TOC

Logi Symphony 24

Manage Symphony 24 in Visual Data Discovery

Quick Start Reference - Manage Symphony in Visual Data Discovery

This applies to: *Visual Data Discovery*

Managing Symphony covers multiple areas:

Trusted Access

- Trusted Access
- Trusted Access API Endpoints
- Trusted Access Client Properties

Single Sign On, Authentication, CORS

- Symphony Single Sign On
- Create And Use Logon And Session Tokens For Symphony
- Supported Authentication Tools
- Authorize Symphony Access
- Configure Kerberos Single Sign-On (SSO) Settings
- Kerberos Authentication For Connectors
- Enable Visual Data Discovery Component Access From Other Sites Using Cross-Origin Resource Sharing (CORS)

Groups

- Manage User Groups In Visual Data Discovery
- About Supplied Groups
- Group Privilege Reference

User Auditing

- User Auditing
- Enable And Configure User Auditing
- Enable User Audit Data For Symphony Accounts
- Consuming Audit Data
- User Audit Events
- User Auditing For Multi Tenancy Environments
- Enable User Audit Data for Multi Tenant Accounts

Tenants and User Management

- Manage Symphony Tenants
- About the Supplied Symphony Tenants
- List and Review Tenants
- Modify Tenants

- Add New Users
- Manage Users in Symphony
- Advanced User Management
- Add and Remove Members of a Group
- Supplied Users and User Groups
- Modify Users
- Assign and Remove Users in Tenants
- Set the Current Tenant for a User
- Specify Custom User Attributes
- Specify A User's Regional Settings

Activity Logging

- Manage Activity Logs
- Activity Logging
- Activities Log Reference Sheet

Trusted Access

This applies to: *Visual Data Discovery*

Symphony provides its own security methodology that allows for machine-to-machine authorization of Symphony resources when embedded in your application (the “parent” application). This is a form of “delegated” authorization where the parent application can determine, on demand, how and when to authorize any given embedded Symphony component to an end-user logged into the parent application. This methodology is called *Trusted Access*.

Note: insightsoftware recommends using Trusted Access for all embed-related workflows.

Similar to "single sign-on," this arrangement allows users to log in once to the parent application and yet have their security information propagated to Symphony, creating a seamless and secure user experience. This, of course, means that users can't be allowed to "go around" the parent application and directly access Symphony. In the stateless world of web applications, this requires some special mechanisms to ensure security that are provided for applications through our SecureKey technology.

On request from the parent application, Trusted Access provides a user access token with defined authorization rules that account for user privileges, object permissions, security filters and any specific user attributes used in interpolation. This user access token can then be used in the parent application to serve any Symphony specific embedded components such as dashboards for the respective user. For information on how tokens are initiated and requested in your applications, see Embed ComposerSymphony Components Using JavaScript And Trusted Access.

Note: In environments where you use Typescript for your client side code, you can use Embed Manager as an npm package. See https://www.npmjs.com/package/logi-embed.

Trusted Access tokens are encrypted when stored in Symphony metadata. The encryption mode used can be set as described in Change The Encryption Mode.

This topic also describes:

- Trusted Access Prerequisites
- Trusted Access Recommendations
- Register A Client
- Generate A User's Access Token

The following additional topics provide reference information:

- Trusted Access API Endpoints
- Trusted Access Client Properties

Trusted Access Prerequisites

- Every end user must have Symphony user account defined, unless you are using LDAP autoprovisioning with Symphony. See Manage Users in Symphony.
- Trusted Access is enabled by default. If it is disabled, enable Trusted Access by selecting the **Trusted Access** option on the Security page. See Enable Trusted Access.

Trusted Access Recommendations

For security reasons, we recommend that you use short-lived tokens. Tokens that are valid for less than 10 minutes are recommended. The validity time of a user access token is defined when you register a client with Symphony.

Important: If you disable Trusted Access in a multi-tenancy environment, data retrieved through the Data Discovery connector you created is fetched using the tenant admin's credentials instead of each user's credentials. Effectively, all row level security settings are removed on the data retrieved.

Register a Client

To start using Trusted Access, you first need to register your application, as Symphony refers to it, as a *client*.

Registering a client will generate a client ID and client secret. These credentials can then be used to generate user access tokens for any user in the Symphony platform, as needed.

To register your application as a client, POST the `/api/trusted-access/clients` API endpoint. You can also patch, delete, and list Trusted Access clients using the `/api/trusted-access/clients` API endpoint. See Trusted Access API Endpoints.

Generate a User's Access Token

To generate a user's access token, pass the client ID and client secret to HTTP BasicAuth. To obtain the client ID and client secret, use the `/api/trusted-access/clients` API endpoint. See Trusted Access API Endpoints.

Generate a User's Access Token for Existing Symphony Users

```
/********REQUEST TRUSTED ACCESS TOKEN ********/
const AccessToken = (ComposerUrl, Username, callback) => {
	var Client = GetClient();
	if(typeof Client === 'undefined' ||  Client === null)
		callback({ErrorMessage: 'Client Not Found', status : 500});
	else {
		var BasicAuth = Buffer.from(`${Client.client_id}:${Client.client_secret}`).toString('base64');
		Post(BasicAuth, `${ComposerUrl}/api/trusted-access/pull/tokens`, { "username": Username }).then((result) => {
			if(JSON.stringify(result).indexOf('error')>-1)
				callback(result, null);
			else
				callback(null, result);
		});
	}
};
```

Note: You can only generate tokens for regular users and for administrators.

Generate a User's Access Token for New Symphony Users

```
/********REQUEST TRUSTED ACCESS TOKEN ********/
const UserContext = {
	"username": "joe",
	"account": "company",
	"fullname": "Example Inc",
	"email": "joe@example.inc",
	"groups": ["Store Manager", "Cashier"],
	"attributes": [{"key": "city", "values": ["London"]}]
};
const AccessToken = (ComposerUrl, UserContext, callback) => {
	var Client = GetClient();
	if (typeof Client === 'undefined' || Client === null)
		callback({ErrorMessage: 'Client Not Found', status: 500});
	else {
		var BasicAuth = Buffer.from(`${Client.client_id}:${Client.client_secret}`).toString('base64');
		Post(BasicAuth, `${ComposerUrl}/api/trusted-access/push/tokens`, UserContext).then((result) => {
			if (JSON.stringify(result).indexOf('error') > -1)
				callback(result, null);
```

```
                        else
                                callback(null, result);
                });
        }
};
```

Note: You can only generate tokens for regular users and for administrators.

Trusted Access API Endpoints

This applies to: *Visual Data Discovery*

The following API endpoints can be used to manage Trusted Access.

Endpoint	Method	Description
`/api/trusted-access/clients`	GET	Returns all the Trusted Access client information in the metadata. Included with this information is the access token validity time (in seconds), client ID, client name, client secret expiration time (in seconds), and the token authentication method. For a description of these, see Trusted Access Client Properties.
`/api/trusted-access/clients`	POST	Creates a Trusted Access client. The request must specify the number of seconds for which the access token is valid and the client name. The client name must be unique. When you create the client, the client ID, client secret, secret expiration time, and the token authentication method are automatically generated.
`/api/trusted-access/clients/<id>`	GET	Returns the Trusted Access client information for a specific client. The request must specify the client ID.
`/api/trusted-access/clients/<id>`	DELETE	Deletes a specific Trusted Access client. The request must specify the client ID.
`/api/trusted-access/clients/<id>`	PATCH	Updates the Trusted Access client information for a specific client. The request must specify the client ID and the number of seconds for which the access token is valid.
`/api/trusted-access/pull/tokens`	POST	Use pull to request a user access token for users that already exist in Symphony. The user must already exist, and have an active Symphony user account (unless you are using LDAP with automatic provisioning for Symphony). You can't update user context such as user attributes or groups using `pull`.
`/api/trusted-access/push/tokens`	POST	Use push to request a user access token for new and existing users by sending their context to Symphony. Existing users are updated if their context has changed. Context must contain `username`, and `account`. It can optionally include `email`, `fullname`, `groups` and other individual `attributes` Several reserved keys are restricted from use in Symphony as custom user attributes: `composerUserName` and `accountId`. If you push these reserved keys, a 400 Bad Request

Endpoint	Method	Description
		error is returned via API.

Trusted Access Client Properties

This applies to: *Visual Data Discovery*

The following table describes the Trusted Access client properties. All of this information is encrypted when it is stored in the Symphony metadata store, except the client secret, which is hashed using BCrypt.

Endpoint	Description
`client_name`	The human-readable string name of the client application. The client name must be unique.
`client_id`	The client ID issued to the client when the client is registered with Trusted Access. This is generated by the system and cannot be modified.
`client_secret`	The client secret string issued to the client when the client is registered with Trusted Access. This value is used by applications to authenticate to the token endpoint. The client secret is not available after it is initially created -- developers must store it locally in order to retain it.
`access_token_validity_seconds`	The validity (in seconds) of the access token. Short-lived tokens (less than 10 minutes) are recommended.
`client_secret_expires_at`	The time (in seconds) at which the `client_secret` will expire. A value of zero (0) means that the client secret never expires.
`token_endpoint_auth_method`	A string indicator of the authentication method used for the token endpoint. For Trusted Access, this value is always "`client_secret_basic`", which means that the client must always use HTTP basic authentication to request an access token.

Symphony Single Sign On

This applies to: *Managed Dashboards, Managed Reports*

Single sign-on (SSO) allows users to log in once on their workstation, and gain access to multiple systems without being prompted to log in again. This article provides guidelines and summarizes how to use SSO within the Symphony application.

There are multiple ways to accomplish single sign-on:

- Federated authentication: see Enabling Federated Authentication.
  - Federated authentication behaves as Single Sign On (SSO), enabling the user to access multiple services without the need for further authentication.
  - Authentication is possible using SAML 2.0, OpenID Connect (OIDC), Azure Active Directory/Microsoft Identity, JWT, and other protocols.
  - Automatic logon with federated authentication can be accomplished by setting the custom logon page configuration setting to the authentication URL: `https://yourinstance/AuthBridge/Auth/ExternalAuth`

- Anonymous logon: see How To Enable Anonymous Log On.
  - Use to provide access to some or all Symphony Managed content without any users needing to take any action to log on.
  - This is ideal for implementation of a public dashboard/reporting site or kiosk.
  - Users are automatically logged onto a specific account specified in the application configuration settings **Anonymous User Name** and **Anonymous Password**.

- Embedding Symphony content: see Create And Use Logon And Session Tokens For Symphony.
  - Your application can use optionally use Symphony's API to create one-time tokens to log in users automatically and securely.
  - Create and pass a one time **logon token** to create a session for a specified user and log them on.
  - Log on through the API and create a session for the user, optionally setting up session properties such as custom attributes, and create a one-time **session token** the user can use to access that session.
  - Use server-side code to create these one-time time-limited tokens and pass them to your client application's JavaScript or HTML.

    Note: You can also use the federated authentication feature to authenticate using JSON Web Tokens (JWTs). These are different from Symphony logon and session tokens.

Supported Authentication Tools

This applies to: *Visual Data Discovery*

Symphony supports several approaches to authenticating users. Your organization must choose the best approach given your existing constraints and objectives.

- Symphony provides basic login access to the Symphony application. See Authorize Symphony Access.
- Trusted Access can be used to allow for machine-to-machine authorization of Symphony resources when embedded in your application. It allows users to log in once to the parent application and yet have their security information propagated to Symphony, creating a seamless and secure user experience. See Trusted Access.

> **Note:** insightsoftware recommends using Trusted Access for all embed-related workflows.

