



WHAT'S NEW AND RELEASE NOTES

Release: 2021.1



IN THIS RELEASE

- NEW FEATURES AND IMPROVEMENTS..... 2**
- DATA PREPARATION..... 2
 - Save data connections without testing it 2
- STORYBOARDS..... 3
 - Load data incrementally in Table 3
 - Filter published and embedded storyboards with query parameters..... 4
- INSTALLATION & CONFIGURATION..... 6
 - Upload TLS/SSL certificates for secure web connectors and data connections 6
 - Upload TLS/SSL certificates for secure LDAP connections 7
 - Use Personal Token authentication for Tableau web connector 8
 - Configure revision history limit for a Portals application 9
 - Manage authentication for Kubernetes during installation 10

NEW FEATURES AND IMPROVEMENTS

DATA PREPARATION

Save data connections without testing it

Previously, you could save a data connection only if it passed the connection test successfully. Starting from this release, you can save a data connection even if the connection cannot be established. This may be useful when you don't have the proper credentials at that time or need to verify other connection details. However, to save a connection, you need to complete all the fields.

MICROSOFT SQL SERVER

Connection details Caching

Username and Password

Server
10.15.16.184

Port
1433

Database
airlines

Username
airlines_admin

Password
.....

Ask user for credentials
Off

Connection name
Microsoft SQL Server

Microsoft SQL Server
Microsoft SQL Server data source connection

Test your connection
After you fill in the data connection details, test your connection. If something goes wrong, an error message will be displayed.

SAVE TEST CONNECTION

STORYBOARDS

Load data incrementally in Table

Starting with this release, the **Table** widget can load data differently depending on your needs—load all the data at once or load the rows incrementally.

By default, all the data rows are loaded at once, limited by the value in the **Maximum data rows** field. If you have a large dataset with many rows, loading all of them at once may take some time and make your interaction with the Table inconvenient. In this case, you can choose to load data incrementally (server-side), where each set of data (increment) is queried separately. The number of rows to load is defined in the **Incremental load** field, set to 50 by default.

The incremental data loading improves performance but has some nuances:

- The Totals row appears on each page, showing the sum of the data that is already loaded. For example, if the incremental load is 50 rows, you will see the total for the first 50 rows, then for 100, and so on.
- The pagination bar shows the page total number after you load all the data.
- Grouping and pivoting are not supported.

The screenshot shows the configuration panel for the Table widget on the left and the resulting data table on the right. The configuration panel includes:

- Table** widget header with a plus sign.
- Data** source selection with a plus sign.
- Data view options** section with a minus sign.
- Minimized SQL** toggle set to **On**.
- Data loading** dropdown menu set to **Incrementally**.
- Incremental load** spinner set to **50** rows.
- Maximum data rows** spinner set to **10000**.
- VISUALIZE** button at the bottom.

The data table displays the following columns: Customers, Quantity, Unit Price, and Items per Transaction. The rows are color-coded in a gradient from blue to green to brown. A totals row is visible at the bottom of the table.

Customers	Quantity	Unit Price	Items per Transaction
40.00	56.00	230.00	49.00
47.00	79.00	928.00	59.00
49.00	70.00	317.00	43.00
62.00	120.00	1.32k	58.00
91.00	110.00	563.75	107.00
124.00	65.00	1.08k	93.00
131.00	158.00	1.23k	85.00
181.00	37.00	161.83	295.00
255.00	76.00	1.1k	90.00
291.00	35.00	373.64	421.00
292.00	124.00	372.60	460.00
351.00	46.00	169.93	346.00
402.00	88.00	1.28k	85.00
459.00	274.00	385.38	214.00
668.00	250.00	59.29	167.00
3.88k	1.76k	176.12	1.32k
6.18k	518.00	1.11k	192.00

Filter published and embedded storyboards with query parameters

Now, you can filter the embedded and published storyboards by using query parameters. To do so, add a filter through the **Filter** widget and then specify a parameter name in the **Parameter** field.

