User Guide

Tasktop: Apex Release (17.1)

January 31, 2017

User Guide

This is the user guide and associated resources for Tasktop: Apex Release (17.1).

Table of Contents

Installation Primer

Tasktop: Apex Release (17.1)

Overview

This section describes how to install Tasktop Integration 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.

Hardware Requirements

Tasktop Integration Hub must be installed in a server environment. You will need an account with administrative privileges on your server to install and configure Tasktop Integration Hub.

Supported Operating Systems

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

- Windows 7 SP1
- Windows 10
- Windows Server 2008 R2 SP1
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2016
- Red Hat Enterprise Linux 6.x
- Red Hat Enterprise Linux 7.x
- Ubuntu Linux 12.04 LTS
- Ubuntu Linux 14.04 LTS
- Ubuntu Linux 16.04 LTS
- SUSE Linux Enterprise Server 11.x
- SUSE Linux Enterprise Server 12.x

Supported Browsers

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

- Internet Explorer 11 or later
- Firefox 46.0.1 and up
- Chrome 50.0.2661.102 and up

Supported Databases

The following databases and versions are supported for use in an Enterprise Reporting Integration:

- Microsoft SQL Server 2008 (including SP1, SP2, SP3, SP4)
- Microsoft SQL Server 2008 R2 (including SP1, SP2, SP3)
- Microsoft SQL Server 2012 (including SP1, SP2)
- Microsoft SQL Server 2014 (including SP1)
- Microsoft SQL Server 2016
- MySQL 5.5
- MySQL 5.6
- MySQL 5.7
- Oracle 11g
- Oracle 12c

A database is required for only the Enterprise Reporting Integration (which is available if you have purchased the Tasktop Data add-on). A database is not otherwise required for running Tasktop.

Java Runtime Environment

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

Hardware Sizing for Deployment Scenarios

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

Tasktop Integration 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. These sizing recommendations apply to the server machine running Tasktop Integration Hub.

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 Integration Hub should be deployed in an environment that has good network throughput and low latency to all repositories and databases involved in an integration.

Small Deployment

A deployment managing up to 20,000 ALM artifacts and up to 200 active users.

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

Small deployment system sizing is roughly equivalent to an EC2 T2 Medium instance.

Medium Deployment

A deployment managing up to 100,000 ALM artifacts and up to 1,000 active users.

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

Medium deployment system sizing is roughly equivalent to an EC2 T2 Large Instance.

Large Deployment

A deployment managing many ALM repositories and 200,000+ ALM artifacts and over 2,000 active users.

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

Large deployment system sizing is roughly equivalent to an EC2 M4 Large or M3 Large Instance.

Sandbox Environment

It is recommended that you prepare a sandbox environment to test your Tasktop Integration Hub configuration before deploying it in production. This sandbox environment should include a sandbox server to install Tasktop Integration Hub on, and sandbox instances of all ALM systems you will be integrating, with the same project structure and customizations as, and a comparable number of artifacts to your production ALM systems.

After you have configured Tasktop Integration Hub on the sandbox server and are happy with the way it is running against your sandbox ALM systems, you can install Tasktop Integration Hub on your production server and recreate the configuration against your production ALM systems.

Installation

Where to Download Tasktop Integration Hub

To get the latest version of Tasktop Integration Hub, first create an account on http://my.tasktop.com, then contact your Solutions Architect or Tasktop Support (support@tasktop.com) and ask them to enable the latest Tasktop Integration Hub download for your account. Click the 'My Downloads' button. This will lead you to http://my.tasktop.com/download_products.php, where you will be able to download the latest version of Tasktop Integration Hub.

