



# **TIBCO® Data Virtualization**

## **Email Adapter Guide**

Version 8.7.0 | October 2023

# Contents

---

|                          |          |
|--------------------------|----------|
| <b>Contents</b>          | <b>2</b> |
| <b>Email Adapter</b>     | <b>4</b> |
| Getting Started          | 4        |
| Basic Tab                | 5        |
| Logging                  | 5        |
| Changelog                | 7        |
| Advanced Features        | 10       |
| User Defined Views       | 11       |
| SSL Configuration        | 14       |
| Firewall and Proxy       | 14       |
| Query Processing         | 15       |
| Logging                  | 15       |
| SQL Compliance           | 18       |
| SELECT Statements        | 19       |
| SELECT INTO Statements   | 22       |
| INSERT Statements        | 22       |
| UPDATE Statements        | 23       |
| DELETE Statements        | 24       |
| EXECUTE Statements       | 24       |
| PIVOT and UNPIVOT        | 25       |
| Supported Mail Protocols | 26       |
| IMAP                     | 27       |
| POP                      | 29       |
| SMTP                     | 30       |
| Data Model               | 31       |
| Tables                   | 31       |
| Stored Procedures        | 37       |

|   |           |
|---|-----------|
| <b>Connection String Options</b>                        | <b>46</b> |
| Authentication  | 50        |
| SMTP  | 54        |
| SSL   | 55        |
| Firewall  | 57        |
| Proxy   | 60        |
| Logging   | 66        |
| Schema  | 67        |
| Miscellaneous   | 68        |
| <b>TIBCO Product Documentation and Support Services</b> | <b>77</b> |
| How to Access TIBCO Documentation                       | 77        |
| How to Contact TIBCO Support                            | 78        |
| Release Version Support                                 | 78        |
| How to Join TIBCO Community                             | 79        |
| <b>Legal and Third-Party Notices</b>                    | <b>80</b> |

# Email Adapter

---

## Email Version Support

The adapter is a standard IMAP and POP client, as well as an SMTP server. The adapter models your account's mailboxes as bidirectional tables, enabling you to send, move, delete, and search messages through standard SQL.

## SQL Compliance

The [SQL Compliance](#) section shows the SQL syntax supported by the adapter and points out any limitations.

# Getting Started

## Connecting to Email

[Basic Tab](#) shows how to authenticate to Email and configure any necessary connection properties. Additional adapter capabilities can be configured using the available [Connection](#) properties on the Advanced tab. The Advanced Settings section shows how to set up more advanced configurations and troubleshoot connection errors.

## Deploying the Email Adapter

To deploy the adapter, you can execute the `server_util` utility via the command line by

1. Unzip the `tdv.email.zip` file to the location of your choice.
2. Open a command prompt window.
3. Navigate to the `<TDV_install_dir>/bin`
4. Enter the `server_util` command with the `-deploy` option:

```
server_util -server <hostname> [-port <port>] -user <user> -  
password <password> -deploy -package <TDV_install_  
dir>/adapters/tdv.email/tdv.email.jar
```

Note: When deploying a build of an existing adapter, you will need to undeploy the existing adapter using the `server_util` command with the `-undeploy` option.

```
server_util -server <hostname> [-port <port>] -user <user> -password  
<password> -undeploy -version 1 -name Email
```

## Basic Tab

The User and Password properties in the Authentication section must be set to valid credentials. The Server must be specified to retrieve emails and the SMTPServer must be specified to send emails.

### Deprecation notice for Gmail users

From May 30, 2022, Google no longer supports the use of third-party apps or devices which ask you to sign in to your Google Account using only your username and password. There are alternatives that allow your continued use of our Email drivers to connect to your Google account, which is why Gmail-specific columns, pseudo-columns, and stored procedures are only getting deprecated. We recommend you move to our Gmail drivers that offer more secure methods of authentication.

## Logging

The adapter uses TDV Server's logging (log4j) to generate log files. The settings within the TDV Server's logging (log4j) configuration file are used by the adapter to determine the type of messages to log. The following categories can be specified:

- Error: Only error messages are logged.
- Info: Both Error and Info messages are logged.
- Debug: Error, Info, and Debug messages are logged.

The Other property of the adapter can be used to set Verbosity to specify the amount of detail to be included in the log file, that is:

```
Verbosity=4;
```

You can use Verbosity to specify the amount of detail to include in the log within a category. The following verbosity levels are mapped to the log4j categories:

- 0 = Error
- 1-2 = Info
- 3-5 = Debug

For example, if the log4j category is set to DEBUG, the Verbosity option can be set to 3 for the minimum amount of debug information or 5 for the maximum amount of debug information.

Note that the log4j settings override the Verbosity level specified. The adapter never logs at a Verbosity level greater than what is configured in the log4j properties. In addition, if Verbosity is set to a level less than the log4j category configured, Verbosity defaults to the minimum value for that particular category. For example, if Verbosity is set to a value less than 3 and the Debug category is specified, the Verbosity defaults to 3.

The following list is an explanation of the Verbosity levels and the information that they log.

- 1 - Will log the query, the number of rows returned by it, the start of execution and the time taken, and any errors.
- 2 - Will log everything included in Verbosity 1 and HTTP headers.
- 3 - Will additionally log the body of the HTTP requests.
- 4 - Will additionally log transport-level communication with the data source. This includes SSL negotiation.
- 5 - Will additionally log communication with the data source and additional details that may be helpful in troubleshooting problems. This includes interface commands.

## Configure Logging for the Email Adapter

By default, logging is turned on without debugging. If debugging information is desired, uncomment the following line in the TDV Server's log4j.properties file (default location of this file is: C:\Program Files\TIBCO\TDV Server <version>\conf\server):

```
log4j.logger.com.cdata=DEBUG
```

The TDV Server must be restarted after changing the log4j.properties file, which can be accomplished by running the composite.bat script located at: C:\Program Files\TIBCO\TDV Server <version>\bin. Note that reauthenticating to the TDV Studio is required after restarting the server.

Here is an example of the calls:

```
.\composite.bat monitor restart
```

All logs for the adapter are written to the "cs\_server\_dsrc.log" file as specified in the log4j properties.

**Note:** The "log4j.logger.com.cdata=DEBUG" option is not required if the **Debug Output Enabled** option is set to true within the TDV Studio. To set this option, navigate to **Administrator > Configuration**. Select **Server > Configuration > Debugging** and set the Debug Output Enabled option to **True**.

## Changelog

### General Changes

| Date       | Build Number | Change Type | Description  |
|------------|--------------|-------------|--|
| 12/14/2022 | 8383         | General     | <b>Changed</b> <ul style="list-style-type: none"> <li>Added the Default column to the sys_procedureparameters table.</li> </ul>                          |
| 09/30/2022 | 8308         | General     | <b>Changed</b> <ul style="list-style-type: none"> <li>Added the IsPath column to the sys_procedureparameters table.</li> </ul>                           |
| 08/17/2022 | 8264         | General     | <b>Changed</b> <ul style="list-style-type: none"> <li>We now support handling the keyword "COLLATE" as standard function name as well.</li> </ul>        |
| 06/02/2022 | 8188         | Email       | <b>Deprecated</b> <ul style="list-style-type: none"> <li>Due to Google deprecating the use of third-party apps that allow you to sign in with</li> </ul> |

|            |      |         |   |
|------------|------|---------|---|
|            |      |         | <p>only your username and password, the following Gmail-specific pseudo-columns are deprecated: ShowLabels, ShowThreadId, HasAttachments.</p> <ul style="list-style-type: none"> <li>• The following Gmail-specific columns are deprecated: Labels, ThreadId.</li> <li>• The following Gmail-specific stored procedures are deprecated: SetLabels.</li> </ul>   |
| 09/02/2021 | 7915 | General | <p><b>Added</b></p> <ul style="list-style-type: none"> <li>• Added support for the STRING_SPLIT table-valued function in the CROSS APPLY clause.</li> </ul>   |
| 08/07/2021 | 7889 | General | <p><b>Changed</b></p> <ul style="list-style-type: none"> <li>• Added the KeySeq column to the sys_foreignkeys table.</li> </ul>   |
| 08/06/2021 | 7888 | General | <p><b>Changed</b></p> <ul style="list-style-type: none"> <li>• Added the new sys_primarykeys system table.</li> </ul>   |
| 07/23/2021 | 7874 | General | <p><b>Changed</b></p> <ul style="list-style-type: none"> <li>• Updated the Literal Function Names for relative date/datetime functions. Previously relative date/datetime functions resolved to a different value when used in the projection vs the predicate. I.e: SELECT LAST_MONTH() AS lm, Col FROM Table WHERE Col &gt; LAST_MONTH(). Formerly the two LAST_MONTH() methods would resolve to different datetimes. Now they will match.</li> <li>• As a replacement for the previous behavior, the relative date/datetime functions in the criteria may have an 'L' appended to them. I.e: WHERE col &gt; L_LAST_MONTH(). This will</li> </ul> |



|            |      |         |  |
|------------|------|---------|--|
|            |      |         | <p>continue to resolve to the same values that previously were calculated in the criteria. Note that the "L_" prefix will only work in the predicate - it not available for the projection.</p>  |
| 07/08/2021 | 7859 | General | <p><b>Added</b></p> <ul style="list-style-type: none"> <li>Added the TCP Logging Module for the logging information happening on the TCP wire protocol. The transport bytes that are incoming and ongoing will be logged at verbosity=5.</li> </ul>  |
| 04/23/2021 | 7785 | General | <p><b>Added</b></p> <ul style="list-style-type: none"> <li>Added support for handling client side formulas during insert / update. For example: UPDATE Table SET Col1 = Concat (Col1, " - ", Col2) WHERE Col2 LIKE 'A%'</li> </ul>   |
| 04/23/2021 | 7783 | General | <p><b>Changed</b></p> <ul style="list-style-type: none"> <li>Updated how display sizes are determined for varchar primary key and foreign key columns so they will match the reported length of the column.</li> </ul>   |
| 04/16/2021 | 7776 | General | <p><b>Added</b></p> <ul style="list-style-type: none"> <li>Non-conditional updates between two columns is now available to all drivers. For example: UPDATE Table SET Col1=Col2</li> </ul> <p><b>Changed</b></p> <ul style="list-style-type: none"> <li>Reduced the length to 255 for varchar primary key and foreign key columns.</li> <li>Updated implicit and metadata caching to improve performance and support for multiple connections. Old metadata caches are not compatible - you would need to</li> </ul> |

---

|             |      |         |  |
|-------------|------|---------|--|
|             |      |         | generate new metadata caches if you are currently using CacheMetadata. <ul style="list-style-type: none"> <li>• Updated index naming convention to avoid duplicates</li> <li>• Updated and standardized Getting Started connection help.</li> <li>• Added the Advanced Features section to the help of all drivers.</li> <li>• Categorized connection property listings in the help for all editions.</li> </ul> |
| 04/15 /2021 | 7775 | General | <b>Changed</b> <ul style="list-style-type: none"> <li>• Kerberos authentication is updated to use TCP by default, but will fall back to UDP if a TCP connection cannot be established</li> </ul>   |

---

## Advanced Features

This section details a selection of advanced features of the Email adapter.

