Installation Primer

Overview

The Installation Primer describes how to install Tasktop Hub and covers some basic information you should know before proceeding with the installation. If you are working on a deployment with Tasktop, your Solutions Architect will assist you with the installation.

System Requirements

On the System Requirements page, you can learn about:

- Supported Operating Systems
- Supported Browsers
- Supported Databases
- Java Runtime Environment
- Hardware Sizing for Deployment Scenarios

Installation

On the Installation page, you can learn about:

- Sandbox Environment
- Where to download Tasktop
- Installation on Windows
- Installation on Linux
- SSL Certificate Installation
- Port Configuration
- Default File Locations
- Repository Preparations

Advanced Configuration

On the Advanced Configuration page, you can learn about:

- Container Configuration
- Increasing Available Memory
- Logging

Upgrading

On the Upgrading page, you can learn about:

- Performing Tasktop Integration Hub version upgrades
- Back up and Restore practices

Business Continuity

On the Business Continuity page, you can learn about:

- Best practices for data loss prevention
- Impacts of Tasktop downtime
- Failover strategy/high availability guidelines

System Requirements

> ⚠️ Beginning in April 2022, the following operating systems will be no longer supported by Tasktop:
>
> - Windows Server 2012 R2
> - Windows Server 2012
> - Red Hat Enterprise Linux 6.x
> - Ubuntu Linux 14.04 LTS
> - SUSE Linux Enterprise Server 11.x
>
> Beginning in Tasktop Hub version 21.4, **TLS 1.2 is required for all encrypted connections**. The database used for storing Tasktop operational data and any repositories being used must support TLS 1.2.
>
> If you have any questions, please contact Support.

General Requirements

Tasktop Hub is a web application which runs centrally on a server. Users interact with it through a web browser from any computer that has network access to the server.

For best results, Tasktop Hub should be deployed in an environment that has good network throughput and low latency to its operational databases and all repositories involved in an integration.

Below are general requirements to meet the needs of typical deployment scenarios.

- Tasktop Hub **must** be installed in a server environment and only **one** instance of Tasktop should be installed on each server.
- The Hub operational database should have its own machine and should be co-located with the Hub server to reduce latency.

User Requirements

To install and configure Tasktop Hub, you need an account with administrative privileges on your server. The account must also have read/write access to the default file locations.

Supported Operating Systems

⚠️ **Note**: For Windows, Powershell 4 must be installed on your server.

The following 64-bit operating systems and versions are supported:

- Windows 10
- Windows 11
- Windows Server 2019
- Windows Server 2022
- Oracle Enterprise Linux 7+
- Oracle Enterprise Linux 8+
- Red Hat Enterprise Linux 8.x
- Ubuntu Linux 18.04 LTS
- Ubuntu Linux 20.04 LTS
- Ubuntu Linux 22.04 LTS
- SUSE Linux Enterprise Server 12.x
- SUSE Linux Enterprise Server 15.x

Available under Extended support:

- Windows Server 2016 *(End-of-Service-Life Date:* 18 Jan 2023*)*
- Red Hat Enterprise Linux 7.x *(End-of-Service-Life Date: 18 Jan 2023)*
- Ubuntu Linux 16.04 LTS *(End-of-Service-Life Date:* 18 Jan 2023*)*

💡 Note that certain connectors (e.g., IBM DOORS) only run on Windows operating systems. Before installing Tasktop Hub, we recommend consulting with your Tasktop contact to determine which operating system best fits your integration scenario.

Supported Browsers

💡 **Note**: Tasktop Hub runs with a minimum screen resolution of **1280 pixels x 800 pixels**.

The Tasktop Hub web interface is supported on the following browsers:

- Firefox 101+
- Google Chrome 102+
- Microsoft Edge 102+

Available under Extended Support:

- Firefox 94-100 *(End-of-Service-Life Date: 18 Oct 2023)*
- Google Chrome 95-101 *(End-of-Service-Life Date: 18 Oct 2023)*
- Microsoft Edge 95-101 *(End-of-Service-Life Date: 18 Oct 2023)*
- Firefox 83-93 *(End-of-Service-Life Date:* 18 Jan 2023*)*
- Google Chrome 87-94 *(End-of-Service-Life Date:* 18 Jan 2023*)*
- Microsoft Edge Chromium 87-94 *(End-of-Service-Life Date:* 18 Jan 2023*)*

💡 If you are interested in extended support, please reach out to your Tasktop contact.

Supported Databases for storing Tasktop Operational Data

This feature is not applicable to Tasktop Cloud.

⚠️ Tasktop automatically stores operational data to a pre-configured Derby database. This is suitable for evaluation purposes *only*, and is not supported for production environments. Configuring Tasktop to utilize an external database enables you to perform frequent back-ups without stopping Tasktop Hub, and ensures that your Tasktop Hub practices are consistent with your existing disaster and recovery process.

⚠️ **Note**: The database used for Tasktop Hub operational data must support **TLS 1.2**.

To reduce latency, the Hub operational database should have its own machine and should be co-located with the Hub server.

Minimal User Permissions

For all supported databases, the user **must** have sufficient permissions to connect, create, alter and drop tables and indexes and create temporary tables. Users must also have sufficient permissions to select, insert, update, delete, and truncate tables.

Tasktop supports this operational database policy for scenarios where your database is on **any** cloud infrastructure like AWS or Azure. You can refer to the resources below for more information on encrypting communication between Hub and Database:

- For AWS, we recommend implementing a VPC. Click here for more information.
- For Azure, we recommend a VPN gateway. Click here for more information.

💡 **Note**: A separate database must be used for Tasktop Operational Data and Enterprise Data Stream integrations.

Operational Database Recommendations

The recommendations below offer a **general guideline** only. We recommend consulting with Tasktop Support to determine the exact needs for your integration scenario, and for guidance on how to efficiently configure Hub.

You can see guidelines regarding external database sizing here.

We strongly recommend using the latest supported version of **PostgreSQL** for storing Tasktop Operational Data. At scale, Tasktop Hub performs better, more reliably, and requires fewer resources with PostgreSQL than with the other available database options.

Supported Versions and Configuration Details

In the section below, you will find supported database versions for storing Tasktop Operational Data and configuration details for each database.

PostgreSQL

Supported Versions

- 10
- 11
- 12
- 13
- 14

Extended Support

- 9.6 (*End-of-Service-Life Date:* 18 Jan 2023*)*

💡 If you are interested in extended support, please reach out to your Tasktop contact.

Configuration Settings

- The database must be case-sensitive (this is the default configuration).
- The database must be configured with the **UTF8** character set.

```
CREATE DATABASE dbName
ENCODING 'UTF8'
LC_COLLATE = 'en_US.UTF-8'
LC_CTYPE = 'en_US.UTF-8'
TEMPLATE template0
```

Necessary User Permissions

The following provides sufficient permissions for the `tasktop_hub` user on the `tasktop_hub` database, when using a public schema:

```
REVOKE ALL PRIVILEGES ON DATABASE tasktop_hub
        FROM tasktop_hub;

GRANT CONNECT, TEMP ON DATABASE tasktop_hub
        TO tasktop_hub;

GRANT CREATE ON SCHEMA public
        TO tasktop_hub;

GRANT SELECT, INSERT, UPDATE, DELETE, TRUNCATE
        ON ALL TABLES IN SCHEMA public
        TO tasktop_hub;
```

The following provides sufficient permissions for the `tasktop_hub` user on the `tasktop_hub` database, when using a custom schema:

💡 If you use a custom schema, please note that when configuring the external database connection you will need to add "`?currentSchema=tasktop`" to the database connection URL, e.g. `jdbc:postgresql://example.com:5432/dbName?currentSchema=tasktop`

```
CREATE SCHEMA TASKTOP;

REVOKE ALL PRIVILEGES ON DATABASE tasktop_hub
    FROM tasktop_hub;

GRANT CONNECT, TEMP ON DATABASE tasktop_hub
    TO tasktop_hub;

GRANT CREATE ON SCHEMA Tasktop
    TO tasktop_hub;

GRANT SELECT, INSERT, UPDATE, DELETE, TRUNCATE
    ON ALL TABLES IN SCHEMA Tasktop
    TO tasktop_hub;
```

Microsoft SQL Server

Supported Versions

- 2017
- 2019

Extended Support

- N/A

Configuration Settings

- The database must be case sensitive. We recommend Latin1_General_100_CS_AS_KS_WS.
  - This can be configured using the following command (replace **dbName** with the name of your database):

    ```
    ALTER DATABASE dbName COLLATE Latin1_General_100_CS_AS_KS_WS;
    ```

- We recommend monitoring the size of your transaction log, as very large transaction logs can cause database connection errors.
- We recommend using JDBC driver `mssql-jdbc-7.0.0.jre8.jar` when transferring from operational database to SQL Server.

Necessary User Permissions

- The user must be a SQL authenticated user (not a Windows authenticated user)
- Additionally, the user must have the following roles granted:

  - `db_datareader`
  - `db_datawriter`
  - `db_ddladmin`

💡 **Note**: Instance and Database name options can be specified by attaching "`;instanceName=;databaseName=`" to the end of the JDBC URL in Tasktop Hub. If using JDBC driver `mssql-jdbc-10.2.x` or later, the "`;trustServerCertificate=`" parameter and its corresponding value is required by the driver.

Oracle

Supported Versions

- 19c
- 21c

Extended Support

- 18c *(End-of-Service-Life Date:* 18 Jan 2023*)*

💡 If you are interested in extended support, please reach out to your Tasktop contact.

Configuration Settings

- The database must be case-sensitive (this is the default configuration).
- The database must be configured with the **AL32UTF8** character set.

```
ALTER DATABASE dbName CHARACTER SET AL32UTF8;
```

Necessary User Permissions:

User must have `CREATE SEQUENCE`, `CREATE TABLE`, `CREATE SESSION` permissions, as well as sufficient quota. Typical user creation might look as follows:

```
CREATE USER tasktop_hub IDENTIFIED BY a_password DEFAULT TABLESPACE tasktop_hub;

GRANT CREATE SESSION TO tasktop_hub;

GRANT CREATE SEQUENCE, CREATE TABLE TO tasktop_hub;
ALTER USER tasktop_hub QUOTA UNLIMITED ON tasktop_hub;
```

Troubleshooting

- To resolve error **ORA-30036 (UNABLE TO EXTEND SEGMENT BY 8 IN UNDO TABLESPACE)**, please refer to the following documentation.

MySQL

Supported Versions

- 5.7.7+ (excluding 5.7.0 - 5.7.6) (Enterprise Edition only)
- 8.0 (Enterprise Edition and Community Edition)

Extended Support

- N/A

Configuration Settings

The following settings must be applied before connecting Tasktop to MySQL:

- The database must be case sensitive.
  - This can be configured using the following command (replace **dbName** with the name of your database):

    ```
    ALTER DATABASE dbName COLLATE = 'utf8_bin'
    ```

- The database default charset must be UTF-8, `ALTER DATABASE dbName CHARACTER SET = 'utf8'`
  - You can also create the database with these settings: `CREATE DATABASE dbName CHARACTER SET = 'utf8'`
- We recommend using JDBC driver version 8.0 or later when transferring from an operational database to MySQL Server.
- `innodb_default_row_format` must be `DYNAMIC`
- `innodb_file_format` must be `Barracuda`
- `innodb_file_per_table` must be `ON`
- `innodb_large_prefix` must be `ON`
- `innodb_buffer_pool_size` must be minimum 1G

  - This size is highly dependent on customer hardware and data size — the number above is only a recommendation. Please consult with Tasktop Support if you have any questions.
- `max_allowed_packet` property must be minimum 64M
  - If this is set too low, you will see a **Packet for query is too large** error on the Activity screen.
- `max_connections` property should be minimum 500
  - **Note**: The number of connections Tasktop uses is highly dependent on customer configuration, hardware, and load — the number above is only a recommendation. Please consult with Tasktop Support if you have any questions.

💡 **Note**: `innodb` settings are the default settings for MySQL, so you will not need to make any changes to those settings unless they have been changed previously. The `innodb` settings apply globally to all MySQL databases on the server, while the `character set` is specific to the database.

⚠️ Configuring Tasktop Hub with the MySQL external operational database will prohibit the synchronization of 4-byte characters due to MySQL's default UTF8 encoding being limited to 3 bytes. Examples of 4-byte characters include but are not limited to some emojis and some Chinese characters. If you may be synchronizing 4-byte characters, consider using another supported database.

Necessary User Permissions

The following provides sufficient permissions for the `tasktop_hub` user on the `tasktop_hub` database:

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM tasktop_hub;

GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, INDEX, LOCK TABLES, REFERENCES ON tasktop_hub.* TO
tasktop_hub
```

Supported Databases for use in Enterprise Data Stream Integrations

The Tasktop Database add-on allows you to create integrations that send artifact information to one central database.

💡 **Note**: A separate database must be used for Tasktop Operational Data and Enterprise Data Stream in tegrations.

You can see guidelines regarding external database sizing here.

Supported Versions

If your license includes the Tasktop Database add-on and you would like to configure an Enterprise Data Stream Integration, the following databases and versions are supported:

PostgreSQL

General Support

- 10
- 11
- 12
- 13
- 14

Extended Support

- 9.6 *(End-of-Service-Life Date:* 18 Jan 2023*)*

💡 If you are interested in extended support, please reach out to your Tasktop contact.

Microsoft SQL Server

General Support

- 2017
- 2019

Extended Support

- N/A

Oracle

General Support

- 19c
- 21c

Extended Support

- 18c *(End-of-Service-Life Date:* 18 Jan 2023*)*

MySQL

We recommend using JDBC driver version 8.0 or later when creating a SQL connection for Enterprise Data Stream integrations.

General Support

- 5.7
- 8.0

Extended Support

- N/A

💡 **Note**: The user must be a SQL authenticated user (and not a Windows authenticated user)

Database Connections and Encryption

The following section describes different ways to configure your database connection. If you choose not to encrypt your connection, data will be transmitted over the network unprotected and will be at risk of being intercepted. Likewise use of self-signed certificates or other certificates not signed by a trusted Certificate Authority puts your data at risk as Tasktop cannot verify the identity of the server at the end of the connection.

Please ensure your connection is configured in a way that is aligned with your security policy and the associated risks are understood and accepted.

Configuration Details

PostgreSQL

or PostgreSQL, please refer to PostgreSQL documentation for more information.

Location

- Example Format: *jdbc:postgresql://hostServerName:postgreSqlServerPort/MyDatabaseName*

You can enable encrypted connections by setting 'ssl=true' (e.g., **jdbc:postgresql://<server-name>: <port>/?ssl=true**).

If the certificate for the PostgreSQL server is self-signed you'll need to set 'sslfactory=org.postgresql.ssl. NonValidatingFactory' and 'sslmode=require' (e.g., **jdbc:postgresql://<server-name>:<port>/? ssl=true&sslmode=require&sslfactory=org.postgresql.ssl.NonValidatingFactory**).

If the certificate for the PostgreSQL server is not self-signed you'll need to add the certificate to the JDBC's truststore.

Microsoft SQL Server

For SQL Server, please refer to Microsoft documentation for more information.

Location

- Example Format: *jdbc:sqlserver://hostServerName;instanceName=MyInstance; databasename=MyDatabaseName*

You can enable encrypted connections by setting 'encrypt=true' (e.g., **jdbc:sqlserver://<server-name>: 1433;encrypt=true;trustServerCertificate=false**). If the certificate for the MySQL server is self-signed you'll need to set 'trustServerCertificate=true' (e.g., **jdbc:sqlserver://<server-name>:1433;encrypt=true; trustServerCertificate=true**)

If using JDBC driver `mssql-jdbc-10.2.x` or later, the "`;trustServerCertificate=`" parameter and its corresponding value is required by the driver.

💡 **Note**: Some older editions may be missing security updates and will need to apply security service packs to use a self-signed certificate and encryption. You may experience certificate errors if the SQL Server is using a self-signed or corporate certificate. To work around this, you will need to disable certificate validation in the JDBC driver or add the certificate to the JDBC's truststore.

Oracle

For Oracle, please refer to this whitepaper for an overview of how to set up connections to encrypted Oracle server. For a guide to configuring the Oracle server to support SSL, please refer to Oracle documentation.

Location

- Example Format: *jdbc:oracle:thin:@hostServerName:oracleServerPort/SID*

For the most part assuming that the server is set up properly, you can follow Case#2 in the white paper and simply use a URL with the following format: **jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS= (PROTOCOL=TCPS)(PORT=2484)(HOST=<hostname>))(CONNECT_DATA= (SERVICE_NAME=<servicename>)))**. On the server, make sure to disable client authentication by setting 'SSL_CLIENT_AUTHENTICATION=FALSE ' in the listener.ora and sqlnet.ora files.

For unencrypted connections, the protocol should be **TCP** and the port would generally be 1521, but the URL would otherwise be the same. The above example connection string is formatted in the 'Oracle Net connection descriptor' format, but Tasktop also accepts 'Thin-style service name' connection strings such as **jdbc:oracle:thin:@<hostname>:1521:<servicename>**.

If the certificate for the Oracle server is self-signed, but you still want to use SSL, you will need to follow Case#1 in the white paper. As described in the paper, only anonymous cipher suites are permitted when trying to use SSL without server authentication. You can specify the cipher suites in the sqlnet.ora file on the Oracle server.

💡 **Note**: Some versions of Oracle do not by default support anonymous cipher suites. Thus, they will need to be imported to the server before enabling them.

MySQL

For MySQL, refer to MySQL documentation for the details on how to set up your connection.

Location

- Example Format: *jdbc:mysql://hostServerName:mysqlServerPort/MyDatabaseName*

To enable encryption on older MySQL servers (5.6.25 and earlier or 5.7.5 and earlier) you need to set the connection property 'useSSL=true' (e.g., **jdbc:mysql://<server-name>:3306?useSSL=true**). Later versions will implicitly try to connect using an encrypted connection. Regardless of the version, the client will only enforce that the server uses TLS if the property 'requireSSL=true' is set.

If the certificate for the MySQL server is self-signed you will need to set 'verifyServerCertificate=false' (e.g., **jdbc:mysql://<server-name>:3306?useSSL=true&verifyServerCertificate=false**).

Java Runtime Environment

Tasktop Hub is packaged with a JRE; there is no need to install a JRE separately. Tasktop Hub uses and ships with Oracle Java.

💡 **Note**: Partner branded editions of Tasktop Hub use and ship with **Azul OpenJDK**.

Deploying Hub on a Cloud Environment

⚠️ To ensure reliable performance, all virtual machines (on-prem and private cloud) **must** meet the requirements listed in the General Requirements section.

Tasktop Hub can be deployed in operating systems on physical servers within virtual machines hosted on dedicated on-prem virtual machine hosts. Tasktop Hub can also be deployed within private cloud deployments, such as AWS or Azure. If deploying on a private cloud environment, Hub and its operational database must be deployed using a full image and not a container, with the exception of AWS RDS PostgreSQL. Tasktop Hub **cannot** run in containerized deployments (Kubernetes, OpenShift, etc.).

Tasktop has qualified AWS RDS PostgreSQL deployments for use as the operational database for Hub instances hosted in AWS private cloud environments. Tasktop does not offer direct support for private cloud hosting infrastructure (i.e., AWS networking and configuration) beyond the operation of Tasktop's own products within the hosted environment. See the section below for recommended configuration settings.

AWS RDS PostgreSQL Recommendations

The recommendations below offer a **general guideline** only. We recommend consulting with Tasktop Support to determine the exact needs for your integration scenario, and for guidance on how to efficiently configure Hub.

Note: Tasktop does not troubleshoot or maintain AWS RDS PostgreSQL. Please ensure your database is configured in a way that is aligned with your security policy and that the associated risks are understood and accepted.

Setting	Recommended Value
`"DBInstanceClass"`	`"db.t3.small"`
`"Engine"`	`"postgres"`
`"AllocatedStorage"`	50
`"BackupRetentionPeriod"`	30
`"MultiAZ"`	true
`"EngineVersion"`	`"13.6"`
`"AutoMinorVersionUpgrade"`	false
`"PubliclyAccessible"`	false
`"StorageType"`	`"gp2"`
`"StorageEncrypted"`	true

"CopyTagsToSnapshot"	true

Hardware Sizing for Deployment Scenarios

General Notes and Considerations

Below are recommendations on sizing hardware and virtual machine capacity to meet the needs of typical deployment scenarios.

These recommendations are guidelines intended to provide a starting point when deciding on hardware allocation for a specific deployment. We recommend monitoring system load including CPU usage, memory pressure and disk queue length, and adjusting the system sizing accordingly.

For best results, Tasktop Hub should be deployed in an environment that has good network throughput and low latency to all repositories and databases involved in an integration.

Based on real-life metrics, we approximate database sizing at about 40 KB per artifact. For 100,000 artifacts total (including artifacts on both sides of an integration), that equates to about 4 GB of database storage, not including log files, rollback space, etc.

This is a rough estimate, and will depend on customer-specific configuration and usage. For example, artifacts that have hundreds of fields and many large comments will require more space. Likewise, short change detection intervals, frequent full scans, or frequent changes to large numbers of artifacts will require more processing power.

Hub Server Sizing Recommendations

⚠️ The recommendations below offer a **general guideline** only. The performance needs of Hub integrations depend on how integrations are configured, the specifications of connected end systems, and the volume and type of changes made in the end systems.

Note that it is possible for a deployment to have a low number of integrations and users, but a high number of artifact updates (or vice versa). We recommend consulting with Tasktop Support to determine the exact sizing needs for your integration scenario, and for guidance on how to efficiently configure Hub.

Small Deployment

A deployment managing up to 20,000 artifacts in up to 100 projects with up to 10,000 updates/month (typically up to 200 active users, and up to 5 integrations).

- 4 GB system memory
- 3 GHz processor, 2 cores
- 50 GB free disk space

Medium Deployment

A deployment managing up to 150,000 artifacts in up to 500 projects with up to 50,000 updates
/month (typically up to 1,000 active users, and up to 15 integrations).

- 8 GB system memory
- 2 x 3 GHz processor, 4 cores
- 150 GB free disk space

Large Deployment

A deployment managing up to 1,000,000 artifacts in up to 2000 projects with up to 200,000 updates
/month (typically more than 2,000 active users, and 20+ integrations).

- 16 GB system memory
- 4 x 3 GHz processors, 8 cores
- 250 GB free disk space

Extra-Large Deployment

If your deployment exceeds any of the guidelines from the **Large Deployment**, please consult with Taskt
op Support.

For extra-large deployments, the specific characteristics of the integrations are crucial when
determining proper instance sizing. As a result, no general recommendations can be offered for extra-
large deployments.

External Database Sizing

The system that the external database is deployed on should also follow the sizing recommendations
listed above. For example, the database for a large deployment should run on a separate machine with
16 GB of memory, 8 cores, and 250 GB of disk space.

Java Heap Size

We recommend setting the maximum Java heap size value to 50-75% of your system's memory.

Learn more about setting Java heap size here.

Installation

Sandbox Environment

It is recommended that you prepare a sandbox environment to test your Tasktop Hub configuration before deploying it in production.

The sandbox environment should include the following:

- A sandbox server to install Tasktop Hub on
- Sandbox instances of all repositories you will be integrating
    - These instances should include the same project structure and customizations as your production repositories.
    - These instance should also include a comparable number of artifacts to your production repositories.

After you have configured Tasktop Hub on the sandbox server and are satisfied with the way it is running with your sandbox repositories, you can install Tasktop Hub on your production server and recreate the configuration for your production repositories.

Installation

Where to Download Tasktop Hub

To get the latest version of Tasktop Hub, create an account on our Customer Portal, then contact your Solutions Architect or Tasktop Support.

Once logged in to the Customer Portal, click **Product Downloads**.
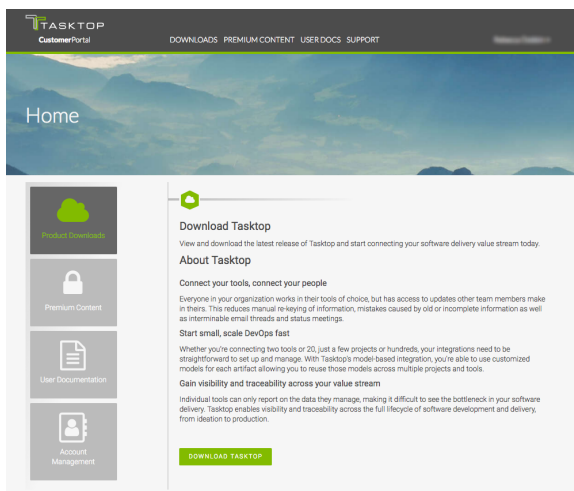
This will lead you to the **Downloads** section, where you can download the latest version of Tasktop Hub.
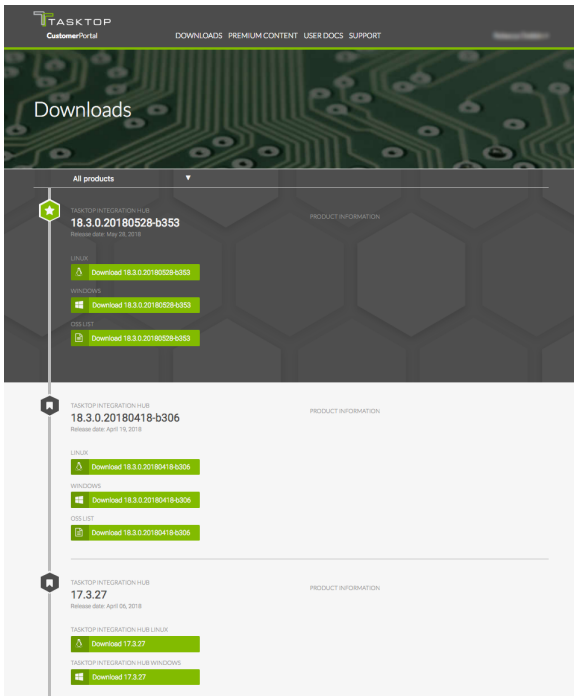
Installation on Windows

Click the **Windows** download link on the **Product Downloads** page of the Customer Portal.

You will be provided with an installation package for Tasktop Hub as a standard Windows MSI installer.

If prompted, click **Save File** and open the file once downloaded.
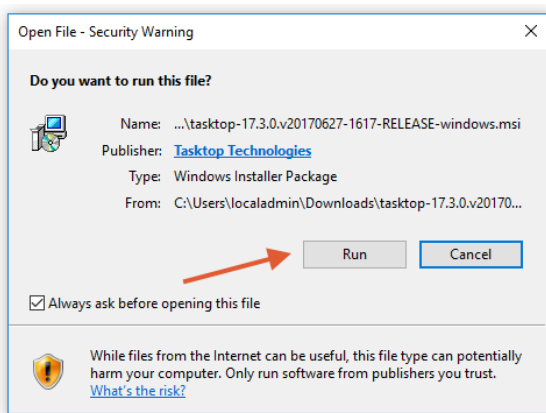
The Tasktop Setup Wizard will guide you through the installation process.

⚠️ **Note**: If you decide to change the location of the ProgramData directory, do **not** include spaces in the new directory name. If the directory includes spaces, Tasktop's UI will **not** be accessible.
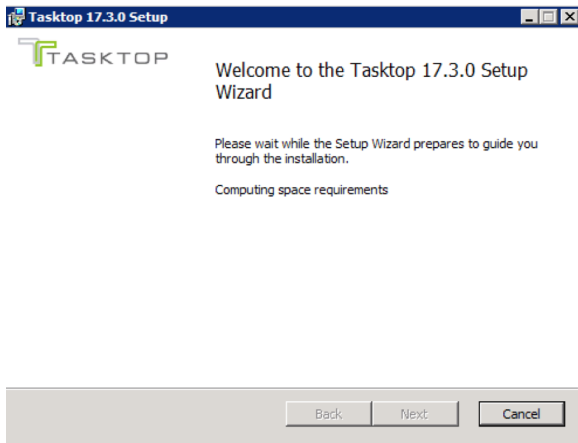
After installing Tasktop, open the **Start** menu and click **Start Tasktop** to start both Tasktop and User Management services.
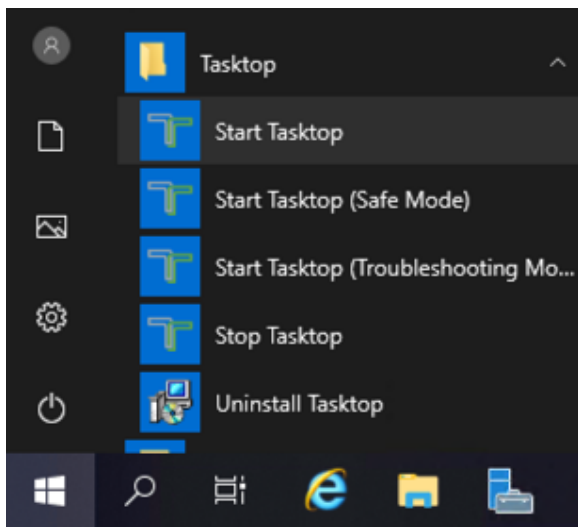
To stop both Tasktop and User Management services, click **Stop Tasktop**.
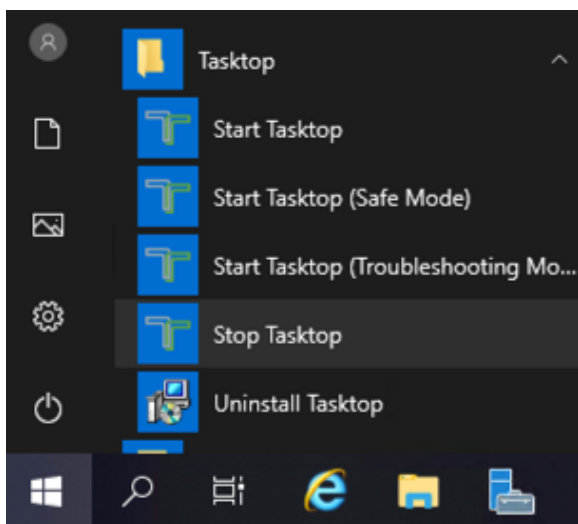
⚠️ **Note**: The Tasktop application is available via HTTPS on port 8443. A default SSL certificate is provided for testing purposes, however this SSL certificate is insecure. **Before using in a production environment, the provided SSL certificate must be replaced**. Please see details in the SSL Certificate Installation section below.

💡 Please follow the steps in the Getting Started section when starting Tasktop Hub for the first time.

Powershell

> ✅ Tasktop Hub: 22.2.6 and later

Tasktop supports Windows environments that require PowerShell scripts to be signed, but Tasktop must be added to the Trusted Publishers store to run on such environments. To do this, see the steps below.

1. Install the latest version of Tasktop Hub on the machine.
2. Log in as the user that will be used to run Tasktop.
3. Run PowerShell as administrator and navigate to `C:\Program` `Files\Tasktop\container\bin\setenv.ps1`
4. Type 'A' and press enter.

Alternatively, PowerShell commands can be used to add Tasktop to the Trusted Publishers store for the entire machine.

Note that the following commands can be run by any user once Tasktop is installed and must be entered into a PowerShell terminal and not run as a PowerShell script.

```
$Cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2
$Cert.Import(((((Get-AuthenticodeSignature "C:\Program Files\Tasktop\container\bin\setenv.ps1").
SignerCertificate).Export([System.Security.Cryptography.X509Certificates.X509ContentType]::Cert)))
$store = Get-Item "cert:\LocalMachine\TrustedPublisher"
$store.Open([System.Security.Cryptography.X509Certificates.OpenFlags]"ReadWrite")
$store.Add($Cert)
$store.Close()
```

Installation on Linux

For Direct Customers

Click the **Linux** download link on the **Product Downloads** page of the Customer Portal.

You will be provided with an installation package for Tasktop Hub as a `.tar.gz` archive.

To extract this archive to your desired location, copy the archive to the correct location on your Linux system.

You must choose a location with **no spaces** in its path and use the following command to extract:

```
$ tar -xzvf tasktop-linux-x64-<version>.tar.gz
```

After extracting, run the `start-tasktop.sh` script from the installation directory (see note on permissions below) to start Tasktop and User Management services.

To stop Tasktop and User Management services, use the `stop-tasktop.sh` script in the same folder.

💡 Please follow the steps in the Getting Started section when starting Tasktop Hub for the first time.

For OEM Customers

You will be provided with a `.bin` installation package for Tasktop Hub.

To execute the file, run these commands:

```
chmod +x tasktop-linux-x64.bin
```

```
./tasktop-linux-x64.bin
```

After approving the End User License Agreement, the file will automatically unzip, allowing you to run Tasktop Hub.

Run the `start-tasktop.sh` script from the installation directory (see note on permissions below) to start Tasktop and Keycloak User Management services.

To stop Tasktop and User Management services, use the `stop-tasktop.sh` script in the same folder.

⚠️ The Tasktop application is available via HTTPS on port 8443. A default SSL certificate is provided for testing purposes, however this SSL certificate is insecure. **Before using in a production environment, the provided SSL certificate must be replaced.** Please see details in the SSL Certificate Installation section below.

💡 Please follow the steps in the Getting Started section when starting Tasktop Hub for the first time.

Note on Permissions

We recommend creating a dedicated user for running Tasktop Hub. We do **not** recommend running Tasktop Hub as root, as it may create files that cannot be accessed when running Tasktop as another user. Running an application on a Linux system as root may also interfere with your system's security.

For this reason, `start-tasktop.sh` will not start if it detects the current user is root.

If you would like to run Tasktop Hub as root despite these risks, you can do so by deleting or commenting lines 3-7 of `start-tasktop.sh` as shown below:

```
#!/bin/sh
#if [ "`id -u`" -eq "0" ]
#then
#    echo "Tasktop should not be run as root"
#    exit 1
#fi
currentdir="$( cd "$(dirname "$0")" ; pwd -P )"
keycloak_running() {
  pgrep -n -f "${currentdir}"/keycloak/bin/standalone.sh
}
```

Tasktop Hub Service on Linux

There are several ways to configure a Tasktop Service that starts automatically on system startup. We recommend using a dedicated account for running Tasktop Hub.

You can see the examples below for **Systemd** and **SysVinit**.

Tasktop Hub Service with Systemd

1. Navigate to `/etc/systemd/system`
2. Create a file named `tasktop.service`
3. Paste the following into the file:

   ```
   # Systemd unit file for tasktop
   [Unit]
   Description=Tasktop Integration Hub
   After=syslog.target network.target

   [Service]
   Type=forking

   ExecStart=/path/to/tasktop/start-tasktop.sh
   ExecStop=/path/to/tasktop/stop-tasktop.sh

   User=user
   Group=group

   [Install]
   WantedBy=multi-user.target
   ```

   a. Change both instances of `/path/to/tasktop` to the full path to your Tasktop Hub installation directory
   b. Change the User and Group variables to the username and group of the account you want to run the Tasktop Hub service
4. Reload Systemd

   ```
   $ systemctl daemon-reload
   ```

5. Enable the new Tasktop Hub service to start on system startup

   ```
   $ systemctl enable tasktop
   ```

💡 To manually start and stop the Tasktop Hub Service, use the following commands:

```
$ systemctl start tasktop
$ systemctl stop tasktop
```

1. Navigate to `/etc/init.d`
2. Create a file named `tasktop`
3. Paste the following into the file:

```
#!/bin/bash
# description: Tasktop Start Stop Restart
# processname: tasktop
# chkconfig: 2345 20 80
TASKTOP_HOME=/path/to/tasktop
case $1 in
start)
sh $TASKTOP_HOME/start-tasktop.sh
;;
stop)
sh $TASKTOP_HOME/stop-tasktop.sh
;;
restart)
sh $TASKTOP_HOME/stop-tasktop.sh
sh $TASKTOP_HOME/start-tasktop.sh
;;
esac
exit 0
```

   a. Change the TASKTOP_HOME variable to the full path to your Tasktop Hub installation directory
   b. If you'd like, you can change the `chkconfig` run levels and start and stop priorities
4. Set the permissions of Tasktop to make it executable:

```
$ chmod 755 tasktop
```

5. Use the `chkconfig` utility to enable Tasktop Hub start at system startup

```
$ chkconfig --add tasktop
$ chkconfig --level 2345 tasktop on
```

    a. If you'd like, you can change the run levels in this command

💡 To manually start and stop the Tasktop Hub Service, use the following commands:

```
$ service tasktop start
$ service tasktop stop
$ service tasktop restart
```

SSL Certificate Installation

⚠️The Tasktop application is available via HTTPS on port 8443. **A default SSL certificate is provided for testing purposes and should be replaced after installation.**

Replacing the default SSL certificate used by Tasktop Hub involves the following:

1. Preparing a Java keystore file with all keys and certificates
    a. The Tasktop and Keycloak SSL configuration require a JKS format keystore.

i. If your corporate CA provides a JKS keystore file, you can skip to the **Configure Tasktop to use the keystore** section and follow the steps using the JKS keystore file from your CA.

ii. If your CA requires you to provide a CSR and returns a certificate response to you, use the following steps to generate your own keystore file and CSR:

1. Create a Java keystore file and generate a new key pair
2. Generate a certificate request file
3. Submit the file to a Certificate Authority (CA) and obtain the certificate and CA certificate trust chain
4. Import the certificates to the keystore file

2. Configuring Tasktop to use the keystore (i.e., new key and certificate)

The SSL certificate should contain DNS names where the Tasktop server is accessible. The user's browser will verify that the name in the address bar matches the names listed in the certificate.Certificate Authority may be your internal corporate service, or you may use a public CA (e.g., Comodo, Let's Encrypt). If you are planning to use a certificate from a public CA, your Tasktop instance must have a publicly recognizable DNS name that is owned by your organization.

SSL-related instructions on this page are provided as a **reference only**. Your Certificate Authority will have more detailed instructions on creating and importing certificates. These instructions are based on the use of a GUI tool Portecle, which can be downloaded here.

Note that Tasktop does not provide support for this third-party tool beyond the instructions shown below.

💡 **Tip**: You can create the Java keystore file on any machine and move the file to the server running Tasktop software; there is no need to install Portecle on the server running Tasktop.

If you cannot use Portecle and need to utilize standard Java command line utility keytool, please refer to Tomcat documentation. Upon following the documentation, use JRE installed with Tasktop software in the Tasktop installation directory (default `C:\Program Files\Tasktop`). Tasktop's `server.xml` file is located in Tasktop's data directory (default: `C:\ProgramData\Tasktop`, or the location where Tasktop is installed on Linux) under `container/conf/server.xml`.

Running Portecle for SSL certificate installation

To run Portecle for SSL certificate installation, see the instructions below:

1. Download and unzip Portecle.
2. Open the command prompt.
   a. For **Windows**, navigate to `C:\Program Files\Tasktop\jre\bin\`
   b. For **Linux**, navigate to `<tasktop-install>/jre/bin/`
3. Run the following command (changing `/path/to/portecle/` to the location where you unzipped Portecle):
   a. `java -jar /path/to/portecle/portecle.jar`

# Prepare a Java keystore file with all the keys and certificates

To replace Tasktop's default SSL certificate using Portecle, follow the instructions below:

💡**Tip**: Details on accessing Portecle can be found in the section above.

1. Create a key pair and keystore:
   a. Start Portecle and click **New Keystore** in the toolbar and select **JKS** as the keystore type.
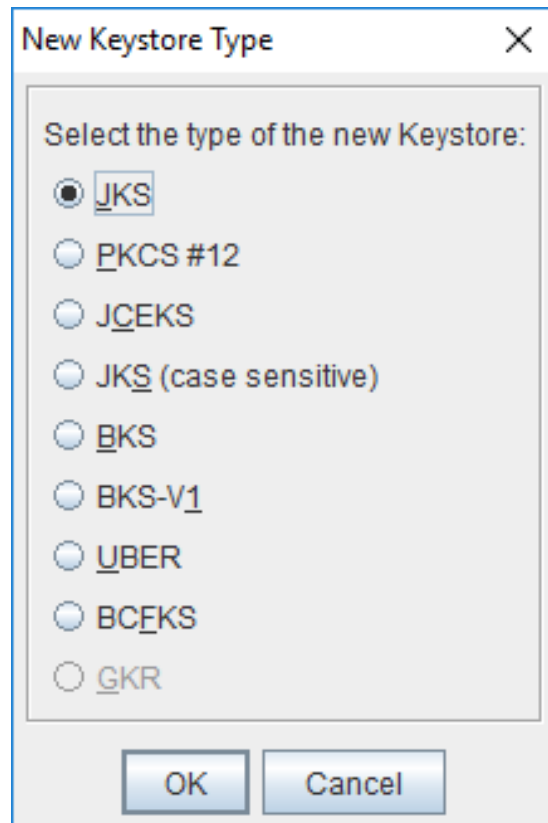
   a.
   b. Click **Generate Key Pair** in the toolbar. You can leave the default settings for 2118 bit RSA key, or choose different settings if required by your company's security policy.
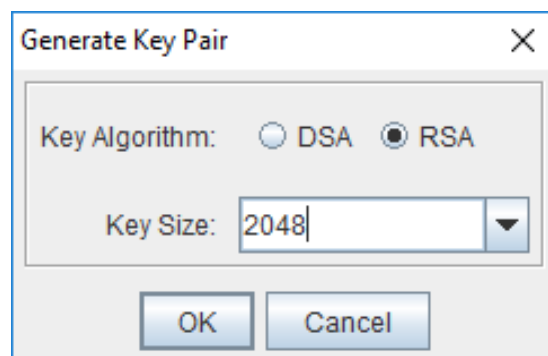
   a.
   c. In the **Generate Certificate** pop-up, enter the Fully Qualified Domain Name (FQDN) of your Tasktop server in the Common Name (CN) field and enter other fields as needed.

      a. In the **Subject Alternative DNS Name** field, enter the alternative domain name of the server, if one exists. Your certificate should include all DNS names that your users may use to connect to Tasktop. For internal corporate CA you can also use "short"

names (i.e., tasktop, in addition to tasktop.acme.corp). In CA, these additional DNS names are called Subject Alternative Names, or SAN. You can specify one SAN at this point, and can usually add more names later when submitting your request to the CA.
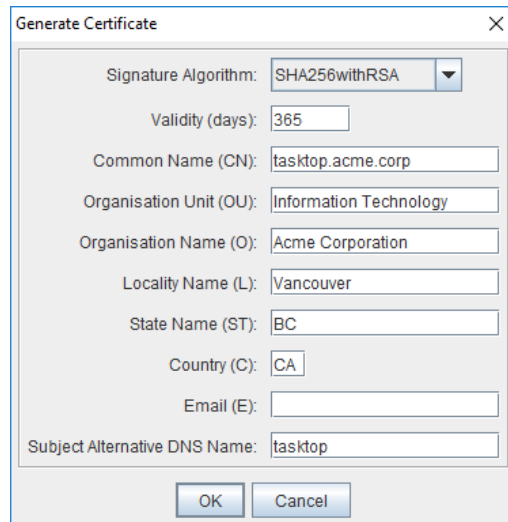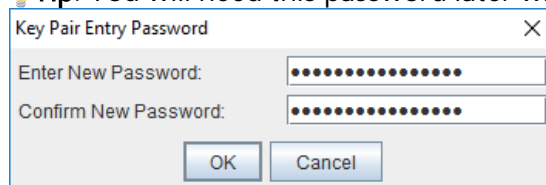

a.

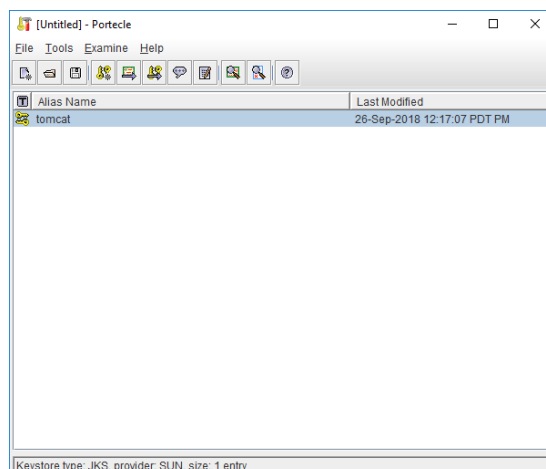d. Enter **tomcat** as alias.
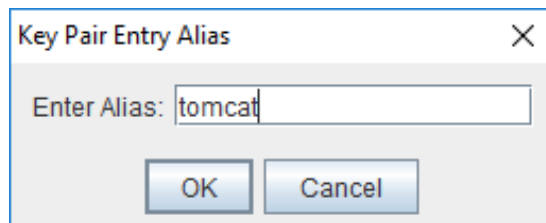
a.

e. Create a new password for the key pair.

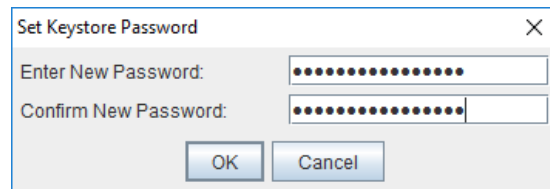a. 💡**Tip**: You will need this password later when configuring Tomcat.


b.

f. You will see your newly created key pair in the list.

a.

g. Click **Save Keystore** in the toolbar to save the newly created keystore file. Here, use the same password that you entered for the key pair earlier.

a.

2. To generate a certificate request file (also known as Certificate Signing Request or CSR), right click on the **tasktop** key and select **Generate Certification Request** and save it to a file.
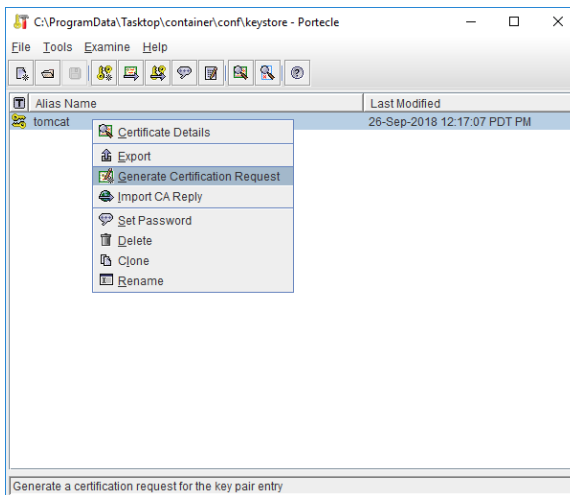
1.

3. Submit your CSR to a CA to obtain a Certificate.

💡 **Note**: For some CAs you will need to provide the list of all DNS names for your Tasktop server separately as they will ignore the SAN values in the certificate request. See your CA's documentation for more information.

4. Import the certificates to the keystore file.
    a. If your CA provided a separate file with the CA certificate or trust chain, import it by selecting **Import Trusted Certificate** in the toolbar. If your CA provided only one file in response to your CSR, skip to 4b.
    b. Import the server certificate by right clicking on the **tasktop** key, selecting **Import CA Reply** and choosing the server certificate file received from the CA.
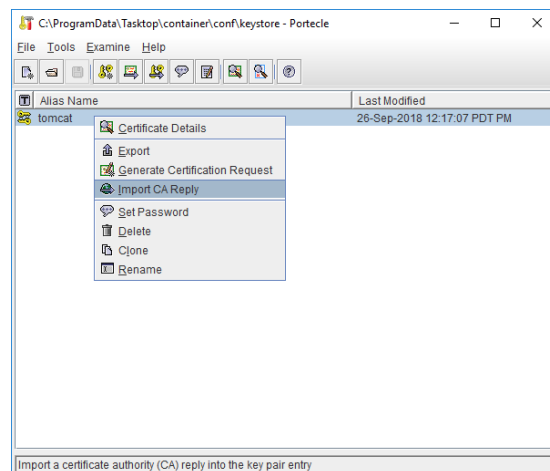
a.

    c. To verify the certificate chain, click **Tools** and then click **Keystore Repor**t.

Configure Tasktop to use the keystore

1. Place your keystore file in a protected location that will not be wiped on Tasktop upgrade. We suggest using Tasktop data directory (default `C:\ProgramData\Tasktop`, or the home directory of the user that Tasktop service is running as on Linux).
2. Open the `tasktop-hub.properties` file and configure the following properties:
   a. `server.ssl.key-store` - Location where the keystore file exists
   b. `server.ssl.key-store-password` - Password of keystore file
   c. `server.ssl.key-store-type` - Type of keystore file (e.g., JKS, PKCS12)
3. Restart Tasktop Hub Service

💡 To learn more about creating a `tasktop-hub.properties` file, please see the section below.

By default, the SSL configuration has been configured to disable known weak ciphers. As new security information becomes available, the list of enabled ciphers should be updated accordingly.

Configure Keycloak User Management to use and trust Tasktop's keystore

In Tasktop Hub 20.4, both Tomcat and Jboss share the same properties in the `tasktop-hub.properties` file as they share the same keystore file. See more details above.

💡 To learn more about creating a `tasktop-hub.properties` file, please see the section below.

Port Configuration

By default, Tasktop utilizes the ports listed in the table below.

If any of those ports are already being utilized for other purposes, you will need to change them. To view a list of all ports being used on your system, you can use the netstat-a command. This will help you determine which available ports you would like to use for Tasktop.

Here is a summary of each port Tasktop utilizes and the location where you can change it if it is already being used:

Port	Location	Purpose

8080
 	
8443
	`tasktop-hub.properties`
	  #server.port=8443
	
	  #server.redirect.port=8080
	
	More details here
		Default port Tasktop uses for HTTP (8080) / HTTPS (8443)

8081 8444		User Management (Keycloak) HTTP Ports

	`tasktop-hub.`
	`properties`
	    #jboss.http.
	port=8081
	
	    #jboss.https.
	port=8444
	
	More details [here](#)	

Additional Keycloak Ports:
 	
- 9990
- 9993
- 8009
- 4712
- 4713
- 25
 	
More details [here](#)
 	
(**Note**: the following ports have been modified from the Keycloak defaults: 80808081, 8443 8444)
	`tasktop-hub.`
	`properties`
	    #jboss.ajp.
	port=8009
	
	    #jboss.
	management.http.
	port=9990
	
	    #jboss.
	management.https.
	port=9993
	
	    #jboss.txn.
	recovery.
	environment.
	port=4712
	
	    #jboss.txn.status.
	manager.port=4713
	
	    #jboss.mail.smtp.
	port=25
		User Management (Keycloak)
		
		More details [here](#)

8005	`tasktop-hub.`
	`properties`
	    #server.shutdown.
	port=8005
		Tomcat Shutdown Port

Tasktop Hub Port

The default port Tasktop uses is 8443 for HTTPS and 8080 for HTTP, which redirects to HTTPS. If you'd like to change these ports to ease access for your users, or to accommodate a proxy, follow these instructions:

1. Open the tasktop-hub.properties file and configure the following properties:
   a. `server.port` - The http or https port
   b. `server.redirect.port` - The port that, if accessed, redirects to **`server.port`**
2. After changing the port, the address used to access Tasktop (i.e., http://localhost:8080) will need to be updated with the new port number in place of '8080.'

Please refer to the official documentation for additional configuration options.

To learn more about creating a `tasktop-hub.properties` file, please see the section below.

User Management (Keycloak) Port

The default port for User Management is 8081. If you'd like to change the port that User Management (Keycloak) utilizes, follow the instructions here. If your User Management (Keycloak) utilizes a port other than 8081, you can instruct Tasktop to access User Management (Keycloak) via the correct port by following the instructions below.

1. Open the tasktop-hub.properties file and configure the following properties:
   a. `jboss.http.port` - Jboss http port
   b. `jboss.https.port` - Jboss https port

💡**Note**: If you change the default jboss management-http port setting in the `/keycloak/standalone /configuration/standalone.xml` to something other than 9990, you must also update the port referenced in `/keycloak/bin/jboss-cli.xml.`

To learn more about creating a `tasktop-hub.properties` file, please see the section below.

Getting Started

Once installation is complete, you can begin using Tasktop Hub by opening **https://localhost:8443/** in any of our supported browsers.

Before logging on to Tasktop, you must log into the **User Administration Console** in order to create your admin user(s). The Tasktop User Administration Console can be accessed via the **User Administration Console link** at the bottom of the Tasktop Hub login page. Please review the User Management section for detailed instructions on how to create a user, login, and manage your user accounts.

Once logged in, you will be prompted to set a Master Password, which will be used to encrypt your repository credentials.

You will also need to apply your license before configuring your integrations. You can learn how to apply your license here.

Externalized Configuration

Tasktop enables you to externalize configurations from Tomcat, Jboss/Keycloak, and certain application properties in a single place. This allows you to use property files to override default values such as:

- **Jboss**: ports (e.g., http, https, management port), Keycloak database paths, Keycloak trust stores, java memory variables, and custom system properties
- **Tomcat**: ports (e.g., http https), keystores (e.g., files, passwords, types), java memory variables, and custom system properties
- **Application Properties**: Derby, Tasktop Hub, Liquibase, log4j, and keycloak host

To override default values through a properties file, you must provide the `tasktop-hub.properties` file in a directory that Hub can scan and read.

This can be done as follows:

1. Rename the file `tasktop-hub.properties.default` to `tasktop-hub.properties`.

    a. For **Windows**, this file can be found in the **App Data Directory**.
    b. For **Linux**, this file can be found in the root level of the `.tar.gz` package.
        a. **Note**: For Linux users, we recommend creating an environment variable named `TASK TOP_HOME` with its value pointing to an exclusive directory where the `tasktop-hub.properties` file will be placed.
2. Provide values to properties that need to be overridden.
    a. For example, if you'd like to change the Tomcat https port to port 9443, uncomment the property from `#server.port=8443` to `server.port=9443`

Good to Know:

- Only properties/lines uncommented within the `<AppDataDirectory>/tasktop-hub.properties` file will be applied, otherwise Tasktop Hub will assume default values for commented properties.
- Only properties at `<AppDataDirectory>/tasktop-hub.properties` file will be used; the file `<AppDataDirectory>/tasktop-hub.properties.default` is just a template and will not work in Tasktop Hub.

Upgrading

Upgrading on Windows

The `tasktop-hub.properties` file will not be replaced or deleted during the installation/upgrade process. For this reason, newer versions of Hub can retain settings automatically after upgrading.

Upgrading on Linux

Because the properties file is placed in the `$TASKTOP_HOME` directory, newer versions of Hub will automatically apply all configurations.

If the properties file is not placed in the `$TASKTOP_HOME` directory, it is necessary to copy the properties file from the old installation directory to the new installation directory.

Upgrading from a Version Earlier than 20.4

If you have made manual changes to Tomcat and/or Jboss files, you have two options upon upgrading to 20.4:

Option One

You can apply all configurations that have been applied manually to `server.xml, standalone.xml ,standalone.conf, standalone.conf.bat, setenv.sh,` and `Manage Tasktop -> Java -> Java Options` to the `tasktop-hub.properties` file .

During an upgrade, it is not necessary to override the `server.xml` file from the old version to the new installation directory. This can be done by simply providing the `tasktop-hub.properties` file in a directory that Tasktop Hub is able to read and ensuring that there is an uncommented line as shown below:

```
...
server.port=9443
...
```

💡 Other properties can be configured the same way as shown in the example above.

Option Two

You can copy all configuration files from Tomcat and/or Jboss that were previously modified and override them in the new version directories.

Properties

The `tasktop-hub.properties` file contains three main blocks:

- Jboss/Keycloak Properties
- Tomcat Properties
- Tasktop Hub Properties

Jboss/Keycloak

The properties listed in the table below are used only if Tasktop Hub is using Keycloak as an Authentication Provider. When provided, the properties file will be passed as an argument of `standal one.sh/standalone.bat` (e.g., `standalone.sh|bat --properties=<path>/tasktop-hub. properties`), which means that the file will override Jboss variables.

Property	Purpose	Notes

`jboss.`
`ajp.`
`port`
	Use this property to provide a value for the tag < `socket-binding`
	`name="ajp" />` within the `standalone.xml` descriptor.	

`jboss.`
`http.`
`port`
	Use this property to provide a value for the tag < `socket-binding`
	`name="http" />` within the `standalone.xml` descriptor.	

`jboss.`
`https.`
`port`
	Use this property to provide a value for the tag < `socket-binding`
	`name="https" />` within the `standalone.xml` descriptor.	

`jboss.`
`managem`
`ent.`
`http.`
`port`
	Use this property to provide a value for the tag < `socket-binding`
	`name="management-`
	`http"/>` within the `stand`
	`alone.xml` descriptor.
		If this property is provided, the script `<installation>`
		`/keycloak/bin/jboss-cli-tasktop.sh|bat` will call the script `<installation>/keycloak/bin`
		`/jboss-cli.sh|bat` passing these arguments: `--contr`
		`oller=localhost:`**`<jboss.management.http.`**
		`port>` `--properties=`**`<path>`**`/tasktop-hub.`
		`properties`.

`jboss.`
`managem`
`ent.`
`https.`
`port`
	Use this property to provide a value for the tag < `socket-binding`
	`name="management-`
	`https"/>` within the `stan`
	`dalone.xml` descriptor.	

`jboss.`
`txn.`
`recover`
`y.`
`environ`
`ment.`
`port`
	Use this property to provide a value for the tag < `socket-binding`
	`name="txn-recovery-`
	`environment"/>` within the `standalone.xml` descriptor.	

`jboss.` `txn.`	Use this property to provide a value for the tag < `socket-binding` `name="txn-status-`	

`status. manager .port`	`manager"/>` within the standalone.xml descriptor.	
`jboss. mail. smtp. port`	Use this property to provide a value for the tag `< remote-destination host="localhost" />` within the `standalone. xml` descriptor.	
`jboss. server. data. dir`	Use this property if you want to place the keycloak database in a custom directory.	This is the same directory where keycloak database lives. For both Windows and Linux, the directory separator needs to be '/'.
`jboss. java. memory`	Use this property to change memory settings.	We recommend setting the maximum Java heap size value to 50-75% of your system's memory.
`jboss. custom. system. propert ies`	Use this property to load custom system properties. For example: `-Djboss. *=value, - Dkey=value, -XX: key=value, - javaagent:value, - agentlib:value`	

Tomcat

The properties listed in the table below are used to override some properties from Tomcat.

Property	Purpose	Notes
`server.port`	Use this property to provide a value for the attribute port in the tag `<Connector/>` within the `se rver.xml` descriptor.	After changing the port, if Keycloak is being used, you will need to go into the User Administration Console and adjust the client to the new port.
`server. redirect.port`	Use this property to provide a value for the attribute redirectPort in the tag `<Connect or/>` within the `server.xml` descriptor.	

`server.shutdown.port`	Use this property to provide a value for the attribute port in the tag `<Server/>` within the `server.xml` descriptor.	
`server.tomcat.connection-timeout`	Use this property to provide a value for the attribute connectionTimeout in the tag `<Connector/>` within the `server.xml` descriptor.	

`server.ssl.key-store=/path/to/keystore-file`	Use this property to provide a value for the attribute keystoreFile in the tag `<Connector/>` within the `server.xml` descriptor.	This property is shared with Jboss /Keycloak. The `standalone.xml` file is reading this property:
		
		`<spi name="truststore">`
		
		...
		
		`<property name="file"value="${server.ssl.key-store:${jboss.home.dir}/../insecureKeystore}"/>`
		
		...

`server.ssl.key-store-password=changeit`	Use this property to provide a value for the attribute keystorePass in the tag `<Connector/>` within the `server.xml` descriptor.	This property is shared with Jboss /Keycloak. The `standalone.xml` file is reading this property:
		
		`<spi name="truststore">`
		
		...
		
		`<property name="password"value="${server.ssl.key-store-password:changeit}"/>`
		
		...

`server.ssl.key-store-type=JKS`	Use this property to provide a value for the attribute keystoreType in the tag `<Connector/>` within the `server.xml` descriptor.	
`server.ssl.key-alias`	Use this property to provide a value for the attribute keyAlias in the tag `<Connector/>` within the `server.xml` descriptor.	Enable this property only if your custom Keystore has an alias and it is different than Tomcat.

`tomcat.java.`
`memory=-`
`Xms256M -`
`Xmx2118M`
	Use this property to change memory settings.	We recommend setting the maximum Java heap size value to 50-75% of your system's memory.
		
		For Windows: Initial memory pool size (-Xms) and maximum memory pool size (-Xmx) needs to be in MB. That means that the value needs to be suffixed with 'M'.
		
		Values suffixed with 'G' will cause an error at the start of Hub.
		
		For Linux: Values can be specified in MB or GB. Both suffixes 'M' and 'G' work.

`tomcat.java.`
`errorFile=/path`
`/to`
`/hs_err_pid%%p.`
`log`
	Use this property to provide a custom path for `-XX:ErrorFile`.	

`tomcat.java.io.`
`tmpdir=path/to`
`/temp`
	Use this property to provide a custom path for `java.io.`
	`tmpdir` directory.	

`tomcat.java.`
`util.logging.`
`config.`
`file=path/to`
`/logging.`
`properties`
	Use this property to provide a custom path for Tomcat's `loggi`
	`ng.properties` file.	

`tomcat.jdk.tls.`
`rejectClientIni`
`tiatedRenegotia`
`tion=true`
	Use this property to provide `jdk.`
	`tls.`
	`rejectClientInitiatedRene`
	`gotiation` value.	

`tomcat.custom.`
`system.`
`properties`
	Use this property to load custom system properties such as:
	
	`-XX:key=value, -`
	`javaagent:value, -`
	`agentlib:value`	

Tasktop Hub

The properties listed in the table below are used to override some Tasktop Hub values.

Property	Purpose	Notes
`derby.storage.pageCacheSize`	Use this property to change the data page cache in the database.	Reference: https://db.apache.org/derby/docs/10.14/ref/rrefproper81359.html
`derby.system.home=/path/to/db`	Use this property to provide a custom path to the Derby database directory.	Providing the Derby database directory is useful for Linux environments when upgrading, as you do not need to copy files from the old installation directory to the new installation directory.
`hub.database.configuration.directory=/path/to/db`	Use this property to provide a custom path to the Derby database.	
`liquibase.ignoreRecycleBinWarning=true`	Use this property to whether or not suppress liquibase warnings.	
`log4j.configuration=file:/path/to/log4j2.xml`	Use this property to provide a custom path to the `log4j2.xml` file.	
`log4j.configuration.verbose=file:/path/to/log4j2-troubleshooting.xml`	Use this property to provide a custom path to the `log4j2-troubleshooting.xml` file.	
`hub.security.cors.exclusionPaths`	Use this property to provide a list of paths that will be excluded from the CORS verification. For example: `/first-path, /second-path`	Prior to version 21.1, this property was configured in `/tasktop/container/webapps/root/WEB-INF/web.xml`

Good to Know

Windows

- It is not possible to use environment variables to compound values. Properties related to paths must be configured using an absolute path.
- Properties must be modified in the `tasktop-hub.properties` file as this file has more priority than properties modified in `Manage Tasktop > Java > Java Options | Initial memory pool | Maximum memory pool`.

Linux

- It is possible to use environment variables to compound a specific value. As an example, it is possible to use $CATALINA_BASE to compound a path.

```
hub.database.configuration.directory=$CATALINA_BASE/../../directory
log4j.configuration.verbose=file:$CATALINA_BASE/../../log4j2-troubleshooting.xml
```

Default File Locations

Windows

When Tasktop Hub is installed on Windows using the MSI installer, the program files (i.e., the executable files and binaries) are located in `C:\Program Files\Tasktop`; configuration files and logs are located in `C:\ProgramData\Tasktop`.

💡 **Tip**: ProgramData may be a hidden folder, so you will need to change your Windows Explorer settings to show hidden files and folders to find it.

⚠️ **Note**: If you change the location of the `ProgramData` directory to an alternate location, do **not** include spaces in the name of the new directory. If the directory has spaces in its name, Tasktop's UI will not be accessible.

Linux

When Tasktop Hub is installed on Linux, the program files (i.e., the executable files and binaries), configuration files, and logs are all located in the installation directory where you extracted the distribution archive.

⚠️ **Note**: You must choose a location with **no spaces** in its path, or Tasktop's UI will not be accessible.

Repository Preparations

Preparing Your Repositories

In Tasktop, the term **repository** refers to the external tools Tasktop connects to (e.g., Atlassian Jira, ServiceNow, BMC Remedy, etc).

Before connecting Tasktop Hub to your external repositories, you will need to perform some simple preparation on each repository you will be integrating. This preparation includes creating a user account for Tasktop Hub with the appropriate permissions. Please refer to our Connector Docs for detailed instructions for each repository.

Firewalls and Proxies

If Tasktop is installed behind a firewall, you may need to connect to external repositories (e.g. hosted or cloud ALM tools) through a proxy. To create a connection to such external repositories in Tasktop, you can make Tasktop connect through your proxy by configuring the proxy settings when creating a new repository connection. It is recommended to create login credentials specifically for Tasktop on the proxy server.

💡 Note that the Proxy Location must be a URL in order for the proxy connection to work. If a .pac script is used in your browser, you will need to open the script and find the URL/port to enter in the Location field.

To use a proxy server, check the **user proxy server** box and fill in your proxy details in the **Proxy Server** section on the New Repository Screen:

Troubleshooting

Troubleshooting Mode

Troubleshooting Mode allows you to turn on verbose logging when the UI doesn't appear due to issues upon startup.

To start Tasktop in Troubleshooting Mode, see the following instructions:

- **For Linux**: Run the following script from the command line in the installation folder:

```
start-tasktop-troubleshooting-mode.sh
```

- **For Windows**: Click **Start Tasktop (Troubleshooting Mode)** in the Start menu.
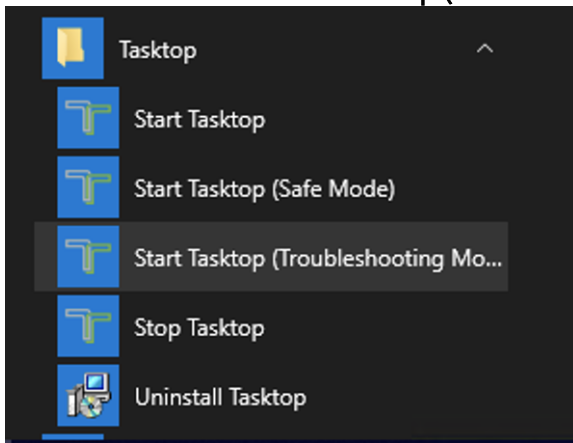
Note: The default troubleshooting duration is set to two hours when Troubleshooting Mode is enabled. You can view the Troubleshooting timer in the Troubleshooting tab on the Settings screen.

SafeMode

SafeMode allows you to start Hub without running your integrations (i.e., no synchronization or change detection will occur).

To start Tasktop in SafeMode, see the following instructions:

- **For Linux**: Run the following script from the command line in the installation folder:

```
start-tasktop-safe-mode.sh
```

- **For Windows**: Click **StartTasktop (Safe Mode)** in the Start Menu.
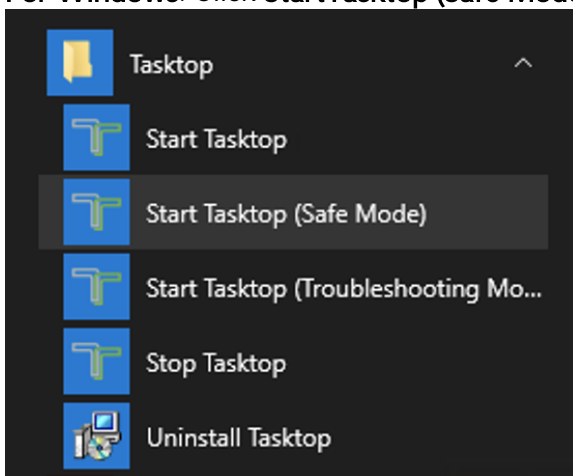
Advanced Configuration

Container Configuration

Tasktop is distributed with the **Apache Tomcat Servlet Container**.

For information on configuring the container, refer to Apache Tomcat documentation.

On **Windows**, configuration and log files are installed under `C:\ProgramData\Tasktop` while program files are located under `C:\Program Files\Tasktop`.

For information on configuring the service, refer to Apache Tomcat Service How to.

Further configuration, including JVM options and memory allocation, can be performed for the Windows service by launching **Tasktop Properties** located at `C:\Program Files\Tasktop\container\bin\tasktopw.exe`.

Increasing Available Memory

Beginning in Hub version 20.4, configurations are externalized from **Tomcat**, **Jboss/Keycloak**, and certain application properties in a single place. This allows you to use property files to override memory variables and custom system properties.

For more information on changing memory settings, please refer to the properties table.

Logging

Logging is configured with `log4j2`. See the included `log4j2.xml` to configure log levels, location, and rolling policy.

The included `log4j2-troubleshooting.xml` configures `log4j2` for the troubleshooting log level when set via the Settings screen.

Upgrading

> ⚠ Beginning in the Tasktop Hub 20.4 release, a properties file is used for configurations such as Ports & SSL. To learn more about this update, click here.

> ⓘ Please be aware of additional upgrade steps needed for the scenarios outlined below.
>
> **When upgrading from earlier versions to 21.1.x:**
>
> - As 21.1.x is a checkpoint release, be aware that you **must** upgrade to version 21.1.x **before** upgrading to a later version.
> - If you do not follow instructions here, you may encounter issues with artifact association management that prevent you from viewing and deleting artifact pairs.
> - **Note**: Be aware that upgrading to 21.1.x from an earlier version may take longer than usual.
>
> **When upgrading from a version earlier than 19.3.0.20190603:**
>
> - If you do not follow instructions here, you may experience errors that prevent pages from loading or be unable to log out of Tasktop.
> - If you are upgrading from a version that is *also earlier than 19.2.1*, *please follow the additional instructions below:*
>
>   - While we always recommend backing up the operational database, it is imperative that a backup is made prior to upgrading to 19.2.1 or later. Upon upgrade from a version earlier than 19.2.1 to 19.2.1 or later, a one-time change to the operational database will occur that may take an hour or longer to complete. During the upgrade process, the UI will not be available. To monitor the upgrade process, please inspect the log files. You can find more details in our FAQ here.
>
> **When upgrading from earlier versions (e.g., 18.3 and earlier) of Hub to newer versions:**
>
> - You may need to perform a two-step upgrade to prevent an upgrade failure.

Backup

A working backup strategy is a critical element of disaster recovery, since only backups can mitigate complete hardware failure and user error. A strategy that ensures correct and current backups is essential. Backups of the Tasktop database include both configuration and operational data.

Backup frequency should mirror your practices for all software tools your organization utilizes. Backup frequency should be daily, ideally with incremental backups performed more frequently.

General Application Configuration

The recommended practice is to back up the entire installation/program data directory to cover all customizations (excluding logs)

- Back up Tomcat customizations (in Linux install directory or Windows Program Data)
  - `container/conf/server.xml`
  - Any keystores for certificates
  - For **Linux**: `bin/setenv.sh`
  - For **Windows**: any changes to the Java section of the Manage Tasktop application (e.g., memory, command line parameters, etc)
- Back up keycloak data and customizations
  - `keycloak/standalone/data`
  - `keycloak/standalone/configuration/standalone.xml`

Note: We recommend backing up the `standalone.xml` file in the event of a rollback; however, when upgrading, this file should not be preserved to prevent keycloak errors. Instead, the `standalone.xml` file in the new Hub package should be used.

Operational Data

⚠️ Tasktop automatically stores operational data to a built-in database. However, for production environments, we strongly recommend that operational data is stored to an external database for improved maintainability. This enables you to perform frequent backups without stopping Tasktop Hub and ensures that your Tasktop practices are consistent with your existing disaster and recovery process. For details on how to store your operational data to an external database rather than Tasktop's built-in database, please refer to General (Settings).

If utilizing Tasktop's built-in Derby Database, ensure you've backed up the following:

- File backup of db directory (in Linux install directory or Windows Program Data)

Restore from Backup

If Tasktop fails to restart after an upgrade or if there are unresolvable errors preventing your integrations from running, Tasktop may need to be returned to the previous version. Please ensure to stop Tasktop before restoring to a previous version.

⚠️ **Note**: If restoring from backup, you should be cautious as the state of the integration is maintained in the database and restoring to an older version could result in duplicated items and data (e.g., comments and attachments). It is recommended to only restore when directed by Tasktop support or after a failed upgrade where no items were processed.

💡 **Tip**: If integrations were resumed individually during an upgrade, you can prevent duplicating items and data when restoring to an earlier version by utilizing the upgrade from backup file feature described below.

General Application Configuration

You should restore any changes identified in the backup.

Operational Data

In order to restore Tasktop Hub, follow the instructions below:

1. Copy the database directory from backup to the Tasktop data folder.
2. Restore the backed up Tomcat and Catalina configuration files from part 4 of the backup instructions.

Operating Systems

1. Shut down Tasktop.
2. Uninstall Tasktop, then run the previous installer.
3. Restore from backup as described in section above.
4. Restart Tasktop.
   a. If you'd like, you can restart Tasktop in 'safe mode' which ensures all integrations are paused.

Upgrading

Before you Upgrade

⚠ Before upgrading Tasktop, be sure to do the following:

1. Shut down Tasktop and afterwards follow the backup instructions outlined above. The first time that Tasktop restarts after an upgrade, the internal database will be migrated to the new version and it will no longer be possible to return to the prior version without the backup.
2. Additionally, ensure that backups are made of the Tomcat, Catalina, and Keycloak configuration files that have been customized. The upgrade process will overwrite these configuration files and customizations will need to be re-applied.
3. When Tasktop is upgraded, a service-downtime for the Tasktop service is required in order to upgrade the database. Note that a second instance cannot be running while the first instance is attempting to upgrade the database.
   a. To understand implications of Tasktop downtime, please see here.
4. Please review the release notes for all Tasktop versions that have been released after the version you are upgrading from. Ensure that any upgrade steps outlined in the release notes are followed.

1. Ensure a copy of the old installer is available in case a roll-back is required.
2. Click the **Stop Tasktop** button on your desktop, and make sure services are stopped:

3. Backup as described in section above.
4. Run the installer of the new version of Tasktop.
5. Re-apply Tomcat/Keycloak configurations.

    a. Upgrading from versions *earlier* than 20.4:

        i. Apply all customizations done in (`<install-location>/container/conf/server.xml`) to the `tasktop-hub.properties` file. More details about translating configurations from `server.xml` to the new properties file can be found here.

        ii. Apply all customizations that have been done in (`<install-location>/keycloak/standalone/configuration/standalone.xml`) to the `tasktop-hub.properties` file. More details about translating configurations from `standalone.xml` to the properties file can be found here.

    b. Upgrading from versions *earlier* than 21.1:

        i. If any configuration was applied to `exclusion-paths` property in the `web.xml`, it needs to be migrated to the tasktop-hub.properties file. See the following example:

            1. Copy `/auth/realms/Tasktop/broker/saml` and `/auth/realms/Tasktop/login-actions` from the `web.xml` file.

```
a. <filter>
        <filter-name>CORSFilter</filter-name>
        <filter-class>com.tasktop.servlet.cors.
   CorsHeaderScrutinyServletFilter</filter-class>
        <init-param>
            <description>A comma or whitespace separated list of paths to
   exclude from the CORSFilter</description>
            <param-name>exclusion-paths</param-name>
            <param-value>
                /auth/realms/Tasktop/broker/saml
                /auth/realms/Tasktop/login-actions
            </param-value>
        </init-param>
   </filter>
```

            2. Place them in the `tasktop-hub.properties` file.