FILTERS ? ×

+ NEW ▾

Name	Dataset	Column	Type
<input checked="" type="checkbox"/> Store Type	Sample - World Sales	Store Type	Multi Select
<input checked="" type="checkbox"/> Department	Sample - World Sales	Department	Multi Select

Dataset: ▾

Column: ×

Type: ▾

Area × Single Value ×

Name:

Parameter:

CLOSE
APPLY

Warehouse
259,836

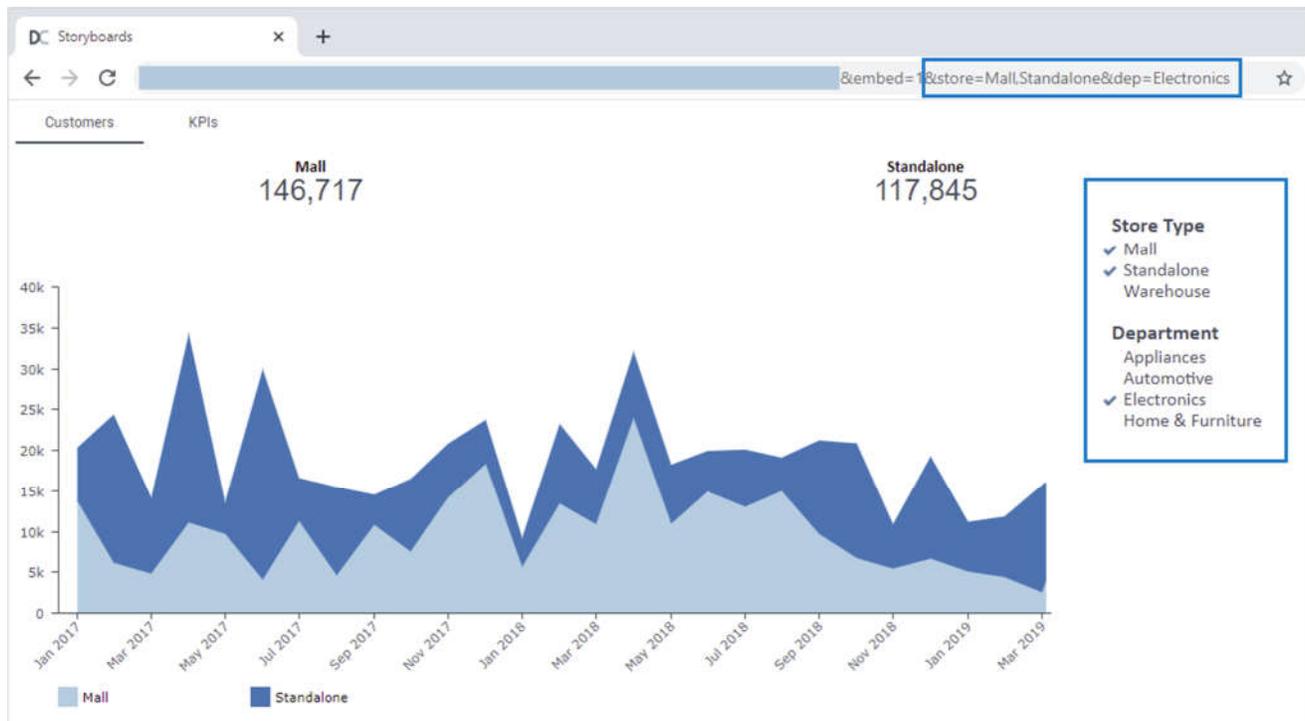
Store Type

- Mall
- Standalone
- Warehouse

Department

- Appliances
- Automotive
- Electronics
- Home & Furniture

After generating the URL for a storyboard, you can add the defined parameter name with one or multiple values to the URL as needed. For example, if you add “&store=Mall,Standalone&dep=Electronics” to the published storyboard URL, the visualization shows data only for the Mall and Standalone store types with the Electronics department.



When using query parameters, consider the following:

- Values for the filter parameter should be the same as listed in the Filter widget.
- If using multiple values, separate them by a comma. For example, “&city=London,New%20York”.
- Each parameter starts with an “&” sign. For example, “&store=Mall&dep=Electronics”.
- If the values have spaces or other special symbols, convert them appropriately for the URL encoding. For example, “&city=New%20York”, where a space is specified as %20.
- Filtering applies to all the widgets in a storyboard where filters use the same parameter name.

INSTALLATION & CONFIGURATION

Upload TLS/SSL certificates for secure web connectors and data connections

When the applications need to communicate with other secure servers, it is required to import the third-party certificates into the Java certificate stores for the respective docker containers.

Now, administrators can upload and manage TLS/SSL certificates to secure connections for web connectors and any data connections right from Configuration Manager. In the **TLS/SSL Certificates** section, go to the **Data Connectors** page and click **Upload certificate**. After adding a valid and trusted certificate file (.crt or .cer), click **Restart** on the toolbar and proceed with the secure connectors in the platform.