### User Defined Views

The adapter allows you to define virtual tables, called *user defined views*, whose contents are decided by a pre-configured query. These views are useful when you cannot directly control queries being issued to the drivers. See [User Defined Views](#) for an overview of creating and configuring custom views.

### SSL Configuration

Use [SSL Configuration](#) to adjust how adapter handles TLS/SSL certificate negotiations. You can choose from various certificate formats; see the [SSLServerCert](#) property under "Connection String Options" for more information.

## Firewall and Proxy

Configure the adapter for compliance with [Firewall and Proxy](#), including Windows proxies and HTTP proxies. You can also set up tunnel connections.

## Query Processing

The adapter offloads as much of the SELECT statement processing as possible to Email and then processes the rest of the query in memory (client-side).

See [Query Processing](#) for more information.

## Logging

See [Logging](#) for an overview of configuration settings that can be used to refine CData logging. For basic logging, you only need to set two connection properties, but there are numerous features that support more refined logging, where you can select subsets of information to be logged using the [LogModules](#) connection property.

## User Defined Views

The Email Adapter allows you to define a virtual table whose contents are decided by a pre-configured query. These are called *User Defined Views*, which are useful in situations where you cannot directly control the query being issued to the driver, e.g. when using the driver from a tool. The User Defined Views can be used to define predicates that are always applied. If you specify additional predicates in the query to the view, they are combined with the query already defined as part of the view.

There are two ways to create user defined views:

- Create a JSON-formatted configuration file defining the views you want.
- DDL statements.

### Defining Views Using a Configuration File

User Defined Views are defined in a JSON-formatted configuration file called *UserDefinedViews.json*. The adapter automatically detects the views specified in this file.

You can also have multiple view definitions and control them using the UserDefinedViews connection property. When you use this property, only the specified views are seen by the adapter.

This User Defined View configuration file is formatted as follows:

- Each root element defines the name of a view.
- Each root element contains a child element, called **query**, which contains the custom SQL query for the view.

For example:

```
{
  "MyView": {
    "query": "SELECT * FROM [INBOX] WHERE MyColumn = 'value'"
  },
  "MyView2": {
    "query": "SELECT * FROM MyTable WHERE Id IN (1,2,3)"
  }
}
```

Use the UserDefinedViews connection property to specify the location of your JSON configuration file. For example:

```
"UserDefinedViews",
"C:\\Users\\yourusername\\Desktop\\tmp\\UserDefinedViews.json"
```

## Defining Views Using DDL Statements

The adapter is also capable of creating and altering the schema via DDL Statements such as CREATE LOCAL VIEW, ALTER LOCAL VIEW, and DROP LOCAL VIEW.

### Create a View

To create a new view using DDL statements, provide the view name and query as follows:

```
CREATE LOCAL VIEW [MyViewName] AS SELECT * FROM Customers LIMIT 20;
```

If no JSON file exists, the above code creates one. The view is then created in the JSON configuration file and is now discoverable. The JSON file location is specified by the UserDefinedViews connection property.

## Alter a View

To alter an existing view, provide the name of an existing view alongside the new query you would like to use instead:

```
ALTER LOCAL VIEW [MyViewName] AS SELECT * FROM Customers WHERE  
TimeModified > '3/1/2020';
```

The view is then updated in the JSON configuration file.

## Drop a View

To drop an existing view, provide the name of an existing schema alongside the new query you would like to use instead.

```
DROP LOCAL VIEW [MyViewName]
```

This removes the view from the JSON configuration file. It can no longer be queried.

## Schema for User Defined Views

User Defined Views are exposed in the **UserViews** schema by default. This is done to avoid the view's name clashing with an actual entity in the data model. You can change the name of the schema used for UserViews by setting the UserViewsSchemaName property.

## Working with User Defined Views

For example, a SQL statement with a User Defined View called *UserViews.RCustomers* only lists customers in Raleigh:

```
SELECT * FROM Customers WHERE City = 'Raleigh';
```

An example of a query to the driver:

```
SELECT * FROM UserViews.RCustomers WHERE Status = 'Active';
```

Resulting in the effective query to the source:

```
SELECT * FROM Customers WHERE City = 'Raleigh' AND Status = 'Active';
```

That is a very simple example of a query to a User Defined View that is effectively a combination of the view query and the view definition. It is possible to compose these

queries in much more complex patterns. All SQL operations are allowed in both queries and are combined when appropriate.

## SSL Configuration

### Customizing the SSL Configuration

By default, the adapter attempts to negotiate SSL/TLS by checking the server's certificate against the system's trusted certificate store.

To specify another certificate, see the [SSLServerCert](#) property for the available formats to do so.

## Firewall and Proxy

### Connecting Through a Firewall or Proxy

#### HTTP Proxies

To connect through the Windows system proxy, you do not need to set any additional connection properties. To connect to other proxies, set [ProxyAutoDetect](#) to false.

In addition, to authenticate to an HTTP proxy, set [ProxyAuthScheme](#), [ProxyUser](#), and [ProxyPassword](#), in addition to [ProxyServer](#) and [ProxyPort](#).

#### Other Proxies

Set the following properties:

- To use a proxy-based firewall, set [FirewallType](#), [FirewallServer](#), and [FirewallPort](#).
- To tunnel the connection, set [FirewallType](#) to TUNNEL.
- To authenticate, specify [FirewallUser](#) and [FirewallPassword](#).
- To authenticate to a SOCKS proxy, additionally set [FirewallType](#) to SOCKS5.

# Query Processing

## Query Processing

CData has a client-side SQL engine built into the adapter library. This enables support for the full capabilities that SQL-92 offers, including filters, aggregations, functions, etc.

For sources that do not support SQL-92, the adapter offloads as much of SQL statement processing as possible to Email and then processes the rest of the query in memory (client-side). This results in optimal performance.

For data sources with limited query capabilities, the adapter handles transformations of the SQL query to make it simpler for the adapter. The goal is to make smart decisions based on the query capabilities of the data source to push down as much of the computation as possible. The Email Query Evaluation component examines SQL queries and returns information indicating what parts of the query the adapter is not capable of executing natively.

The Email Query Slicer component is used in more specific cases to separate a single query into multiple independent queries. The client-side Query Engine makes decisions about simplifying queries, breaking queries into multiple queries, and pushing down or computing aggregations on the client-side while minimizing the size of the result set.

There's a significant trade-off in evaluating queries, even partially, client-side. There are always queries that are impossible to execute efficiently in this model, and some can be particularly expensive to compute in this manner. CData always pushes down as much of the query as is feasible for the data source to generate the most efficient query possible and provide the most flexible query capabilities.

## More Information

For a full discussion of how CData handles query processing, see [CData Architecture: Query Execution](#).

## Logging

Capturing adapter logging can be very helpful when diagnosing error messages or other unexpected behavior.

## Basic Logging

You will simply need to set two connection properties to begin capturing adapter logging.

- Logfile: A filepath which designates the name and location of the log file.
- Verbosity: This is a numerical value (1-5) that determines the amount of detail in the log. See the page in the Connection Properties section for an explanation of the five levels.
- MaxLogFileSize: When the limit is hit, a new log is created in the same folder with the date and time appended to the end. The default limit is 100 MB. Values lower than 100 kB will use 100 kB as the value instead.
- MaxLogFileCount: A string specifying the maximum file count of log files. When the limit is hit, a new log is created in the same folder with the date and time appended to the end and the oldest log file will be deleted. Minimum supported value is 2. A value of 0 or a negative value indicates no limit on the count.

Once this property is set, the adapter will populate the log file as it carries out various tasks, such as when authentication is performed or queries are executed. If the specified file doesn't already exist, it will be created.

## Log Verbosity

The verbosity level determines the amount of detail that the adapter reports to the Logfile. Verbosity levels from 1 to 5 are supported. These are described in the following list:

- |   |   |
|---|---|
| 1 | Setting <u>Verbosity</u> to 1 will log the query, the number of rows returned by it, the start of execution and the time taken, and any errors. |
| 2 | Setting <u>Verbosity</u> to 2 will log everything included in <u>Verbosity</u> 1 and additional information about the request.                  |
| 3 | Setting <u>Verbosity</u> to 3 will additionally log HTTP headers, as well as the body of the request and the response.                          |
| 4 | Setting <u>Verbosity</u> to 4 will additionally log transport-level communication with the data source. This includes SSL negotiation.          |
| 5 | Setting <u>Verbosity</u> to 5 will additionally log communication with the data source and  |



---

additional details that may be helpful in troubleshooting problems. This includes interface commands.

---

The Verbosity should not be set to greater than 1 for normal operation. Substantial amounts of data can be logged at higher verboisities, which can delay execution times.

To refine the logged content further by showing/hiding specific categories of information, see LogModules.

## Sensitive Data

Verbosity levels 3 and higher may capture information that you do not want shared outside of your organization. The following lists information of concern for each level:

- Verbosity 3: The full body of the request and the response, which includes all the data returned by the adapter
- Verbosity 4: SSL certificates
- Verbosity 5: Any extra transfer data not included at Verbosity 3, such as non human-readable binary transfer data

## Best Practices for Data Security

Although we mask sensitive values, such as passwords, in the connection string and any request in the log, it is always best practice to review the logs for any sensitive information before sharing outside your organization.

## Java Logging

When Java logging is enabled in Logfile, the Verbosity will instead map to the following logging levels.

- 0: Level.WARNING
- 1: Level.INFO
- 2: Level.CONFIG
- 3: Level.FINE
- 4: Level.FINER

- 5: Level.FINEST

## Advanced Logging

You may want to refine the exact information that is recorded to the log file. This can be accomplished using the LogModules property.

This property allows you to filter the logging using a semicolon-separated list of logging modules.