TASKTOP

HOME OUR PRODUCTS ACCOUNT SIGN OUT BLOG SUPPORT

PRODUCTS INTEGRATIONS INSIGHTS PARTNERS RESOURCES COMPANY

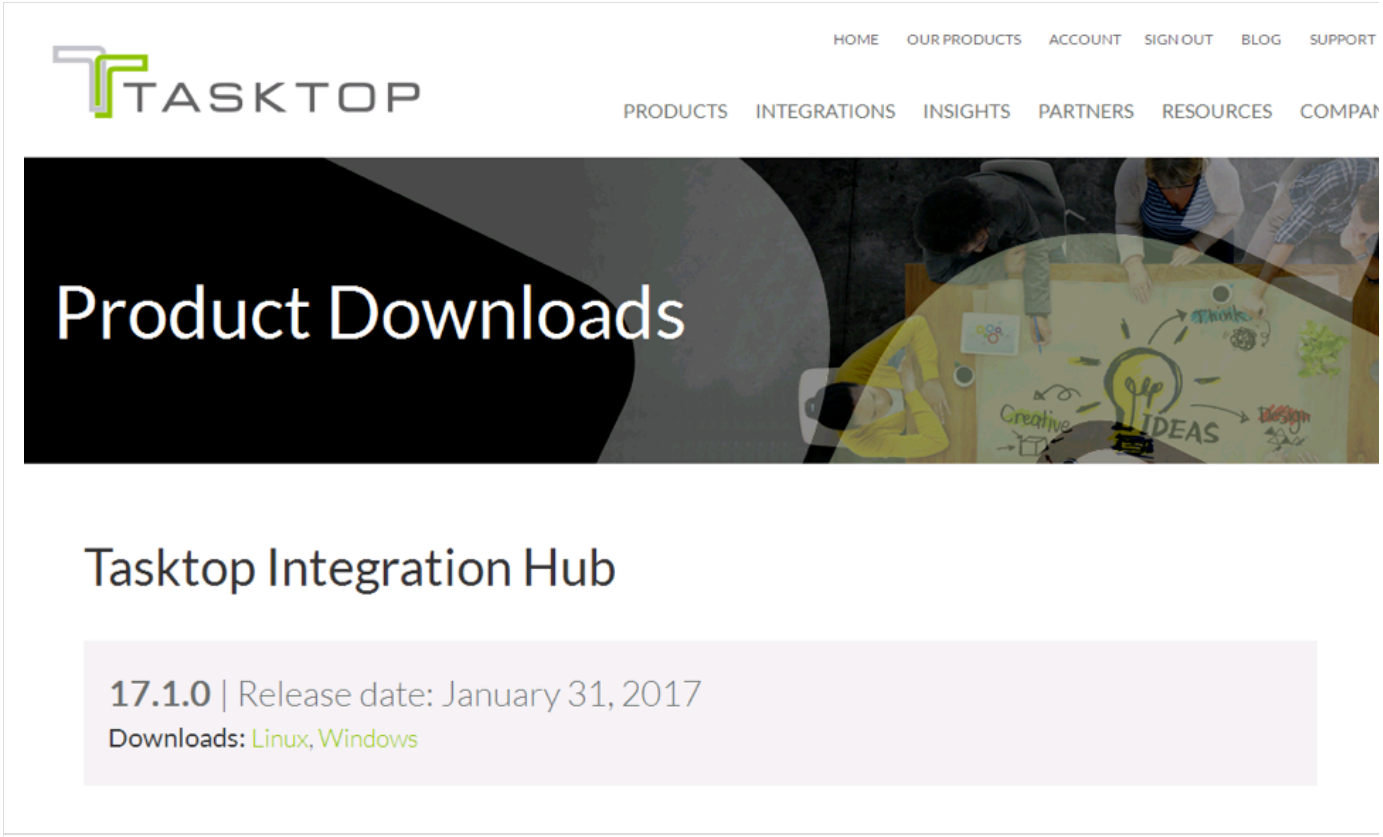
Customer Portal

SIGN OUT

ACCOUNT

MY DOWNLOADS

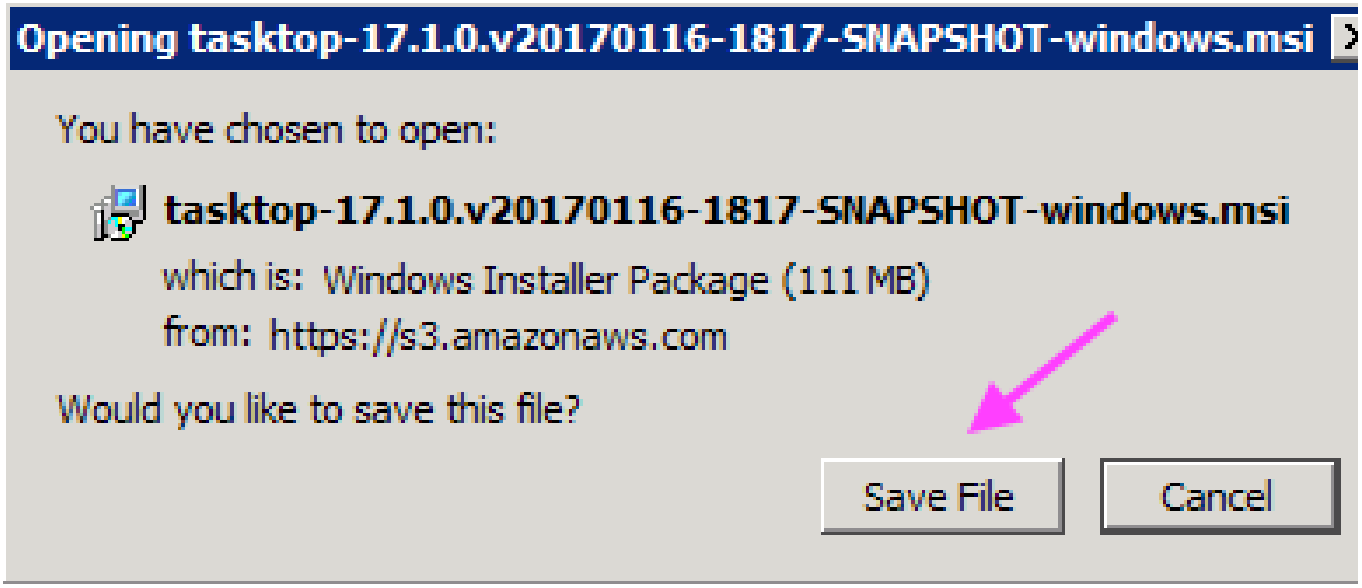
MY RESOURCES

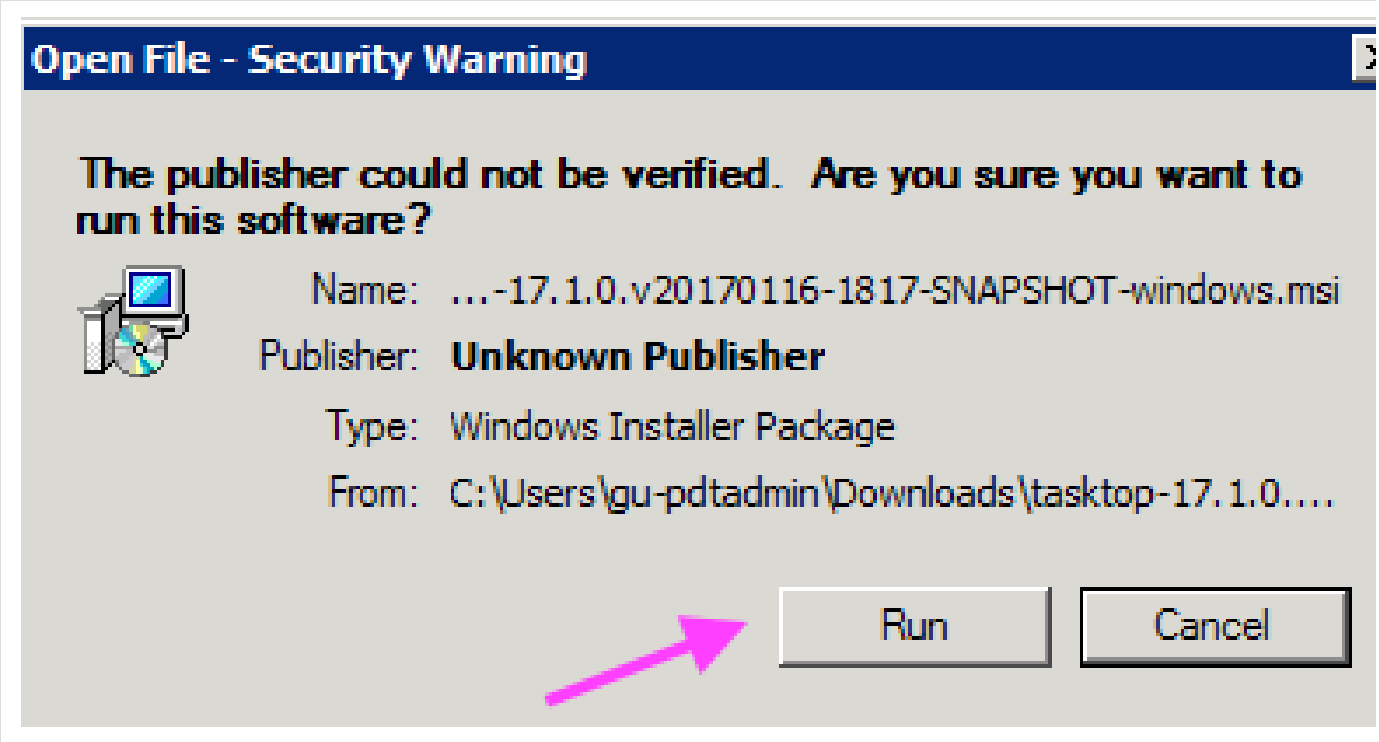


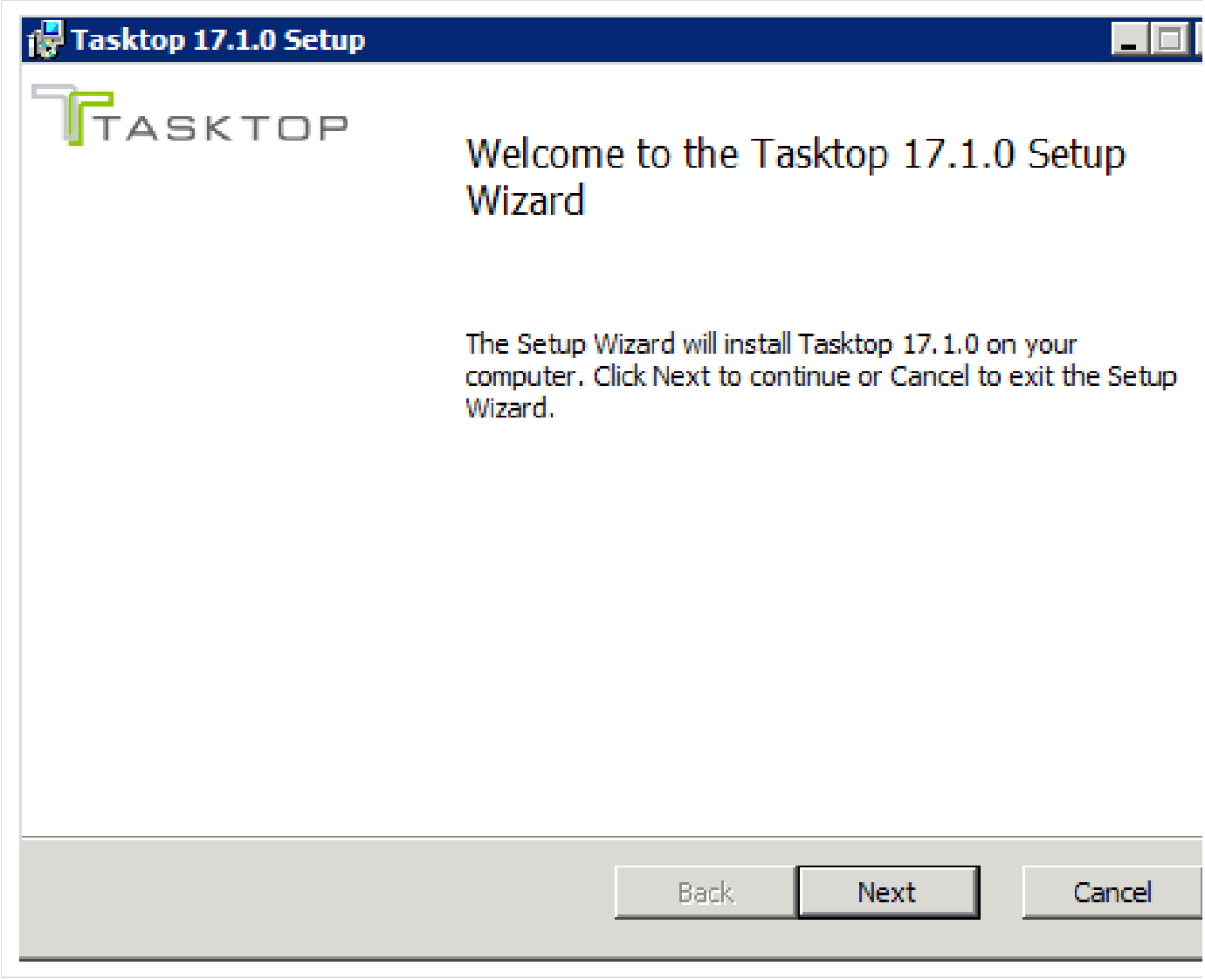
Installation on Windows

Click on the 'Windows' download link on the Product Downloads page of my.tasktop.com.

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



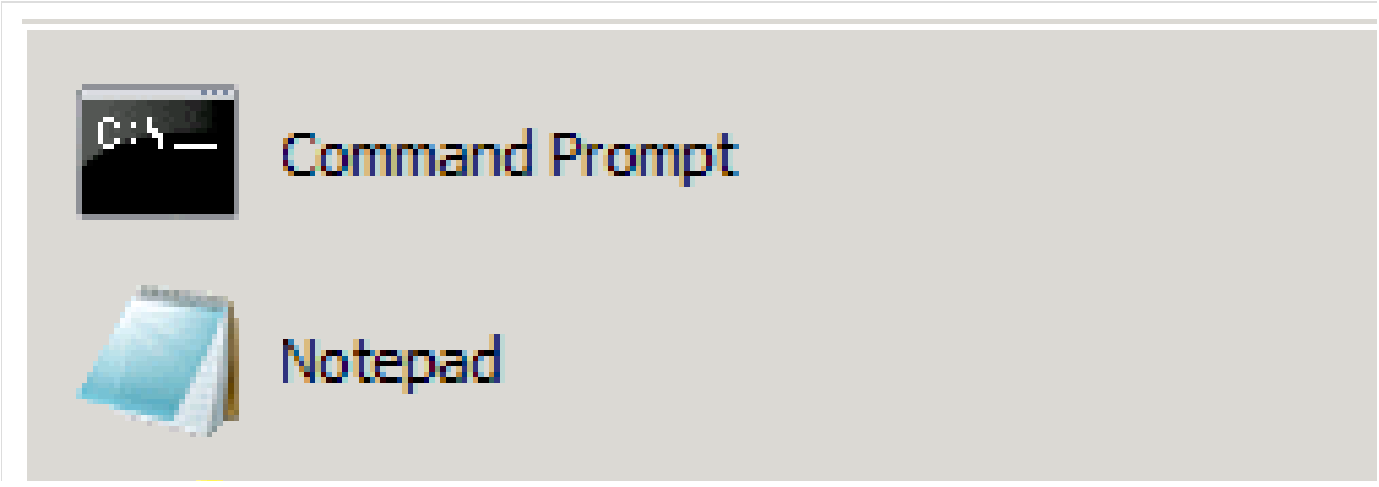

Open File - Security Warning
The publisher could not be verified. Are you sure you want to run this software?
Name: ...-17.1.0.v20170116-1817-SNAPSHOT-windows.msi
Publisher: Unknown Publisher
Type: Windows Installer Package
From: C:\Users\gu-pdtadmin\Downloads\tasktop-17.1.0....
Run
Cancel

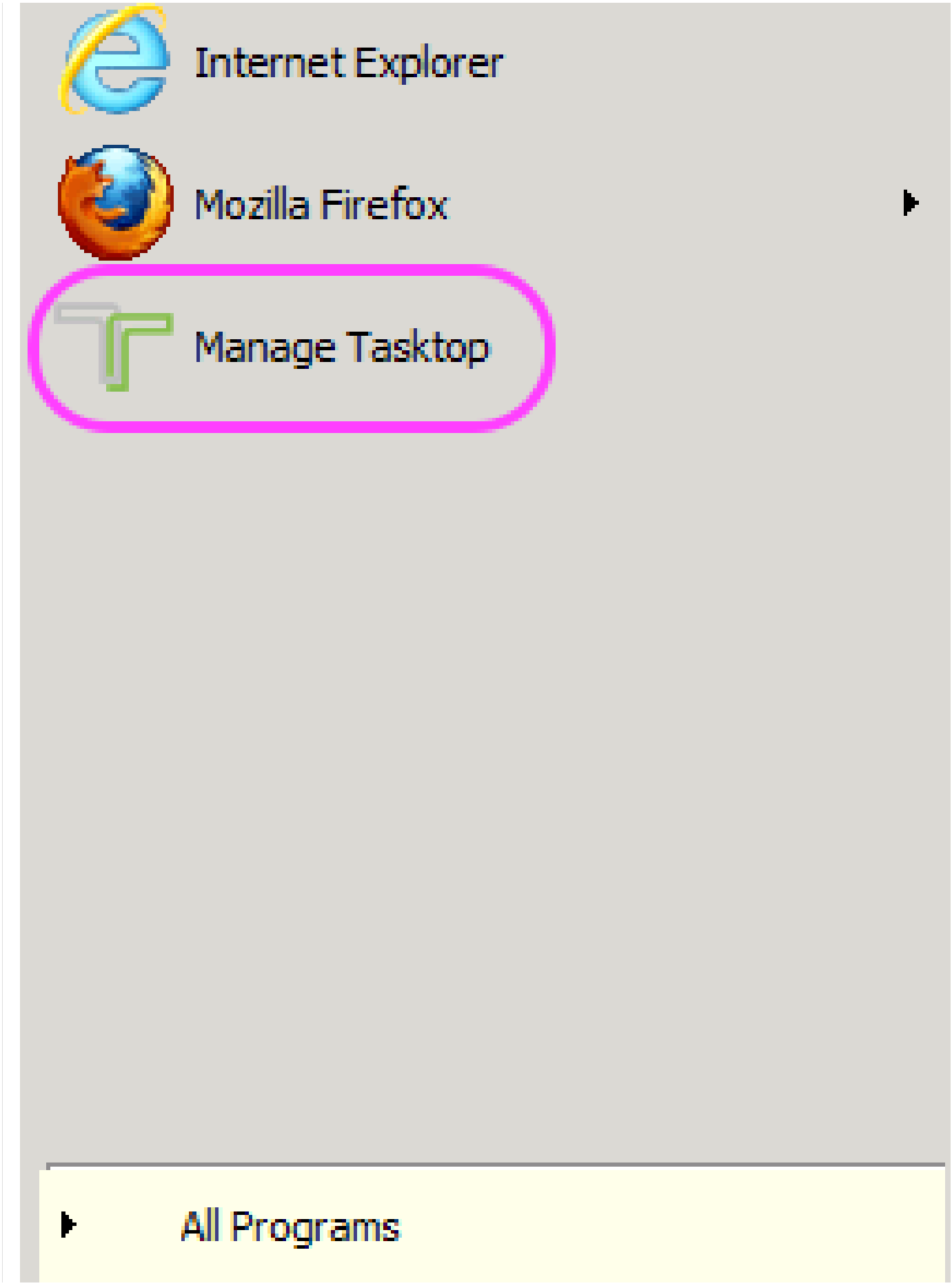


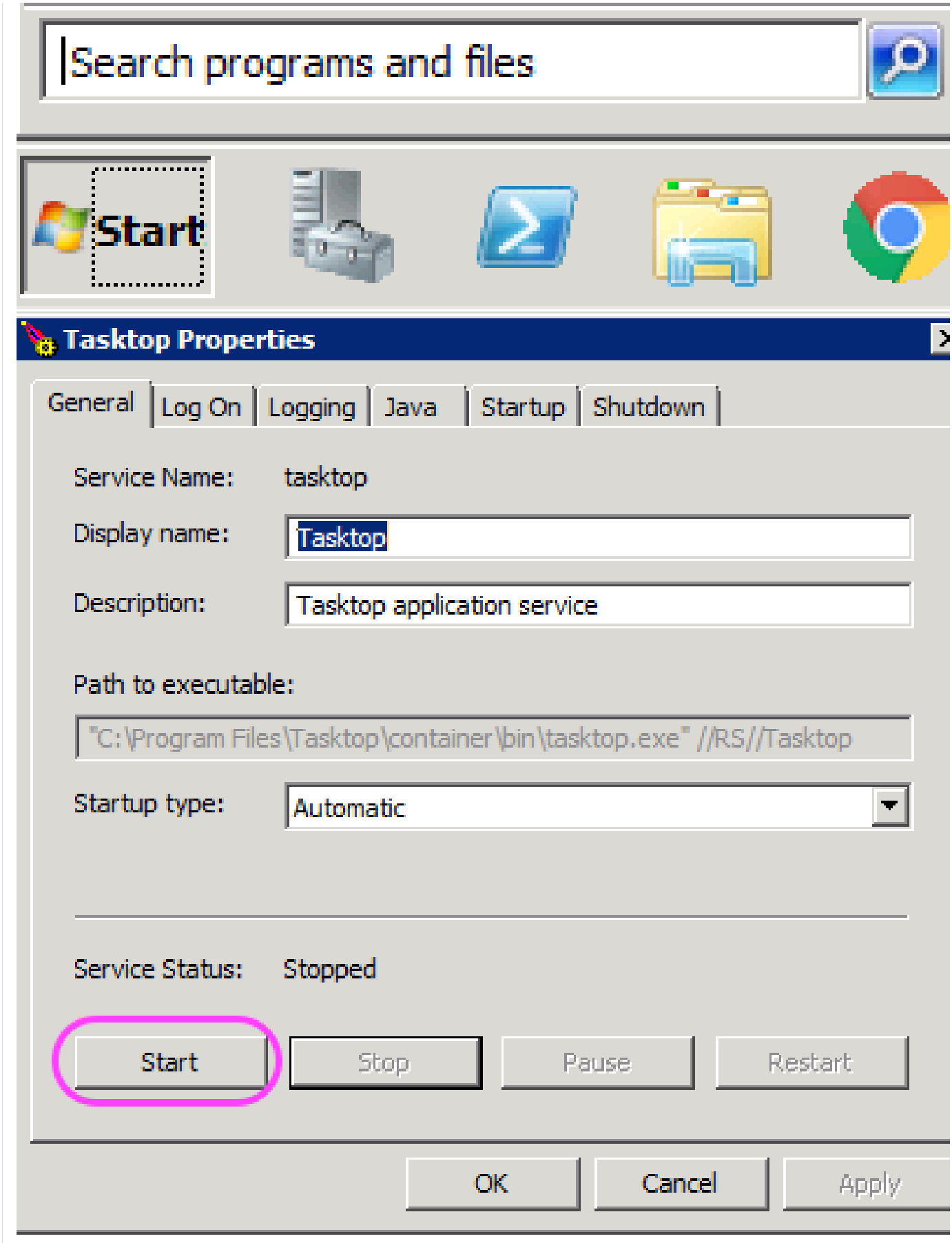
You will then be lead through the installation wizard. Follow the prompts to install Tasktop.

To start Tastkop Integration Hub as a service, click on the Manage Tasktop shortcut on the desktop. Click on 'Start' on the dialog.

