example, you may have a dataflow with two Read from File stages connected to an Import To Hub stage. While the Import To Hub stage is reading records from one Read from File stage, the other will be idle, possibly causing its connection to the FTP server to time out. Consider setting the timeout value on your FTP server to 0 to prevent connections from timing out.

Note: The FTP server must be running in active connection mode. Passive connection mode is not supported.


1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **FTP**.
6. In the **User name** and **Password** fields, enter the credentials to use to authenticate to the FTP server. This is required only if the FTP server requires it.
7. In the **Host** field, enter the hostname or IP address of the FTP server.
8. In the **Port** field, enter the network port number the server uses for FTP.
9. Click **Test** to verify that the Spectrum™ Technology Platform server can connect to the FTP server.
10. Click **Save**.

Connecting to Google Cloud Storage

In order for Spectrum™ Technology Platform to access data in Google Cloud Storage, you must define a connection to Google Cloud Storage using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Google Cloud Storage.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Cloud**.
6. In the **Cloud service** field, choose **GoogleCloudStorage**.
7. In the **Bucket name** field, enter the bucket name as defined in your Google cloud service. This is the bucket where Spectrum™ Technology Platform will read and write files.
8. Enter your the application name, service account, and private key file provided by Google.

Note: Ensure the private key file is present on the Spectrum™ Technology Platform server.

9. You can set access permissions in the **Permissions** section.

| | |
|--|---|
| Manage your data and permission | Allows the user to manage the data and permissions. |
| View your data | Allows the user to view data. |
| Manage your data | Allows the user to manage data. |


10. To test the connection, click **Test**.
11. Click **Save**.

For more information, see Google's [Service Account Authentication](#) documentation.

Connecting to Hadoop

In order for Spectrum™ Technology Platform to access data in Hadoop, you must define a connection to Hadoop using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Hadoop.

Attention: Spectrum™ Technology Platform does not support *Hadoop 2.x* for Kerberos on Windows platforms.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **HDFS**
6. In the **Host** field, enter the hostname or IP address of the NameNode in the HDFS cluster.
7. In the **Port** field, enter the network port number.
8. In **User**, select one of:

Server user Choose this option if authentication is enabled in your HDFS cluster. This option will use the user credentials that the Spectrum™ Technology Platform server runs under to authenticate to HDFS.

User name Choose this option if authentication is disabled in your HDFS cluster.

9. Check **Kerberos** if you wish to enable Kerberos authentication feature for this HDFS file server connection.
10. If you have opted to enable **Kerberos** authentication, then enter the path of the keytab file in the **Keytab file path** field.

Note: Ensure the key tab file is placed on the Spectrum™ Technology Platform server.

11. In the **Protocol** field, select one of:

WEBHDFS Select this option if the HDFS cluster is running HDFS 1.0 or later. This protocol supports both read and write operations.

HFTP Select this option if the HDFS cluster is running a version older than HDFS 1.0, or if your organization does not allow the WEBHDFS protocol. This protocol only supports the read operation.

HAR Select this option to access Hadoop archive files. If you choose this option, specify the path to the archive file in the **Path** field. This protocol only supports the read operation.

- 12 Expand the **Advanced options**.

13. If you selected the WEBHDFS protocol, you can specify these advanced options as required:

Replication factor Specifies how many data nodes to replicate each block to. For example, the default setting of 3 replicates each block to three different nodes in the cluster. The maximum replication factor is 1024.

Block size Specifies the size of each block. HDFS breaks up a file into blocks of the size you specify here. For example, if you specify the default 64 MB, each file is broken up into 64 MB blocks. Each block is then replicated to the number of nodes in the cluster specified in the **Replication factor** field.

File permissions Specifies the level of access to files written to the HDFS cluster by Spectrum™ Technology Platform. You can specify read and write permissions for each of these options:

Note: The *Execute* permission is not applicable to Spectrum™ Technology Platform.

User This is the user specified above, either **Server user** or the user specified in the **User name** field.

Group This refers to any group of which the user is a member. For example, if the user is john123, then Group permissions apply to any group of which john123 is a member.

Other This refers to any other users as well as groups of which the specified user is not a member.

14. In the grid below the **File permissions** table, specify the server properties for Hadoop to ensure that the sorting and filtering features work as desired when the connection is used in a stage or activity.

To add a new property, click .

Define the properties, as described in this table, depending on the stage or activity that will use the Hadoop connection, and whether *Hadoop 1.x* or *Hadoop 2.x* is being used.

Stage or Activity using the
HDFS Connection

Required Server Properties

- Stage **Read from Sequence File**
- Activity **Run Hadoop Pig**

Stage or Activity using the HDFS Connection

Required Server Properties

Hadoop 1.x Parameters

fs.default.name

Specifies the node and port on which Hadoop runs.

For example,
`hdfs://152.144.226.224:9000`

mapred.job.tracker

Specifies the hostname or IP address, and port on which the MapReduce job tracker runs. If the host name is entered as local, then jobs are run as a single map and reduce task.

For example, `152.144.226.224:9001`

dfs.namenode.name.dir

Specifies where on the local files system a DFS name node should store the name table. If this is a comma-delimited list of directories, then the name table is replicated in all of the directories, for redundancy.

For example,
`file:/home/hduser/Data/namenode`

dfs.datanode.data.dir

Specifies where on the local file system a DFS data node should store its blocks. If this is a comma-delimited list of directories, then data will be stored in all the named directories that are usually on different devices. Directories that do not exist are ignored.

For example,
`file:/home/hduser/Data/datanode`

hadoop.tmp.dir

Specifies the base location for other temporary directories.

For example, `/home/hduser/Data/tmp`

Hadoop 2.x Parameters

fs.defaultFS

Stage or Activity using the
HDFS Connection

Required Server Properties

Specifies the node and port on which Hadoop runs.

For example,
`hdfs://152.144.226.224:9000.`

Note: It is recommended that the parameter name `fs.defaultFS` be used Spectrum™ Technology Platform 11 SP1 onwards.

yarn.resourcemanager.resource-tracker.address

Specifies the hostname or IP-address of the Resource Manager.

For example, `152.144.226.224:8025`

yarn.resourcemanager.scheduler.address

Specifies the address of the Scheduler Interface.

For example, `152.144.226.224:8030`

yarn.resourcemanager.address

Specifies the address of the Applications Manager interface that is contained in the Resource Manager.

For example, `152.144.226.224:8041`

mapreduce.jobhistory.address

Specifies the host name or IP address, and port on which the MapReduce Job History Server is running.

For example, `152.144.226.224:10020`

Stage or Activity using the HDFS Connection

Required Server Properties

mapreduce.application.classpath

Specifies the CLASSPATH for Map Reduce applications. This CLASSPATH denotes the location where classes related to Map Reduce applications are found.

Note: The entries should be comma separated.

For example

```
$HADOOP_CONF_DIR,
$HADOOP_COMMON_HOME/share/hadoop/common/*,
$HADOOP_COMMON_HOME/share/hadoop/common/lib/*,
$HADOOP_HDFS_HOME/share/hadoop/hdfs/*,
$HADOOP_HDFS_HOME/share/hadoop/hdfs/lib/*,
$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*,
$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*,
$HADOOP_YARN_HOME/share/hadoop/yarn/*,
$HADOOP_YARN_HOME/share/hadoop/yarn/lib/*
```

mapreduce.app-submission.cross-platform

Handles various platform issues that arise if your Spectrum server runs on a Windows machine, and you install Cloudera on it. If your Spectrum server and Cloudera are running on different Operating Systems, then enter the value of this parameter as `true`. Else, mark it as `false`.

Note: Cloudera does not support Windows clients. Configuring this parameter is a workaround and not a solution to all resulting platform issues.

If you have checked the **Kerberos** checkbox above, then add the below Kerberos configuration properties additionally:

| Stage or Activity using the HDFS Connection | Required Server Properties |
|---|--|
| | <p data-bbox="613 352 1045 384">hadoop.security.authentication</p> <p data-bbox="797 401 1321 464">The type of authentication security being used. Enter the value <code>kerberos</code>.</p> <p data-bbox="613 489 1057 520">yarn.resourcemanager.principal</p> <p data-bbox="797 533 1338 638">The Kerberos principal being used for the resource manager for your Hadoop YARN resource negotiator.</p> <p data-bbox="797 659 1344 690">For example, <code>yarn/_HOST@HADOOP.COM</code></p> <p data-bbox="613 716 1073 747">dfs.namenode.kerberos.principal</p> <p data-bbox="797 760 1344 865">The Kerberos principal being used for the namenode of your Hadoop Distributed File System (HDFS).</p> <p data-bbox="797 886 1344 917">For example, <code>hdfs/_HOST@HADOOP.COM</code></p> <p data-bbox="613 942 1057 974">dfs.datanode.kerberos.principal</p> <p data-bbox="797 987 1338 1092">The Kerberos principal being used for the datanode of your Hadoop Distributed File System (HDFS).</p> <p data-bbox="797 1113 1344 1144">For example, <code>hdfs/_HOST@HADOOP.COM</code></p> |

| Stage or Activity using the HDFS Connection | Required Server Properties |
|--|--|
| <ul style="list-style-type: none"> • Stage Read from File • Stage Write to File • Stage Read from Hive ORC File • Stage Write to Hive ORC File | <p><i>Hadoop 1.x Parameters</i></p> <p>fs.default.name</p> <p>Specifies the node and port on which Hadoop runs.</p> <p>For example, <code>hdfs://152.144.226.224:9000</code></p> |
| | <p><i>Hadoop 2.x Parameters</i></p> <p>fs.defaultFS</p> <p>Specifies the node and port on which Hadoop runs.</p> <p>For example, <code>hdfs://152.144.226.224:9000.</code></p> <p>Note: It is recommended that the parameter name <code>fs.defaultFS</code> be used Spectrum™ Technology Platform 11 SP1 onwards.</p> |

15. To test the connection, click **Test**.

16. Click **Save**.


After you have defined a connection to an HDFS cluster, it becomes available in source and sink stages in Enterprise Designer, such as Read from File and Write to File. You can select the HDFS cluster when you click **Remote Machine** when defining a file in a source or sink stage.

Compression Support for Hadoop

Spectrum™ Technology Platform supports the compression formats `gzip (.gz)` and `bzip2 (.bz2)` on Hadoop. While using the **Read from File** and **Write to File** stages with an HDFS connection, include the extension corresponding to the required compression format (`.gz` or `.bz2`) in the **File name** field. The file is decompressed or compressed based on the specified compression extension. Spectrum™ Technology Platform handles the compression and decompression of the files.

Connecting to a JDBC Database

In order for Spectrum™ Technology Platform to access data from a JDBC database, you must define a database using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to a JDBC database.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.


5. In the **Type** field, select type of database you want to connect to.

Spectrum™ Technology Platform comes with JDBC drivers for SQL Server and Oracle databases. If you want to connect to a different database type, you must add the JDBC driver before defining a connection.

6. In the **URL** field, enter the JDBC connection URL. Your database administrator can provide this URL.

For example, to connect to a MySQL database named "SampleDatabase" hosted on a server named "MyServer" you would enter:

```
jdbc:mysql://MyServer/SampleDatabase
```

7. There may be additional fields you need to fill in depending on the JDBC driver. The fields represent properties in the connection string for the JDBC driver you selected in the **Type** field. See the JDBC driver provider's documentation or your database administrator for information about the specific connection properties and values required by the connection type.
8. Click **Save**.
9. Test the connection by checking the box next to the new connection and clicking the Test button .

Importing a JDBC Driver

Spectrum™ Technology Platform can access data from any database using a JDBC driver. Drivers for SQL and Oracle are provided with Spectrum™ Technology Platform, and some modules include drivers for other types of databases. If Spectrum™ Technology Platform does not come with a driver for the type of database you need, you can add a JDBC driver.

In this procedure you will import a JDBC driver by copying the driver files to the Spectrum™ Technology Platform server. When you complete this procedure the driver will be available to use when defining a JDBC database connection in Management Console.

Note: This procedure works for JDBC 4.x drivers. If the driver you want to add uses an older version of JDBC you must add the driver manually in Management Console. For more information, see [Manually Adding a JDBC Driver](#) on page 31

1. Put all the JDBC driver files for the database into a folder named:

```
Name.jdbc
```

Where *Name* is any name you want. The folder name must end with `.jdbc`.

2. Log in to the server running Spectrum™ Technology Platform.
3. Copy the folder containing the driver to this folder:

```
Spectrum Location\server\app\drivers
```

The driver is automatically imported.

4. To verify that the driver was successfully imported, log in to Management Console and go to **System > Drivers**. The driver should be listed.


If the driver is not listed, open the System Log in Management Console and look for errors related to deploying JDBC drivers.

Manually Adding a JDBC Driver

Spectrum™ Technology Platform can access data from any database using a JDBC driver. Drivers for SQL and Oracle are provided with Spectrum™ Technology Platform, and some modules include drivers for other types of databases. If Spectrum™ Technology Platform does not come with a driver for the type of database you need, you can add a JDBC driver.

In this procedure you will add JDBC driver files to the server then manually define the connection string and connection properties. Before you begin, be sure that you understand the connection string format and properties required by the driver. You must define these accurately in order for the driver to function. You can typically find information about a driver's connection string and properties from the driver provider's website.

Note: We recommend that you use this procedure only when adding a JDBC driver that uses JDBC 1.x, 2.x, or 3.x. If the driver uses JDBC 4.x, we recommend that you use the import method to add the driver. For more information, see [Importing a JDBC Driver](#) on page 30.

1. Open Management Console.
2. Go to **System > Drivers**.
3. Click the Add button .
4. In the **Name** field, enter a name for the driver. The name can be anything you choose.
5. In the **JDBC driver class name** field, enter the Java class name of the driver. You can typically find the class name in your JDBC driver's documentation.

For example, to use the Microsoft JDBC driver, you might enter the following:

```
com.microsoft.sqlserver.jdbc.SQLServerDriver
```

6. In the **Connection string template** field, enter the JDBC connection URL to use to connect to the database, including any properties you want to set in the connection string. Different database vendors use different connection strings so check your database's documentation for more information about the connection string.

If the driver will be used by more than one database connection, consider using property tokens in the connection string instead of hard-coding property values that may be different for each connection. For example, if you want to have some connections use encryption and others not, you may want to define a property token for the encryption property.

To use a property token in the connection string, use this syntax:

```
${PropertyToken}
```

Any property tokens you include in the connection string template will be required fields when defining a database connection.

For example, this connection string for SQL contains property tokens for host, port, instance, and encryption:

```
jdbc:sqlserver://${host}:${port};databaseName=${instance};encrypt=${encryption};  
TrustServerCertificate=true
```

These tokens are required fields when defining a database connection that uses this driver:

Add Data Source

*Name

Connection


*Type

*Host

*Port

*Instance

*encryption

7. If there are properties that you want to make optional for database connections, define them in the **Connection Properties** section.
 - a) In the **Connection properties** section, click the Add button .
 - b) In the **Label** field, enter a user-friendly description of the property. The label you enter here is used as the field label in the connections window when creating a connection using this driver.
 - c) In the **Property token** field, enter the token for the optional property. See the database driver's documentation for the properties supported by the driver.


For example, if you want to make encryption optional for database connections that use this driver, you could define the encryption property like this:

[Home](#) / [System: Drivers](#) / [Edit Driver](#)

Edit Deployed Driver

*Name

com.mysql.jdbc.Driver.5.1


*JDBC driver class name 

com.mysql.jdbc.Driver

*Connection string template 

jdbc:mysql://\${host}/\${instance}

Properties & Drivers

Connection properties 



| Label | Property Token |
|-----------------------------------|----------------|
| <input type="checkbox"/> username | user |
| <input type="checkbox"/> password | password |
| <input type="checkbox"/> Use SSL | useSSL |

When a database connection uses this driver, the encryption property would be displayed as an optional property in the database connection:

[Home](#) / [Data Sources](#) / [Add Data Source](#)

Add Data Source

*Name

Connection

*Type


*Host

*Instance

User Name


Password

Use SSL

8. Log in to the server running Spectrum™ Technology Platform and place the database driver file in a folder on the server. The location does not matter.
9. In the **Driver files** section, click the Add button .
10. In the **File path** field, enter the path to the database driver file on the server.
11. Click **Save**.

Editing a JDBC Driver

You can edit any JDBC driver that you have added to Spectrum™ Technology Platform. You cannot edit or delete any of the drivers that come with Spectrum™ Technology Platform, including the drivers for SQL Server, Oracle, and any drivers included with modules.

1. Open Management Console.
2. Go to **System > Drivers**.
3. Select the driver you want to modify then click the Edit button .
4. In the **JDBC driver class name** field, enter the Java class name of the driver. You can typically find the class name in your JDBC driver's documentation.

For example, to use the Microsoft JDBC driver, you might enter this:

```
com.microsoft.sqlserver.jdbc.SQLServerDriver
```

Note: You cannot edit the class name if you imported the driver files by copying them to the *Spectrum Location\server\app\drivers* folder.

5. In the **Connection string template** field, enter the JDBC connection URL to use to connect to the database, including any properties you want to set in the connection string. Different database vendors use different connection strings so check your database's documentation for more information about the connection string.

If the driver will be used by more than one database connection, consider using property tokens in the connection string instead of hard-coding property values that may be different for each connection. For example, if you want to have some connections use encryption and others not, you may want to define a property token for the encryption property.

To use a property token in the connection string, use this syntax:

```
${PropertyToken}
```

Any property tokens you include in the connection string template will be required fields when defining a database connection.

For example, this connection string for SQL contains property tokens for host, port, instance, and encryption:

```
jdbc:sqlserver://${host}:${port};databaseName=${instance};encrypt=${encryption};TrustServerCertificate=true
```

These tokens are required fields when defining a database connection that uses this driver:

Add Data Source

*Name

Connection


*Type

*Host

*Port

*Instance

*encryption

6. If there are properties that you want to make optional for database connections, define them in the **Connection Properties** section.
 - a) In the **Connection properties** section, click the Add button .
 - b) In the **Label** field, enter a user-friendly description of the property. The label you enter here is used as the field label in the connections window when creating a connection using this driver.
 - c) In the **Property token** field, enter the token for the optional property. See the database driver's documentation for the properties supported by the driver.


For example, if you want to make encryption optional for database connections that use this driver, you could define the encryption property like this:

[Home](#) / [System: Drivers](#) / [Edit Driver](#)

Edit Deployed Driver

*Name

com.mysql.jdbc.Driver.5.1

*JDBC driver class name 

com.mysql.jdbc.Driver

*Connection string template 

jdbc:mysql://\${host}/\${instance}

Properties & Drivers

Connection properties 



| Label | Property Token |
|-----------------------------------|----------------|
| <input type="checkbox"/> username | user |
| <input type="checkbox"/> password | password |
| <input type="checkbox"/> Use SSL | useSSL |

When a database connection uses this driver, the encryption property would be displayed as an optional property in the database connection:

[Home](#) / [Data Sources](#) / [Add Data Source](#)

Add Data Source

*Name

Connection

*Type

*Host

*Instance

User Name

Password

Use SSL

7. Click **Save**.

Deleting an Imported JDBC Driver

JDBC drivers cannot be deleted using Management Console if the JDBC driver was imported to Spectrum™ Technology Platform rather than being added manually in Management Console. Instead, follow this procedure to delete the driver.