Symphony system administrators can enable or disable Symphony's authentication services as required. The available services are listed on the Security Services tab.

View and edit security options

1. Log in as a system admin.
2. Select **Tools > Security** from the main menu.

   The Security page appears. It consists of several sections: **Security Services**. The Security Services tab is selected. Other tabs shown are accessible only when the corresponding service is enabled on the Security Services tab.

Enabling or disabling any of these security services requires a restart of the Symphony service. Basically, any time you switch a security feature, the Symphony service needs to be restarted before the change takes effect. The following switch status may appear for each of the authentication services: **Started**, **Stopped**, **Will start or stop on next restart**.

When working with security authentication services, bear in mind that you cannot use them all at the same time. If you switch a particular security service on, others will become disabled. If you want to use a security service that is disabled, you must switch the running services off and then start the service you want.

Security services compatibility

Security Service	Can Be Used With
SAML SSO	Trusted Access
Trusted Access	Kerberos, x.509

Configure Client Certificate Authentication

This applies to: *Visual Data Discovery*

Symphony supports X.509 client certificate authentication. However, note that auto-provisioning of user accounts is not available for client certificate authentication.

To use the X.509 authorization you need to:

- Enable the X.509 option in the Security Services section
- Configure the required properties in the `zoomdata.properties` file

Caveat

Symphony does not support auto-provisioning of user accounts for client certificate authentication.

Configuration Steps

For guidance on accessing and editing a Symphony property file, refer to the topic Configure ComposerSymphony.

Add the following settings to your `zoomdata.properties` file:

```
server.port= 8443
server.ssl.enabled= true
server.ssl.client-auth= want
server.ssl.key-store= .../server.jks
server.ssl.key-store-password= <Your_password>
server.ssl.key-store-type= <use_either_jks_or_pkcs12>
server.ssl.trust-store= .../truststore.jks
server.ssl.trust-store-password=<Your_password>
server.ssl.trust-store-type= <use_either_jks_or pkcs12>
```

For each user, create an user account in Symphony with the username set to the 'CN' in the user's certificate.

Troubleshooting

Challenges you may run into:

- User is never prompted to select a certificate:
  - Make sure you have added at least one CA to the trust-store file.
  - Verify `server.ssl.client-auth` is set to `want`.
- Selecting login brings me back to the login page:
  - Make sure the username matches the CN of the certificate being used.
  - Make sure the client certificate is signed by a CA in the trust-store.

For further troubleshooting assistance, contact Technical Support.

Configure Kerberos Single Sign-On (SSO) Settings

This applies to: *Visual Data Discovery*

Kerberos is an enterprise authentication protocol that uses the concept of tickets and three-way authentication to enable users and computers to identify themselves and secure access to resources.

Symphony supports Kerberos as a Single Sign On (SSO) mechanism. Using Kerberos SSO, users can seamlessly log into Symphony and administrators can completely externalize and centrally manage users or group memberships using their existing Kerberos infrastructure. You can learn more about Kerberos here.

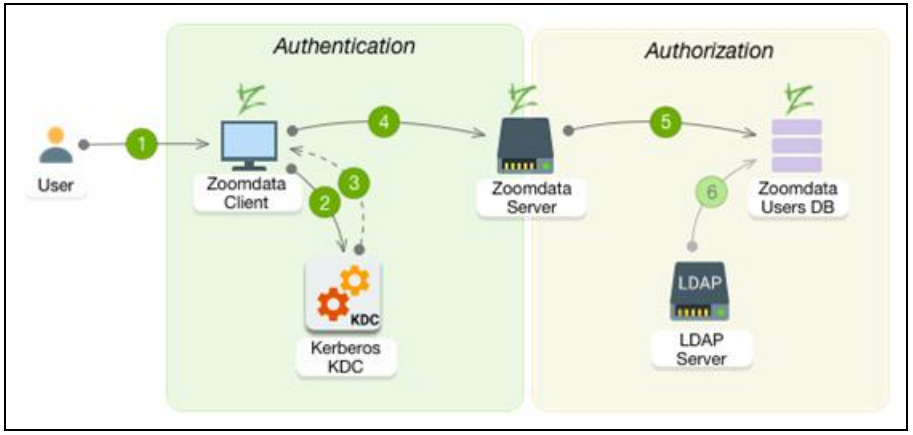
How It Works

When using Kerberos with Symphony, the workflow is as follows:

1. A user logs in to his company domain (for example, logging into his windows workstation) and is authenticated with Kerberos. In the case of Windows environments, this is likely Active Directory.

2. In a browser window, a user visits the Symphony application and Symphony then leverages the user's Kerberos identity to automatically log them into Symphony. If this is a users first visit to Symphony, then Symphony will auto-provision them as a new user in the Symphony environment.

3. Kerberos authentication is often paired with LDAP to look up a user's authorization or group membership. Symphony will look up the user's group membership.



Prerequisites

Before you start configuring settings for Symphony to use Kerberos, you must:

- Create a Kerberos Principal. For more information on how to create a Kerberos Principal, refer to Kerberos documentation.
- Create the keytab file on the instance where KDC runs and upload it to Symphony Server.

Note: If you have Kerberos enabled, you must use a static context path for the Symphony login URL. Multiple or dynamic context paths are not supported at this time.

How to Generate the Keytab File

To generate the `.keytab` file for existing AD users, run the following command on AD server:

```
ktpass
-out <file_name>.keytab
-princ <HTTP/zoomdata_host> @<realm>
-mapUser <user_name>
-pass <user_password>
-crypto RC4-HMAC
-pType KRB5_NT_PRINCIPAL
```

To generate the `.keytab` file for existing MIT LDAP user, run the following command on LDAP server side:

```
$kadmin
kadmin: xst -e "rc4-hmac" -k <file_name>.keytab <HTTP/zoomdata_host> @<realm>
```

Configure General Settings

To configure the Kerberos settings for Symphony Server, complete the following steps:

1. Enable the Kerberos SSO service on the Security Services tab. Keep in mind, that if you have SAML or x509 authentication enabled, you will have to disable them first to use Kerberos.
2. Restart the Symphony server by running the following command:

   ```
   sudo service zoomdata restart
   ```

3. Log in as a system admin or a member of the Supervisors group and select **Tools > Security**. Navigate to the **Kerberos Settings** tab.

4. Slide the **Enable Kerberos** switch on (to the right).
5. Specify the **Kerberos Service Principal**.
6. Select **Upload Kerberos Keytab File** and upload the `.keytab` file, that you have generated before.
7. Select the **Include Kerberos realm/domain name in auto provisioned Symphony username** if you want to have the user name in the following format: `username@realm`.
8. Save your settings.

Configure the Settings on the Client Side

Perform the steps listed below on the client instance that will connect to kerberized Symphony.

1. Install the Kerberos command line tools:

```
sudo yum install krb5-workstation
sudo yum install krb5-libs
```

2. Navigate to the `krb5.conf` file and specify the host on which Symphony server runs:

```
[libdefaults]
default_realm = ZOOMDATA.LOCAL
[realms]
ZOOMDATA.LOCAL = {
kdc = <composer_host>.local
admin_server = <composer_host>.local
}
[logging]
default = FILE:/var/log/krb5.log
kdc = FILE:/var/log/krb5.log
[domain_realm]
.zoomdata.local = ZOOMDATA.LOCAL
zoomdata.local = ZOOMDATA.LOCAL
```

3. List all the Kerberos tickets available on the current instance:

```
klist
```

4. Remove all the Kerberos tickets (if there are any):

```
kdestroy -A
```

5. Obtain a Kerberos ticket for a user (the realm is not required if there is a default one specified in krb5.conf):

```
kinit user@ZOOMDATA.LOCAL
```

6. Configure your browser to support Kerberos SSO to Symphony.

Note: To authenticate a user with a Kerberos ticket in Symphony, you must either enable LDAP autoprovisioning or create a user with the same name in Symphony.

Kerberos Authentication for Connectors

This applies to: *Visual Data Discovery*

Kerberos is an enterprise authentication protocol that uses the concept of tickets and three-way authentication to enable users and computers to identify themselves and secure access to resources. Kerberos support does not apply to some connectors.

Support for this feature by connector is shown in the following table.

Key:Y - Supported; **N** - Not Supported; N/A - not applicable

Connector	Supported?	Notes
Amazon Redshift	**N**	
Amazon S3	**N**	
Apache Drill	**Y**	
Apache Phoenix	**Y**	Apache Phoenix supports Kerberos, but Apache Phoenix Query Server does not. For more information, see Enable Kerberos Authentication For Apache Phoenix Connectors.
Apache Phoenix Query Server (QS)	**N**	
Apache Solr	**Y**	
BigQuery	**N**	If you need to access a BigQuery partition, explicitly include an alias for the built in partition column in your select clause, such as `select *, _PARTITIONTIME as pt from projectId.datasetId.tableId`.
Cloudera Impala	**Y**	
Cloudera Search	**Y**	
Couchbase	N/A	
Dremio	**N**	
Elasticsearch 7.0	**N**	
Elasticsearch 8.0	**N**	
File Upload	**N**	
HDFS	**Y**	
Hive	**Y**	
Jira	**N**	
MemSQL	**N**	
Microsoft SQL Server	**N**	
MongoDB	**N**	

Connector	Supported?	Notes
MySQL	**N**	
Oracle	**N**	
PostgreSQL	**N**	
Python	**N**	
Real Time Sales	N/A	
Salesforce	**N**	
SAP Hana	**N**	
SAP S/4HANA	**N**	
SAP IQ	**Y**	
Spark SQL	**Y**	To enable Kerberos authentication for Spark SQL connectors, see Connect To Spark SQL Sources On A Kerberized HDP Cluster.
Snowflake	**N**	
Teradata	**N**	
TIBCO DV	**N**	
Trino	**N**	
File Upload (Upload API)	**N**	
Vertica	**N**	

Enable Visual Data Discovery Component Access From Other Sites Using Cross-Origin Resource Sharing (CORS)

This applies to: *Visual Data Discovery*

Cross-origin resource sharing (CORS) is a standard introduced in HTML 5 that allows web applications to use HTTP headers to specify which origins are permitted to request resources on the server. Cross-origin resource sharing provides a web application access to selected resources running at a different location. Modern web browsers will consult the Access-Control-Allow-Origin header on how to relax the same origin policy for a given page. By default, this setting was disabled starting version 1.5.0SR1 due to security vulnerability concerns. However, CORS can be enabled for certain or all domains by editing the `zoomdata.properties` file (located in `/etc/zoomdata`). For more information about CORS, see https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS.

Modify CORS request permissions

1. From your terminal, open a command line session.
2. Connect to your Symphony server via a command prompt.
3. Use the following command to access and open the `zoomdata.properties` file:

   ```
   vi /etc/zoomdata/zoomdata.properties
   ```

4. Add the following variables to the file on new lines:

   ```
   http.response.header.content-security-policy.frame-ancestors=<source1> <source2>
   access.control.allow.origin=<origin1>,<origin2>
   ```

   Replace `<source>` with the actual sources that may embed resources. The default is an asterisk (`*`), or all sources. For more information about the Content-Security-Policy (CSP) frame-ancestors directive, see https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/frame-ancestors.

   Replace `<origin1>` and `<origin2>` with your specific URI including `http` or `https` as appropriate. Use `*` as a wildcard instead of a specific origin, thereby allowing any origin to access the resource, keeping in mind that this is a potential security vulnerability and may not work with all browsers. Refer to your Tomcat documentation to verify the appropriate syntax to use.

5. Save the document using standard `vi` commands.
6. Restart Symphony microservices after making this change and also make sure to clear your browser cache. See Restart ComposerSymphony Microservices.