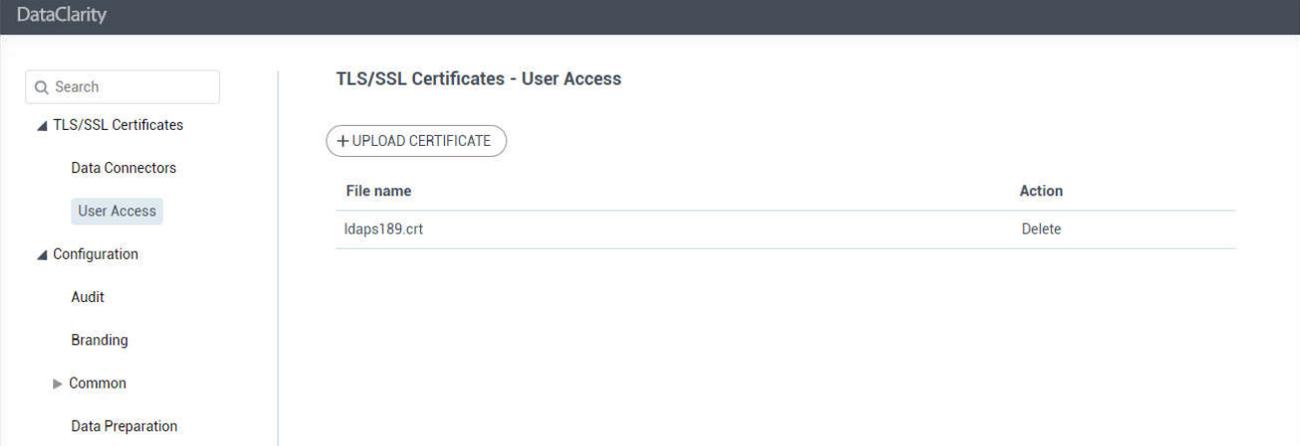
The screenshot displays the DataClarity Configuration Manager interface. On the left is a navigation sidebar with a search bar and menu items: TLS/SSL Certificates (expanded), Data Connectors (selected), User Access, Configuration (expanded), Audit, Branding, Common, and Data Preparation. The main content area is titled "TLS/SSL Certificates - Data Connectors" and features a "+ UPLOAD CERTIFICATE" button. Below this is a table with two columns: "File name" and "Action".

File name	Action
tab189.crt	Delete
cog224.crt	Delete

Upload TLS/SSL certificates for secure LDAP connections

When Access Manager needs to communicate with a secure LDAP server to work with users' lists, it is required to import the third-party certificate into the Java certificate store for the "user access" docker container.

Administrators can now upload and manage TLS/SSL certificates for secure LDAP connections right from Configuration Manager. In the **TLS/SSL Certificates** section, go to the **User Access** page and click **Upload certificate** and select a valid and trusted certificate file (.crt or .cer). After adding a valid and trusted certificate file (.crt or .cer), click **Restart** on the toolbar and proceed with the secure LDAP connection in the platform.



The screenshot displays the DataClarity configuration interface. On the left is a navigation sidebar with a search bar and a tree view containing: TLS/SSL Certificates (expanded), Data Connectors, User Access (selected), Configuration (expanded), Audit, Branding, Common, and Data Preparation. The main content area is titled "TLS/SSL Certificates - User Access" and features a "+ UPLOAD CERTIFICATE" button. Below this is a table with two columns: "File name" and "Action". The table contains one entry: "ldaps189.crt" with a "Delete" action link.

File name	Action
ldaps189.crt	Delete

Use Personal Token authentication for Tableau web connector

If Tableau is configured to use SAML authentication, it cannot authenticate users by credentials. In this case, you can use the Personal Access Token authentication—instead of the username and password, you specify the user’s token name and secret.

The **Personal Access Token authentication** option is now available in the **Web Connectors** section, under **Tableau Software**. If you choose the token authentication, enter the user’s token name and secret as generated in a user’s account settings in Tableau. For details, see [Manage Your Account Settings \(Tableau\)](#).

In the **Token username** field, you may enter the username or email address associated with the token. This way, when the token expires, you can easily find the user for whom you need to regenerate it. Depending on the personal access token usage, it expires in 15 days (if not used during this period) or in 1 year (if used regularly). You can also change the default refresh token expiry time span in Tableau. For details, see [Personal Access Tokens \(Tableau\)](#).