```
a. # A list of paths that will be excluded from the CORS verification.
   # This list is separated by comma. Example: /first-path,/second-path
   hub.security.cors.exclusionPaths=/auth/realms/Tasktop/broker/saml,/auth
   /realms/Tasktop/login-actions
```

    c. Upgrading from version 20.4 and later:

        i. No action needs to be taken. Tomcat/Keycloak configurations will be applied automatically.

6. If you have connected to the Microsoft TFS repository in the past:

    1. Remove all files and folders, except for the com.tasktop files, under `<install-location>\Tasktop\libraries\microsoft-tfs` and `<program-data>\Tasktop\libraries\microsoft-tfs`. Note that the parent folders (marked in red here) for each location could differ if they were customized during original installation.

2. Once Tasktop is started up again, navigate to the TFS repository connection screen. There, you will see instructions on how to provide the updated SDK and CLC files to Tasktop by adding them to the connector-requirements directory on the machine that hosts Tasktop.
3. Restart Tasktop after uploading the files.
7. Start Tasktop.
8. Navigate to the Activity screen.

   a. Review the Background Jobs tab to review status on Integration Data Migration jobs.
      i. Resolve any associated issues shown here. These issues will need to be resolved and their associated data migration jobs completed before the affected integrations can run (unrelated integrations should continue to process).
      ii. Once the associated issue is resolved, failed Integration Data Migration processes can be prioritized using the 'prioritize' button on the Background Jobs tab. Ensure that these jobs complete successfully.
   b. Review the Issues tab to resolve any configuration issues.
      i. If there are configuration migration issues, those will be shown in the Issues tab. They will block affected integrations from running (but unrelated integrations should continue to process). Once the source of the issue is resolved, configuration migration issues can be retried using the 'retry' button on the Issues tab.
      ii. If using TFS, you may see issues related to unsatisfied connector requirements since you may need to upload new versions of the TFS SDK and CLC zip files.
   c. Review the Errors tab to resolve any errors related to specific integration activities.
   d. Once all issues and errors are resolved, the internal upgrade will complete and information will begin processing for those affected integrations.
9. If you are upgrading from a version earlier than 19.4.1, please see details regarding the Troubleshooting User here.

Upgrade from Backup File

This feature is not applicable to Tasktop Cloud and is only available when upgrading from Tasktop Hub versions 20.1 and later. To utilize this feature, please see the section below.

To restore Hub to a previous version in cases where integrations were resumed individually during an upgrade, you must use the upgrade backup file available on the **Advanced Configuration** screen. The downloaded data in the file corresponds to artifacts that may have been modified when migrations were still running to ensure artifact updates aren't duplicated when restoring.

⚠ **Note**: You must download the backup file from your Hub instance **before** beginning the steps to restore.

To use this feature, you will first need to download the upgrade backup file. This file can be downloaded on the **Advanced Configuration** screen.
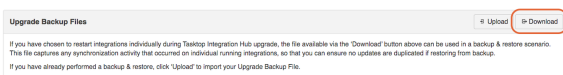
After clicking **Download**, the following message will appear:
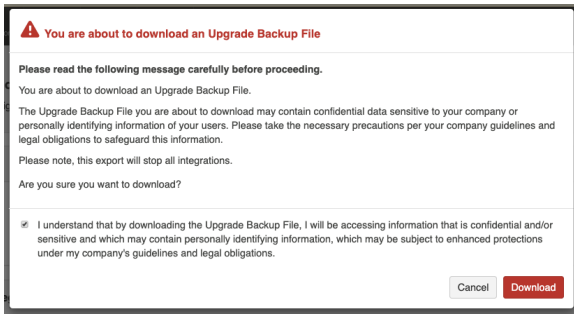
Once the file has been downloaded, you will need to restore Tasktop to the prior version. Please see the section above for more details on how to restore to a prior version.

After restoring to the earlier version, you can then select the backup file you would like to import and click **Upload**.
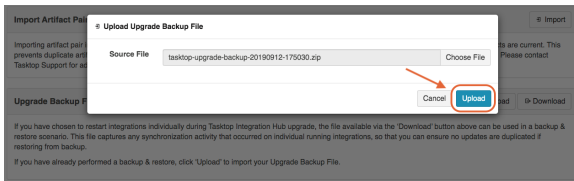
If the backup file is imported successfully, the following message will appear and your integrations will resume.
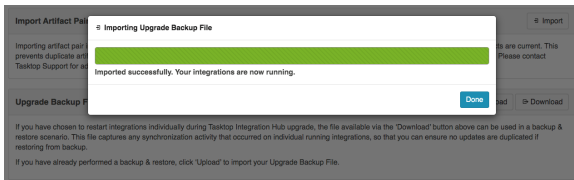
⚠ **Note**: If the backup file fails to upload, you will need to contact Tasktop support for further assistance.

Business Continuity

Overview

Tasktop Hub maintains information critical to organizational business processes, and therefore should be included in a comprehensive business continuity plan that safeguards data and ensures business continuity in hardware and operational failure scenarios.

💡 For additional information, please contact Tasktop Support or your Sales Rep to access our Business Continuity & Disaster Recovery materials.

Data Loss Prevention

An important aspect of disaster avoidance is avoidance of data loss. Tasktop Hub should be configured to use a reliable external database such as Oracle or Microsoft SQL Server. Please see the Supported Databases for storing Tasktop Operational Data section to determine supported databases.

External databases should be set up with sufficient redundancy to maximize uptime and to reduce the probability of data loss due to hardware failure. For details on how to set up your external database, please see our General (Settings) screen.

Monitoring

You can append `/api/health` to your Tasktop URL (e.g., https://server.tasktop.com/api/health) to get information on general health of your Tasktop instance (e.g., to confirm that Tasktop is not experiencing downtime or that your license is valid).

Customers may wish to leverage this API call into a monitoring tool to allow them to determine if a failover instance need be brought up in case of issues.

Below is a definition of what each term means:

- **notificationsettingsValid**
  - *Pass:* Testing the connection to the email server succeeded.
- **databaseisAvailable**
  - *Pass:* Connecting to the operational database succeeded.
  - *Fail:* Tasktop could not connect to the Operational Database; Tasktop cannot function until this is resolved.
- **licenseConfigured**

  - *Pass*: Tasktop has been configured with a license
- **configurationMigration**

- *Pass*: No errors from configuration migration are present, i.e., configuration migration completed successfully the last time it ran.
- **memoryUsage**

  - *Pass*: No "out of memory" errors are present.
- **passwordEncryptionInitialized**

  - *Pass*: No cryptography errors (error type CCRRTT-60001) exist (i.e., the Java runtime environment supports 256-bit AES encryption).
- **tasktopIsRunning**

  - *Pass*: Tasktop has initialized and is running, meaning the UI should be accessible. Tasktop is not currently restarting or shutting down.
  - *Fail*: Tasktop is initializing, restarting, or shutting down.
- **securityIsSetUp**

  - *Pass*: Tasktop has been configured with a master password, and the master password has been entered if necessary.
  - *Fail*: Either the master password has not yet been set up, needs to be re-entered, or Tasktop has been configured in insecure mode (no longer supported or possible to configure).
- **licenseExpired**

  - *Fail*: The license has expired.
- **licenseIsValid**

  - *Pass*: All configured integrations are allowed by the configured license.
  - *Fail*: There is no license, or there is an integration whose integration style is not licensed, or there is an integration using a connector that is not licensed.
- **activitiesCanProcess**

  - *Pass:* All valid integrations can detect changes and process activity.
  - *Fail:* Tasktop is not detecting changes or processing activity for any of the *valid Work Item or Work Item + Container integrations. Check for any error messages in Hub to resolve the issue.

    - ***Note**: Valid integrations are the ones not blocked by issues caused by user configuration errors or external factors (e.g., unavailable repositories).

Downtime

When Tasktop service is unavailable, changes may be taking place in integrated repositories. Normal Tasktop operation ensures that data flows between these repositories in a timely manner. When the server is unavailable, however, information is no longer propagating between integrated systems.

This has the following impacts:

1. Synchronization integrations will not create or update artifacts in synchronized repositories

2. Enterprise Data Stream integrations will not record artifact changes from their integrated source repositories to their target databases, which may cause a loss of fidelity in reporting data
3. Gateway integrations cannot accept payloads from integrated gateway collections; this can result in data loss if the integrated tools cannot handle the downtime

Upon restarting Tasktop Hub, integrations will resume with the following effects:

1. All Synchronization integrations will begin processing where they left off when the server became unavailable; there may be a backlog of changes to process, but no synchronizations will be lost
2. Enterprise Data Stream integrations will begin detecting artifact changes; any changes that occurred when service was unavailable will be detected, but multiple changes to the same field will have lost fidelity (only one change to that field will be reported)
3. Tasktop will begin accepting Gateway collection payloads, and if the integrated repositories are configured correctly to retry payloads, they will be processed as usual without data loss

Backup

A working backup strategy is a critical element of disaster recovery, since only backups can mitigate complete hardware failure and user error. A backup strategy that ensures correct and current backups is essential. Backups of the Tasktop database include both configuration and operational data.

See details on Backup procedures in the Upgrading section.

Restore

In order to restore Tasktop Hub, follow the instructions outlined in the Upgrading section.

High Availability

To learn more about Tasktop High Availability strategies, please reach out to Tasktop Support or your Sales Rep to access our Business Continuity & Disaster Recover materials.

Load Balancing

To learn more about Tasktop's recommendation for handling REST API traffic to a repository, see our FAQ page.

Troubleshooting

Overview

Tasktop provides several methods for troubleshooting your integration — from our easy-to-use Activity screen which outlines errors, past activity, and more to our Support and Usage Reports which can be used to troubleshoot issues with our support team and to help track Tasktop usage.

Our Configuration History screen contains up to **six months** of changes that have been made on your general settings or configuration elements (e.g., integrations, models, collections, mappings, etc). For information on how to migrate these changes from one Tasktop instance to another, please see the Configuration History page.

Configuration History

Introduction

The Configuration History screen shows you up to **six months** of changes that have been made on your general settings or configuration elements (e.g., integrations, models, collections, mappings, etc).

To view your **Configuration History**, click the History icon in the upper right corner.
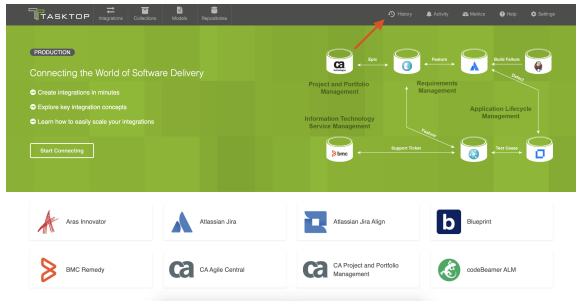
On this screen, you can see detailed information about the configuration changes and filter these changes by name, date, and the user who initiated the change.
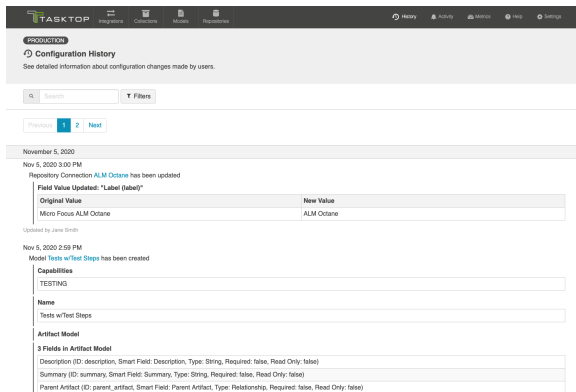
💡 **Tip:** To easily view an updated configuration element, just click on the hyperlink on the element.

Migrating Configuration Changes

This feature allows you to export configuration changes and migrate them to another Tasktop instance. For example, during testing or major upgrades, changes made in a test environment may need to be replicated in the production environment. Manually replicating these changes is often tedious and time-consuming.

With this feature, you can easily move configuration changes from one instance to another in just a few clicks.

Before you Begin

Before using this feature, please review the following requirements:

1. The source environment version should match the target environment version.
2. The source configuration element should mirror the target configuration element as closely as possible.

   a. 💡 **Note**: Changes are applied based on the names/labels of configuration elements in Hub. For example, if you export a change that adds Artifact Filtering to an integration named **Jira to ServiceNow**, upon import Tasktop will search for an integration named **Jira to ServiceNow** and apply the change.

Exporting Changes

To begin exporting configuration changes, click **Export Changes.**
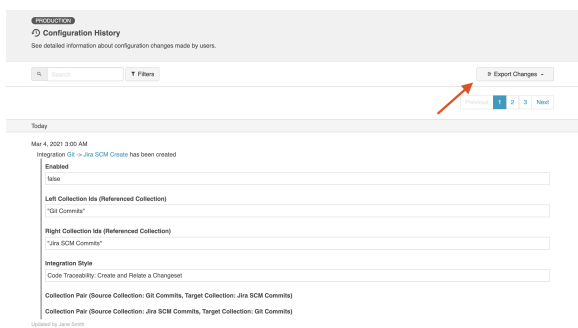
Click **Select to Export** to export selected changes or **Export All** to export all configuration changes.

💡 **Note**: Any changes made on the **Settings** screen cannot be exported (e.g., adding a license or updating the change detection interval). Additionally, any credentials will not be exported and will need to be re-entered upon import.
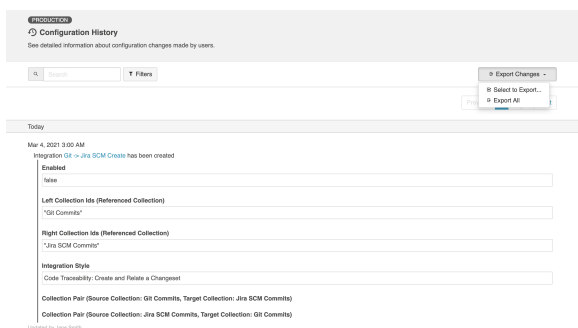
If you'd like to narrow down changes, you can filter by type, date range, and user.
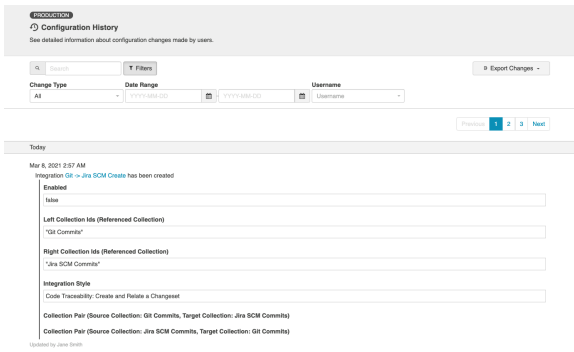
After you've selected the desired changes, click **Export Changes** and a **.zip** file will be generated with the changes.

💡 **Note**: If you attempt to modify the contents of the **.zip** file, you may encounter issues upon import.
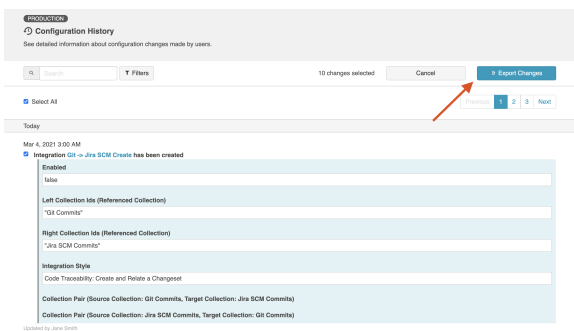
Importing Changes

You can import the exported changes in the **Advanced** tab on the **Settings** screen.
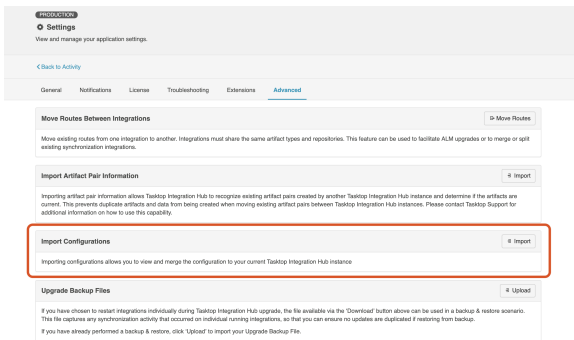
Click **Import** in the **Import Configurations** section.
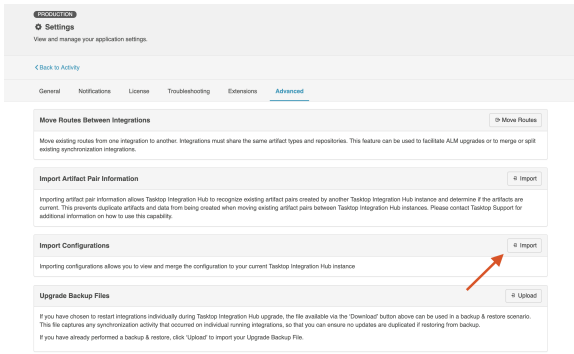
On the **Import Configurations** screen, select the **.zip** file with the changes you'd like to import.

Click **Next** to review the changes before importing.
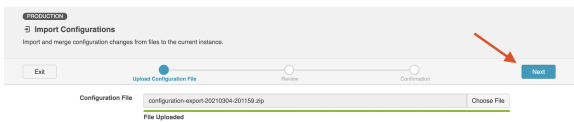
After reviewing, click **Import** to apply the changes.
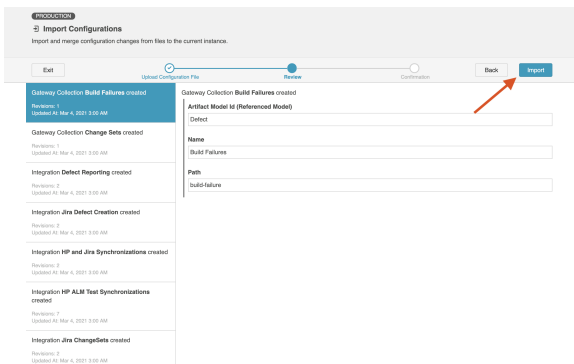
If a change fails to import, an error message will appear to explain why the change failed. Once a change fails to import, Hub will not attempt to import the subsequent changes and will provide a new **.zip** file containing the remaining changes to import after troubleshooting the failed change.

Possible causes of import failure:

- **Missing element in the target Hub instance**. This error can be resolved by manually creating the missing required element in the target system or importing the changes that created that element from the source Hub instance to the target Hub instance.
- **Element already exists in the target Hub instance**. This error can be resolved by deleting the element, or not exporting the changes that try to create the duplicate element.
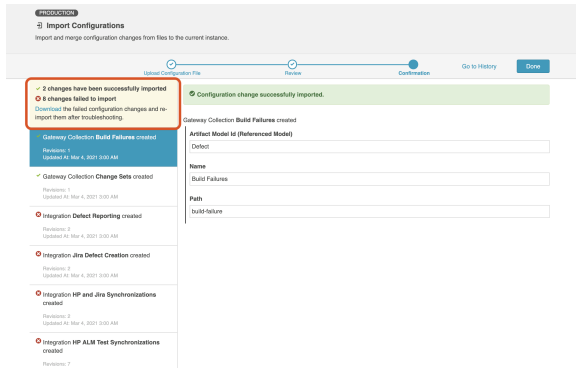
Once you're finished importing configuration changes, click **Done**.
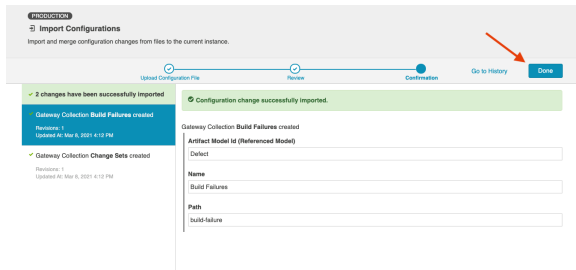
The imported changes will be visible on the **Configuration History** screen.
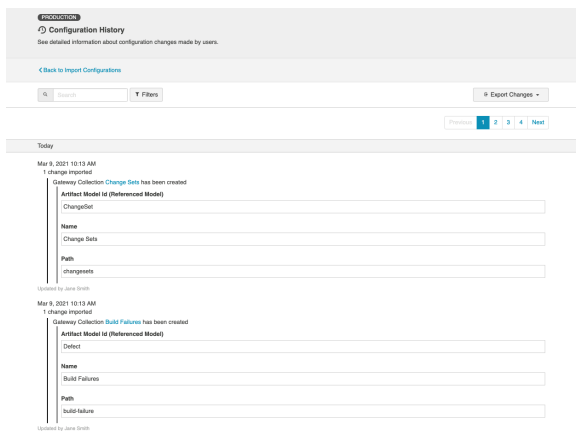
Activity Screen

Introduction

Most problems can be solved by navigating to the **Activity** screen and following the steps described on the listed errors. The Activity screen can be accessed by clicking **Activity** in the top right corner of the screen.
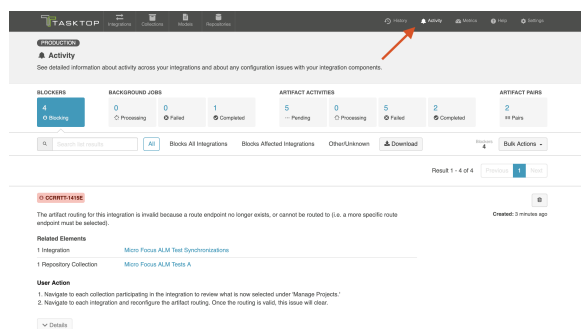
Blockers

The **Blockers** tab shows issues that arise from **invalid Tasktop configuration** or from more **global issues,** such as having an **invalid or expired license**. These are issues that can generally be resolved within the Tasktop application itself.

💡 **Tip**: Blockers can block integrations from running, so it is recommended that you monitor the Blockers tab regularly.
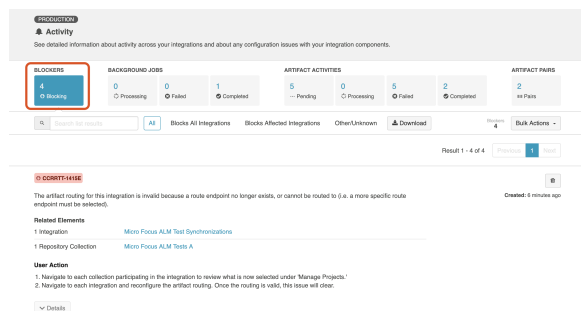
An additional warning icon appears when blockers are so fundamental that they will prevent integrations from running.

The hover message will indicate whether the blocker will prevent **all** integrations from running (e.g., licensing errors), or just **affected** integrations from running (e.g., a configuration error that impacts just one integration).
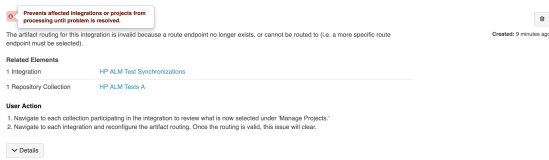
You can filter based on blocker impact using the filters at the top of the screen.
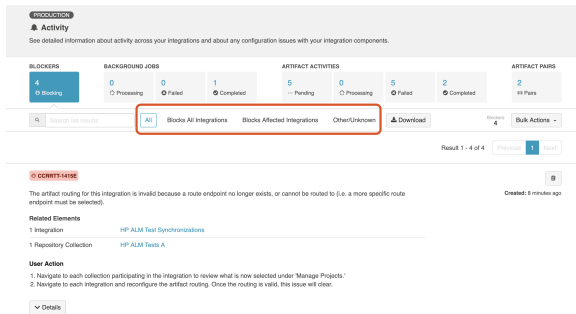
Or, you can use the search box to refine your results.
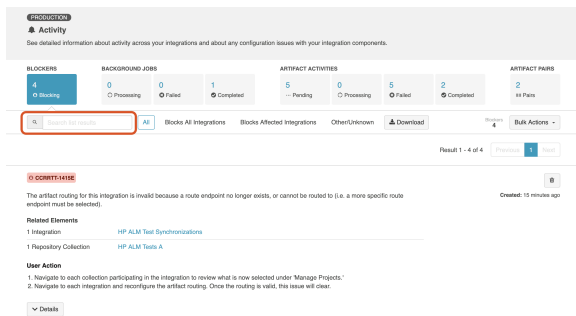
Click **Download** to export a **.csv** file containing your Blockers.
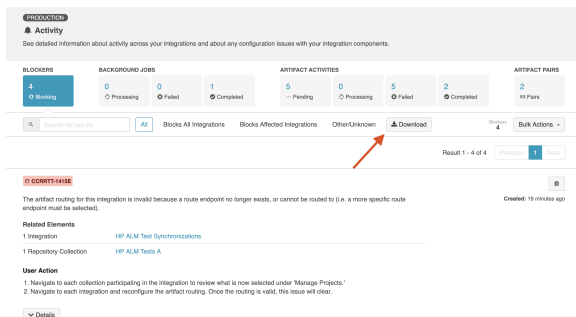
You can take the following actions on the Blockers tab:

- ↻ **Retry**: Retries a blocker. This action is only available for configuration migration blockers.
- **Resolve:** Resolves a blocker. This action is only available for certain blocker types, and can be taken to acknowledge that the user has reviewed the blocker and taken any required user actions.

- 🗑 **Remove**: Removes a blocker. If the blocker is blocking an integration, the integration will become unblocked. However, if the cause of the blocker has not been resolved, the blocker will return to the Blockers tab the next time configuration validation occurs (once an hour).

You can also take the following Bulk Actions:

- **Refresh**: Refreshes the blockers tab.
- **Remove All**: Removes all blockers. If the blockers were blocking an integration, the integration will become unblocked. However, if the cause of the blocker has not been resolved, the blocker will return to the Blockers tab the next time configuration validation occurs (once an hour).

Background Jobs

The **Background Jobs** section shows progress on background Tasktop processes such as: Upgrades, Redeployments from Sync, Project Replacements for invalid projects in Tasktop collections, and Moving Routes between Integrations.

Background Jobs consists of three subcategories:

- **Processing**: Background Jobs that are currently processing
- **Failed**: Background Jobs that Tasktop attempted to process, but were not successful
- **Completed**: Background Jobs that have successfully completed

You can take different actions on the background jobs in these subcategories, which are outlined in the sections below.
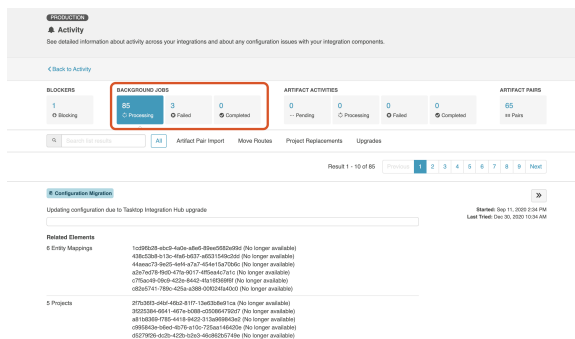
You can filter based on job type for each category.
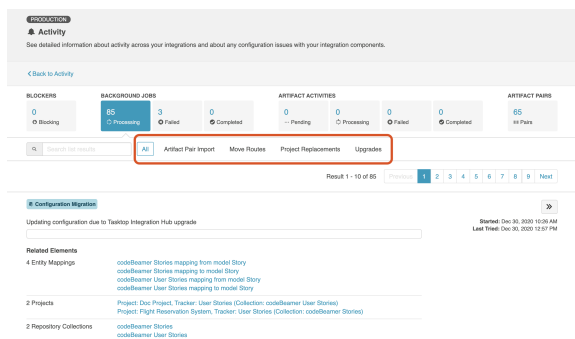
Or, you can use the search box to refine your results.
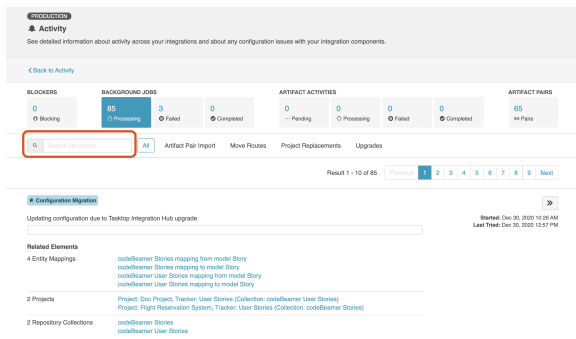
Processing

In the **Processing** tab, you can take the following actions:

- **»Prioritize**: Prioritizes the processing background job in the queue

While background jobs are processing, you will see a progress bar to track progress.

💡 **Note**: Jobs that are in progress cannot be canceled.
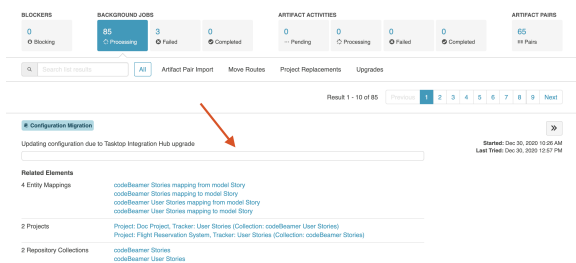
Failed

In the **Failed** tab, you can take the following actions:

- **»Prioritize**: Prioritizes the retry of the background job in the queue. This option is especially useful if you have made changes in your repository or in Tasktop that will likely clear up the failed job.
  - You will see this action if the event is already set to be retried, and is hence both in **failed** and **processing** states simultaneously.

If a background job fails, it will appear color coded in red. If there is an associated issue, a link will be shown to navigate to that issue. These jobs will be retried automatically until they complete, and can be prioritized using the **prioritize** button.
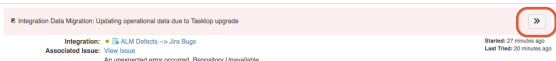
Completed

In the **Completed** tab, you can take the following actions:

- ✖ **Remove Completed**: Removes all completed background jobs.

Once jobs complete, you will see them in the **Completed** tab of the Background Jobs section color coded in green. For Project Replacement jobs, you can expand the **Projects Updated** section to see additional details:

You can remove all completed background jobs using the Bulk Actions dropdown.

💡 **Tip**: Activity listed on the Background Jobs tab will be cleared after each Tasktop upgrade.
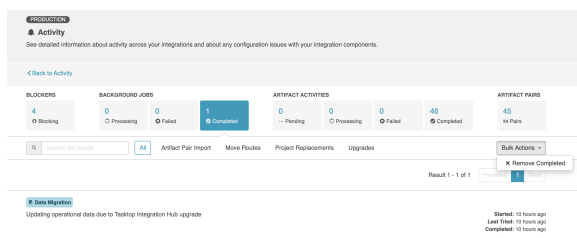
Artifact Activities

The **Artifact Activities** section shows **activities that are active in an integration.**

Artifact Activities consist of four subcategories:

- **Pending**: Activities that are queued up to be processed.
- **Processing**: Activities that are currently processing.
- **Failed**: Activities that Tasktop tried to process, but was not successful.
- **Completed**: Activities that have completed processing.

You can take different actions on the events in these subcategories, which are outlined in the sections below.
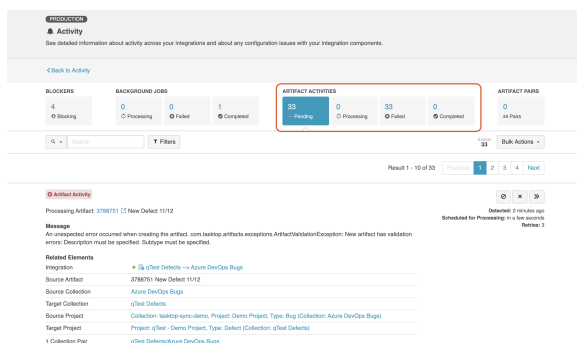
Each category allows you to expand your filter options.
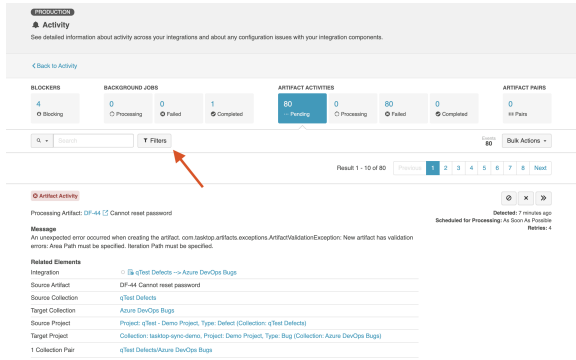
You can filter by search, integration status (e.g., running or stopped), integration name, or created date.
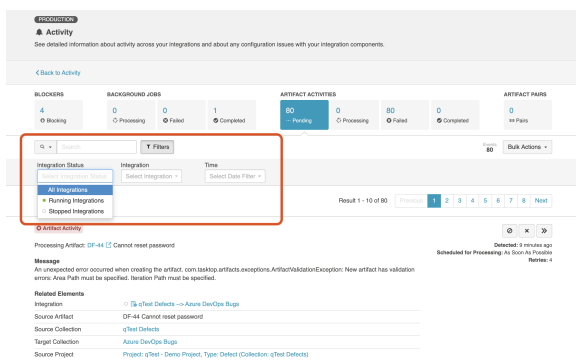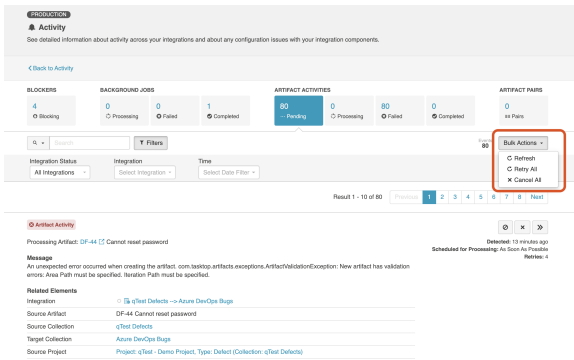

You can also filter to exclude specific text.

To do this, click the search icon and choose **Exclusive** in the dropdown menu. Specify the text you would like to exclude, and only artifact activities without this text will be displayed.
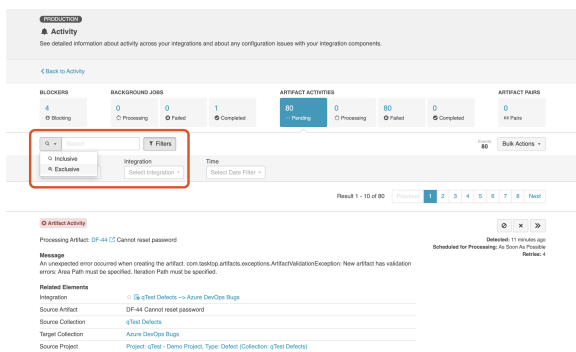
Each category also allows you to take bulk actions.

The number of events in the summary banner will update regularly, but the list of events themselves will need to be refreshed to show new activity. This is to avoid items unexpectedly appearing and disappearing when you might be examining them.
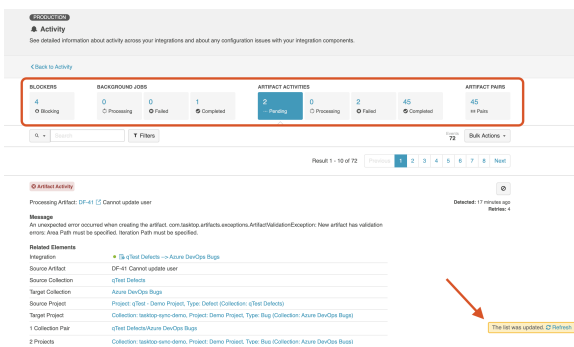
Pending

On **Pending** Activity, you can take the following actions:

- **» Prioritize**: Prioritize this pending event in the queue.
- **✕ Cancel**: Remove this event from the pending queue. It will not be processed, though subsequent changes to artifacts will trigger another event.
- **⊘ Ignore**: If an error is pending, you have the option of moving it to the Ignored Errors tab. See Errors section for details.

Processing

The **Processing** tab shows activity that is currently processing. There are no actions that can be taken here.

Failed

The **Failed** tab shows any failed activity related to specific activities that have occurred. In contrast to the Blockers tab, failed activity here typically blocks **individual artifacts** rather than **entire integrations**, and therefore are less severe.

You can take the following actions:

- ⊘ **Ignore**: Moves the failed activity to the **Ignored** list. Once ignored, it will no longer show up in the Failed list (or in Pending), and it will not be counted in the Failed summary counts at the top of the screen.
- ✕ **Cancel**: Removes the failed activity from the list. It will not be retried, though subsequent changes to artifacts will trigger another event.
- » **Prioritize**: Prioritizes the retry of this failed activity in the queue. This option is especially useful if you have made changes in your repository or in Tasktop that will likely clear up the error.
    - You will see this action if the event is already set to be retried, and is hence both in **failed** and **pending** states simultaneously.
- ╋ **Recreate**: If a previously synchronized artifact has been deleted in one of your repositories, you have the option of recreating it from the Activity screen. This will keep the newly recreated artifact in sync with the source artifact.
- ↻ **Retry**: Retries the failed activity.
    - You will see this action if the event is not already set to be retried.

💡 **Note**: Most failed activity will automatically be retried on a gradually decreasing interval (granted that Tasktop can locate the artifact that is to be changed). Retry-able failed activity will be retried approximately 30 seconds after they are first encountered, and then on a gradually decreasing interval over time.

You can see information about retries on the failed activity itself. In the example below, you can see that the failed activity has been retried 6 times, and that it has been scheduled for processing as soon as possible. If a failed activity will not be retried, this information will not be be relevant and hence will not be displayed.
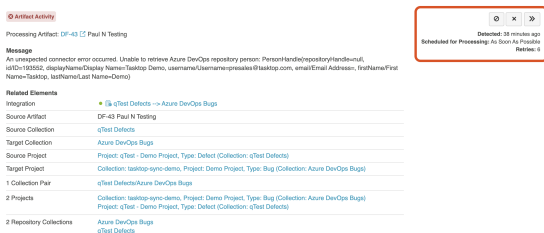
Click **Download** to download a **.csv** file containing your failed artifact activities.

A complete list of failed artifact activities (i.e., errors) is available in the appendix.

You can find additional information on select errors in our FAQ.

Ignored

If you ignore a failed activity, it will be moved to the **Ignored** list, and no longer be counted in the **Failed** total at the top of the screen.

💡 **Note**: Ignored artifact activities must be manually retried to be resolved.

You can move a failed activity back to the Failed list by clicking **Stop Ignoring**.

If you enable **Auto retry activities below**, all ignored artifact activities will be retried automatically.
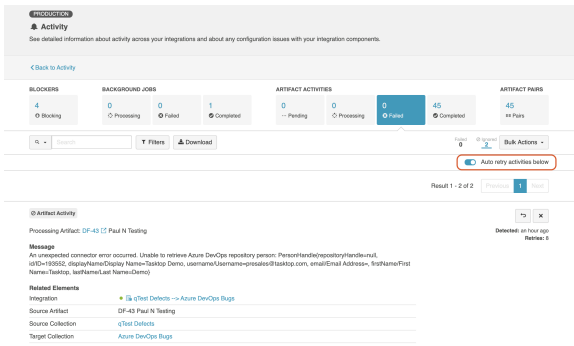
If you'd like to use the bulk action, **Stop Ignoring All**, you must first apply a filter to the Ignored list. This will move all failed activities that meet your search filters back to the Failed list.
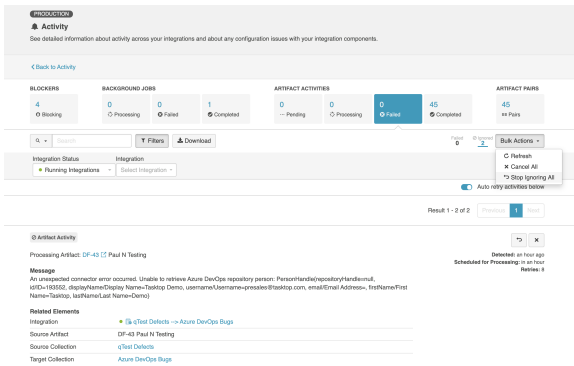
Completed

The **Completed** tab allows you to view all past integration activity, so that you can understand **what has successfully completed**.

There are three types of Completed activities:

- **Created Artifact**: When a new target artifact is created in a repository
- **Updated Artifact:** When an existing artifact is updated in a repository
- **Associated Artifacts**: When existing artifacts are auto-matched, and therefore associated with one another. Currently this is only supported for containers, when utilizing Container Matching fo r a Work Item + Container Mirroring synchronization integration.
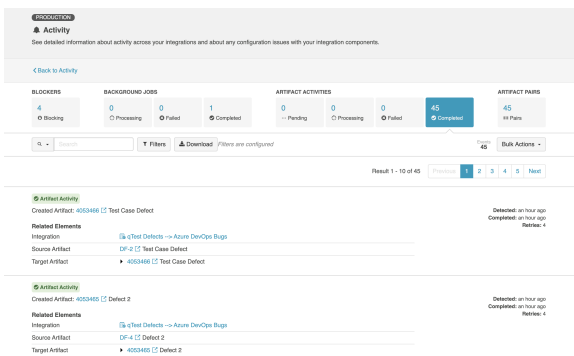
You can click the drop down arrow on each activity to see more details on the activity that has occurred.

If past activity is indicating that a new artifact was created, you'll see that the Original Values listed are blank, and that the Activity type is **Created Artifact** as opposed to **Updated Artifact**.
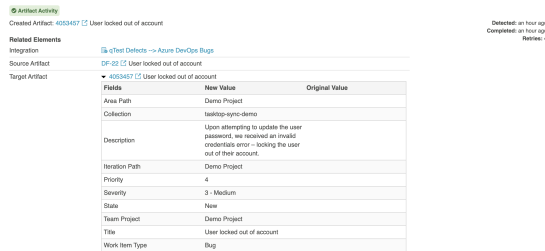
If you'd like to filter your results, you can use the search box.

Additionally, you can click **Filters** to expand filtering options. You can use the integration filter to filter by integration, or the date filter to filter by a fixed date range or by a set number of days in the past (which will dynamically update your results as days pass).
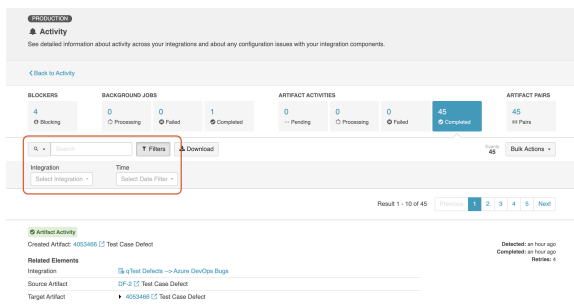
You can use **Bulk Actions** to refresh, or remove all artifact activities that meet your filters. If you have not configured any filters, all completed artifact activities will be refreshed or removed.
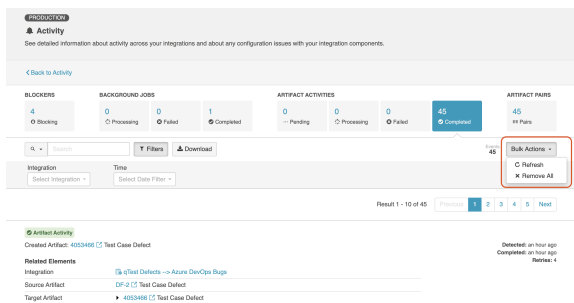
Click **Download** to download a **.csv** file containing your completed artifact activities.
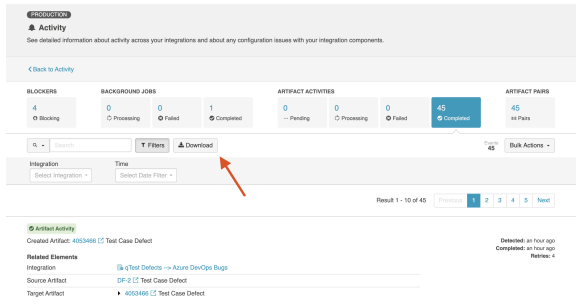
⚠️ **Note**: Tasktop will store up to 100,000 entries on the Completed screen. Once 100,000 entries are met, older entries will be deleted as new entries come in. You can also opt to clear your entries when approaching 100,000 to have better visibility into more recent completed artifact activities.

Artifact Pairs

⚠️ If using a Keycloak installation, please follow the instructions here to enable full functionality of this feature.

The **Artifact Pairs** tab allows you to view and manage artifact associations so you can promptly address problems related to specific artifact pairs.

💡 **Note**: For each artifact that is displayed, there is no directional information involved with the pair (i. e., Tasktop does not display which artifact is source or target) — this tab only shows the artifact association.
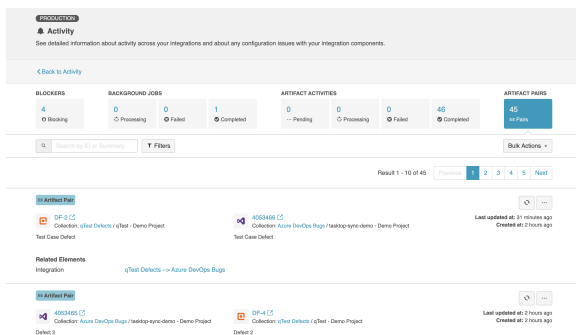
Synchronizing Artifact Pairs

Rather than synchronizing the full collection of artifacts in an integration, Tasktop allows you to synchronize a **single artifact pair**.

Click **Sync now** to synchronize an artifact pair.

💡 **Tip**: If you'd like to synchronize **multiple** artifact pairs, refer to the section below.
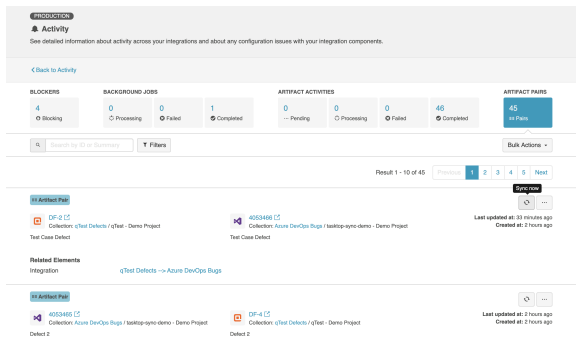
A pop-up will appear prompting you to select the artifact from which you'd like to initiate the synchronization.
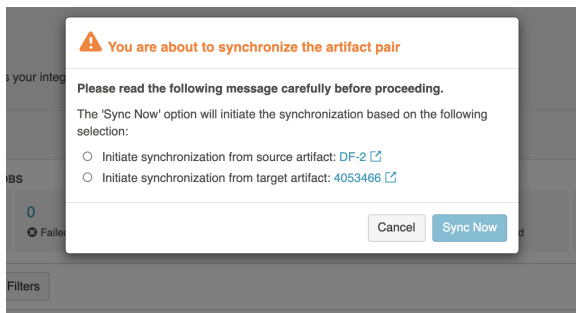
Viewing Artifact Pair Details

> ⚠️ If using a Keycloak installation, you **must** follow the instructions here before you can proceed.

If you'd like to view the details of an artifact pair, click the ellipses and select **View Detail**s.

💡 **Tip**: The View Details option may not appear if there is missing information on either side of the artifact pair (e.g., if a collection or repository is deleted on either side of the pair).

⚠️ **Note**: The artifact summary may not appear if using a custom string field as the summary field.
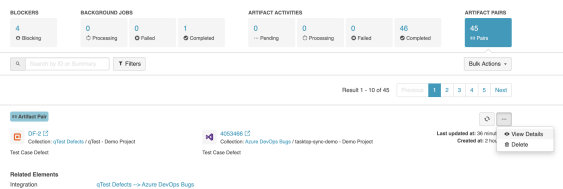
A pop-up will appear with a table of field values belonging to the artifact pair. Any greyed out area represents the source field in a unidirectional mapping.

💡 **Note**: Each row in the table corresponds to a field mapping. For more complex mappings such as one-to-many, each field mapping will remain in a single row.
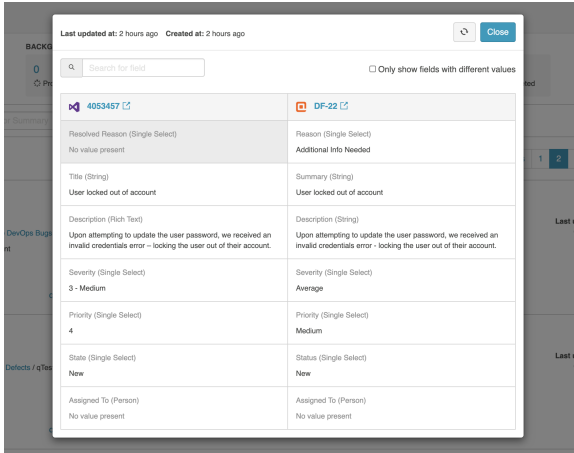
Within the pop-up, you can initiate synchronization of the pair by clicking **Sync Now**.
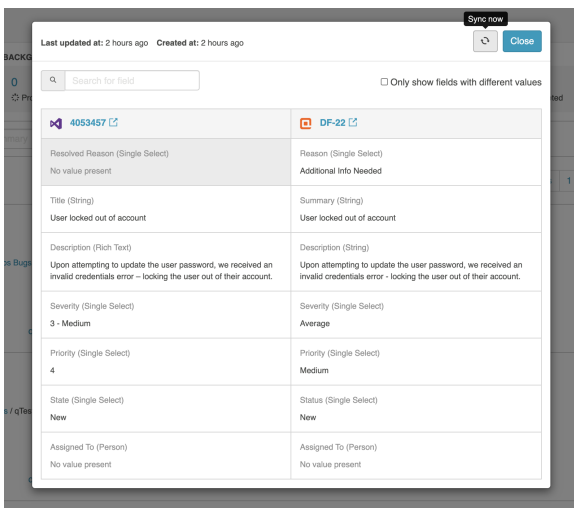
If you'd like to find a field and many details exist, you can use the search box in the upper left corner to find a specific field.

💡 **Note**: The search option within the pop-up will only search for fields and **not** values.
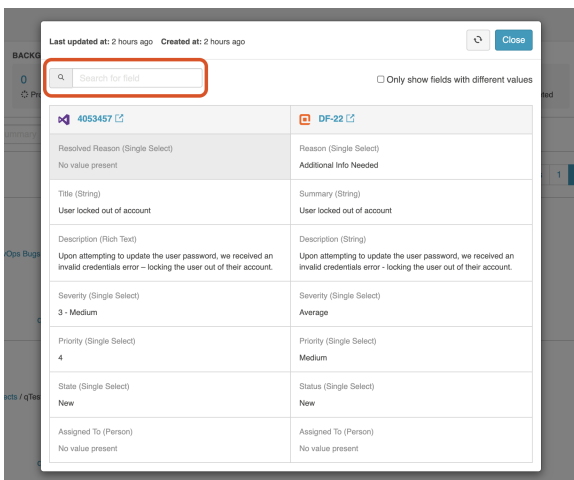
Deleting Artifact Pairs

⚠️ If using a Keycloak installation you **must** follow the instructions here before you can proceed.

If you'd like to delete an artifact pair, click the ellipses located in the right corner of the artifact pair and select **Delete**.

💡 **Tip**: To delete multiple artifact pairs, refer to the section below.
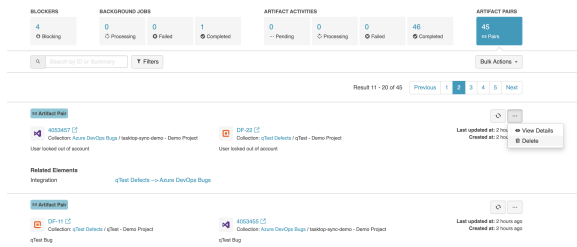
A pop-up will appear confirming you'd like to proceed.

⚠️ **Note**: Artifact pairs will be stored indefinitely unless deleted. Deleting an artifact pair will remove all associations from the database, which may result in duplicate artifacts if change detection picks up the deleted artifact again.
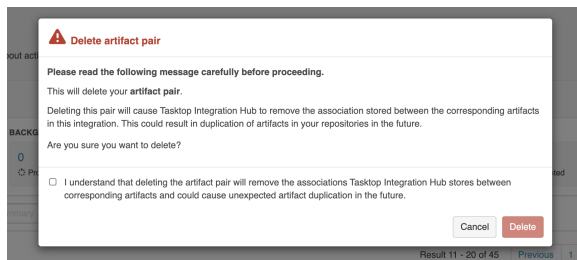
💡 **Note**: Deleted pairs are logged under a file named `deleted-pairs.log` in the Support and Usage Reports . If you're unsure if an artifact pair has been deleted, you can revisit the log file to confirm.

Filtering Artifact Pairs

If you'd like to filter your artifact pairs, you can use the search box to refine your results by ID or summary. Additionally, you can expand your filter options by clicking **Filters**.
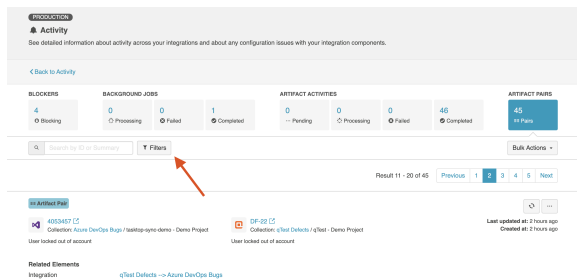
Using expanded filters, you can filter by integration.
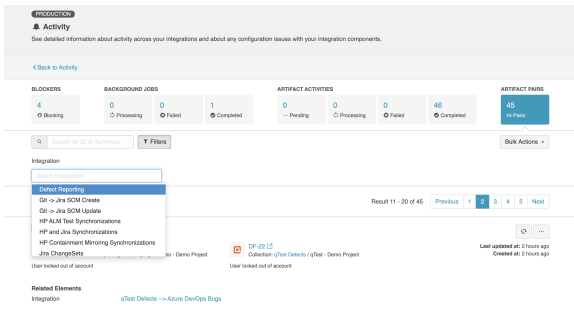
Bulk Actions

You can also take bulk actions like synchronizing or deleting multiple artifact pairs.

⚠ **Note**: If too many artifacts are created, this may cause many artifact events to be created — resulting in an influx of pending events that may delay subsequent processing. Please proceed with caution.

Click **Bulk Actions** and then click ✅ **Select ...** to select the artifact pairs you'd like to synchronize or delete.
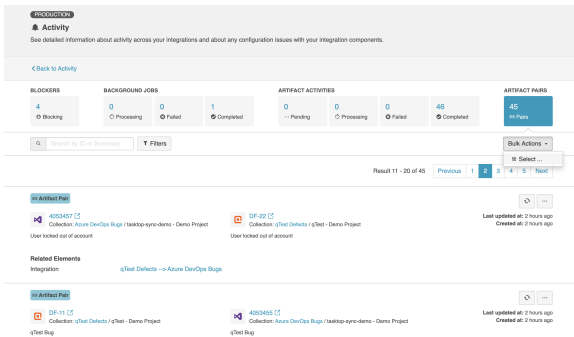
Once the pairs are selected, the option to synchronize or delete selected pairs will be enabled.

⚠ **Note**: Conflicts may occur as Tasktop will generate events for both artifacts within the pair. Please refer to the conflict resolution strategy configured within your integrations.
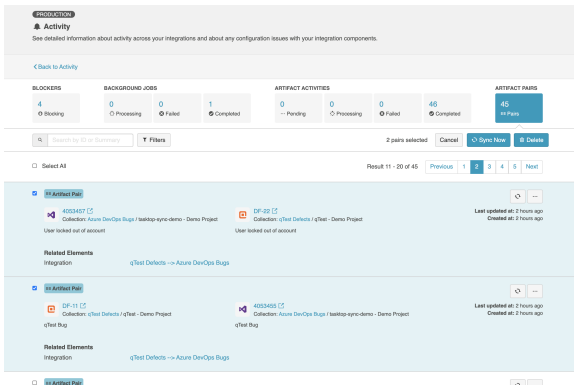
If you'd like to use the bulk option **Sync All Now**, you must first apply a filter to the Artifact Pairs list. This will move all artifact pairs that meet your criteria back to the Artifact Pairs list where you can bulk synchronize them.

To do this, click **Sync All Now**.

💡 **Note**: This option will **only** appear when filters are configured or search text is entered.
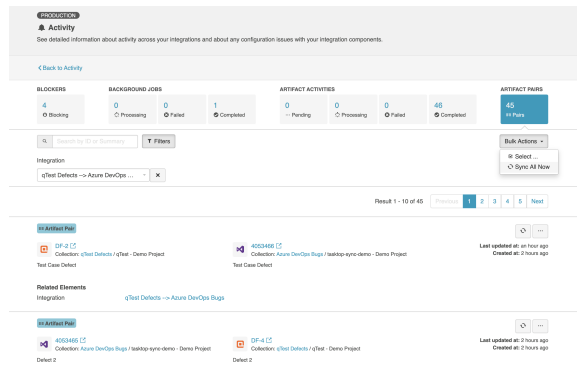
A pop-up will appear confirming you'd like to synchronize all selected pairs. Click **Sync Now** to synchronize the artifact pairs.
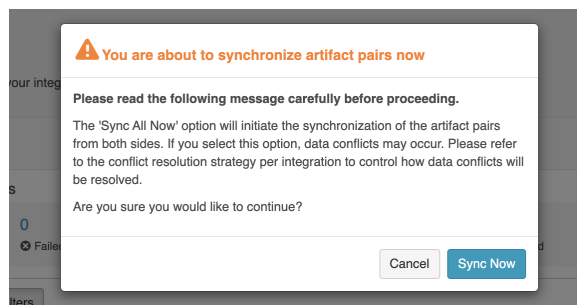
Exporting Artifact Pairs

This functionality should only be used under the guidance of Tasktop support.

You can also migrate artifact pairs from an On-prem instance to a cloud instance using the export functionality.

Before you Begin

Before using this feature, please ensure that the following requirements are met:

1. To use this feature, you must be enabled for access by Tasktop support.
2. The On-prem instance should be upgraded to the latest (major) version that matches the Cloud instance.

   💡 **Note**: This ensures that the artifact handles from the On-prem instance are compatible with the Cloud instance.

3. The Cloud instance should have the same repositories, integrations, collections, and field mappings configured.
4. If using a Keycloak version of Hub, the user logged in must be a **TasktopAdmin** user.
5. You **must** be on an On-Premise instance (export **cannot** be enabled <u>from</u> a Cloud instance).
6. Before exporting, it is recommended to <u>stop</u> all other integrations as you may encounter performance issues that could impact running integrations and the export.

To begin exporting your artifact pairs, navigate to the **Artifact Pairs** tab on the **Activity Screen**.
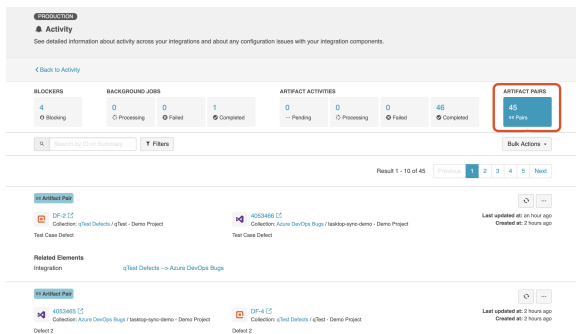
Click the **Export** option under the **Bulk Actions** dropdown.
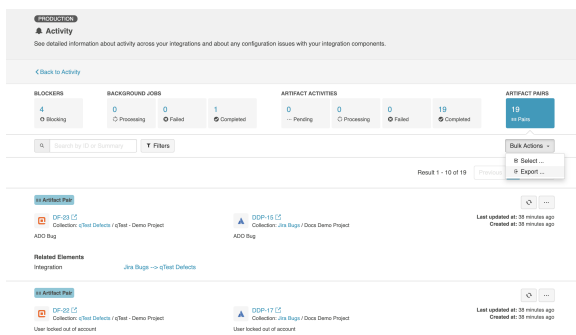
A pop-up will appear where you can select the integrations from which you'd like to export artifact pairs.

💡 **Note:** Only artifact pairs from work-item synchronization integrations can be exported.
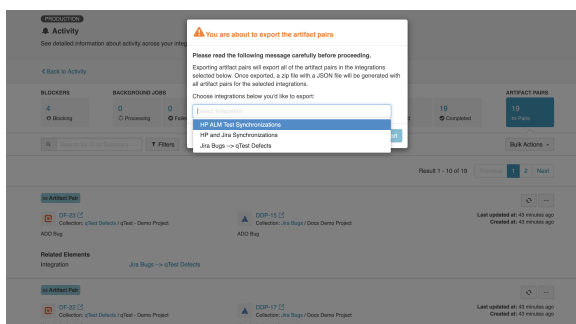
After selecting the integration you'd like to export, click **Export** and a **.zip** file will be generated with a **.js on** file containing all the artifact pairs.

During Export

Upon export, the selected integration will be disabled and no artifacts will be processed.

If the integration is in the middle of processing artifact events, the export will not start until the artifact events have finished processing. If the event cannot finish processing within the specified time limit, an error will appear alerting you to manually disable the integration before exporting the integration.

💡 **Note:** Only **one import** and **one export** can occur simultaneously.

If you encounter any issues upon exporting your artifact pairs, please reach out to Tasktop support.

Importing Artifact Pairs

⚠️ **Note**: If you do not **stop** the exported integration *before* importing into the cloud instance, you may encounter **duplicate** artifacts.

After extracting the exported **.zip** file, you can import the **.json** file in the **Advanced** tab on the **Settings** screen.
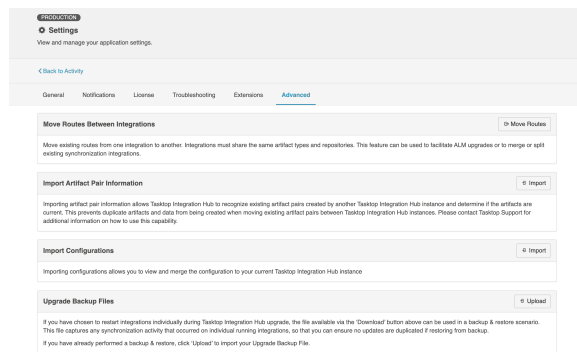
Click **Import** in the **Import Artifact Pair Information** section.
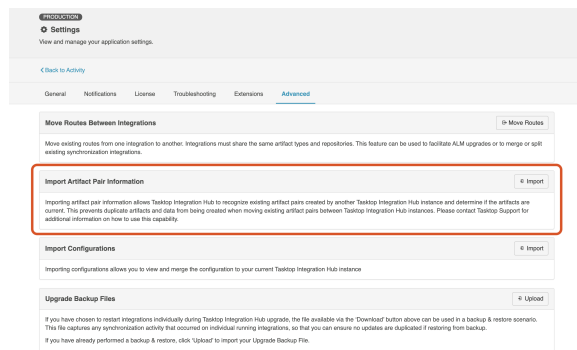
A pop-up will appear where you can select the target integration and the exported **.json** file containing all the artifact pairs.
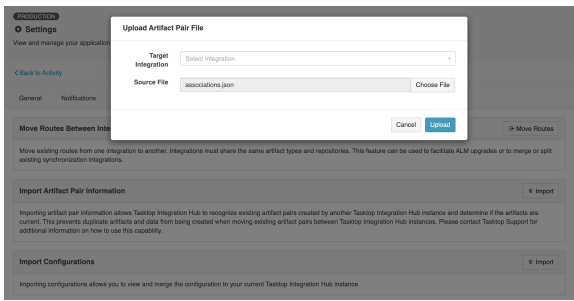
After you've selected the target integration and source file, click **Upload** to import your artifact pairs.
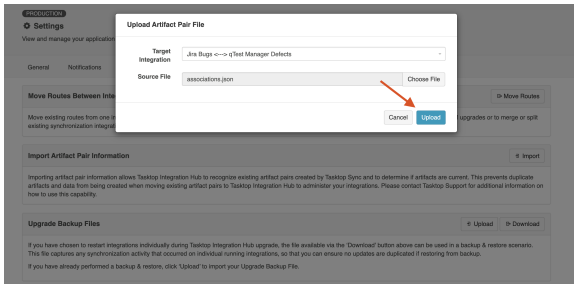
If you encounter any issues upon importing your artifact pairs, please reach out to Tasktop support.

Specific Error Messages

Errors on Activity Screen

You can find details on some specific error messages in our FAQ (in the Troubleshooting section) and in our connector pages (for connector-specific errors). We've also outlined errors below which require specific steps in the Tasktop UI.

Repository collection project cannot be found
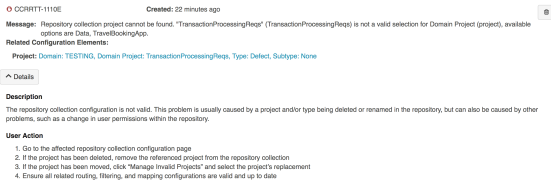
This error message is usually caused by a project type being deleted or renamed in the repository, but can also be caused by other problems, such as a change in user permissions within the repository, or moving the project to a new domain within that repository.

To resolve this error, go to the Collection configuration screen. Here, you will see a message alerting you to the fact that previously selected project(s) cannot be found in the repository.

💡 Note: You may not see the alert message on the Collections screen until Tasktop's cache refresh occurs. To 'force' the message to appear, click 'Manage Projects' and then refresh the project schema. This will cause the alert to appear.
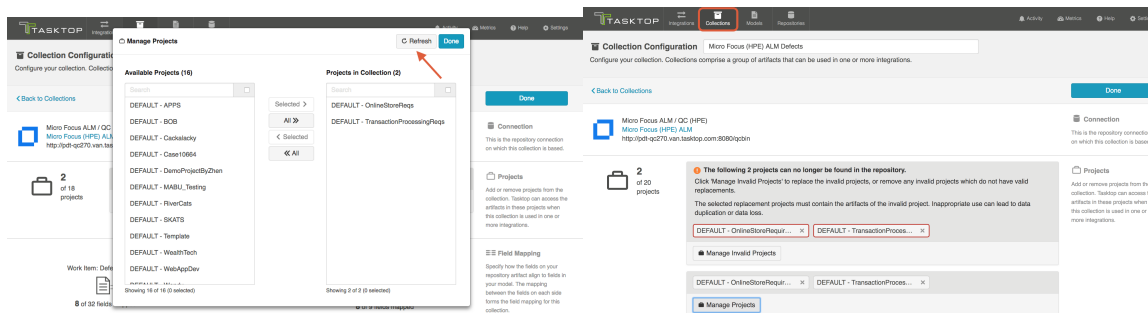
You can click the 'x' to remove any projects which do not have valid replacements, or click the 'Manage Invalid Projects' button to select replacement projects.

💡 Note: If you remove a invalid project (instead of replacing it via the 'Manage Invalid Projects' button) and then add its replacement to the collection later, you risk creating duplicate artifacts. Project replacements should always be executed via the 'Manage Invalid Projects' button, and all project replacements should be done at the same time.
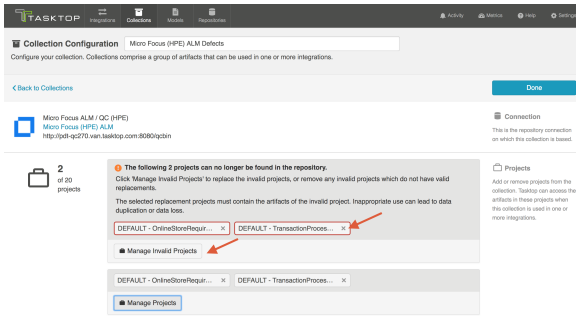
After clicking 'Manage Invalid Projects,' you will see the 'Manage Invalid Projects' picker, where you can search for available project replacements:
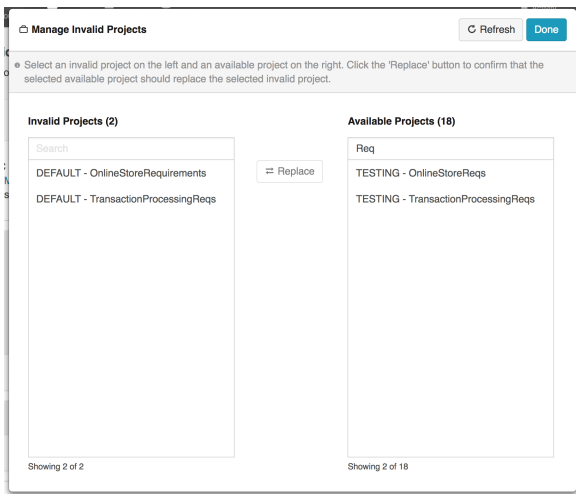
Highlight the invalid project on the left, and its replacement project on the right. Then click 'Replace.' Repeat the steps for any invalid projects you'd like to replace, and then click 'Done.'
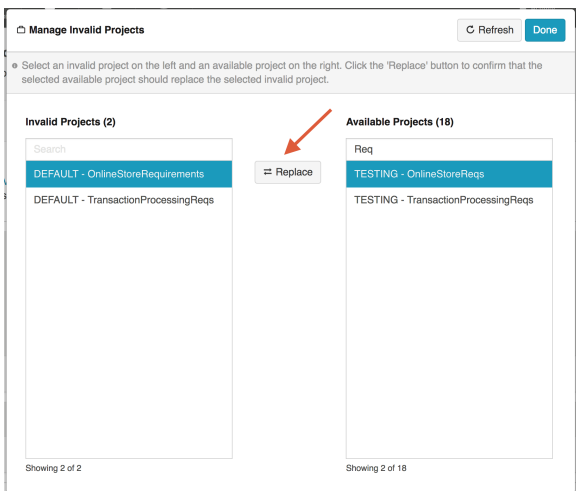
You will be prompted to save your collection in order to apply the updates (note that until the collection is saved, the invalid project names may display).
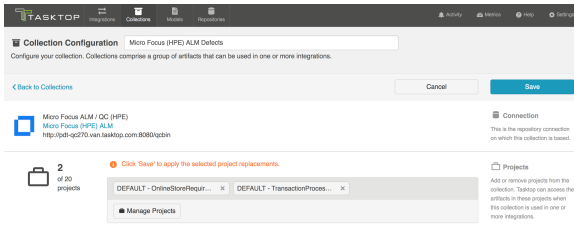
You will get a pop-up message warning you that the integrations associated with this collection cannot be used until the project update is complete:
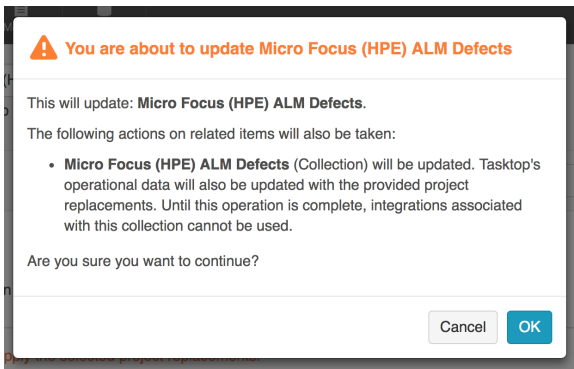
You can view progress for your project replacements on the Background Jobs tab of the Activity screen.

In-Application Errors

There are some scenarios where you may see an error message within the application itself, rather than on the Activity Screen.

External Database Error

If you have exported your Tasktop configuration information to an external database (see information here), and your database is not reachable, you will notice that your configuration elements (i.e. repositories, collections, integrations, etc.) will not be visible, and an error message will appear. To resolve this error, please ensure that your external configuration database is online.
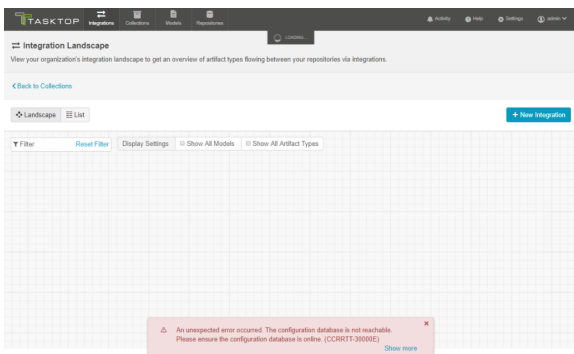
Support and Usage Reports

Overview

In cases where the Activity screen is not enough to resolve a problem, a Support and Usage Report is available to provide additional information.

The Support and Usage Report can be downloaded from the **Help** screen.

To download, click the **Download Support and Usage Report** link in the **System Information** section on the Help screen.
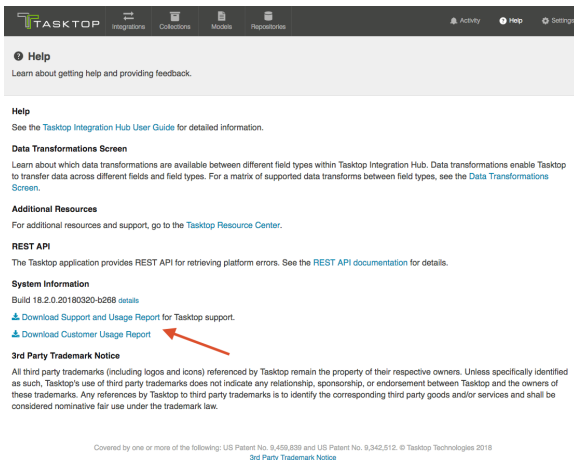
Upon downloading, you can specify how many days of log files you'd like to include in the Support and Usage Report.

💡 **Note**: The default value for this field is set to 30 days.
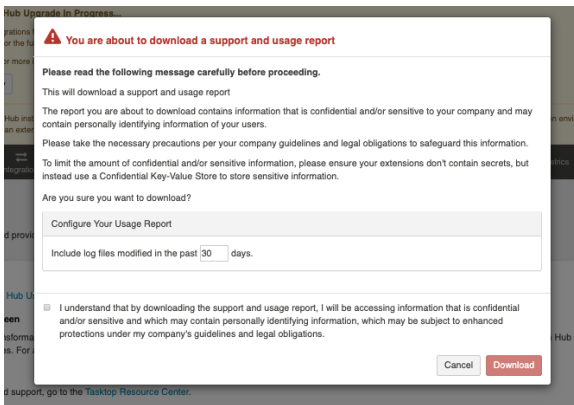
Report Contents

The downloaded report file is named tasktop-state-EnvironmentName-DATE-TIME.zip. Once unzipped, there will be five folders. The folders and contents are listed below.

💡 **Note**: The environment name will only appear in the file name if set in your Tasktop instance. Only the first 80 alphanumeric, underscore, or dash characters will be included in the environment name and all spaces will be converted to dashes.

1. activity
   - issues.json
2. configuration
   - configuration.json
   - hub-details.json
3. crash-reports
   - hs_err_pid*.log
4. logs
   - logs by day for past 14 days
   - configuration-changes.log
   - extensions.log
   - thread-dump.log
   - localhost.log
   - localhost_access_log.txt
   - catalina.log
   - tasktop-service.log
   - keycloak-service.log
   - keycloak-stderr.log
   - keycloak-stdout.log
5. mappings
   - text file for each collection configured
6. metrics
   - metrics.json
7. repository metadata
   - file for each repository connection configured
8. schemas
   - JSON file for each collection configured
9. usage
   - usage report
   - overview.json

Folder	File Name	Contents
activity	issues.json	Contains issues shown on the Activity screen.
config uration	configurati on.json	Contains all the configuration of your application instance.

config uration	hub-details. json	Contains details about the specific build and license of the application.
crash-reports	hs_err_pid *.log	Contains log files generated when the Java Virtual Machine crashes.
logs	logs	A separate file is created for every day of logs — 14 days of logs are saved.
logs	configurati on-changes. log	Contains details on configuration changes made in Tasktop Integration Hub, broken out by user (if applicable) and date/time. Note that the user is identified by their user ID, which can be found in the user administration screen (accessible by Tasktop admins only).
logs	extensions .log	Contains any logs generated when an extension is called. The extension will write out a log whenever the console.log function is called.
logs	migration-event-trace.log	Contains logs populated only when migrations are running.
logs	thread-dump.log	Contains all Tasktop thread information at the point of time the Support and Usage report is downloaded. This file will only be included if your Tasktop instance has crashed or if you have forced Tasktop to close.
logs	localhost. log	Tomcat's host log
logs	localhost_ access_log .txt	Tomcat's log of requests
logs	catalina. log	Tomcat's container log
logs	tasktop-service.log	Tasktop Windows service log, showing service start and stop
logs	keycloak-service.log	Keycloak Windows service log, showing service start and stop
logs	keycloak-stderr.log	Keycloak standard error output
logs	keycloak-stdout.log	Keycloak standard output
mappi ngs	collection-label.txt (i.	Contains information about collection mappings with transformation identifiers from Collection to Model and from Model to Collection.

	e. jira-defects.txt)	
metrics	metrics.json	Contains various metrics of the application.
metrics	change-detection-metrics.json	Contains metrics relating specifically to integrations and change detection.
repository metadata	repository-label.json (i.e. jira.json)	Contains repository metadata (i.e., repository version, repository timezone, repository api rate limit, repository default pagination size, repository additional metadata, connector timezone, repository state) for each repository connection configured.
schemas	collection-label.json (i.e. jira-defects.json)	Contains collection schema information (i.e. the same fields that would display on the mapping screen).
usage	usage-report.csv	Contains details on Tasktop usage without any with personal information included (i.e., names, e-mail addresses, etc).
usage	overview.json	Contains details such as repository versions, number of integrations, integration routes and last processed times, number of activities (creates and updates) by integration and repository, and number of person IDs seen by integration and repository. 💡 **Note**: Some integrations that do not have artifact associations will not have last processed times.

Usage Reports

Tasktop supplies a Usage Report to enable customers to review and understand their Tasktop usage.

Two reports are provided:

- A sanitized report that does not contain personal information (such as names, email addresses, or usernames), that is part of the Support and Usage Report file
- A Customer Usage Report which contains personal information (such as names, email addresses, and usernames), that can be used to analyze and reconcile user counts
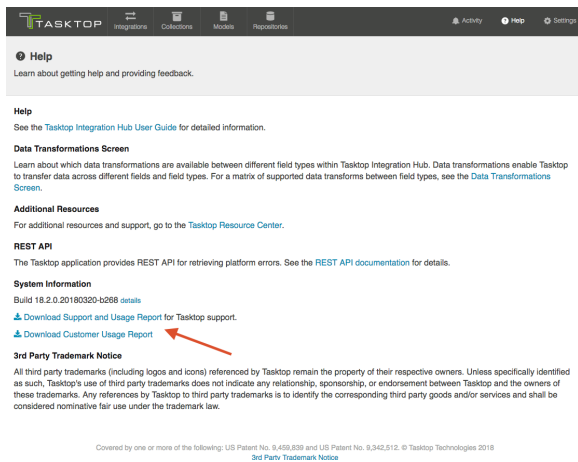
Both reports contain the following fields:

- **Tasktop Generated Person Identifier**:
    - This is generated to identify a person that flows between two or more repositories. If Person Reconciliation is in effect, the users that are the same across repositories will have the same Tasktop Generated Person Identifier. This field may be blank in scenarios where a person existed on an artifact seen by Tasktop, but where the field that contained that person did not flow to another repository.
- **Tasktop Generated Repository Person Identifier**:
    - This is generated for each unique person Tasktop sees within one repository. Note that the person field does not need to flow in order to be counted here. Since this is repository-specific, you could see two (or more) different Tasktop Generated Repository Person Identifiers that share the same Tasktop Generated Person Identifier.
- **Connector**:
    - Tasktop's name for the connector
- **Repository Label**:
    - The name (label) supplied by the customer for the repository
- **Integration Name**:
    - The name supplied by the customer for the integration within Tasktop
- **Collection Project**:
    - The collection and project names that contain the person
- **Repository Fields**:
    - The repository fields that the person was seen on during the course of a month
- **Model Fields**:
    - The model fields mapped to the repository fields listed above
- **Count**:
    - The number of times the Tasktop Generated Repository Person Identifier was seen for the given integration/collection/project combo in one month
- **Month**:
    - The month that the count (above) applies to

The customer-facing report also contains the following fields:

- **First Name**

- **Last Name**
- **Display Name**
- **Email**
- **Username**
- **Repository Person ID:**
    - A repository specific identifier. Some repositories provide an ID that is unique from the username.

💡 **Note**: The customer-specific fields above may be blank depending on the associated repository and whether Tasktop has retrieved them yet (these fields are retrieved periodically).

Both reports contain data collected over a rolling 2 year span.

Logging Settings

Tasktop provides two logging levels for the logs in the support and usage reports: Normal and Troubleshooting. Please see the Logging section of the Troubleshooting (Settings) screen for more details on how to configure each setting.

Error Message Appendix

The following is a complete list of error messages. Error messages are displayed on the Activity screen. More details on specific errors can be found under Troubleshooting and in our FAQ.

CCRRTT-0001E – An unexpected error occurred.

Description

An unexpected error has occurred.

User Action

Attempt to resolve error according to the specific error message.

CCRRTT-0002E – The maximum number of allowable errors has been reached.

Description

The maximum number of allowable errors has been reached. Any errors encountered after the maximum number will be discarded.