All modules are four characters long. **Please note that modules containing three letters have a required trailing blank space.** The available modules are:

- **EXEC**: Query Execution. Includes execution messages for original SQL queries, parsed SQL queries, and normalized SQL queries. Query and page success/failure messages appear here as well.
- **INFO**: General Information. Includes the connection string, driver version (build number), and initial connection messages.
- **HTTP**: HTTP Protocol messages. Includes HTTP requests/responses (including POST messages), as well as Kerberos related messages.
- **SSL** : SSL certificate messages.
- **OAUT**: OAuth related failure/success messages.
- **SQL** : Includes SQL transactions, SQL bulk transfer messages, and SQL result set messages.
- **META**: Metadata cache and schema messages.
- **TCP** : Incoming and Ongoing raw bytes on TCP transport layer messages.

An example value for this property would be.

```
LogModules=INFO;EXEC;SSL ;SQL ;META;
```

Note that these modules refine the information as it is pulled after taking the Verbosity into account.

## SQL Compliance

The Email Adapter supports several operations on data, including querying, deleting, modifying, and inserting.

## SELECT Statements

See [SELECT Statements](#) for a syntax reference and examples.

See [Data Model](#) for information on the capabilities of the Email API.

## INSERT Statements

See [INSERT Statements](#) for a syntax reference and examples, as well as retrieving the new records' Ids.

## UPDATE Statements

The primary key Id is required to update a record. See [UPDATE Statements](#) for a syntax reference and examples.

## DELETE Statements

The primary key Id is required to delete a record. See [DELETE Statements](#) for a syntax reference and examples.

## EXECUTE Statements

Use EXECUTE or EXEC statements to execute stored procedures. See [EXECUTE Statements](#) for a syntax reference and examples.

## Names and Quoting

- Table and column names are considered identifier names; as such, they are restricted to the following characters: [A-Z, a-z, 0-9, \_:@].
- To use a table or column name with characters not listed above, the name must be quoted using double quotes ("name") in any SQL statement.
- Strings must be quoted using single quotes (e.g., 'John Doe').

## SELECT Statements

A SELECT statement can consist of the following basic clauses.

- SELECT
- INTO
- FROM
- JOIN
- WHERE
- GROUP BY
- HAVING
- UNION
- ORDER BY
- LIMIT

## SELECT Syntax

The following syntax diagram outlines the syntax supported by the Email adapter:

```

SELECT {
  [ TOP <numeric_literal> ]
  {
    *
    | {
      <expression> [ [ AS ] <column_reference> ]
      | { <table_name> | <correlation_name> } .*
    } [ , ... ]
  }
  [ INTO csv:// [ filename= ] <file_path> [ ;delimiter=tab ] ]
  {
    FROM <table_reference> [ [ AS ] <identifier> ]
  }
  [ WHERE <search_condition> ]
  [
    LIMIT <expression>
  ]
} | SCOPE_IDENTITY()
<expression> ::=
  | <column_reference>
  | @ <parameter>
  | ?
  | COUNT( * | { [ DISTINCT ] <expression> } )
  | { AVG | MAX | MIN | SUM | COUNT } ( <expression> )

```

```

| NULLIF ( <expression> , <expression> )
| COALESCE ( <expression> , ... )
| CASE <expression>
    WHEN { <expression> | <search_condition> } THEN { <expression> |
NULL } [ ... ]
    [ ELSE { <expression> | NULL } ]
    END
| <literal>
| <sql_function>
<search_condition> ::=
{
    <expression> { = | > | < | >= | <= | <> | != | LIKE | AND | OR |
BETWEEN | IN } [ <expression> ]
} [ { AND | OR } ... ]

```

## Examples

1. Return all columns:

```
SELECT * FROM [INBOX]
```

2. Rename a column:

```
SELECT "Subject" AS MY_Subject FROM [INBOX]
```

3. Cast a column's data as a different data type:

```
SELECT CAST(AnnualRevenue AS VARCHAR) AS Str_AnnualRevenue FROM
[INBOX]
```

4. Search data:

```
SELECT * FROM [INBOX] WHERE Subject = 'Test'
```

5. The Email APIs support the following operators in the WHERE clause: =, >, <, >=, <=, <>, !=, LIKE, AND, OR, BETWEEN, IN.

```
SELECT * FROM [INBOX] WHERE Subject = 'Test';
```

6. Return the number of items matching the query criteria:

```
SELECT COUNT(*) AS MyCount FROM [INBOX]
```

## SELECT INTO Statements

You can use the SELECT INTO statement to export formatted data to a file.

### Data Export with an SQL Query

The following query exports data into a file formatted in comma-separated values (CSV):

```
boolean ret = stat.execute("SELECT From, Subject INTO "csv://c:/[INBOX].txt" FROM "[INBOX]" WHERE Subject = 'Test'");
System.out.println(stat.getUpdateCount()+" rows affected");
```

You can specify other file formats in the URI. The following example exports tab-separated values:

```
Statement stat = conn.createStatement();
boolean ret = stat.execute("SELECT * INTO "[INBOX]" IN 'csv://filename=c:/[INBOX].csv;delimiter=tab' FROM "[INBOX]" WHERE Subject = 'Test'");
System.out.println(stat.getUpdateCount()+" rows affected");
```

## INSERT Statements

To create new records, use INSERT statements.

### INSERT Syntax

The INSERT statement specifies the columns to be inserted and the new column values. You can specify the column values in a comma-separated list in the VALUES clause, as shown in the following example:

```
INSERT INTO <table_name>
( <column_reference> [ , ... ] )
VALUES
( { <expression> | NULL } [ , ... ] )
```

```

<expression> ::=
  | @ <parameter>
  | ?
  | <literal>

```

You can use the `executeUpdate` method of the `Statement` and `PreparedStatement` classes to execute data manipulation commands and retrieve the rows affected. To retrieve the Id of the last inserted record use `getGeneratedKeys`. Additionally, set the **RETURN\_GENERATED\_KEYS** flag of the `Statement` class when you call `prepareStatement`.

```

String cmd = "INSERT INTO [INBOX] (Subject) VALUES (?)";
PreparedStatement pstmt = connection.prepareStatement(
    cmd, Statement.RETURN_GENERATED_KEYS);
pstmt.setString(1, "Spam");
int count = pstmt.executeUpdate();
System.out.println(count+" rows were affected");
ResultSet rs = pstmt.getGeneratedKeys();
while(rs.next()){
    System.out.println(rs.getString("Id"));
}
connection.close();

```

## UPDATE Statements

To modify existing records, use UPDATE statements.

### Update Syntax

The UPDATE statement takes as input a comma-separated list of columns and new column values as name-value pairs in the SET clause, as shown in the following example:

```

UPDATE <table_name> SET { <column_reference> = <expression> } [ , ... ]
WHERE { Id = <expression> } [ { AND | OR } ... ]
<expression> ::=
  | @ <parameter>
  | ?
  | <literal>

```

You can use the `executeUpdate` method of the `Statement` or `PreparedStatement` classes to execute data manipulation commands and retrieve the rows affected, as shown in the following example:

```
String cmd = "UPDATE [INBOX] SET Subject='Spam' WHERE Id = ?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "1");
int count = pstmt.executeUpdate();
System.out.println(count + " rows were affected");
connection.close();
```

## DELETE Statements

To delete information from a table, use DELETE statements.

### DELETE Syntax

The DELETE statement requires the table name in the FROM clause and the row's primary key in the WHERE clause, as shown in the following example:

```
<delete_statement> ::= DELETE FROM <table_name> WHERE { Id =
<expression> } [ { AND | OR } ... ]
<expression> ::=
    | @ <parameter>
    | ?
    | <literal>
```

You can use the executeUpdate method of the Statement or PreparedStatement classes to execute data manipulation commands and retrieve the number of affected rows, as shown in the following example:

```
Connection connection = DriverManager.getConnection
("jdbc:email:Port=993;Server=outlook.office365.com;Password=password;Use
r=user;Protocol=IMAP;SMTP Port=587;SMTP Server=smtp.office365.com;");
String cmd = "DELETE FROM [INBOX] WHERE Id = ?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "1");
int count=pstmt.executeUpdate();
connection.close();
```

## EXECUTE Statements

To execute stored procedures, you can use EXECUTE or EXEC statements.



EXEC and EXECUTE assign stored procedure inputs, referenced by name, to values or parameter names.

## Stored Procedure Syntax

To execute a stored procedure as an SQL statement, use the following syntax:

```
{ EXECUTE | EXEC } <stored_proc_name>
{
  [ @ ] <input_name> = <expression>
} [ , ... ]
<expression> ::=
  | @ <parameter>
  | ?
  | <literal>
```

## Example Statements

Reference stored procedure inputs by name:

```
EXECUTE my_proc @second = 2, @first = 1, @third = 3;
```

Execute a parameterized stored procedure statement:

```
EXECUTE my_proc second = @p1, first = @p2, third = @p3;
```

## PIVOT and UNPIVOT

**PIVOT** and **UNPIVOT** can be used to change a table-valued expression into another table.

### PIVOT

PIVOT rotates a table-value expression by turning unique values from one column into multiple columns in the output. PIVOT can run aggregations where required on any column value.

### PIVOT Syntax

```
"SELECT 'AverageCost' AS Cost_Sorted_By_Production_Days, [0], [1], [2],
[3], [4]
FROM
(
SELECT DaysToManufacture, StandardCost
FROM Production.Product
) AS SourceTable
PIVOT
(
AVG(StandardCost)
FOR DaysToManufacture IN ([0], [1], [2], [3], [4])
) AS PivotTable;"
```

## UNPIVOT

UNPIVOT carries out nearly the opposite to PIVOT by rotating columns of a table-valued expressions into column values.

## UNPIVOT Syntax

```
"SELECT VendorID, Employee, Orders
FROM
(SELECT VendorID, Emp1, Emp2, Emp3, Emp4, Emp5
FROM pvt) p
UNPIVOT
(Orders FOR Employee IN
(Emp1, Emp2, Emp3, Emp4, Emp5)
)AS unpvt;"
```

For further information on PIVOT and UNPIVOT, see [FROM clause plus JOIN, APPLY, PIVOT \(Transact-SQL\)](#)

## Supported Mail Protocols

The Email Adapter connects to mail servers using standard mail protocols for [IMAP](#), [POP](#), and [SMTP](#), as defined by the following RFCs:

For [IMAP](#), the adapter implements a standard IMAP client, as specified in RFC 1730 and RFC 2060.

For [POP](#), the adapter implements a standard Internet post office (POP3) client, as specified in RFC 1725.