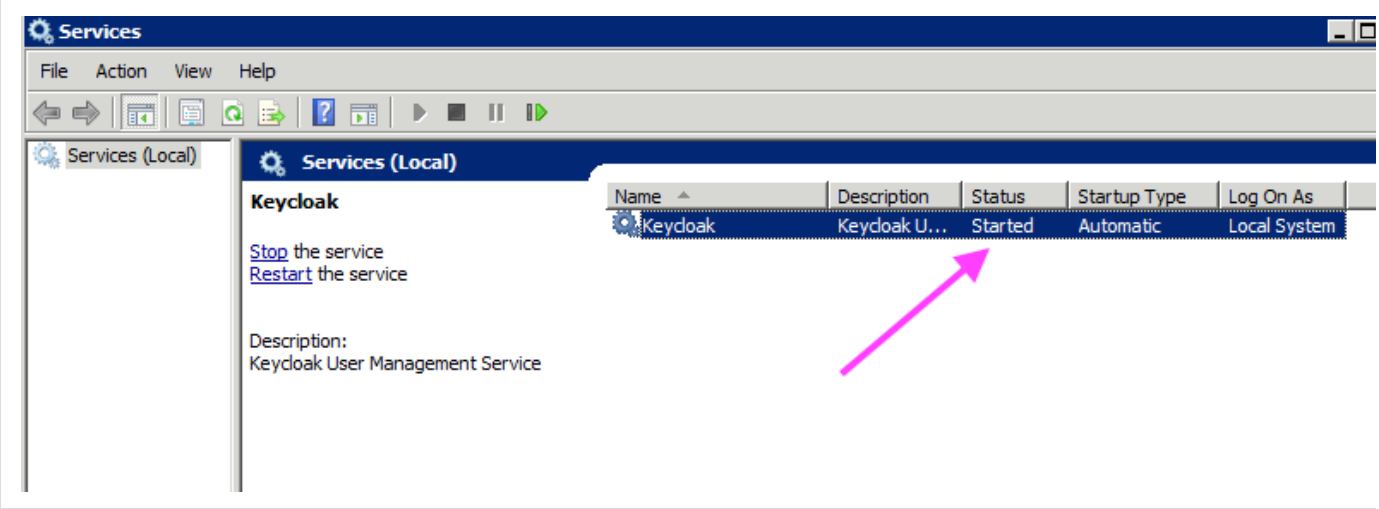
⚠ Note that this does not start or stop Keycloak (for User Management). Please ensure that you read and execute the following step to start Keycloak, or else you may encounter errors.




Internet Explorer
Mozilla Firefox
Manage Tasktop
All Programs


Search programs and files
Start
Tasktop Properties
General
Log On
Logging
Java
Startup
Shutdown
Service Name:
tasktop
Display name:
Tasktop
Description:
Tasktop application service
Path to executable:
"C:\Program Files\Tasktop\container\bin\tasktop.exe" //RS//Tasktop
Startup type:
Automatic
Service Status:
Stopped
Start
Stop
Pause
Restart
OK
Cancel
Apply

To ensure that the User Management Service is started, go to Control Panel>Administrative Tools>Services, and search for 'Keycloak,' and 'Tasktop.' Both should say 'started' under status and 'automatic' under startup type. If not started, right click each one, and select 'start.' If not automatic, right click and select 'properties.' Under 'start-up type,' select 'automatic.' Click 'apply' and 'ok.'



To begin using Tasktop Integration Hub, open http://localhost:8080 in any of our supported browsers.

Installation on Linux

You will be provided with an installation package for Tasktop Integration Hub as a .tar.gz archive. Extract this archive to your desired location, then refer to the README.txt file inside for instructions on starting and stopping Tasktop Integration Hub.

To start tasktop, in the folder where Tasktop is installed, run start-tastkop.sh.

To begin using Tasktop Integration Hub, open http://localhost:8080 in any of our supported browsers.

Port Configuration

Tasktop Integration Hub

The default port Tasktop listens for requests on is 8080. To change this port, follow these instructions:

In the Tasktop installation directory, open container/conf/server.xml

1. Find the HTTP connector configuration (the <Connector> element with protocol="HTTP/1.1")
2. Change the port attribute to the port you wish to use
    - e.g. to use port 8888: <Connector port="8888" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />

If you change the port, the address used to access Tasktop (http://localhost:8080) will need to be updated with the new port number in place of '8080.'

User Management Service

Tasktop user management runs on a separate sevice on the same machine as the integration application. Requests to Tasktop redirect to the user management server for validation. The user management server listens via port 8081. Port 8081 must be available for local communication, but does not need to be open

for remote access.

Default File Locations

Default File Locations on Windows

When Tasktop Integration 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, and the configuration files and logs are located in C:\ProgramData\Tasktop (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).

Default File Locations on Linux

When Tasktop Integration 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.

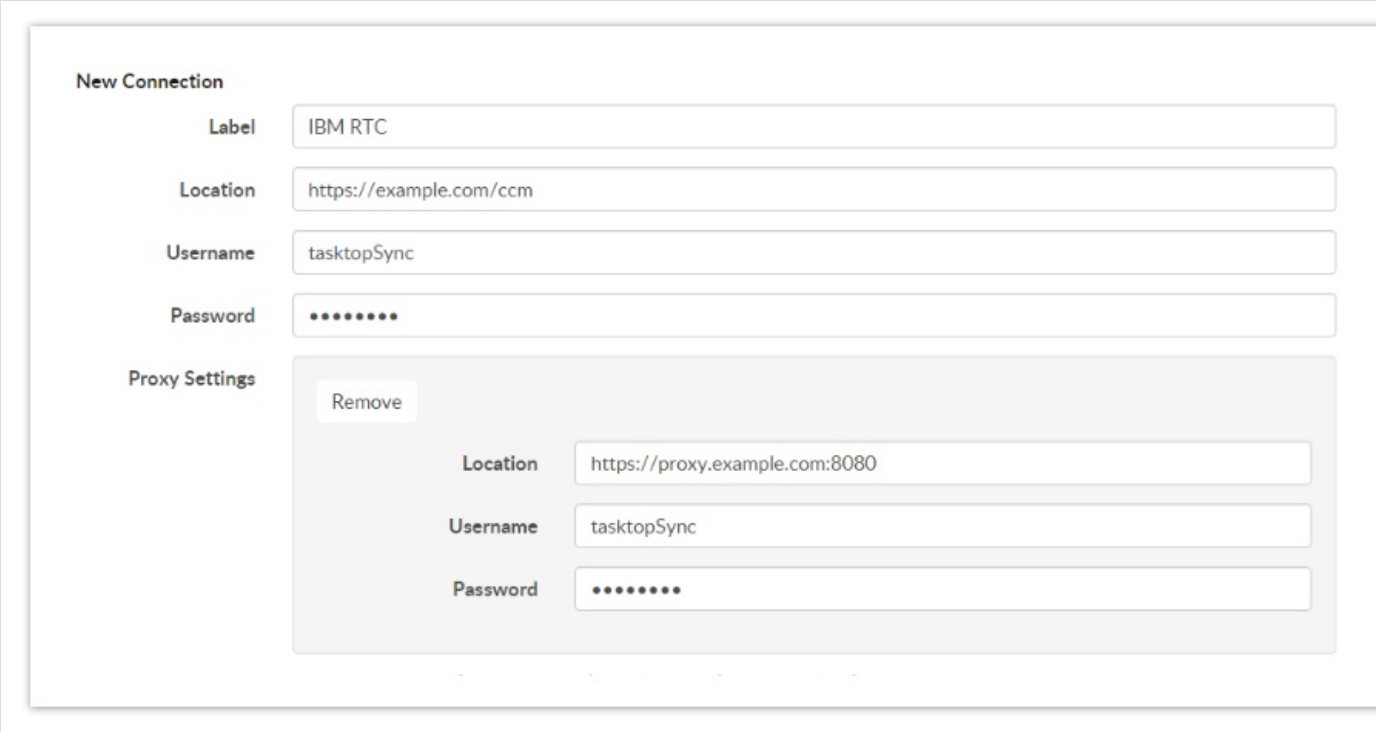
Preparing Your ALM Systems

Before using Tasktop Integration Hub with your ALM Systems you will need to perform some simple preparation on each ALM System you will be integrating. This preparation includes creating a user account for Tasktop Integration Hub with the appropriate permissions, and possibly other steps. Please refer to the specific preparation document for each of your ALM systems for detailed instructions.

Firewalls and Proxies

If Tasktop Integration Hub is installed behind a firewall, you may need to connect to external ALM systems (e.g. hosted or cloud ALM systems) through a proxy. To create a connection to such external

ALM systems in Tasktop Integration Hub, you can make Tasktop Integration Hub connect through your proxy by configuring the proxy settings when creating a new repository connection (see Figure 1). It is recommended to create login credentials specifically for Tasktop Integration Hub on the proxy server.



User Management

Tasktop: Apex Release (17.1)

Getting Started

Tasktop comes pre-configured with an admin user.

Username: admin

Password: Tasktop123

After logging in for the first time, the admin will be prompted to change his or her password.

Types of Users

There are two types of users: Admins and Users

The only differences between the two user types are regarding user management. An admin can create

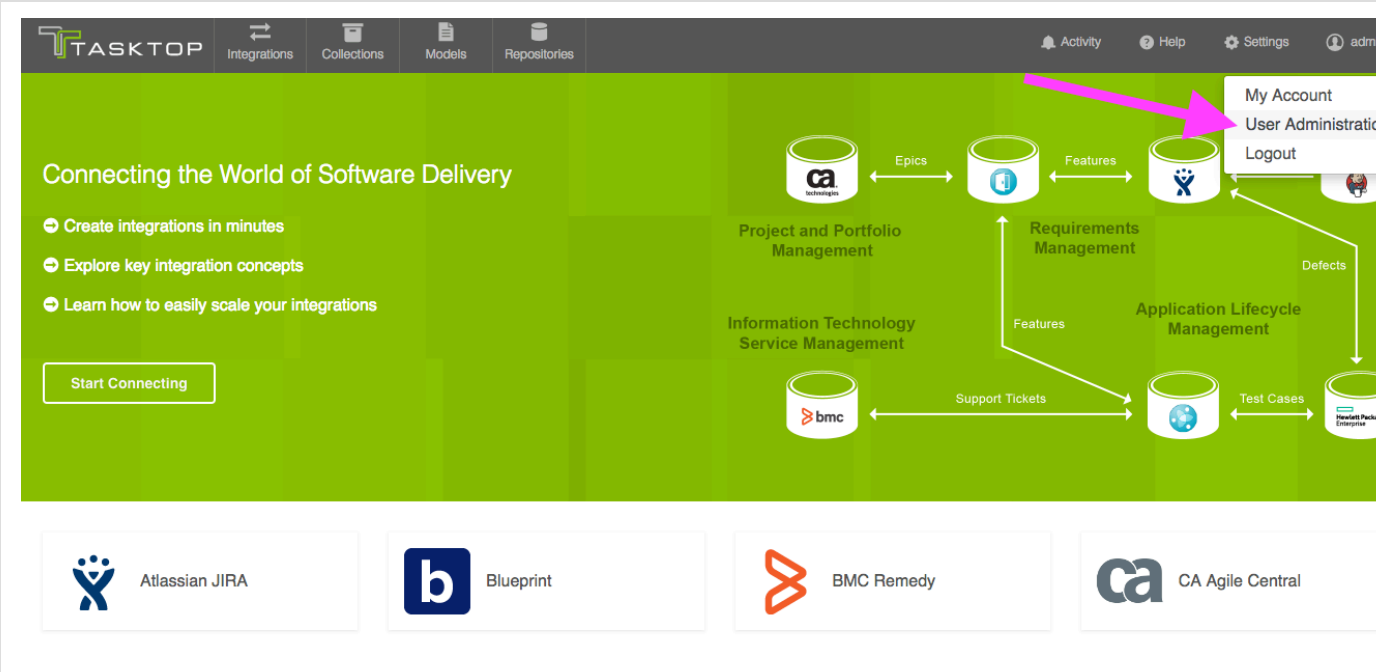
new users, update users' passwords, and change users' group membership (from user to admin or vice-versa). Both user types have the same permissions with regard to Tasktop functionality (meaning that both have all permissions needed to create, modify, and run integrations).

💡 We recommend configuring at least two admin users. This way, if one admin forgets their password, the other admin will be able to log in and re-set the other admin user's password.