Manage User Groups in Visual Data Discovery

This applies to: *Visual Data Discovery*

Create and update groups in the Groups work area.

Use groups to assign privileges to groups of users in data discovery. Groups are most useful when a number of users require the same access restrictions. Users can be assigned to multiple groups.

Note: SAML and LDAP groups that are automatically created in Symphony must be manually assigned privileges.

The main advanced task you can complete in data discovery instead of the Symphony Groups work area is assigning specific group privileges to your groups.

- System Administrators: Edit and update group privileges for groups you've created for Global Users. This does not include system-created groups, such as the administrators group.
- Tenant Administrators: Edit and update group privileges for groups created for your tenants. This does not include system-created groups, such as the administrators group.

See the following topics:

- About Supplied Groups
- Symphony User Groups
- Create And Edit Symphony Groups
- Advanced Group Management
- Group Privilege Reference

About Supplied Groups

This applies to: *Visual Data Discovery*

Several default groups are used in data discovery for working specifically with content in this module for your users. System admins belong to all of these groups by default, and can perform all of the tasks associated with these groups. You can't delete, rename, or edit the privileges of these default groups.

Add non-admin users to these groups to allow them to perform specific tasks for global users and tenant users when they are working in the data discovery module or with data discovery content.

- **Administrators** - This group is for administrator users. The default admin user and any system admins are members of this group. If your environment includes tenants, each tenant includes an administrators group: all tenant admins belong to that group. As a member of an administrators group, you can perform tasks to manage your environment, users, groups, and content.

- **Supervisors** - Add users as group members to allow them to perform specific functions without giving them access to all tasks members of the Administrators group can perform in the *Visual Data Discovery* tenant.

- **Content Distributors** - Add global users as group members to allow them to create, maintain, and distribute content to any tenant. Edit the user to enable the **VDD Content Distributor** toggle.

  Important: If you use the Content Distributors group, add them to another user group you define to give them access to the features you designate beyond Content Distributors.

Important: If you do not use multi-tenancy in Symphony, all users are part of *Global Users* by default (called the **Visual Data Discovery** tenant in the Data Discovery module). If your environment includes multiple tenants, users can be members of different groups in each tenant depending on your organization's needs.

Administrators Group (System Admins)

Administrators group members include the default admin user, who is a system administrator associated with the *Visual Data Discovery* (Global User in Managed Dashboards) tenant.

Assign users to the Administrators group to give them administrative privileges to perform actions such as:

- Create and manage users, system admins, and groups.

- Create and manage tenants.

- Manage connectors, licenses, and actions.
- Enable security privileges, and more.

Note: The Administrators group is an integral part of Symphony management. Global Users can be system administrators. Tenants also have Administrators group: tenant admins are members of that group for one or more tenants.

See Add New Users, and Symphony User Groups, and The Composer UI MenuSymphony Main Menu.

Administrators Group (Tenant Admins)

Administrators group members in tenants can:

- Create and manage users, groups, and content in their tenants.
- Define custom charts in their tenants.
- Perform Console and Actions related tasks.

Supervisors Group

The Supervisors group is designed to give you a group of users who can perform specific functions without giving them access to all tasks members of the Administrators group can perform in Visual Data Discovery.

All system admins are automatically added to this group. No further actions related to this group are needed to create system admins. Add non-admin users to this group if needed.

Content Distributors Group

All system admins are automatically added to this group. No further action is needed.

Add global users as group members to allow them to create, maintain, and distribute content to any tenant. Edit the user to enable the **VDD Content Distributor** toggle.

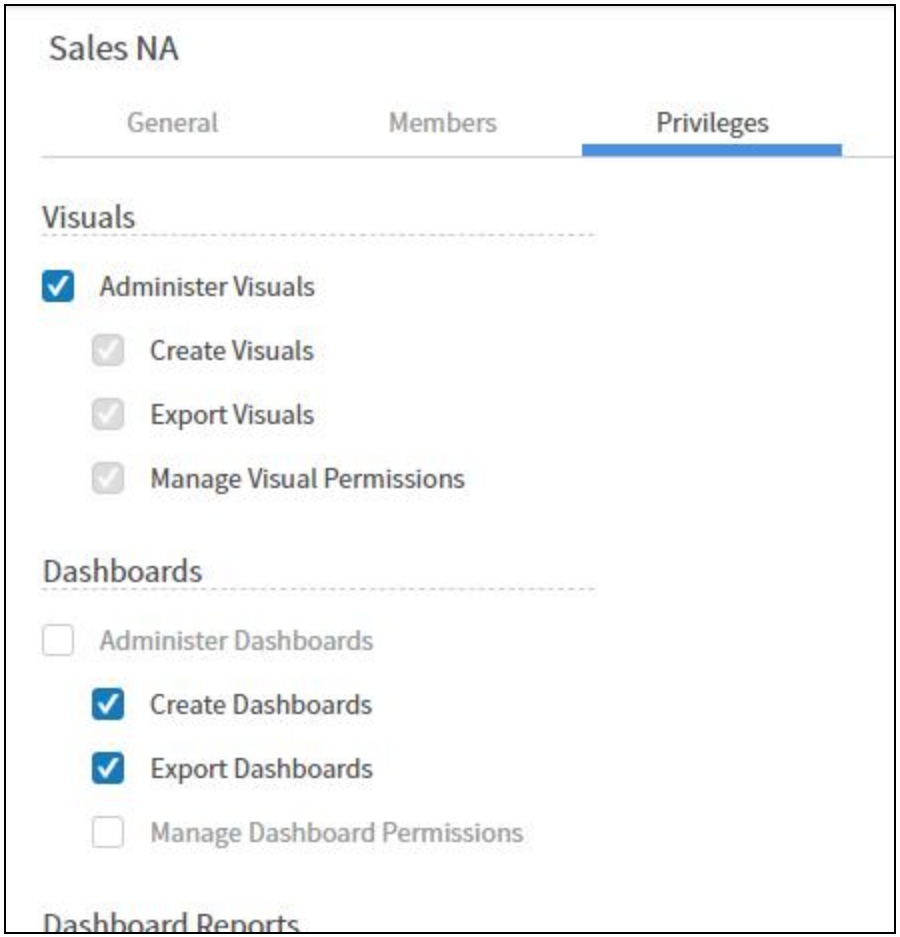
The Content Distributors group is part of the Visual Data Discovery tenant. Members can create, maintain, and distribute content to all tenants in your environment. Members of the Content Distributors group can:

- Import and export sources directly, or as part of importing and exporting dashboards.
- Import and export local and visual gallery visuals when importing and exporting dashboards.
- Import and export visual gallery visuals.
- Import and export dashboards.
- Import and export connections.

Group Privilege Reference

This applies to: *Visual Data Discovery*

Group privileges are specified on the **Privileges** tab for a group. Privileges allow a member of the Administrators group to grant permission to perform specific functions to all members of a group.



UI Privilege Name	Assign this privilege to allow group members to...	Enabled by default in these groups:
Administer Visuals ROLE_ADMINISTER_VISUALS	Create, read, update, and delete visuals from dashboards or from the Visual Gallery in the account. When the **Administer Visuals Privilege** is granted, the privileges **Create Visuals**, **Export Visuals**, and **Managed Visual Permissions** are automatically granted. In addition, users with the **Administer Visuals** privilege are automatically granted read, write, and delete permissions to all visuals in the account. However, if they do not also have Data Access permission for the data source associated with a visual, they cannot see any data on the visual.	■ Administrators group
Create Visuals ROLE_CREATE_VISUALS	Create visuals in Symphony. Users must also have Read permission for the data source selected for the visual. When the **Administer Visuals** privilege is granted, this privilege is also granted.	■ Administrators group
Export Visuals ROLE_EXPORT_VISUALS	Import and export visuals in Symphony. Users must also have Read permission for the data source selected for the visual. When the **Administer Visuals** privilege is granted, this privilege is also granted.	■ Administrators group
Manage Visual Permissions ROLE_PERMISSION_VISUALS	Assign permissions to a visual. When the **Administer Visuals** privilege is granted, this privilege is also granted. If this privilege is *not* granted, the  icon and the Permissions column in the Visual Gallery do not appear in the UI. See About Visual Permissions.	■ Administrators group
Administer Dashboards ROLE_ADMINISTER_DASHBOARDS	Add, modify, or remove dashboards in the account, including dashboards created by other users. When this privilege is granted, the **Create Dashboards**, **Export Dashboards**, and **Manage Dashboard Permissions** privileges are automatically granted.	■ Administrators group
Create Dashboards ROLE_CREATE_DASHBOARDS	Create dashboards and reports. If this privilege is not granted, your users: ■ **Add Dashboard** and **Add Report buttons** are not available in the dashboard library or reports library.	■ All groups

UI Privilege Name	Assign this privilege to allow group members to...	Enabled by default in these groups:
	■ Cannot import a dashboard or make of copy a dashboard using the Save As dialog. ■ Cannot make a copy of a report using the Save As dialog. When the **Administer Dashboards** privilege is granted, the **Create Dashboards**privilege is automatically granted.	
Export Dashboards ROLE_EXPORT_DASHBOARDS	Export dashboard configuration JSON files and self service reports as PDFs. If this privilege is not granted, users in the group can still export dashboards as screen shots (PNG) or PDF files, but they cannot export the dashboard configuration. When the **Administer Dashboards** privilege is granted, this privilege is automatically granted.	■ Administrators group ■ Content Distributors group
Manage Dashboard Permissions ROLE_PERMISSION_DASHBOARDS	Assign permissions to a dashboard. If your user has the **Administer Dashboards** privilege, they automatically have this privilege as well. If this privilege is not granted, the permissions () icon and the Permissions column on the Library page do not appear in the UI. See About Dashboard Permissions.	■ Administrators group
Administer Scheduled Reports ROLE_ADMINISTER_DASHBOARD_REPORTS	Create, edit, and delete all scheduled dashboard reports. Grant Read access for dashboards to users who receive reports.	■ Administrators group
Create Scheduled Reports ROLE_CREATE_DASHBOARD_REPORTS	Create, edit, and delete only your own scheduled dashboard reports.	■ Administrators group
Administer Tags ROLE_ADMINISTER_TAGS	Create, assign, remove, and delete all content tags.	■ Administrators group
Create Tags	Create, assign, and remove tags. Delete your own content tags.	■ Administrators group

UI Privilege Name	Assign this privilege to allow group members to...	Enabled by default in these groups:
ROLE_CREATE_TAGS		
Administer Sources ROLE_ADMINISTER_SOURCES	Create, import, export, modify, review, and remove data source configurations in the account. When this privilege is granted, the **Create New Data Sources**, **Manage Source Permissions** and **Edit Calculations** privilege are also granted. In addition, users with the **Edit Calculations** privilege are automatically granted read, write, and delete permissions to all sources in the account.	■ Administrators group
Create New Data Sources ROLE_CREATE_SOURCES	Create new data source configurations. When the **Administer Sources** privilege is granted, this privilege is also granted.	■ Administrators group
Manage Source Permissions ROLE_PERMISSION_SOURCES	Assign permissions to a data source configuration and manage data source row and column security filters. If your user has the **Administer Sources** privilege, they automatically have this privilege as well. If this privilege is *not* granted, the  icon and the Permissions column on the Sources page do not appear in the UI. See About Source Permissions.	■ Administrators group
Edit Calculations ROLE_EDIT_FORMULAS	Add or edit custom metrics and derived fields. Users with Read permission for a source and **Edit Calculations** can create and edit custom metrics and derived fields for the source. Users with Write permission for a source can create and edit custom metrics and derived fields for the source.	■ All groups
Administer Alerts ROLE_ADMINISTER_ALERTS	Add, modify, or remove alert definitions in the account, including alerts created by other users. When this privilege is granted, the **Create Alerts** privilege is automatically granted.	■ Administrators group
Create Alerts ROLE_CREATE_ALERTS	Create alert definitions in Visual Data Discovery. When the **Administer Alerts** privilege is granted, this privilege is also granted.	■ Administrators group
Manage Connections	Add, modify, or remove the data store connection definitions used by connectors and the query engine to connect to your data stores.	■ Administrators group