And, finally, for [SMTP](#), the adapter implements a standard SMTP client, as specified in RFC 821.

# IMAP

## IMAP Specific Information

Email address fields accept the following formats for email addresses:

- "Friendly Name" <address@company.com>
- address@company.com

## SELECT

Email will be listed newest to oldest. By default, the MessageBody and Headers will not be returned when listing multiple messages.

*Note:* By default, the max number of messages returned will be 100. This can be changed by setting either LIMIT or MaxItems. If you wish to return all email within a mailbox, you will need to specify a value of 0 or -1.

### List Mail Messages

To list messages within a specific mailbox, specify the mailbox as the table. This will list the most recent messages.

Alternatively, you can specify a range of message Ids to list. This includes the following formats:

```
WHERE Id='10' //For returning only the message with an Id of 10.  
WHERE Id BETWEEN 10 AND 20 //For returning all messages from 10 to 20.  
WHERE Id IN (1,3,5) //For returning messages 1, 3, 5.
```

### Search Email

To search mail, simply specify a value for any of the following columns in the WHERE clause: To, From, BCC, CC, Subject, MessageBody, Flags, Labels, Size, or Date. This includes complex statements. For example:

**Note that the CONTAINS function requires full words (For example Test, not Tes), just like the search bar from the UI.** For example:

```
SELECT * FROM [Inbox] WHERE CONTAINS(MessageBody, "Test") AND ([From] = test1@email.com OR [From] = test2@email.com) AND Date>'1-1-2012'
```

This will return all messages where the message body contains the text 'Test', and the message is from either test1@email.com or test2@email.com.

In addition to searching by standard columns, you can also search by attachment file names and by whether or not an attachment is on the email. For example:

```
SELECT * FROM [Inbox] WHERE HasAttachment='True' AND Attachments='.txt'
```

Use of parentheses is also supported for complex WHERE clauses.

As an alternative to the above method of creating search criteria, the SearchCriteria pseudo column is also available. This will accept any valid IMAP search criteria as specified by RFC.

## INSERT

Please refer to the [SMTP](#) section.

## UPDATE

To move an email from one mailbox to another, you will need to specify the Mailbox in the SET clauses of the message and the ID in the WHERE clause. For example:

```
UPDATE [Inbox] SET Mailbox='NewMailboxName' WHERE Id='MessageId'
```

Id may consist of a single message number, a range of messages specified by two message numbers separated by ':' (e.g. "1:5"), and/or individual message numbers separated by ',' (e.g. "1:5,7,10").

When moving an email from one mailbox to another, you may not specify any additional updates.

## Additional Notes

By default, the number of messages returned per page will be 25. To change this, you can set the ItemsPerPage pseudo column.

All message Ids returned are temporary Ids and may change in subsequent requests to the server. To use static Ids, you can set `UIDMode=true`. However, only listing messages is supported with UIDs.

## POP

### POP Specific Information

Email address fields contained within this table accept the following formats:

- 'Friendly Name' <address@company.com>
- address@company.com

### SELECT

Email will be listed newest to oldest. By default, the `MessageBody` and `Headers` will not be returned when listing multiple messages.

*Note:* By default, the max number of emails returned will be 100. This can be changed by setting either `LIMIT` or `MaxItems`. If you wish to return all emails, you will need to specify a value of 0 or -1.

#### List Emails

To list messages, simply call `SELECT` after specifying the account information. This will list the most recent messages first.

To specify a single Id, which will return headers and the message body, you can specify the following:

```
WHERE Id='10' //For returning only the message with an Id of 10.
```

To list messages beginning with a specific Id, you can set the `StartId` property in the `WHERE` clause. This will cause messages to begin with the Id you specified, to the oldest message available within the `LIMIT`.

```
WHERE StartId='45' //Returns messages 45 through 1
```

#### Search Emails

Searching is not supported by the POP protocol. In order to search emails, you must use IMAP.

## INSERT

Please refer to the [SMTP](#) section.

## UPDATE

Updating is not supported by the POP protocol.

## Additional notes

By default, the number of messages returned per page will be 25. To change this, you can set the `ItemsPerPage` pseudo column.

All message `Ids` returned are temporary `Ids` and may change in subsequent requests to the server. To use static `Ids`, you can set `UIDMode=true`. However, only listing messages is supported with `UIDs`.

# SMTP

## SMTP Specific Information

Email address fields accept both the email addresses and the email address accompanied by the username; for example, "Friendly Name" <address@company.com>

## INSERT

When sending mail with either POP or IMAP, SMTP is used to send the message to the mail server.

To send mail, you can insert a row into the table. Required fields are `Subject`, `To`, and `MessageBody`. For example:

```
INSERT INTO [Inbox] (Subject, MessageBody, To) VALUES ('Test  
Subject', 'Body Text', 'address@company.com')
```

## Additional notes

By default, the `From` field will be populated with the email address supplied in the account. If it is not a complete email address, be sure to set `From` before sending a message.

# Data Model

The Email Adapter models Email entities as relational tables and stored procedures.

## Tables

[Tables](#) describes how to query mailboxes as a table. The adapter detects the available mailboxes at run time.

## Stored Procedures

[Stored Procedures](#) are function-like interfaces to the data source. They can be used to search, update, and modify information in the data source.

## Tables

The adapter models the data in Email into a list of tables that can be queried using standard SQL statements.

Generally, querying Email tables is the same as querying a table in a relational database. Sometimes there are special cases, for example, including a certain column in the WHERE clause might be required to get data for certain columns in the table. This is typically needed for situations where a separate request must be made for each row to get certain columns. These types of situations are clearly documented at the top of the table page linked below.

## Email Adapter Tables

| Name                         | Description  |
|------------------------------|--|
| <a href="#">MailMessages</a> | List, search, delete, and move IMAP messages. List and delete POP messages. And send messages with SMTP. |

## MailMessages

List, search, delete, and move IMAP messages. List and delete POP messages. And send messages with SMTP.

This schema is defined for all tables. Each table name will be a mailbox folder on the server.

## Table Specific Information

You can use square brackets to escape the table names of mailbox folders.

## Protocol Information

For more specifics on using this table with [IMAP](#), [POP](#), and [SMTP](#), refer to [Supported Mail Protocols](#).

## Deprecation notice for Gmail users

From May 30, 2022, Google no longer supports the use of third-party apps or devices which ask you to sign in to your Google Account using only your username and password. There are alternatives that allow your continued use of our Email drivers to connect to your Google account, which is why Gmail-specific columns, pseudo-columns, and stored procedures are only getting deprecated. We recommend you move to our Gmail drivers that offer more secure methods of authentication.

## Columns

| Name        | Type   | ReadOnly | Description                                     |
|-------------|--------|----------|---|
| Id<br>[KEY] | String | True     | Identifier of the mail message.                 |
| Mailbox     | String | True     | The current mailbox that the item is listed in. |
| Subject     | String | False    | The subject of the current message.             |



|                |               |       |   |
|----------------|---------------|-------|---|
| From           | <i>String</i> | True  | The sender email address of the current message.  |
| FullFrom       | <i>String</i> | True  | The sender name and email address of the current message.   |
| To             | <i>String</i> | False | The semicolon-separated list of email addresses of the recipients.  |
| FullTo         | <i>String</i> | False | The semicolon-separated list of names and email addresses of the recipients.  |
| CC             | <i>String</i> | False | The semicolon-separated list of names and email addresses of the CCed recipients.   |
| BCC            | <i>String</i> | False | The semicolon-separated list of names and email addresses of the BCCed recipients (only applicable for sent mail).                              |
| Date           | <i>String</i> | True  | The date the current message was sent.  |
| MessageBody    | <i>String</i> | False | The message body. Available only if you specify a single Id in SELECT.  |
| Attachments    | <i>String</i> | False | Semicolon-separated list of the attachment file names (with path if reading from a file) included in the message.                               |
| AttachmentData | <i>String</i> | False | Semicolon-separated list of the base-64-encoded attachment data included in the message. (You must still specify the file name in Attachments.) |
| Size           | <i>Int</i>    | True  | The size in bytes of the current message.   |
| Flags          | <i>String</i> | False | The flags set on the current message.   |

|                   |               |       |   |
|-------------------|---------------|-------|---|
| Labels            | <i>String</i> | False | The labels of the message, separated by spaces. Gmail treats labels as mailboxes.   |
| ThreadId          | <i>String</i> | True  | The Gmail thread Id associated with the current message. Several messages may have the same thread Id.  |
| MsgId             | <i>String</i> | True  | The message Id associated with the current message. For Gmail, this is a globally unique Id assigned by Google separate from the IMAP Id/UID. |
| PartIds           | <i>String</i> | True  | The Id of the message part.   |
| PartFilenames     | <i>String</i> | True  | The file name of the message part.  |
| PartContentTypes  | <i>String</i> | True  | The content type of the message part.   |
| PartSizes         | <i>String</i> | True  | The size of the message part.   |
| Headers           | <i>String</i> | True  | The message headers for the specified mail message. Available only if you specify a single Id in SELECT.                                      |
| AttachmentHeaders | <i>String</i> | True  | A newline-separated list of the headers associated with each attachment.  |
| ContentIds        | <i>String</i> | True  | A semicolon-separated list of Content-Id headers for each attachment.   |

## Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

| Name                | Type   | Description   |
|---------------------|--------|---|
| ItemsPerPage        | String | Maximum number of items to return per page.<br><br>The default value is 25.   |
| SearchCriteria      | String | The search criteria to use for the search operation (e.g., SENTSINCE dd-MMM-yyyy). The exact format of the search criteria is specified by the IMAP RFCs. The string consists of one or more search keys and their corresponding values (if any) separated by spaces; for example, 'SINCE 1-Feb-1994 NOT FROM Smith'. |
| UIDMode             | String | If true, permanent message Ids are used instead of the default temporary Ids.<br><br>The default value is <i>FALSE</i> .  |
| DownloadAttachments | String | If true, attachments will be downloaded with the messages. Available only when specifying a single message Id or the Include Messages connection property is true.<br><br>The default value is <i>FALSE</i> .   |
| AttachmentFolder    | String | Folder to download attachments to. If this is set, the AttachmentData column will not contain data: The data will be written to files.  |
| EMLFolder           | String | Folder to download the complete MIME message, which is saved as an EML file. If multiple messages are selected, subfolders will be created for each message.  |
| Overwrite           | String | Whether or not to overwrite attachments and EML files.<br><br>The default value is <i>FALSE</i> .   |
| IsHTML              | String | Whether the email is HTML or plain text.<br><br>The default value is <i>TRUE</i> .  |

|                      |               |  |
|----------------------|---------------|--|
| Importance           | <i>String</i> | <p>Importance of the mail message.</p> <p>The allowed values are <i>UNSPECIFIED</i>, <i>LOW</i>, <i>NORMAL</i>, <i>HIGH</i>.</p> <p>The default value is <i>UNSPECIFIED</i>.</p>                       |
| Priority             | <i>String</i> | <p>Priority of the mail message.</p> <p>The allowed values are <i>UNSPECIFIED</i>, <i>NONURGENT</i>, <i>URGENT</i>, <i>NORMAL</i>.</p> <p>The default value is <i>UNSPECIFIED</i>.</p>                 |
| Sensitivity          | <i>String</i> | <p>Sensitivity of the mail message.</p> <p>The allowed values are <i>UNSPECIFIED</i>, <i>PERSONAL</i>, <i>PRIVATE</i>, <i>COMPANYCONFIDENTIAL</i>.</p> <p>The default value is <i>UNSPECIFIED</i>.</p> |
| DeliveryNotification | <i>String</i> | Email address to send a delivery notification to.  |
| ReadReceipt          | <i>String</i> | Email address to send a read receipt to.   |
| ShowLabels           | <i>String</i> | <p>Whether to show labels or not. The EmailService connection property must be set to Gmail.</p> <p>The default value is <i>TRUE</i>.</p>  |
| ShowThreadId         | <i>String</i> | <p>Whether to show the thread Id or not. The EmailService connection property must be set to Gmail.</p> <p>The default value is <i>TRUE</i>.</p>   |
| ShowMsgId            | <i>String</i> | <p>Whether to show the Msg Id or not. The EmailService connection property must be set to either Gmail or Outlook.</p> <p>The default value is <i>TRUE</i>.</p>  |

|                |               |   |
|----------------|---------------|---|
| HasAttachments | <i>String</i> | If true, only messages with attachments will be listed. If false, all messages will be listed. The EmailService connection property must be set to Gmail. |
|----------------|---------------|---|

## Stored Procedures

Stored procedures are function-like interfaces that extend the functionality of the adapter beyond simple SELECT/INSERT/UPDATE/DELETE operations with Email.

Stored procedures accept a list of parameters, perform their intended function, and then return, if applicable, any relevant response data from Email, along with an indication of whether the procedure succeeded or failed.

### Email Adapter Stored Procedures

| Name                            | Description   |
|---------------------------------|---|
| <a href="#">GetPartIMAP</a>     | Gets the message part specified by LocalFile or PartId from the message specified by Id on the IMAP server. |
| <a href="#">GetPartPOP</a>      | Gets the message part specified by LocalFile or PartId from the message specified by Id on the POP server.  |
| <a href="#">MoveEmails</a>      | Moves the messages specified by Ids to the mailbox specified by Mailbox .                                   |
| <a href="#">SendMailMessage</a> | Sends the message to the specified recipient.   |
| <a href="#">SetFlags</a>        | Adds or Rplaces exeisting flags with the specified flags.   |
| <a href="#">SetLabels</a>       | Adds or Rplaces exeisting Labels with the specified Labels.   |

## GetPartIMAP

Gets the message part specified by LocalFile or PartId from the message specified by Id on the IMAP server.

## Input

| Name         | Type          | Description  |
|--------------|---------------|--|
| ID           | <i>String</i> | The Id of the message that contains the part.  |
| LocalFile    | <i>String</i> | The local file where the content of the part will be written to.   |
| Mailbox      | <i>String</i> | The mailbox of the message that contains the part.   |
| PartFileName | <i>String</i> | The part file name of the part to retrieve. Either the PartId or the PartFileName input must be specified. |
| PartID       | <i>String</i> | The part Id of the part to retrieve. Either the part Id or the part file name input must be specified.     |
| UIDMode      | <i>String</i> | If true, permanent message Ids are used instead of the default temporary Ids.                              |

## Result Set Columns

| Name        | Type          | Description  |
|-------------|---------------|--|
| Content     | <i>String</i> | If the LocalFile input is not specified, the base64 encoded content will be output. Otherwise, the path to the file containing the content will be output. |
| ContentType | <i>String</i> | The content type.  |

|          |               |  |
|----------|---------------|--|
| Size     | <i>String</i> | The size of the message part.          |
| FileName | <i>String</i> | The filename of the message part.      |
| To       | <i>String</i> | The list of recipients                 |
| From     | <i>String</i> | The sender of the current message.     |
| CC       | <i>String</i> | The list of CCed recipients.           |
| Subject  | <i>String</i> | The subject of the current message.    |
| Flags    | <i>String</i> | The flags set on the current message.  |
| Date     | <i>String</i> | The date the current message was sent. |

## GetPartPOP

Gets the message part specified by LocalFile or PartId from the message specified by Id on the POP server.

### Input

| <b>Name</b> | <b>Type</b>   | <b>Description</b>   |
|-------------|---------------|--|
| ID          | <i>String</i> | The Id of the message that contains the part.                    |
| LocalFile   | <i>String</i> | The local file where the content of the part will be written to. |

|              |               |  |
|--------------|---------------|--|
| PartFileName | <i>String</i> | The part file name of the part to retrieve. Either the part Id or the part file name input must be specified.            |
| PartID       | <i>String</i> | The part Id of the part to retrieve. Either the part Id or the part file name input must be specified.                   |
| UIDMode      | <i>String</i> | If true, permanent message Ids are used instead of the default temporary Ids.<br><br>The default value is <i>FALSE</i> . |

## Result Set Columns

| <b>Name</b> | <b>Type</b>   | <b>Description</b>   |
|-------------|---------------|--|
| Content     | <i>String</i> | If the LocalFile input is not specified, the base64 encoded content will be output. Otherwise, the path to the file containing the content will be output. |
| ContentType | <i>String</i> | The content type.  |
| Size        | <i>String</i> | The size of the message part.  |
| FileName    | <i>String</i> | The filename of the message part.  |
| To          | <i>String</i> | The list of recipients   |
| From        | <i>String</i> | The sender of the current message.   |
| CC          | <i>String</i> | The list of CCed recipients.   |



|         |               |  |
|---------|---------------|--|
| Subject | <i>String</i> | The subject of the current message.    |
| Flags   | <i>String</i> | The flags set on the current message.  |
| Date    | <i>String</i> | The date the current message was sent. |

## MoveEmails

Moves the messages specified by Ids to the mailbox specified by Mailbox .

### Input

| Name        | Type          | Description   |
|-------------|---------------|---|
| Mailbox     | <i>String</i> | The mailbox where the message is currently located.<br>The default value is <i>Inbox</i> .  |
| Id          | <i>String</i> | This inputs indicates the set of messages to operate on. It may consist of a single message id, a range of messages specified by two message numbers separated by ':' (e.g. '1:5'), and/or individual message numbers separated by ',' (e.g. '1:5,7,10'). |
| Destination | <i>String</i> | The mailbox where the message will be moved.<br>The default value is <i>Inbox</i> .   |

### Result Set Columns

| Name    | Type          | Description  |
|---------|---------------|--|
| Success | <i>String</i> | Boolean response indicating whether or not the operation was successful. |

## SendMessage

Sends the message to the specified recipient.

**Note:** This procedure makes use of **indexed parameters**. These input parameters are denoted with a '#' character at the end of their names.

Indexed parameters facilitate providing multiple instances a single parameter as inputs for the procedure.

Suppose there is an input parameter named Param#. Input multiple instances of an indexed parameter like this:

```
EXEC ProcedureName Param#1 = "value1", Param#2 = "value2", Param#3 = "value3"
```

## Input

| Name        | Type          | Accepts Input Streams | Description  |
|-------------|---------------|-----------------------|--|
| To          | <i>String</i> | <i>False</i>          | The semicolon-separated list of names and email addresses of the recipients. |
| Subject     | <i>String</i> | <i>False</i>          | The subject of the mail message.   |
| From        | <i>String</i> | <i>False</i>          | The email address of the sender.   |
| MessageBody | <i>String</i> | <i>False</i>          | The message body.  |

|                    |               |              |  |
|--------------------|---------------|--------------|--|
| CC                 | <i>String</i> | <i>False</i> | The semicolon-separated list of names and email addresses of the CCed recipients.  |
| BCC                | <i>String</i> | <i>False</i> | The semicolon-separated list of names and email addresses of the BCCed recipients.   |
| Attachment#        | <i>String</i> | <i>False</i> | Semicolon-separated list of the attachment file names (with path if reading from a file) included in the message.  |
| AttachmentData#    | <i>String</i> | <i>False</i> | Semicolon-separated list of the base-64-encoded attachment data included in the message. (You must still specify the filename in Attachments.)                                 |
| AttachmentContent  | <i>String</i> | <i>True</i>  | The content as InputStream to be uploaded.   |
| InlineImage#       | <i>String</i> | <i>False</i> | Semicolon-separated list of the inline image identifiers (cids) to be included in the message.   |
| InlineImageData#   | <i>String</i> | <i>False</i> | Semicolon-separated list of the base-64-encoded image data to be included in the message.  |
| InlineImageContent | <i>String</i> | <i>True</i>  | The content as InputStream to be uploaded.   |
| IsHTML             | <i>String</i> | <i>False</i> | Whether the email is HTML or plain text.<br>The default value is <i>TRUE</i> .   |
| Importance         | <i>String</i> | <i>False</i> | Importance of the mail message.<br><br>The allowed values are <i>UNSPECIFIED</i> , <i>LOW</i> , <i>NORMAL</i> , <i>HIGH</i> .<br><br>The default value is <i>UNSPECIFIED</i> . |

|                      |               |              |  |
|----------------------|---------------|--------------|--|
| Priority             | <i>String</i> | <i>False</i> | <p>Priority of the mail message.</p> <p>The allowed values are <i>UNSPECIFIED</i>, <i>NONURGENT</i>, <i>URGENT</i>, <i>NORMAL</i>.</p> <p>The default value is <i>UNSPECIFIED</i>.</p>                 |
| Sensitivity          | <i>String</i> | <i>False</i> | <p>Sensitivity of the mail message.</p> <p>The allowed values are <i>UNSPECIFIED</i>, <i>PERSONAL</i>, <i>PRIVATE</i>, <i>COMPANYCONFIDENTIAL</i>.</p> <p>The default value is <i>UNSPECIFIED</i>.</p> |
| DeliveryNotification | <i>String</i> | <i>False</i> | Email address to send a delivery notification to.  |
| ReadReceipt          | <i>String</i> | <i>False</i> | Email address to send a read receipt to.   |
| Charset              | <i>String</i> | <i>False</i> | <p>The character set to use in the message.</p> <p>The default value is <i>UTF-8</i>.</p>  |

## Result Set Columns

| Name      | Type          | Description                                       |
|-----------|---------------|---|
| MessageId | <i>String</i> | The Id for the message as returned by the server. |

## SetFlags

Adds or Rplaces exexisting flags with the specified flags.