User Action

1. Open the errors page and resolve the listed errors

CCRRTT-0003E – The system has run out of memory.

Description

The system has run out of memory. Services have been stopped.

User Action

1. Increase the amount of memory available (see help docs).
2. Restart Tasktop Integration Hub.

CCRRTT-0005E – There is a conflicting artifact association.

Description

The artifact association could not be imported as an existing artifact association conflicts with it.

User Action

Contact support for assistance.

CCRRTT-0006W – Upgrade data migration cancelled.

Description

Data migration required to run an updated version of Tasktop Integration Hub was cancelled due to a configuration change or because Tasktop Integration Hub was shut down.

User Action

None, data migration will be resumed automatically.

CCRRTT-0006E – Migration cannot be completed as there are errors related to disabled repositories.

Description

Migration cannot be completed as there are errors related to disabled repositories.

User Action

1. Open the Activity page and delete all errors related to the specified repository, or
2. Navigate to the Repositories page and enable the specified repository

CCRRTT-1000E – Unable to communicate with repository.

Description

There was a network error when attempting to communicate with a repository.

User Action

1. Check the network connection between Tasktop Integration Hub and the repository.
2. Try connecting again later.

If the problem persists, contact your network administrator.

CCRRTT-1002E – An unexpected connector error occurred.

Description

An unexpected connector exception has occurred.

User Action

Attempt to resolve error according to the specific error message.

CCRRTT-1003E – An error occurred while executing an operation.

Description

An exception has occurred during the execution of a connector operation.

User Action

Attempt to resolve error according to the specific error message.

CCRRTT-1004E – Connection to LDAP directory failed.

Description

An unexpected error has occurred while attempting to establish a connection with an LDAP directory.

User Action

Attempt to resolve error according to the specific error message.

CCRRTT-1005E – An unexpected error occurred while communicating with an LDAP directory.

Description

An unexpected error has occurred while communicating with an LDAP directory.

User Action

Attempt to resolve error according to the specific error message.

CCRRTT-1104W – Authentication state for repository connection has expired.

Description

The authentication state for a repository connection has expired.

User Action

Typically, the authentication state for a repository connection expires on a periodic basis and authentication will be retried automatically. If the error persists, verify that the repository credentials for the associated repository are correct.

CCRRTT-1105E – The project configuration is invalid.

Description

The project configuration is not valid. This problem is usually caused by a project and/or type being deleted or renamed in the repository, but can also be caused by other problems, such as a change in user permissions within the repository.

User Action

1. Determine the cause of the problem from the specific error message
2. Navigate to the referenced repository collection or artifact union
3. Correct the problem on the repository and then click ? *Refresh Projects*?, or
4. Remove the referenced project from the repository collection or artifact union
5. If a project has been renamed add the renamed project back
6. Ensure all related routing, filtering, and mapping configurations are valid and up to date

CCRRTT-1107E – Connection could not be established with a repository due to a failure during authentication.

Description

There was an unexpected error while attempting to authenticate with a repository.

User Action

Attempt to resolve error according to the specific error message.

CCRRTT-1109E – Project configuration is outdated.

Description

The project configuration is outdated.

User Action

1. Identify the outdated project configured from the specific error message
2. Remove the outdated project from the associated Repository Collection or Artifact Union
3. Select ? *Manage Projects*? and press the ? *Refresh*? button
4. Add the project back

CCRRTT-1110E – Repository collection project cannot be found.

Description

The repository collection configuration is not valid. This problem is usually caused by a project and/or type being deleted or renamed in the repository, but can also be caused by other problems, such as a change in user permissions within the repository.

User Action

1. Go to the affected repository collection configuration page
2. If the project has been deleted, remove the referenced project from the repository collection
3. If the project has been moved, click ? *Manage Invalid Projects*? and select the project???s replacement
4. Ensure all related routing, filtering, and mapping configurations are valid and up to date

CCRRTT-1111E – Repository collection contains duplicate projects.

Description

The high-level container (i.e. the type of container chosen when clicking ? *Manage Projects*? on the Collections screen) has changed.

User Action

Before resolving this issue, please:

1. Review and write down the current artifact routing configuration for any integrations utilizing this collection as these must be reconfigured once the issue is resolved.
2. To ensure you understand the changes made to your collection, please navigate to the collection and review what is now selected under ? *Manage Projects.*? No changes will need to be made on this screen.

Once this issue is resolved, your artifact routing will be removed from any relevant integrations, and need to be manually reconfigured.

CCRRTT-1112E – Artifact is locked.

Description

The artifact is locked by another user or process.

User Action

See the specific error message for details on what artifact is locked. Ensure that no other user or process is currently using the artifact, and retry the operation.

CCRRTT-1113E – Connection could not be established with a repository due to an insecure connection.

Description

The repository connection could not be established due to an insecure connection.

User Action

- Attempt to resolve error according to the specific error message, or
- Navigate to the specific Repository and enable the setting for allowing insecure connections

CCRRTT-1114E – The artifact union configuration is invalid.

Description

The artifact union configuration is invalid.

User Action

1. Navigate to the artifact union configuration screen,
2. Correct the invalid configuration according to the specific error message

CCRRTT-1115I – This message is to notify you that events processing for this repository may be delayed due to the repository???s event rate limit.

Description

The events processing for this repository may be delayed because the repository???s event rate limit is set too low.

User Action

1. Navigate to each integration utilizing this repository and update the change detection interval, or
2. Navigate to the repository connection screen and update the event rate limit.

CCRRTT-1116E – Unable to create repository query.

Description

The repository query could not be created.

User Action

1. Verify that the query has not been renamed or deleted on the repository

2. Ensure that the repository user has sufficient permissions to access the query

CCRRTT-1401E – Integration must specify at least one route.

Description

An integration must contain at least one route.

User Action

1. Navigate to the integration routing page
2. Add at least one route

CCRRTT-1402E – Integration must satisfy style constraints.

Description

An integration must satisfy the constraints of its style. This type of error should not happen when an integration is built using the UI.

See the detailed message for more details about the parts of the integration that are invalid.

User Action

1. Navigate to the integration page
2. Adjust the configuration to be valid (according to the messages)
3. If this integration was created via the web UI, consider contacting support

CCRRTT-1403E – Integration must have all collections attached to the same model.

Description

Collections used in an integration must all be attached to the same model.

User Action

1. Determine which model the integration should be using
2. Navigate to the integration and determine which collections are not using this model
3. Either remove the identified collections from the integration, or
4. For each identified collection, set the mapping to the correct model

CCRRTT-1404E – Collection must have a mapping to a model.

Description

Repository Collections used in an integration must have a mapping to a model.

User Action

1. Navigate to the collection
2. Select a Model to create a mapping

CCRRTT-1405E – Integration must have a source Collection.

Description

An integration must have a source collection.

User Action

1. Navigate to the Integration
2. Add a collection to be used as a source

CCRRTT-1406E – Integration must have a target Collection.

Description

An integration must have a target collection.

User Action

1. Navigate to the Integration
2. Add a collection to be used as a source

CCRRTT-1408E – Integration failed to lookup artifact.

Description

An integration failed to locate the artifact to be modified. This can be caused by:

- a missing formatted ID value on the source artifact,
- an invalid formatted ID value on the source artifact, or
- the absence of a target collection which contains an artifact matched by the formatted ID.

See the detailed message for more details about the parts of the lookup that failed.

User Action

1. Navigate to the integration page
2. Ensure the key field is configured correctly on the field flow page
3. Ensure the data on the source artifact is correct
4. Ensure a matching artifact is contained in a target collection

CCRRTT-1409E – Integration has invalid filter.

Description

The filter used in the integration has become invalid.

User Action

1. Navigate to the integration filter in error.
2. Resolve each error that appears in the filter.

CCRRTT-1410E – Integration must specify a key identifier.

Description

An integration must specify a key identifier for the given collections. Key identifiers are used to determine how to locate artifacts in a target collection. They do this by specifying the field on the source model that contains the target artifact formatted id.

User Action

1. Navigate to the integration page
2. Select the two collections missing a key identifier
3. Navigate to the field flow page and configure a key identifier

CCRRTT-1411E – All specified routes of an integration must be configured.

Description

All specified routes of an integration must be configured.

User Action

1. Navigate to the integration routing page
2. Configure all routes which require configuration

CCRRTT-1412E – Integration has a conditional route with invalid configuration.

Description

The conditional routing configuration of the integration has become invalid.

User Action

1. Navigate to the integration route in error.
2. Resolve each error that appears in the routing configuration.

CCRRTT-1413E – Collection has invalid repository query.

Description

The repository query used in the collection has become invalid.

User Action

1. Navigate to the collection.
2. Resolve the error by selecting a different repository query.

CCRRTT-1414I – Tasktop Integration Hub is currently updating its operational data for this integration.

Description

Tasktop Integration Hub is currently updating its operational data for an integration.

User Action

1. Wait for the data to be updated.

CCRRTT-1415E – The routing configuration is invalid.

Description

The artifact routing for this integration is invalid because a route endpoint no longer exists, or cannot be routed to (i.e. a more specific route endpoint must be selected).

User Action

1. Navigate to each collection participating in the integration to review what is now selected under 'Manage Projects.'
2. Navigate to each integration and reconfigure the artifact routing. Once the routing is valid, this issue will clear.

CCRRTT-1416E – The twinless artifact update configuration is invalid.

Description

The twinless artifact update for this integration is invalid.

User Action

1. Navigate to the twinless update configuration for this integration.
2. Resolve the error according to the specific error message.

CCRRTT-10004E – Enterprise Data Stream Integration must have exactly one target SQL Collection.

Description

An Enterprise Data Stream Integration must reference a single SQL collection.

User Action

- Select a SQL Collection for the target of the Integration that is in error.

CCRRTT-10005E – Enterprise Data Stream Integration must have a source Collection.

Description

An Enterprise Data Stream Integration must reference at least one Collection to be used as a source of artifacts.

User Action

Select a source Collection for the Integration that is in error.

CCRRTT-10006E – Enterprise Data Stream Integration target Collection must have appropriate mapping.

Description

An Enterprise Data Stream Integration???s data Collection must be mapped to a model. This corresponds to the model desired to be reported on.

User Action

Add mappings for the Collection used in the Enterprise Data Stream Integration.

1. navigate to the Collection
2. add a mapping to a model

CCRRTT-10007E – Enterprise Data Stream Integration source Collection must provide the correct model.

Description

An Enterprise Data Stream Integration source Collection must be mapped to the same model as the target Collection.

User Action

Add relationship to the model for the source Collection used in the Enterprise Data Stream Integration

1. navigate to the Integration
2. identify the model of the target Collection
3. navigate to the source Collection in error, and ensure that its model matches the model of the target Collection
   - if the source collection is a Repository Collection, add a mapping to the corresponding model
   - if the source collection is a Gateway Collection, ensure its model is set to the corresponding model

CCRRTT-10008E – Enterprise Data Stream Integration target Collection must have exactly one project.

Description

An Enterprise Data Stream Integration???s Collection must have exactly one project.

User Action

1. Navigate to the Collection
2. Ensure it has exactly one project which corresponds to the database table

CCRRTT-10009E – Enterprise Data Stream Integration is missing required column.

Description

An Enterprise Data Stream SQL Collection???s underlying database table is missing a required column.

User Action

Add the required column to the underlying database table. See error message for missing column id.

CCRRTT-15002E – Integration services cannot be started due to a problem with the license.

Description

Tasktop Integration Hub integration services cannot be started due to a problem with the license. This problem can be caused by running the software without a license, using features that are not included in the installed license, or by having an invalid or expired license.

User Action

This problem can be resolved by installing a valid license using the following steps:

1. Obtain a valid license by contacting the Tasktop Support Center
2. Navigate to the settings page
3. Press the Apply New License button under License
4. Paste in the license text and press Save

CCRRTT-15005E – Repository cannot be used due to a problem with the license.

Description

The repository connection cannot be used because connections to repositories of this type are not enabled by the license.

User Action

This problem can be resolved by installing a valid license using the following steps:

1. Obtain a valid license by contacting the Tasktop Support Center
2. Navigate to the settings page
3. Press the Edit button under License
4. Paste in the license text and press Save

CCRRTT-15012E – Conditional field flow is not licensed.

Description

The integration cannot be used because conditional field flow configurations exist in this integration but are not enabled by the license.

User Action

Perform one of the following:

- Remove the conditional field flow from the integration

- Contact the Tasktop Support Center to obtain and install a license that includes the conditional field flow feature

CCRRTT-15013E – Artifact unions are not licensed.

Description

The collection cannot be used because mappings using a field from an artifact union exist within this collection but are not enabled by the license.

User Action

Perform one of the following:

- Remove the mappings using a field from an artifact union from the collection
- Contact the Tasktop Support Center to obtain and install a license that includes the artifact union feature

CCRRTT-16001E – Services cannot be started until Tasktop Integration Hub security has been initialized.

Description

Tasktop Integration Hub integration services cannot be started because secure password storage has not been configured and initialized.

User Action

1. Navigate to the Settings page
2. Specify the Master Password under Secure Password Storage

CCRRTT-16002W – The Tasktop Integration Hub services restart is taking longer than expected.

Description

The Tasktop Integration Hub services restart is taking longer than expected.

User Action

- Wait for the Tasktop Integration Hub services to restart and this issue will be removed automatically.
- If the Tasktop Integration Hub services do not restart (this issue is still present) after 30 minutes, please contact the Tasktop Support Center for assistance: ? *https://links.tasktop.com/support*?.
- Do not restart Tasktop Integration Hub without assistance from support

CCRRTT-17001E – Mapping cannot be applied since it is not valid within the current context.

Description

The mapping cannot be applied since the mapping is not valid for the artifacts in the current context.

User Action

1. Determine the source of the problem from the specific error message
2. Either update the mapping to match the artifacts and model in use, or
3. Update the corresponding artifact schema to match the mapping, for example by changing a field type

CCRRTT-17002E – Collection model mapping is invalid.

Description

The collection model mapping is not valid due to inconsistencies between the collection schema, the model schema and the mapping.

User Action

1. Determine the cause of the problem from the specific error message
2. Navigate to the mapping
3. Update the mapping to match the collection and model in use, or
4. Update the corresponding collection artifact schema to match the mapping, for example by changing a field type, or
5. Update the model to match the mapping, for example by adding a field, or changing a field type

CCRRTT-17003E – Artifact could not be created or updated because one or more values cannot be accepted.

Description

An artifact could not be updated or created because one or more of its values are not valid. See the specific error message for details.

User Action

1. Identify the fields and values that are in error from the specific error message
2. Correct the source data, either by
   - updating the source artifact, or
   - by making changes to the mapping, or
   - by making changes to the target system so that the provided data is valid, or
   - by providing a new artifact via a Gateway Collection

CCRRTT-17004W – Artifact cannot be processed since it is currently in use.

Description

Artifact cannot be processed since it is currently in use. This temporary problem occurs when Tasktop Integration Hub attempts to process changes to an artifact concurrently.

User Action

This error will resolve itself automatically, no user action required.

CCRRTT-17005E – Field flow is invalid.

Description

The field flow configuration is not valid due to inconsistencies between the the model schema and the field flow.

User Action

1. Determine the cause of the problem from the specific error message
2. Navigate to the integration
3. Select the collection pair
4. Navigate to the field flow
5. Update the field flow to match the model in use, or
6. Update the model to match the field flow, for example by adding a field

CCRRTT-17006E – Artifact was created but some values could not be set.

Description

An artifact was created by an integration but some values on the artifact could not be set. The resulting artifact has some field values that may not be correct.

User Action

1. Determine the cause from the specific error message
2. Either retry the corresponding activity, or
3. Verify the state of the created artifact and manually adjust values as necessary

CCRRTT-17007E – Conflict resolution strategy is invalid.

Description

The conflict resolution strategy configuration is invalid.

User Action

1. From the integration, navigate to the conflict resolution strategy
2. Select an option for the conflict resolution strategy

CCRRTT-17008E – Artifact could not be processed as it did not meet any of the configured conditions on the Conditional Artifact Routing page.

Description

Artifact could not be processed as it did not meet any of the configured conditions on the Conditional Artifact Routing page.

User Action

- Update the conditions configured on the Conditional Artifact Routing page to ensure the artifact???s field value is accounted for, or

- Update fields on the artifact to ensure that it meets the conditions set on the Conditional Artifact Routing page, or
- Update specification for handling artifacts not matched by conditions configured on the Conditional Artifact Routing page to ? *Ignore*? or ? *Default Route*? instead of ? *Error*?.

CCRRTT-17009E – Invalid state transition.

Description

An extension provided invalid values when attempting to transition an artifact.

User Action

1. Identify the extension that produced invalid values
2. Identify the fields and values that are in error from the specific error message
3. Modify the extension to produce a valid transition

CCRRTT-17010E – Repeated state transition.

Description

An extension attempted to transition an artifact with the same transition more than once.

User Action

1. Identify the extension from the error message
2. Modify the extension to avoid repeated transitions of the same type for an artifact

CCRRTT-17011E – Extension completed with an error.

Description

An extension completed with an error. See the specific error message for details.

Extensions complete with errors for one of two reasons:

- the extension intentionally raised an error, for example to indicate that a business rule was not satisfied
- the extension itself has an error in its implementation

User Action

1. Determine from the specific error message the cause of the error
2. Either modify the extension to prevent the error from occurring, or
3. Modify the source or target artifact to satisfy the condition that caused the error

CCRRTT-17013E – The state transition requires the selection of model fields.

Description

A state transition extension is configured in a collection that has no model fields selected.

User Action

Either disable the state transition of the collection or select model fields for the state transition.

To select the fields for the state transition:

1. navigate to the collection
2. navigate to the collection state transitions via the ? *Configure State Transition*? link
3. add the model fields required by the state transition in "State Transition Fields"

To disable state transitions in the collection:

1. navigate to the collection
2. navigate to the collection state transitions via the ? *Configure State Transition*? link
3. select ? *None*? for "State Transition"

CCRRTT-17014E – Relationship values could not be resolved during synchronization.

Description

One or more relationship links could not be resolved as part of a synchronization.

This problem occurs when two artifacts that link to each other are synchronized out of order.
This commonly occurs when one artifact (A) links to another (B), but the linked-to artifact B has not yet been synchronized.

When the copy of artifact A (A') is created in the target repository, a link to a copy of B (B') cannot be created at that time since B' has not yet been created.
This problem usually resolves itself once B' is created; the link from A' to B' is created once B' becomes available.

User Action

- None; wait for the error to be resolved automatically, or
- Remove the unresolved link from the artifact being synchronized

CCRRTT-17015E – Relationship values could not be resolved during synchronization.

Description

One or more relationship links could not be resolved as part of a synchronization.

This commonly occurs when one artifact (A) links to another (B), but the linked-to artifact B has more than one corresponding copy in the target repository. This can be caused by having separate synchronization integrations that cause B to be copied into the target repository.

User Action

- Remove the link from A to B, or
- Use the Artifact Pairs tab on the Activity page to delete any invalid or outdated associations involving B, or
- Remove any unnecessary synchronization integrations which can affect B

CCRRTT-17016E – An unexpected error occurred when creating the artifact.

Description

An unexpected error occurred when creating the artifact. The artifact may or may not have been created.

User Action

1. Do not retry the event without guidance from Tasktop Support,
2. Contact the Tasktop Support Center for assistance: "https://links.tasktop.com/support"

CCRRTT-17017E – The repository does not support artifact creation.

Description

The repository does not support artifact creation.

User Action

1. Navigate to the corresponding integration,
2. Disable artifact creation flow into the specified collection,
3. Remove all routes flowing into the specified collection.

CCRRTT-17018E – Model does not have all fields required by the state transition.

Description

A state transition extension is configured in a collection that requires fields that are not configured in the model.

User Action

Either remove the missing fields in the state transition configuration, or ensure that the model has the required fields.

To add the fields to the model:

1. navigate to the model
2. add the fields

To change the required fields of the state transition extension from the collection:

1. navigate to the collection
2. navigate to the collection state transitions via the ? *Edit state transition*? link
3. modify the list of model fields

CCRRTT-17019E – Target collection partition could not be resolved during synchronization.

Description

The work item artifact could not be synchronized due to a missing or invalid route.

User Action

1. Verify which container this artifact is in in the repository, and ensure that either that container or one of its ancestors has been configured as part of a mirrored container structure; or
2. Ensure that a route has been created for the container in which this artifact originates in the work item integration; or
3. Ensure that the target container has not been deleted. If it has, and if an error exists for it, re-create the container on the Errors screen. To ensure you see an error for the deleted container, make a change to the still-existing corresponding container in the other collection.

CCRRTT-17020E – Associated target container could not be resolved during synchronization.

Description

The artifact could not be synchronized because the target container could not be found.

User Action

1. No action needed, the synchronization should be fixed automatically when the containers synchronize.

CCRRTT-17021E – An error occurred when performing state transitions.

Description

A transition was attempted on an artifact but an error resulted.
The artifact may be in an incorrect state.

User Action

Either address the cause from the specific error message, or disable/reconfigure the state transition of the collection.

1. If the specific error message has a cause, verify the state of the target artifact and manually adjust values as necessary

To disable/reconfigure state transitions in the collection:

1. navigate to the collection
2. navigate to the collection state transitions via the ? *Configure state transition*? link

3. adjust the relevant state transitions

CCRRTT-17022E – The associated container could not be found.

Description

The container associated with the parent container of this artifact could not be found.

User Action

- If the parent container is configured in a route, update the routing configuration to use an existing container
- If the parent container is synchronized by an integration, update the parent container to generate an event for the parent container, and use the ? *Recreate Artifact*? action

CCRRTT-17023E – For the artifact pair import to succeed, the associated integration must be running.

Description

For the artifact pair import to succeed, the associated integration must be running.

User Action

- Run the integration associated with the artifact pair file.

CCRRTT-17024E – An error occurred when processing the output of an extension.

Description

An error occurred when processing the output of an extension. See the specific error message for details.

User Action

1. Determine from the specific error message the cause of the error
2. Either modify the extension to prevent the error from occurring, or

3. Modify the source or target artifact to satisfy the condition that caused the error

CCRRTT-17025E – Field value configuration required.

Description

Field value configuration required. The field type for a field in your integration has changed and its field values must be re-mapped.

User Action

1. Navigate to the appropriate collection (linked above)
2. Go to Field Mapping screen
3. Click ? *Configure*? next to any fields with a ? *!*? icon
4. Map the field values and save
5. Important: Remove this issue to re-enable the integration

CCRRTT-17026E – Comment flow is invalid.

Description

The comment flow configuration is not valid. This can happen if a repository???s comment visibility support has changed.

User Action

1. From the integration, navigate to the comment flow
2. Configure the comment flow as desired

CCRRTT-17027E – The conditional field flow has an invalid condition.

Description

The Conditional Field Flow configuration is not valid. This can happen if either a field or value is not resolvable in the model schema.

User Action

1. From the integration, navigate to the Field Flow

2. Configure the Conditional Field Flow so that all fields and values configured in the conditions are in the respective model used.

CCRRTT-17028E – Test step model mapping is invalid.

Description

The test step model mapping is not valid due to inconsistencies between the test step schema, the test step model schema and the mapping.

User Action

1. Determine the cause of the problem from the specific error message
2. Navigate to the test step mapping
3. Update the mapping to match the collection and model in use, or
4. Update the corresponding test step schema to match the mapping, for example by changing a field type, or
5. Update the test step model to match the mapping, for example by adding a field, or changing a field type

CCRRTT-17029E – Time entry model mapping is invalid.

Description

The time entry model mapping is not valid due to inconsistencies between the time entry schema, the time entry model schema and the mapping.

User Action

1. Determine the cause of the problem from the specific error message
2. Navigate to the time entry mapping
3. Update the mapping to match the collection and model in use, or
4. Update the corresponding time entry schema to match the mapping, for example by changing a field type, or
5. Update the time entry model to match the mapping, for example by adding a field, or changing a field type

CCRRTT-17030W – Configuration delta table is currently unavailable as it is being migrated.

Description

Configuration delta table is unavailable as it is being migrated. This temporary problem may occur when a user attempts to access the list of configuration changes before the migration completes shortly after an upgrade.

User Action

The configuration delta table migration will complete as a background job, no user action required.

CCRRTT-17031E – Failed to transform imported configuration change.

Description

Failed to transform imported configuration change into change applicable for this instance due to missing elements.

User Action

Create the required missing elements before re-importing the configuration change.

CCRRTT-20000E – No integration is listening to the Gateway Collection.

Description

A Gateway Collection has been used, but the collection is not configured as a source in an integration. The payload has been lost.

User Action

1. Use the Gateway Collection in an integration, or
2. Stop pushing to the collection (from the external source)

CCRRTT-20004E – Relationship fields of a Gateway Collection must be configured to specify the related repository.

Description

A Gateway Collection must configure the Relationship(s) fields to associate them with the repository having referenced artifacts.

User Action

1. Navigate to the Gateway collection
2. Locate the ? *Relationship Field Configuration*? section in the UI
3. For each field, select the repository that is associated with that relationship.

CCRRTT-20005E – Gateway collection must have a model.

Description

A Gateway Collection must have a model configured.

User Action

1. Navigate to the Gateway collection
2. Select a model and save the changes

CCRRTT-20006E – Gateway Collection cannot be used with the configured payload transformation extension due to a restriction in the license.

Description

A gateway collection has been configured with a payload transformation extension, which is not permitted by the current license.

User Action

Perform one of the following:

- Delete the offending gateway collection
- Remove the payload transformation extension from the offending gateway collection

CCRRTT-20007E – Gateway collection must use a token.

Description

A Gateway Collection must use a token.

User Action

1. Navigate to the Gateway collection
2. Generate a token and save the changes

CCRRTT-21001E – An unexpected error occurred while sending an email.

Description

An error occurred while attempting to send an email.

User Action

1. Verify that the email settings are specified correctly in the settings
2. Attempt to resolve error according to the specific error message

CCRRTT-21002E – Failed to authenticate with the email server.

Description

The mail server rejected the client connection because it was not able to authenticate.

User Action

1. Verify that the email settings are specified correctly in the settings
   - Double-check the email server hostname and port
   - Double-check the email server credentials
2. Attempt to resolve error according to the specific error message

CCRRTT-22001E – Artifact Association records with unknown Artifact Handles found and deleted during upgrade.

Description

During database upgrade, one or more associations were discovered to have an invalid reference. The records that reference nonexistent associated records were logged and deleted.

User Action

Do not cancel this error or run the associated integration without consulting Tasktop Support. (https://links.tasktop.com/support)

CCRRTT-30000E – An unexpected error occurred.

Description

An unexpected error has occurred. Check the specific error message for details.

User Action

Check the specific error message for details of the failure. If possible correct the problem described in the error message, or contact your administrator for assistance.

CCRRTT-30001E – Not found.

Description

The entity was not found because the entity no longer exists on the server.

User Action

Ensure that the provided entity id is correct, and if not correct the id and try again.

CCRRTT-30002E – The data provided was not valid.

Description

The data provided was not valid. See the specific error message for details.

User Action

Correct the problem described in the specific error message and try again.

CCRRTT-30003E – The connector kind was not found.

Description

The connector kind was not found.

User Action

Ensure that the connector kind is specified correctly and try again.

CCRRTT-30004E – The request entity was not valid JSON.

Description

The request entity was not valid JSON.

User Action

Ensure that the request payload is formatted as a valid JSON entity and try again.

CCRRTT-30005E – Secure password storage must be initialized.

Description

Secure password storage has not been initialized.

User Action

Configure secure password storage via the settings page.

CCRRTT-30006E – Error communicating with {0} repository.

Description

Error connecting to repository. See the specific error message for details.

User Action

Check the specific error message for details of the failure. If possible correct the problem described in the error message, or contact your administrator for assistance.

CCRRTT-30007E – Error processing request MIME attachment.

Description

The request MIME attachment could not be accepted either due to a bad request or an I/O failure.

This problem can be caused by insufficient disk space or lack of write permissions in the Tasktop Integration Hub application temporary directory.

User Action

1. Verify that the temporary directory of the Tasktop Integration Hub application is writable,
   - The Tasktop Integration Hub application must have write permissions to the directory
   - The directory must have sufficient available space
2. Try again

CCRRTT-30008E – Tasktop Integration Hub is stopped, see the Activity View and error log for more details.

Description

Tasktop Integration Hub has been stopped due to unrecoverable errors. See error log for more details.

User Action

Correct the problem described in the specific error message and restart.

CCRRTT-30009E – The database is not available.

Description

The configuration database is unavailable.

User Action

Ensure the configuration database is online and can be reached and ensure Tasktop Integration Hub???s database settings are correct.

CCRRTT-30010E – Connection settings are not valid.

Description

The provided connection settings are not valid. See the specific error message for details.

User Action

Correct the problem described in the specific error message and try again.

CCRRTT-30011E – The database is locked for maintenance and cannot currently be used.

Description

The configuration database is locked for maintenance and cannot be used.

User Action

Wait for the ongoing maintenance to complete.

CCRRTT-30012E – The database is in use by another instance of the application.

Description

The Configuration database is in use by another instance of the application.

User Action

If this is the Tasktop Integration Hub instance which should be running, then shut down any other instances of Tasktop Integration Hub using the same database and restart this instance. Otherwise shut down this instance of Tasktop Integration Hub.

CCRRTT-30013E – Temporary error communicating with {0} repository.

Description

Temporary error connecting to repository. See the specific error message for details.

User Action

Retry your action. If the problem persists, contact your administrator for assistance.

CCRRTT-30014E – Error communicating with repository. Insecure connections are not allowed.

Description

The repository connection could not be established due to an insecure connection.

User Action

- Attempt to resolve error according to the specific error message, or
- Navigate to the specific Repository and enable the setting for allowing insecure connections

CCRRTT-30015E – Deployment configuration error.

Description

Configuration applicable to the current deployment is incomplete or invalid.

User Action

Contact Tasktop customer support.

CCRRTT-30016E – Unauthorized user error.

Description

The current user is not authorized due to a restriction in the license.

User Action

Contact your Tasktop Integration Hub administrator.

CCRRTT-30017E – The license is expired.

Description

Tasktop integration services cannot be started because the current license has expired.

CCRRTT-30018E – No license has been configured.

Description

Tasktop integration services cannot be started because the no license has been configured.

CCRRTT-30019E – The application is currently starting up.

Description

The application is starting up and cannot be accessed at this time.

User Action

Wait for the application to finish starting up

CCRRTT-30020E – An error occurred when reading the database connection settings.

Description

The connection settings for the operational database are inaccessible.

User Action

Ensure the tasktop-db.json file is correctly formatted and the Tasktop Integration Hub user has permission to read it.

CCRRTT-30021E – The history page is not available before the necessary data migrations are complete.

Description

The history page is not available before the necessary data migrations are complete.

User Action

Wait for the migrations to complete.

CCRRTT-30022W – Repository Connection disabled.

Description

The repository connection will be disabled until it is manually enabled.

User Action

- Enable the repository connection manually on the Repository Connection page, or
- Leave the repository connection disabled and ignore this warning

CCRRTT-50001E – Unable to propagate artifact changes since the target artifact has been removed.

Description

Changes to an artifact cannot be propagated to the corresponding artifact in the alternate repository of a synchronization integration since the target artifact has been removed.

User Action

- Use the ? *Recreate Artifact*? action to have Tasktop Integration Hub recreate the artifact that was deleted in the end system and associate it with the still-existing artifact in the other repository (putting them in sync with one another), or
- Delete the associated artifact, or
- Move the associated artifact out of its collection such that the artifact is no longer synchronized, or

- Apply an artifact filter to ensure updates to the artifact will not be synchronized. To do so, make sure the artifact does not meet the filter criteria specified and make sure to configure the filter to apply to artifact updates

CCRRTT-50002E – A conflict has occurred during synchronization.

Description

A field conflict was detected when synchronizing artifacts. A field conflict occurs when the value of a field that is set to flow bidirectionally conflicts across your repositories.

The synchronization of these artifacts was halted with an error because a conflict resolution strategy of ? *Error Upon Conflict*? was configured and the system was unable to propagate the value from either artifact without overwriting a change from the other artifact.

User Action

- Change the conflict resolution strategy to have one of the repositories dominate in case of a conflict, or
- Manually change the conflicting value on at least one of the artifacts such that there is no longer a conflict, or
- Change the field flow of the affected field to be unidirectional (in which case a conflict is not possible)

CCRRTT-50005E – A conflict has occurred during synchronization.

Description

A conflict was detected when synchronizing artifact containment. A conflict occurs when one or more containers of synchronized artifacts is changed for both artifacts.

User Action

- Change the container of one or both artifacts to its original value or
- Change the conflict resolution strategy to have one of the repositories dominate

CCRRTT-50006E – Unable to update artifact due to values for dependent single selects not found.

Description

Unable to find a new value for an unchanged dependent field.

User Action

- From the error message find the field that the field in error depends on
- In the repository add a value with the same label as the one provided in the error message

OR

- Change the field that the field in error depends on back to its original value

OR

- Remove the mapping for the field that the field in error depends on

CCRRTT-50007E – Multiple matching containers were found.

Description

Multiple matching containers were found when attempting to match containers.

User Action

- Disable container matching in the container mirroring configuration, or
- Rename the containers such that only one container matches, or
- Change the container matching configuration to choose the first matching container, or
- Change the container matching configuration to match containers differently

CCRRTT-50008E – This integration cannot be started because a required relationship cannot be resolved.

Description

The integration cannot be started because a required Relationship field cannot be resolved.

User Action

- Create an integration to synchronize the artifacts referenced by the specified field, or
- Add a constant mapping to the specified field.

CCRRTT-50009E – Time Tracking integration model must have a field of type Time Entries.

Description

Model used in a Time Tracking integration must have a field of type Time Entries.

User Action

Either

1. Navigate to the model
2. Add a field of type Time Entries

Or

1. Create or select another model having a field of type Time Entries
2. Ensure that each collection used in the integration is using the selected model

CCRRTT-50010E – Time Tracking integration Collection must have a field mapping to a field of type Time Entries in the Model.

Description

Collections used in a Time Tracking integration must have a field mapped to the model Time Entries field.

User Action

1. Navigate to the collection model mapping
2. Add a field mapping to the model Time Entries field

CCRRTT-50011W – Time Tracking integration target Collection does not support impersonation of the Worker field.

Description

The selected collection does not support worklog impersonation and so has limited use as the target in a Time Tracking integration.
The worklogs will be filed under the user of the target repository connection.

CCRRTT-50012E – Time Tracking integration Collection does not support time entry filtering.

Description

Time entry filtering is configured for a collection pair but the source collection does not support it.

User Action

1. Navigate to the integration
2. Select the collection pair
3. Navigate to Time Entry Filtering
4. Disable the filter

CCRRTT-50013W – Artifact cannot be created currently as other artifact creations are being processed.

Description