UI Privilege Name	Assign this privilege to allow group members to...	Enabled by default in these groups:
ROLE_MANAGE_CONNECTIONS		
Manage File Uploads ROLE_MANAGE_UPLOADS	Remove unused files uploaded for a data source if they are not used by any other sources. When the **Manage Connections** privilege is granted, this privilege is also granted.	■ Administrators group
Manage Action Templates ROLE_MANAGE_ACTION_TEMPLATES	Add, modify, or remove the action templates required to integrate Symphony visual and dashboard data into your third-party applications. Action templates define your third-party application to Symphony.	■ Administrators group
Invoke Actions ROLE_INVOKE_ACTIONS	Invoke an action template from a visual.	■ Administrators group
Administer Themes ROLE_ADMINISTER_THEMES	Create, read, update, delete, list, activate, and otherwise manage themes for the UI.	■ Administrators group
Administer Users ROLE_ADMINISTER_USERS	Administer other Symphony user definitions. When this privilege is granted, group users can: ■ Add, disable, and remove user definitions ■ Reset user passwords ■ Define user custom attributes and regional settings This privilege does not allow group members to update groups or the groups to which a user is assigned. If your user ID is not assigned the **Administer Groups** privilege or is not an administrator, you cannot assign groups to a user. In addition, only administrators can assign users to the **Administrators** group.	■ Administrators group
Administer Groups ROLE_ADMINISTER_GROUPS	Administer other Symphony group definitions. When this privilege is granted, group users can: ■ Add or remove group definitions	■ Administrators group

UI Privilege Name	Assign this privilege to allow group members to...	Enabled by default in these groups:
	■ Assign and remove users in a group definition ■ Authorize users in the group to perform specific Symphony functions This privilege does not allow group members to add or otherwise maintain user definitions.	
Save Filters ROLE_SAVE_FILTERS	Save (and share) filters created in visuals and dashboards. When this privilege is not granted, the **Save Filter** privilege does not appear on Filter dialogs in the UI.	■ All groups
Manage Custom Charts ROLE_MANAGE_VISUALIZATION_TYPES	When you have this privilege, you can use the CLI to create custom charts, and manage custom charts using the Symphony UI.	■ Administrators group
Generate Embed Code ROLE_GENERATE_EMBED_CODE	Generate an embeddable dashboard or visual gallery snippet for a dashboard in the dashboard library or the visual gallery. See Embed Components Into Your Application.	■ Administrators group
Administer Initial Visuals ROLE_ADMINISTER_INITIAL_VISUALS	Users with this privilege have permission to update the list of available visualizations for a source. See Available Visual Types.	■ Administrators group
Administer Calendars ROLE_ADMINISTER_CALENDARS	Users with this privilege can create, update, and delete alternative fiscal calendars. See Fiscal Calendars.	■ Administrators group
ROLE_DISTRIBUTE_CONTENT	Users who belong to the Administrators group or Content Distributors group can make content available to your users and tenant users.	■ Administrators group ■ Content Distributors group

User Auditing

This applies to: *Visual Data Discovery*

User auditing allows you as a Symphony administrator or client application administrator to track your users' access and actions to data considered sensitive. This includes tracking:

- What period of time any user interacted with data from dashboards, visuals, sources, and custom SQL queries.
- How the data was accessed: directly in the Symphony user interface, via an API, or as embedded objects in your applications.
- Actions performed with the data. See User Audit Events.
- For all sensitive data your applications need to track data access for, such as regulated industry information, PII, HIPAA, GDPR, and more.

Write your audit logs to the PostgreSQL database installed with Symphony, or an alternative PostgreSQL database of your choice. This enables your administrators to access and manage the data collected to suit your organization's needs.

The following topics cover enabling, configuring, and consuming user auditing and user audit data:

- Enable and Configure User Auditing
- Consuming Audit Data
- User Audit Events
- User Auditing for Multi Tenancy Environments
- Enable User Audit Data for Symphony Accounts
- Enable User Audit Data for Multi Tenant Accounts

Enable and Configure User Auditing

This applies to: *Visual Data Discovery*

Write your audit logs to the PostgreSQL database installed with Symphony, or an alternative PostgreSQL database of your choice. Create the database, update the `zoomdata.properties` file, and restart Symphony to begin capturing audit events.

Note: If you expect a large number of user audit events, accumulating a significant amount of information over time, consider using a separate PostgreSQL database.

Create a Database

Create the database for user auditing. There are two ways to do so:

Note: PostgreSQL databases are the only supported database type.

1. Bootstrap creation. When you install or upgrade Symphony using a `bootstrap-zoomdata` script, a user audit data table is created on the local PostrgreSQL instance. The user audit database is named `zoomdata-user-auditing`, and installed at the same location as the Logi metadata for your Symphony instance.

2. Manual creation. To use a separate database or PostgreSQL instance from your Symphony metadata database, edit the `zoomdata.properties` file to reflect the target parameters to use. Set the default value of `destination.params.password` to the same database password used in your default PostgreSQL database. Symphony creates the collections automatically in that database as needed.

Enable User Auditing

After you have created or linked your database, copy these properties into the `zoomdata.properties` file.

```
user-auditing.enabled=true
user-auditing.destination.name=PostgreSQL
user-auditing.destination.type=postgresql
user-auditing.destination.schema=public
user-auditing.destination.collection=audit_records
user-auditing.destination.collection-per-account=false
user-auditing.destination.params.user_name=${db.username:zoomdata}
user-auditing.destination.params.password=${db.password:}
user-auditing.destination.params.jdbc_url=jdbc:postgresql://localhost:5432/zoomdata-user-auditing
```

```
user-auditing.tenant.attribute=
```

Required and optional properties include:

Property	Default Value	Description
enabled	false	Use to enable or disable user auditing. Default is false, to disable user auditing. To enable, set to true. Required.
destination.name	PostgreSQL	The name of the type of database. Only PostgreSQL is supported.
destination.type	postgresql	The database type. Only postgresql is supported.
destination.schema	public	
destination.collection		The name of the collection of the user audit data. Use to separate data by accounts to prevent access to audit data by unauthorized users. For example, `audit_records<Account ID>`.
destination.collection-per-account	fals	Default is false, to disable To enable, set to true and include a field for Account ID in destination.collection.
destination.params.user_name	zoomdata	The user name sent to the database to write audit data. Required.
destination.params.password		The password sent to the database to write audit data. Required. Can be the same as your default PostgreSQL database.
destination.params.jdbc_url	jdbc:postgresql://localhost:5432/zoomdata-user-auditing	The path to the audit data database. This can be the same database as the user audit schema as shown, or a separate database or partition, as needed. Required.
tenant.attribute		Include any new or existing user attribute to support data capture by tenant customer users. Use in addition to account information that is always

Property	Default Value	Description
		captured and retained in the database.

Restart Symphony

Restart Symphony to begin capturing audit events.

If you used the bootstrap script to install Symphony and audit setup, the audit database table is created when the first event is logged. Test your setup by performing any of the event-triggering actions. See User Audit Events.

If you've set up your database manually, set up the Data Writer microservice to support the user audit processes.

Enable User Audit Data for Symphony Accounts

This applies to: *Visual Data Discovery*

Audit data for Symphony users, including administrators, are not visible to other Symphony accounts. You define this level of separation in the database in one of three ways:

- Tables: Configure Symphony to collect and write user audit data to different tables, then give users database access only to their own audit data table.
- Views: Disable user auditing by account, then give users access to their own audit data, using specific views and database access rights.
- Row level security: Disable user auditing by account, then configure row level security, limiting access for each user to their own data.

Important: Database tables that contain user audit data are created automatically when you first trigger an audit event. Trigger table creation with user auditing enabled and properly configured before you create a data source for the tables or apply access controls at the database level.

Tables for a Symphony Account

1. Use separate collections for accounts by defining `user-auditing.destination.collection-per-account=true` in the `zoomdata.properties` file.
2. Generate an audit event for all accounts to trigger audit data table creation.
3. Create user database accounts for each user. Define user access rights for each user to query only their own account's audit data table.
4. Create connections to the audit database using each account.

Views by Symphony Account

1. Disable use of separate collections for accounts by defining `user-auditing.destination.collection-per-account=false` in the `zoomdata.properties` file.
2. Generate an audit event for any account to trigger audit data table creation.
3. Create a view for each account that filters the audit data table, using a condition such as `accountID='<Account ID>'`.

4. Create user database accounts for each user. Define user access rights for each user to query only their own account's audit data view.
5. Create connections to the audit database using each account.

Row Level Security in the Database

1. Enable user auditing by defining `user-auditing.destination.collection-per-account=false` in the `zoomdata.properties` file.
2. Generate an audit event for any account to trigger audit data table creation.
3. Create user database accounts for each user, and configure row level security for the audit data table to limit users to their own data. For example, use a filter such as `accountID='<Account ID>`.
4. Create connections to the audit database using each account.

User Auditing for Multi Tenancy Environments

This applies to: *Visual Data Discovery*

Symphony supports two different levels of separation for user audit data. By default, user audit data is separated between accounts. If needed, you can further separate user audit data within a Symphony account by adding and using a custom attribute to separate the data.

Use Symphony to write audit data to a single table in your database or multiple tables to suit your organization's needs.

Important: Database tables that contain user audit data are created automatically when you first trigger an audit event. Trigger table creation with user auditing enabled and properly configured before you create a data source for the tables or apply access controls at the database level.

Separation Between Symphony Accounts

Audit data for Symphony users, including administrators, are not visible to other accounts. You define this level of separation in the database in one of three ways:

- Tables: Configure Symphonyto collect and write user audit data to different tables, then give users database access only to their own audit data table.
- Views: Disable user auditing by account, then give users access to their own audit data, using specific views and database access rights.
- Row level security: Disable user auditing by account, then configure row level security, limiting access for each user to their own data.

See Enable User Audit Data for Symphony Accounts.

Separation Between Tenants Within the Same Symphony Account

Configure user auditing for tenants to restrict access to user audit data to their own user tenant data. Symphony system administrators can see aggregated data for all tenants in the Symphony account.

You can define this level of separation in the database in one of several ways, depending on tenant connection creation privileges.

- If tenant users don't have permission to create new connections, add a user attribute value, then make the data available by applying the appropriate row level filter to that value.
- If tenant users do have permission to create new connections, you must set up database-level access control mechanisms.

- Views per tenant: Define a tenant attribute, then define what user audit data users can access, using specific views and database access rights.
- Database level row level security: Define a tenant attribute, then configure row level security for the audit table data, allowing users access to appropriate user audit data through individual user database accounts.

See Enable User Audit Data for Multi Tenant Accounts.

Enable User Audit Data for Multi Tenant Accounts

This applies to: *Visual Data Discovery*

Configure user auditing for tenants to restrict access to user audit data to their own user tenant data. Symphony system administrators can see aggregated data for all tenants in the Symphony account.

You can define this level of separation in the database in one of several ways, depending on tenant connection creation privileges.

Important: Database tables that contain user audit data are created automatically when you first trigger an audit event. Trigger table creation with user auditing enabled and properly configured before you create a data source for the tables or apply access controls at the database level.