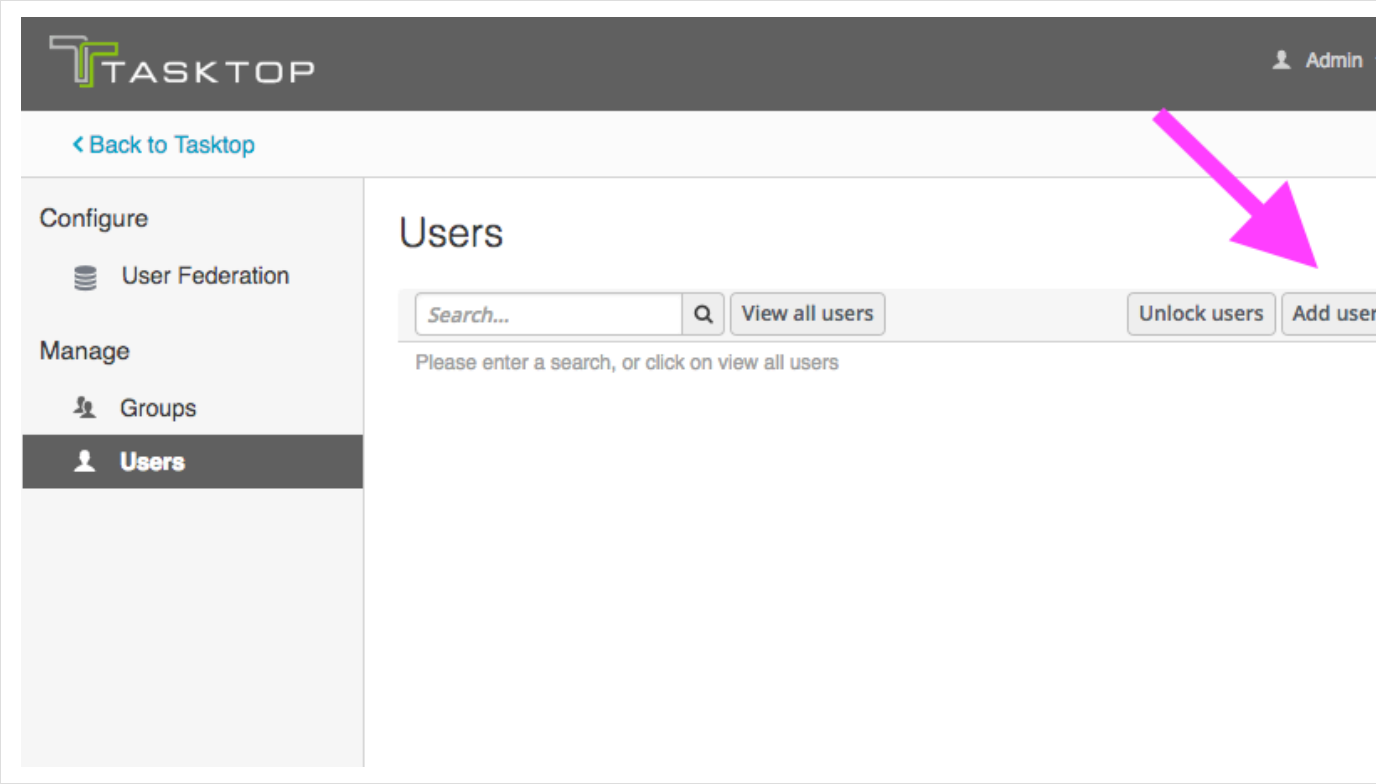
Capability	Admin	User
Create New User	✅	❌
Reset Any User's Password	✅	❌
View and Modify Any User's Group Membership	✅	❌
Reset Own Password, Name, or E-mail	✅	✅
Create and Modify Repository Connections	✅	✅
Create and Modify Models	✅	✅
Create and Modify Collections	✅	✅
Create, Modify, and Run Integrations	✅	✅

Creating a User

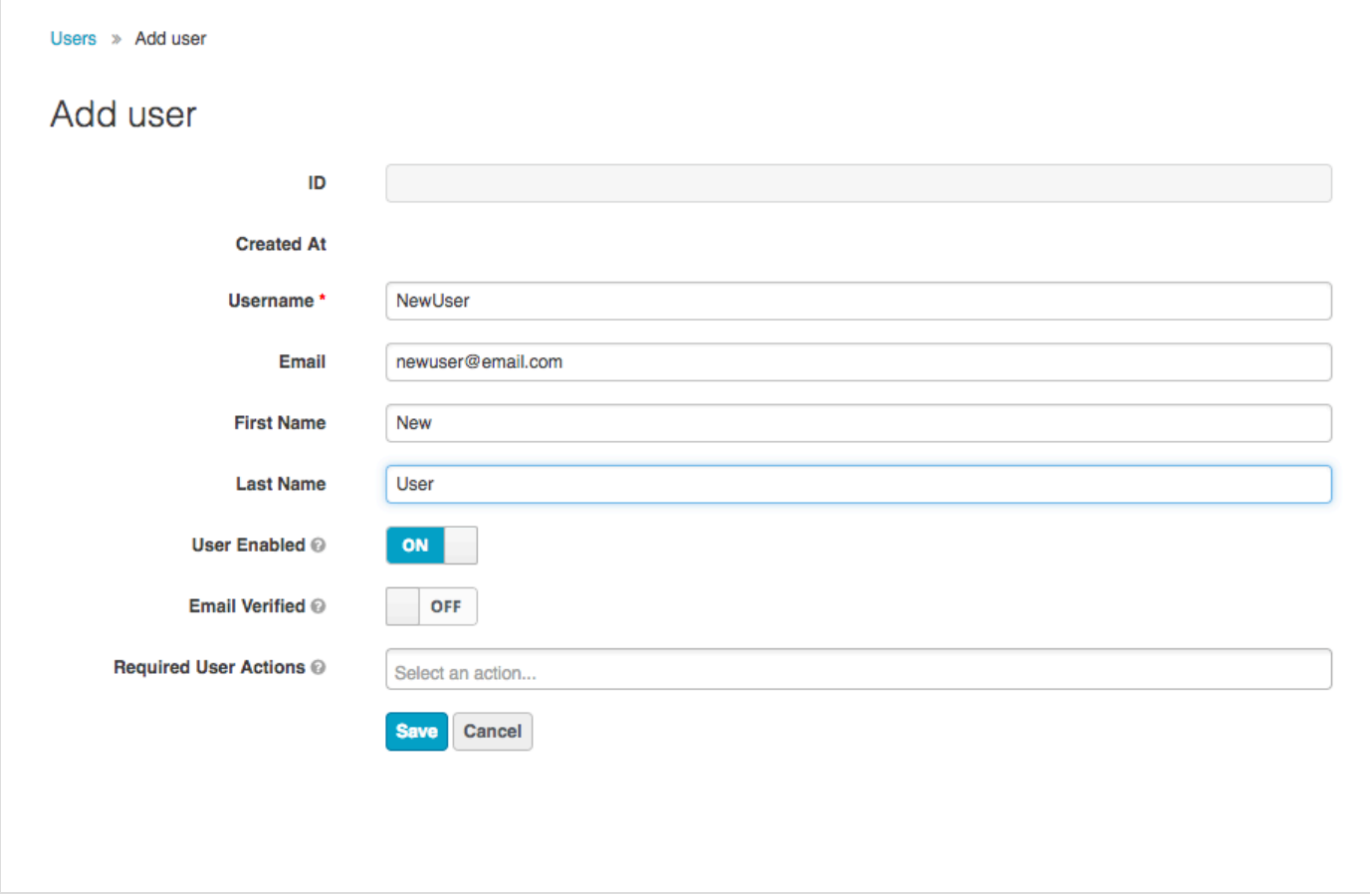
To create a user, you must have admin capabilities. To create a user, select 'User Administration from the upper right corner of the application.



From the User Administration screen, select 'Add user'

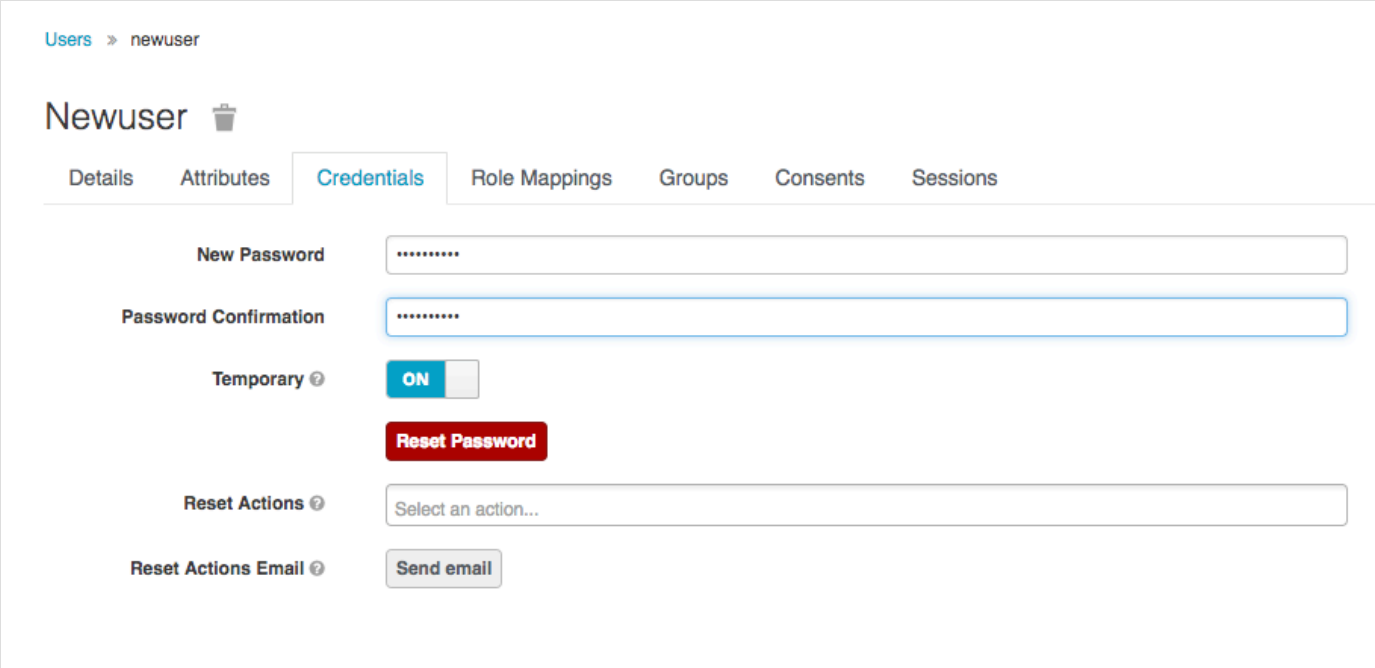


On the Add User screen, populate the Username, Email, First Name, and Last Name sections. The rest of the sections can be ignored.



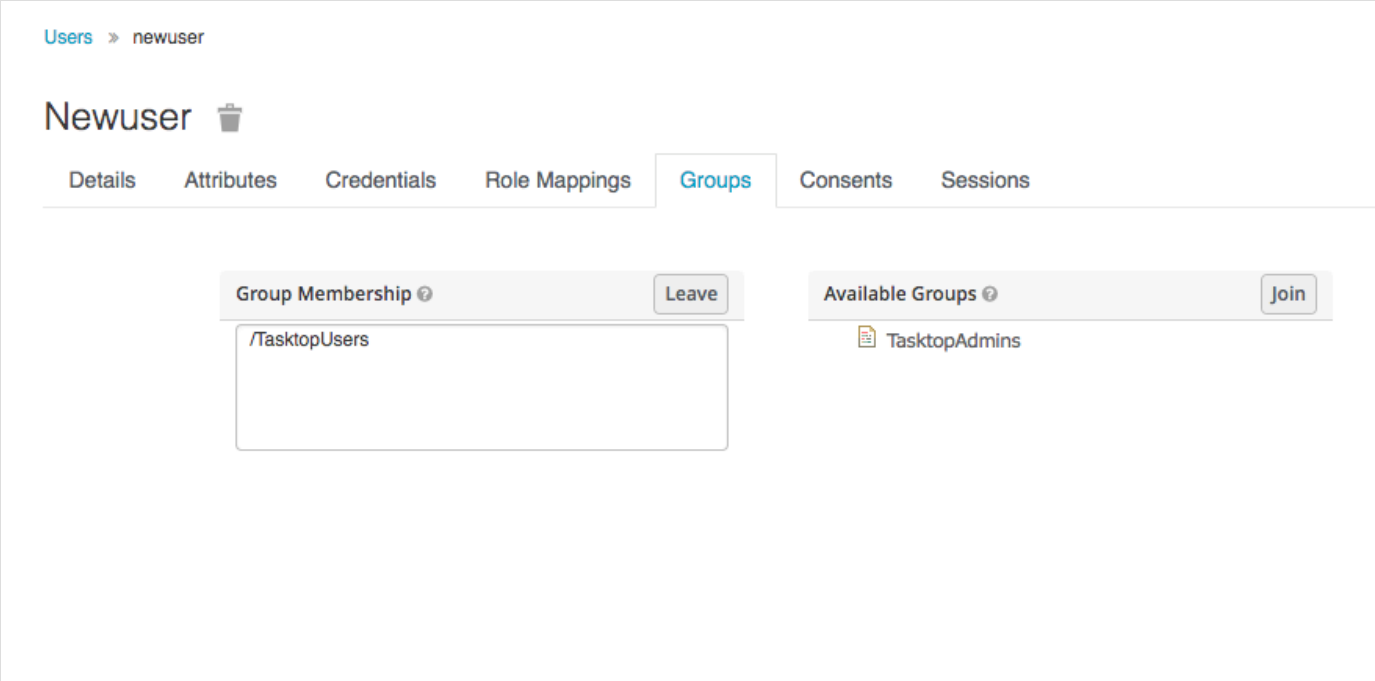
Click the 'Credentials' tab and give the user a temporary password. Make sure 'temporary' is set to 'on'.

This will allow them to set a new password upon their first log-in. Then click 'Reset Password.'



Click on the 'Groups' tab. Add the user to a group - either TasktopUsers or TasktopAdmins, depending on the permissions you'd like the user to have.

⚠️ If the new user is not added to a group, they will not be able to successfully access the Tasktop Integration Hub.



You can ignore the following tabs: Attributes, Role Mappings, Consents, and Sessions.

Your user has been added, and can log in with their temporary password.