Important: Before deleting a driver, verify that there are no database connections using the driver.

1. Stop the Spectrum™ Technology Platform server.
2. Go to this folder:

`Spectrum Location\server\app\drivers`

3. In the `drivers` folder, delete folder containing the driver.
4. Start the Spectrum™ Technology Platform server.
5. To verify that the driver has been deleted, log in to Management Console, go to **System > Drivers**, and verify that the driver is no longer listed.

Supported Database Data Types

Spectrum™ Technology Platform supports these data types commonly used in databases:

| | |
|-------------------|--|
| 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 6:15:00 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 -1.79769313486232E+308 to 1.79769313486232E+308. |
| 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 -3.402823E+38 to 3.402823E+38. |
| 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. |
| Raw | An Oracle datatype for storing variable length binary data. Maximum size is 2000 bytes (the maximum length in Oracle 7 was 255 bytes). |

Other database data types are automatically mapped to one of the supported data types as follows:

| Database Data Type | Supported Data Type |
|------------------------|---------------------|
| Date/Time Types | |
| TIMESTAMP | datetime |

| Database Data Type | Supported Data Type |
|----------------------|---------------------|
| String Types | |
| CHAR | string |
| CLOB | string |
| LONGVARCHAR | string |
| NCHAR | string |
| NVARCHAR | string |
| VARCHAR | string |
| Numeric Types | |
| BIGINT | long |
| DECIMAL | double |
| FLOAT | double |
| NUMERIC | bigdecimal |
| REAL | float |
| SMALLINT | integer |
| TINYINT | integer |
| Boolean Types | |
| BIT | boolean |

Supported Database Data Types for the Location Intelligence Module

These database data types are automatically mapped to one of the supported data types for the Location Intelligence Module.

| Database Data Type | Supported Data Type |
|--------------------|---------------------|
|--------------------|---------------------|

| | |
|-------------------|--|
| SQL Server | |
|-------------------|--|

| | |
|---------|---------------|
| tinyint | SHORT_INTEGER |
|---------|---------------|

| | |
|----------|---------------|
| smallint | SHORT_INTEGER |
|----------|---------------|

| | |
|-----|---------|
| int | INTEGER |
|-----|---------|

| | |
|--------|--------------|
| bigint | LONG_INTEGER |
|--------|--------------|

| | |
|-------|--------|
| float | DOUBLE |
|-------|--------|

| | |
|------|--------|
| real | DOUBLE |
|------|--------|

| | |
|----------------|--------|
| decimal(10, 5) | DOUBLE |
|----------------|--------|

| | |
|----------------|--------|
| numeric(10, 5) | DOUBLE |
|----------------|--------|

| | |
|------|------|
| date | DATE |
|------|------|

| | |
|------|------|
| time | TIME |
|------|------|

| | |
|----------|-----------|
| datetime | DATE_TIME |
|----------|-----------|

| | |
|---------------|-----------|
| smalldatetime | DATE_TIME |
|---------------|-----------|

| | |
|----------|--------|
| char(10) | STRING |
|----------|--------|

| | |
|-------------|--------|
| varchar(10) | STRING |
|-------------|--------|

| | |
|-----------|--------|
| nchar(10) | STRING |
|-----------|--------|

| | |
|---------------|--------|
| nvarchar(10)G | STRING |
|---------------|--------|

| | |
|------------|--------|
| binary(10) | BINARY |
|------------|--------|

| | |
|---------------|--------|
| varbinary(10) | BINARY |
|---------------|--------|

| Database Data Type | Supported Data Type |
|--------------------|---------------------|
|--------------------|---------------------|

| | |
|----------------|--|
| PostGIS | |
|----------------|--|

| | |
|----------|---------------|
| smallint | SHORT_INTEGER |
|----------|---------------|

| | |
|---------|---------|
| integer | INTEGER |
|---------|---------|

| | |
|--------|--------------|
| bigint | LONG_INTEGER |
|--------|--------------|

| | |
|----------------|--------|
| numeric(10, 5) | DOUBLE |
|----------------|--------|

| | |
|------|--------|
| real | DOUBLE |
|------|--------|

| | |
|------------------|--------|
| double precision | DOUBLE |
|------------------|--------|

| | |
|--------|---------|
| serial | INTEGER |
|--------|---------|

| | |
|-----------|--------------|
| bigserial | LONG_INTEGER |
|-----------|--------------|

| | |
|-------|--------|
| bytea | BINARY |
|-------|--------|

| | |
|------|------|
| date | DATE |
|------|------|

| | |
|------|------|
| time | TIME |
|------|------|

| | |
|-----------|-----------|
| timestamp | DATE_TIME |
|-----------|-----------|

| | |
|---------------|--------|
| character(10) | STRING |
|---------------|--------|

| | |
|-----------------------|--------|
| character varying(10) | STRING |
|-----------------------|--------|

| | |
|-----------|--------|
| nchar(10) | STRING |
|-----------|--------|

| | |
|---------------|--|
| Oracle | |
|---------------|--|

| | |
|--------|--------|
| NUMBER | DOUBLE |
|--------|--------|


| | |
|----------|--------|
| CHAR(10) | STRING |
|----------|--------|

| Database Data Type | Supported Data Type |
|--------------------|---------------------|
| VARCHAR(10) | STRING |
| VARCHAR2(10) | STRING |
| NCHAR(10) | STRING |
| NVARCHAR2(10) | STRING |
| DATE | DATE_TIME |
| TIMESTAMP | DATE_TIME |
| BLOB | BINARY |
| SAP HANA | |
| tinyint | SHORT_INTEGER |
| smallint | SHORT_INTEGER |
| integer | INTEGER |
| bigint | LONG_INTEGER |
| smalldecimal | DOUBLE |
| decimal(10, 5) | DOUBLE |
| real | DOUBLE |
| double | DOUBLE |
| float(30) | DOUBLE |
| varchar(30) | STRING |
| nchar(10) | STRING |

| Database Data Type | Supported Data Type |
|--------------------|---------------------|
| nvarchar(30) | STRING |
| alphanum(30) | STRING |
| date | DATE |
| time | TIME |
| seconddate | DATE_TIM |
| timestamp | DATE_TIM |
| varbinary(30) | BINARY |

Connecting to Knox

An Apache Knox Gateway allows you to access a Hadoop service through the Knox security layer. In order for Spectrum™ Technology Platform to access data in Hadoop via Knox, you must define a connection to Knox using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Hadoop via Knox.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose *Gateway*.
6. In the **Gateway Type** field, choose **Knox**.
7. In the **Host** field, enter the hostname or IP address of the node in the HDFS cluster running the gateway.
8. In the **Port** field, enter the port number for the Knox gateway.
9. In the **User Name** field, enter the user name for the Knox gateway.
10. In the **Password** field, enter the password to authorize you access to the Knox gateway.
11. In the **Gateway Name** field, enter the name of the Knox gateway you wish to access.
12. In the **Cluster Name** field, enter the name of the Hadoop cluster to be accessed.
13. In the **Protocol** field, choose *webhdfs*.
14. In the **Service Name** field, enter the name of the Hadoop service to be accessed.

15 To test the connection, click **Test**.


16 Click **Save**.

After you have defined a Knox connection to an HDFS cluster, the connection can be used in Enterprise Designer, in the stages **Read from File** and **Write to File**. You can select the HDFS cluster when you click **Remote Machine** when defining a file in a source or sink stage.

Connecting to Marketo

In order for Spectrum™ Technology Platform to access data in Marketo, you must define a connection to Marketo using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Marketo.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Marketo**.
6. In the **Endpoint URL** field, enter the endpoint URL of your Marketo account.

To find your endpoint URL, log in to your Marketo account and go to **Admin > Integration > Web Services**. The endpoint URL is under the heading **REST API** and is in this format:

```
https://AccountID.mktorest.com/rest
```

Copy the portion of the URL before `/rest`. For example, `https://AccountID.mktorest.com`.

7. Enter the client ID and secret key for your Marketo account.

To find your client ID and secret key, log in to your Marketo account and go to **Admin > Integration > LaunchPoint > API Rest > View Details**. The pop-up window displays the details.

8. To test the connection, click **Test**.
9. Click **Save**.

Marketo Limitations

1. This query applies only to `List` and `Activity_type` entities. For others, provide the filter type.

```
Select * from Marketo_Table
```


2. Marketo does not support joins except between `Lead` and `Lead_List` entities. The join query between `Lead` and `Lead_List` for a `List_Id` is as follows:

```
Select Lead.* from Lead Inner Join Lead_List
On Lead.ID = Lead_List.Lead_ID
And Lead_List.List_ID = <List ID>
```

Connecting to Microsoft Dynamics 365 Online

In order for Spectrum™ Technology Platform to access data in Microsoft Dynamics 365 Online, you must define a connection to Microsoft Dynamics 365 Online using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Microsoft Dynamics 365 Online.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Dynamics 365 Online**.
6. In the **Username** field, enter your Microsoft Dynamics user name.
7. In the **Password** field, enter your Microsoft Dynamics password.
8. In the **Organization Name** field, enter your organization unique name, which identifies your CRM instance.

To find your organization unique name, log in to Microsoft Dynamics and go to **Settings > Customization > Customizations > Developer Resources**. Your organization unique name is displayed.

9. In the **Region** field, select the geographical region of your Microsoft Dynamics account.
10. To test the connection, click **Test**.
11. Click **Save**.

Connecting to a Model Store

Connect to a model store to use the data federated from various sources such as databases, file servers, and cloud services. Once a connection is defined, you can use the data in the logical and physical models of a model store in the **Read from DB** and **Write to DB** stages of Enterprise Designer. For more information about model stores, see [Model Store](#) on page 86.

1. Open Management Console.

2. Go to **Resources > Data Sources**.

3. Click the Add button .

4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Model Store**.

6. In the **Model store** field, enter the name of the model store that you are establishing connection with.

To find the names of the available model stores, open Metadata Insights, to Modeling, and click the **Model Store** tab.

7. In the **Host** field, enter the hostname or IP address of the Spectrum™ Technology Platform server.

Note: You can leave the **Host** field blank if the server is running on localhost.

8. In the **User Name** field, enter the user name to access the server.

9. In the **Password** field, enter the password to authorize your access to the server.

10. To test the connection, click **Test**.

11. Click **Save**.

Connecting to NetSuite

In order for Spectrum™ Technology Platform to access data in NetSuite, you must define a connection to NetSuite using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, NetSuite. While reading from and writing to a NetSuite connection, both interactive and batch modes are supported.

Note: This connection is for use in the Metadata Insights module.

Spectrum™ Technology Platform supports these NetSuite entity types:

- Standard records
- Custom records
- Saved searches
- Joins between Standard records

To connect to NetSuite:

1. Open Management Console.

2. Go to **Resources > Data Sources**.

3. Click the Add button .

4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **NetSuite**.
6. In the **Email** field, enter the e-mail linked to the NetSuite account to be used for the connection.
7. In the **Password** field, enter the password of the NetSuite account.
8. In the **Account** field, enter the user name for the NetSuite account.
9. In the **Role** field, select the appropriate role for this connection from the roles mapped to the particular NetSuite user account.

The **Role** field is optional. If you leave the **Role** field blank, the default role is used to log in through the connection.

Attention: Only standard roles are supported. Custom roles are not supported.

10. To test the connection, click **Test**.
11. Click **Save**.

Note: To `INSERT` a record using a NetSuite connection, use an `UPSERT` query with the primary key (`internalId`) blank.

NetSuite Limitations

1. When querying using joins, you must cite specific columns. For example, this query is not supported:

```
select * from CUSTOMER_M
```

2. Simultaneous connections to NetSuite are not supported because NetSuite allows only a single login per account.
3. You can only write Standard and Custom records.
4. For both `UPDATE` and `UPSERT` queries, an `UPSERT` operation is performed.
5. In the Write to DB stage, the maximum batch size permitted for an `insert` operation is 200 and the maximum batch size for an `update` operation is 100.
- 6.


Connecting to NoSQL

In order for Spectrum™ Technology Platform to access data in a NoSQL database, you must define a connection to the NoSQL database using Management Console.

The supported NoSQL database types are:

1. Couchbase
2. MongoDB

Once the desired NoSQL connection is defined, you can create flows in Enterprise Designer that can read data from and write data to this database.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.


Note: Once you save a connection you cannot change the name.

5. In the **Type** field, select any one of:
 - Couchbase
 - MongoDB
6. Specify the **Host, Port, Database, Username** and **Password** of the specific NoSQL database you wish to access.
7. Click **Test** to check that the connection to the database is successful.
8. Click **OK**.

Connecting to Salesforce

In order for Spectrum™ Technology Platform to access data in Salesforce, you must define a connection to Salesforce using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Salesforce.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Salesforce**.
6. In the **Username** field, enter the email ID registered on the Salesforce data store.
7. In the **Password** field, enter a combination of the Salesforce portal password and the security token generated through the Salesforce portal.

For example, if your password is Sales@Test, and the security token provided to you by Salesforce is 56709367, then the Password to authenticate this Salesforce connection would be Sales@Test56709367.

8. To test the connection, click **Test**.

9. Click **Save**.

Note: Audit fields are enabled on all tables by default. The Salesforce audit fields are:


- created date
- last modified date
- created by
- last modified by

Attention: Physical model created in Spectrum™ Technology Platform version 10 and earlier using Salesforce connections need to be opened and saved again in order to enable audit fields on their tables.

Connecting to SAP NetWeaver

Creating a SAP NetWeaver connection in the Management Console using OData Services allows you to read, write and synchronize your CRM and ERP data. While reading from and writing to a SAP connection, both interactive and batch modes are supported.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **SAP**.
6. In the **Username** field, enter the username to access the SAP web service.
7. In the **Password** field, enter the password of the SAP web service.
8. In the OdataURL field, enter the address of the Odata web service to be used for this connection.
9. Click **Test**.
A message confirms the successful test of the connection.
10. Click **Save**.
A message confirms the successful creation of the connection.


Note: To perform fetch operations, an OData service must support the `$skip` and `$top` operations. If the service does not support these operations, the fetched records show inconsistencies in the model store preview.

SAP NetWeaver Limitations

For both `UPDATE` and `UPSERT` operations, an `UPDATE` operation is performed.

Connecting to SharePoint

In order for Spectrum™ Technology Platform to access data in SharePoint, you must define a connection to SharePoint using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, SharePoint.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Cloud**.
6. In the **Cloud service** field, choose **Sharepoint**.
7. In the **Version** field, select **v2010**. Spectrum™ Technology Platform currently supports SharePoint version 2010.
8. In the **Protocol** field, select the protocol required to connect SharePoint.
9. In the **Server address** field, enter the host name or IP address of the SharePoint server you want to connect to.
10. Enter the user name and password to use to authenticate to SharePoint.
11. In the **Project** field, enter the specific project whose SharePoint location you want to access.
12. To test the connection, click **Test**.
13. Click **Save**.

Example

For example, say you want to create a connection to this SharePoint URL:

```
https://sharepoint.example.com/sites/myportal
```


You would fill in the **Protocol**, **Server address**, and **Project** fields as follows:

- **Protocol:** https
- **Server address:** sharepoint.example.com
- **Project:** myportal

Connecting to Siebel

In order for Spectrum™ Technology Platform to access data in Siebel, you must define a connection to Siebel using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to Siebel.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Siebel**.
6. In the **Host** field, enter the host name of the server where Siebel is installed.
7. In the **Port** field, enter the listening port number for the SCBroker component.

Note: Use 2321 for SCBPort.

8. In the **Server** field, enter the name of your Siebel Enterprise Server.
9. In the **Object Manager** field, enter the name of the Object Manager.
10. Enter the user name and password to use to authenticate to Siebel.
11. In the **Locale** field, select the type of Object Manager and Language Pack for the Siebel Business application.

Note: For locales other than **English**, you must install locale-specific JAR files. For more information, see [Required JAR Files](#).

12. To test the connection, click **Test**.
13. Click **Save**.

Within a physical model, Siebel business components are displayed in the format `<Business Object>.<Business Component>`.

Note: Business Components that are not a part of any Business Object are not displayed in physical model.

Siebel Limitations

1. Siebel Business Components fields with the same names but differing cases are handled by leaving the first occurrence unchanged, while appending all subsequent occurrences with `_PB_` and an increasing numeral value.

For example, fields like `DeDup Token`, `Dedup Token`, and `DEdup Token` in a Siebel business component are renamed to `DeDup Token`, `Dedup Token_PB_1`, and `DEdup Token_PB_2` respectively.

2. If a field name contains a dot (.) in a Siebel schema, the dots are removed from the field name for use in Spectrum™ Technology Platform.

3. While creating a physical model from a Siebel connection, links between any two Siebel Business Components are displayed only if the link name matches the Business Component name in the Siebel schema.


For example, in a Siebel schema, if a link is defined between the Business Components `Account` and `Contact`, while another Business Component `Contact Custom` uses the links of `Custom`, then in the physical model created for this Siebel connection, no link is displayed between `Account` and `Contact Custom`. Links are displayed only between `Account` and `Contact`, as the original link is between these two in the Siebel schema.

4. Fields of a Business Component which are marked as inactive in the Siebel schema are not displayed in physical model, logical model, and model store.

Connecting to Splunk

In order for Spectrum™ Technology Platform to access data in Splunk, you must define a connection to Splunk using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, Splunk.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **Splunk**.
6. In the **Username** field, enter the Splunk account user name to authenticate the Splunk instance.
7. In the **Password** field, enter the password of the Splunk account.
8. In the **Host** field, enter the address or host name of the server on which the Splunk data source is hosted.
9. In the **Port** field, enter the port number of the Splunk data source.
10. To test the connection, click **Test**.
11. Click **Save**.

Splunk Limitations


This query is not supported:

```
select count(*) from SplunkTable
```

Connecting to SuccessFactors

In order for Spectrum™ Technology Platform to access data in SuccessFactors, you must define a connection to SuccessFactors using Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, SuccessFactors.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.
2. Go to **Resources > Data Sources**.
3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **SuccessFactors**.
6. In the **Company ID** field, enter your company ID, which identifies your company's unique instance in a specific SuccessFactors data center.
7. In the **Service URL** field, enter the URL for the SuccessFactors server you want to connect to. This URL is specific to the global data center to which your company ID is mapped.
8. Enter your user name and password for your SuccessFactors client instance.
9. To test the connection, click **Test**.
10. Click **Save**.