Tenants Without Connection Creation Permissions

In environments where tenants don't have permission to create connections, you can set up and use user attribute value, such as `${User.tenant}` define row level filters to limit access to user audit data.

1. Set up an attribute for each user, such as `${User.tenant}`, to identify the tenant for each user.
2. Add the attribute name to the `zoomdata.properties` file. In this example, add `tenant: user-auditing.tenant.attribute=tenant`.
3. Generate an audit event for any account to trigger audit data table creation.
4. As a system administrator, create connection and data sources for the audit data in Symphony.
5. Add a row level filter to the data source. Compare the column tenant with the user attribute you created. In this example, `tenant=${User.tenant}`

Tenants With Connection Creation Permissions

If your tenants do have permission to create connections, there are two ways to restrict user audit data access: by setting up views at the tenant level, or applying row level security. Use the features of your database to control access to the data, once attributes are defined and applied.

Views Per Tenant

1. Set up an attribute for each user, such as `${User.tenant}`, to identify the tenant for each user.
2. Add the attribute name to the `zoomdata.properties` file. In this example, add `tenant: user-auditing.tenant.attribute=tenant`.

3. Generate an audit event for any account to trigger audit data table creation.
4. Create a view for each tenant that filters the audit data table, using a condition such as `tenant='<Tenant>'`.
5. Create user database accounts for each tenant. Define access rights for account user to query only their own tenant audit data table.
6. Create connections to the audit database using each user database account.

Row Level Security

1. Set up an attribute for each user, such as `${User.tenant}`, to identify the tenant for each user.
2. Add the attribute name to the `zoomdata.properties` file. In this example, add `tenant`: `user-auditing.tenant.attribute=tenant`.
3. Generate an audit event for any account to trigger audit data table creation.
4. Create a separate database user for each tenant, and configure row level security for each account user to query only their own tenant audit data table, using a condition such as `tenant='<Tenant>'`
5. Create connections to the audit database using each user database account.

Consuming Audit Data

This applies to: *Visual Data Discovery*

Users with appropriate access can review your audit data in one of three ways:

- Create visuals and a dedicated dashboard using Symphony.
- Direct database queries using any third-party tool.
- Add the user auditing database as a source and connection, then export the information using the Symphony API.

Consume Audit Data Using Symphony

Use Symphony to create a User Auditing dashboard.

1. Create a connection to your user auditing database.
2. Create a source from this connection.
3. Create a dashboard for User Auditing with visuals as needed.

Note: If you use a table visualization for user auditing data, add cross-visual filters such as List Filters and Tree Maps for a contextual look at your data.

Consume Audit Data Using Third-Party Tools

You can use any third-party tools you prefer to access the information for user auditing. Access to the tool or to the database using that tool should be restricted to appropriate users.

User Audit Events

This applies to: *Visual Data Discovery*

Specific user auditing events captured include:

Event	Description
COLLECTION_PREVIEW	User sees a collection data preview while creating or editing a source.
FACECT_VALUES_ACCESS	User sees values of a text search facet.
FIELD_STATS_ACCESS	User sees a field minimum and maximum values while trying to apply a filter or while reviewing an applied filter. User opens a visual or dashboard that includes a Time Bar.
FIELD_VALUES_ACCESS	User sees a list of field values while trying to apply a filter or while reviewing an applied filter.
KEY_SET_PREVIEW	User creates a keyset from a visual result set or data point, showing distinct values for a field. One keyset event per field.
KEY_SET_REVIEW	User reviews a keyset before applying it to a field, or after application, or prior to deletion.
RAW_DATA_EXPORT	User exports raw data for a visual or Details view, in .csv format.
VISUAL_DATA_EXPORT	User exports displayed data for a visual, in .csv format.
VISUAL_DATA_ACCESS	User creates, opens, changes a dashboard or loads data as new or a refresh.
TEXT_SEARCH	User performs a text search.

Manage Activity Logs

This applies to: *Visual Data Discovery*

The Symphony server records user- and server-based activities in log files. These files can be used by Symphony administrators to troubleshoot issues that may occur with the Symphony server, for example, activities related to setting up data sources or creating visuals and dashboards. When Symphony is installed in your network environment, a log file is created.

By default, activity logging is enabled, but some activities are not logged by default. For details, see the Activities Log Reference Sheet.

Important: Activity logging in `zoomdata-activity.log` is deprecated and will be removed in a future release. Information previously captured in `zoomdata-activity.log` can be found in other log files such as `access.log`, `service.log`, and by using Symphony's User Auditing feature.

You can enable or disable activity logging by calling REST API endpoints. See Activity Logging. A list of the types of activities that are logged can be found in the Activities Log Reference Sheet.

For more information about other Symphony log files, see ComposerSymphony Log Files Reference.

Activity Logging

This applies to: *Visual Data Discovery*

The Symphony server records user and server-based activities in log files. These files can be used by Symphony administrators to troubleshoot issues that may occur with the Symphony server (for example, activities related to setting up data sources or creating visuals and dashboards).

Symphony logs activities in `zoomdata-activity.log`. By default, logging to this file is enabled. Some activity types are not automatically logged. To determine which activities are logged by default, see the Activities Log Reference Sheet.

Important: Activity logging in `zoomdata-activity.log` is deprecated and will be removed in a future release. Information previously captured in `zoomdata-activity.log` can be found in other log files such as `access.log`, `service.log`, and by using Symphony's User Auditing feature.

This page covers the following topics:

If you want to use unified logging, see Set Up Unified Logging Using Fluentd. Fluentd unified logging is not enabled by default.

Configure the Symphony Activity Log

The Symphony activity log defaults can be configured through properties set in the `zoomdata.properties` file. See Configure ComposerSymphony for more information about modifying property files.

Property	Description	Default
log.file.count	Number of log files to keep	1
activity.log.file.size	Maximum log file size in Mb	10
activity.logs.dir	File path to store log files. Verify that this log directory has all the necessary permissions and that the owner of the directory is set to `zoomdata`. The `/home` directory cannot be used for logging.	Linux: /opt/zoomdata/logs/ Windows: <install-path>/logs/

Enable or Disable the Logging of Specific Activities

Enabling or disabling the logging of specific activities is performed via a series of REST API calls or by directly updating the appropriate activity property in the `zoomdata.properties` file. For a list of the activities that can be logged and the file in which they can be logged, see the Activities Log Reference Sheet.

To enable or disable logging for a specific activity using the REST API, run the following cURL command:

```
curl -u supervisor: <password> -XPUT 'http://<host>:<port>/composer/api/system/activity/type/<activityType> ' -H "Content-Type: application/vnd.composer.v3+json" --data '<option>'
```

- `<host>:<port>` - Specify the host IP address and port of the Symphony instance.
- `<activityType>` - Specify the name of selected activity (for example, `AUTHENTICATION` or `USER`).
- `<option>` - Specify either `true` or `false`. Set the value to `true` to enable the selected activity output or `false` to disable it.

To enable or disable logging for a specific activity by changing the appropriate activity property in the properties file:

1. Locate the `zoomdata.properties` file. For information about accessing Symphony configuration (properties) files, see Configure ComposerSymphony.
2. Add or edit the appropriate activity property in the file using the following format:

   ```
   activity.<activity-name>=<option>
   ```

   Activity names (`<activity-name>`) are listed in the Activities Log Reference Sheet.

   For `<option>`, specify either `true` or `false`. Set the value to `true` to enable the selected activity output or `false` to disable it.
3. Save the properties file.
4. Restart Symphony microservices. See Restart ComposerSymphony Microservices.

Determine Whether an Activity Is Being Logged

To determine whether a specific activity type is being logged in the log file, run the following cURL commsand:

```
curl -u supervisor: <password> -XGET 'http://<host>:<port>/composer/api/system/activity/type/<activityType>'
```

- `<host>:<port>` - Specify the address of the instance on which Symphony is installed.
- `<activityType>` - Specify the type of selected activity (for example, `AUTHENTICATION` or `USER`). For a list of the activities that can be logged, see the Activities Log Reference Sheet.

Example of Symphony Activity Logging Monitored by Fluentd

This example shows how to set up Fluentd to monitor Symphony activity log files. It does not show how to connect your Fluentd service to an external data store or how to pipe the log information to the external data store.

Before you use this example, make sure Fluentd is installed. Next, create a log file and grant the `td-agent` service permissions to write to it. Finally, modify the `td-agent.conf` file.

Linux

Create the file `/opt/zoomdata/logs/zoomdata-activity.log.pos` and run the following command to grant the `td-agent` service permissions to write to it:

```
chmod 777 /opt/zoomdata/logs/zoomdata-activity.log.pos
```

Finally, modify the `td-agent.conf` file in the `/etc/td-agent` folder using the template below.

Windows

Create the file `<install-path>/logs/zoomdata-activity.log.pos`. Grant the `td-agent` service permissions to read, write and execute permission to the owner, group and public.

Finally, modify the `td-agent.conf` file in the `/etc/td-agent` folder using the template below. Replace `pattern /zoomdata/service/health` with `<install-path>/service/health`.

Template

- Substitute Symphony fully qualified log file name for ***`<logfile>`*** and ***`<logfilen>`*** (for example, `opt/zoomdata/logs/zoomdata-activity.log`) for Linux, or `<install-path>/logs/zoomdata-activity.log` for Windows. Symphony log files names are provided in ComposerSymphony Log Files Reference.
- For each log file you select, specify rules in `<rule>` sections for the information you want to extract from the Symphony log files.

```
<source>
  @type forward
  @id input1
  port 24224
</source>
<source>
  @type tail
  path <logfile>[,<logfilen>]...
  pos_file zoomdata-activity.log.pos
  tag all.activity
  format json
  refresh_interval 1
</source>
<match all.activity>
  @type rewrite_tag_filter
  <rule>
    key     activityType
    pattern ^ACCOUNT$
    tag     account_logs
  </rule>
  <rule>
    key     activityType
    pattern ^AUTHENTICATION$
    tag     authentication_logs
  </rule>
  <rule>
    key     activityType
    pattern ^SOURCE$
    tag     source_logs
  </rule>
  <rule>
    key     activityType
    pattern ^RAW_DATA_EXPORT$
    tag     raw_data_export_logs
  </rule>
  <rule>
    key     activityType
    pattern ^RAW_DATA_EXPORT_CSV$
    tag     raw_data_export_csv_logs
  </rule>
  <rule>
    key     activityType
    pattern ^UPLOAD$
    tag     upload_logs
```

```
  </rule>
  <rule>
    key     activityType
    pattern ^USER$
    tag     user_logs
  </rule>
  <rule>
    key     activityType
    pattern ^VIS$
    tag     vis_logs
  </rule>
  <rule>
    key     activityType
    pattern ^GROUP$
    tag     group_logs
  </rule>
</match>

### Uncomment the section below to shift Zoomdata eventDates (UTC by default) to another timezone
### Adjust the quoted value in `Time.zone_offset("EDT")` below to the desired timezone value
### Additional details: https://docs.ruby-lang.org/en/2.4.0/Time.html#method-c-zone_offset
#<filter *_logs>
#  @type record_transformer
#  enable_ruby true
#  <record>
#     eventDate ${(Time.parse(record["eventDate"]).utc + Time.zone_offset("EDT")).strftime("%Y-%m-%d %H:%M:%S.%L")}
#  </record>
#</filter>

#type topology timeline data as json
<filter topology_performance_logs>
  @type record_transformer
  enable_ruby true
    <record>
      timeline ${record["timeline"].to_json}
    </record>
</filter>

#expand topology timeline json to new keys (attributes)
<filter topology_performance_logs>
  @type parser
  key_name timeline
  reserve_data true
  <parse>
```

```
    @type json
  </parse>
</filter>

#derive source information from the request payload
<filter vis_command_logs>
  @type record_transformer
  enable_ruby true
  <record>
    source_id ${if record["status"] == "STARTED" then JSON.parse(record["request"])["source"]["id"] end}
    source_name ${if record["status"] == "STARTED" then JSON.parse(record["request"])["source"]["name"] end}
    source_type ${if record["status"] == "STARTED" then JSON.parse(record["request"])["source"]["subStorageType"] end}
    source_schema ${if record["status"] == "STARTED" then JSON.parse(record["request"])["source"]["storageConfiguration"]
["schema"] end}
    source_collection ${if record["status"] == "STARTED" then JSON.parse(record["request"])["source"]
["storageConfiguration"]["collection"] end}
    connection_id ${if record["status"] == "STARTED" then JSON.parse(record["request"])["source"]["storageConfiguration"]
["connectionId"] end}
  </record>
</filter>

#exclude health check requests
#<filter request_logs>
#  @type grep
#  <exclude>
#    key uri
#    pattern /zoomdata/service/health
#  </exclude>
#</filter>

#send fluent log entries to stdout
<match fluent.**>
  @type stdout
</match>
```

Activities Log Reference Sheet

This applies to: *Visual Data Discovery*

This reference sheet lists the types of activities and corresponding fields that are logged in the Symphony activity log. By default, activity logging is enabled with a few exceptions (described later in this topic).