## Input

| Name          | Type          | Description   |
|---------------|---------------|---|
| Mailbox       | <i>String</i> | The mailbox where the message is located.<br><br>The default value is <i>Inbox</i> .  |
| Id            | <i>String</i> | This inputs indicates the set of messages to operate on. It may consist of a single message id, a range of messages specified by two message numbers separated by ':' (e.g. '1:5'), and/or individual message numbers separated by ',' (e.g. '1:5,7,10'). |
| Flags         | <i>String</i> | Sets the specified flags to the message specified by message set. (e.g., Seen, Deleted, Draft). Flags should be specified as comma separated list(e.g. Seen,Deleted)<br><br>The default value is <i>Seen</i> .  |
| OperationType | <i>String</i> | This indicates weather the specified flags should be added,removed or replace existing flags list.<br><br>The allowed values are <i>ADD</i> , <i>REPLACE</i> , <i>REMOVE</i> .<br><br>The default value is <i>REPLACE</i> .                               |

## Result Set Columns

| Name    | Type          | Description  |
|---------|---------------|--|
| Success | <i>String</i> | Boolean response indicating whether or not the operation was successful. |

## SetLabels

Adds or Rplaces exeisting Labels with the specified Labels.

## Input

| Name          | Type          | Description   |
|---------------|---------------|---|
| Mailbox       | <i>String</i> | The Gmail mailbox where the message is located.<br><br>The default value is <i>Inbox</i> .  |
| Id            | <i>String</i> | This inputs indicates the set of messages to operate on. It may consist of a single message id, a range of messages specified by two message numbers separated by ':' (e.g. '1:5'), and/or individual message numbers separated by ',' (e.g. '1:5,7,10'). |
| Labels        | <i>String</i> | List of labels to be set to the message specified by the MessageSet property. This input should be set to a comma separate list of labels.  |
| OperationType | <i>String</i> | This indicates weather the specified labels should be added,removed or replace existing labels list.<br><br>The allowed values are <i>ADD</i> , <i>REPLACE</i> , <i>REMOVE</i> .<br><br>The default value is <i>REPLACE</i> .                             |

## Result Set Columns

| Name    | Type          | Description  |
|---------|---------------|--|
| Success | <i>String</i> | Boolean response indicating whether or not the operation was successful. |

## Connection String Options

The connection string properties are the various options that can be used to establish a connection. This section provides a complete list of the options you can configure in the

connection string for this provider. Click the links for further details.  
For more information on establishing a connection, see [Basic Tab](#).

## Authentication

---

| Property                 | Description   |
|--------------------------|---|
| <a href="#">Protocol</a> | The type of email server to connect to.   |
| <a href="#">User</a>     | The user of the Email account used to authenticate.                             |
| <a href="#">Password</a> | The password of the email account used to authenticate.                         |
| <a href="#">Server</a>   | The name or address of the mail server.   |
| <a href="#">Port</a>     | The port of the mail server. The default value is 143 for IMAP and 110 for POP. |
| <a href="#">SSLMode</a>  | The authentication mechanism to be used when connecting to the Email server.    |

---

## SMTP

---

| Property                   | Description   |
|----------------------------|---|
| <a href="#">SMTPServer</a> | The name or address of the mail server (SMTP server). |
| <a href="#">SMTPPort</a>   | The server port for SMTP (default 25).                |

---

## SSL

---

| Property                      | Description   |
|-------------------------------|---|
| <a href="#">SSLServerCert</a> | The certificate to be accepted from the server when connecting using TLS/SSL. |

---

## Firewall

| Property                         | Description   |
|----------------------------------|---|
| <a href="#">FirewallType</a>     | The protocol used by a proxy-based firewall.                      |
| <a href="#">FirewallServer</a>   | The name or IP address of a proxy-based firewall.                 |
| <a href="#">FirewallPort</a>     | The TCP port for a proxy-based firewall.                          |
| <a href="#">FirewallUser</a>     | The user name to use to authenticate with a proxy-based firewall. |
| <a href="#">FirewallPassword</a> | A password used to authenticate to a proxy-based firewall.        |

## Proxy

| Property                        | Description   |
|---------------------------------|---|
| <a href="#">ProxyAutoDetect</a> | This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings. |
| <a href="#">ProxyServer</a>     | The hostname or IP address of a proxy to route HTTP traffic through.  |
| <a href="#">ProxyPort</a>       | The TCP port the ProxyServer proxy is running on.   |
| <a href="#">ProxyAuthScheme</a> | The authentication type to use to authenticate to the ProxyServer proxy.  |
| <a href="#">ProxyUser</a>       | A user name to be used to authenticate to the ProxyServer proxy.  |
| <a href="#">ProxyPassword</a>   | A password to be used to authenticate to the ProxyServer proxy.   |
| <a href="#">ProxySSLType</a>    | The SSL type to use when connecting to the ProxyServer proxy.   |
| <a href="#">ProxyExceptions</a> | A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .  |



## Logging

---

| Property                   | Description                                  |
|----------------------------|--|
| <a href="#">LogModules</a> | Core modules to be included in the log file. |

---

## Schema

---

| Property                 | Description   |
|--------------------------|---|
| <a href="#">Location</a> | A path to the directory that contains the schema files defining tables, views, and stored procedures. |

---

## Miscellaneous

---

| Property                       | Description   |
|--------------------------------|---|
| <a href="#">EmailService</a>   | Optimizes the IMAP connection for the service you are working with.   |
| <a href="#">IncludeMessage</a> | Whether to include message body content and attachment data or not.   |
| <a href="#">IsHTML</a>         | Determines whether to return the MessageBody as HTML or plain-text.   |
| <a href="#">KeepAlive</a>      | Determines whether to keep the connection alive across instances.   |
| <a href="#">ListMailboxes</a>  | Whether to list all mailboxes or just the subscribed IMAP mailboxes.<br>IMAP Only.  |
| <a href="#">MaxItems</a>       | Maximum number of items to return.  |
| <a href="#">MaxRows</a>        | Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time. |

---

|                                  |   |
|----------------------------------|---|
| <a href="#">MessageCharset</a>   | Specifies the session character set for decoding character data transferred from the IMAP server. |
| <a href="#">Other</a>            | These hidden properties are used only in specific use cases.                                      |
| <a href="#">Readonly</a>         | You can use this property to enforce read-only access to IMAP from the provider.                  |
| <a href="#">Timeout</a>          | The value in seconds until the timeout error is thrown, canceling the operation.                  |
| <a href="#">UIDMode</a>          | If true, permanent message Ids are used instead of the default temporary Ids.                     |
| <a href="#">UserDefinedViews</a> | A filepath pointing to the JSON configuration file containing your custom views.                  |

## Authentication

This section provides a complete list of the Authentication properties you can configure in the connection string for this provider.

| Property                 | Description   |
|--------------------------|---|
| <a href="#">Protocol</a> | The type of email server to connect to.   |
| <a href="#">User</a>     | The user of the Email account used to authenticate.                             |
| <a href="#">Password</a> | The password of the email account used to authenticate.                         |
| <a href="#">Server</a>   | The name or address of the mail server.   |
| <a href="#">Port</a>     | The port of the mail server. The default value is 143 for IMAP and 110 for POP. |
| <a href="#">SSLMode</a>  | The authentication mechanism to be used when connecting to the Email server.    |

## Protocol

The type of email server to connect to.

### Possible Values

IMAP, POP

### Data Type

string

### Default Value

"IMAP"

### Remarks

Possible values include POP and IMAP. The default value of this field is IMAP.

## User

The user of the Email account used to authenticate.

### Data Type

string

### Default Value

""

### Remarks

Together with [Password](#), this field is used to authenticate to the email servers.

## Password

The password of the email account used to authenticate.

## Data Type

string

## Default Value

""

## Remarks

Together with [User](#), this field is used to authenticate to the email servers.

## Server

The name or address of the mail server.

## Data Type

string

## Default Value

""

## Remarks

This property specifies the IP address or the domain name of the mail server. It must be set before a connection is attempted and cannot be changed once a connection is in progress.

If this property is set to a domain name, a DNS request is initiated.

If the connection is configured to use a SOCKS firewall, the value assigned to this property may be preceded with a "\*". If this is the case, the host name is passed to the firewall unresolved and the firewall performs the DNS resolution.

## Port

The port of the mail server. The default value is 143 for IMAP and 110 for POP.

## Data Type

string

## Default Value

""

## Remarks

A valid port number (a value between 1 and 65535) is required for the connection to take place. The property must be set before a connection is attempted and cannot be changed once a connection is established.

## SSLMode

The authentication mechanism to be used when connecting to the Email server.

## Possible Values

AUTO, NONE, IMPLICIT, EXPLICIT

## Data Type

string

## Default Value

"NONE"

## Remarks

If SSLMode is set to NONE, default plaintext authentication is used to log in to the server. If SSLMode is set to IMPLICIT, the SSL negotiation will start immediately after the connection is established. If SSLMode is set to EXPLICIT, the adapter will first connect in plaintext, and then explicitly start SSL negotiation through a protocol command such as STARTTLS. If SSLMode is set to AUTO, if the remote port is set to the standard plaintext port of the protocol (where applicable), the component will behave the same as if SSLMode is set to EXPLICIT. In all other cases, SSL negotiation will be IMPLICIT.

- AUTO
- NONE
- IMPLICIT
- EXPLICIT

## SMTP

This section provides a complete list of the SMTP properties you can configure in the connection string for this provider.

---

| Property                   | Description   |
|----------------------------|---|
| <a href="#">SMTPServer</a> | The name or address of the mail server (SMTP server). |
| <a href="#">SMTPPort</a>   | The server port for SMTP (default 25).                |

### SMTPServer

The name or address of the mail server (SMTP server).

#### Data Type

string

#### Default Value

""

#### Remarks

This property specifies the IP address or domain name of the mail server. It must be set before a connection is attempted and cannot be changed once a connection is in progress.