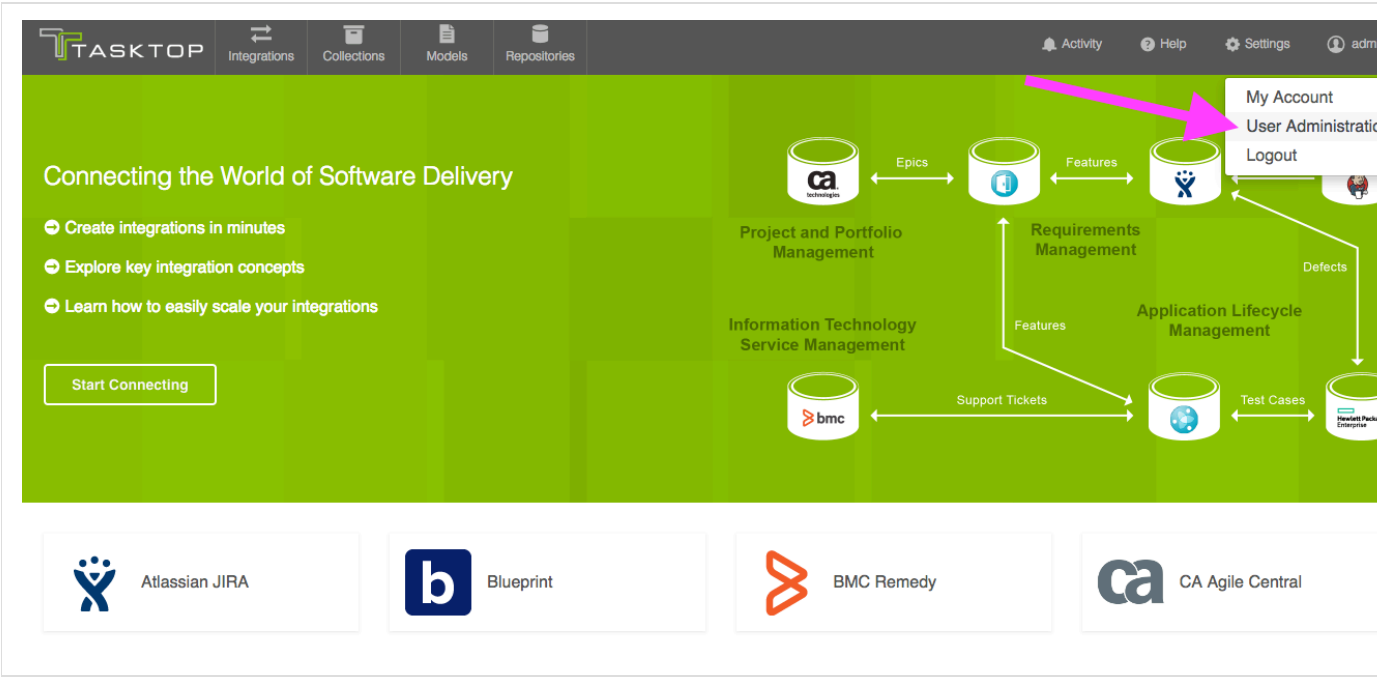
⚠️ Note that Tasktop will not send the new user an e-mail notification. The admin must notify the user of

the new account and password.

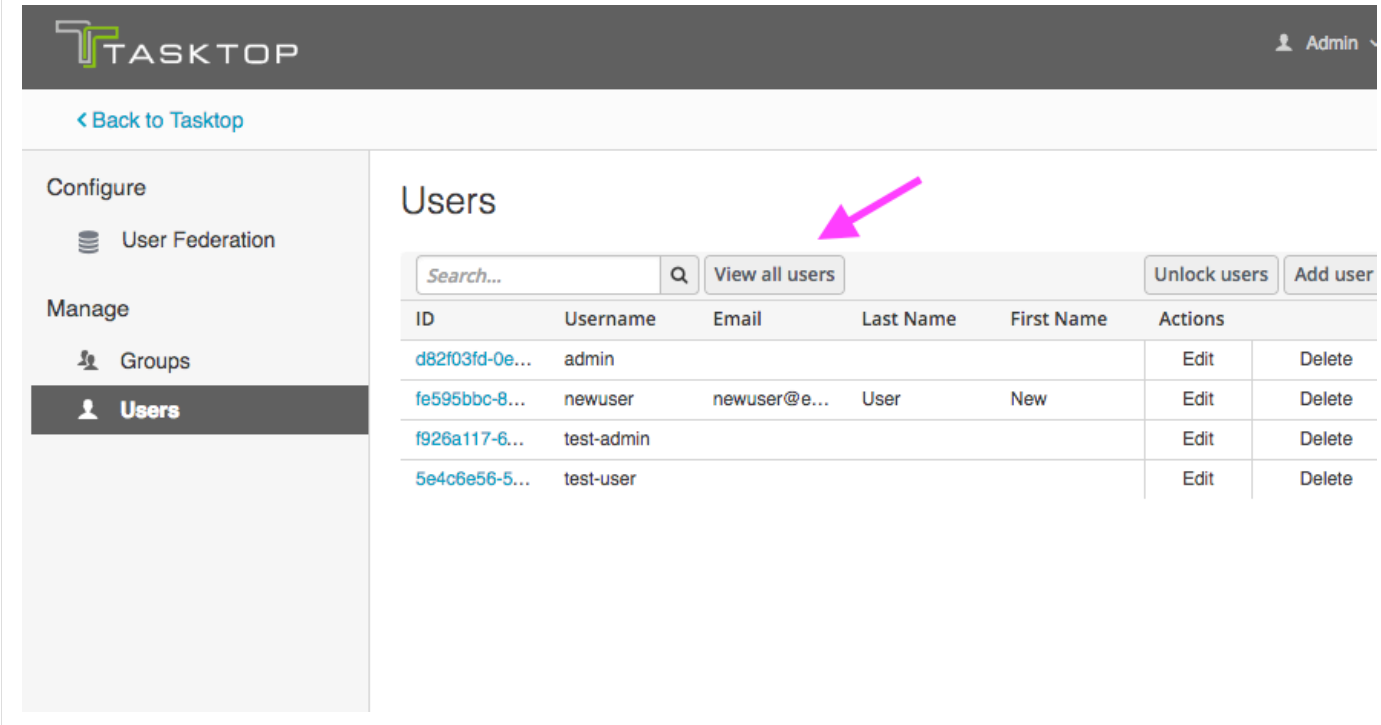
Resetting a User's Password

To re-set a user's password, you must have admin capabilities.

To re-set a user's password, select 'User Administration' from the upper right corner of the application.

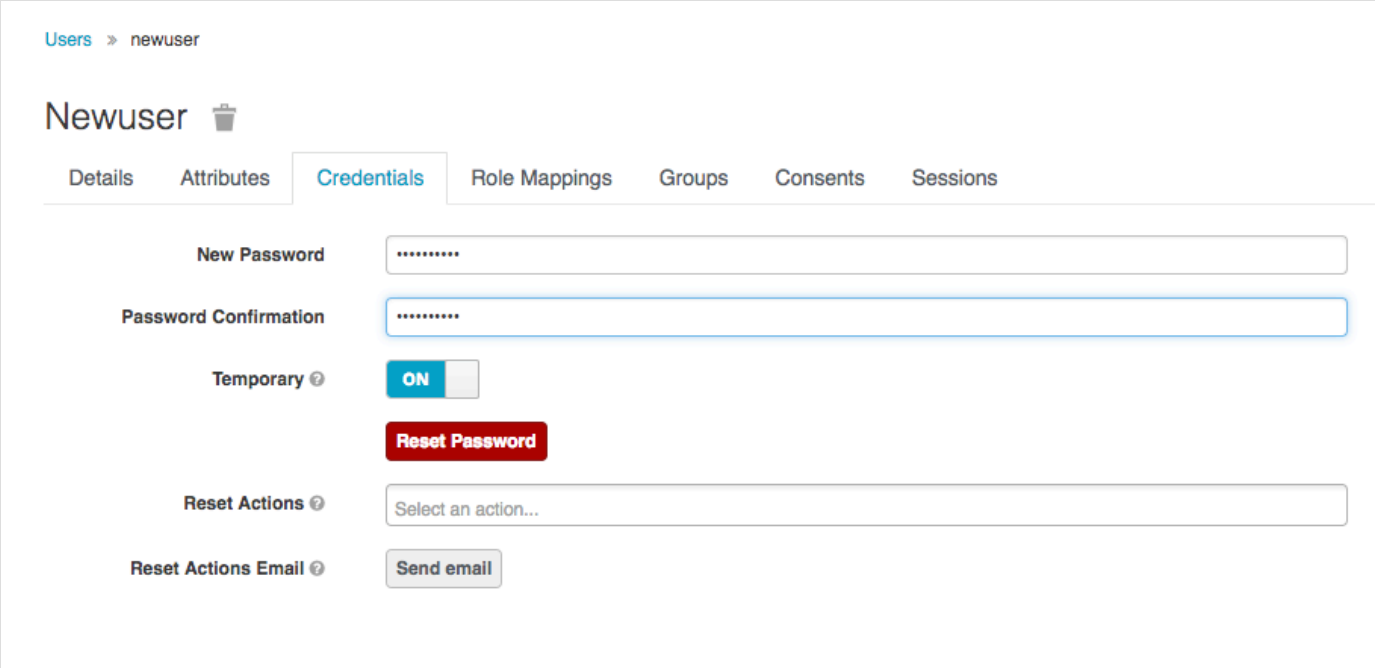


Click 'View all Users.'



Click on the ID for the user whose password you would like to re-set. Then, click on the 'Credentials' tab and give the user a new temporary password. Make sure 'temporary' is set to 'on'. This will allow them to

set a new password upon their first log-in. Then click 'Reset Password.'



⚠️ Note that Tasktop will not send the user an e-mail notification. The admin must notify the user of the new temporary password. The user will be prompted to set a new password upon their next log-in.

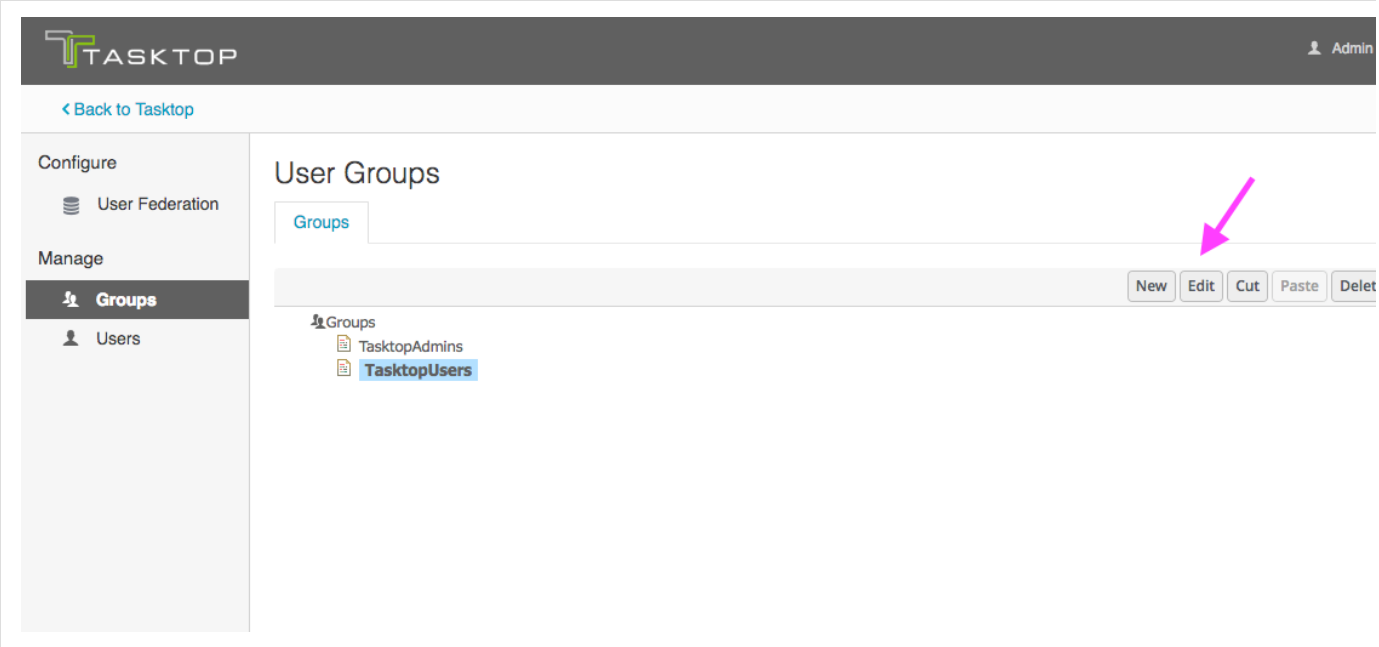
Managing Groups

Viewing Members of a Group

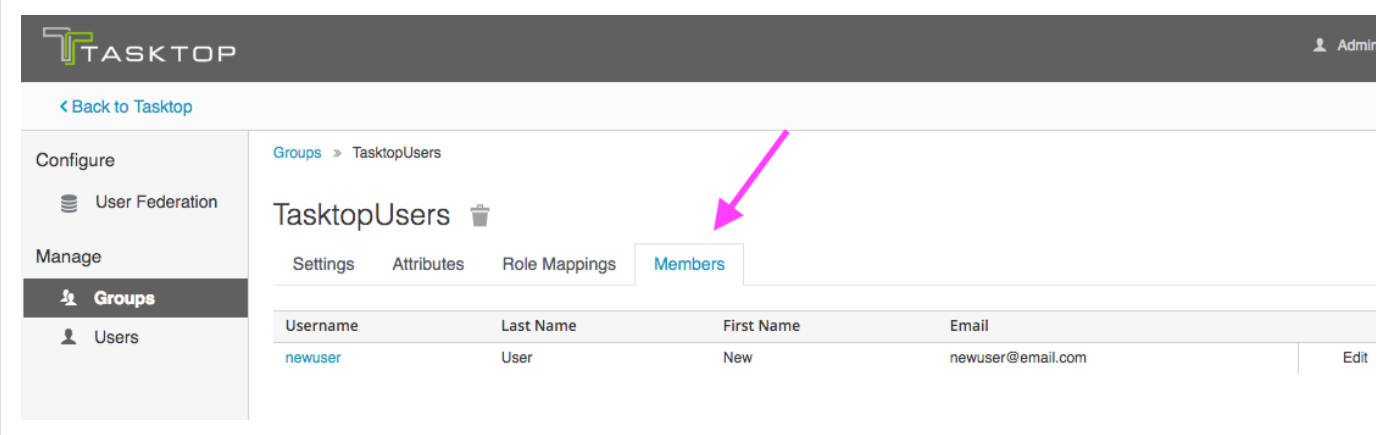
To view members of a group, you must have admin capabilities.