SuccessFactors Limitations

1. Batch operations can only be performed using `upsert` queries. So, `insert` and `update` queries are also performed as `upsert` queries in batch operations.
2. The table/column properties, as displayed in the physical model schema of a SuccessFactors connection, might not function as expected during use of the corresponding operation. For example, a column, which has been marked `updatable`, may throw a system exception when you attempt to update that column.

Connecting to SugarCRM

For Spectrum™ Technology Platform to access data in SugarCRM, you must define a connection to SugarCRM using the Management Console. Once you do this, you can create flows in Enterprise Designer that can read data from, and write data to, SugarCRM. Both on-line and on-premise versions of SugarCRM are supported.

Note: This connection is for use in the Metadata Insights module.

1. Open Management Console.
2. Go to **Resources > Data Sources**.

3. Click the Add button .
4. In the **Name** field, enter a name for the connection. The name can be anything you choose.

Note: Once you save a connection you cannot change the name.

5. In the **Type** field, choose **SugarCRM**.
6. Enter your SugarCRM user name and password.
7. In the **URL** field, enter the URL of the SugarCRM account to be used for this connection.
8. Enter the **Client Id** and **Client Secret** of your SugarCRM account.
9. To test the connection, click **Test**.
10. Click **Save**.

SugarCRM Limitations

1. For both `UPDATE` and `UPSERT` queries, an `UPSERT` operation is performed.
2. When writing records in batches, if a job takes more time than specified in the installer, you may face rollback exceptions. In such cases, you need to either decrease the batch size or increase the timeout limit.

Note: The timeout limit needs to be set as

```
com.atomikos.icatch.default_jta_timeout = 300000 in the
federation_transaction.properties file here: <Spectrum Install
Directory>\server\modules\datafederation\conf.
```

3. The **Nullable** and **Updatable** columns in the table properties, as displayed in the **physical model Schema** of the connection, may not represent the correct operation. For example, a column not marked as `Updatable` may throw system exception when you try to update it and conversely, a column marked as `Nullable` might not throw an exception in updating.
4. When querying using joins you need to use an alias.

Connecting to a Windows Mapped Drive

If you are running the Spectrum™ Technology Platform server on a Windows server you can provide Enterprise Designer and Management Console users with access to data stored on a mapped drive. For example, you may want to design dataflows that read data from, or write data to, files located on a drive mapped to the Windows server running Spectrum™ Technology Platform. Since the Spectrum™ Technology Platform server runs as a Windows service under a particular user account (often the Local System account) you need to define the mapped drive in the server's start-up process in order for it to be visible in Enterprise Designer and Management Console.

1. Stop the Spectrum™ Technology Platform server.
2. Under the folder where the Spectrum™ Technology Platform server is installed, go to `server\bin\wrapper`. For example, `C:\Program Files\Pitney Bowes\Spectrum\server\bin\wrapper`.
3. Open the file `wrapper.conf` in a text editor.

Important: In the following steps you will add new properties to this file. It is important that you follow these instructions precisely and only add and modify the properties described in the following steps. Do not modify any of the other properties in this file.

4. Add these lines:

```
wrapper.share.1.location  
wrapper.share.1.target  
wrapper.share.1.type  
wrapper.share.1.account  
wrapper.share.1.password
```

5. In the `wrapper.share.1.location` property, specify the location of the mapped drive in UNC format.

Note: Do not include a trailing backslash in the UNC.

For example,

```
wrapper.share.1.location=\\myserver\share
```

6. In the `wrapper.share.1.target` property, specify the drive letter to assign to this mapped drive.

For example,

```
wrapper.share.1.target=Y:
```

7. In the `type` property, specify `DISK`.

For example,

```
wrapper.share.1.type=DISK
```

8. If the share you are connecting to requires a user name and password, specify the user name in the `wrapper.share.1.account` property and specify the password in the `wrapper.share.1.password` property.

For example,

```
wrapper.share.1.account=domain\user123  
wrapper.share.1.password=mypassword1
```


Note: If the Spectrum™ Technology Platform server service is running under the Local System user, you cannot specify a user name and password. If the share requires a user name and password you must modify the service to run under a different account.

Example of Mapped Drive Properties in Wrapper.conf

The following example shows two mapped drives being defined in the `wrapper.conf` file.

```
wrapper.share.1.location=\\myserver\data
wrapper.share.1.target=Y:
wrapper.share.1.type=DISK
wrapper.share.1.account=sample\user
wrapper.share.1.password=samplepass
wrapper.share.2.location=\\myserver\moredata
wrapper.share.2.target=Z:
wrapper.share.2.type=DISK
wrapper.share.2.account=sample\user
wrapper.share.2.password=samplepass
```

Deleting a Connection

1. Open the Management Console.
2. Go to **Resources > Data Sources**.
3. Check the box next to the connection you want to delete, then click the Delete button .

2 - Modeling

Producing data-driven insights requires that technical teams and business teams have a common understanding of the organization's data assets and how those assets will be used to support business decisions. Technical teams understand the design of databases while business teams understand the business objects (such as customer, store, or vendor) that are of interest. Metadata Insights helps bridge this gap by providing physical modeling and logical modeling tools that are visually rich and independent of each other, enabling you to create both a technical view of data assets and a business view of objects of interest, and to link the two through mapping.

A *physical model* organizes your organization's data assets in a meaningful way. A physical model makes it possible to pull data from individual tables, columns, and views to create a single resource that you can then use to supply data to logical models or to perform profiling.

A *logical model* defines the objects that your business is interested in and the attributes of those objects, as well as how objects are related to each other. For example, a logical model for a customer might contain attributes for name and date of birth. It might also have a relationship to a home address object, which contains attributes for address lines, city, and postal code. Once you have defined the attributes of the objects your business is interested in, you can map physical data sources to the logical model's attributes, thereby identifying the specific data asset that will be used to populate that attribute.

Once the models are created, you can create a *model store* which is a virtual aggregation of the data referenced by physical and logical models. Model stores enable data federation from sources such as databases, file servers, and cloud services. When you create a model store, the data from the columns and tables that are used by the model is pulled and aggregated to the Spectrum™ Technology Platform.

In this section

| | |
|----------------|----|
| Logical Model | 61 |
| Physical Model | 63 |

Mapping
Model Store

71
86

Logical Model

A logical model is a representation of business-specific entities. It defines the objects that your business is interested in and the attributes of those objects, as well as how these objects are related to each other.


Creating a Logical Model


A logical model is created by adding business entities to it and defining relationships among the entities. You can also define the properties of the entities as well as the relationships. When complete, you can map the logical model to one or more physical models.

1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Log in using your credentials.
3. Click **Modeling**.
4. Click the Add Logical Model button .

Note: You can also choose to create a new model by copying an existing model and renaming it. To do this, on the **Models** tab, select the model and click on the **Copy Model**  icon. Rename the model on the **Create Logical Model** page as needed and click the **Save** button. For more information about modifying a physical model, see [Modifying a Logical Model](#) on page 62.

5. Enter a name for the model in the **Name** field.
6. From the **Entities** palette, drag the entity you want to add to the model and drop it to the canvas.


Tip: Use the drop-down list in the palette to sort entities on the basis of their categories. You can also use the type-ahead filter box below the drop-down list for this.

The entity is added to the canvas and its properties are displayed in the **Properties** panel on the right side of the page. To define the properties of an entity, see [Defining Properties of Entity and relationship](#) on page 62.

7. To create relationship between two entities, hover the cursor on the entity, drag the mouse and drop it on the entity to which you want to relate it.

The relationship between entities is indicated by a relationship link. When you hover the cursor on the primary entity, the related entity gets highlighted.

Note: Like an entity, you can add properties to the relationship by using the **Properties** panel. For more information, see [Defining Properties of Entity and relationship](#) on page 62.

8. To delete an entity or relationship, select it and click the Delete icon .


If you delete a relationship entity, the related relationship links also get deleted.

9. Click **Save**.

Defining Properties of Entity and relationship

The **Properties** dialog box displays default name of all the business entities and relationships. You can modify the name and add attributes to entities and relationships in a logical model using this dialog box.

1. Right-click on the entities or relations for which you want to define properties, and click the **Properties** option.
2. Use the **Add Properties** section to add attributes and their data types to the selected entity or relation. Enter the attribute name in the **Name** field and select its data type from the **Type** drop down list.
3. Click the **Add** button.
The defined attribute and data type are displayed below the entry fields.

Note: You can delete the attribute details by clicking the Delete icon  adjacent to it.

Modifying a Logical Model

You can modify details of a logical model using the **Edit Logical Model - <model name>** page.


1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Click **Modeling**.
3. Select the check box next to the model you want to modify.

Note: You can search for a specific model using the **Filter** field on the top of the page. It searches the models by the model name.

4. Click the Edit Model icon .
5. Modify details as needed.
6. Click **Save**.

The selected model is updated and listed in the **Models** tab of the **Modeling** page.

Deleting a Logical Model

If you delete a logical model which has a dependent physical model, the dependent physical model becomes unusable.


1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Click **Modeling**.
3. Select the check box next to the model you want to modify.

Note: You can search for a specific model using the **Filter** field on the top of the page. It searches the models by the model name.

4. Click the Delete Model button .
5. Click **Yes** to confirm.

Physical Model

A *physical model* organizes your organization's data assets in a meaningful way. A physical model makes it possible to pull data from individual tables, columns, and views to create a single resource that you can then use to supply data to logical models or to perform profiling.

Adding a Physical Data Model


Physical models are created by selecting the tables and columns you want to include from the data source. You can create multiple physical models that use data from the same data source, but each physical model can only have one data source. For example, a model for a system administrator


will include the system tables from the data source, while the model created for end users will have the required data tables from the same data source.

1. If you have not already done so, use Management Console to connect Spectrum™ Technology Platform to the data sources you want to model. For more information, see [Introduction to Data Sources](#) on page 8.
2. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

3. Log in using your credentials.
4. Click **Modeling**.
5. Click the **Add Physical Model** button .

Note: You can also choose to create a new model by copying an existing model and renaming it. To do this, on the **Models** tab, select the model and click on the **Copy Model**  icon. Rename the model on the **Copy Physical Model** page as needed and click the **Save** button. For more information about modifying a physical model, see [Modifying a Physical Model](#) on page 65.

6. In the **Name** field, enter a name for the model.
7. In the **Source connection type** field, choose the type of data source that you want to model, then select the specific data source in the **Source connection name** field.

The **Source connection name** field displays the data sources defined in Management Console for the connection type you select. For example, if you select Apache Cassandra in the **Source connection type** field, the **Source connection name** field will list all the Apache Cassandra data sources defined in Management Console.

Note: A physical model can have only one data source.

8. Click **Fetch Metadata**.
9. Select the data you want to include.

To select a table or view, check the box in the **Include** column for the table or view you want to include. To select all the tables listed in the page, select the **Include** check-box in the header row. When you select a table, all its columns are included in the model.

To select specific columns to include in the model, hover over the table containing the columns you want and click the **Select columns** button that appears.

You can search for tables on the basis of their name or type using the **Filter** field above the list of tables. For example, if you want to view only those tables that have names containing the word "new", enter `new` in the filter field. Similarly, you can filter out tables on the basis of their type. However, you can use only one of the filter criteria, either name or type.

10. Click Save.

The model is created and is listed on the **Models** tab of the **Modeling** page.

Modifying a Physical Model

1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.


2. Click Modeling.

3. Select the check box next to the model you want to modify.

Note: You can search for a specific model using the **Filter** field on the top of the page. It searches the models by the model name.

4. Click the Edit Model button .

The **Edit Physical Model** page is displayed for the selected model with the **Include** check-box displaying the already selected tables and columns.

Note: Physical models are displayed in two views: list and tabular. The list view displays the selected table as a list below the connection specification fields. The tabular view shows only the selected tables and their details. To use the tabular view, click the Tabular View icon  on the top of the page. For more information, see [Viewing a Physical Model in Tabular View](#) on page 65.

5. Modify the connection for the model, if required, using the **Source connection name drop down list.**

6. Remove the added table, or add a table, as required, by clicking the respective rows.

7. Click Save.

The selected model gets updated and listed on the **Models** tab of the **Modeling** page.

Viewing a Physical Model in Tabular View

The tabular view of a physical model displays details of all the selected data tables. Each table displays the attributes and its data type. Other properties of the tables are displayed in the **Properties** panel.

1. In a web browser, go to:


```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Click **Modeling**.
3. Select the check box next to the model you want to view.

Note: You can search for a specific model using the **Filter** field on the top of the page. It searches the models by the model name.

4. Click the View Model icon .

Note: You can also go to the tabular view by clicking the Tabular View icon  on the **Edit Physical Model** page.

5. To view the properties of a table, select it, and click the **Properties** collapsible tab on the right side of the page.

Note: You cannot modify these properties.

The **Properties** panel displays these details:

Note: The details displayed here are based on the type of the selected data source. For more information, see [Table Properties](#) on page 67.

- **Columns:** Column details, such as the name of the column, the contained data type, its size and whether it is nullable and updatable.
- **Primary Keys:** Name of the primary key (if defined) and related column and key sequence
- **Foreign Keys:** Name of the foreign key (if defined) and related column and table name

Deleting a Physical Model

Follow this procedure to delete a physical model from Metadata Insights.

Note: If you delete a physical model that is mapped to a logical model, the logical model becomes unusable.


1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Click **Modeling**.
3. Select the check box next to the model you want to delete.

Note: You can search for a specific model using the **Filter** field on the top of the page. It searches the models by the model name.

4. Click the Delete Model button .
5. Click **Yes** to confirm.

Connection-specific Details

Table Properties

Table properties, such as columns, primary keys, and foreign keys displayed in the **Properties** collapsible tab on the **Edit Physical Model** page are based on the type of source connection selected in the physical model. The detail for various connection types are given below.

Note: For the steps to view the **Properties** of tables, see [Viewing a Physical Model in Tabular View](#) on page 65.

| | | |
|------------------------------------|---------------------|---|
| JDBC connection | Columns | Displays the name, data type, and size of each column. The section also displays whether a column is nullable and/or a primary key. |
| | Primary Keys | Displays the primary key, primary key name, and key sequence. |
| | Foreign Keys | Displays the foreign key column name, foreign key name, primary key catalog name, primary key table name, primary key column name, and primary key name. |
| | Indexes | Displays the index name, column name, and whether the indexes are unique or non unique. If an index is non unique then the NU check box is displayed as checked otherwise it is displayed as unchecked. |
| Amazon DynamoDB connection | Columns | Displays the name and data type of each column. The section also displays whether a column can be updated, searched, and/or is a primary key. |
| | Primary Keys | Displays the primary key and primary key name. |
| Amazon SimpleDB connection | Columns | Displays the column name and data type for the table you select. |
| Apache Cassandra connection | Columns | Displays the column name and data type. The section also displays whether a column can be updated, searched, and/or is a primary key. |
| | Primary Keys | Displays the primary key and primary key name. |
| Flat File connection | Columns | Displays the column name and data type for the table you select. |

Marketo connection

| | |
|---------------------|--|
| Columns | Displays the column name and data type. The section also displays whether a column is a primary key, readable or updatable. |
| Primary Key | Displays the primary key and primary key name. |
| Foreign Keys | Displays the foreign key column name, foreign key name, and the reference table name. |
| Permissions | Indicates which functions can be performed on the selected physical model table. The functions Read, Insert, Update, Upsert and Delete can be performed on Marketo tables. |
| Filter Types | Indicates the column on which search is applicable. For example, <i>id</i> as Filter Type in Lead indicates Lead can be searched with <i>id</i> . |

Note: Marketo does not support multiple Filter Types in `Select` query.

For example:

Valid Query:

```
Select * from Lead where id = 1234
```

```
Select * from Lead where email = 'abc@txyz.com'
```

Invalid Query:

```
Select * from Lead where id = 1234 and email = 'abc@txyz.com'
```

MS Dynamics 365 Online connection

| | |
|---------------------|--|
| Columns | Displays the column name and data type. The section also displays whether a column is nullable and/or a primary key. |
| Primary Keys | Displays the primary key and primary key name. |
| Foreign Keys | Displays the foreign key column name, foreign key name, and the reference table name. |
| Permissions | Indicates which functions can be performed on the selected physical model table. Currently, the functions create, read, write, and delete can be performed on an MS Dynamics 365 Online table. |

NetSuite connection

| | |
|---------------------|---|
| Columns | Displays the name, data type, and size of each column. The section also displays whether a column is nullable and/or a primary key, and which of the functions create, read, and update can be performed on the column. |
| Primary Keys | Displays the primary key and primary key name. |

| | | |
|----------------------------------|---------------------|--|
| Salesforce connection | Columns | Displays the name, data type, and size of each column. The section also displays whether a column is nullable and/or a primary key. |
| | Primary Keys | Displays the primary key and primary key name. |
| | Foreign Keys | Displays the foreign key column name, foreign key name, and the reference table name. |
| SAP NetWeaver connection | Columns | Displays the name, data type, and size of each column. The section also displays whether a column is nullable and/or a primary key. |
| | Primary Keys | Displays the primary key and primary key name. |
| | Foreign Keys | Displays the foreign key column name, foreign key name, and the reference table name. |
| | Permissions | Indicates which functions can be performed on the selected physical model table. Currently, the functions create, write, and delete can be performed on an SAP NetWeaver table. |
| Splunk Connection | Columns | Displays the name and data type of each column. The section also displays whether a column can be updated, searched, and/or a primary key. |
| | Primary Keys | Displays the primary key and primary key name. |
| SuccessFactors connection | Columns | Displays the name, data type, and size of each column. The section also indicates whether a column is nullable and/or a primary key, and which of the functions filter, insert, sort, update, upsert, and view can be performed on the column. |
| | Primary Keys | Displays the primary key columns and the respective primary key name of which each is a part. |
| | Foreign Keys | Displays the foreign key column name, foreign key name, and the reference table name. |
| | Permissions | Indicates which of the functions insert, update, upsert, and delete can be performed on the selected physical model table. |
| SugarCRM connection | Columns | Displays the name, type and size of each column. It also shows whether a column is nullable and updatable. |
| | Primary Keys | Displays the primary key names, column and key sequence. |
| | Foreign Keys | Displays the foreign key column name, foreign key name, and the reference table name. |
| Siebel connection | Columns | Displays the name, type and size of each column. The section also displays whether a column is nullable and updatable. |

| | |
|---------------------|---|
| Primary Keys | Displays the primary key names, column and key sequence. |
| Foreign Keys | Displays the foreign key column name, foreign key name, and the reference table name. |

Read and Write Limitations

Read limitations

- Metadata Insights does not interpret case-sensitive field names as different names. If two tables or column names use the same name but different cases, they are interpreted as duplicate names. If a model store contains such field names, the deployment of that model store fails.

For example, fields like `DeDup`, `Dedup`, and `DEdup` are interpreted as duplicate names. The deployment of a model store containing such case-differentiated fields fails.

- If a Data Source, physical model, logical model or model store has a dependency on a JDBC-specific data source, it cannot be imported or exported using the Administration Utility.

Write limitations

While writing to a database source using the **Write to DB** stage in **Enterprise Designer**, there are certain limitations.

- Creation of tables while writing to a database source using a Metadata Insights connection is not supported. If you click **Create Table** while using a Metadata Insights connection, a runtime exception is displayed.
- Metadata Insights does not support unified writes to multiple connections simultaneously. You can write to a single source at a time.
- Dropping and recreating any table while writing to a database using a Metadata Insights connection is not supported.
- Delete and truncate operations are not supported while writing to a database using Metadata Insights. Only insert and update operations are supported.
- Metadata Insights performs a commit after completion of each batch insert. The **Batch count to commit** cannot be changed to any other value other than 1.

In the **Runtime** tab of the **Write to DB** stage in **Enterprise Designer**, the **Batch count to commit** field can only have the value 1 for Metadata Insights.

Note: A batch consists of 1000 records by default. You can configure the batch size according to your system, in the **Runtime** tab of the **Write to DB** stage in **Enterprise Designer**.

- For batch inserts, the feature **Commit at the end** is not supported. This feature refers to performing a commit after all batches are complete. In Metadata Insights, a commit is performed at the end of each batch insert completion.

Mapping

Mapping Models

The logical models must be mapped to one or more physical data models to define the data sources to use to populate the entities in the logical model. This mapping can be done using one or a combination of these ways:

- Directly: Drag a table from the physical model to the canvas and establish a link between the columns of the physical and logical tables.
- Transformation: Transform the values before populating it to the logical model tables. The transformation can be application of a conversion function or functions of string numeric, and datetime to a logical model column. It may also involve applying a constant value to a column of a logical model table.
-

You can filter the data to be populated in the logical model entities:

- Building Criteria: Define a criteria for filtering the data retrieved from the physical model columns. The filtered data is fed to the logical model.
- Joining columns and tables: Combine data from columns of two or more physical model tables before it is populated in the logical model columns.



If you map columns of two different data types, the data type of the logical model column gets updated to that of the physical model column.

Mapping Logical Entities to Physical Data

1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Log in using your credentials.
3. Click **Modeling**.
4. Select the logical model you want to map and click the Map Model icon  in the tool bar or the mapping icon  in the **Mapped** column .


The color of the icon in the **Mapped** column indicates if it is mapped. Green indicates partial or complete mapping, while red indicates that the model is not yet mapped.

5. In the **Physical Model** panel, select the physical model you want to map or its table or column to which you want to map the entities of your logical model. There are a few ways to create the mapping:



- To map using drag-and-drop:
 1. From the list of models, click the required physical model. A list of all the tables contained in the model is displayed.
 2. Drag the table you want to the canvas. The table displays the columns and the data type of each column.
 3. Map a column to an attribute of a logical entity by dragging the mouse from the column to the attribute.
- To map using the search box (useful when you know only the table name and not the model to which it belongs):
 1. Enter the search term in the search field and from the adjacent drop-down list, select **Columns**, **Tables**, or **All**, based on whether the term needs to be only in columns or tables, or in columns, tables as well as the models.

The results are displayed below the search fields, showing the name of the column, data type, name of the table or view, and name of the physical model to which the table belongs.

2. If you searched for a table, click it, and click the **Add to Canvas** button. The selected table gets added to the canvas and you can map it using the drag-and-drop method described above.
3. If you searched for a column, click the column name. The **Entity** and **Attribute** fields appear. The **Entity** field lists all the entities in the currently mapped logical model, while the **Attribute** field lists all the attributes of the selected entity.

Select the entity and attribute you want to map, then click the Mapping icon .

Note: One attribute can be mapped to only one column at a time.

- To convert a value in the physical model column before mapping it to a logical entity, use transformation. For more information, see [Applying a Transformation](#) on page 74.
 - To filter data from a logical model, click the Criteria Builder icon . For more information, see [Criteria](#) on page 82.
 - To combine data from columns of two physical model tables before populating the logical model columns, see [Joins](#) on page 84.
6. To delete any of the mappings or tables on the canvas, select it and click the **Delete mapping** icon  in the tool bar.
7. Click **Save**.

The mapping is saved and the mapping icon changes from read to green for this logical model in the **Models** tab of the **Modeling** page.

Transformations

This feature gives you the flexibility of transforming a value before populating it to a column in a logical model entity. The values are sourced from the columns of physical model tables.

On the basis of the functions you need to apply on values before populating the logical model columns, you can do transformation on:

- **Columns:** To populate values from a column of a physical model table
- **Constant:** To apply a constant value to a logical model column or to one of the parameters of a function transformation
- **Function:** To apply functions of string, numeric, conversion, and datetime to a logical model column, or to one of the parameters of a function transformation.

You can create complex nested transformations within the parent transformation.

Conversion Function Transformation

A conversion function allows you to convert values from one data type into another data type, from one format into another format, or parse values according to desired formats.

A conversion function is applied to convert the data from a source table column data or a constant into the desired data type or format. This converted data populates the selected column of a logical entity.

The various conversion functions available are:

| | |
|-----------------|--|
| cast | This function allows you to convert values of one data type into another data type. |
| format | This function allows you convert a value of any data type into a string value using a provided format. |
| parse | This function allows you to parse a string value into the desired data type using a provided format. |
| to_bytes | This function allows you to convert a CLOB type value into bytes. |
| to_chars | This function allows you to convert a BLOB type value into chars. |

Parameters of Conversion Functions

The `cast`, `format`, and `parse` conversion functions have two parameters:

- Source value of a certain data type and format
- Target data type or format into which the source value needs to be converted, or using which the source value needs to be parsed.


Note: This convention is not applicable for `to_bytes` and `to_chars` conversion functions.

Applying a Transformation

1. On the **Create Mapping <logical model name>** page, select the entity corresponding to the attribute to which you want the transformed value applied, and click the **Transformations** collapsible tab on the right side of the page.

The **Transformations** panel is displayed showing the name of all the attributes in the selected entity. The **Transformation** column remains blank initially. It displays the transformation type after you save the transformation.

Note: The **Transformation** panel doesn't display details of the physical model table. In case you click a physical model table after an entity and open the **Transformation** panel, it displays details of the last selected entity.

2. Select the attribute to which you want the transformed value applied, and click the **Transformation** icon .

The **Transformation of column '<column name>'** pop-up window is displayed.


3. Select an option from **Column**, **Constant**, or **Function** to define the type of transformation to be applied on the values from the physical model tables.

| Option | Description |
|-----------------|--|
| Column | Derive the value from a physical model table column. For more information, see Applying a Column Transformation on page 74. |
| Constant | Define the resulting value as a constant. For more information, see Applying a Constant Transformation on page 75. |
| Function | Define the resulting value using String, Numeric, Conversion, and Datetime functions. For more information, see Applying a Function Transformation on page 76. |

4. Define the transformation using either one of the above three options, or a desired combination of any of the three.
5. Click **OK**.
The defined transformation is applied on the attribute and is displayed with an arrow on the canvas. The transformation type is also displayed in the **Transformations** panel of the **Create Mapping** page.


Applying a Column Transformation

A column transformation allows you to link the required column of a physical model table to define an attribute of a logical entity or one of the parameters of a function transformation. You can do a column transformation by using the **Transformations** panel as described below or by linking a physical model column with an attribute in a logical entity directly on the canvas.

1. On the **Create Mapping <logical model name>** page, select the entity corresponding to the attribute to which you want the transformed value applied, and click the **Transformations** collapsible tab on the right side of the page.
The **Transformations** panel is displayed showing the name of all the attributes in the selected entity. The **Transformation** column remains blank initially. It displays the transformation type after you save the transformation.
Note: The **Transformation** panel doesn't display details of the physical model table. In case you click a physical model table after an entity and open the **Transformation** panel, it displays details of the last selected entity.
2. Select the attribute to which you want the transformed value applied, and click the **Transformation** icon .
The **Transformation of column '<column name>'** pop-up window is displayed.
3. Select the **Column** option, and from the **Physical Model Name** drop-down list, select the required physical model.
4. From the **Table Name** drop-down list, select the required table.
5. From the **Column Name** drop-down list, select the required column that you want to map with the attribute.
6. Click **OK**.
The selected column gets mapped with the attribute and it is indicated by a connecting arrow between the two on the canvas. The **Transformation** panel displays this as a **Simple** transformation.
Note: The data type of the attribute is inherited from the mapped physical model column.

Applying a Constant Transformation

A constant transformation allows you to apply a constant value to define an attribute of a logical entity or one of the parameters of a function transformation.

1. On the **Create Mapping <logical model name>** page, select the entity corresponding to the attribute to which you want the transformed value applied, and click the **Transformations** collapsible tab on the right side of the page.
The **Transformations** panel is displayed showing the name of all the attributes in the selected entity. The **Transformation** column remains blank initially. It displays the transformation type after you save the transformation.
Note: The **Transformation** panel doesn't display details of the physical model table. In case you click a physical model table after an entity and open the **Transformation** panel, it displays details of the last selected entity.
2. Select the attribute to which you want the transformed value applied, and click the **Transformation** icon .
The **Transformation of column '<column name>'** pop-up window is displayed.


3. Select the **Constant** option, and from the **Type** drop-down list, select the data type of the constant value you want to define.
4. In the **Value** field, enter a constant value for the selected data type.
5. Click **OK**.

The applied constant transformation is reflected on the canvas with a transformation node  and it is indicated as **Constant** transformation in the **Transformation** panel.

Note: The **Transformation of column '<column name>'** pop-up window is displayed when you double-click this node.


Applying a Function Transformation

A function transformation allows you to apply functions of string, numeric, conversion, and datetime to define an attribute of a logical entity, or one of the parameters of a parent function transformation.

1. On the **Create Mapping <logical model name>** page, select the entity corresponding to the attribute to which you want the transformed value applied, and click the **Transformations** collapsible tab on the right side of the page.
The **Transformations** panel is displayed showing the name of all the attributes in the selected entity. The **Transformation** column remains blank initially. It displays the transformation type after you save the transformation.
2. Select the attribute to which you want the transformed value applied, and click the **Transformation** icon .
The **Transformation of column '<column name>'** pop-up window is displayed.
3. Select the **Function** option, and from the **Category** drop-down list, select the category for the function you want to apply. You can choose from **String**, **Numeric**, **Conversion**, and **Datetime**.
4. From the **Name** drop-down list, select the function you want to apply.

Note: The options in this drop-down depends on the function **Category** you select.

The function is defined and is displayed on the window below the entry fields.

Note: You can see the description of the function by hovering the mouse on  icon below the **Name** drop-down list.

5. Define the parameters of the selected function by expanding it and using one of these options: **Column**, **Constant**, or **Function**.

Note: Ensure that the resulting data type matches the expected datatype of the parameter in the function.

6. Click **OK**.

The applied function transformation is reflected on the canvas with a transformation node.

7. Click **Save**.

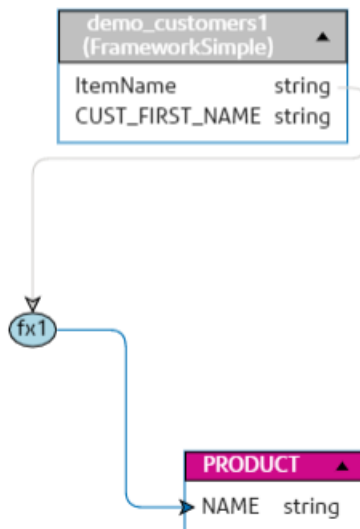
The mapping is saved and the mapping icon changes from read to green for this logical model in the **Models** tab of the **Modeling** page.

Example - Applying a Function Transformation to an attribute of a logical model entity

For example, if you wish to trim spaces from the right of each value of the column `ItemName` of the physical model table `demo_customers1` and fill in the trimmed value into the linked `Name` column of the logical table `Product`:

1. Select the `Product` table on the canvas.
2. In the **Transformations** panel, click Δ adjacent to the `Name` column, and in the **Transformation of column 'NAME'** pop-up window, select the **Function** option.
3. In the **Category** drop-down list, select `String`.
4. In the **Name** drop-down list, select the `rtrim(string string)` function.
5. The **String** function gets added to the pop-up below the entry fields. Click this function, and define the parameter on which this function needs to be applied by using the **Column** option such that the resulting data type of the parameter is a string.
6. Click **OK**.

The applied function transformation is depicted on the canvas as below:




Applying a Conversion Transformation

Note: These steps do not apply to the conversion functions `to_bytes` and `to_chars`.

1. On the **Create Mapping <logical model name>** page, select the entity corresponding to the attribute to which you want the transformed value applied, and click the **Transformations** collapsible tab on the right side of the page.

The **Transformations** panel is displayed showing the name of all the attributes in the selected entity. The **Transformation** column remains blank initially. It displays the transformation type after you save the transformation.

Note: The **Transformation** panel doesn't display details of the physical model table. In case you click a physical model table after an entity and open the **Transformation** panel, it displays details of the last selected entity.

2. Select the attribute to which you want the transformed value applied, and click the **Transformation** icon .

The **Transformation of column '<column name>'** pop-up window is displayed.

3. Select the **Function** option, and from the **Category** drop-down list, select the **Conversion** option.
4. In the **Name** field, select the required conversion function.

The function is defined as conversion and value and target get displayed below the entry fields.

- **Value:** The source value which needs to be converted.
- **Target:** The target data type or format into which the source value needs to be converted, or using which the source value needs to be parsed.

5. Select the **value** using one of these options: **Column**, **Constant**, or **Function**.

Note: Ensure the resulting data type matches the expected data type for the first parameter of the cast function (selected in the **Name** field above).

6. Select the **target** using one of these options: **Column**, **Constant**, or **Function**. For more information, see [Supported Datatypes for Conversion Transformations](#) on page 79.

7. Click **OK**.


The applied conversion function transformation is reflected on the canvas with a link and the function node.

8. Click **Save**.

The mapping is saved and the mapping icon changes from read to green for this logical model in the **Models** tab of the **Modeling** page.

Example - Applying a Cast Conversion Function Transformation

For example: To convert an integer value from `BALANCE` column of the `CUSTOMER` table of `ClientSheet` physical model into a float value to populate the `CUSTBALANCE` column of a `CUSTPROFILE` logical entity:

1. Select the `CUSTPROFILE` entity on the canvas.
 2. Click the **Transformations** panel and in the panel, click the **Transformation** icon  against the `CUSTBALANCE` column.
 3. In the popup, select the **Function** option.
 4. In the **Category** field, select `Conversion`.
 5. In the **Name** field, select the `cast(Integer value, string target)` function. You get option to set the value and target below the entry fields.
 6. Click **Value**, and select the **Column** option.
 - In the **physical model Name** drop-down, select the physical model `ClientSheet`.
 - In the **Table Name** drop-down, select a table `CUSTOMER`.
 - In the **Column Name** drop-down, select the column `BALANCE`.

The Integer value from this column becomes the first parameter of the cast function.
 7. Click **Target** and do these selections.

Note: The expected target data type is string in this example.

 - Select the **Constant** option.
 - From the **Type** drop-down, select the datatype `String`.
 - In the **Value** field, enter `Float`. This is the target datatype.

The target datatype `Float` becomes the second parameter of the cast function.
 8. Click **OK**.
- The applied cast conversion function transformation is depicted on the canvas.

Supported Datatypes for Conversion Transformations

Metadata Insights supports the datatypes mentioned in the table below while applying conversion transformations.

Table 1: Supported Datatypes for Conversion Transformations

| Type | Description |
|-------------------|---|
| String or Varchar | Variable length character string with a maximum length of 4000 |
| Varbinary | Variable length binary string with a nominal maximum length of 8192 |
| Char | A single Unicode character |

| Type | Description |
|-----------------------|---|
| Boolean | A single bit, or Boolean, with two possible values |
| Byte or Tinyint | Numeric, integral type, signed 8-bit |
| Short or Smallint | Numeric, integral type, signed 16-bit |
| Integer or Serial | Numeric, integral type, signed 32-bit. The serial type also implies not null and has an auto-incrementing value that starts at 1. Serial types are not automatically UNIQUE. |
| Long or Bigint | Numeric, integral type, signed 64-bit |
| Biginteger | Numeric, integral type, arbitrary precision of up to 1000 digits |
| Float or Real | Numeric, floating point type, 32-bit IEEE 754 floating-point numbers |
| Double | Numeric, floating point type, 64-bit IEEE 754 floating-point numbers |
| Bigdecimal or Decimal | Numeric, floating point type, arbitrary precision of up to 1000 digits |
| Date | Datetime, representing a single day (year, month, day) |
| Time | Datetime, representing a single time (hours, minutes, seconds, milliseconds) |
| Timestamp | Datetime, representing a single date and time (year, month, day, hours, minutes, seconds, milliseconds, nanoseconds) |


Editing a Transformation

For Column, Constant, and Function transformations, the editing can be done by clicking the Transformation icon in the **Transformations** panel for the relevant entity. Additionally, in case of Constant and Function transformations, you can double-click the Transformation node on the canvas and edit the values in the **Transformation of column '<column name>'** pop-up window that gets displayed.

1. To edit a transformation, select the related entity and click the **Transformations** collapsible tab on the right side of the page.

Note: In case of Constant and Function transformations, you can also double-click the Transformation node on the canvas, which takes you to the **Transformation of column '<column name>'** pop-up window for editing.

The **Transformations** panel is displayed showing the name of all the attributes in the entity and indicating the type of transformations applied. The **Transformation** column remains blank in case no transformation is applied on the attribute.

2. Select the attribute to which you want the transformed value applied, and click the **Transformation** icon .

In case of constant and function transformations, you can also double-click the transformation node on the canvas to edit it.


The **Transformation of column '<column name>'** pop-up window is displayed.

3. In the pop-up, select the column, constant, or function parameter that you wish to modify.
4. Select an option from **Column**, **Constant**, or **Function** to define the type of transformation to be applied on the values from the physical model tables.

| Option | Description |
|-----------------|--|
| Column | Derive the value from a physical model table column. For more information, see Applying a Column Transformation on page 74. |
| Constant | Define the resulting value as a constant. For more information, see Applying a Constant Transformation on page 75. |
| Function | Define the resulting value using String, Numeric, Conversion, and Datetime functions. For more information, see Applying a Function Transformation on page 76. |

The edited transformation is displayed on the canvas for the respective logical entity.

Deleting a Transformation

1. Delete a transformation using either of these steps:
 - Select the transformation function node or the transformation link on the canvas and press **Delete** on your keyboard.
 - In the **Transformations** panel of the logical entity, click  against the transformation you want to delete.
2. Click **Save**.

Criteria

Use Criteria Builder to define a criteria for filtering data retrieved from the physical model columns. The filtered data is fed to the logical model columns.

Note: This is an optional function.

Components of a Criterion

The criteria builder of a logical entity consists of a parent group, which comprises of a hierarchy of child criteria and child groups.

Group A group is defined using multiple criteria and child groups. You can create a hierarchy of child groups and criteria up to any desired level within the parent group.

For each group, you need to specify whether the data fetched from the source physical model tables needs to satisfy all, at least one, or none of the defined criteria of that group.

Criterion A criterion is defined using either one or two operands and a logical operator.

The selected logical operator either compares the left operand to the values defined in the right operand, or places a condition on the left operand.

Left Operand The column of the physical model table on which you want to define the criterion.

Right Operand The right operand is defined as any one of these:


- A constant
- A set of constants
- A column of any of the physical model tables
- A set of columns of any of the physical model tables

Operator The logical operator defines the condition to be fulfilled for the criterion. The operator can:

- Place a condition on the left operand.
- Place a condition comparing the left operand with the right operand.


Creating a Criterion



The **Criteria** section is displayed for the selected logical model entity only if at least one transformation is defined for it.

1. On the **Create Mapping <logical model name>** page, select the required logical entity, and click the **Criteria Builder** icon on the tool bar .

Note: The **Transformation** panel doesn't display details of the physical model table. In case you click a physical model table after an entity and open the **Transformation** panel, it displays details of the last selected entity.

The **Criteria Builder for logical table <name of the table>** pop-up window is displayed, with the **Group satisfying criterion** drop-down list.

Note: To add another group, hover the mouse on the group and click the **Add group** icon  that is displayed.

2. Select a grouping method for the criteria by clicking on **Group satisfying criterion** drop-down list. Specify whether it is necessary to fulfill all, at least one, or none of the criteria defined in the criteria group.
3. To add a child group to the parent group, hover the mouse on the parent group and click the **Add group** icon  that is displayed.
4. To add a criterion to a group, hover the mouse on the parent group, and click the **Add Comparison** icon . The added criterion shows below the group.
 - a) To define the left operand of the criterion, in the **Expression Builder**, click **<Column 1>** to select one of the physical table columns.
 - a) To select a logical operator from the provided list, in the **Expression Builder**, click **<comparison>**.
If the selected logical operator requires a right operand, a placeholder for the right operand is displayed to be defined.
 - b) If the right operand is required, click **<Column 2>**.

The right operand can be defined using either a constant or a physical model table column.

Constant Enter the constant, and click **Add constant**.

column(s) Select one of the available physical model columns.


Based on the number of operands the logical operator expects in the right operand, define the right operand as any of these:

- A constant
- A set of constants
- A column of any of the physical model tables
- A set of columns of any of the physical model tables



5. Repeat steps 3 and 4 as many times as is required to create the desired criteria.
6. Click **OK**.

All, any, or none of the defined set of child criteria need to hold true in order to retrieve the required data from the connected physical model tables into the target logical model columns. This is as defined for the parent criterion.

Editing a Criterion

1. Select the required logical entity, and click the **Criteria Builder** icon on the tool bar .
The **Criteria Builder for logical table <name of the table>** pop-up window is displayed, with the defined groups and criteria.
2. Select the required group, child group or criterion.
3. Make the required changes.
4. Click **OK**.

Deleting a Criterion

1. Select the required logical entity, and click the **Criteria Builder** icon on the tool bar .
The **Criteria Builder for logical table <name of the table>** pop-up window is displayed, with the defined groups and criteria.
2. Select the required group, child group or criterion.
3. Click the **Delete** icon .
4. Click **Save**.

Joins


Use the **Joins builder** to combine data from columns of two physical model tables before it is populated to the logical model columns.

Example: The `Name` column in the `Product` table of your logical model is linked to the `Product Name` column of the `Product_A` physical table. However, you want to pull in data from another physical table `Product_B`. In such a case, you can join the `Product Name` column of the `Product_A` physical table with the `Product Name` column of the `Product_B` physical table so that data populated in the logical model is a combination of data between the two physical tables (on the basis of the selected join type).

Note: You need to define one or more transformations on the logical table before using the Joins builder.

Note: You will be able to perform joins only if you have at least two physical model tables dragged to the canvas.

Creating a Join

1. On the **Create Mapping <logical model name>** page, select the required logical entity, and click the **Join builder** icon on the tool bar .

Note: The **Transformation** panel doesn't display details of the physical model table. In case you click a physical model table after an entity and open the **Transformation** panel, it displays details of the last selected entity.

The **Join Builder for logical table <name of the table>** pop-up window is displayed.


2. Click the **Add table** button.

A table is displayed with **Left table**, **Right table**, and **Join type** columns.

3. Use the **<select table>** and **<select column>** drop-down options of the **Left table** and **Right table** columns to specify the columns (of the physical model tables) from which data is to be joined.

4. From the **Join type** drop-down list, select the type of join you want between the columns. The options are:

- **Inner join:** Returns matching records between the selected left and the right table columns.
- **Left join:** Returns all records from the left column, in addition to any matching records from the right column.
- **Right join:** Returns all the records from the right column, in addition to any matching record from the left column.
- **Full outer join:** Returns all the records from the left and the right columns.

5. To establish joins on multiple columns of the table, hover cursor over the table row, and click the **Add comparison** icon  that appears. A new **<select column>** row is displayed.

6. Repeat steps 4 and 5 to define the join criteria.

7. To establish join between another set of tables, click the **Add table** button.

Note: You need to have the set of table dragged on to the canvas to be able to establish join between those.

A new set of **<Select table>** and **<Select column>** rows are displayed.


8. Repeat steps 3 and 4 to define the join.

9. Click **OK**.

The join between the tables is established and data is populated in the logical model column from the selected physical model columns on the basis of the defined join.

Note: Joins are not represented by any physical link between the tables on the canvas.

Editing a Join

1. On the **Create Mapping <logical model name>** page, select the required logical entity, and click the **Join builder** icon on the tool bar .

Note: The **Transformation** panel doesn't display details of the physical model table. In case you click a physical model table after an entity and open the **Transformation** panel, it displays details of the last selected entity.

The **Join Builder for logical table <name of the table>** pop-up window is displayed.

2. Modify your selection, as needed, and click the **OK** button.

Note: For details on using the fields for modification, see [Creating a Join](#) on page 84.

The join definitions are modified for the selected tables and columns.


Note: Joins are not represented by any physical link between the tables on the canvas.

Deleting a Join

1. On the **Create Mapping <logical model name>** page, select the required logical entity, and click the **Join builder** icon on the tool bar .

Note: The **Transformation** panel doesn't display details of the physical model table. In case you click a physical model table after an entity and open the **Transformation** panel, it displays details of the last selected entity.

The **Join Builder for logical table <name of the table>** pop-up window is displayed.

2. To delete a column or a table row, hover the cursor over the row, and click the Delete icon .
3. Click the **OK** button.

Model Store

A model store is a virtual representation of mapped logical and physical models, which once deployed, can be accessed from Spectrum™ Technology Platform as well as from other client applications.

The mapped logical and physical models in a model store have data federated from multiple sources, such as databases, file servers, and cloud services. You can design and customize your model store according to your requirement and for each of the logical models in the model store, you can define the cache configuration which ensures data gets refreshed periodically. All these features make model store a convenient option to store and use customized data without having to endure time lag due to network congestion. This local cache improves query performance when flows or services of Spectrum™ Technology Platform reference this data.

You can create multiple model stores to suit the different business requirements and deploy these. However, to access a deployed model store through Spectrum™ Technology Platform you need to first create a connection to it using the **Management Console**. For more information, see [Connecting to a Model Store](#) on page 47.

You can also edit and delete a model store, however, for it you need to undeploy the model store first.

Creating a Model Store

To create a new model store, you need to perform these tasks on the **Create Model Store** page.

1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Log in using your credentials.

3. Click **Modeling**.

4. Click the **Model Store** tab.

5. Click **Create Model Store** icon .

The **Create Model Store** page is displayed, listing the **Name** and **Type** (physical or logical) of all the available models.

6. In the **Name** text box, enter a name for the Model Store.

7. Select the check box corresponding to the models you want to include in the model store. To select all the listed models, select the **Include** check box in the header row.

To remove any selected model, de-select the corresponding row.

Note: You can use the **Filter** type ahead text box to customize the grid display. For example, if the models you want to select begin with the text "Cust", you can type *Cust* in the filter box, and it will display all the models having "Cust" in their name.

The selected models get included in the model store.

Note: If you click the **Save** button including any model, and error message asks you to include at least one model from the table.

8. To define the cache configuration of the logical models included in the model store, click the respective **Cache configuration** button.

Note: The **Cache configuration** button is displayed after you select the **Include** check box for the logical model.

The **Cache configuration for model <name of the model>** pop-up window is displayed. For details on configuring the cache, see [Configuring Cache](#) on page 95.

Note: You can configure the cache of only one logical model at a time.

9. To save as well as deploy the model store, click **Save & Deploy**. If you selected a logical model, but did not select all the mapped physical models, a message prompts you to add the dependent physical model as well. Click **Yes** and proceed.

For successful deployment of a model store, ensure you avoid these two scenarios.

- A constituent logical model is not mapped to any physical model.
- For a logical model, any of its mapped physical model has not been selected for addition.

The model store gets saved and deployed for use in the client application.

Note: In case you did not fulfill any of the criteria mentioned above, your model store still gets saved and an error message is displayed indicating that the model store has not been deployed. For more information, see [.Deploying a Model Store](#) on page 89.

10. To save the model store and deploy it later, click **Save**.


A model store needs to be deployed to be accessed from Spectrum™ Technology Platform as well as other client applications.

- To deploy a model store, see [Deploying a Model Store](#) on page 89.
- To access a model store from within Spectrum™ Technology Platform, you need to create a connection to it using the **Management Console**. For more information, see [Connecting to a Model Store](#) on page 47 in the **Administration Guide**.
- To access a model store from a client application other than Spectrum™ Technology Platform, you need to download the JDBC driver and install it manually. For more information about downloading and installing the JDBC driver, see [Installing JDBC Driver](#) on page 90.

Editing a Model Store

To modify an existing model store, perform these tasks on the **Edit Model Store** page.

Only an undeployed model store can be modified. To modify a deployed model store, you need to first undeploy it.

1. Click **Modeling**.
2. Click the **Model Store** tab.
3. From the list of model stores displayed, select the model store that you want to modify, and click the **Edit Model Store** icon .

Note: You can modify only those model stores that are not already deployed. For a deployed model store, the **Edit Model Store** icon will remain unavailable, till you undeploy it.

The **Edit Model Store** page is displayed.

4. Add or remove the physical or logical models, as required, by clicking the respective **Include** rows.
5. Click **Save** or **Save & Deploy**, as needed.

For successful deployment of a model store, ensure you avoid these two scenarios.

- A constituent logical model is not mapped to any physical model.
- For a logical model, any of its mapped physical model has not been selected for addition.

Deleting a Model Store

Only an undeployed model store can be deleted. To delete a deployed model store, you need to first undeploy it.


1. Click **Modeling**.
2. Click the **Model Store** tab.
3. From the list of models displayed, select the model store that you want to delete, and click the **Delete** icon

A message confirms successful deletion of the selected model store.


Deploying a Model Store

You can deploy an available model store for use in your application, or undeploy an already deployed model store to stop it from being used. To do this, perform these tasks on the **Model Store** page.

1. Click **Modeling**.
2. Click the **Model Store** tab.
3. From the list of models displayed, select the model that you want to deploy (or undeploy), and click one of these icons, as needed.

- **Deploy Model Store** : To deploy the selected model store. This icon is enabled only when you select an undeployed model store from the list.

Note: The deployment will fail if the model store contains any logical model that has all the entities unmapped.

- **Undeploy Model Store** : To undeploy an already deployed model store. This icon is enabled only when you select a deployed model store from the list.


Note: You can deploy (or undeploy) only one model store at a time. If you select multiple model stores, the icons get disabled.

Note: If a dependent physical or logical model is deleted from a model store, the model store becomes unusable.

Previewing a Model Store

To preview a deployed model store, select it from the list of model stores, click **Preview**, and run the required SQL query. The preview result displays a maximum of 50 records.

Note: An un-deployed model store cannot be previewed.

1. Click **Modeling**.
2. Click the **Model Store** tab.
3. From the list of model stores displayed, select the model store that you want to preview, and click the **Preview** icon . The **Preview Model Store - <name of the model>** page is displayed.
4. In the **SQL query** textbox, type in the query to retrieve the desired data from the selected model store. Enclose the table name within double-quotes.