If this property is set to a domain name, a DNS request is initiated.

If the connection is configured to use a SOCKS firewall, the value assigned to this property may be preceded with a "\*\*\*". If this is the case, the host name is passed to the firewall unresolved and the firewall performs the DNS resolution.

## SMTPPort

The server port for SMTP (default 25).

### Data Type

string

### Default Value

"25"

### Remarks

A valid port number (a value between 1 and 65535) is required for the connection to take place. The property must be set before a connection is attempted and cannot be changed once a connection is established.

## SSL

This section provides a complete list of the SSL properties you can configure in the connection string for this provider.

---

| Property                      | Description   |
|-------------------------------|---|
| <a href="#">SSLServerCert</a> | The certificate to be accepted from the server when connecting using TLS/SSL. |

---

## SSLServerCert

The certificate to be accepted from the server when connecting using TLS/SSL.

## Data Type

string

## Default Value

""

## Remarks

If using a TLS/SSL connection, this property can be used to specify the TLS/SSL certificate to be accepted from the server. Any other certificate that is not trusted by the machine is rejected.

This property can take the following forms:

| Description  | Example   |
|--|---|
| A full PEM Certificate (example shortened for brevity)                       | -----BEGIN CERTIFICATE-----<br>MIICHTCCAe4CAQAwDQYJKoZIhvd.....Qw<br>== -----END CERTIFICATE----- |
| A path to a local file containing the certificate                            | C:\cert.cer   |
| The public key (example shortened for brevity)                               | -----BEGIN RSA PUBLIC KEY-----<br>MIGfMA0GCSq.....AQAB -----END RSA<br>PUBLIC KEY-----            |
| The MD5 Thumbprint (hex values can also be either space or colon separated)  | 34a929226ae0819f2ec14b4a3d904f801c  |
| The SHA1 Thumbprint (hex values can also be either space or colon separated) | bb150d  |

If not specified, any certificate trusted by the machine is accepted.

Certificates are validated as trusted by the machine based on the System's trust store. The trust store used is the 'javax.net.ssl.trustStore' value specified for the system. If no value is specified for this property, Java's default trust store is used (for example, JAVA\_HOME\lib\security\cacerts).



Use '\*' to signify to accept all certificates. Note that this is not recommended due to security concerns.

## Firewall

This section provides a complete list of the Firewall properties you can configure in the connection string for this provider.

---

| Property                         | Description   |
|----------------------------------|---|
| <a href="#">FirewallType</a>     | The protocol used by a proxy-based firewall.                      |
| <a href="#">FirewallServer</a>   | The name or IP address of a proxy-based firewall.                 |
| <a href="#">FirewallPort</a>     | The TCP port for a proxy-based firewall.                          |
| <a href="#">FirewallUser</a>     | The user name to use to authenticate with a proxy-based firewall. |
| <a href="#">FirewallPassword</a> | A password used to authenticate to a proxy-based firewall.        |

## FirewallType

The protocol used by a proxy-based firewall.

### Possible Values

NONE, TUNNEL, SOCKS4, SOCKS5

### Data Type

string

### Default Value

"NONE"

## Remarks

This property specifies the protocol that the adapter will use to tunnel traffic through the [FirewallServer](#) proxy. Note that by default, the adapter connects to the system proxy; to disable this behavior and connect to one of the following proxy types, set [ProxyAutoDetect](#) to false.

| Type   | Default Port | Description   |
|--------|--------------|---|
| TUNNEL | 80           | When this is set, the adapter opens a connection to Email and traffic flows back and forth through the proxy.   |
| SOCKS4 | 1080         | When this is set, the adapter sends data through the SOCKS 4 proxy specified by <a href="#">FirewallServer</a> and <a href="#">FirewallPort</a> and passes the <a href="#">FirewallUser</a> value to the proxy, which determines if the connection request should be granted.                       |
| SOCKS5 | 1080         | When this is set, the adapter sends data through the SOCKS 5 proxy specified by <a href="#">FirewallServer</a> and <a href="#">FirewallPort</a> . If your proxy requires authentication, set <a href="#">FirewallUser</a> and <a href="#">FirewallPassword</a> to credentials the proxy recognizes. |

To connect to HTTP proxies, use [ProxyServer](#) and [ProxyPort](#). To authenticate to HTTP proxies, use [ProxyAuthScheme](#), [ProxyUser](#), and [ProxyPassword](#).

## FirewallServer

The name or IP address of a proxy-based firewall.

### Data Type

string

### Default Value

""

## Remarks

This property specifies the IP address, DNS name, or host name of a proxy allowing traversal of a firewall. The protocol is specified by [FirewallType](#): Use [FirewallServer](#) with this property to connect through SOCKS or do tunneling. Use [ProxyServer](#) to connect to an HTTP proxy.

Note that the adapter uses the system proxy by default. To use a different proxy, set [ProxyAutoDetect](#) to false.

## FirewallPort

The TCP port for a proxy-based firewall.

### Data Type

int

### Default Value

0

## Remarks

This specifies the TCP port for a proxy allowing traversal of a firewall. Use [FirewallServer](#) to specify the name or IP address. Specify the protocol with [FirewallType](#).

## FirewallUser

The user name to use to authenticate with a proxy-based firewall.

### Data Type

string

### Default Value

""

## Remarks

The `FirewallUser` and `FirewallPassword` properties are used to authenticate against the proxy specified in `FirewallServer` and `FirewallPort`, following the authentication method specified in `FirewallType`.

## FirewallPassword

A password used to authenticate to a proxy-based firewall.

## Data Type

string

## Default Value

""

## Remarks

This property is passed to the proxy specified by `FirewallServer` and `FirewallPort`, following the authentication method specified by `FirewallType`.

## Proxy

This section provides a complete list of the Proxy properties you can configure in the connection string for this provider.

| Property                     | Description   |
|------------------------------|---|
| <code>ProxyAutoDetect</code> | This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set <code>ProxyAutoDetect</code> to <code>FALSE</code> in order use custom proxy settings. |
| <code>ProxyServer</code>     | The hostname or IP address of a proxy to route HTTP traffic through.  |

|                        |  |
|------------------------|--|
| <b>ProxyPort</b>       | The TCP port the ProxyServer proxy is running on.  |
| <b>ProxyAuthScheme</b> | The authentication type to use to authenticate to the ProxyServer proxy.   |
| <b>ProxyUser</b>       | A user name to be used to authenticate to the ProxyServer proxy.   |
| <b>ProxyPassword</b>   | A password to be used to authenticate to the ProxyServer proxy.  |
| <b>ProxySSLType</b>    | The SSL type to use when connecting to the ProxyServer proxy.  |
| <b>ProxyExceptions</b> | A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer . |

## ProxyAutoDetect

This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.

### Data Type

bool

### Default Value

true

### Remarks

This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.

NOTE: When this property is set to True, the proxy used is determined as follows:

- A search from the JVM properties (**http.proxy**, **https.proxy**, **socksProxy**, etc.) is performed.
- In the case that the JVM properties don't exist, a search from

**java.home/lib/net.properties** is performed.

- In the case that `java.net.useSystemProxies` is set to `True`, a search from **the SystemProxy** is performed.
- In Windows only, an attempt is made to retrieve these properties from the **Internet Options** in the **registry**.

To connect to an HTTP proxy, see [ProxyServer](#). For other proxies, such as SOCKS or tunneling, see [FirewallType](#).

## ProxyServer

The hostname or IP address of a proxy to route HTTP traffic through.

### Data Type

string

### Default Value

""

### Remarks

The hostname or IP address of a proxy to route HTTP traffic through. The adapter can use the HTTP, Windows (NTLM), or Kerberos authentication types to authenticate to an HTTP proxy.

If you need to connect through a SOCKS proxy or tunnel the connection, see [FirewallType](#).

By default, the adapter uses the system proxy. If you need to use another proxy, set [ProxyAutoDetect](#) to `false`.

## ProxyPort

The TCP port the ProxyServer proxy is running on.

### Data Type

int

## Default Value

80

## Remarks

The port the HTTP proxy is running on that you want to redirect HTTP traffic through. Specify the HTTP proxy in [ProxyServer](#). For other proxy types, see [FirewallType](#).

## ProxyAuthScheme

The authentication type to use to authenticate to the ProxyServer proxy.

## Possible Values

BASIC, DIGEST, NONE, NEGOTIATE, NTLM, PROPRIETARY

## Data Type

string

## Default Value

"BASIC"

## Remarks

This value specifies the authentication type to use to authenticate to the HTTP proxy specified by [ProxyServer](#) and [ProxyPort](#).

Note that the adapter will use the system proxy settings by default, without further configuration needed; if you want to connect to another proxy, you will need to set [ProxyAutoDetect](#) to false, in addition to [ProxyServer](#) and [ProxyPort](#). To authenticate, set [ProxyAuthScheme](#) and set [ProxyUser](#) and [ProxyPassword](#), if needed.

The authentication type can be one of the following:

- **BASIC:** The adapter performs HTTP BASIC authentication.
- **DIGEST:** The adapter performs HTTP DIGEST authentication.
- **NEGOTIATE:** The adapter retrieves an NTLM or Kerberos token based on the

applicable protocol for authentication.

- **PROPRIETARY:** The adapter does not generate an NTLM or Kerberos token. You must supply this token in the Authorization header of the HTTP request.

If you need to use another authentication type, such as SOCKS 5 authentication, see [FirewallType](#).

## ProxyUser

A user name to be used to authenticate to the ProxyServer proxy.

### Data Type

string

### Default Value

""

### Remarks

The [ProxyUser](#) and [ProxyPassword](#) options are used to connect and authenticate against the HTTP proxy specified in [ProxyServer](#).

You can select one of the available authentication types in [ProxyAuthScheme](#). If you are using HTTP authentication, set this to the user name of a user recognized by the HTTP proxy. If you are using Windows or Kerberos authentication, set this property to a user name in one of the following formats:

```
user@domain  
domain\user
```

## ProxyPassword

A password to be used to authenticate to the ProxyServer proxy.

### Data Type

string



## Default Value

""

## Remarks

This property is used to authenticate to an HTTP proxy server that supports NTLM (Windows), Kerberos, or HTTP authentication. To specify the HTTP proxy, you can set [ProxyServer](#) and [ProxyPort](#). To specify the authentication type, set [ProxyAuthScheme](#).

If you are using HTTP authentication, additionally set [ProxyUser](#) and [ProxyPassword](#) to HTTP proxy.