To view the members of a group, click 'Groups' on the left pane of the User Management screen.

Select the group you'd like to review, and click 'edit.'



Click the 'Members' tab to view current members.



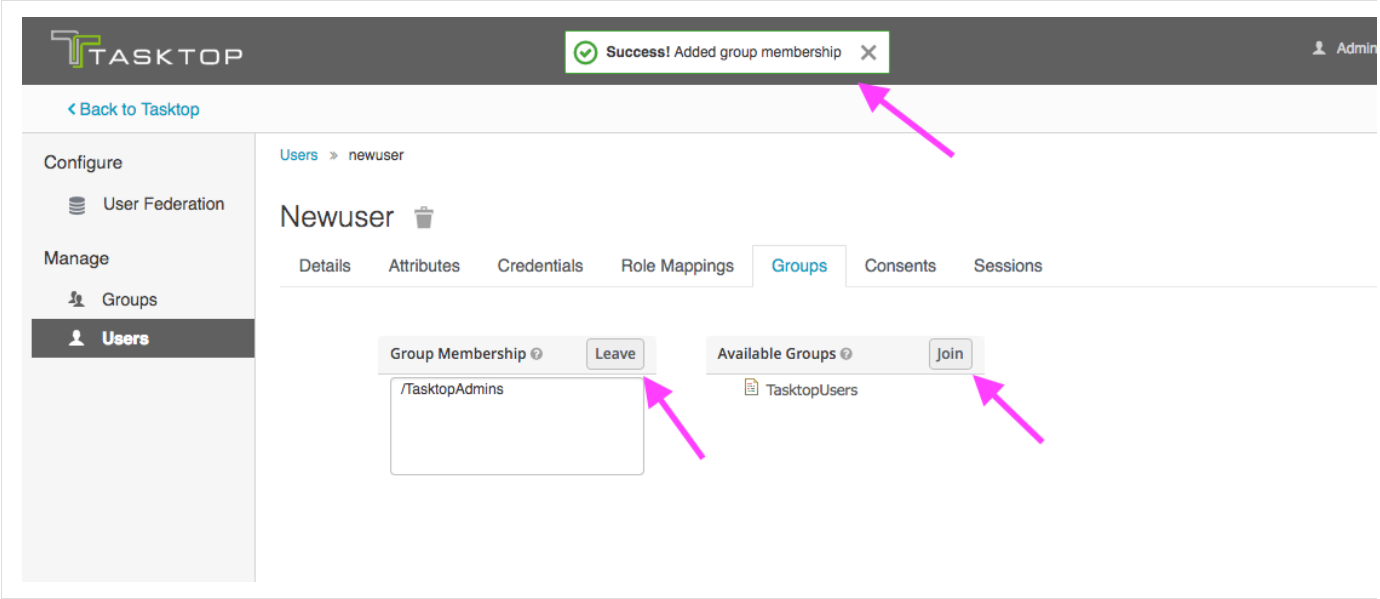
Adding or Removing Users From a Group

To modify a user's group membership, you must have admin capabilities.

Select 'Users' from the left pane of the User Administration screen. Click 'View all Users' and select the ID of the user you would like to modify.

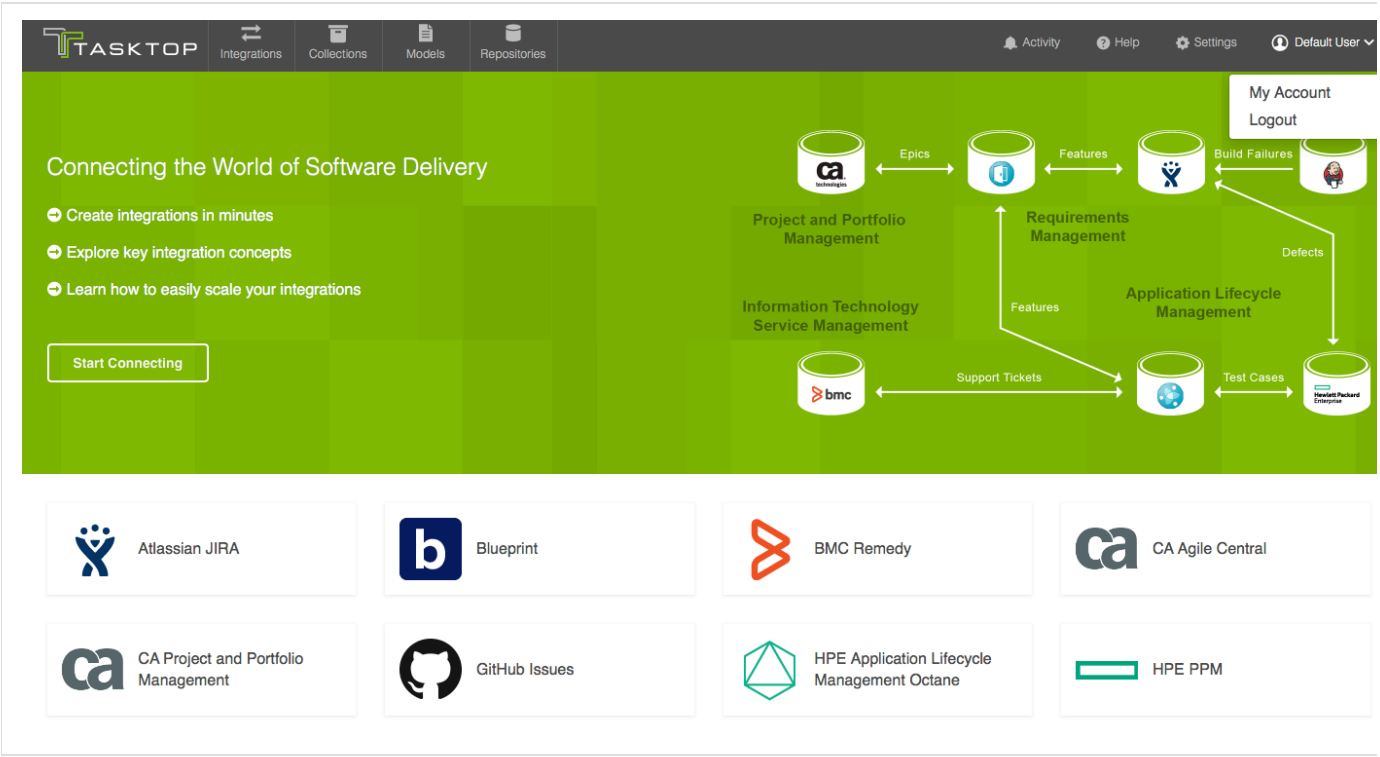
Click on the 'Groups' tab, select the group whose membership you'd like to modify, and use the 'leave' and 'join' buttons to modify their group membership. There is no saving necessary here; once you click the 'leave' and/or 'join' button, you will see a notification at the top of the screen letting you know that your change has been made.

⚠️ Note that a user must be a member of at least one group in order to be able to log into Tasktop successfully.

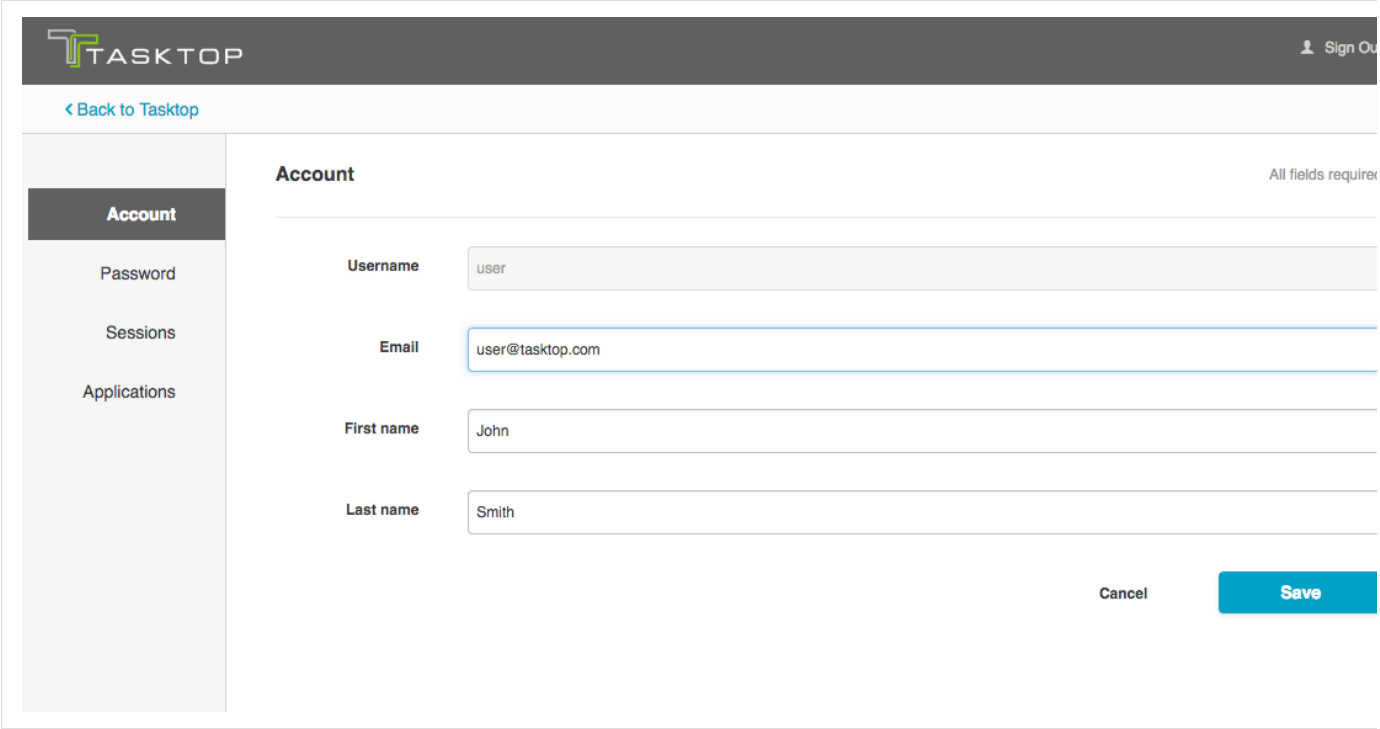


Modifying Your Own User Information

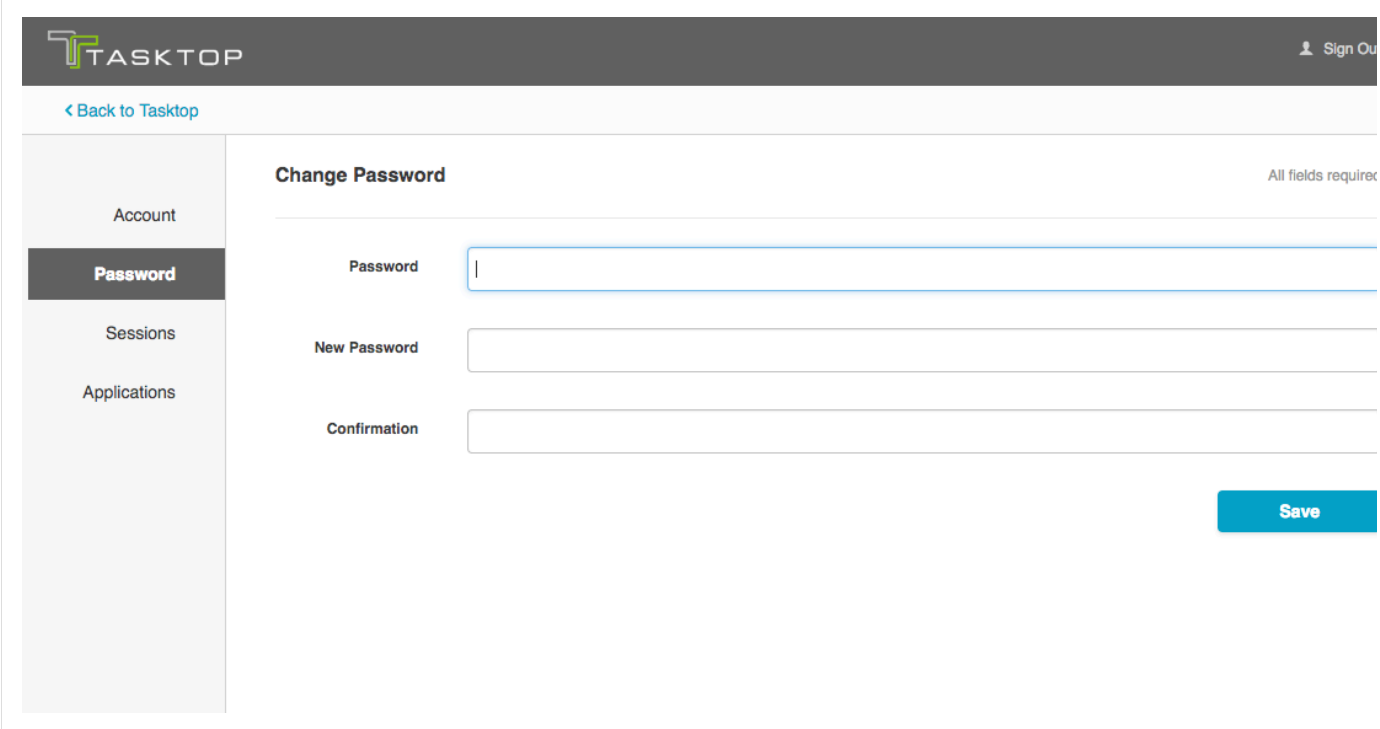
Both Users and Admins can modify their own account information. To change your own password or other user information, right click your name at the upper right corner of the screen, and select 'My Account.'