```
select * from "EMPLOYEE_SALARY";
```

Note: If the same table name exists in two different physical models, you need to provide the <physicalmodelname>.table name to run the query successfully.

5. Click **Preview**.
A preview of the data fetched from the constituent physical and logical models is displayed.

Accessing a Model Store

A deployed model store can be accessed from within the Spectrum™ Technology Platform as well as other client applications. If you are accessing it from Spectrum™ Technology Platform, you just need to create a connection to required model store using the **Management Console**, as the JDBC driver gets installed during the Spectrum™ Technology Platform installation. For more information about creating connection to a model store, see [Connecting to a Model Store](#) on page 47 in the **Administration Guide**.

However, if you are accessing it from any other application, you need to download the JDBC driver and install it manually. Once installed, you need to create connection to Metadata Insights using simple code so that you are able to use the model store.

Installing JDBC Driver

Use these steps to download the JDBC driver, install it, and create connection to the model store.

1. On the Spectrum™ Technology Platform welcome page, click **Platform Client Tools**. In the options on the right pane, expand the **Driver** cascading menu, and click the **Download Jar** button for JDBC Driver.

2. Save the *modelstore-jdbc.jar* file in the default browser directory.

Note: You do not need to restart the server.

Once the JDBC driver is installed, use these details to connect to the Spectrum™ server using a simple code or any application:

1. Spectrum™ username
2. Password to authenticate the Spectrum™ username
3. JDBC Driver Class Name =
`com.pb.spectrum.modelstore.driver.modelstoreDriver`
4. Connection String Template =
`jdbc:spectrum://${host}:${port}/${modelstore}`.

Fill in the server IP or machine name in place of `{host}`, the port for the Metadata Insights in `{port}`, and the name of the model store you wish to access in `{modelstore}`.

Note: Default for `{port}` is 32750.

You can now connect to any desired model store.

Installing ODBC Driver

You can access a model store through an ODBC connection in applications supporting an ODBC interface, such as MS Excel and MS Access.

1. The version of the ODBC driver depends on the architecture of the client application.
 - If the client application version is 32-bit, download the 32-bit PostgreSQL ODBC driver [here](#).
 - If the client application version is 64-bit, download the 64-bit PostgreSQL ODBC driver [here](#).

For example, if you are using the 32-bit version of MS Office, download the 32-bit ODBC driver.

2. Run the installer `psqlodbc.msi` in case of a 32-bit ODBC driver.

In case of a 64-bit ODBC driver, run the `.exe` installer.

Follow the steps to complete the driver installation.

The ODBC driver is installed.

Creating a Data Source Name (DSN)

These instructions are specific to Windows platforms.

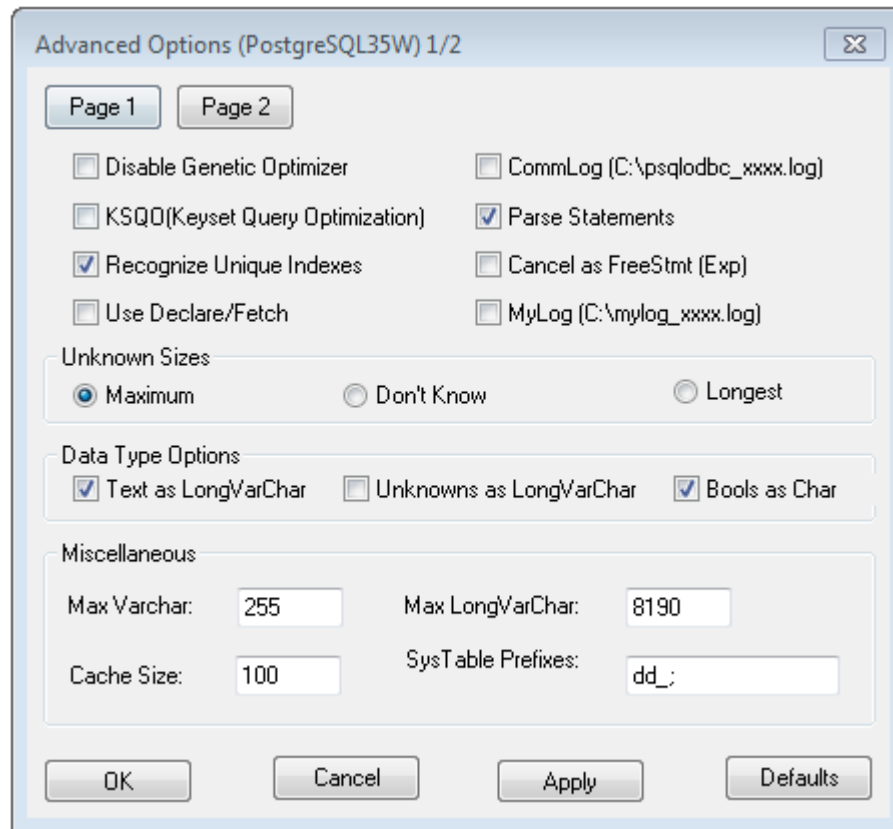
Attention: Ensure you have logged in to the client machine with *administrative rights*.

1. From the **Start** menu, go to **Control Panel > Administrative Tools**.
2. Double-click **Data Sources (ODBC)**.
The **ODBC Data Source Administrator** window opens.

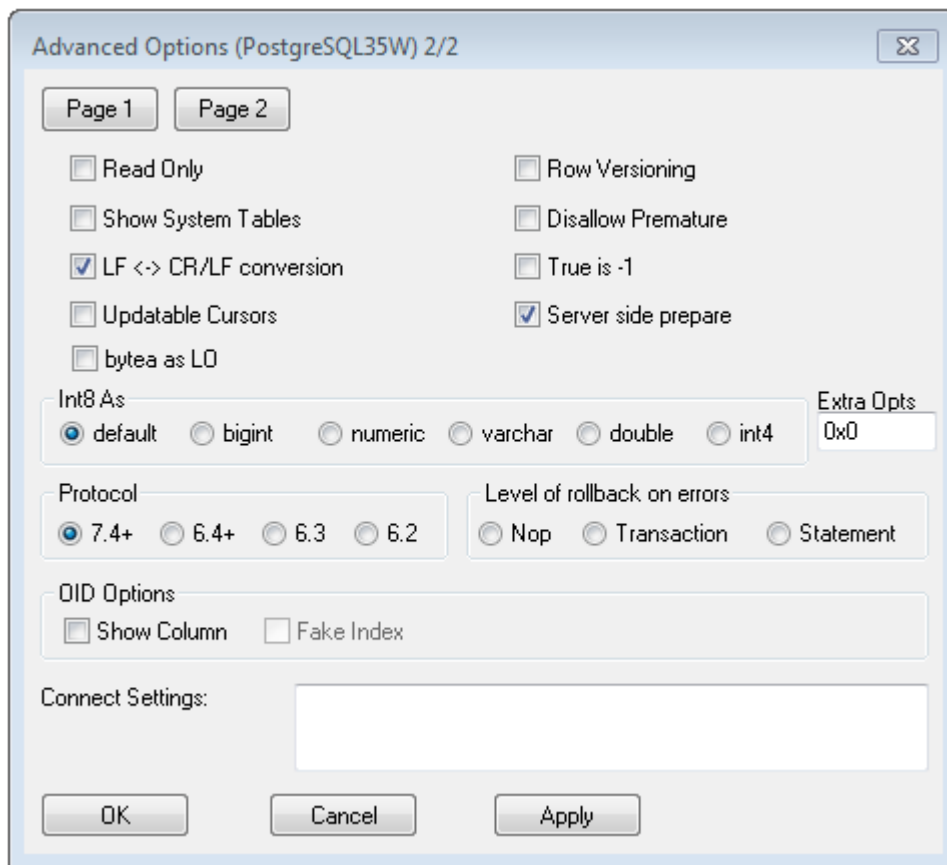
3. Click either the **User DSN** tab or the **System DSN** tab, depending on the type of DSN you wish to create.
4. Click **Add**.
5. From the list of installed drivers on the machine, select `PostgreSQL ANSI`, and click **Finish**. The **PostgreSQL ANSI ODBC Driver (psqlODBC) Setup** popup opens.

Attention: Select `PostgreSQL ANSI` or `PostgreSQL Unicode`, based on the client application through which the ODBC connection is to be used.

6. In the popup, enter the following details to map this DSN to the desired model store:
 - a) In the **Data Source** field, enter the name you wish to assign to the DSN.
 - b) In the **Database** field, enter the name of the model store to which you wish to connect through this DSN.
 - c) In the **Server** field, enter the name of the machine on which the Spectrum™ Technology Platform server is running.
 - d) In the **Port** field, enter the value of the property `edf.odbc.socket.port`, as configured in the `federation_config.properties` file. See [Common Configuration Properties](#).
By default, the port is `32751`.
 - e) In the **User Name** field, enter the username of the Spectrum™ Technology Platform server.
 - f) In the **Password** field, enter the password of the Spectrum™ Technology Platform server.
 - g) Click **Datasource** in the **Options** section.
The **Advanced Options (PostgreSQL30)** popup opens.
 - h) On **Page 1**, configure the options as below:



i) Click **Page 2**, and configure the options as below:



- j) Click **OK** to close the **Advanced Options (PostgreSQL30)** popup.
- k) Click **Test** to check if the created DSN connects to the configured model store successfully.

Important: Ensure the model store is deployed to test the DSN successfully.

- l) Click **Save**.

The created DSN allows you to connect to the configured model store using applications like MS Excel, MS Access.

Accessing a DSN through MS Excel

You can connect to a configured ODBC DSN from a client application, and access the mapped model store to perform the required operations. A DSN is accessible by various client applications, such as MS Excel and MS Access. This example illustrates the steps to connect to an ODBC DSN and access a model store in MS Excel.

1. Open MS Excel.
2. Go to **Data > From Other Sources > From Data Connection Wizard**.
The **Data Connection Wizard** opens.
3. Select ODBC DSN as the type of data source.
4. Click **Next**.
5. From the list of ODBC data sources, select the DSN you wish to connect to.

6. Click **Next**.
7. Ensure the correct model store is selected in the database dropdown.
8. If you wish to select any one of the listed tables of the model store, check the **Connect to a specific table** checkbox, and select the desired table.
Else, to select all the listed tables, leave the checkbox unchecked.

9. Click **Next**.

The contents of the selected table(s) are imported and displayed on the spreadsheet.

Note: The data displayed in MS Excel is read-only.

However, writes are possible while accessing a DSN using MS Access and other client applications.

Note: The example illustrates the process for a 32-bit ODBC driver.

Cache Configuration

You can cache data of all the logical models (and their entities) in a model store. This reduces the delays in accessing data due to network congestion.

Caching can be switched on or off for all or selected entities in a logical model. You can define the time for which the cached data will remain accessible, after which it will expire and will have to be reloaded.

Cache Load The cache for each model is loaded when the first query is run to access it after caching is switched on.

After caching is switched on for a model store, the execution of the first query on it might take longer than usual. This is because the cached data is first refreshed from the data source, after which the required data is fetched from the refreshed cache.

Model's Time to Live This is the duration between a cache load until the next purge of the cache. Once a cache expires, the cached data is reloaded from the data sources when the immediate next query is run.

The next **Time To Live** is calculated from the moment the cache data reload is completed. Once the expiry time is reached, the cached data is purged again.

The default **Time To Live** is 120 minutes.


Configuring Cache

You can configure the cache time for the logical models in a model store by using the **Cache Configuration** button.

1. In a web browser, go to:

`http://server:port/metadata-insights`

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Log in using your credentials.
3. Click **Modeling**.
4. Click the **Model Store** tab.
5. From the list of model stores displayed, select the model store for which you want to define the cache configuration, and click the **Edit Model Store** icon .

Note: You cannot define the cache configuration of already deployed model stores. In such cases, you need to undeploy the model store. The **Edit Model Store** icon will be enabled after deployment.

The **Edit Model Store** page is displayed, listing all the models in this model store.

6. Select the **Include** check box of the logical model and click the **Cache Configuration** button that appears.

Note: This button is displayed only for the logical models in the model store.

Once configuration is defined for a logical model in model store, the **Cache Configuration** button remains enabled for it till you de-select the **Include** check box.

The **Cache configuration for model <name of the logical model>** pop-up is displayed showing a default cache time of **120** minutes (in the **Model's Time To Live** field) for all the entities in the model.

7. In the **Model's Time To Live** field, enter the duration (in minutes) for which you want to enable caching. The default duration is 120 minutes.

Note: The duration you enter in this field is automatically populated in the **Time To Live** fields for all the entities in the model. However, you can override the cache duration for any entity by turning the **Override TTL** selection **On**.

8. To enable caching of the entities in the model, select the respective **Enable Cache** check box. To enable caching of all the entities in the model, select the **Enable Cache** check box in the header.
9. To customize the cache duration for any of the entities in the model, click the **Override TTL** slider to **On**, and enter the required cache override duration in the respective **Time to Live** field.
10. To clear all the cache configuration overrides for the model, click the **Clear all overrides** button. The customizations done in the **Override TTL** and **Time To Live** fields are overridden and the **Time To Live** for all the entities is set to the value defined in the **Model's Time To Live** field.
11. Click the **OK** button.

The cache time is set, as specified for all the entities in the logical model and you return to the **Edit Model Store** page.

3 - Profiling

Making informed business decisions requires quality data. So, it is important for you to have confidence in the completeness, correctness, and validity of your data. Incomplete records, malformed fields, and a lack of context can result in misleading or inaccurate data being delivered to your business users, which can result in flawed decisions.

Data profiling can help you be confident in your data. Profiling scans your data and generates reports that identify problems related to correctness, completeness, and validity. With these reports, you can take actions to fix incorrect or malformed data.

Metadata Insights provides profiling tools to run profiling on your data assets, as well as the data feeding into the logical and physical models defined in Metadata Insights. Using this information, you can determine the reliability of your data, design data quality rules, and perform standardization and normalization routines to fix data quality issues.

The tasks involved in profiling are:

- Creating a profile for your data sources
- Specifying rules for analyzing the quality of your data
- Creating a schedule for profile analysis, if needed
- Specifying a list of users who should be notified of the analysis report

Note: You need to have the Data Integration module installed in order to use the profiling feature of Metadata Insights.

In this section

| | |
|--------------------------|-----|
| Creating a Profile | 99 |
| Running a Profile | 103 |
| Scheduling a Profile | 104 |
| Viewing Analysis History | 106 |
| Data Profiling Result | 107 |

Creating a Profile

You need to create a profile to run the data quality analysis rules. A profile can include data set from any of the logical or physical models created using the Modeling feature of Metadata Insights or it can include data from any of the connections configured in Spectrum™ Technology Platform.

To create a new profile, perform these steps:



1. In a web browser, go to:



```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Log in using your credentials.
3. On the **Home** page, click **Profiling**.
4. The **Profiles** tab shows two alert messages. The first message asks you to create a one-time **ProfilerInternalConnection**, while the second message needs you to expose ProfilerJobInternal job by clicking the **Expose Job** button. Click the **Click here to create connection...** link, which takes you to the **Add Data Source** page in the **Management Console**. Follow the instructions to create the **ProfilerInternalConnection** and **Save** it.

Note: Ignore the test failure message if you get one on clicking the **Test** button.

5. Click the **Expose Job** button and refresh the page.
You can now use the **Profiles** tab to add a new profile.
6. Click the Add Profile button .
7. In the **Profile name** field, enter a name for this profile.
8. In the **Description** field, enter a description for the profile.
9. In the **Select datasource** field, click file browser icon .
10. Choose the type of source you want to profile.
 - To profile a physical or logical model, select **Model**, choose the model you want to profile, then click **OK**.
Tip: If you need to add or modify a model, click **Manage model**.
 - To profile a data source connection, click **Connections**, choose the data source you want to profile, then click **OK**.
Tip: If you need to add or modify a connection, click **Manage connections**.

11. From the list of tables, select tables and columns that you want to include in the profile. To include all the tables and columns, select the check box in the header row. To select specific columns of a given table, click the arrow  beside the table name.
12. In the **Sample size per table** field, specify the sample size for data analysis. For example, if you specify 10000, the analysis will take into consideration 10000 rows of data in every analysis sample and generate report for it. The maximum sample size is 1,000,000.
13. In the **Profiling rules** section, select the rules you want to use to analyze your data. For information about the profiling rules, see [Profiling Rules](#) on page 100.
14. In the **Notification** section, select the option **When the data quality analysis is completed send an email to** if you want to email the analysis report to users. Specify the e-mail addresses of the users. To add multiple email addresses, click the Add button .

Note: You need to configure a mail server to be able to send email. For information about configuring a mail server, see the *Administration Guide*.

15. Click **Save**.

The profile is created and listed on the **Profiles** tab of the **Profiling** page.

Profiling Rules


Profiling rules perform different types of analysis on your data. When setting up a profile, choose the profiling rules that perform the types of data analysis you're interested in. For information about creating a profile, see [Creating a Profile](#) on page 99.

Character Analysis

This rule identifies patterns, scripts, and character types in string fields. When you enable this rule, you will get this information for the string fields in your data:

- **Frequency:** The most-used phrases in the selected string column.
- **Scripts Distribution:** The different scripts identified in the selected string column and their count.
- **String Lengths:** The distribution of string lengths in the selected string column.
- **Character Categories:** The types of characters in the selected column, such as letter, punctuation, and number.
- **Text Patterns:** Converts data in the string column to pattern and displays the pattern, its count, and percentage occurrence. The pattern is determined using this rule:
 - Latin upper-case characters are replaced with "A"
 - Latin lower-case characters are replaced with "a"
 - Digits are replaced with "9"
 - Control characters are replaced with "c"

Custom Pattern Analysis

This rule identifies any pattern in the string column. You can configure as many regular expressions as you want to match your data against. To configure this rule when creating a profile, click the Configure icon , and enter these details:

1. **Key:** Name of the pattern to be identified
2. **Value:** The regular expression for the pattern

For example, if you want to identify email addresses in string columns, enter the regular expression `^(.+)\@(.+)\$` in the **Value** field and email in the **Key** field.

3. To add another expression, click the Add icon and add details of the next key value pair. You can add as many expressions as you would like to match the data against.

When you enable this rule, you will get this information:

- **Validity:** The values that matched at least one of the regular expression patterns in the rule.
- **Pattern Match Distribution:** The distribution of records that matched the regular expressions.

Date Analysis

This rule detects and validates dates in string columns. It also identifies date patterns in the columns and their distribution. This analysis can be useful in detecting date entries in erroneous columns, for example in email data.

If you select this rule, **Data Profiling Results** page displays an additional **Date Summary** tab for the string columns that have dates. This tab shows these details:

- **Validity:** The valid and invalid values.
- **Date Patterns:** The date patterns detected in the selected columns, their total count, and percentage of that pattern in the data set.

Email Analysis

This rule detects and validates the email addresses and determines the distribution of email domains in the selected data column.

If you select this rule, **Data Profiling Results** page displays an additional **Email Summary** tab showing these details:

- **Validity:** The valid and invalid values.
- **Domain Distribution:** The top ten email domains in the selected data column.

Phone Number Analysis

Select this rule to detect and validate phone numbers and identify phone numbers as fixed line numbers, mobile numbers, or any other type of number. This rule also gives the distribution of the phone numbers by country and region. You need to configure this rule to define the default country to use when a phone number does not have a country code.

If you select this rule, **Data Profiling Results** page displays an additional **Phone Number Summary** tab showing these details:

- **Validity:** The the valid and invalid phone numbers.
- **Phone Number Types:** The types of phone numbers, such as mobile, land line, fixed line, VOIP, Pager, voice mail, or toll-free.
- **Phone Numbers by Country:** The country-wise distribution of the detected phone numbers.
- **Phone Numbers by Region:** The region-wise distribution of the detected phone numbers.


Semantic Analysis

Select this rule to detect semantic types, such as first name, city, country, ISO country code 2 and 3, last name (family name), and states. This rule can help find values in incorrect columns, such as city names in a Country column.

If you select this rule, the **Data Profiling Results** page displays an additional **Semantic Type** tab showing the detected semantic types and their frequency.

U.S. Address Analysis

This rule determines the quality of your address data using the U.S. database of Universal Addressing Module. To run this rule, you need to:


- Install the Universal Addressing Module U.S. database and define it as a resource in Management Console. For more information about adding this database resource, the *Administration Guide*.
- Configure the U.S. Address Analysis rule by clicking the Configure button  and entering this information:
 1. **US Address Coder database:** Select the Universal Addressing Module database resource configured in Management Console.
 2. **AddressLine1 field to AddressLine5 field:** Map these fields to the columns of the table you are analyzing. You do not necessarily need to enter column names in all the fields. However, the more specific you are the better the matching score will be.
 3. Map the columns from your table to the **City, Country, USUrbanName, FirmName field, PostalCode, and StateProvince** fields.

If you select this rule, the **Data Profiling Results** page shows the **Address Summary** tab.

- The legend below the chart shows the match score for the data along with the color coding.
- Point anywhere in the graph area to view the match score. The scores are in ranges (0, 1-25, 26-50, 51-80, 81-99, and 100), with zero representing no match of the data to the database. The graph also shows the percentage of matching records detected (color coded).
- Click the area in the graph to view the data that matched or did not match to the database.

International Address Analysis

This rule determines the quality of your address data using the database of the Global Address Validation Module. To run this rule, you need to:

- Install the Global Address Validation database and define it as a resource in Management Console. For more information, see the *Administration Guide*.
- Configure the International Address Analysis rule by clicking the Configure button , and entering this information:
 1. **Addressing Engine Database:** Select the Global Address Validation database resource configured in the **Management Console**.
 2. **AddressLine1 field** and **Country** field: Map these fields to the columns in the table you are analyzing.
 3. Map the columns from your table to the **LastLine**, **City**, **CitySubdivision**, **PostalCode**, **State**, **StateSubdivision**, and **FirmName** fields. You do not necessarily need to enter column names in all the fields. However, the more specific you are the better the matching score will be.

If you select this rule, the **Data Profiling Results** page shows and **Address Summary** tab, which displays:

- **International Address Confidence Distribution:** The match score for the data. The score is color coded. Point anywhere in the donut chart to view the score range (0, 1-25, 26-50, 51-80, 81-99, and 100), with zero representing no match of the data in the database. Click the area in the chart to preview the matching or non-matching data.
- **International Address Precision:** This distribution of validation levels of addresses, such as state, house, postal code, city, city sub-division, and street.

Running a Profile

After a profile is created and listed on the Profiles tab, you can send it for analysis. The profile is analyzed based on the rules selected for it and a report gets displayed on the **Data Profiling Results** page. The report can also be sent directly to the targeted users if you specify this on the **Add Profile** page and you can also download it as a PDF document.

The profile analysis can be run as a one time activity or a recurring event by scheduling the recurrence in the **Add Schedule** page. You can also start analysis directly from the **Profiles** tab. In all these cases the analysis is run and the status displayed on the **Profiles** and the **History** tab.

When a profile starts getting analyzed, a unique ID is assigned to it, which is displayed on the **History** tab. For more information, see [Viewing Analysis History](#) on page 106.


Note: If the same profile is sent for analysis multiple times, every analysis is assigned a different ID.

To run profile analysis from the list of profiles displayed on the **Profiles** tab, perform these steps:

1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Log in using your credentials.
3. On the **Home** page, click **Profiling**.
4. From the list of profiles displayed, select the check box corresponding to the one you want to analyze, and click the **Run Profile** icon .

Note: You can search for the required profile by using the type-ahead **Filter** text box. It helps you search on the basis of the Name, User, or Data Source of the profile.

The selected dataset of the profile starts getting analyzed, with the **Status** column showing the in-progress icon. The icon turns to a tick mark when the profiling is complete. The other details displayed for the profile are:

- **Name:** The name of the profile. Clicking on the profile name takes you to the **Data Profiling Results** page. For more information, see [Table and Column Details in Profiling Result](#) on page 108.
- **Data Source:** The source of the data in the profile being analyzed. It can be a physical or logical model or a direct data source connection.
- **User:** The user who created the profile
- **Last Run Start Time:** The date (in mm/dd/yy format) and time (in hh:mm and AM/PM format) when the most recent profile analysis started
- **Last Run End Time:** The date (in mm/dd/yy format) and time (in hh:mm and AM/PM format) when the most recent profile analysis completed
- **Duration:** The duration (in mm:ss) of the analysis
- **Status:** If the analysis is in progress, complete, or failed
- **Tables analyzed**
- **Records analyzed**
- **More:** Click the PDF icon to download the analysis report. In case the profiling hits some roadblock, you can view the error in this field.

Scheduling a Profile

To schedule a profile analysis, use the **Add Schedule** page. This page gives you various combinations of pattern, time, and date range to schedule the profiling in the way that suits your need. You can also have it as a one-time activity.

To schedule analysis of a profile perform these steps:

1. On the **Profiling** page, click the **Schedules** tab, and click the **Add Schedules** icon . The **Add Schedule** page is displayed.

2. In the **Name** field enter a name for the schedule.
3. From the **Profile** drop down list, select the profile that you want to schedule for analysis.

Note: It displays all the defined profiles.

4. In the **Schedule** section, choose one of the triggers (in the **Trigger** drop down list), as needed.
 - **Date/Time:** To schedule a one-time analysis of the profile. For this analysis, select the **Start date**(in dd/mm/yy format) and **Start time** (in the hh/mm and AM/PM format) from the respective calendars.
 - **Recurring Date/Time:** To schedule a recurring analysis of the profile. The recurrence can be set as a **Pattern** (of days or week or month), **Time** (for example, every three hours), **Range** (at a specified time between a range of dates), or a combination of all the three. For more information, see [Scheduling Recurring Analysis](#) on page 105.
5. To activate this schedule, ensure the **Schedule** slider is maintained at **On**.
6. Click the **Save** button.