Activity Logging Defaults

The following table lists the activities that are logged by default and those that are not. Activities that are not logged by default must have logging manually enabled. See Activity Logging.

Important: Activity logging in `zoomdata-activity.log` is deprecated and will be removed in a future release. Information previously captured in `zoomdata-activity.log` can be found in other log files such as `access.log`, `service.log`, and by using Symphony's User Auditing feature.

Activity Log File	Activities Logged by Default	Activity Not Logged by Default
`zoomdata-activity.log`	account authentication group SOURCE USER VIS	raw_data_export raw_data_export_csv upload

If logging or specific activity logging is disabled in a previous version, the disabled status is retained after an upgrade.

Common Log Fields

For each activity, the following common fields are logged as part of each activity record:

Field Name	Description
user	contains information about the tenant user and the actions performed
accountID	contains the user's current tenant ID
userGroups	list of the groups to which the user belong
IP	contains IP address from which the event has been

Field Name	Description
	performed

Additional fields are logged as part of each activity.

Activity Log Fields

The following activity types can be logged by Symphony. For each activity, specific fields are logged, in addition to the common fields.

- account: Account Maintenance
- authentication: User Authentication
- Group: Group Maintenance
- Raw_data_export: Text Search
- Raw_data_export_csv: Data Export To CSV
- Source: Data Source Maintenance
- Upload: Flat File Upload
- User: User Account Maintenance
- Vis: Visual Maintenance

`account`: Account Maintenance

This activity is logged when a Symphony tenant account is created, updated, or deleted. Logging for this activity is enabled by default and captured in the `zoomdata-activity.log` file.

The following fields are recorded for this activity:

Field Name	Description
activityType	account
status	CREATED / UPDATED / DELETED

Field Name	Description
accountId	Generated tenant account ID
name	Name of the tenant account from which the event has been triggered
createdBy	User who created this tenant account.
createdDate	Tenant account creation date
lastModifiedBy	User who modified the tenant account
disabled	TRUE / FALSE

`authentication`: User Authentication

This activity is logged while user authentication at login. Logging for this activity is enabled by default and captured in the `zoomdata-activity.log` file.

The following fields are recorded for this activity:

Field	Value
activityType	authentication
status	SUCCEEDED / FAILED
authenticationType	BASE, SAML, LDAP, x.509

`group`: Group Maintenance

This activity is logged when a user group is created, updated, or updated. Logging for this activity is enabled by default and captured in the `zoomdata-activity.log` file.

The following fields are recorded for this activity:

Field	Value
groupId	Group ID
label	Name of the group
description	Group description added by the user
group	Privileges that have been assigned for the group

`raw_data_export`: Text Search

This activity is logged when a text search (applicable only for Elasticsearch, Apache Solr, and Cloudera Search sources) is performed using the `/api/stream/search` endpoint. Logging for this activity is disabled by default. After you have manually enabled logging for this activity, its log records are captured in the `zoomdata-activity.log` file.

The following fields are recorded for this activity:

Field	Value
activityType	raw_data_export
status	SUCCEEDED
exportType	REST (view details) FILE (export file)
count	Number of exported rows
storageType	Data source
query	Queries sent to the data source
cid	Created ID
actionStartedOn	Export start time
duration	Request processing time

`raw_data_export_csv`: Data Export to CSV

This activity is logged when a user exports the raw data to a CSV file. Logging for this activity is disabled by default. After you have manually enabled logging for this activity, its log records are captured in the `zoomdata-activity.log` file.

The following fields are recorded for this activity:

Field	Value
activityType	raw_data_export_csv
status	SUCCEEDED
count	Number of exported rows
location	DB or FILE
file	location=FILE - the full path to the file in local file system
cid	Created ID

Field	Value
actionStartedOn	Export start time
duration	Request processing time

`source`: Data Source Maintenance

This activity is logged when a source is created, updated, or deleted. Logging for this activity is enabled by default and captured in the `zoomdata-activity.log` file.

The following fields are recorded for this activity:

Field	Value
activityType	source
status	SAVED, UPDATED, DELETED
sourceId	Generated source ID
sourceName	Name of the source
streamType	Demo_record, CSV, API
storageType	Data source type
sourceAsString	Contains microservice info about the data source
sourceDescription	Source description added by the user

`upload`: Flat File Upload

This activity is logged when a user uploads the flat file while creating a new data source. Logging for this activity is disabled by default. After you have manually enabled logging for this activity, its log records are captured in the `zoomdata-activity.log` file.

The following fields are recorded for this activity:

Field	Value
activityType	upload
status	SUCCEEDED
source	Data source ID to which the file will be uploaded
fileName	Name of the uploaded file
contentType	Type of the uploaded file

Field	Value
filesize	Size of the uploaded file

`user`: User Account Maintenance

This activity is logged when user account is created, updated, assigned to or removed from a group, or deleted. Logging for this activity is enabled by default and captured in the `zoomdata-activity.log` file.

The following fields are recorded for this activity:

Field	Value
activityType	user
status	CREATED / UPDATED / DELETED
userID	
userName	
userFullname	
email	
subjectUserGroups	Groups which a user belongs to
userOrigin	NATIVE / SAML / LDAP
accounts	Tenant a ccounts, which a user is assigned to

`vis`: Visual Maintenance

This activity is logged when a visual is created or updated. Logging for this activity is enabled by default and captured in the `zoomdata-activity.log` file.

The following fields are recorded for this activity:

Field	Value
activityType	vis
status	CREATED / UPDATED
visualizationID	
visualizationName	

Manage the Upload API Source

This applies to: *Visual Data Discovery*

Symphony supports live streaming sources. This source uses PostgreSQL as a storage medium for the incoming data flow. By default, Symphony uses PostgreSQL to store the data for Upload API. This PostgreSQL instance is separate from other PostgreSQL instances you may have in existence. You can elect to use another PostgreSQL instance that you may have by changing the configuration properties.

Before You Begin

If you want to use an existing PostgreSQL instance in conjunction with the Upload API microservice, you need to find the above information in your `edc-postgresql.properties` file.

Use Another PostgreSQL Instance for Upload API

1. Use the following command to access and open the properties file:

```
vi /etc/zoomdata/zoomdata.properties
```

2. Add the following parameters to the file:

```
#  that are configure for Upload API
upload.destination.params.user_name=yourusername
upload.destination.params.password=yourpassword
upload.destination.params.jdbc_url=jdbc:postgresql://yourlocalhost:yourport//zd_upload
upload.batch-size=1000
```

3. Save and exit the configuration file.

4. Restart the Symphony server.

Authorize Symphony Access

This applies to: *Visual Data Discovery*

Your users can access Symphony, after you've defined their user accounts, added users to groups, and optionally, added users to tenants to grant access to content creation, content management, content access, and use.

Symphony supports several approaches to authenticating users, including SAML and LDAP. Choose the best approach given your existing constraints and objectives. A complete list of authentication tools supported by Symphony is provided in Supported Authentication Tools.

Note: SAML and LDAP groups that are automatically created in Symphony must be manually assigned group data source access and privileges.

Once authenticated, users have authorization to perform Symphony functions and access Symphony resources as defined by their group membership.

Use the following features to define and provide product access and authorization in Symphony.

- Symphony tenants: Use to separate Symphony product resources as necessary. Assign users to multiple tenants to allow access to each tenant's resources. You can further set up different groups, data connections, data sources, dashboards, and visuals for each tenant. See Manage Symphony Tenants.
- User accounts: Define access for individual users in Symphony. Assign users to one or more groups to give them access to Symphony data sources and product features. Users can belong to multiple groups in multiple tenants.
- Groups: Use to assign privileges to groups of users. Groups are most useful when a number of Symphony users require the same access restrictions. Users can be assigned to multiple groups. See Manage User Groups in Visual Data Discovery.

Manage Symphony Tenants

This applies to: *Visual Data Discovery*

As a member of the Supervisors group or a system admin user, you can manage Symphony tenants.

Use tenants to separate Symphony product resources as necessary. Assign users to multiple tenants to allow access to each tenant's resources. You can further set up different groups, data connections, data sources, dashboards, and visuals for each tenant.

See the following topics:

- About the Supplied Symphony Tenants
- Create a New Tenant
- List and Review Tenants
- Create and Add Additional Tenant Admin Users
- Modify Tenants
- Disable, Enable, and Remove Tenants
- Switch Tenants

About the Supplied Symphony Tenants

This applies to: *Visual Data Discovery*

One unique Symphony tenant is created during the installation process: *Visual Data Discovery* and includes all global users in Symphony.

When you, as a global user, work in the Managed Dashboard and Managed Reports modules, *Visual Data Discovery* is not shown as a tenant in the user interface. When you, as a global user, work in the Visual Data Discovery modules, your tenant is shown as *Visual Data Discovery* in the user interface.

- The *Visual Data Discovery* tenant is a dedicated and permanent tenant that cannot be deleted. The supplied admin user and other system admin users can maintain other tenants and perform system administrator-level functions.

- Other users can be assigned as admins of any tenant. See Create and Add Additional Tenant Admin Users.

Important: When you install v23.4 of Symphony in your environment, a default tenant , *Visual Data Discovery*, is created to manage users and content within your instance.

List and Review Tenants

List Tenants

1. Log in as a system administrator.
2. Select **Tenants** from the main menu. The Tenants work area opens, listing the defined tenants in your environment.
3. When you have reviewed the list of tenants in your environment, navigate away from this work area by selecting an option from the main menu.

Review Tenants

1. Log in as a system administrator.
2. Select **Tenants** from the main menu. The Tenants work area opens, listing the defined tenants in your environment.
3. To review a tenant, select its name in the list.
4. When you have reviewed the tenants in your environment, navigate away from this work area by selecting an option from the main menu.

Modify Tenants

Modify an Existing Tenant

1. Log in as a system administrator.
2. Select **Tenants** from the main menu. The Tenants work area opens, listing the defined tenants in your environment.
3. Scroll through the list of tenants or search for the tenant by name.
4. When you find your tenant, select the **Actions** menu, and select **Edit tenant**. An **Edit tenant** pop-up opens.
5. Make any changes you need in this work area, such as changing the **Tenant name**, or disabling the tenant, then select **Update tenant** to apply your changes.
6. Optionally, select **Edit in Managed dashboards** from the more menu to modify more tenant information. The Managed Dashboards **<tenant name> tenant details** work area opens, where you can modify seat allocation, provide data connector overrides, and more. **Save** your changes when your modifications are complete.