This will bring you to the Account Info screen, where you can update your name or e-mail address:



You can also click 'Password' on the left sidebar in order to change you password:



The 'Sessions' and 'Applications' sections can be ignored.

Advanced User Management

⚠️ *Note: Advanced User Management is available only for Enterprise and Ultimate Editions. Advanced User Management is currently available for Pro users for a limited time, in preview mode.*

Tasktop Integration Hub has some advanced user management capabilities not accessible via the Tasktop

Integration Hub interface.

To access advanced user management capabilities, go to http://<my-tasktop-url>.com/auth and then click on the 'Administration Console' link.

The default username and password of the user management super-user is root/Tasktop123. We highly recommend changing this password upon install.

⚠ WARNING: there is only one initial root user. If the credentials for this user are lost, access to the advanced User Management features will be lost. All functionality of Tasktop Integration Hub, however, will continue uninterrupted.

Some of the advanced features include:

- User Federation Configuration for:
  - LDAP
  - Kerberos
- Identity Provider login for:
  - SAML v2.0
  - OpenID Connect v1.0
- Social Login for:
  - Google
  - LinkedIn
  - GitHub
  - Facebook
  - Twitter
  - Microsoft
  - StackOverflow
- Enforcing custom password policies such as:
  - Set password expiration
  - Require special characters
  - Setting minimum password length
  - Etc

To learn more about these advanced features, go to https://keycloak.gitbooks.io/server-adminstration-guide/content/ or http://www.keycloak.org/documentation.html

⚠ WARNING: Do not make changes or updates to the Roles or Groups section. Altering these settings may prevent your Tasktop Integration Hub users from accessing the tool.

Key Concepts

Tasktop: Apex Release (17.1)

- Integration
- Repository
- Artifact
- Collection
  - 1) Repository Collection
  - 2) Gateway Collection
- Model
- Flow Specification and Templates

    - Integration Style
    - Canvas Layout
- Containment
- Artifact Relationship Management (ARM)

Tasktop is a powerful tool for connecting your software delivery systems to empower teams, enhance communication, and improve the process of software development as a whole. Below is a look at some of the concepts Tasktop utilizes to facilitate integration.

The key concepts to understand are:

- Integration
- Repository
- Artifact
- Collection
- Model
- Flow Specification
- Template

You can learn more about these concepts in the short video below:

Integration

At the highest level, the definition of an integration is simply the flow of information between 2 or more systems. If you dig a little bit deeper, the definition of an integration is the flow of information, defined by the flow specification, between two or more collections. And collections are sets of artifacts. But that is probably too much to swallow right at the beginning – so don't try to! Take a look at a conceptual picture of what an integration looks like in Figure 1 below, and just keep that in mind as we walk through all of the other concepts – then when you come back to this it will make a lot more sense!

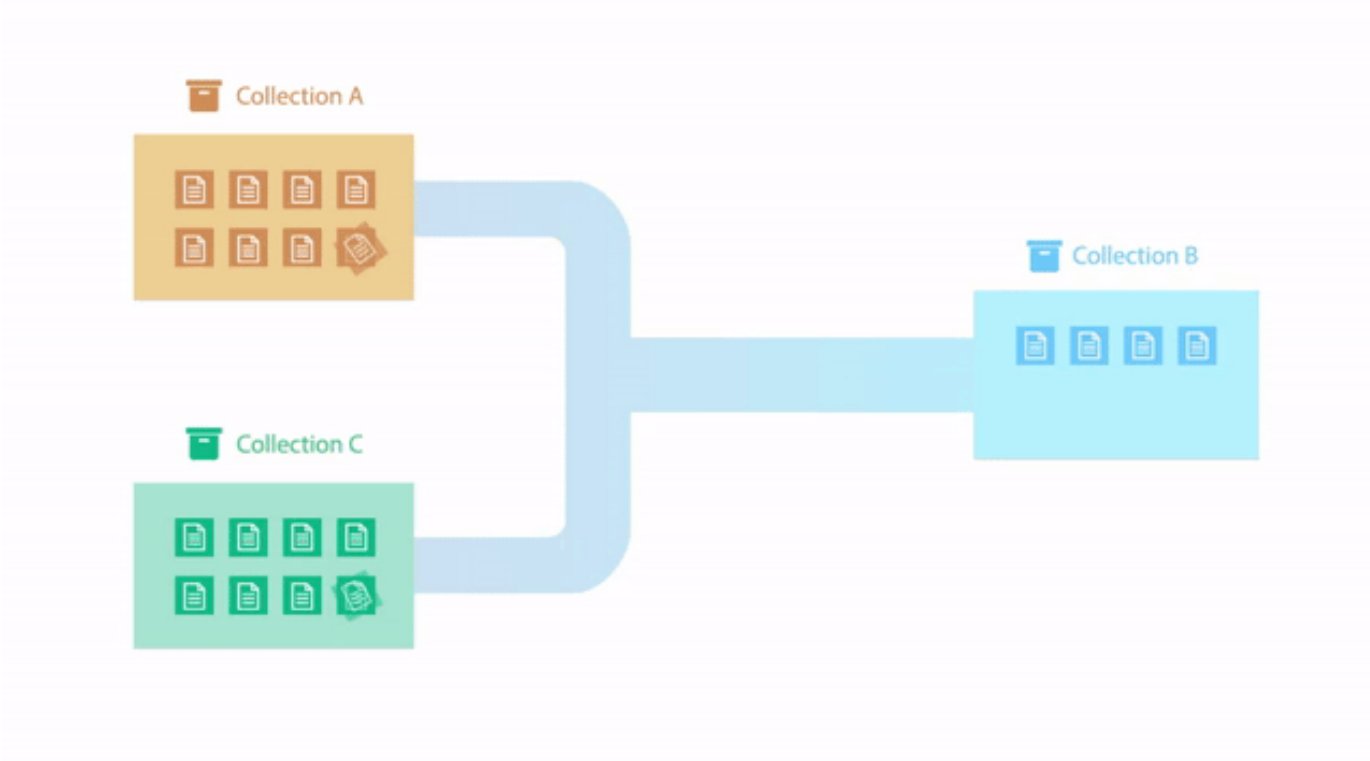
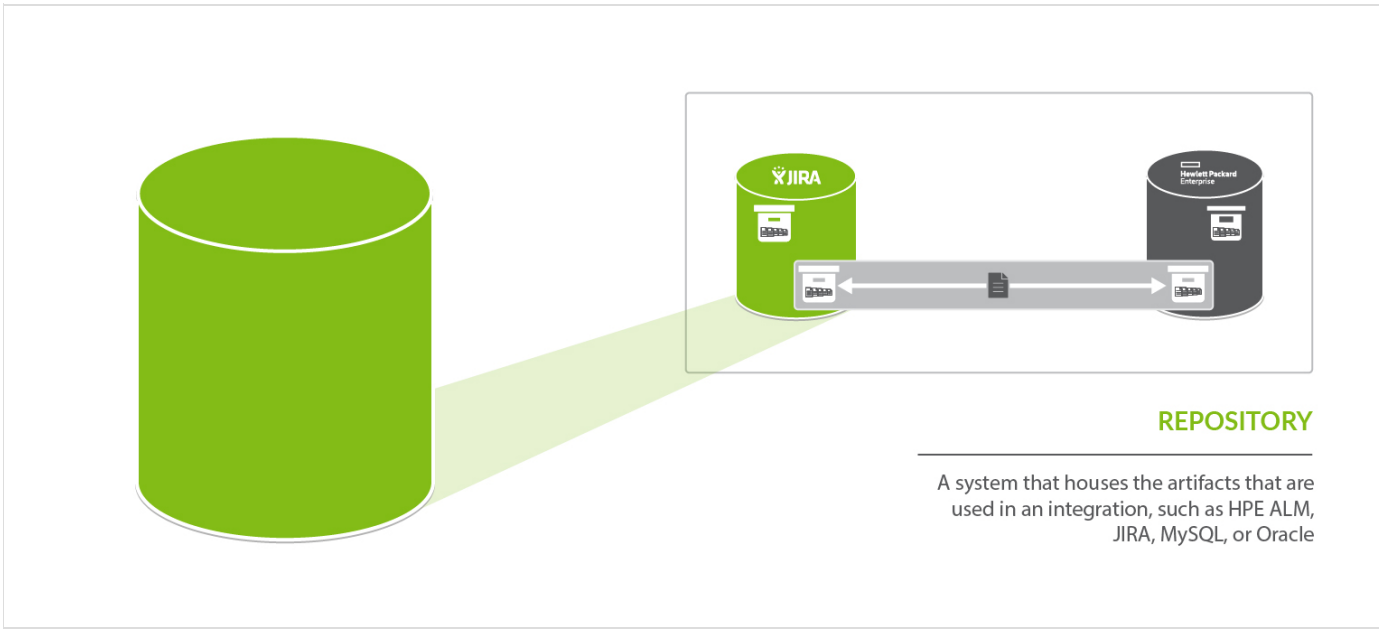


Figure 1: Integration Flow

So let's first talk about the underpinnings of how Tasktop communicates with end systems, which we call *Repositories*. For all repositories Tasktop connects to, we create what we call a *Repository Connection*. Once we've introduced those concepts we'll talk about *Artifacts* and *Collections* and then we will come back to *Integrations* and talk more about the *flow specification*.

Repository

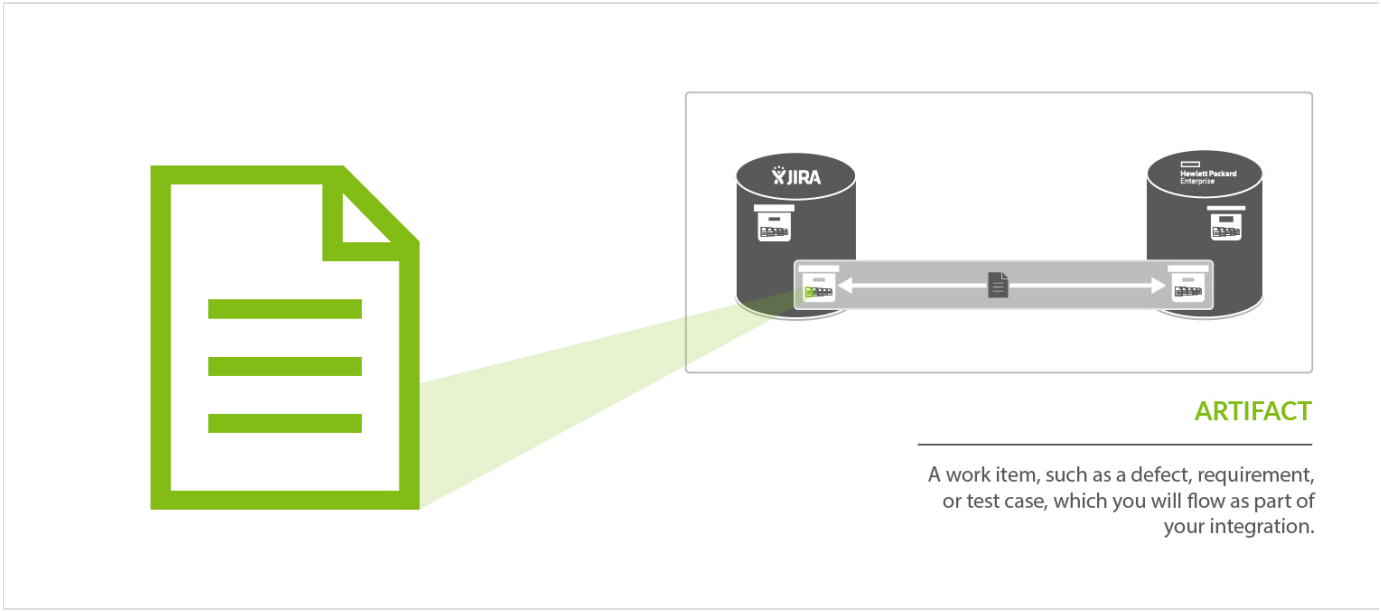


A *repository* is any system that houses the artifacts that can be used in an integration. Repositories can be systems used as part of the software delivery process, like *HPE ALM, CA Agile Central, JIRA,* etc., or repositories can be more generic databases, like *MySQL* or *Oracle*.

A *repository connection* is a connection to a specific instance of a given repository that permits Tasktop to communicate with that repository. To configure a *repository connection*, users will need to provide base credentials such as a server URL, a username, and a password.