Artifact cannot be created currently as other potentially conflicting artifact creations are being processed. This temporary problem can occur when Tasktop Integration Hub attempts to create artifacts on both sides of an integration concurrently.

User Action

This error will resolve itself automatically, no user action required.

CCRRTT-50014E – The test step flow configuration is invalid.

Description

The test step flow configuration is invalid.

User Action

1. Navigate to the test step flow configuration screen,
2. Correct the invalid flow configuration according to the specific error message

CCRRTT-50015E – The routing configuration for this container + work item integration is invalid.

Description

The artifact routing for this container + work item integration is invalid because one side of the integration has multiple container collections of the same type, and artifacts are flowing away from that side.

User Action

1. Navigate to each integration and reconfigure the artifact routing. Once the routing is valid, this issue will clear.
2. When multiple container collections of the same type exist, integrations can only routed toward that side of the integration.
3. Ensure artifact flow is not bidirectional.

CCRRTT-50016E – Artifact could not be processed as it did not meet any of the configured conditions on the Shared Container Mirroring page.

Description

Artifact could not be processed as it did not meet any of the configured conditions on the Shared Container Mirroring page.

User Action

- Update the conditions configured on the Shared Container Mirroring page to ensure the artifact???s field value is accounted for, or
- Update fields on the artifact to ensure that it meets the conditions set on the Shared Container Mirroring page, or
- Update specification for handling artifacts not matched by conditions configured on the Shared Container Mirroring page to ? *Ignore*? or ? *Default Type*? instead of ? *Error*?.

CCRRTT-50017E – The integration is missing required field mappings.

Description

Container + Work Item integrations using Shared Container Mirroring conditions require the model field within the condition to be mapped within its associated collections.

User Action

1. Navigate to the associated collections,
2. Add a field mapping to and from the model field used within the Shared Container Mirroring conditions

CCRRTT-50018E – The artifact could not be processed as the artifact it depends on has not synchronized.

Description

The artifact could not be processed as the artifact it depends on has not synchronized.

User Action

- Ensure that the artifact specified within the detailed error message is included in the integration,
- Wait for it to synchronize

CCRRTT-50019E – Container matching cannot be enabled when the parent field is mapped with an extension.

Description

Container matching cannot be enabled when the parent field is mapped with an extension.

User Action

- Remove the extension from the parent field mapping within the related collections, or
- Disable container matching in the related integration

CCRRTT-50020E – Collections with multiple artifact types cannot be used in a Container + Work Item Synchronization.

Description

Collections with multiple artifact types cannot be used in a Container + Work Item Synchronization.

User Action

1. Use a different collection in the integration
2. Remove unnecessary artifact types from the collection

CCRRTT-60001E – Error initializing password encryption.

Description

Secure password storage requires 256-bit AES encryption which is not available in the Java runtime environment.

User Action

This problem can be resolved by installing the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files in the Java runtime environment. The download is available from oracle.com including a README file with installation instructions.

Alternatively, the unencrypted level of the password store maybe used.

CCRRTT-61001E – Connector is missing requirements.

Description

The connector requirements are not met.

User Action

Read the connector-specific error message to determine which requirements are unsatisfied.

To provide 3rd party components such as a library or SDK, follow the following steps:

1. Navigate to the ? *Repositories*? screen.
2. Select the repository for which the requirements were unsatisfied.

3. On the repository connection screen, provide the required files.

CCRRTT-61101E – Connection credentials were not accepted by the repository.

Description

There was an authentication error while attempting to communicate with a repository.

User Action

1. Verify that the credentials for the associated repository are correct in the settings.

If these steps do not resolve the error, ensure that the user has sufficient permissions in the target repository to create and edit artifacts.

CCRRTT-61102E – Connection HTTP proxy credentials were not accepted by the repository.

Description

There was an authentication error with the proxy server while attempting to communicate with a repository.

User Action

1. Verify that the proxy credentials for the associated repository are correct in the settings.

If these steps do not resolve the error, contact your network administrator for assistance.

CCRRTT-61103E – Connection settings are invalid.

Description

The connection settings are invalid.

User Action

1. Open the connection settings page for the repository that is in error.
2. Update the connection???s settings to valid values.

If these steps do not resolve the error, contact support for additional assistance.

CCRRTT-61104E – Tasktop Integration Hub is unable to communicate with this repository as it is experiencing high server load.

Description

Tasktop Integration Hub is unable to communicate with this repository as it is experiencing high server load. This problem is usually caused by exceeding the number of API calls a repository can receive or otherwise placing a high load on the repository.

User Action

This error will resolve itself automatically when the repository is no longer experiencing high server load. You can also set an event rate limit on the repository connection screen in Tasktop Integration Hub to limit the number of Tasktop Integration Hub events processed for this repository per minute.

CCRRTT-61105I – This message is to notify you that Tasktop Integration Hub has exceeded the allowed rate limit of the target repository. Any events that did not get processed will automatically be retried and processed upon subsequent attempts.

Description

This message is to notify you that Tasktop Integration Hub has exceeded the allowed rate limit of the target repository. Any events that did not get processed will automatically be retried and processed upon subsequent attempts.

User Action

None.

CCRRTT-61106E – Repository Connection disabled.

Description

The repository connection will be disabled until it is manually enabled.

User Action

- Enable the repository connection manually on the Repository Connection page, or
- Leave the repository connection disabled and ignore this warning

CCRRTT-63001E – Integration services cannot be started since the current license has expired.

Description

Tasktop Integration Hub integration services cannot be started because the current license has expired.

User Action

This problem can be resolved by installing a valid license using the following steps:

1. Obtain a valid license by contacting the Tasktop Support Center
2. Navigate to the settings page
3. Press the Apply New License button under License
4. Paste in the license text and press Save

CCRRTT-64001E – Integration cannot be used with the configured repositories due to a restriction in the license.

Description

An integration cannot be run because it is configured with repository pairs which are invalid under the current license restrictions.

User Action

Perform one of the following:

- Delete the offending integration
- Disable the offending integration
- Update the offending integration to use repository pairs allowed under the current license restrictions

CCRRTT-65001E – Extension cannot be used because of a restriction in the license.

Description

A value transformation extension is present which is not permitted by the current license.

User Action

Perform one of the following:

- Provide a license that includes extensions of this type, or
- Remove extension by navigating to the the Settings -> Extensions page

CCRRTT-66001I – Tasktop Integration Hub is currently updating its operational data with a collection's project replacements.

Description

Tasktop Integration Hub is currently updating its operational data with a collection???s project replacements.

User Action

1. Wait for collection update to complete.

CCRRTT-66002I – Tasktop Integration Hub is currently updating its operational data with a collection's project replacements.

Description

Tasktop Integration Hub is currently updating its operational data with a collection???s project replacements.

User Action

1. Wait for collection update to complete.

CCRRTT-66003W – Integration data migration is currently in progress.

Description

A background job is currently in progress.

User Action

Wait for the background job to complete.

CCRRTT-66003I – Project replacement cannot be applied because the handle schema is missing fields from the builder schema.

Description

Project replacement cannot be applied because the handle schema is missing fields from the builder schema.

User Action

1. Check for related upgrade background jobs,
2. If some exist, wait for them to finish, and retry the background job
3. Otherwise, contact the Tasktop Support Center for assistance: "https://links.tasktop.com/support"

CCRRTT-66004W – Background job cancelled.

Description

A background job was cancelled due to a configuration change or because Tasktop Integration Hub was shut down.

User Action

None, the job will be resumed automatically.

CCRRTT-67000E – Target repository must contain valid containers for this integration.

Description

Could not find a container where the target artifact could be created.

User Action

1. Identify the target repository where the container could not be located for the target artifact from the specific error message.
2. Ensure a container that satisfies the following constraints is created in the target repository:
   - Is able to contain the target artifact.
   - Is a sibling of the container of the referenced artifact.

CCRRTT-67001E – A field mapping must exist from a model String or Rich Text field to a target Relationship or Relationships field.

Description

A field mapping must exist from a String or Rich Text field in the model to a Relationship or Relationships field in the target collection. Such a mapping is required to determine where to flow any commit artifacts processed by this integration.

User Action

1. Configure a field mapping from your model Commit Message field to a Relationship or Relationships field in the target collection.

Metrics

See the Tasktop Editions table to determine if your edition contains basic or advanced Metrics functionality.

Introduction

Tasktop Integration Hub provides a **Metrics dashboard** to help you better understand Tasktop activity such as:

- Number of users
- Number of artifacts created by Tasktop
- Number of artifact updates by Tasktop

These metrics are a great tool to:

- Understand and troubleshoot downtime
- Communicate the value of Tasktop to your organization
- Analyze trends and patterns within your organization, such as:
  - Are there certain times of year when higher quantities of customer requests flow from your CRM tool to your Requirements tool?
  - Have defects flowing from your ITSM tool to your Agile tool decreased over time?
  - ...and more!

The data used to create the metrics refreshes each time the page is reloaded.
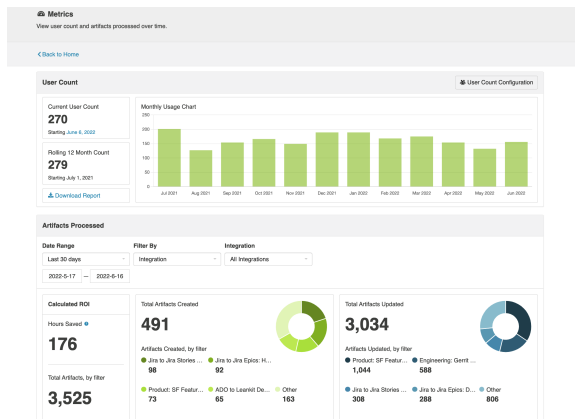
Viewing the Metrics Dashboard

To access the Metrics Dashboard, click **Metrics** in the right corner of the screen.
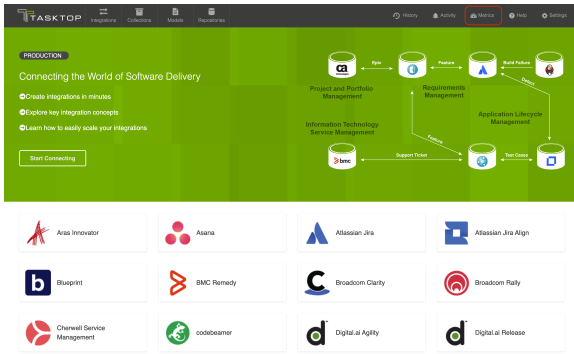
User Counting

> ⚠️ **The User Count dashboard should only be configured under the guidance of your Tasktop contact.** If the dashboard is configured without guidance, the user count reflected in your dashboard may be inaccurate.
>
> Note that the above does not limit the terms of Tasktop's Support and Maintenance Policy.

The User Count dashboard lets you view and configure your user count so you can anticipate precisely how many user licenses you will need to purchase upon renewal.

On the User Count dashboard, you'll see the Current User Count, Rolling 12 Month Count, and Monthly Usage Chart.
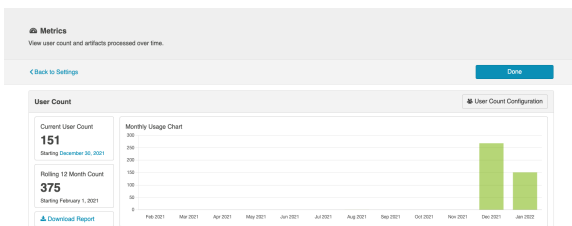
The **Current User Count** represents the number of users from your most recent license renewal date to the current date.

Note: Users that exist in data processed before the current date are not included in the count.
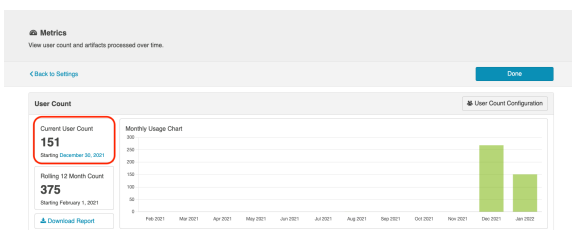
For the **Rolling 12 Month Count**, the same logic as the Current User Count applies; however, this count begins from precisely one year prior to the current date. This count, along with the **Monthly Usage Chart**, can help you understand trends in the number of users Hub processes over time.
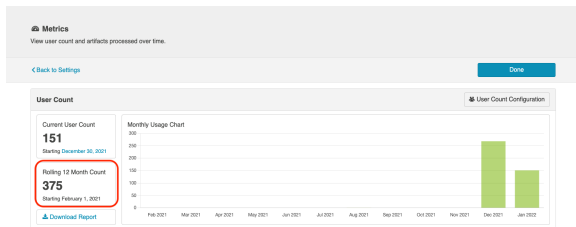
To find the number of distinct users observed in artifacts processed by Tasktop, you can download the **User Count Report**.

This report provides a pipeline view of how many users remain in the user counting logic after each stage. It can also help you understand how exclusions are applied to determine the user count.

The User Count Report includes the following contents:

current-usage-breakdown: The files in this folder can be used to determine the number of users from the most recent license renewal date to the current date.

- **1-raw-data.csv**: This is the starting point for the user count.
- **2-deduplicated-based-on-user-fields.csv**: This intermediate report only contains users who are included in the user count and can be used to trace users if needed.
- **3-reconciled-based-on-settings.csv**: This is the final result after applying person reconciliation overrides.
- **4-unmatched-based-on-settings.csv**: This report contains users who are not associated with another person and could not be reconciled based on the configured settings.

12-month-usage-breakdown: The files in this folder can be used to determine the number of users from precisely one year before the current date.

- **1-raw-data.csv**: This is the starting point for the user count.
- **2-deduplicated-based-on-user-fields.csv**: This is an intermediate report that can be used to trace users.
- **3-reconciled-based-on-settings.csv**: This is the final result after applying person reconciliation overrides.
- **4-unmatched-based-on-settings.csv**: This report contains users who are not associated with another person and could not be reconciled based on the configured settings.

final-counts.csv: This is the final summary report of the user count results.

person-reconciliation-template.csv: This template can be used to configure Person Reconciliation Overrides on the User Count Configuration screen.

user-count-settings.json: These are the settings configured on the Metrics screen that are used to obtain the count.

To download the **User Count** Report, click **Download Report**.
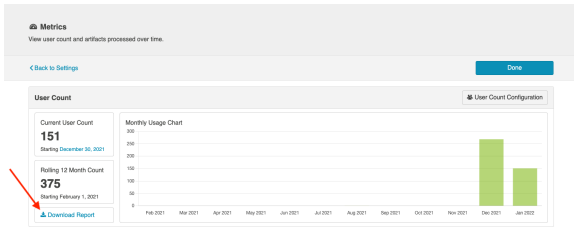
Click **User Count Configuration** to configure your user count settings.
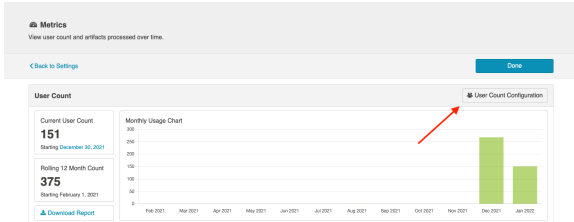
On the **User Count Configuration** screen, you can identify exclusions, person reconciliation overrides, and other options that determine user counts.

Integration Exclusions

In the **Integration Exclusions** tab, you can determine which integrations involve general employees submitting tickets to an IT help desk and non-production integrations, so you can ensure those users are excluded from your user count.

Note: To exclude users involved in an ITSM integration, you must also set the user type to 'Requester' in the User Type Exclusions tab.

User Type Exclusions

In the **User Type Exclusions** tab, you can further refine exclusions from your user count by determining the user type for the person fields in your repositories.
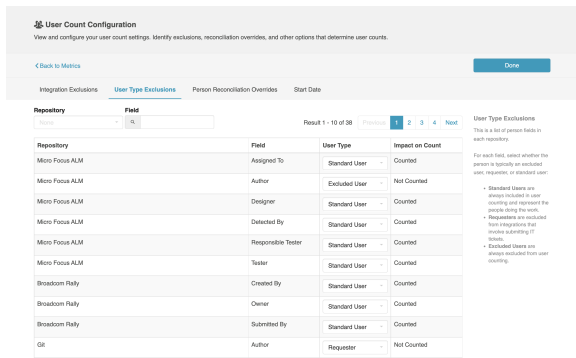
By default, the user type is set to Standard User, which is always included in the count.

- A **Standard User** is always included and represent the people doing the work.
- A **Requester** is always excluded from the count if the corresponding integration is also excluded, since they are seen as outside users (submitting IT or help tickets, for example) and are not employees or subcontractors of your company.
- An **Excluded User** is always excluded regardless of integration.

To better understand which user types will be counted, refer to the Impact on Count column, which identifies if the user type is included based on the Integration and User Type Exclusions configured.

- **Counted** indicates that the user type will be counted in the user count.
- **Not Counted** indicates that the user type will not be counted in the user count.
- **Conditional (See details)** indicates that the user type will not be counted in integrations that were checked as involving ITSM in the Integration Exclusions tab. If an integration containing the user type was not marked as involving ITSM, it will be counted in the user count.
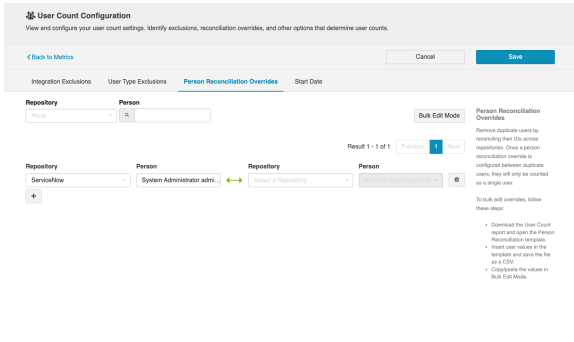
Person Reconciliation Overrides

In the **Person Reconciliation Overrides** tab, you can de-duplicate users by mapping their IDs across tools. This ensures that users are not counted more than once in your user count.
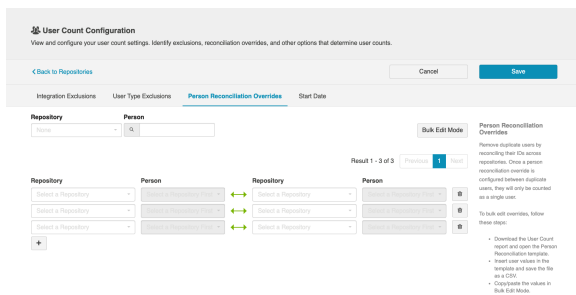
To configure a person reconciliation override, click the **+** icon.
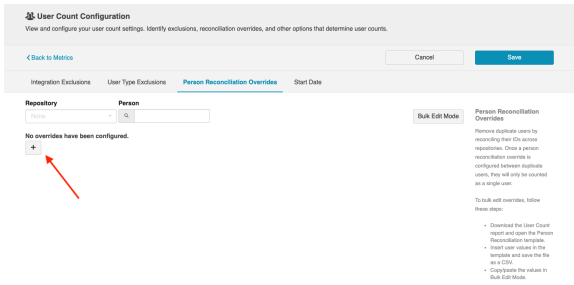
Next, select a repository and choose the user you'd like to reconcile.

Then, select the duplicate user in the other repository to complete the reconciliation override.
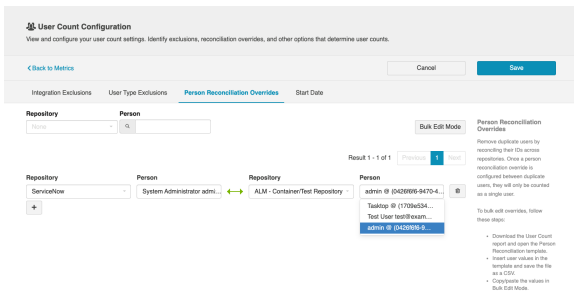
If multiple reconciliation overrides need to be configured at once, you can use Bulk Edit Mode.
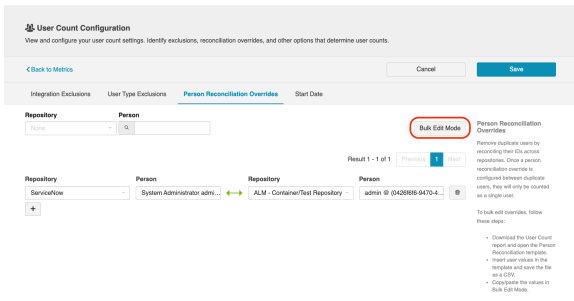
Before adding values in bulk edit mode, you'll first need to open the **1-raw-data.csv** file and **person-reconciliation-template.csv** file (both found in the User Count Report).
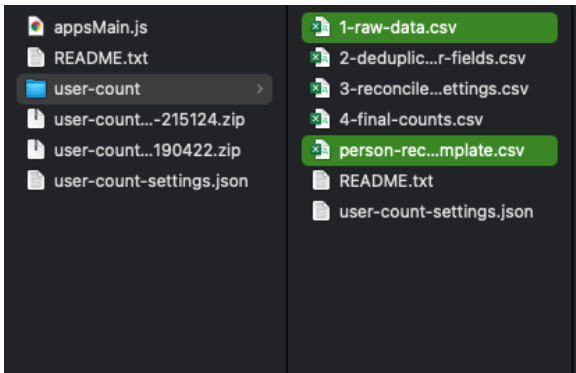
After opening the raw data file, copy the **Repository ID** and **Tasktop Generated Repository Person Identifier** for the values you'd like to reconcile.

Then, paste the values in the template and save the template as a CSV.

After saving the template, return to the Person Reconciliation Overrides tab and click **Bulk Edit Mode**.
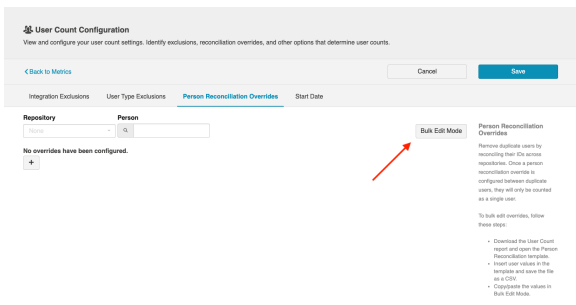
A pop-up will appear where you can copy and paste the values from the template.

Note: Do not include the column headers from the template when inserting the values.
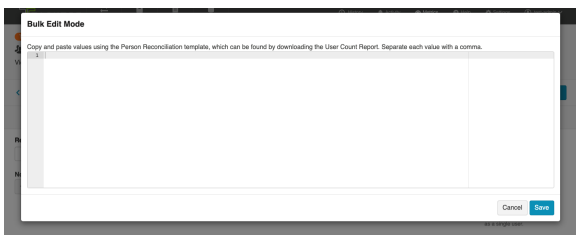
Once filled out, an entry might look like this:
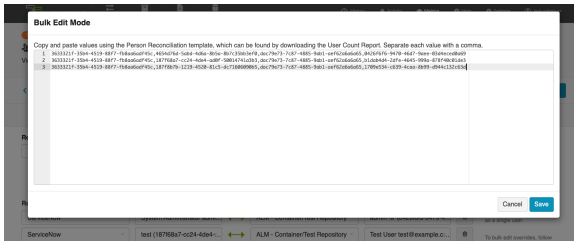
Start Date

In the **Start Date** tab, you can select the start date for the User Count report if your contract date differs from your license start date or expiry date. By default, the User Count report will be generated from the start of your most recent license renewal anniversary.

Basic Functionality

See the Tasktop Editions table to determine if your edition contains basic or advanced Metrics functionality.

Users with basic functionality will be able to view metrics showing the following:

- Total Artifacts Created
- Total Artifacts Updated
- Calculated ROI

Metrics below are displayed to show data for all integrations, over the last 30 days.

Note: Basic functionality does not include the Filter By and Date Range filters.
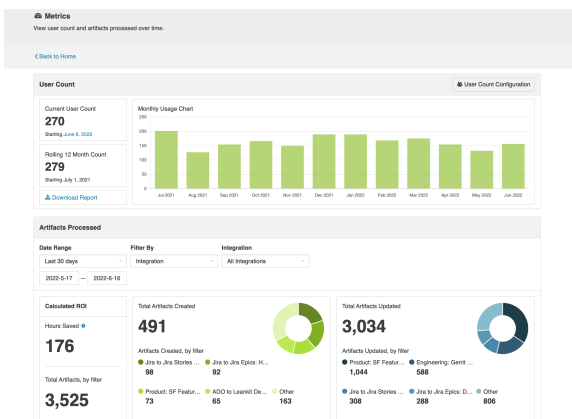
To help understand which artifacts are being synchronized, a pie chart will show the distribution for the metrics based on integration.
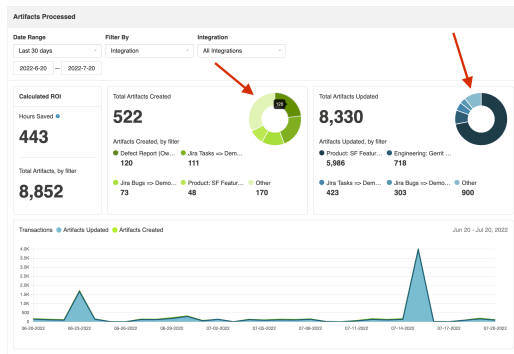
Calculated ROI

The **Calculated ROI** section helps you understand and communicate the direct cost savings enabled by Hub by quantifying the time and effort saved with Hub's model-based tool integration.

In the Calculated ROI section, you'll see two subsections, Hours Saved and Total Transactions.

- **Hours Saved** estimates the efficiency gains from synchronization calculated as total hours saved. This number is derived from an estimate of three minutes saved per transaction.
- **Total Artifacts** represents the total number of artifacts created and updated within the last 30 days.

Note: Basic functionality only displays calculated ROI for all integrations, over the last 30 days.
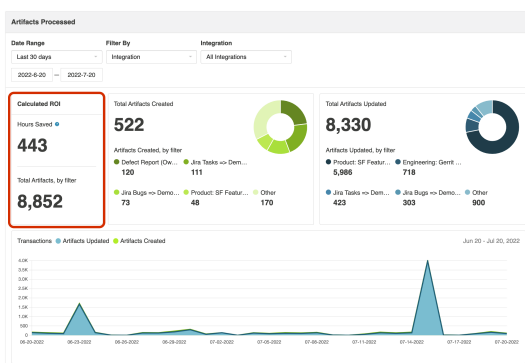
Advanced Functionality

See the Tasktop Editions table to determine if your edition contains basic or advanced Metrics functionality.

Users with advanced functionality will be able to view metrics showing the following:

- Total Artifacts Created
- Total Artifacts Updated
- Calculated ROI
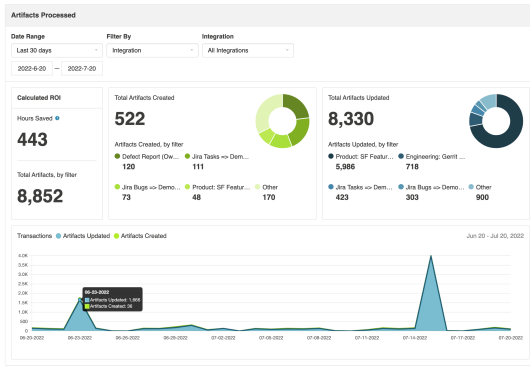
To help understand which artifact types are being synchronized, a pie chart will show the distribution for the metrics above based on model, integration, or repository.
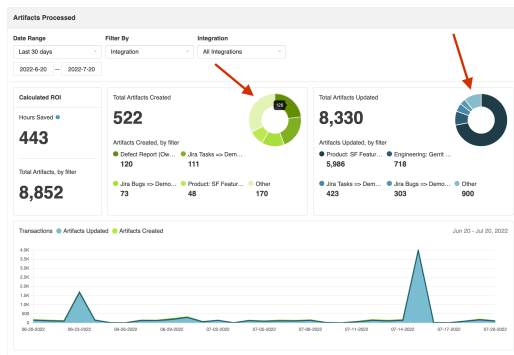
Additionally, users can choose to filter the data above based on

- Date Range
    - Last 7 Days
    - Last 30 Days
    - Last 90 Days
    - Custom Range
- Integration
- Repository
- Model
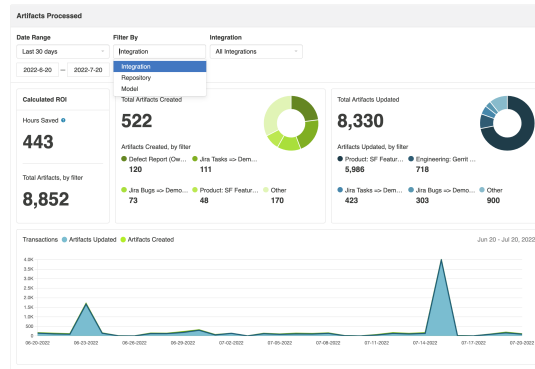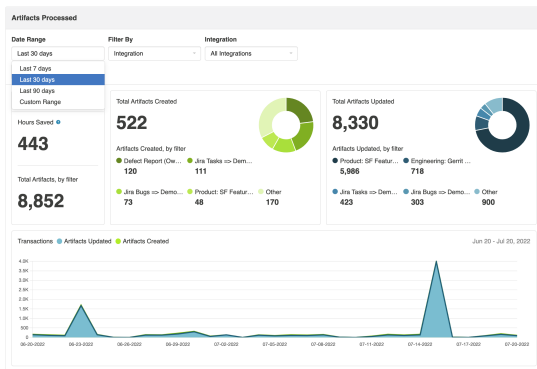
Calculated ROI

The **Calculated ROI** section helps you understand and communicate the direct cost savings enabled by Hub by quantifying the time and effort saved with Hub's model-based tool integration.

Calculated ROI contains two subsections, Hours Saved and Total Transactions.

- **Hours Saved** estimates the efficiency gains from synchronization calculated as total hours saved. This number is derived from an estimate of three minutes saved per transaction.
- **Total Artifacts, by filter** represents the total number of artifacts created and updated based on the filters applied.

Tip: You can use the filters to view ROI for a certain repository, integration, or model.
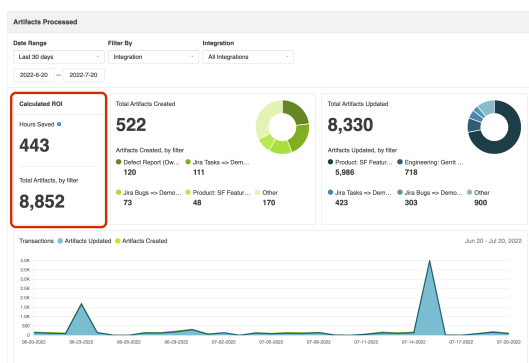
Cumulative Artifact Processing Counts

Users can also view tables showing cumulative totals for Artifacts Created, Artifacts Updated, and User IDs for each integration and repository.

The Artifacts Created and Artifacts Updated metrics show cumulative totals since installing Tasktop Integration Hub version 18.2.0.

The User ID metrics shows the number of unique user IDs on artifacts that have flowed through or been updated by Tasktop since installing Tasktop Integration Hub version 18.3.0. This metric can be used to better understand the value and scope of the integration, and is not intended to be used to assess Tasktop usage for licensing purposes (for licensing purposes, please see the User Counting section above).

PRODUCTION

Metrics
View user count and artifacts processed over time.

‹ Back to Home

User Count

User Count Configuration

Current User Count
182
Starting June 6, 2022

Rolling 12 Month Count
270
Starting August 1, 2021

⬇ Download Report

Monthly Usage Chart

Artifacts Processed

Date Range	Filter By	Integration
Last 30 days	Integration	All Integrations
2022-6-20 — 2022-7-20		

Calculated ROI
Hours Saved
443

Total Artifacts, by filter
8,852

Total Artifacts Created
522
Artifacts Created, by filter
- Defect Report (Ow... 120
- Jira Tasks -> Dem... 111
- Jira Bugs <=> Demo... 73
- Product: SF Featur... 48
- Other 170

Total Artifacts Updated
8,330
Artifacts Updated, by filter
- Product: SF Featur... 5,986
- Engineering: Gerrit ... 718
- Jira Tasks <=> Dem... 423
- Jira Bugs <=> Demo... 303
- Other 900

Transactions ● Artifacts Updated ● Artifacts Created   Jun 20 - Jul 20, 2022

Cumulative Artifact Processing Counts

Integrations

| Integration Name	Template	# Artifacts Created	# Artifact Updates	# User IDs
Defect Reporting	Work Item Synchronization	61	370	34
CS ProServ - UK to BPDC Generic Cards (Ow...	Work Item Synchronization	89	2,140	53
Engineering: Gerrit Changeset linking to Jira (...	Modify via Gateway	0	45,400	213
EngOps: Code Complete ChangeSet (Owner:...	Modify via Gateway	0	0	0
EngOps: IBM Release Work Item (Owner: Sy...	Create via Gateway	395	0	5
Defect Report (Owner: QA Group, Primary: S...	Enterprise Data Stream	4,434	0	190
Product: SF Feature Requests to TP (Owner:...	Work Item Synchronization	3,219	53,817	240
Product: Targetprocess Feature to JIRA Epic (...	Work Item Synchronization	678	13,003	202
Product: Targetprocess Requests to JIRA Tas...	Work Item Synchronization	342	1,561	85
Sales: Salesforce Cloud Instance Request Sy...	Work Item Synchronization	447	1,895	85
TEST - SF Highlighters -> Jira (improved notifi...	Work Item Synchronization	365	4,422	33
ADO <> Jira	Work Item Synchronization	14	44	8
TP User Story <=> Jira New Version Support (...	Work Item Synchronization	167	1,100	39
Jira Arch Stories <=> Demo Jira Arch Stories (...	Work Item Synchronization	457	1,461	37
Jira Bugs <> Demo Jira Bugs	Work Item Synchronization	2,521	10,478	155
Jira Debt <> Demo Jira Debt	Work Item Synchronization	1,074	3,319	101
Viz Demo: Tasktop Jira Epics <=> Demo Jira E...	Work Item Synchronization	696	2,019	91
Viz Demo: Tasktop Jira Risks <=> Demo Jira Ri...	Work Item Synchronization	390	1,732	77
Viz Demo: Tasktop Jira Stories <=> Demo Jira ...	Work Item Synchronization	2,432	9,088	157
Viz Demo: Tasktop Jira Tasks <=> Demo Jira T...	Work Item Synchronization	3,172	9,592	144

Repositories

Repository Name	# Artifacts Created	# Artifact Updates	# User IDs
Hub Defect Database	4,434	0	0
IBM DOORS	392	0	5
Jira	365	4,422	1
Planview LeanKit	181	882	3
Planview E1	0	758	67
Jira Align	10,742	37,887	7
Salesforce	125	38,493	316
Asana	1,662	51,969	1,662
ServiceNow	1,970	5,893	0
Azure DevOps	896,812	0	0
codebeamer	0	1	0
Targetprocess	3,304	30,722	488

Settings

Introduction

To access the **Settings** screen, click **Settings** in the upper right corner of your screen.
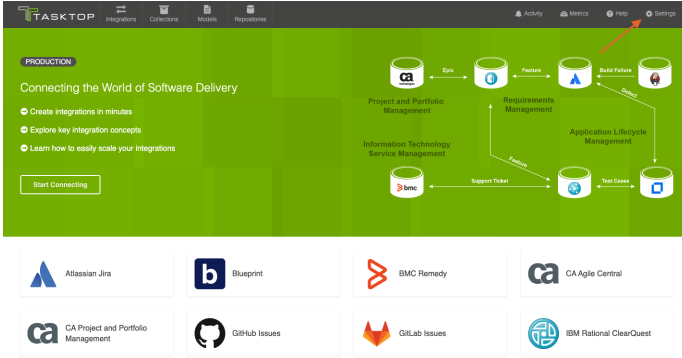
Under General (Settings), you can access:

- Configuration
- Master Password Configuration
- Storage Settings

Configuration

The Configuration section allows the administrator to set the change detection intervals of the connected repositories and to create a label for their Tasktop instance identifying the environment name and type (testing or production).

Learn more here.

Master Password Configuration

This feature is not applicable to Tasktop Cloud.

The Master Password is used to encrypt the credentials used in your repository connections and proxy settings, along with any other configuration values that are considered secret, which could include API keys and tokens.

Learn more here.

Storage Settings

This feature is not applicable to Tasktop Cloud.

Tasktop automatically stores operational data to a pre-configured Derby database. This is suitable for evaluation purposes only, and is not supported for production environments. Configuring Tasktop to utilize an external database will enable you to perform frequent back-ups without having to stop Tasktop Integration Hub, and ensure that your Tasktop Integration Hub practices are consistent with your existing disaster and recovery process.

Learn more here.

General (Settings)

Introduction

General (Settings) can be accessed by clicking the **General** tab on the **Settings** screen.
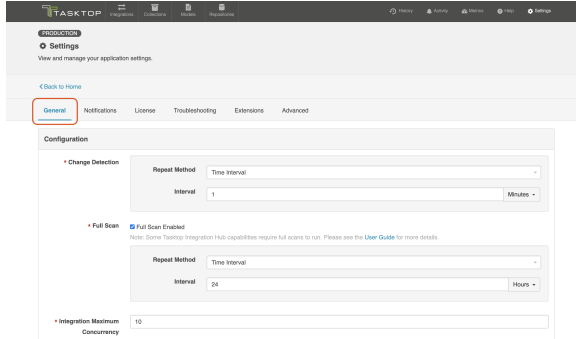
Under **General (Settings)**, you can access:

- Configuration
- Master Password Configuration
- Storage Settings

Configuration

The **Configuration** section allows the administrator to set the change detection intervals of the connected repositories and to create a label for their Tasktop instance identifying the environment name and type (i.e., testing or production).
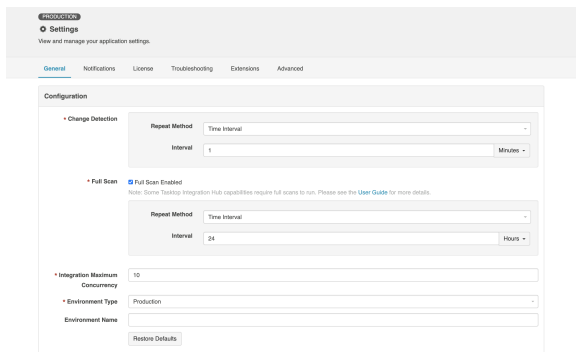
Change Detection

This is the method in which Tasktop detects changed artifacts in your external repositories.

You can configure repeat method to run on a time interval (e.g., 1 minute or 5 minutes) or an advanced schedule using cron expression (e.g., every 30 minutes from 9am-5pm from Monday to Friday).
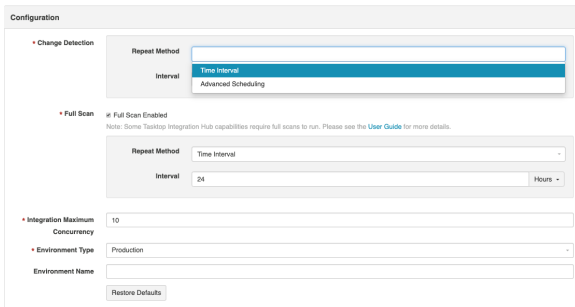
Environment Type and Name

Tasktop administrators can also set an environment type (testing or production) and name for their instance in the Configuration panel. This will create a label visible in the upper left corner of the screen while navigating throughout the Tasktop UI, to allow users to easily identify which Tasktop instance they are utilizing.
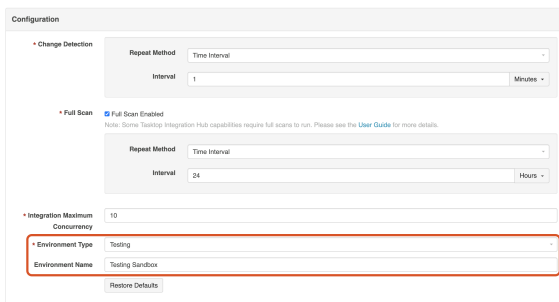
Once set, you will see the environment type and name label displayed in Tasktop.
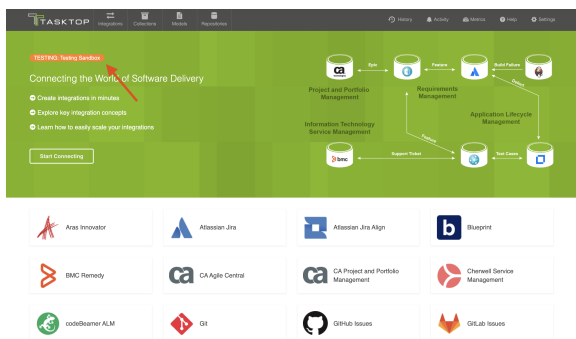
Master Password Configuration

This feature is not applicable to Tasktop Cloud.

After installation, you will be prompted to set a Master Password.
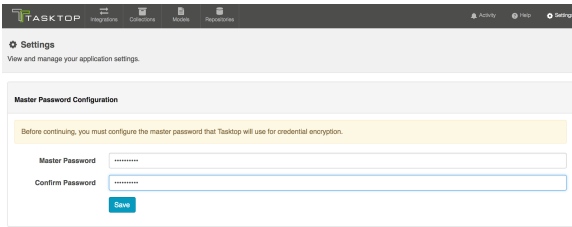
The Master Password is used to encrypt the credentials used in your repository connections and proxy settings, along with any other configuration values that are considered secret, which could include API keys and tokens.

💡 **Note**: 256 bit AES encryption is used.

Tasktop Integration Hub will automatically use the stored Master Password to decrypt repository credentials.

Normally you will not need to re-enter your Master Password. However, if the stored Master Password is missing, or if you'd like to change your Master Password from the General (Settings) screen, you will need to enter your current Master Password.

The Master Password is encrypted and stored separately from the encrypted repository credentials.

- On **Windows**, the encrypted Master Password is stored in the Windows Registry, encrypted using the Windows Data Protection (DPAPI).
- On **Linux**, the encrypted Master Password is stored in the Home Directory of the User running Tasktop Integration Hub.

If desired, you can change or reset the Master Password from the General (Settings) screen.

To do this, click **Change Master Password**.
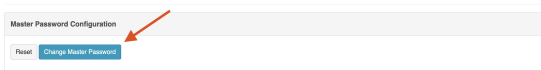
Enter your current Master Password and new Master Password.
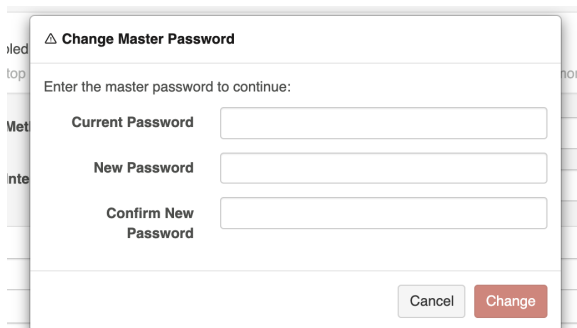
Click **Change** to update the Master Password.
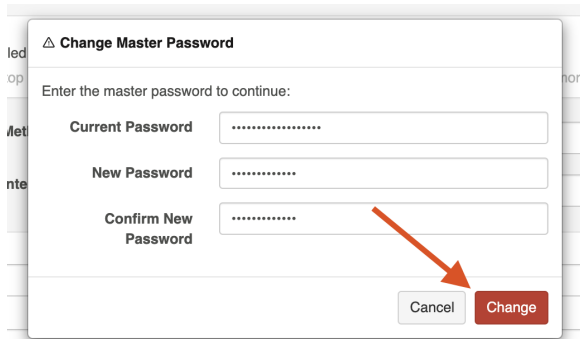
To reset your master password, click **Reset**.

💡 **Note**: If resetting the Master Password, you will not need to enter your current Master Password, but previously encrypted repository passwords will be lost, and must be provided after resetting.
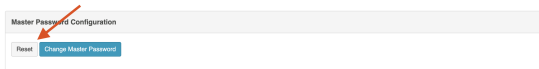
Storage Settings

This feature is not applicable to Tasktop Cloud.

Tasktop automatically stores operational data to a pre-configured Derby database. This is suitable for evaluation purposes only, and is **not** supported for production environments. Configuring Tasktop to utilize an external database will enable you to perform frequent back-ups without having to stop Tasktop Integration Hub, and ensure that your Tasktop Integration Hub practices are consistent with your existing disaster and recovery process.

💡 **Tip**: See our Hardware Requirements to determine which databases are supported for storing operational data.

Migrating Databases

To migrate your Tasktop operational data from the internal database to an external database, click **Use External Database**.
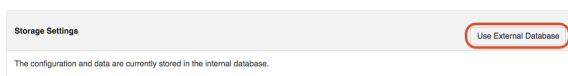
Next, click **Configure JDBC Driver** to select the JDBC driver for your database.
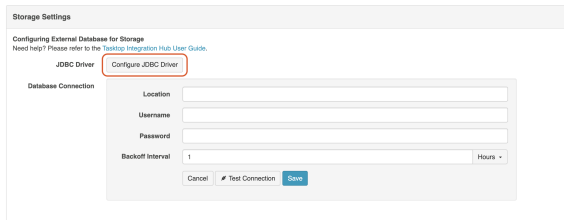
To download the JDBC driver:

The JDBC driver for Microsoft SQL Server can be downloaded from the Microsoft support site.

💡 **Note**: Tasktop currently supports the 9.4 version.

To upload the JDBC driver to Tasktop, a system administrator (a user with file system access to the machine that hosts Tasktop) must extract the ***.jar** file from the downloaded driver file and add the file to the designated directory:

- On **Windows**, the default folder is `C:\ProgramData\Tasktop\jdbc-drivers`
- On **Linux**, the `jdbc-drivers` folder can be found in the Tasktop installation directory

💡 **Note**: If needed, the user can change the location in which Tasktop looks for the files. This is done by changing the system property `jdbc.libraries.path`

Once the JDBC driver is uploaded, select it from the **Choose File** field on the Configure JDBC Driver pop-up.
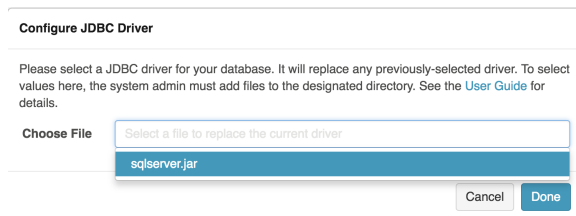
Next, fill out the Database Connection credentials — enter the location, username, and password.

⚠️ **Note**: Authentication credentials **must** be in SQL server authentication mode (i.e., mixed-mode with SQL credentials). Windows authentication mode is **not** supported.
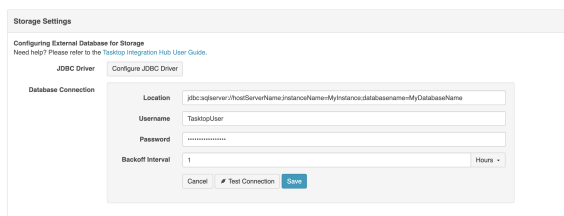
Location formats are as follows:

jdbc:sqlserver://hostServerName;instanceName=MyInstance;databasename=MyDatabaseName

If you'd like, you can also update the **Backoff Interval** setting.

The **Backoff Interval** is the time Tasktop will wait after a database connection failure (e.g., invalid username or password) before retrying the connection. This feature is especially useful for databases with a lockout or brute force policy configured.

⚠ **Note**: While backoff is in effect, processing artifacts will display an error and some operations may not work (e.g., you may be automatically redirected to the Settings screen). Once the backoff interval expires, artifacts will resume processing and operations will return to normal.

The backoff interval defaults to **one hour**, but can be customized as desired.

After you've added your database connection credentials, click **Test Connection** to confirm that your credentials have been accepted by Tasktop.

💡 **Note**: If backoff is in effect, the Test Connection button will continue to work so you can test and save updated credentials.

Once confirmed, click **Save**.
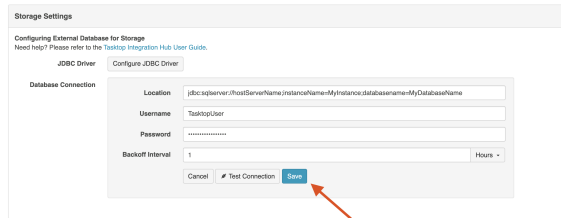
A warning message will appear telling you that you are about to transfer to an External Database. Review the entire message, **ensuring that you have performed the recommended data back-up.**

If you'd like to continue the transfer, click the checkbox and then click **Transfer**.
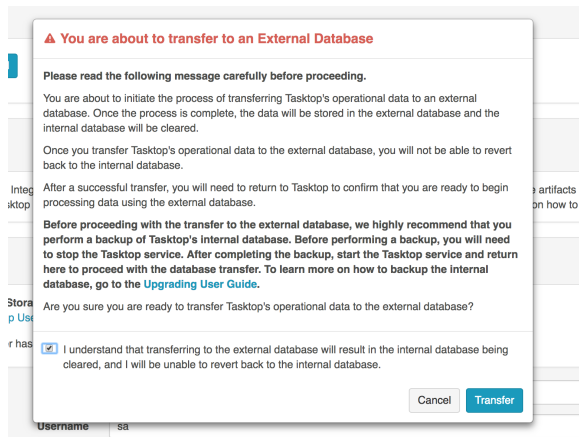
A **Database Transfer Completed** message will appear once the transfer is complete, informing you that your operational data has been successfully transferred from Tasktop's internal database to your own external database.
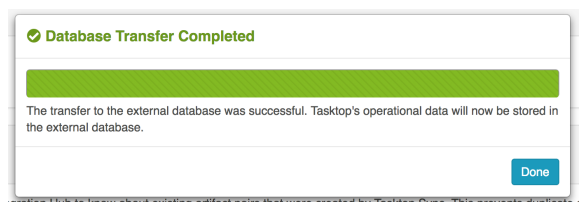
If your Database Transfer Fails or is Aborted

If your database transfer fails or is aborted, Tasktop will continue to use its internal database to store operational data. The internal database is not cleared until a successful transfer is completed, so you should not notice any change in performance.

However, we do recommend reviewing the external database and clearing any data and tables that were created as part of the failed data transfer before starting the transfer process again.

Overriding Database Access

In order to prevent risk of collisions, duplicates, and other errors, Tasktop has functionality to ensure that multiple Tasktop instances cannot run on the same operational database.

If you connect your instance to a database that is already in use by Tasktop (this is **not** recommended), upon start-up of the new instance, the prior instance will lose database access and stop processing events. When you login to the prior instance, you will see an error message prompting you to either update your credentials to connect to a different database, or to override database access. If you override database access, this means that the other instance of Tasktop will lose access to that database.

When overriding, be sure to confirm that no other Tasktop instance is using the database before moving forward. If another Tasktop instance is actively using the database, it is recommended that you shut down the other instance of Tasktop before proceeding.
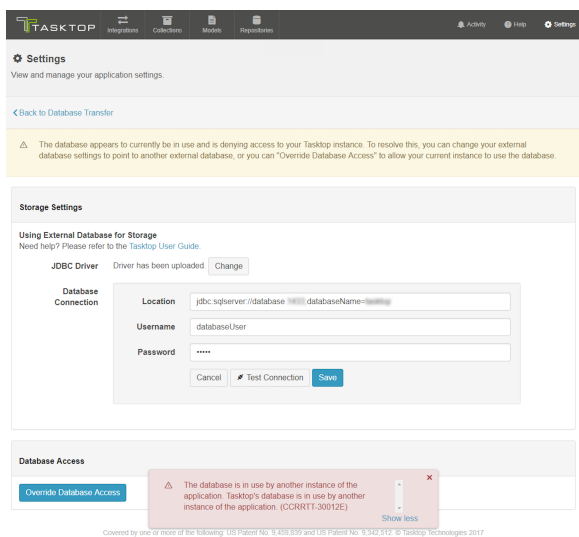
Notifications

Introduction

Notifications can be accessed by clicking the **Notifications** tab on the **Settings** screen.
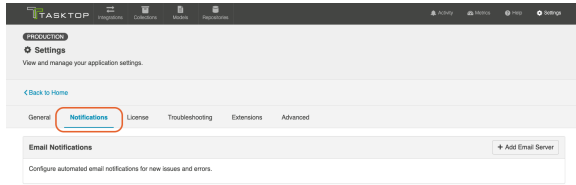
Under **Notifications**, you can access:

- Email Notifications

Email Notifications

To facilitate troubleshooting, you can configure automated email notifications for new errors and issues encountered in Tasktop.

Emails will contain a count of new issues and errors (excluding ignored errors) since the last email notification, and a link to the Activity screen to view the errors/issues. Emails will be sent only if a new error or issue occurs.
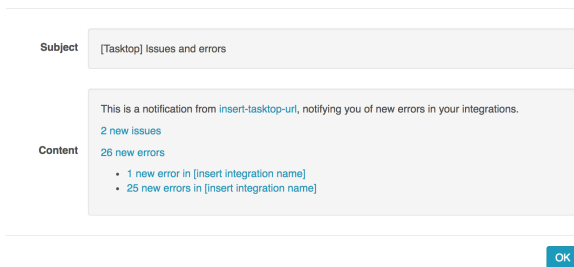
Configuring Email Notifications

To configure email notifications click **+Add Email Server**.
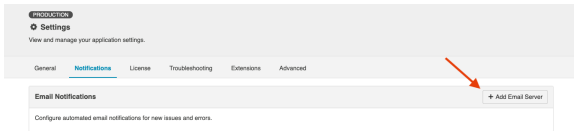
This will bring you to the **Email Notifications** screen.

The form requires that the following fields be filled out:

Basic Details

- **To Email Addresses**: The email addresses that will receive the notifications. You can add multiple email addresses to this field.
- **From Email Address**: The email address listed in the 'sender address' (or 'from') field of notification emails sent by Tasktop. In many cases, this will match the email whose settings are configured in the 'Email Server Settings' section below, though a different email (such as no-reply@email.com) can be configured here. If a user were to hit 'reply' on an email notification, this is the email the reply would be sent to.
- **Subject Prefix (optional)**: The prefix appended to the subject line of the Error Notification emails. This defaults to [Tasktop] but can be edited if needed. This can help users filter email notifications from Tasktop based on prefix.
- **Tasktop Server URL**: The URL used to access your instance of Tasktop. This is used to construct links to errors and issues in the notification emails.
- **Notification Frequency**: The frequency of email notifications. Emails will contain a count of new issues and errors since the last notification and will only send if a new error or issue has occurred since the prior email.

Email Server Settings

These are the email server settings that allow Tasktop to send notifications.

- **Username (optional)**: Username for the authenticated SMTP server.
- **Password (optional)**: Password for the authenticated SMTP server.
- **SMTP Server**: The SMTP host name of your mail server.
- **SMTP Port**: The SMTP Port number to use.
    - If Protocol = SMTP, the value for this will typically be 25.
    - If Protocol = SMTPS, the value for this will typically be 465.
    - If Protocol = SMTP_STARTTLS, the value for this will typically be 587, but can also be port 25.
- **Connection Timeout**: Specifies the maximum period, in seconds, that establishing an email server connection is permitted to take. This defaults to 60 seconds, which should cover most scenarios.
- **Protocol**
    - **SMTP**: Basic unencrypted SMTP Protocol.

160

- **SMTPS**: A more advanced, encrypted SMTP Protocol (SMTP Secure), which will perform server certificate validation.
- **SMTP_STARTTLS**: A modern protocol that wraps the unencrypted SMTP protocol in TLS (formerly known as SSL encryption), and will perform server certificate validation. This will attempt the STARTTLS wrapper, but if it is not supported by the server, the client will fall back to basic SMTP.

💡 **Note**: Google email users should select SMTP_STARTTLS.

Here's an example of a filled in form:

You can test your email server settings by clicking **Test Connection**.

Or, send a test email by clicking **Send Test Email**.

Once settings are filled in and the connection has been tested, click **Save** to save your settings.

Click **Turn On Notifications** to enable email notifications.

Once saved, you can turn email notifications on or off and delete the notification settings from the Notifications screen. You can also click **Configure Notification Settings** to modify your existing settings:
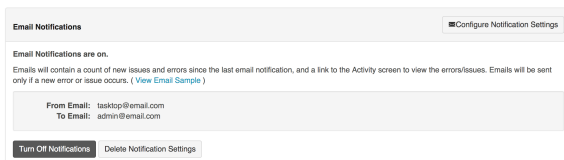
💡 **Note**: If an email notification fails, an issue will be surfaced on the Activity screen in Tasktop.

License

Introduction

You can access your License details by clicking the **License** tab on the **Settings** screen.
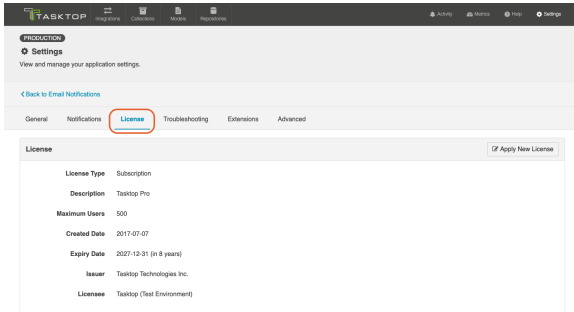
Under **License**, you can access:

- License

License

This feature is not applicable to Tasktop Cloud.

A license is required to run the application. Upon initial log-in, you will see that your product is currently un-licensed:
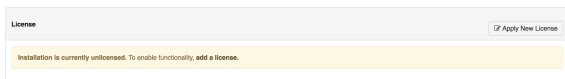
Click **Apply New License** to enter your license.

The Master Password must be set and the License must be entered before the application can be used.

On the License panel, you will see the following information:

- License Type
- Description
- Maximum Users
- Created Date
- Expiration Date
- Issuer
- Licensee
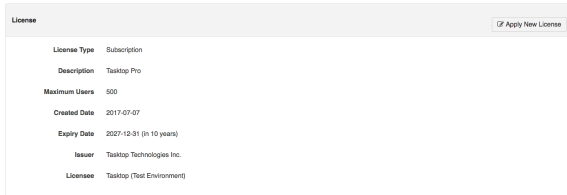
You will also see a warning if your license is expired:
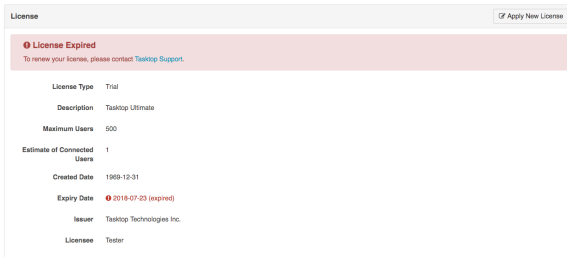
Should your license expire, in addition to seeing a warning on the License screen, you'll also see that an issue is surfaced on the Activity screen:
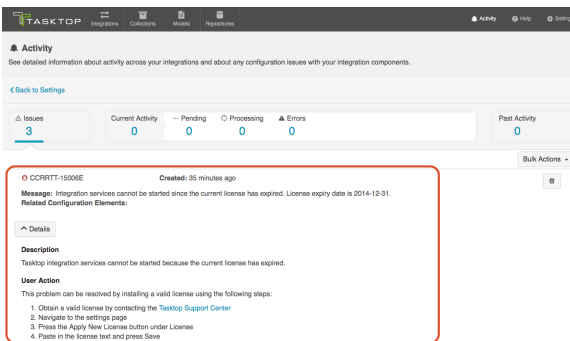
When your license is expired, you can still navigate within the Tasktop UI, but your integrations will be stopped from running. Note that though they will still display the **Run** or **Stopped** state they were in at the time your license expired, no artifacts will process in an integration until a new license is applied.

💡 **Note**: Please consult your license agreement or contact your account representative if you have any questions about your license settings or usage policy.

Troubleshooting (Settings)

Introduction

Troubleshooting (Settings) can be accessed by clicking the **Troubleshooting** tab on the **Settings** screen.
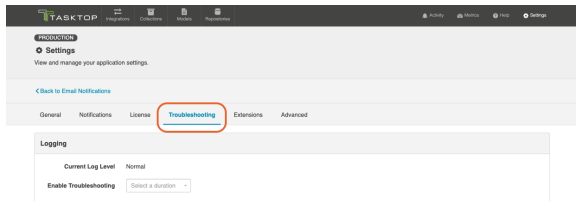
Under **Troubleshooting**, you can access:

- Logging

Logging

For troubleshooting purposes, Tasktop logs various events that the application performs.

There are two logging levels available:

This type of logging is sufficient for most scenarios.

Updating the logging levels immediately changes the logging granularity. Tasktop does not need to be restarted for the change to take effect.
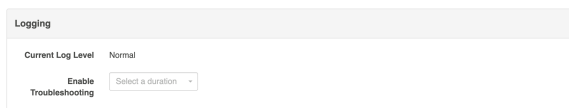
Downloading Logs

Please reference the Troubleshooting page for instructions on downloading the logs as part of the Support and Usage Report.

Extensions (Settings)

Introduction

Extensions can be accessed by clicking the **Extensions** tab on the **Settings** screen.
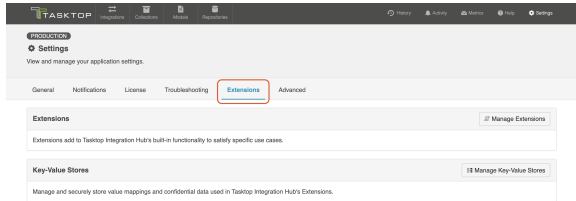
Under **Extensions**, you can access:

- Extensions
- Key-Value Stores

Extensions

Extensions add to Tasktop's basic functionality by facilitating processes such as custom data transformations, payload transformations, advanced person reconciliation, and state transitions.

💡 **Note**: Extensions are written in JavaScript, or more specifically ECMAScript.

Custom Data Transformation Extensions enable you to map fields to one another which do not have out-of-the-box transforms, and to create custom transforms for comments.

You can apply this extension when updating your transform on the Field Configuration screen.
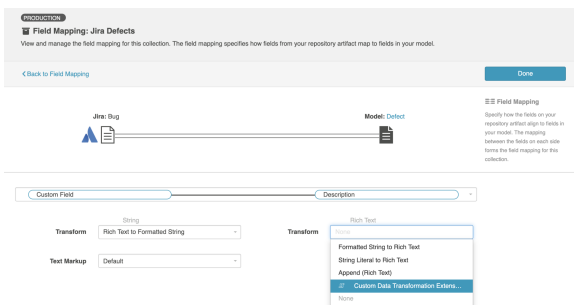
Creating a New Extension

You can create and save custom extensions for use in your integrations on the **Extensions** screen. Extensions are created with a name and optional description so that they can be centrally managed and reused if needed.

💡 **Note**: Fields that are not mapped to the model are not retrieved by Tasktop, and therefore are not available to be used in an extension. If fields are needed for scripting purposes, please map those fields to the model.

To create and edit your extensions, click **Manage Extensions**.
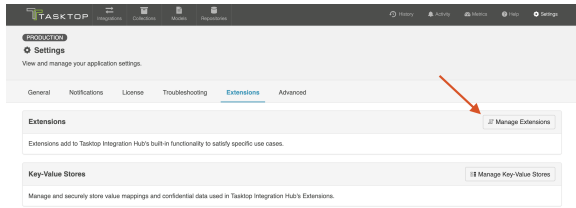
You can copy an existing extension by clicking **Copy** from the Extensions list.
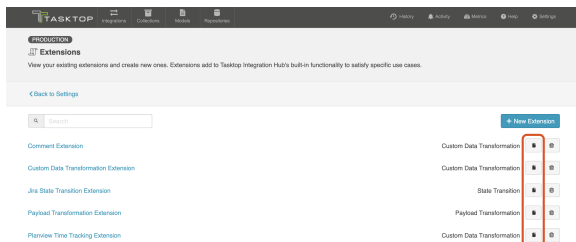
Click **New Extension** to create and customize an extension.
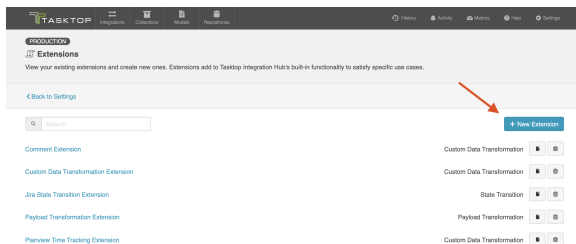
Then, select one of the following extension types: **Custom Data Transformation**, **Payload Transformation**, **Person Reconciliation**, **State Transition**.
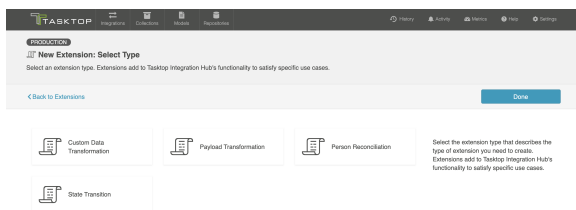
After choosing your extension type and selecting the collection or model, you can view the schemas of the artifact, comment, or person in the Artifact Schema Information section.

This section provides you with useful schema information when composing extensions.
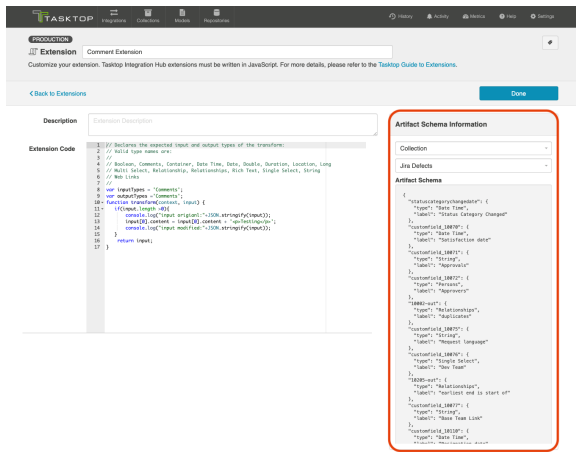
Viewing Associated Configuration Elements

To view associated configuration elements (such as collections or integrations that utilize the extension you are viewing), click the **Associated Elements** tag in the upper right corner of the screen.
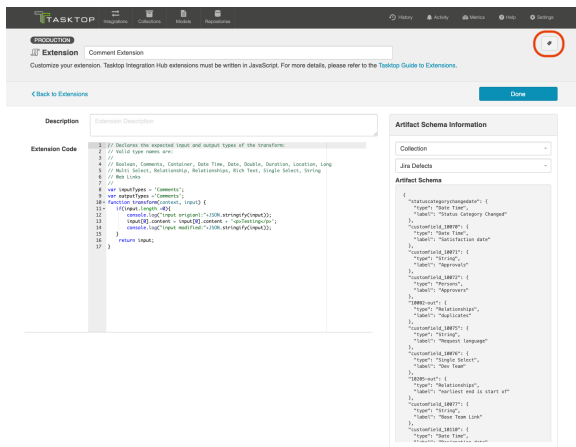
Key-Value Stores

See Tasktop Editions table to determine if your edition contains Key-Value Store functionality.

Key-Value Stores enable you to securely store and manage sensitive data and value mappings. Using key-value stores instead of inlining the data in the extensions reduces the size, complexity, and maintenance of extensions.

Creating a New Key-Value Store

💡 **Note**: Access via the provided JavaScript API is read-only. Stores can only be created, updated, and deleted using the user-interface and import functionality.

To create a new key-value store, navigate to the **Extensions** tab and click **Manage Key-Value Stores**.
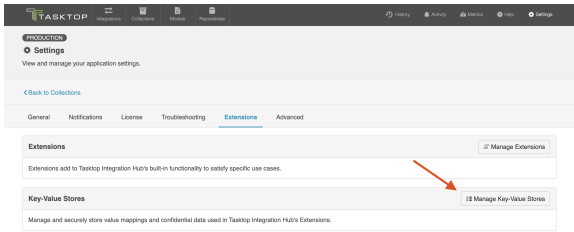
Click **+ New Key-Value Store.**
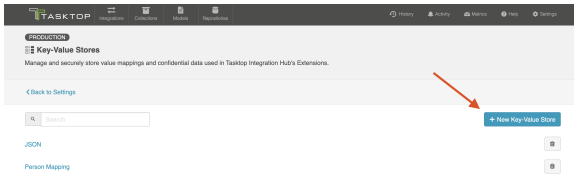
Select the Key-Value store type that best suits your use-case.

- **Confidential:** Enables you to encrypt your key-value pairs.
- **Plain:** Enables you to store your key-value pairs in plain text.
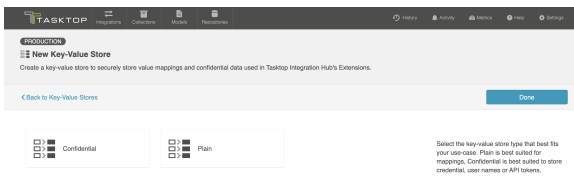
After you have selected the Key-Value store type, you can add each key-value pair individually, or you can import key-value pairs using a **.csv** or **.json** file.

💡 **Note**: Keys in a key-value store are case-sensitive and must match exactly.
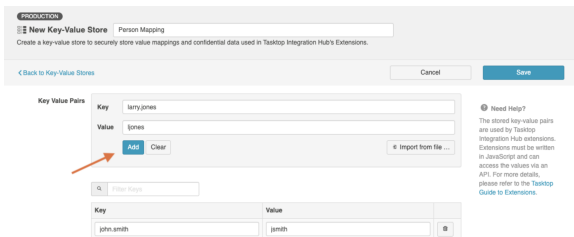
Once you have finished adding your key-value pairs, click **Save** and **Done** to save your changes.

💡 **Tip**: Clicking **ctrl+s** on Windows and **cmd+s** on macOS will save your key-value store.
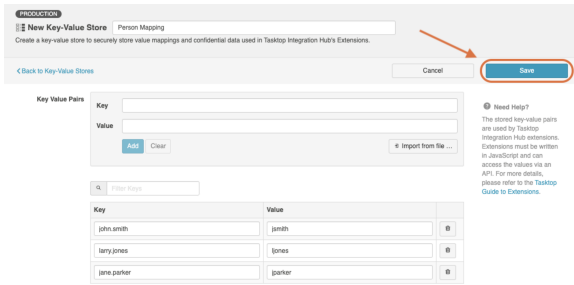
Importing Files to a Key-Value Store

Tasktop Integration Hub allows you to import key-value pairs to your key-value store using **.csv** or **.json** files.

To import your key-value pairs, click **Import from file** and select the **.json** or **.csv** file you'd like to import.
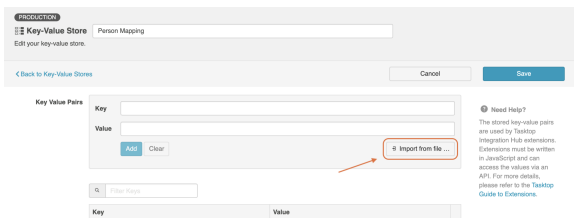
To ensure that your .json or .csv files are imported successfully, please use the following format:

```
{
"jsmith@email.com":"John Smith",
"ljones@email.com":"Larry Jones",
"mbrown@email.com":"Mary Brown"
}
```

⚠ **Note**: If your .json file contains a duplicate key, an error will appear upon importing.

After you have selected the file you'd like to import and it has been imported successfully, the key-value pairs will be displayed in your key-value store.
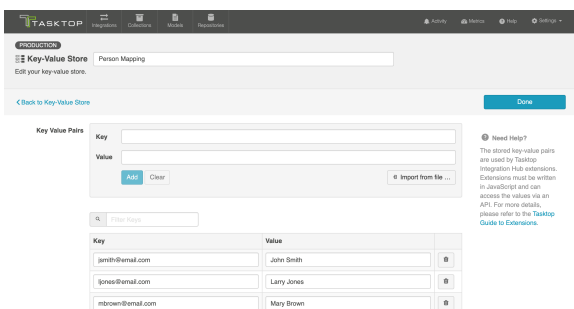
Accessing Individual Values of a Key-Value Store

Tasktop Integration Hub provides a JavaScript API to access the values of a store. All types of key-value stores allow direct access to values.

The API is used as follows:

```
var tokenValue = store.retrieveValue('EngOps Credentials', 'Build Server API token');
```

Accessing All Pairs of a Key-Value Store

The provided JavaScript API allows the retrieval of all pairs of a store. This is especially useful when using the stored pairs as a lookup table.

The API is used as follows:

```
var pairs = store.retrievePairs('Project to Product mapping');
```

Each store is limited to 3000 key-value pairs. For lookup and other purposes, the pairs of multiple stores can be joined in the extension as follows:

```
var pairs = store.retrievePairs('Project to Product mapping');
var morePairs = store.retrievePairs('Value-Stream mapping');

var merged = Object.assign(pairs, morePairs);
```

💡 **Note**: Confidential key-value stores do not allow access to all pairs and can only be accessed by providing the key for individual values.

Key-Value Store API Reference

- `store` - The globally-visible object providing the key-value store API.
- `store.retrievePairs(store)` - Provides an object with all the keys as properties and their values as strings. If the store cannot be found, a 'NotFoundException' is thrown.
- `store.retrieveValue(store, key)` - Provides the string value associated to the specified key. If the store or key cannot be found, a 'NotFoundException' is thrown.

Technical Guide to Extensions

Extensions add to Tasktop's built-in functionality to satisfy specific use cases, such as:

- Performing state transitions incorporating business logic
- Enabling custom data transformations between fields
- Defining person reconciliation strategies between repositories
- Transforming payloads sent to Gateway collections into a format Tasktop can accept

In the following sections, you will find technical implementation details about each extensions type, example extensions, troubleshooting extensions, and how to access web resources and object properties.

State Transitions

State Transitions are used to transition an artifact from one status to another. To illustrate, we use the fictitious example of an artifact of type Defect with the following status values:

- New
- In Progress
- Resolved
- Closed

The status of a Defect cannot be modified directly. In this example, to move a defect from status **New** t o **In Progress**, the **Start Progress** transition is used.

Sometimes multiple status transitions are required. For example, to move a defect from **New** to **Closed**, the following transitions are used in sequence **Start Progress**, **Resolve**, **Close**.

The following diagram shows how state transitions are used to move a defect from one status to another:
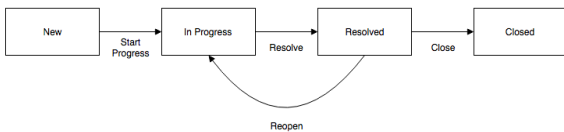
Configuring State Transitions with Extensions

To perform state transitions, an extension can be used. Add a state transition extension from the Extensions screen, accessible from Settings. Once added, the extension can be applied from the State Transition sash on the Collection Configuration screen.

💡 **Note**: Tasktop also provides functionality to configure state transitions using a transition graph. The transition graph is the recommended strategy, as it allows you to configure the state transitions directly within Tasktop's UI.

Authoring State Transition Extensions

State transition extensions are defined by a single function:

```
function transitionArtifact(context,transitions)
```

This function can return a single transition.

For a given artifact, the extension may be called multiple times. Each time the extension is called, the transition that it returns is performed. State transition extensions are called repeatedly until they return undefined, indicating that no more transitions are needed.

To prevent errors, extensions are not called again if they cause an artifact to transition to the same status more than once.

A simple state transition extension could look something like this:

```
function transitionArtifact(context,transitions) {

    if (context.sourceArtifact.status === 'Resolved' && context.targetRepositoryArtifact.status !== 'Resolved')
{

        var transition = findTransitionWithLabel(transitions,'Resolve');
        transition.attributes.resolution = 'Fixed';

        return transition;

    }

}

function findTransitionWithLabel(transitions, label) {

    for each(var transition in transitions) {

        if (transition.label === label) {

            return transition;

        }

    }

}
```

Two parameters are passed to the `transitionArtifact` function:

- `context` – A context object that provides state that the extension can use to determine which transitions are needed
  - `context.sourceArtifact` – A JavaScript object representation of the source artifact, whose structure matches the model configured in the integration
  - `context.targetRepositoryArtifact` – A JavaScript object representation of the target artifact, whose structure matches the structure of the artifact in the repository
- `transitions` – An array of transition objects

Below is an example of a `context` with a target artifact from Jira:

```
{
    "sourceArtifact": {
        "summary": "a summary value",
        "priority": "Critical",
        "status": "Done"
    },
    "targetRepositoryArtifact": {
        "issuetype": "Bug",
        "components": null,
        "timespent": null,
        "formattedid": "TPC-144",
        "timeoriginalestimate": null,
        "project": "Test Project C",
        "description": null,
        "fixVersions": null,
        "resolution": null,
        "customfield_11500": null,
        "api-id": "JIRA",
        "attachment": null,
        "resolutiondate": null,
        "id": 14400,
        "summary": "a summary value",
        "watches": null,
        "created": "2016-09-23T15:22:20.000+0000",
        "$closed": false,
```

```
        "reporter": "*****",
        "priority": "Critical",
        "labels": null,
        "revision": null,
        "customfield_11601": null,
        "customfield_11600": null,
        "customfield_11501": null,
        "environment": null,
        "customfield_11504": null,
        "customfield_11602": null,
        "timeestimate": null,
        "versions": null,
        "duedate": null,
        "web-links": null,
        "location": "http://jira.example.com/browse/TPC-144",
        "assignee": null,
        "worklog": null,
        "updated": "2016-09-23T15:22:20.000+0000",
        "status": "To Do"
    }
}
```

Each transition object in the array appears as follows:

```
 {
  id: 'an-id',
  label: 'A Label'
  attributes: {
    first-attribute: null,
    ...
  }
}
```

For example, transitions corresponding to the Jira artifact example above are as follows:

```
[{
    "attributes": {
        "project": "Test Project C",
        "issuetype": "Bug"
    },
    "id": "11",
    "label": "To Do"
}, {
    "attributes": {
        "project": "Test Project C",
        "issuetype": "Bug"
    },
    "id": "21",
    "label": "In Progress"
}, {
    "attributes": {
        "project": "Test Project C",
        "issuetype": "Bug"
    },
    "id": "31",
    "label": "Done"
}]
```

Attributes of a transition are values that may be set when performing the transition. Attributes should not be set unless needed or required.

The available attributes and whether or not they are required will vary depending on the type of repository of the collection.

Payload Transformations

Gateway collections can accept a JSON payload via HTTP, enabling clients to use a REST API to publish artifacts in Tasktop.

Without further configuration, Gateway Collections require a JSON payload that matches the model of the collection.

By configuring a Gateway Collection with an extension, it is possible to accept arbitrarily complex JSON payloads, enabling integration with third party products that integrate with webhooks.

Examples of such third party webhook notifiers include:

- Jenkins Notification Plugin
- Microsoft VisualStudio Web Hooks
- GitHub Webhooks

Configuring Gateway Collections with Extensions

To configure a Gateway Collection with an Extension, add a payload transformation extension from the Extensions screen, accessible from Settings. Once added, the extension can be referenced from the Gateway Collection screen.

Authoring Payload Transformation Extensions

Payload transformation extensions are defined by a single function:

```
function transformPayload(payload)
```

The function must return an array of 0 or more JSON objects matching the model of the gateway collection.

Given a model representing build jobs with the following fields:

- `created` – A date signifying the creation date
- `summary` – A brief one-line description
- `status` – A single-select indicating the build status

A simple payload transformation extension could look something like this:

```
function transformPayload(payload) {
  var createdTimestamp = new Date(payload.build.completion_time).toISOString();
  var created = createdTimestamp.substring(0,createdTimestamp.indexOf('T'));
  return [
    {
      'created': created,
      'summary': payload.name + ': '+payload.build.full_url,
      'status': payload.status
    }
  ];
}
```

The example above corresponds to the payload provided by the Jenkins Notification plugin, which provides JSON payloads as follows:

```
{
        "name": "Robot Lawnmower",
        "url": "job/Robot%20Lawnmower/",
        "build":
        {
                "full_url": "http://build.example.com:8081/job/Robot%20Lawnmower/4/",
                "number": 4.0,
                "phase": "COMPLETED",
                "status": "FAILURE",
                "url": "job/Robot%20Lawnmower/4/",
                "scm":
                {

                },
                "causes":
                [
                        "Started by user admin"
                ],
                "duration_string": "9 ms",
                "completion_time": 1.476313762942E12,
                "failing_since_build":
                {
                        "full_url": "http://build.example.com:8081/job/Robot%20Lawnmower/1/",
                        "number": 1.0,
                        "change_set":
                        [

                        ],
                        "completion_time": 1.47631304791E12,
                        "failing_since_time": "11 min"
                }
        }
}
```

Ignoring Webhook Payloads

For cases where the gateway collection is called and no corresponding action should be performed, the extension should return a 0-length array:

```
function transformPayload(payload) {
  ...

  if (nothingToDo) {
    return [];
  }
  ...
}
```

Creating Multiple Artifacts From A Single Webhook Payload

There may be cases when multiple artifacts should be created from a single webhook payload depending on the use case. For example, a GitHub PushEvent can contain multiple commits. To link each commit to an artifact separately, a payload transformation extension would be used as follows:

```
function transformPayload(payload) {
  var gatewayPayloads = [];
  for each (var commit in payload.commits) {
    gatewayPayloads.push(createCommitPayload(commit));
  }
  return gatewayPayloads;
}
```

Query Parameters and HTTP headers (optional)

Payload transformations can take two additional parameters — query parameters and HTTP headers:

- **Query Parameters**: Provides the query parameters sent with the payload as a JavaScript object with property names corresponding to parameter names and values as arrays of values of the corresponding query parameter
- **HTTP Headers**: Provides the HTTP request headers sent with the payload as a JavaScript object with property names corresponding to HTTP header names and values as arrays of values of the corresponding HTTP header

This might appear as follows:

```
function transformPayload(payload, parameters, headers)
```

Custom Data Transformations

In cases where specialized value transformations are needed for use in field mappings, such transformations can be added as **custom data transformation** extensions.

⚠️ **Note**: When using a custom data transformation, we recommend **not** mapping an extension that **loses** information from **collection to model**. Because the model is used for change detection, if information is lost, change detection may fail. For example, an extension that transforms a list of links into a single link should not be mapped from **collection to model**, as subsequent change detection would only have a single link to compare rather than the full list. We recommend keeping as much information as possible in the model and mapping any lossy transformations from **model to collection**.

The Context Object

The context object provides information that the extension can use to determine which transformations are needed.

For a custom data transformation, use the following:

- `context.sourceArtifact:` A JavaScript object representation of the source artifact
  - If you are mapping from model to repository, this will match the structure of the artifact in the model
  - If you are mapping from repository to model, this will match the structure of the artifact in the repository
- `context.targetArtifact:` A JavaScript object representation of the target artifact
  - If you are mapping from model to repository, this will match the structure of the artifact in the repository
  - If you are mapping from repository to model, this will match the structure of the artifact in the model

💡 **Note**: If existing scripts are utilizing `targetRepositoryArtifact` instead of `targetArtifact`, they will continue to work.

If you are creating a custom data transformation extension for test steps, use:

- `context.sourceTestStep` to access the source test step
  - If you are mapping from model to repository, this will match the structure of the artifact in the model
  - If you are mapping from repository to model, this will match the structure of the artifact in the repository
- `context.targetTestStep` to access the target test step
  - If you are mapping from model to repository, this will match the structure of the artifact in the repository
  - If you are mapping from repository to model, this will match the structure of the artifact in the model

See Tasktop Editions to determine if your edition contains Test Step functionality

The context parameter also has field properties:

- `context.sourceField`: If processing a single field
- `context.sourceFields`: A list of field objects, if processing more than one field
- `context.targetField`: If processing a single field
- `context.targetFields`: A list of field objects, if processing more than one field

A field object only has two properties: ID, and label:

```
{
    id: "assignee",
    label: "Assignee"
}
```

Creating a Custom Data Transformation Extension

Custom data transformation extensions are created from the Extensions screen, accessible from Settings. Created extensions can be selected when configuring a field mapping of a collection.

Custom data transformation extensions appear as follows:

```
var inputTypes = 'String';
var outputTypes = 'String';

function transform(context, input) {
        // returns the transformation result
}
```

All custom data transformation extensions must declare their input and output types as shown in the example above. Transformations are only available for a field mapping if the input types and output types match the fields selected in the mapping. In the case of a mapping with multiple source and target fields, the order of the declared input and output types must match the order of the source and target fields.

A simple split-and-trim value custom data transformation extension could look like this:

```
var inputTypes = 'String';
var outputTypes = ['String', 'String'];

function transform(context, input) {
    if (input) {
        var values = input.split('/');
        if (values.length != 2) {
            throw 'Unexpected value ' + input;
        }
        return values.map(function(s) {
            return s.trim();
        });
    }
}
```

Single Select and Multi Select in Custom Data Transformation Extensions

Single Select and Multi Select values are specified using their labels. Extensions that accept a Single Select as the input type will receive a string containing the option's label. Extensions that specify a Single Select as the output type should return a string containing the option's label.

To specify the empty option, return `undefined` from the extensions instead of a value. Extensions that accept a Multi Select as the input type will receive an array of strings of the option labels. Extensions that specify a Multi Select as the output type should return an array of strings with the option labels or an empty array to specify no options.

If the field has options, field options are available on the field of the context passed into an extension. For example, the options can be accessed with something like this:

```
context.sourceField.options
context.targetField.options
```

Rich Text Support in Custom Data Transformation Extensions

To perform Rich Text transformations, **Rich Text** must be declared as input or output types of the extension.
A Rich Text input parameter is passed as a valid HTML string.

For Rich Text as output type, the extension is expected to return a valid HTML string.

To escape HTML characters, the following function is provided:

```
html.escape(string)
```

A simple String-to-Rich-Text value transformation could look like this:

```
var inputTypes = 'String';
var outputTypes = 'Rich Text';

function transform(context, input) {
    if (input) {
        return '<pre>' + html.escape(input) + '</pre>';
    }
}
```

Web Links in Custom Data Transformation Extensions

To perform a web links transformation, web links must be declared as the input or output types of the extension. A web links field consists of a list of web link objects. A web link object consists of a location and other attributes.

The following is an example of a web link output:

```
[
    {
        label: 'Tasktop',
        location: 'http://www.tasktop.com'
    },
    {
        location: 'http://www.alt-tasktop.com'
    }
]
```

💡 **Note**: The label attribute is optional and if specified will be used to populate the label of the web link.

Relationships in Custom Data Transformation Extensions

Tasktop provides a JavaScript API for working with relationship fields. This API can retrieve, search, and get associated artifacts for artifacts.

Artifact Service API Reference

Artifacts returned from the Artifact API are the raw JSON representation of a Repository's Artifact. These representations may include internal IDs and other fields not mapped to the model. It may be necessary to manually interpret the results of these calls on a per-repository basis to determine the exact information that is returned.

- `artifacts.retrieveArtifact(relationship):Artifact` - Retrieves the artifact for the provided relationship
- `artifacts.listSearchTypes():SearchType[]` - Lists the valid search types for the targeted repository
- `artifacts.getSearchDefinition(searchTypeId):SearchDefinition` - Returns an object with the parameters that are required for the given search type id
- `artifacts.search(searchType, searchDefinition):Relationship[]` - Searches the target repository with the given search type id and search definition, returns a list of relationships which then can be looked up via `artifacts.retrieveArtifact (relationship)` to retrieve the artifact
- `artifacts.getFormattedIdSearchDefinition():SearchDefinition` - Returns an object with the parameters that are required for a formatted ID search
- `artifacts.searchByFormattedId(searchDefinition):Relationship[]` - Searches by formatted ID with the provided search definition and returns a list of relationships which then can be looked up via `artifacts.retrieveArtifact (relationship)` to retrieve the artifact
- `artifacts.toContainer(relationship, summary):Container` - Converts a relationship into a container, summary is optional

- `artifacts.toRelationship(container):Relationship` - Converts a container into a relationship
- `artifacts.getAssociatedRelationship(relationship):Relationship` - Finds the associated relationship for the given relationship. When mapping from model to collection the input value and source artifact relationship field values are from the source repository and must be converted to their associated value to be used in the target system. An exception is thrown if no artifact is found or multiple artifacts are found.
- `artifacts.getAssociatedContainer(container):Container` - Finds the associated container for the given container. When mapping from model to collection the input value and source artifact container link field values are from the source repository and must be converted to their associated value to be used in the target system. An exception is thrown if no artifact is found or multiple artifacts are found.

A sample relationship transformation extension:

```
var inputTypes = 'Relationship';
var outputTypes = 'Relationship';

function transform(context, input) {
        if (input) {
                return findParentFolder(context.sourceArtifact);
        }
        return null;
}

function findParentFolder(artifact) {
        var parent = artifacts.retrieveArtifact(artifact['parent']);
        if (parent['subtype'] === 'Folder') {
                return artifact['parent'];
        } else if (parent['subtype'] === null) {
                return null;
        }
        return findParentFolder(parent);
}
```

Looking at the above extension, we find the parent artifact and if that artifact is a folder we return that as the parent.

```
var inputTypes = 'Relationship';
var outputTypes = 'Relationship';

function transform(context, input) {
        var searchDefinition = artifacts.getFormattedIdSearchDefinition();

        searchDefinition['formatted-id'] = 'TPA-42';
        var results = artifacts.searchByFormattedId(searchDefinition);
        if (results[0]) {
                return results[0];
        }
        return null;
}
```

The above extensions uses the formatted ID search to find the correct artifact for the link.

The following extension uses a custom search to determine a relationship:

```
var inputTypes = 'Relationship';
var outputTypes = 'Relationship';

function transform(context, input) {
        var searchType = getCustomSearchType();
        var searchDefinition = artifacts.getSearchDefinition(searchType);
```

```
        searchDefinition['domain'] = 'DEFAULT';
        searchDefinition['project'] = 'My Project';
        searchDefinition['summary'] = context.sourceArtifact.summary;
        var results = artifacts.search(searchType, searchDefinition);
        if (results[0]) {
                return results[0];
        }
        return null;
}

function getCustomSearchType() {
        var searchTypes = artifacts.listSearchTypes();
        for (var i=0; i<searchTypes.length; i++) {
                if (searchTypes[i] === 'My Custom Search') {
                        return searchTypes[i];
                }
        }
        return i;
}
```

💡 **Note**: The returned search results are limited to a maximum of 1024 entries.

Containers and Relationships

A **Container** can be used as input and output type in a Custom Data Transformation extension. Tasktop provides a JavaScript API for working with container fields.

The following two functions are provided to handle containers:

```
artifacts.toRelationship(container)
```

```
artifacts.toContainer(relationship[, summary])
```

All container objects provide a `summary` property.

- `.toContainer(relationship[, summary])` - Converts a relationship object into a container. The summary is provided as a String and is optional. Is no summary provided, the summary of the related artifact is used. An exception is thrown if the artifact or the summary field of the artifact cannot be found.
- `.toRelationship(container)` - Converts a container into a relationship object to use with the `artifacts.retrieveArtifact(relationship)`
- API or return as result of the extension.

The following extension finds the first parent folder and returns that as the parent container.

```
var inputTypes = 'Relationship';
var outputTypes = 'Container';

function transform(context, input) {
        if (input) {
                var parentRelationship = findParentFolder(context.sourceArtifact);
                  return artifacts.toContainer(parentRelationship);
        }
        return null;
}

function findParentFolder(artifact){
        var parent = artifacts.retrieveArtifact(artifact['parent']);
        if (parent['subtype'] === 'Folder') {
                return artifact['parent'];
        } else if (parent['subtype'] === null) {
```

```
            return null;
        }
        return findParentFolder(parent);
}
```

The next extension retrieves the parent of our parent container field and returns it as relationship.

```
var inputTypes = 'Container';
var outputTypes = 'Relationship';

function transform(context, input) {
        if (input) {
                var parentRelationship = artifacts.toRelationship(input);
                var parentArtifact = artifacts.retrieveArtifact(parentRelationship);
                var container = parentArtifact['parent'];
                    return artifacts.toRelationship(container);
        }
        return null;
}
```

💡 **Note**: Only containers based on artifacts are supported.

Comments in Custom Data Transformation Extensions

Comment extensions can be used to achieve use cases such as:

- Splitting long comments in a source collection into multiple comments in a target collection
- Excluding comments from integration based on some set criteria
- ... and more!

Once saved, the extension can be applied on the Comment Configuration screen.

To create a comment extension, **comments** must be declared as the input or output types of the extension.

💡 **Note**: Comment extensions will only impact new comments as they flow through Tasktop Integration Hub. Existing comments that have already been synchronized will not be impacted.

If you are creating a custom data transformation for comments,

- The **Comments** type is supported as an array of comment objects
- A comment will be a javascript object with field ids as the key
  - For example, a private comment with ID **1** and content **This is a comment** may look like this:

```
{
    "id":"1",
    "is-private":true,
    "comment-content":"<p>This is a comment</p>"
}
```

Here's an example of an extension that replaces user information with a default user in outbound comments:

```
// The following extension can be set on Collection to Model transformation on a collection.
// It replaces user information at a repository's comment to a default user and returns comments that matches
```

```
to Hub comment model.
var inputTypes = 'Comments';
var outputTypes = 'Comments';

function transform(context, input) {
    if(input.length >0){
        input.forEach(function(element) {
            replacePeople(element);
        });
    }
     return input;
}

function replacePeople(comment){
    var pattern = /user(\d*)/gi;
    comment['creator']='default';
    comment['work_notes']=comment['work_notes'].replace(pattern,'default');  //replace user information at a
repository's comment contents field, work_notes
    comment['content']=comment['work_notes'];  //assign updated repository's comment content to the Hub's
comment object's comment content field
}
```

Here's an example of an extension that adds a header to inbound comments with a default user:

```
// The following extension can be set on Model to Collection transformation on a collection.
// It addes comment header with default user to the given Hub model's comment input.
var inputTypes = 'Comments';
var outputTypes = 'Comments';

function transform(context, input) {
    if(input.length >0){
            addCommentHeader(input[0]);
    }
     return input;
}

function addCommentHeader(comment){
    var headerText = '<p>[Comment from '+'default_user'+']</p>';
    comment['content']=headerText+comment['content'];  //Hub's comment object's comment content field is comment
['content']
}
```

Concatenation

To concatenate two fields on the source artifact into one field on the target artifact, a custom data transformation extension can be used.

Below, we've outlined how to configure a custom data transformation extension in order to concatenate the Formatted ID and Name from CA Agile Central into the Summary model field. The concatenated values will then flow from the model to the chosen field on the target artifact.
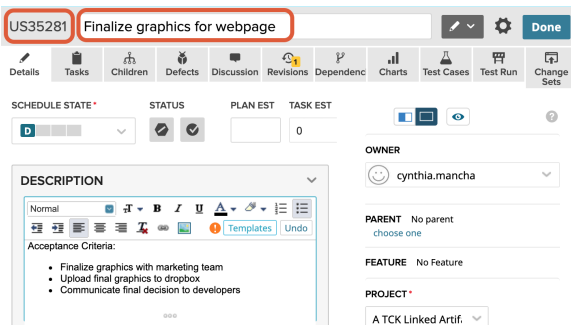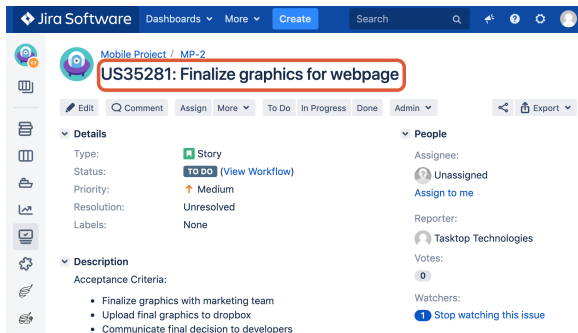
1. Go to the Field Mapping screen for the source (CA Agile Central) collection.
2. If the Summary model field is already mapped in the source collection, delete the mapping.
3. Choose Formatted ID and Name from the left side (repository) dropdown and Summary from the right side (model) dropdown and Press Connect.
4. Make a note of the Type for each of the 2 fields and the order in which they are added. For example, in the below example Formatted ID was added first and is of type **String** and Name was added next and is of type **String**. The Model Field is also of type **String**.

5. Open the Settings in a different tab and go to **Extensions** > **Manage Extensions**.
6. Create a new data transformation extension.
7. Give the extension a name and update the input types based on Step 5. In this case we have 2 Inputs of types **String** and **String**. Update the input types as follows:

```
var inputTypes = ['String', 'String'];
```

      a. 💡 **Note**: This will take the Formatted ID as the 1st parameter and Name as the 2nd parameter.

8. Update the output types based on Step 5. In this example, we have 1 output of type **String**. Update output types as follows:
      a.
```
var outputTypes = 'String';
```

9. In the body of the function, use the following statement to concatenate:
      a.
```
return 'ID: ' + input[0] + ' :: '+input[1];
```

10. Here's an example of the full script:
      a.
```
var inputTypes = ['String', 'String'];
var outputTypes = 'String';


function transform(context, input) {
    // returns the result of the transformation
    return 'ID: ' + input[0] + ' :: '+input[1];
}
```

11. Save and go back to the source collection.

12. Configure the Summary mapping from Step 4:

13. You will now see the extension you created as an option for the transform on the right (model) side. Choose this extension and click **Save** and **Done**.
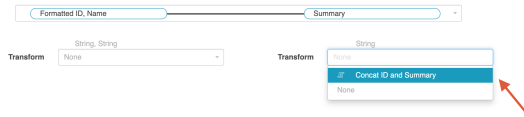
    a.

14. In your target collection, simply map the Summary model field to your chosen field on the target artifact (i.e., Summary, Name, Title, etc).
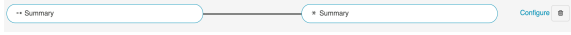
This will concatenate the 2 fields (ID, Name) on the source artifact to a single Summary field on the target artifact.

Person Reconciliation

Integrations that create or update artifacts often need to deal with differences between the representation of persons in different systems.

Tasktop comes with a default **person reconciliation** strategy (**Copy with Default Matching**), which matches based on name, ID, and/or e-mail.

More specifically, the algorithm will compare the metadata from each side as follows:

- Username (person-username) from source to username (person-username) on target
- Username (person-username) from source to ID (person-id) on target
- ID (person-id) from source to username (person-username) on target
- ID (person-id) from source to ID (person-ID) on target
- Email (person-email) from source to email (person-email) from target

Please review the Connector Docs to determine which fields are available for your specific repository. If a field (i.e., person-username) is not available, Tasktop will simply skip that step.

This strategy should cover most use cases. However, if needed, you can also configure a custom Person Reconciliation Extension to match **person** fields from one repository to another.

Configuring Person Reconciliation with Extensions

A person reconciliation extension can be created from the Extensions screen, accessible from Settings. Created extensions are selected in the Person Reconciliation section of the Collection screen. In most cases it makes sense to have one extension per repository, since each repository will have different requirements for mapping persons to and from the repository. Person reconciliation extensions apply to all person fields of an artifact, including person fields in comments and attachments.

The Context Object

The context object provides information that the extension can use to determine how person reconciliation should be handled.

For a custom data transformation, use the following:

- `context:` A context object that provides information that the extension can use to determine which transformations are needed
- `context.sourceArtifact:` A JavaScript object representation of the source artifact
  - If you are mapping from model to repository, this will match the structure of the artifact in the model
  - If you are mapping from repository to model, this will match the structure of the artifact in the repository
- `context.targetArtifact*:` A JavaScript object representation of the target artifact
  - If you are mapping from model to repository, this will match the structure of the artifact in the repository
  - If you are mapping from repository to model, this will match the structure of the artifact in the model

If you are creating a custom data transformation extension for test steps, use:

- `context.sourceTestStep` to access the source test step
  - If you are mapping from model to repository, this will match the structure of the artifact in the model
  - If you are mapping from repository to model, this will match the structure of the artifact in the repository
- `context.targetTestStep` to access the target test step
  - If you are mapping from model to repository, this will match the structure of the artifact in the repository
  - If you are mapping from repository to model, this will match the structure of the artifact in the model

See Tasktop Editions to determine if your edition contains Test Step functionality

The context parameter also has field properties:

- `context.sourceField`: If processing a single field
- `context.sourceFields`: A list of field objects, if processing more than one field
- `context.targetField`: If processing a single field
- `context.targetFields:` A list of field objects, if processing more than one field

A field object only has two properties: ID, and label:

```
{
    id: "assignee",
    label: "Assignee"
}
```

Authoring Person Reconciliation Extensions

Person reconciliation extensions are defined by two functions:

```
mapPersonFromRepository(repositoryPerson, unresolvedPerson)
```

```
mapPersonToRepository(modelPerson)
```

Both functions are expected to return a string value corresponding to the user id of the person. Returning `undefined` sets the person field to empty. In the case where a user cannot be mapped and having the field empty is not an option, throw an exception as follows:

```
if (noMatchFoundCondition) {
    throw 'some descriptive message';
}
```

Such errors will cause processing of an artifact to result in an error with error code CCRRTT-17011E which will display under the Activity screen.

mapPersonFromRepository(repositoryPerson, unresolvedPerson)

`mapPersonFromRepository` is used to create a model representation of a person from a repository representation of a person, which occurs whenever a person is copied from a repository artifact to a model artifact. The return value of this function is used as the id of the person in the model artifact.

Two parameters are passed to the `mapPersonFromRepository` function:

- `repositoryPerson` - An object representing the person corresponding to the repository representation
- `unresolvedPerson` - This parameter contains whatever information may be available about the person from the repository. It contains information only if `repositoryPerson` does not.

An example `repositoryPerson` from Jira On-prem looks like:

```
{
        "person-id": "userA",
        "person-email": "userA@test.tasktop.com",
        "person-display-name":"User A",
        "active": true
}
```

An example `unresolvedPerson` from Jira On-prem might look like:

```
{
        "person-id": "userA",
        "person-email": "userA@test.tasktop.com"
}
```

mapPersonToRepository(modelPerson)

`mapPersonToRepository` is used to create a repository representation of a person from a model representation of a person, which occurs whenever a person is copied from a model artifact to a repository artifact. The return value of this function is used to lookup the corresponding person in the repository.

A single parameter is passed to the `mapPersonToRepository` function:

- `modelPerson` - An object representing the person corresponding to the model representation

A `modelPerson` always has the following properties:

```
{
        "id": "userId",
```

```
        "display-name": "Jane Smith"
  }
```

💡 **Note**: `Display-name` could be empty.

Simple Person Reconciliation Example

A simple person reconciliation mapping extension could look like this:

```
function mapPersonFromRepository(repositoryPerson, unresolvedPerson, context) {
    if (repositoryPerson) {
        return repositoryPerson.id;
    }
}

function mapPersonToRepository(modelPerson, context) {
    if (modelPerson) {
        try {
            var person = persons.searchPerson('id', modelPerson.id);
            console.log("found match " + person.id);
            return person.id;
        } catch (e) {
            console.log("no match found mapping to " + context.targetField.id);
            if (context.targetField.id === "assignee") {
                return "default-assignee";
            } else if (context.targetField.id === "reporter") {
                return "default-reporter";
            } else if (context.targetField.id === "comments") {
                return "default-commenter";
            }
        }
    }
}
```

The SimplePersonReconciliation script is a simple script which makes use of dictionary concept in Javascript to map key and values.

Scenario 1: Using E-mail

Consider an example where Repository 1 has email **john.s@email.com** and Repository 2 has email **john. smith@email.com** and the display names and ID's don't match. Assume that the integration has one-way person flow from Repository 1 (john.s@email.com) to Repository 2 (john.smith@email.com).

In that case, we would edit the var mapping on the `mapPersonToRepository()` function so that the incoming value checks the dictionary (key) and returns a valid email (value) for the repository.

In this example, we would edit the `var mapping = { 'john.s@email.com' : 'john. smith@email.com' }` in the `mapPersonToRepository()` function.

If the integration has two-way person flow, we must also edit the `mapPersonFromRepository()` function. The `mapPersonFromRepository()` function will show the e-mail addresses in the opposite order - i.e. `var mapping = {'john.smith@email.com' : 'john.s@email.com'}`. For two-way integrations, the person reconciliation extension must be added to both the source collection and the target collection.

Scenario 2: Using ID

If the source repository does not provide an e-mail, we can use the Simple Person Reconciliation script above to match person ID to person email.

For example, if Repository 1 has user id "JohnSmith" and the matching user in Repository 2 is "john. smith@email.com," then we should edit the script at `var mapping = { JohnSmith: 'john. smith@email.com'}`.

If the integration has two-way person flow, we will also need to edit the `mapPersonFromRepository ()` as outlined in Scenario 1. We must also remember to edit the extension in `var result` as `modelP erson[person-id]` for scenarios where we are using ID instead of email. The edit must be done on both the `mapPersonFromRepository()` and `mapPersonFromRepository()` functions.

Selecting a Default Person when No Match is Found

Below is a script which uses the context to select a default person when no match is found:

```
function mapPersonToRepository(modelPerson, context) {
        var person;
        try {
                person = persons.searchPerson("email", modelPerson.id);
        } catch (e) {
                // no matching person found
                // select a default person by team
                if (context.sourceArtifact["team"] === "Team A") {
                        person = persons.searchPerson("email", "team.a.lead@company.net");
                } else if (context.sourceArtifact["team"] === "Team B") {
                        person = persons.searchPerson("email", "team.b.lead@company.net");
                }
        }
        // return a match if found
        if (person) {
                return person["person-id"];
        }
}
```

Person Reconciliation Extension Javascript API

Tasktop provides a JavaScript API for working with persons in a person reconciliation extension. This API includes two functions:

- `persons.listPersonSearchFields():Object\` - Allows for the discovery of the searchable fields on a person. Not all fields from a person are searchable and vary between connectors.
- `persons.searchPerson(fieldId, fieldValue):Person\` - Used to search for person in a repository. This person can then be used to return the correct ID for a user in a repository. `pe rsons.searchPerson(fieldId, fieldValue)` will find exactly one person and will throw a `PersonNotFoundException` if no match is found or `TooManyPersonsFoundException` if more than one person is found. These exceptions can be caught and handled by the extension.

Artifacts returned from the artifact API are the raw JSON representation of a repository's artifact. These representations may include internal IDs and other fields not mapped to the model. It may be necessary to manually interpret the results of these calls on a per-repository basis to determine the exact information that is returned.

Below is a person reconciliation extension that will take the id of a model person, retrieve the user by username and return the exact ID from the repository. This is helpful for systems where the person's ID is a number or some other non-human readable value.

```
function mapPersonFromRepository(repositoryPerson, unresolvedPerson) {
    return repositoryPerson['Username'];
}

function mapPersonToRepository(modelPerson) {
    // persons.listPersonSearchFields(); determines the fields usable by .searchPerson(...)
    var repositoryPerson = persons.searchPerson('Username', modelPerson['id']);
    return repositoryPerson['ID'];
}
```

SearchPerson Example Script

Below is an example SearchPerson script. `Persons.searchperson (fieldId, fieldValue)` is used to search for a person in a repository using the two parameters: `fieldId` and `fieldValue`.

This person can then be used to return the matching ID of a user in that repository.

`Persons.searchPerson(fieldId, fieldValue)` will find exactly one person and will throw a `PersonNotFoundException` if no match is found or `TooManyPersonsFoundException` if more than one person is found. These exceptions can be caught and handled by the extension. `searchPerson()` is a native Tasktop call, which means it is functionality that is unique to Tasktop.

```
function mapPersonFromRepository(repositoryPerson) {
        ...
}

function mapPersonToRepository(modelPerson) {
        if (!modelPerson){
                console.log('incoming model person is empty')
                return undefined
        }

        console.log('modelPerson = ' + modelPerson['id']);

   var repoPerson = persons.searchPerson('person-username', modelPerson['id']);

   console.log('repoPerson = ' + repoPerson['id']);
        return repoPerson['id'];
}
```

Scenario 1: Mismatched E-mails

Consider an example where Repository 1 has email **john.s@email.com** and Repository 2 has email **john.smith@email.com**. The `persons.searchPerson(fieldId, fieldValue)` can be used to search the repository for matching person values.

Assume that the integration has one-way person flow from Repository 1 (john.s@email.com) to Repository 2 (john.smith@email.com). In this case, the `mapPersonToRepository()` function should be edited and the incoming values matched by ID. A search persons call based on incoming username is made and then the matching user object is retrieved.

Scenario 2: Returning a Default ID as a Value

```
function mapPersonFromRepository(repositoryPerson) {
        return repositoryPerson['email'];
```

```
        }

function mapPersonToRepository(modelPerson) {
        var defaultUserId = 'SOMEVALUEHERE'

        console.log(persons.listPersonSearchFields())
        try{
                var person = persons.searchPerson('email', modelPerson.id);
                if(person != null) {
                        return person['person-username'];
                }
        } catch(e){
                console.log(e)
        }
        console.log('Falling back to default person')
        return defaultUserId
}
```

The above script allows us to search for persons in the repository based on an incoming email value. In cases where a corresponding person is not found in the repository, Tasktop will return the `defaultUs erId`. To return a default user ID, assign a default value (a user ID) to the `var defaultUserId`.

PersonListSearchFields

`Persons.listPersonSearchFields()` allows for the discovery of the searchable fields on a person. Not all fields from a person are searchable and vary between connectors.

```
function mapPersonFromRepository(repositoryPerson) {
        return repositoryPerson['email'];
}

function mapPersonToRepository(modelPerson) {
            console.log(JSON.stringify(persons.listPersonSearchFields()))
        var person = persons.searchPerson('person-email', modelPerson.id);
        return person['person-id'];
}
```

For example, when using the above Person Reconciliation script/extension on the Jira side in a Jira-Micro Focus (HPE) integration, the `console.log(JSON.stringify(persons. listPersonSearchFields()))` line will give you a list of the searchable fields.

In our demo, we got the following values:

```
 Person listSearch Fields: ["person-username","person-email","person-id","person-display-name"]
```

You can then use one of those available values as part of the `persons.searchPerson() script`. In the example scripts shown above, we make use of `person-id`.

Using LDAP or Active Directory

LDAP (Lightweight Directory Access Protocol) and Active Directory can be used to lookup information required to map persons from one system to another. Tasktop provides a JavaScript API for accessing LDAP and Active Directory as follows:

```
function mapPersonToRepository(modelPerson) {
        ldap.connect('ldap://subdomain.mycompany.com', 'cn=admin,dc=example,dc=mycompany,dc=com', 'mypassword');
        var results = ldap.search('dc=example,dc=mycompany,dc=com', 'cn='+ldap.escape(modelPerson['id']))
        if (results.length == 0) {
            throw 'no person found with id='+modelPerson['id'];
```

```
        }
        return results[0]['sn'];
}
```

Looking at the example above, three steps are involved:

1. establishing a connection
2. looking up the appropriate entries using a search
3. returning a value from the search results

The same approach is used for both LDAP and Active Directory.

The Tasktop JavaScript LDAP API is described as follows:

*   `ldap` - The globally-visible object providing the LDAP API
*   `ldap.connect(connectionUrl, principal, password):void` - A means of establishing a connection with a connection URL, user principal and password
*   `ldap.search(base, query, fields):Map[]` - A means of searching providing a base name of the context to search, a search query, and an optional list of fields to provide in the search results
*   `ldap.escape(value):String` - A means of escaping string literals to use in LDAP search queries or distinguished names

There is no need to close an LDAP connection; LDAP connections are managed implicitly by Tasktop.

Artifacts returned from the Artifact API are the raw JSON representation of a repository's artifact. These representations may include internal IDs and other fields not mapped to the model. It may be necessary to manually interpret the results of these calls on a per-repository basis to determine the exact information that is returned.

Accessing Web Resources

Extensions may access resources using HTTP. For example, extensions may access a REST API which could provide data necessary for the extension.

Tasktop provides a fluent JavaScript API for making HTTP requests, inspired by the Java 9 HTTP client API. The API is used as follows:

```
var response = httpClient.request()
        .uri('http://example.com/my/rest/api')
        .parameter('first-param','first-value')
        .parameter('second-param','second-value')
        .header('my-special-header','header-value')
        .GET().response()

if (response.statusCode() == 200) {
        var responseJson = JSON.parse(response.content());
        // do something with response data
}
```

HTTP Client API Reference

- `httpClient` - The globally-visible object providing the HTTP client API
- `httpClient.request():RequestBuilder` - Provides a RequestBuilder object
- `RequestBuilder.uri(uriString):RequestBuilder` - Specifies the URI of the request
- `RequestBuilder.parameter(key,value):RequestBuilder` - Adds a query parameter to the request with the given key and value
- `RequestBuilder.header(key,value):RequestBuilder` - Adds an HTTP header value to the request with the given key and value
- `RequestBuilder.GET():Request` - Creates a Request object for an HTTP GET request
- `Request.response():Response` - Creates a Response object with the result of the HTTP request
- `Response.statusCode():int` - Provides the HTTP status code of the response
- `Response.content():String` - Provides the body of the HTTP response as a string
- `Response.headers():Map` - Provides the HTTP response headers as a JavaScript object with property names corresponding to HTTP header names, and values as arrays of values of the corresponding HTTP header
- `RequestBuilder.proxy(hostname, port[, username, password]): RequestBuilder` - Specifies an HTTP proxy to be used for this request

Artifacts returned from the Artifact API are the raw JSON representation of a Repository's Artifact. These representations may include internal IDs and other fields not mapped to the model. It may be necessary to manually interpret the results of these calls on a per-repository basis to determine the exact information that is returned.

Example extension `Response.headers()` return value:

```
{
        "Transfer-Encoding": [
                "chunked"
        ],
        "Server": [
                "Jetty(9.2.13.v20150730)"
        ],
        "Vary": [
                "Accept-Encoding, User-Agent"
        ],
        "Content-Type": [
                "application/json;charset=UTF-8"
        ]
}
```

Using an HTTP Proxy Server

Extensions can specify an HTTP proxy server with the following API:

```
var response = httpClient
    .request()
    .proxy("myproxy.mycompany.com", 3128)
    .uri('https://www.example.com')
    .GET()
    .response();
```

To use a proxy server with BASIC proxy authentication, credentials can be specified as shown below assuming username and password are strings:

```
var response = httpClient
    .request()
    .proxy("myproxy.mycompany.com", 3128, username, password)
    .uri('https://www.example.com')
    .GET()
    .response();
```

Good to Know

- All communication to the proxy server uses HTTP, not HTTPS, so even if an HTTPS connection to the target server is tunneled through the proxy, it is important that the connection to the proxy server is through a trusted network if sending proxy credentials.
- We recommend storing the proxy password in a confidential key-value store and not hard coding it in the extension because extensions are stored unencrypted in Hub's operational database.

Causing Extensions to Complete With An Error

There are occasions where extensions should complete with an error. In such cases, simply use the JavaScript `throw` keyword as follows:

```
if (somethingUnexpected) throw 'some descriptive message'
```

Such errors will cause processing of an artifact to result in an error with error code CCRRTT-17011E which will display on the Activity screen.

Troubleshooting Extensions

Extension troubleshooting usually involves trial and error. To make the troubleshooting process easier, a global logging function is exposed as follows:

`console.log(message)`

`console.log` takes a single argument which is converted to a string.

For example:

```
function transitionArtifact(context,transitions) {
    if (someUnexpectedCondition) {
        console.log('source artifact: '+JSON.stringify(context.sourceArtifact));
        console.log('target artifact: '+JSON.stringify(context.targetRepositoryArtifact));
        console.log('transitions: '+JSON.stringify(transitions));
        throw 'message describing that something bad happened';
    }
}
```

The output of `console.log` goes to the Tasktop log file at `logs/extensions.log`

Extensions and State

Extensions should not rely on declared variables to retain state between invocations. Doing so is not supported and has undefined behavior.
For example:

// This is not supported:

```
var myGlobalState = // some state


function someFunction() {
    if (myGlobalState == someValue) {
      ...
    }
}
```

Accessing Object Properties

There are two ways to access object properties:

You can use the dot notation if the property name only contains alpha-numeric and characters that are allowed in JavaScript variables such as '$' or '_'.
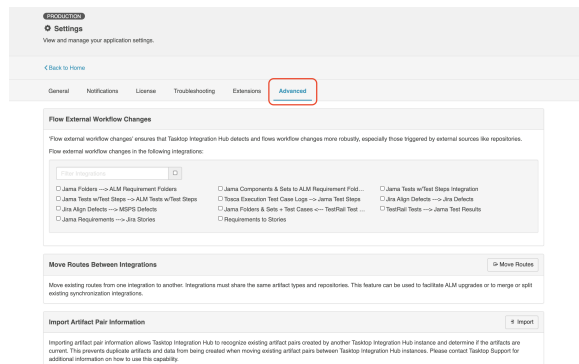
For example:

```
person.email
```

Advanced (Settings)

Introduction

Advanced (Settings) can be accessed by clicking the **Advanced** tab on the **Settings** screen.

Under **Advanced**, you can access:

- Flow External Workflow Changes
- Move Routes Between Integrations
- Import Artifact Pair Information
- Upgrade Backup Files

Flow External Workflow Changes

Flowing external workflow changes allows you to specify the integrations that should receive high-fidelity scans to ensure that changes that might have been missed during a regular change detection interval are detected.

⚠️ **Note**: Enabling this setting for all integrations may have a negative impact on Hub's performance.

Move Routes Between Integrations

There may be rare scenarios when you must move routes from an existing integration to another integration. For example, you may have configured Tasktop incorrectly and mistakenly created multiple integrations which should be combined into one. Or you could be upgrading Micro Focus ALM, which requires moving projects from an instance running an old version of ALM to a server running a newer instance of ALM. Since existing projects are moved to a completely new ALM instance with a different URL, users must create a new repository connection, collection(s), and integration(s) in Tasktop. Once the new integration is created, existing routes must be migrated to prevent the risk of duplicate artifacts. This feature will allow users to easily migrate routes from an existing integration to a new one.

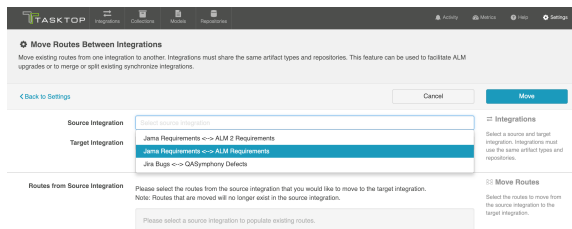To move routes from one integration to another, they must both:

- Be synchronize integrations
- Use the same artifact types
- Use the same repository connections (except for Micro Focus ALM connections used in an upgrade scenario)

💡 We recommend stopping both integrations before moving routes so that you can review your mappings and configuration before running.
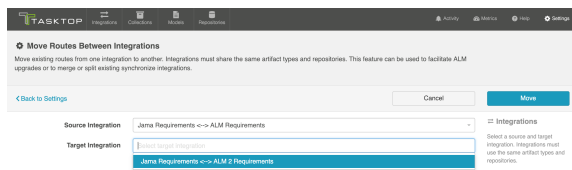
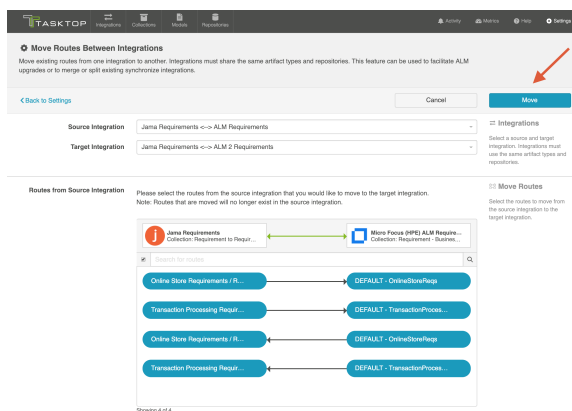To use this feature, click **Move Routes**.

Select your source integration:

Then select your target integration:

Select the routes from the source integration that you'd like to move to the target integration. Once moved, they will no longer exist in the source integration. Click **Move** in the upper right corner.

Review the pop-up message and if approved, click **I understand...** and **Move**. This process may take some time. Progress can be tracked on the Background Jobs tab of the Activity Screen .

Once the move is complete, review your integration configuration, field mappings, etc, before clicking **run** on the target integration.

Import Artifact Pair Information

Importing artifact pairs allows Tasktop Integration Hub to import existing artifact pairs that were created by Tasktop Sync. This prevents duplicate artifacts from being created when you switch from using Tasktop Sync to Tasktop Integration Hub to administer your integrations.

Please contact Tasktop Support for additional information on how to use this capability.

Upgrade Backup Files

This feature is not applicable to Tasktop Cloud and is only available when upgrading from Tasktop Integration Hub versions 20.1 and later.

Upgrading backup files enables you to download and upload artifact data in cases where integrations were resumed individually during an upgrade. The downloaded data corresponds to artifacts that were modified when migrations were still running. These files capture any synchronization activity that occurred on individually running integrations, so that you can ensure no updates are duplicated if restoring from backup.

Learn more about utilizing this capability here.