Supplied Users and User Groups

This applies to: *Visual Data Discovery*

Symphony supplies one user: **admin**. The admin is the default system administrator, who can define other users and tenants, enable security features, and define user groups and privileges. Add more system administrators as needed to perform all of the same functions as the default system admin.

After you've install and deployed Symphony in your operating environment, access the application as the admin user from a web browser.

Note: If you are installing or upgrading to v23.4 or later, the default admin user (system administrator) can perform all functions the former supervisor user previously performed, and provide authentication for embedded use cases.

admin User

The supplied **admin** user is defined as a system administrator for the supplied tenant. The **admin** user cannot be deleted.

The first time you log into the Symphony UI as the **admin** user, you may be prompted to change the **admin** user password. Thereafter, you can change the password for the **admin** user only if you are logged into the UI as an administrator.

Other users can be defined as administrators or can be assigned to groups with some or all administrator privileges. See Add New Users, and Symphony User Groups.

Administrators Group

Administrators group members include the default admin user, who is a system administrator associated with the *Visual Data Discovery* (Global User in Managed Dashboards) tenant.

Assign users to the Administrators group to give them administrative privileges to perform actions such as:

- Create and manage users, system admins, and groups.
- Create and manage tenants.
- Manage connectors, licenses, and actions.
- Enable security privileges, and more.

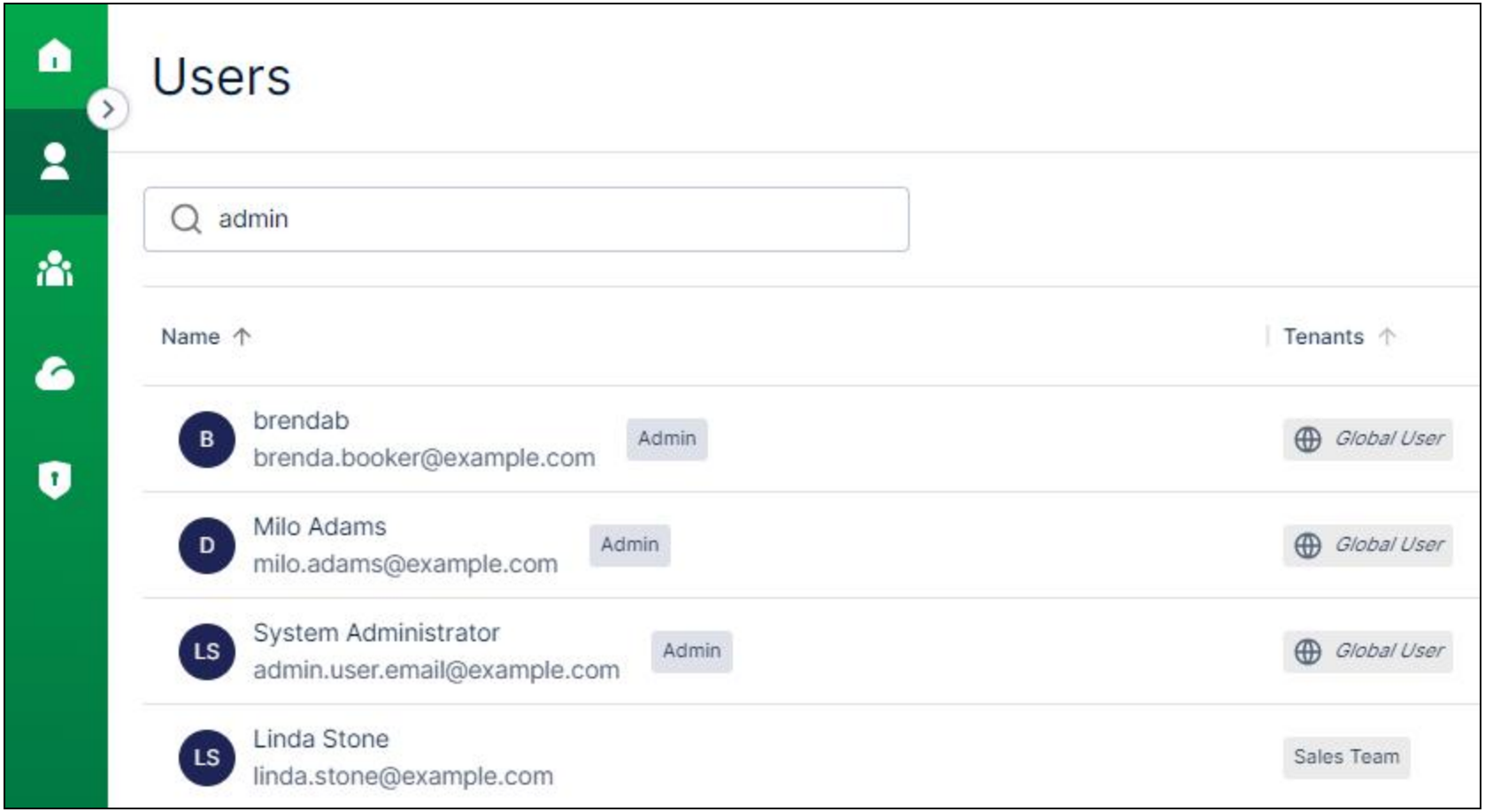
Note: The Administrators group is an integral part of Symphony management. Global Users can be system administrators. Tenants also have Administrators group: tenant admins are members of that group for one or more tenants.

See Add New Users, and Symphony User Groups, and The Composer UI MenuSymphony Main Menu.

Find Admins in Symphony

- System admins are listed with an **Admin** tag in the Name column on the main **Users** landing page.
- Only Global Users can be system admins, and show a **Global User** tag in the Tenants column.

- Search for all admins, including tenant admins, by entering **Admin** in the search box.



Administrators Group (Tenant Admins)

Administrators group members in tenants can:

- Create and manage users, groups, and content in their tenants.
- Define custom charts in their tenants.

- Perform Console and Actions related tasks.

Supervisors Group

The Supervisors group is designed to give you a group of users who can perform specific functions without giving them access to all tasks members of the Administrators group can perform in Visual Data Discovery.

All system admins are automatically added to this group. No further action is needed.

Content Distributors Group

All system admins are automatically added to this group. No further action is needed.

The Content Distributors group is part of the Visual Data Discovery tenant. Members can create, maintain, and distribute content and objects such as sources and connections to all tenants in your environment.

See About Supplied Groups.

Modify Users

This applies to: *Visual Data Discovery*

Symphony administrators, tenant administrators, or a user who has been assigned to a group with user management privileges can modify user accounts.

Modify a User

1. Log in as an administrator or a user who has been assigned to a group with user management privileges. If you are logged in as a tenant admin, verify you're in or switch to the appropriate tenant.
2. Select **Users** from the main menu. The Users work area opens. Scroll to find an existing user, or use the Search field to find a user.
3. Select **Edit user** from the **Actions** menu for the user you want to update. An Edit user pop up dialog opens.Select the actions menu button, then **Edit in Visual Data Discovery**.
4. The Users and Groups page appears. It consists of two sections: **Users** and **Groups**. Search for the user, or scroll to find the user you want to edit.
5. On the left side of the work area, select the name of the user you want to modify. The user information appears on the right side of the work area.
6. On the **Info** tab, update the values for **Full Name** and **Email** if needed. Select (check the checkbox) **Disable User** to disable this user.
7. On the **Custom Attributes** tab, add or change custom attributes for the user as needed. See Specify Custom User Attributes.
8. On the **Regional Settings** tab, change the regional language for the user as needed. See Specify A User's Regional Settings.
9. Select **Save** to save the changes you've made to this user.

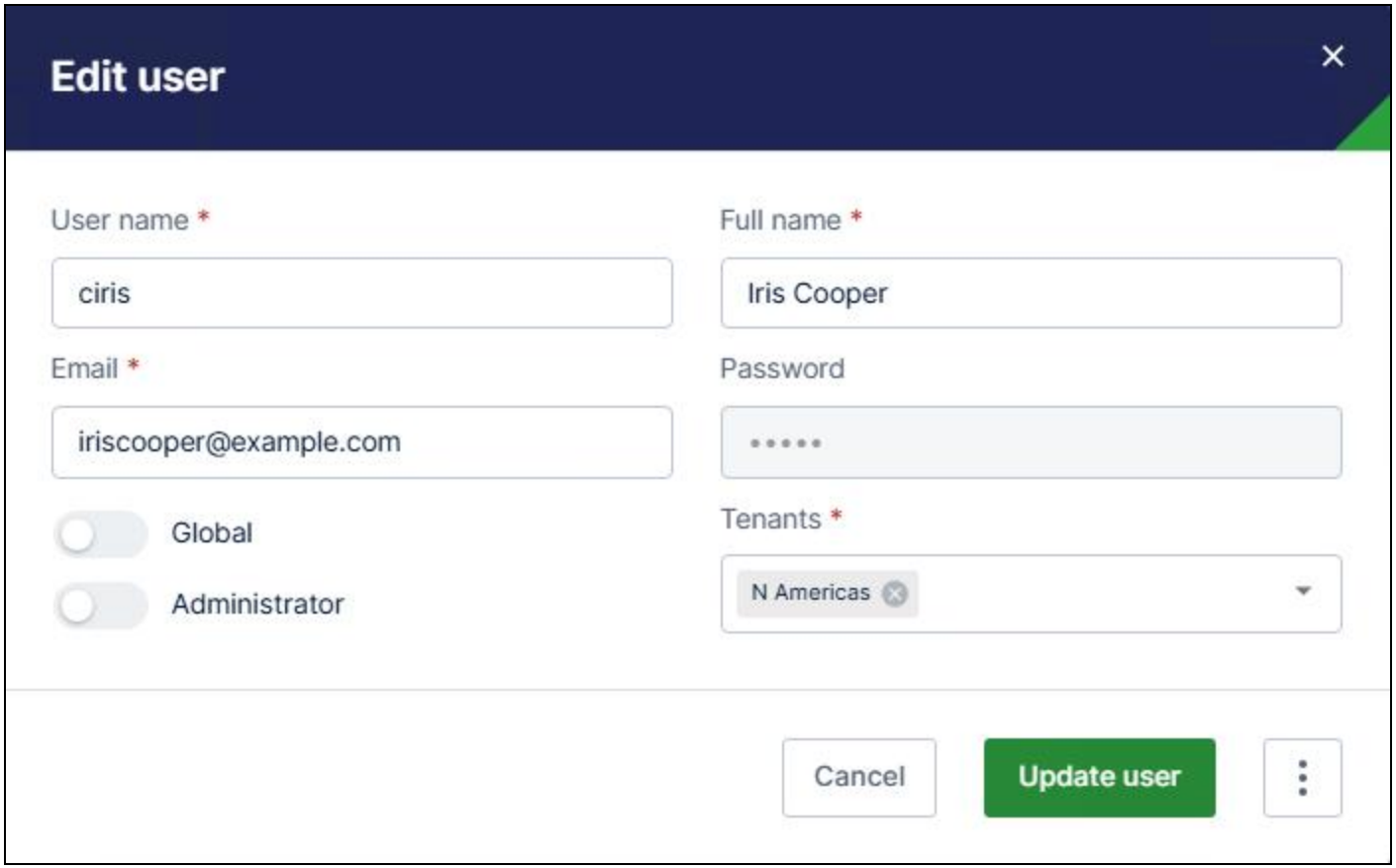
Assign and Remove Users in Tenants

This applies to: *Visual Data Discovery*

You can assign or remove users in Symphony tenants as a system admin user.