The screenshot shows the DataClarity configuration interface for Tableau Software. The left sidebar contains a search bar and a navigation menu with categories like TLS/SSL Certificates, Configuration, Audit, Branding, Common, Data Preparation, Data Science, Data Server, Notifications, Portals, Storyboards, User Access, Web Connectors, and Cognos Analytics. The 'Tableau Software' section is selected under Web Connectors. The main content area contains the following fields:

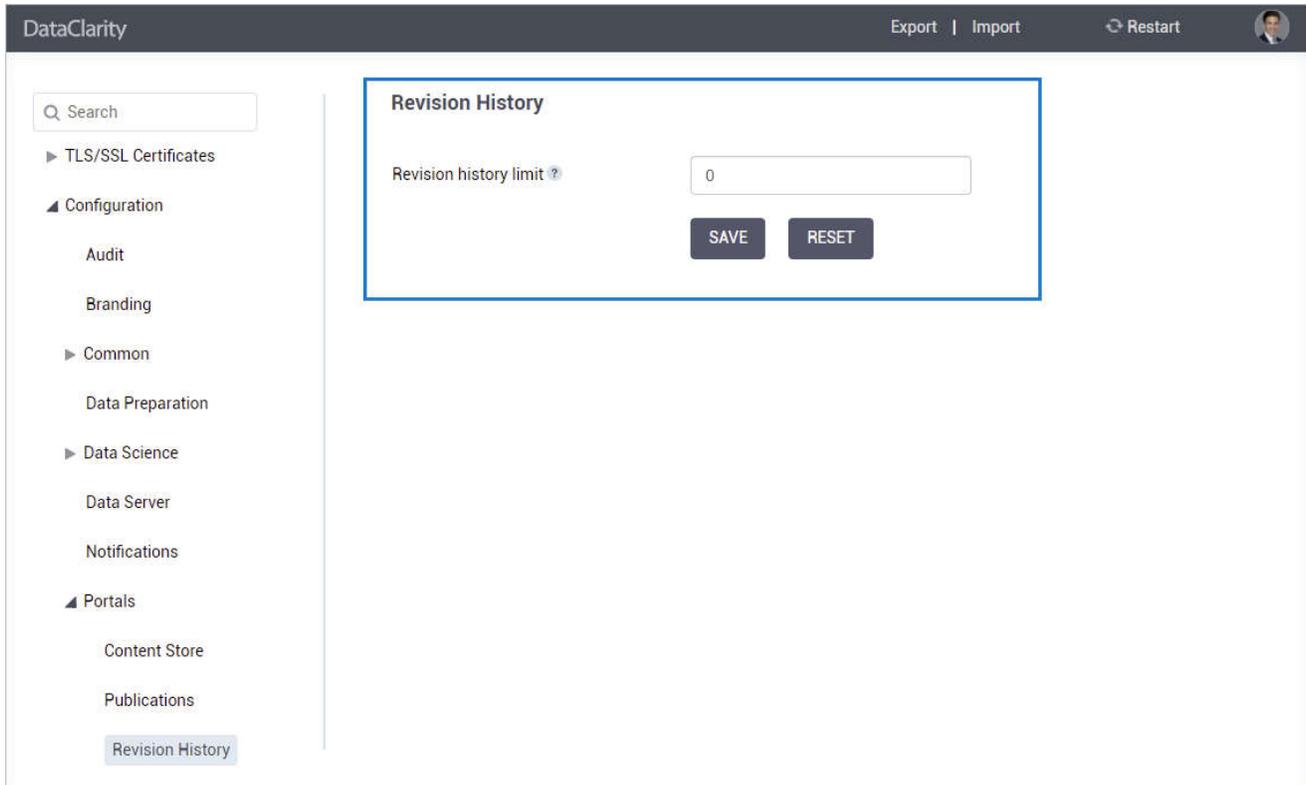
- Storyboard widget: ON
- Host: https://dc-rnd-050
- Site name: none
- API version: 3.10
- Username: dragos
- Password: [masked]
- Personal Access Token authentication: ON (highlighted in blue)
- Token username: [empty field]
- Token name: [empty field]
- Token secret: [empty field]

At the bottom of the highlighted section are 'SAVE' and 'RESET' buttons.

Configure revision history limit for a Portals application

Administrators can now control the number of portal revisions available as backups in Configuration Manager. In the **Revision history limit** field, specify the maximum number of portal revisions. By default, the revision history limit is set to **0**. This means that the system does not keep any backups, and by saving a Portals application, you overwrite the current version. If the limit is **2**, the system keeps two backup versions that you can restore.

To change the revision history limit, go to the **Portals** section, and open the **Revision History** page.



The screenshot displays the DataClarity Configuration Manager interface. The top navigation bar includes the DataClarity logo, 'Export | Import' links, a 'Restart' button, and a user profile icon. A left-hand sidebar contains a search bar and a list of configuration categories: TLS/SSL Certificates, Configuration, Audit, Branding, Common, Data Preparation, Data Science, Data Server, Notifications, Portals, Content Store, Publications, and Revision History. The 'Revision History' page is highlighted in the sidebar. The main content area features a 'Revision History' section with a text input field labeled 'Revision history limit' containing the value '0'. Below the input field are two buttons: 'SAVE' and 'RESET'.