If you are using NTLM authentication, set [ProxyUser](#) and [ProxyPassword](#) to your Windows password. You may also need these to complete Kerberos authentication.

For SOCKS 5 authentication or tunneling, see [FirewallType](#).

By default, the adapter uses the system proxy. If you want to connect to another proxy, set [ProxyAutoDetect](#) to false.

## ProxySSLType

The SSL type to use when connecting to the [ProxyServer](#) proxy.

## Possible Values

AUTO, ALWAYS, NEVER, TUNNEL

## Data Type

string

## Default Value

"AUTO"

## Remarks

This property determines when to use SSL for the connection to an HTTP proxy specified by [ProxyServer](#). This value can be AUTO, ALWAYS, NEVER, or TUNNEL. The applicable values are the following:

|               |  |
|---------------|--|
| <b>AUTO</b>   | Default setting. If the URL is an HTTPS URL, the adapter will use the TUNNEL option. If the URL is an HTTP URL, the component will use the NEVER option. |
| <b>ALWAYS</b> | The connection is always SSL enabled.  |
| <b>NEVER</b>  | The connection is not SSL enabled.   |
| <b>TUNNEL</b> | The connection is through a tunneling proxy. The proxy server opens a connection to the remote host and traffic flows back and forth through the proxy.  |

## ProxyExceptions

A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

### Data Type

string

### Default Value

""

### Remarks

The [ProxyServer](#) is used for all addresses, except for addresses defined in this property. Use semicolons to separate entries.

Note that the adapter uses the system proxy settings by default, without further configuration needed; if you want to explicitly configure proxy exceptions for this connection, you need to set [ProxyAutoDetect](#) = false, and configure [ProxyServer](#) and [ProxyPort](#). To authenticate, set [ProxyAuthScheme](#) and set [ProxyUser](#) and [ProxyPassword](#), if needed.

## Logging

This section provides a complete list of the Logging properties you can configure in the connection string for this provider.

---

| Property                   | Description                                  |
|----------------------------|--|
| <a href="#">LogModules</a> | Core modules to be included in the log file. |

---

## LogModules

Core modules to be included in the log file.

### Data Type

string

### Default Value

""

### Remarks

Only the modules specified (separated by ';') will be included in the log file. By default all modules are included.

See the [Logging](#) page for an overview.

## Schema

This section provides a complete list of the Schema properties you can configure in the connection string for this provider.

---

| Property                 | Description   |
|--------------------------|---|
| <a href="#">Location</a> | A path to the directory that contains the schema files defining tables, views, and stored procedures. |

---

## Location

A path to the directory that contains the schema files defining tables, views, and stored procedures.

## Data Type

string

## Default Value

"%APPDATA%\\CData\\Email Data Provider\\Schema"

## Remarks

The path to a directory which contains the schema files for the adapter (.rsd files for tables and views, .rsb files for stored procedures). The folder location can be a relative path from the location of the executable. The Location property is only needed if you want to customize definitions (for example, change a column name, ignore a column, and so on) or extend the data model with new tables, views, or stored procedures.

If left unspecified, the default location is "%APPDATA%\\CData\\Email Data Provider\\Schema" with **%APPDATA%** being set to the user's configuration directory:

| Platform | %APPDATA%                                     |
|----------|---|
| Windows  | The value of the APPDATA environment variable |
| Mac      | ~/Library/Application Support                 |
| Linux    | ~/.config                                     |

## Miscellaneous

This section provides a complete list of the Miscellaneous properties you can configure in the connection string for this provider.

| Property                         | Description   |
|----------------------------------|---|
| <a href="#">EmailService</a>     | Optimizes the IMAP connection for the service you are working with.   |
| <a href="#">IncludeMessage</a>   | Whether to include message body content and attachment data or not.   |
| <a href="#">IsHTML</a>           | Determines whether to return the MessageBody as HTML or plain-text.   |
| <a href="#">KeepAlive</a>        | Determines whether to keep the connection alive across instances.   |
| <a href="#">ListMailboxes</a>    | Whether to list all mailboxes or just the subscribed IMAP mailboxes.<br>IMAP Only.  |
| <a href="#">MaxItems</a>         | Maximum number of items to return.  |
| <a href="#">MaxRows</a>          | Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time. |
| <a href="#">MessageCharset</a>   | Specifies the session character set for decoding character data transferred from the IMAP server.   |
| <a href="#">Other</a>            | These hidden properties are used only in specific use cases.  |
| <a href="#">Readonly</a>         | You can use this property to enforce read-only access to IMAP from the provider.  |
| <a href="#">Timeout</a>          | The value in seconds until the timeout error is thrown, canceling the operation.  |
| <a href="#">UIDMode</a>          | If true, permanent message ids are used instead of the default temporary ids.   |
| <a href="#">UserDefinedViews</a> | A filepath pointing to the JSON configuration file containing your custom views.  |

## EmailService

Optimizes the IMAP connection for the service you are working with.

## Data Type

string

## Default Value

"Other"

## Remarks

Possible values include AOL, Gmail, Outlook, Yahoo, and Other.

## IncludeMessage

Whether to include message body content and attachment data or not.

## Data Type

bool

## Default Value

false

## Remarks

Whether to include message body content and attachment data or not.

## IsHTML

Determines whether to return the MessageBody as HTML or plain-text.

## Data Type

bool

## Default Value

true

## Remarks

Determines whether to return the MessageBody as HTML or plain-text.

## KeepAlive

Determines whether to keep the connection alive across instances.

## Data Type

bool

## Default Value

true

## Remarks

Determines whether to keep the connection alive across instances.

## ListMailboxes

Whether to list all mailboxes or just the subscribed IMAP mailboxes. IMAP Only.

## Possible Values

All, Subscribed

## Data Type

string

## Default Value

"All"

## Remarks

Whether to list all mailboxes or just the subscribed IMAP mailboxes. IMAP Only.

- All
- Subscribed

## MaxItems

Maximum number of items to return.

## Data Type

string

## Default Value

"100"

## Remarks

Maximum number of items to return.

## MaxRows

Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.

## Data Type

int

## Default Value

-1



## Remarks

Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.

## MessageCharset

Specifies the session character set for decoding character data transferred from the IMAP server.

### Data Type

string

### Default Value

"UTF-8"

## Remarks

Specifies the session character set for decoding character data transferred from the IMAP server. The default value is UTF-8. Set it to 'None' to leave the charset unspecified. This can resolve some timeout errors that can be caused by an incorrect charset configuration.

## Other

These hidden properties are used only in specific use cases.

### Data Type

string

### Default Value

""

## Remarks

The properties listed below are available for specific use cases. Normal driver use cases and functionality should not require these properties.

Specify multiple properties in a semicolon-separated list.

## Integration and Formatting

|                       |  |
|-----------------------|--|
| DefaultColumnSize     | Sets the default length of string fields when the data source does not provide column length in the metadata. The default value is 2000. |
| ConvertDateTimeToGMT  | Determines whether to convert date-time values to GMT, instead of the local time of the machine.   |
| RecordToFile=filename | Records the underlying socket data transfer to the specified file.   |

## Readonly

You can use this property to enforce read-only access to IMAP from the provider.

## Data Type

bool

## Default Value

false

## Remarks

If this property is set to true, the adapter will allow only SELECT queries. INSERT, UPDATE, DELETE, and stored procedure queries will cause an error to be thrown.

## Timeout

The value in seconds until the timeout error is thrown, canceling the operation.

## Data Type

int

## Default Value

60

## Remarks

If Timeout = 0, operations do not time out. The operations run until they complete successfully or until they encounter an error condition.

If Timeout expires and the operation is not yet complete, the adapter throws an exception.

## UIDMode

If true, permanent message ids are used instead of the default temporary ids.

## Data Type

bool

## Default Value

false

## Remarks

The default value for UIDMode is false.

## UserDefinedViews

A filepath pointing to the JSON configuration file containing your custom views.

## Data Type

string

## Default Value

""

## Remarks

User Defined Views are defined in a JSON-formatted configuration file called *UserDefinedViews.json*. The adapter automatically detects the views specified in this file.

You can also have multiple view definitions and control them using the UserDefinedViews connection property. When you use this property, only the specified views are seen by the adapter.

This User Defined View configuration file is formatted as follows:

- Each root element defines the name of a view.
- Each root element contains a child element, called **query**, which contains the custom SQL query for the view.

For example:

```
{
  "MyView": {
    "query": "SELECT * FROM [INBOX] WHERE MyColumn = 'value'"
  },
  "MyView2": {
    "query": "SELECT * FROM MyTable WHERE Id IN (1,2,3)"
  }
}
```

Use the UserDefinedViews connection property to specify the location of your JSON configuration file. For example:

```
"UserDefinedViews",
"C:\\Users\\yourusername\\Desktop\\tmp\\UserDefinedViews.json"
```

# TIBCO Product Documentation and Support Services

---

For information about this product, you can read the documentation, contact TIBCO Support, and join the TIBCO Community.

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The [TIBCO Product Documentation](#) website is updated frequently and is more current than any other documentation included with the product.

## Product-Specific Documentation

The following documentation for this product is available on the [TIBCO® Data Virtualization](#) page.

- **Users**
  - TDV Getting Started Guide
  - TDV User Guide
  - TDV Web UI User Guide
  - TDV Client Interfaces Guide
  - TDV Tutorial Guide
  - TDV Northbay Example
- **Administration**
  - TDV Installation and Upgrade Guide
  - TDV Administration Guide
  - TDV Active Cluster Guide
  - TDV Security Features Guide
- **Data Sources**

TDV Adapter Guides

TDV Data Source Toolkit Guide (Formerly Extensibility Guide)

- **References**

TDV Reference Guide

TDV Application Programming Interface Guide

- **Other**

TDV Business Directory Guide

TDV Discovery Guide

- *TIBCO TDV and Business Directory Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

## How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on the website.

## Release Version Support

TDV 8.5 is designated as a Long Term Support (LTS) version. Some release versions of TIBCO® Data Virtualization products are selected to be long-term support (LTS) versions. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also

[https://docs.tibco.com/pub/tdv/general/LTS/tdv\\_LTS\\_releases.htm](https://docs.tibco.com/pub/tdv/general/LTS/tdv_LTS_releases.htm).

## How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, visit [TIBCO Community](#).

# Legal and Third-Party Notices

---

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, TIBCO logo, TIBCO O logo, ActiveSpaces, Enterprise Messaging Service, Spotfire, TERR, S-PLUS, and S+ are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the



readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2002-2023 Cloud Software Group, Inc All Rights Reserved.