Assign a User to a Tenant

1. Log in as a system admin, and select **Users** from the main menu. The Users work area opens.
2. Scroll to find an existing user, or use the Search field to find a user.
3. Select **Edit user** from the **Actions** menu for the user you want to update. An Edit user pop up dialog opens.

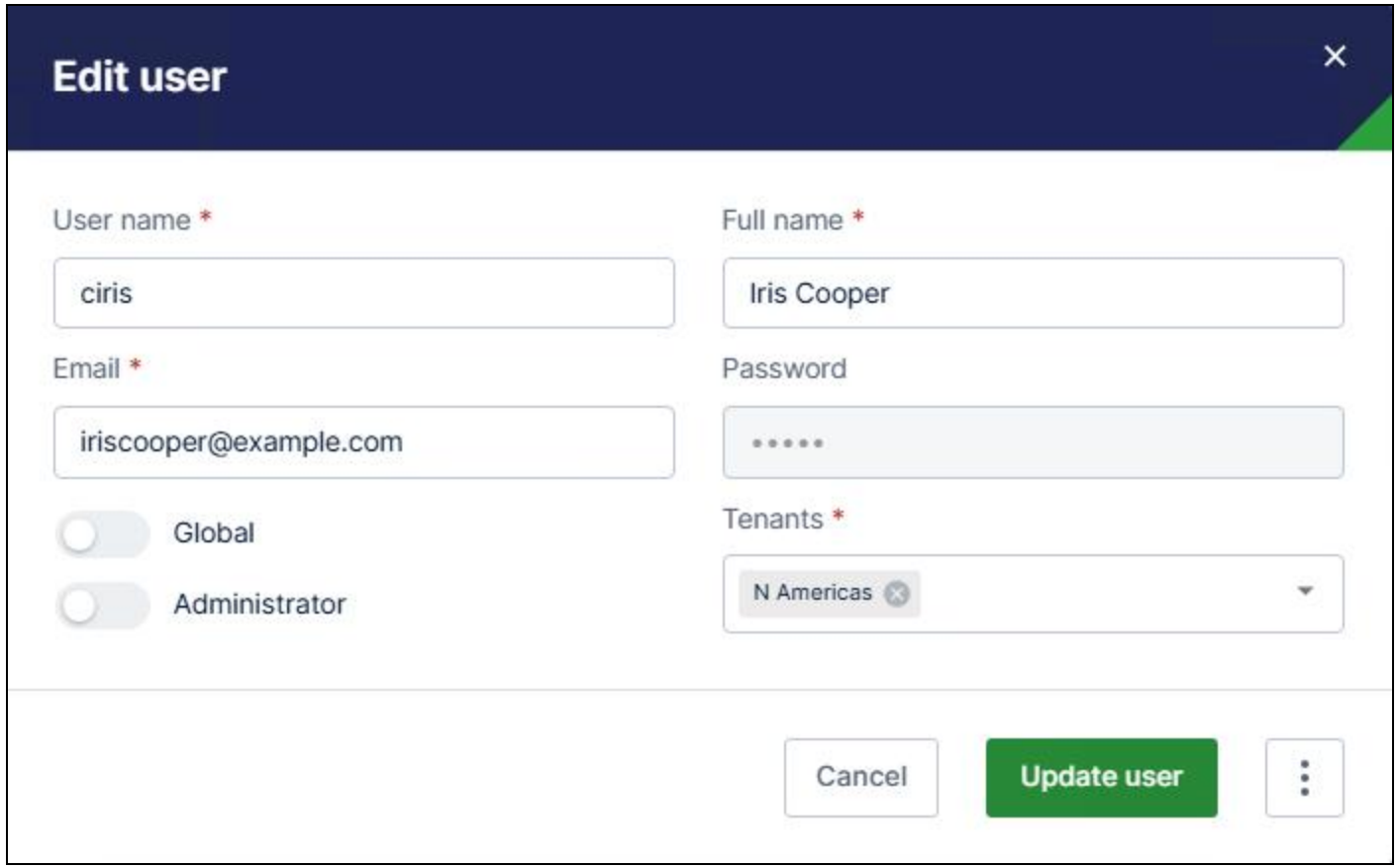


4. Select the **Tenants** field to open a list of available tenants. If the user belongs to a tenant, they are shown in this field and selected in the list.
5. Select (check) any tenants to assign the user to those tenants.

   If the user is a **Global** user, disable the toggle to see the Tenants selection field.
6. Select **Update user** when finished. Symphony returns a success message.

Remove a User from a Tenant

1. Log in as a system admin, and select **Users** from the main menu. The Users work area opens.
2. Scroll to find an existing user, or use the Search field to find a user.
3. Select **Edit user** from the **Actions** menu for the user you want to update. An Edit user pop up dialog opens.



4. Select the **Tenants** field to open a list of available tenants. Clear (uncheck) the checkbox associated with a tenant here to remove the user from that tenant.

   If needed, select **Global** to make the user a Global user and remove them from all tenants.

5. Select **Update user** when finished. Symphony warns you about removing a user and user data from a tenant. Select **Cancel** to cancel your changes, or **Remove** to continue.
6. Symphony returns a success message if you select Remove, or closes the work area if you select Cancel.

Set the Current Tenant for a User

This applies to: *Visual Data Discovery*

If a user has access to multiple Symphony tenants, you can specify the tenant they should use the next time they log in. After logging in, they can switch tenants, as needed.

After a user switches tenants, the most recent Symphony tenant they worked in is remembered and used the next time they log in. If a user is assigned to only one tenant, that tenant is always the current tenant.

Set the Current Tenant for a User

1. Log in as a system administrator. Select **Users** from the main menu, select a user, and select the option to **Edit a user in Visual Data Discovery**.
2. On the left side of the Manage Users work area, select the name of the user whose tenants you want to modify. The user information appears on the right side of the work area.
3. Select the **Tenant(s)** tab. This tab lists all the tenants to which the user is assigned and the number of groups to which the user is assigned in each tenant.
4. Select a tenant for the user to use the next time they log in from available tenants in the **Current Tenant** field.
5. Select **Save** to save the user.

Specify Custom User Attributes

This applies to: *Visual Data Discovery*

Custom user attributes are defined and managed in the Symphony interface. For more information, see Add a Custom Attribute.

- If you are hosting users who connect solely to embedded Visual Data Discovery content as trusted access users in a specific tenant, use the Custom User Attributes enumerated here to define what your users can access and what they see. This requires use of the trusted access method of Pull for user information and connection.

- If you use the trusted access method of Push for user information and connection, custom attributes must be set using the API. API documentation is provided with your Symphony installation at this link: `<symphony-URL>/discovery/swagger-ui.html`.

You can define custom attributes for a user's definition if you are logged in as a Symphony system administrator, tenant administrator, or a user who is assigned to a group with user management privileges.

Use custom attributes to store values that can be used as parameters, or variables, in connection definitions, data source row security filters, and in custom SQL.

A primary use of custom attributes is for user credential pass-through. Add credentials to a user's custom attributes so users with access to a particular data source connected to your instance can use these saved credentials to maintain access privileges for that source within Composer Symphony.

> **Note:** User attributes set for users via the UI are encrypted when stored in metadata. To specify the encryption modes, see Change the Encryption Mode.

Custom attributes are defined using the **Custom Attributes** tab when you edit the user.

When you reference custom attributes in connecton definitions, action URLs, attribute definitions, or a data source's custom SQL, they take the form of `${User.<custom-attribute-name>}`.

This section covers the following topics:

- Supplied Context Variables
- Add A Custom Attribute for a User
- Remove a Custom Attribute from a User
- Specify Multivalue (Array) Custom Attributes

- Specify Numeric Values in Custom Attributes
- Specify Time Values in Custom Attributes

Supplied Context Variables

The following context variables are provided with Symphony.

- `${User.composerUserName}`: include to insert the name of the user who is currently logged in.
- `${User.accountId}`: include to insert the user account ID of the user who is currently logged in.
- `${User.credentials}`: include to pass the session ID or trusted access token (in embedded environments) of the user.

You do not need to create custom user attributes for the user name or account ID. Use these supplied context variables instead.

Add A Custom Attribute for a User

Add a custom attribute for a user

1. Access the Custom Attributes tab for the user. See Modify Users.
2. Select Add Custom Attribute. A blank line is added to the Custom Attributes tab.
3. Supply values for the attribute, as described in the following table:

Tab Field	Description
Key	Specify the name of the custom attribute in the Symphony displays. The name cannot include braces.
Value	Specify one or more values for the custom attribute. See below for more options.
Usage	Shows how the attribute appears in the text entry fields of the application.
Secure	Select this checkbox if you want to encrypt the custom attribute values.
Delete	Select the delete button to remove the attribute.

4. When you have specified all values, select **Save** to save the user.

Remove a Custom Attribute from a User

Remove a custom attribute from a user

1. Access the Custom Attributes tab for the user. See Modify Users. The defined custom attributes are listed.
2. Select the delete button corresponding to the attribute you want to remove.
3. When are done removing your specific attributes, select **Save** to save the user.

Specify Multivalue (Array) Custom Attributes

For multivalue user attributes (arrays), the elements must be comma-separated and contain no redundant white spaces between elements. These multivalue custom attributes should only be used as arrays in INCLUDE and EXCLUDE filter operations. With all other filter operations, the custom attribute values are used as-is.

For example, an INCLUDE or EXCLUDE row security filter that uses a multivalue custom attribute containing the array (`["apples", "oranges", "bananas"]`) will interpret the values as three separate values, splitting the values on commas.

However, an EQUAL or NOT EQUAL row security filter using the same multivalue custom attribute would interpret the array values as a single value (`"apples,oranges,bananas"`).

Null values in multivalue custom attributes are processed as a special text marker - `[Null]`.

Specify Numeric Values in Custom Attributes

Numeric value in custom attributes are supported as integers or floating-point values represented as text.

Specify Time Values in Custom Attributes

For date-time values, the following formats are supported:

- ISO-8601 format with optional milliseconds (but no timezone): `yyyy-MM-ddTHH:mm:ss[.SSS]`
- Symphony's default API time format: `yyyy-MM-dd HH:mm:ss.SSS`, milliseconds required.

Unix time stamp format (seconds/milliseconds) is not supported for date-time values.

The BETWEEN filter operator expects an array with two elements.

When you use a time-value custom attribute as a filter variable for a filter using BETWEEN, two explicit elements must be specified (for example `start_date` and `end_date`). The dates can be specified as static date-time values, dynamic time patterns, or single-value user attributes.

Specifications such as `[${User.date_array}]`, with the `date_array` attribute containing two elements are not supported.

Specify A User's Regional Settings

This applies to: *Visual Data Discovery*

Set regional settings for a user to determine the language of each user and the format of numeric fields in a data source. For example, a user based in the United States sees number fields formatted for Italian currency differently than a user based in Italy. See Number Formatting for Data Sources for more information.

These settings can be changed by an administrator or a user who has been assigned to a group with user management privileges.

The user locale can be set using the **Regional Settings** tab, as well.

To access the **Regional Settings** tab, see Modify Users. For information on the user settings that can be changed on other tabs for a user, see:

- Add New Users
- Assign and Remove Users in Tenants
- Specify Custom User Attributes

Set the User Locale for a User

1. Navigate to the **Regional Settings** tab for a user. See Modify Users.
2. Select the location for the user in the drop-down list of the **Regional Presets** box.
3. When all values have been specified, select **Save** to save the user.