The profiling schedule is saved and gets displayed on the **Schedules** tab of the **Profiling** page. This page lists all the profiles that are scheduled for analysis and displays these details for each of those:

- **Profile:** Name of the profile
- **Schedule Name:** Name of the schedule
- **User:** The user who created the schedule
- **Trigger:** It specifies **SCHEDULE** to indicate that the profile is scheduled for analysis
- **Next Run:** Specifies the date (in yyyy/mm/dd format), time (in hh/mm/ss format), and period (AM/PM) when the schedule will be run again.
- **Last Run:** Specifies the date (in yyyy/mm/dd format), time (in hh/mm/ss format), and period (AM/PM) when the schedule was last run.

Note: When analysis of a profile begins, the in-progress profile also gets displayed on the **Profiles** tab of the **Profiling** page in case of recurring scheduling. If the scheduling is non-recurring, the in-progress profile gets moved from the **Scheduling** tab to the **Profiles** tab the moment the processing begins.

Scheduling Recurring Analysis

You can set the recurrence of profile analysis by using the required combination of **Pattern**, **Time** and **Range** in the **Schedule** section of the **Add Schedule** page.

Pattern

- **Every weekday:** To run the analysis on every day of the week

- **Every:** At a specified periodicity. For example, to run analysis every alternate day, choose this option, and enter 2 in the selection box.
- **Day <> of every <> month(s):** To run analysis on a specified day and month. For example, to run analysis on the third day of every alternate month, enter 3 in the **Day** selection box and 2 in the **month(s)** selection box.
- **Every <> month(s) <on a specified day> <of a week>:** For example, to run the analysis every third month on the second Tuesday, enter 3 in the **month(s)** selection box, and from the drop downs, select **on the second** and **Tuesday** options respectively.
- **Days of the week:** Select the days of the week on which you want to run the analysis.

Note: You can select multiple days.

Time

To run the analysis at a specified interval of time, select this option and in the selection boxes specify the number of hours or minutes after which you want to run the analysis.

Range

To specify the range of analysis, select the **Start date** and **End date** (in the dd/mm/yy format) from the respective calendars and select the time (in hh:mm format) and period (AM/PM) from the **Start time** option.

Note: Select the **No end date** check box if you do not want to specify an end date for the recurring analysis.

Viewing Analysis History

The **History** tab of the **Profiling** page displays a list of all the analysis run on various profiles along with details, such as the start and completion time of the analysis, whether it was completed successfully, the unique ID assigned by the system to the analysis, the number of tables and records analyzed in each case, and the data source of the profiles analyzed.

Note: It is recommended to keep deleting the obsolete profile runs from history.

To view the history of all the analysis run by the application, perform these steps:

1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Log in using your credentials.
3. On the **Home** page, click **Profiling**.
4. Click the **History** tab.

A list of all the analysis run on various profiles is displayed, along with these details:

- **ID**: The unique identification assigned by the application to this profile analysis.



Note: In case the same profile is analyzed multiple times, each run will have a unique ID.

- **Name**: The name of the profile. Clicking on the profile name takes you to the **Data Profiling Results** page. For more information, see [Table and Column Details in Profiling Result](#) on page 108.
- **Data Source**: The source of the data in the profile being analyzed. It can be a physical or logical model or a direct data source connection.
- **User**: The user who created the profile
- **Last Run Start Time**: The date (in mm/dd/yy format) and time (in hh:mm and AM/PM format) when the most recent profile analysis started
- **Last Run End Time**: The date (in mm/dd/yy format) and time (in hh:mm and AM/PM format) when the most recent profile analysis completed
- **Duration**: The duration (in mm:ss) of the analysis
- **Status**: If the analysis is in progress, complete, or failed

5. To view profiling report, select the required profile, and click the **View History**  icon.

Note: Use the type-ahead **Filter** text box to search the profile faster. It helps you search on the basis of the Name, Data Source, and User field details.

The **Data Profiling Results** page is displayed. For more information, see [Table and Column Details in Profiling Result](#) on page 108.

6. To refresh the page, click the **Refresh History** icon .
7. To delete an analysis, select it and click the **Delete History**  icon.

Data Profiling Result

The **Data Profiling Result** page displays a comprehensive analysis of a profile, including the completeness and uniqueness of the contained data. The report is generated at the profile level as well as table and column level. At the column level the result displayed varies depending upon the profiling rules selected by you. For example, if you select **Semantic Analysis** rule on the **Add Profile** page, the **Data Profiling Result** will show the detected semantic types in the selected string columns.

To access the **Data Profiling Result** perform these steps:

1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Log in using your credentials.
3. On the **Home** page, click **Profiling**.
4. Perform one of these actions:
 - On the **Profiles** tab, click the **Name** of the profile for which you want to view the analysis report.

Note: The profile **Name** becomes hyper-linked only if the analysis has been run for it.

- On the **Profiles** tab, click the document icon in the **More** column corresponding to the required profile.
- On the **History** tab, click the ID of the profile for which you want to view the analysis report.

The **Data Profiling Result** page is displayed showing the analysis results.

Table and Column Details in Profiling Result

The **Data Profiling Results** page can be divided into three parts.

- A header showing the profile details
- The left pane that displays the profile name and all the tables and column included in it
- The report pane that give a pictorial view of the analyzed data

The report displayed in the right pane is based on what you select in the left pane (profile, table, or column). Details, such as Standard deviation (**Stdev**), Maximum (**Max**), Minimum (**Min**), and **Average** value are displayed at the column level for numerical data, while length frequency and text pattern frequency is displayed for string columns. Besides, based on the rules selected for profiling, additional tabs are displayed in the report, such as **Semantic Type** for columns having semantic type values if **Semantic Analysis** rule is selected. For more information, see [Profiling Rules](#) on page 100.

Profile details

The header of the report displays these details of the analyzed profile.

- **Profile name:** Name of the profile
- **Description:** A brief description of the profile
- **Started:** The date (YYYY/mm/dd), time (hh:mm:ss), and period (AM or PM) when the profile analysis was initialized

- **Data Source:** The source of the data in the profile being analyzed. It can be a physical or logical model or a direct data source connection.
- **User:** The user who created the profile
- **Tables:** Number of tables in the selected model
- **Data source type:** Whether the data source for the profile is a model a database.
- **Records:** Total number of records analyzed

Model and table details

The left navigation pane of the **Data Profiling Results** page displays the profile analyzed and the tables and columns included in it. The pictorial report displayed on the right pane corresponds to the selection made in the left pane. For example, if you click the profile name, the report displays:

- **Summary** for the profile showing the percentage of complete and incomplete rows in the profile data.
- An overview of the tables in the profile data in the **Tables Summary** tab.

You can use the type-ahead **Search** text box above the **Summary** grid to find the relevant table or column.

Note: For the `date` data type, results display time zone of the client machine.

The Data Profiling Results

Based on what you select in the left pane of this page, these reports are displayed in the right pane.

- **Profile analysis report:** Click the data source name in the left pane to display these details:
 - **Completeness:** The percentage completeness of rows in the profile data
 - **Tables Summary:** Displays these details for every table in the profile
 - **Table Name:** Names of all the tables in the profile
 - **Number of Columns:** Number of columns in the table
 - **Number of Records:** Number of records in the table
 - **Completeness (%):** Completeness of records in the table
- **Table report:** Click any of the table names in the left pane to display these details:
 - **Completeness:** The percentage completeness of rows in the table
 - **Column Summary:** Displays these details for every column in the table
 - **Column Name:** Names of all the columns in the table
 - **Column Type:** The data type in the column
 - **Completeness (%):** Completeness of records in the column
 - **Uniqueness (%):** Uniqueness of the data contained in the column
 - **Detected Type:** Displays the semantic types, such as email, phone, city, first name, and last name detected in the string in this column.

- **Null Count Frequency:** Displays the number of null values in every row of the table
- **Column report:** Click any of the columns in the left pane to display these details:
 - **Completeness:** The percentage completeness of data in the column
 - **Unique:** How unique is the data in the column (%)
 - **Min Length** and **Max Length** of characters in case the column data is a string
 - Standard deviation (**Stdev**), Maximum (**Max**), Minimum (**Min**), and **Average** value in case the column has numerical data.
 - **Frequency** of data in the column
 - **Length Frequency** in case the column data type is string
 - **Text Pattern Frequency** in case the column data type is string

4 - Lineage and Impact Analysis

In this section

| | |
|--|-----|
| Lineage and Impact Analysis Technology Preview | 112 |
| Viewing Lineage and Impact Analysis | 112 |
| Icons in Lineage and Impact Analysis | 115 |
| Uses | 119 |

Lineage and Impact Analysis Technology Preview

The Spectrum team is excited to share with you a preview of a major new feature: Lineage and Impact Analysis. This technology preview is an early implementation of Lineage and Impact Analysis containing the fundamental capabilities that we think you'll find most useful. We intend to add significant new capabilities in future releases, so to help make sure we are adding the features you want, we are making this technology preview available to you now. Your feedback will help guide the evolution of Lineage and Impact Analysis.

As you explore Lineage and Impact Analysis, keep this in mind:

- This technology preview contains a limited set of features. When you find yourself saying "I wish I could...", let us know by filing an enhancement request with technical support. Your suggestions will help determine what features we add in future releases. For information about contacting technical support, see www.pitneybowes.com/us/contact-dcs.html.
- Even the best software has some bugs. When you encounter a bug, let us know by submitting a bug report to technical support. Since this is a technology preview, we cannot guarantee that we will resolve your specific problem right away. For information about contacting technical support, see www.pitneybowes.com/us/contact-dcs.html.
- Feel free to use this technology preview in your production environment. Keep in mind that we cannot adhere to normal service level agreements (SLAs) for technology previews.
- We expect to get interesting and unanticipated feedback which could dramatically affect the next release of Lineage and Impact Analysis, so we cannot guarantee that you will be able to preserve all the work you do with Lineage and Impact Analysis when you upgrade to future releases.
- Use your judgment when it comes to making business decisions based on insights generated with this technology preview. We cannot adhere to normal service level agreements (SLAs) for features that are in technology preview.

We hope you enjoy experimenting with Lineage and Impact Analysis and look forward to hearing your feedback.

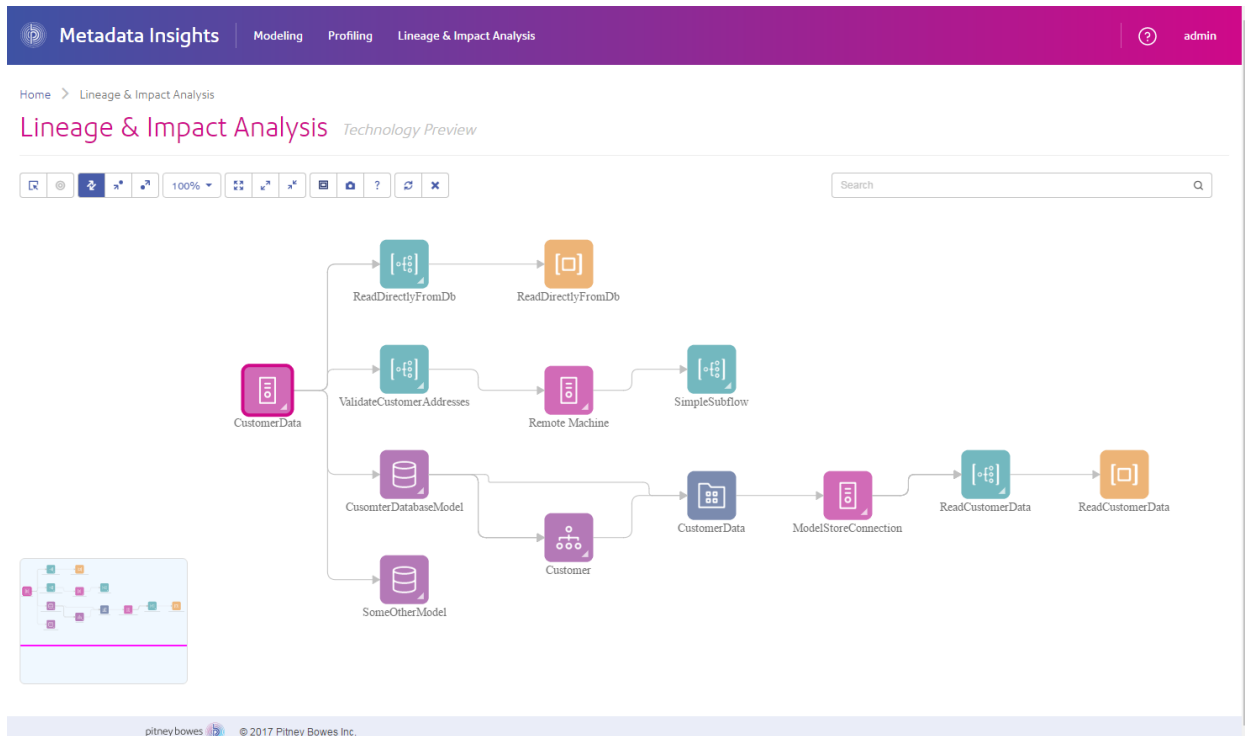
Viewing Lineage and Impact Analysis

The Lineage and Impact Analysis view shows how data flows from data sources to data destinations and through Spectrum™ Technology Platform flows. Lineage and impact analysis are similar concepts that describe different ways of tracing the flow of data.

Lineage shows where data comes from. You can use it to trace the path of data back to its source, showing all the systems that process and store the data along the way, such as Spectrum™ Technology Platform flows, databases, and files.

Impact analysis shows where data goes and the systems that depend on data from a selected data resource. You can use it to view the flows, databases, and files that use a data resource directly or indirectly. Looking at impact analysis is useful if you want to understand how a modification to a database, file, or flow will affect the processes and systems that use the data.

Metadata Insights can show lineage and impact analysis in a single diagram that shows the complete flow of data from source to destination. You can also choose to view lineage only or impact only. By viewing data lineage and impact analysis together you can pinpoint issues in your data processes and plan for upgrades and modifications to your data processes.



1. In a web browser, go to:

`http://server:port/metadata-insights`

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Log in.



3. Click **Lineage & Impact Analysis**.

4. Click the Select Entity button .

5. Select the type of entity you want to analyze. For example, if you want to view lineage or impact analysis for a dataflow, select **Dataflows**.

6. Select the specific entity whose data you want to analyze.

7. Click **OK**.

The resulting diagram shows the flow of data into and out of the entity you selected. The entity's lineage is to the left of the entity. The entity's impact is to the right. You can change the diagram to show only the entity's lineage  or only the entity's impact .

Some entities are expandable, allowing you to view the flow of data within the entity, such as the specific tables that the data flows into in a database, or the stages in a Spectrum™ Technology Platform flow. To expand an entity, click the tab in the lower-right corner of the entity's icon:







Once expanded, you can see the details of the flow of data through the entity. For example, here we have expanded the database entity named Prod SQL Server. You can see that data in the table Customer Table comes from the Spectrum™ Technology Platform flow named Add to Customer DB. In addition, you can see that data from Customer Table is used by two flows: Query Customer Job and Geocode Job. Note as well that the flow Geocode Job writes data back to Customer Table, so in this example Customer Table is both an input to and an output from Geocode Job. This example could represent a situation where Geocode Job reads customer addresses from Customer Table, determines the latitude and longitude coordinates for the addresses, then adds the latitude and longitude fields to the customer records in Customer Table.




Icons in Lineage and Impact Analysis


The Lineage & Impact Analysis canvas in Metadata Insights contains a variety of icons that represent different parts of Spectrum™ Technology Platform as well as the data resources connected to Spectrum™ Technology Platform.

Dataflows and Stages


| Icon | Entity | Lineage | Impact Analysis |
|---|---|--|---|
|  | Dataflow type Expand this icon to see the exposed and last saved versions of the dataflow. | The lineage of a dataflow type shows the data sources that feed into any version of the dataflow (either exposed or last saved). | The impact analysis of a dataflow type shows the destinations where any version of the dataflow writes data. If any version of the dataflow is an exposed job, the impact analysis also shows an activity type icon which represents the fact that the dataflow may be used as an activity in a process flow. |
|  | Dataflow version (exposed or last saved) Expand this icon to see the stages in the dataflow. | The lineage of a dataflow version shows the data sources that feed into the version. It also shows subflows contained in the dataflow. | The impact analysis of a dataflow version shows the destinations where the selected version of the dataflow writes data. If the dataflow is an exposed job, the impact analysis also shows an activity type icon which represents the fact that the dataflow may be used as an activity in a process flow. |
|  | Stage type | Stage types have lineage if the stage represents a subflow. In this case the subflow shows as lineage of the stage type. For other stage types, there is no lineage. | The impact analysis of a stage type shows all the flows that use the stage. For example, if you are looking at the impact analysis of the Read from File stage type, the impact analysis shows all the flows that contain a Read from File stage. |
|  | Stage instance | The lineage of a stage instance shows data used by the stage, and also the stage type. | The impact analysis of a stage instance shows where the data goes to after being processed by the stage. This could be another stage in a dataflow, a database, a file, or any other place where the data is sent. |

| Icon | Entity | Lineage | Impact Analysis |
|---|--|--|---|
|  | <p>Default service options</p> <p>This entity represents the default options for the selected service. Default options are those configured in Management Console.</p> | <p>The lineage of default service options shows entities that the service depends on by default. For example, if a service has a default option that specifies a Spectrum database such as a postal database or a geocoding database, that Spectrum database is shown in the default service option's lineage.</p> | <p>The impact analysis of default service options shows the stages that are configured to use the default service options rather than overriding them in the stage.</p> |



Data Sources and Destinations

| Icon | Entity | Lineage | Impact Analysis |
|--|-------------------|--|--|
|  | <p>Connection</p> | <p>The lineage of a connection shows the entities that write data to the connection.</p> | <p>The impact of a connection shows the entities that read data from the connection.</p> |




Spectrum Databases


| Icon | Entity | Lineage | Impact Analysis |
|---|--------------------------|--|---|
|  | <p>Spectrum database</p> | <p>Spectrum databases have no lineage.</p> | <p>The impact analysis of Spectrum databases shows the stages that use the database. It also shows when the database is configured as a default service option.</p> |

Reports





| Icon | Entity | Lineage | Impact Analysis |
|---|-----------------|---|--|
|  | Report type | Report types have no lineage. | The impact analysis of a report type shows the flows where the report type is used. For example, if you are looking at the impact analysis of the ValidateAddressSummary report type, the impact analysis shows all the flows that generate a ValidateAddressSummary report. |
|  | Report instance | The lineage of a report instance shows the report type. | Report instances have no impact. |

Process Flows and Activities

| Icon | Entity | Lineage | Impact Analysis |
|---|---|---|--|
|  | Process flow type Expand this icon to see the exposed and last saved versions of the process flow. | The lineage of a process flow type shows the activities, such as dataflows, that are called by any version of the process flow, and the data sources that feed into those activities. The output from the activities is also shown in the lineage of the process flow, not in the impact. | Process flow types have no impact since any data produced by a result of running a process flow is shown in the lineage, where the called activities are shown along with the output from those activities. |
|  | Process flow version Expand this icon to see the activities in the process flow. | The lineage of a process flow version shows the activities, such as dataflows, called by the selected version of the process flow, and the data sources that feed into the version. | Process flow versions have no impact since any data produced by a result of running a process flow is shown in the lineage, where the called activities are shown along with the output from those activities. |
|  | Activity type An activity type represents a flow that can be used inside a process flow. | The lineage of an activity type shows the dataflow that the activity represents, and the data sources for the dataflow. | The impact of an activity type shows the process flows that contain the activity. If an activity type has no impact, it is not used in any process flow. |

| Icon | Entity | Lineage | Impact Analysis |
|---|-------------------|---|--|
|  | Activity instance | The lineage of an activity instance is the activity type and the dataflow represented by the activity type. If the activity is preceded in a process flow by another activity, that activity is also shown in the activity's lineage. | The impact of an activity is the activity that follows it in a process flow. |

Modeling

| Icon | Entity | Lineage | Impact Analysis |
|---|---|---|---|
|  | Model store | The lineage of a model store shows the physical and logical models that comprise the model store. | The impact of a model store shows the connections to the model store and the flows that use the model store. |
|  | Physical model | The lineage of a physical model shows the connections that supply data to the physical model, such as database connections, file server connections, or cloud connections. | The impact of a physical model shows the logical models that are mapped to the physical model. The impact also shows the model stores that contain the model. |
|  | Logical model | The lineage of a logical model shows the physical model that is mapped to entities in the logical model. | The impact of a logical model shows the model stores that contain the model. |
|  | Entity in a physical or logical model, or relationship in a logical model | <p>Inside a physical model, entities do not have lineage.</p> <p>Inside a logical model, the lineage of an entity is the relationship between the entity and other entities</p> | <p>Inside a physical model, entities do not have impact.</p> <p>Inside a logical model, relationships have impact on the entities that they connect.</p> |

Uses

Modifying a Database Table


When you make a change to a database, that change can have unintended consequences for systems that rely on its data. Modifications to schemas, keys, columns, or tables can affect downstream processes and ultimately reports in ways that are sometimes obvious and sometimes subtle. So when planning a database modification, you want to be confident that you understand the downstream impact of your changes so that you can modify other systems if needed to ensure your business users continue to have accurate and reliable data on which to base business decisions.

Spectrum™ Technology Platform Metadata Insights provides a visual representation of the flow of data throughout your flows, databases, and files. You can select the database you are modifying and see the flows, databases and files that use the data. With this information you can determine the impact of database modifications. For more information, see [Viewing Lineage and Impact Analysis](#) on page 112.

1. In a web browser, go to:

```
http://server:port/metadata-insights
```

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Click the Select Entity button .
3. Click **Connections**.
4. Click the name of the connection for the table you want. Do not select the radio button since this will show you the lineage and impact analysis of the connection, not a table..
5. Select the table you want then click **OK**.

The resulting diagram shows the connection that contains the table. To the left of the connection (the table's lineage) is any entity that writes data to the table. To the right of the table are any entities that use data from the table, such as flows.

Example

You are working on a project to standardize column names across several databases. You have found a column the table `Customer_Data` that you want to rename. Before modifying the column name, you want to find out which entities are using data from this table so that you can analyze them to find out if they must be updated to accommodate the new column name.

To accomplish this, you open Metadata Insights, go to Lineage & Impact Analysis, and open the entity browser. You click **Connections**, then the name of the connection for the database containing the table. Then you select the table and click **OK**. In the resulting diagram, you expand the connection icon containing the table, which shows you which flows are connected to the table:



The diagram reveals that the HouseholdRelationships flow writes data to the Customer_Data table, and also reads data from it. In addition, the Promotions flow reads data from the Customer_Data table. As a result of reviewing this diagram, you can open the HouseholdRelationships and Promotions flows in Enterprise Designer to find out if there are any changes you need to make to accommodate the renaming of columns in the Customer_Data table.

Updating a Spectrum Database

Spectrum databases contain reference data that some modules use to validate or enhance your data. Examples of Spectrum databases include the postal database used by the Universal Addressing Module and the geocoding databases used by the geocoding modules. Spectrum databases need to be updated on a regular basis to ensure that you have the latest reference data. When it comes time to update a Spectrum database, you want to be confident that the update will go smoothly and not disrupt your operations.

To accomplish a smooth update, you need to know which processes are using the old database. If you know which processes are using the old database, you can:


- Update them to use the new database
- Test them with the new database to ensure you continue to get the expected results

The Lineage & Impact Analysis tool provides the visibility you need to achieve a smooth update of a Spectrum database. It shows the services and flows that use a Spectrum database so that you can update them to point to the new database and test them to ensure that they continue to produce the expected results after the update.

1. In a web browser, go to:

`http://server:port/metadata-insights`

Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Click the Select Entity button .
3. Click **Spectrum Databases**.
4. Click the button next to the module name to select the module then click **OK**. This will display all the Spectrum databases for the selected module.

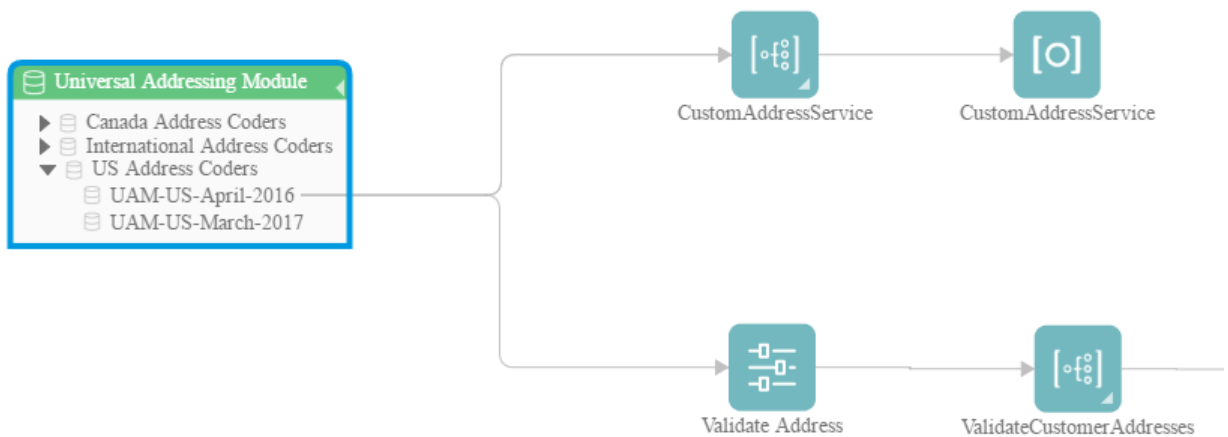
Tip: Even though you can click the module name itself to view the individual databases that are used by the module, we recommend viewing all the module's databases first so that you have a complete picture of all the Spectrum databases in use for the module.

The resulting diagram shows the entities that rely on each Spectrum database for the module you selected.

Example

You use the Universal Addressing Module to validate addresses in your customer database. You are currently using outdated postal database and want to update your system to use the latest database. You have installed the database and defined a Spectrum database for it in Management Console. Now you need to find out where the old database is used so that you can update everything to use the database, and test all affected processes to ensure they are producing the expected results using the new database.

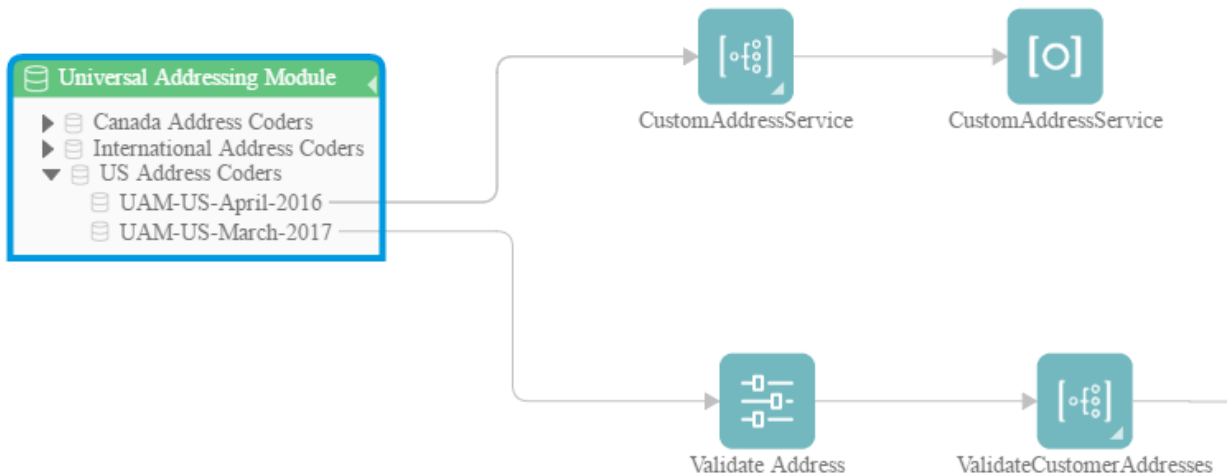
You open Metadata Insights, go to Lineage & Impact Analysis, and open the entity browser. You click **Spectrum Databases**. Since you want to view all database resources for the Universal Addressing Module, you select the button next to **Universal Addressing Module** and click **OK**. The resulting diagram is:



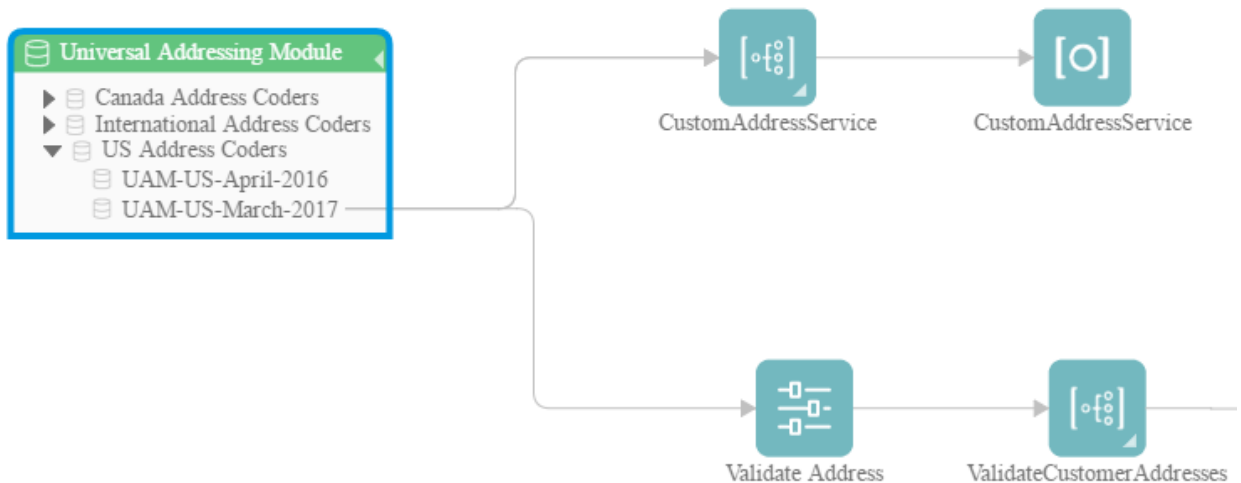
The diagram shows that there are two Spectrum databases on the system, the old one (UAM-US-April-2016) and the new one that you just created in Management

Console (UAM-US-March-2017). There are two entities using the old database: a flow named CustomAddressService and a flow named ValidateCustomerAddresses. There are no entities using the new database.

You update the default service options for Validate Address to use the new database. After refreshing the diagram, you now see this:



As you can see, the default service options for Validate Address are now using the new database (UAM-US-March-2017) but the flow CustomAddressService is still using the old database (UAM-US-April-2016). You open the flow CustomAddressService in Enterprise Designer and change the Validate Address stage to use the new database. When you refresh the diagram you now see this:



Now you are certain that all entities are using the new data. You can now perform tests on CustomAddressService and ValidateCustomerAddresses to verify that the update has not adversely impacted the results. By taking these actions as indicated

by Lineage & Impact Analysis, you can be confident that the update to the new postal data has gone smoothly.

Changing a Service's Default Options


Spectrum™ Technology Platform services have default options which you configure in Management Console. The default options are in effect when a service is used as a stage in a flow and the stage is configured to use the system's default options. So, before you make changes to service options in Management Console, you need to know which flows use the service's default options so that you can ensure that the change does not have undesirable consequences for those flows, or for any business processes that rely on the flows.

To find the flows that use a service's default options:

1. In a web browser, go to:

```
http://server:port/metadata-insights
```

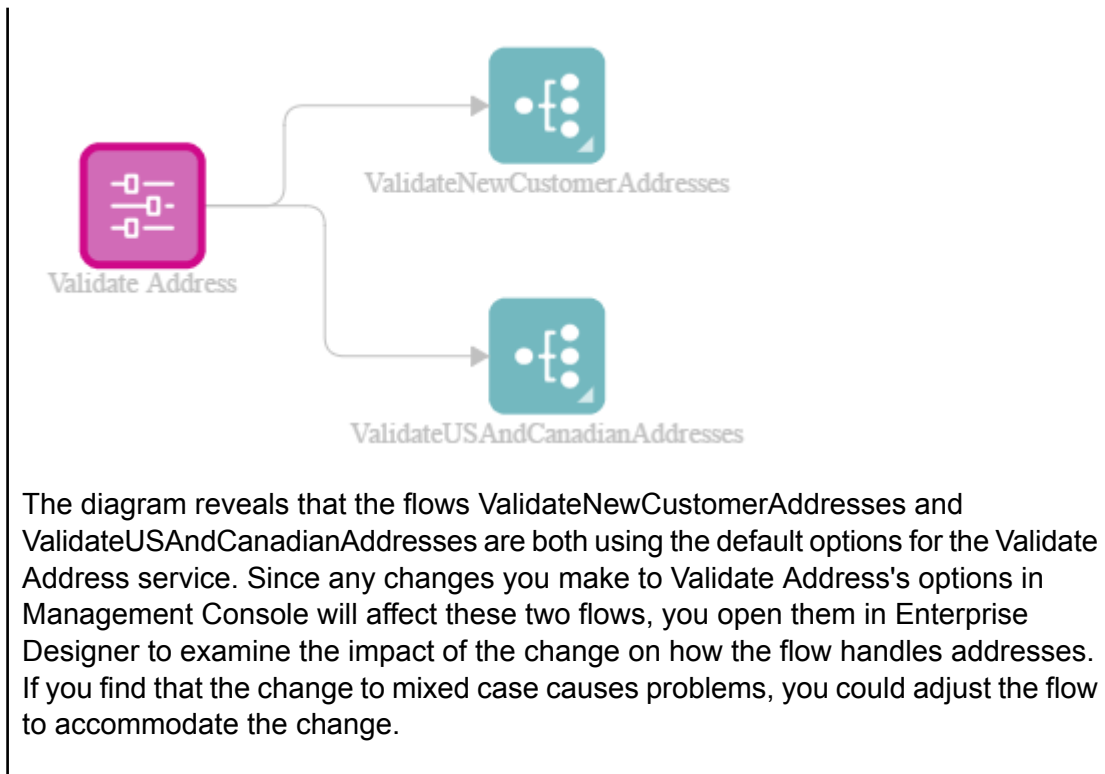
Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Click the Select Entity button .
3. Click **Default Service Options**.
4. Select the service whose default option you are considering modifying then click **OK**.

The resulting diagram shows all the flows that use the service's default options. In other words, these are the flows where the stage option **Override system default options with the following values** is *not* checked. If a flow contains the service, but has this option checked, it is not included in the diagram because it is not using the service's default options, and so would not be impacted by changes to the service's default options in Management Console.

Example

You have configured the service Validate Address to return addresses in upper case and you want to change the casing option to mixed case in Management Console, which means that all Validate Address stages that use the service's default options will start returning addresses in mixed case. To ensure that this change does not have undesirable consequences for these flows, you use the Lineage & Impact Analysis view in Metadata Insight to find out which flows use the Validate Address default options. It shows this diagram:



Modifying a Subflow


A subflow is a flow that can be reused within other flows. Subflows are useful if you want to create a reusable process that can be easily incorporated into multiple flows. When subflows are used extensively, it can be difficult to keep track of which flows use each subflow. This can be a problem if you want to modify a subflow, because you need to make sure that the modification does not cause undesirable results in the flows that use it. Using the Lineage & Impact Analysis feature in Metadata Insights, you can find all the flows that use a particular subflow so that you can be fully aware of the impact of changes you make to a subflow.

To view the flows that use a subflow:

1. In a web browser, go to:

```
http://server:port/metadata-insights
```

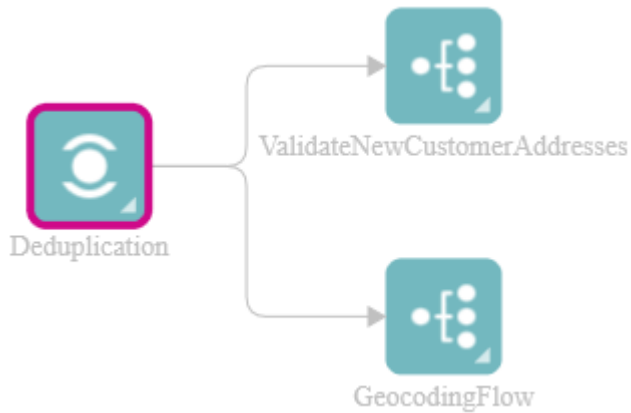
Where *server* is the server name or IP address of your Spectrum™ Technology Platform server and *port* is the HTTP port. By default, the HTTP port is 8080.

2. Click the Select Entity button .
3. Click **Stages**.
4. Select the subflow you want then click **OK**.

The resulting diagram shows all the flows that use the subflow.

Example

You have a subflow named Deduplication. It contains logic for identifying duplicate records in your customer data. You use this subflow in multiple flows. You want to make a change to the deduplication logic and need to make sure that you test the effects of the change in all the flows that use the subflow. So, you use Lineage & Impact Analysis to view the impact of the Deduplication subflow:



The Lineage & Impact Analysis diagram shows that the Deduplication subflow is used in two flows: ValidateNewCustomerAddresses and GeocodingFlow. When you make your change you can check these flows in Enterprise Designer to confirm that you are getting the results you want.

Notices

© 2017 Pitney Bowes Software Inc. 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. The following 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 Software's 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 the following 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 1993-2007 by Nova Marketing Group Inc. All Rights Reserved

© Copyright Second Decimal, LLC

© Copyright Canada Post Corporation

This CD-ROM contains data 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 Software, Inc. 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