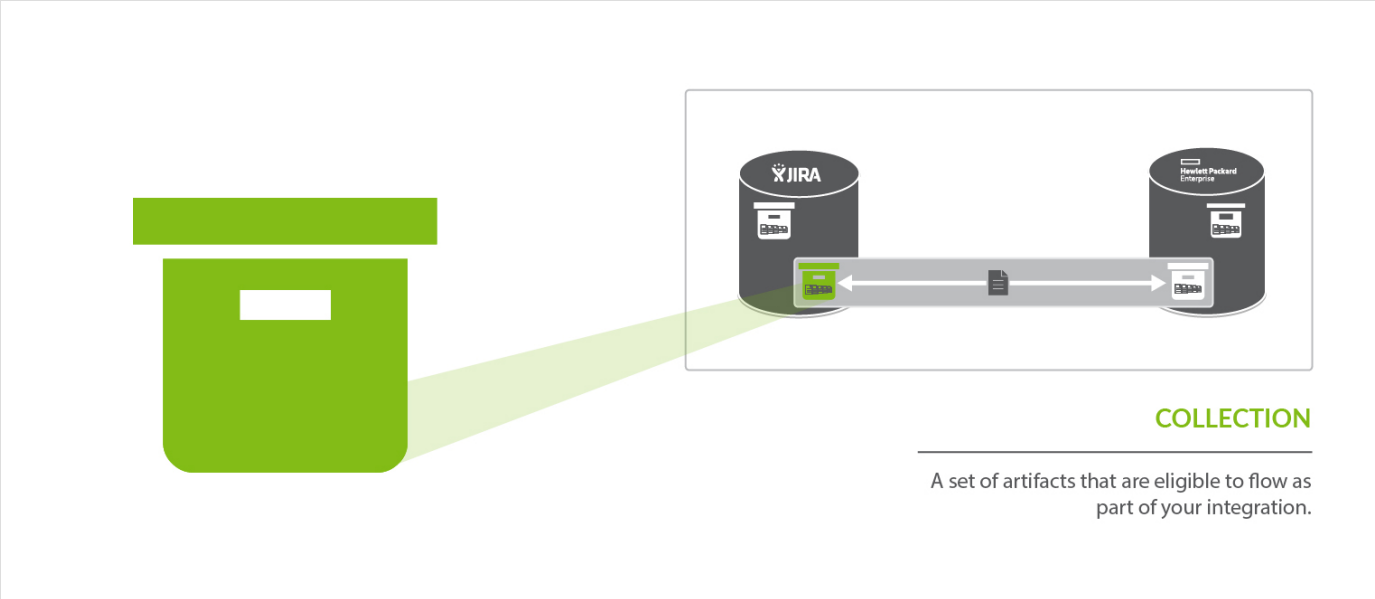
You can learn how to set up a *repository connection* here.

Artifact



Some examples of common artifacts are defects, stories, requirements, test cases, and help tickets, to name just a few.*Artifacts* are the work items, such as defects, requirements, or test cases, that are produced by different teams during software development. Artifacts are the core items which will flow as part of your integration. Serving as the core currency of communication, artifacts are the means by which all the work around software production is recorded and tracked. Artifacts are at the core of any integration and are the entities that Tasktop can create or modify as a part of an integration.

Collection



A *collection* is the set of artifacts that are eligible to flow as part of your integration. They have the following characteristics:

1. All artifacts in the collection are the same core artifact type (e.g. defect, user story, feature, etc)
2. The artifacts in the collection are mapped to one model
3. Artifacts can be sourced from multiple projects

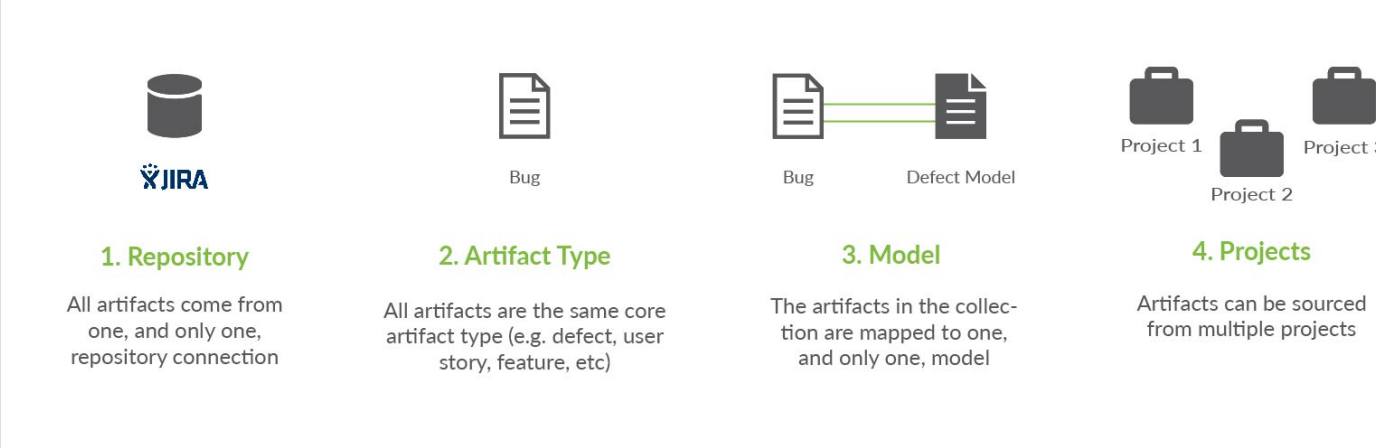


Figure 2: Collection Definition

A concrete example of a collection would be a set of defects from an organization's *JIRA* instance.

The artifacts in a collection can come from one or more projects from a given repository connection. Getting back to the example provided, if your JIRA instance had 50 projects, you could include artifacts from any or all of those projects. Once projects are added to a collection, those artifacts are eligible for inclusion in an integration.

(Note: The term "project" is used here generically– sometimes repositories have different names for "project", or may not have more granular projects at all, but let's stick with this for simplicity's sake.)

The artifacts in a collection share a set of fields that have repository-specific names and values. Part of creating a collection involves choosing a model on which to base the collection and then mapping these

repository specific fields and values to those defined in the model. The concept of models will be discussed in the next section.

There are two types of collections in Tasktop: *Repository Collections (*which include collections from typical repositories, such as *JIRA* or *HPE Octane*, as well as *Database Collections*, which connect to databases such as *MySQL*) and *Gateway Collections*.

You can learn how to create your collection(s) here.

1) Repository Collection

A) Standard Repository Collections

Standard Repository Collections comprise artifacts from an ALM, PPM, or ITSM repository like *Atlassian JIRA*, *HPE ALM*, *CA Clarity*, or *Zendesk*. When used in an integration, artifacts in a repository collection can be created, can be updated, and/or can trigger the creation of artifacts in another collection.

What can Tasktop do to artifacts in a repository collection?

Action	Permissable
Create artifacts in collection	✅
Update artifacts in collection	✅
Detect additions or updates to artifacts in collection in order to create or update artifacts in another collection	✅

B) Database Collections (a type of Repository Collection)

Databases collections, a type of Repository Collection, comprise artifacts from a database repository like *MySQL*, *Oracle*, or *MS SQL Server*. When used in an integration, artifacts in a database collection can be created, but cannot be updated nor trigger the creation of artifacts in another collection.

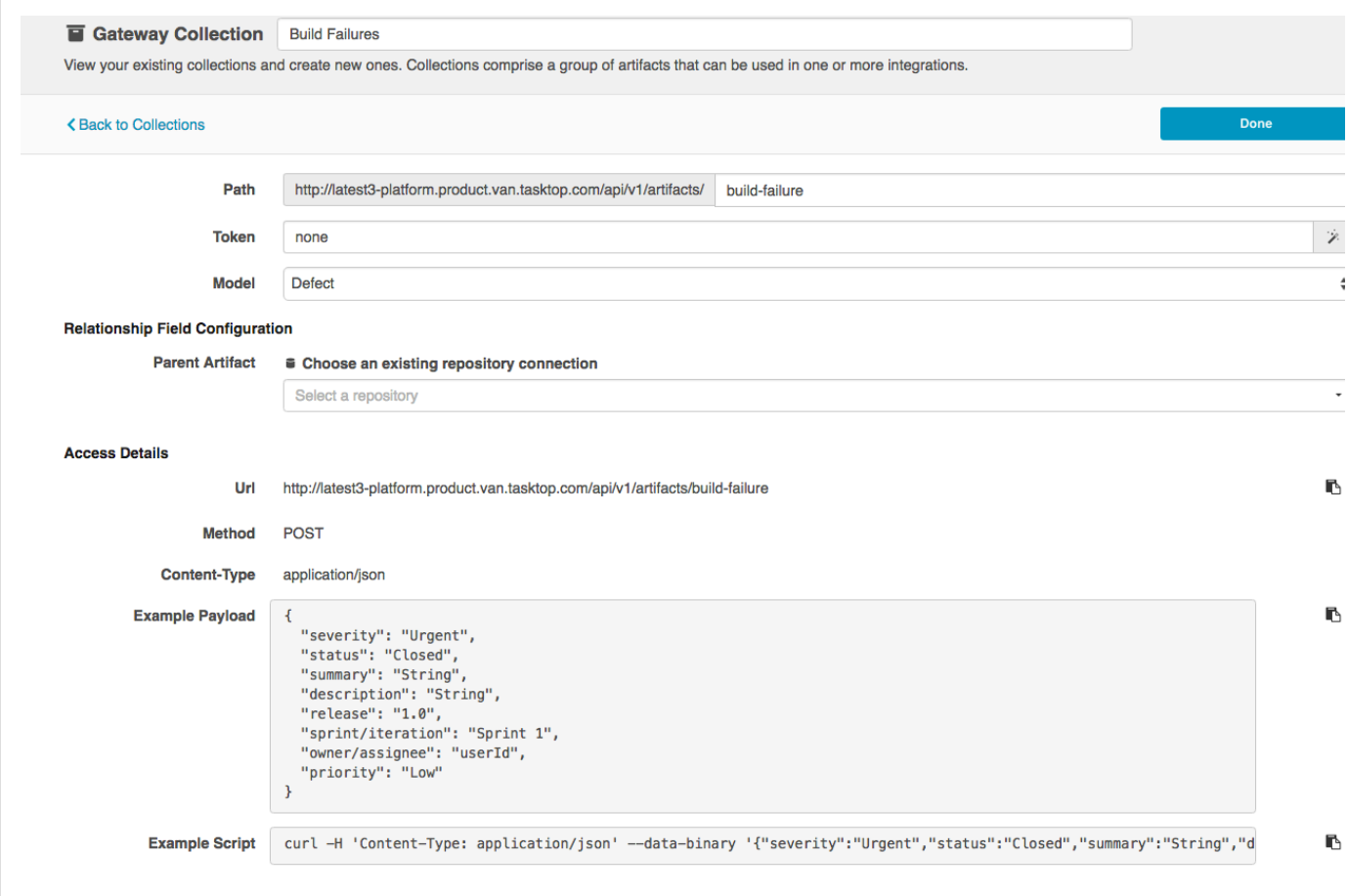
What can Tasktop do to artifacts in a database collection?

Action	Permissible
Create artifacts in collection	✅
Update artifacts in collection	❌
Detect additions or updates to artifacts in collection in order to create or update artifacts in another collection	❌

2) Gateway Collection

Unlike repository collections and database collections, which rely on Tasktop actively making various API calls to communicate with a given repository, artifacts in a Gateway collection are sent to Tasktop via our own REST API. This means that you don't need to create a repository connection to create a gateway collection--as long as you can send Tasktop a simple REST call, those artifacts can then be used to achieve a specific goal within the context of an integration.

Gateway collections are particularly useful when the artifacts you want to integrate come from smaller, purpose-built systems for practitioners in various disciplines, such as Selenium for QA; when the artifacts you want to integrate come from systems that are largely event-driven, such as an application performance monitoring repositories; when artifacts come from home-grown tools your organization might have developed on their own; or when you'd like to pull information that is not considered a standard artifact from a repository supported by Tasktop, like capacity information from a PPM tool. When creating a gateway collection, you'll specify a path to generate a webservice to which you'll post information. You'll also choose the model to which you would like incoming artifacts from this collection to conform. You'll then be given an example payload and script that can be used to send artifacts to Tasktop:

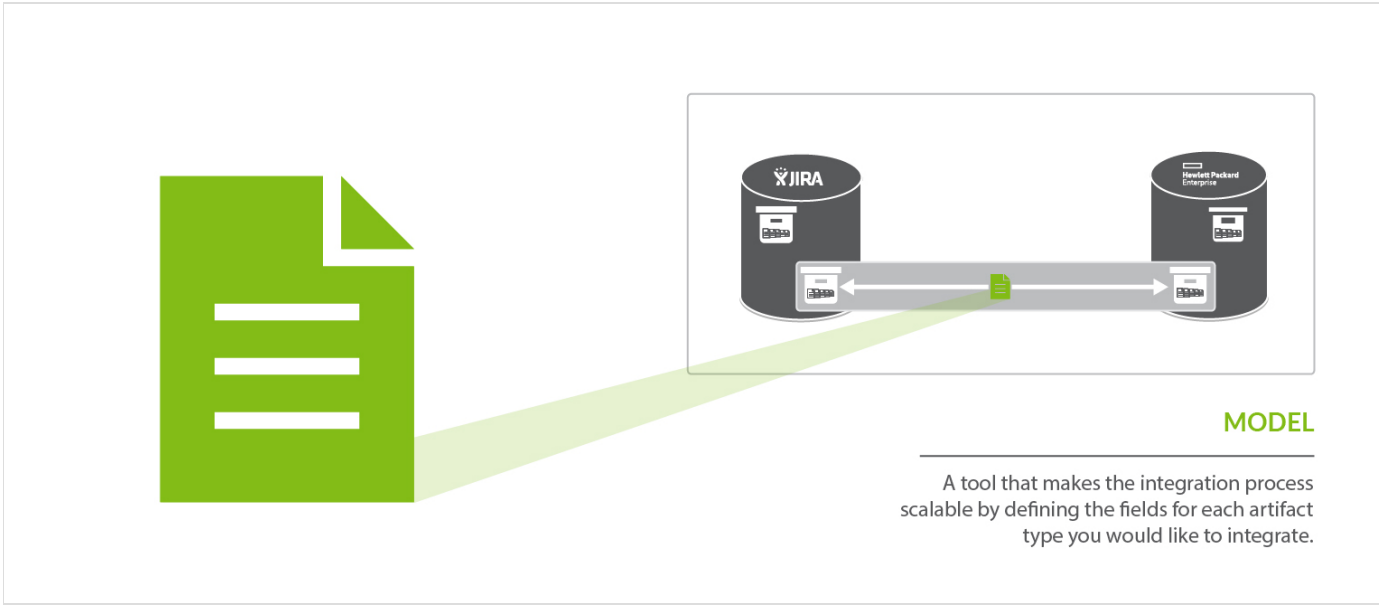


When used in an integration, artifacts in a gateway collection can trigger the creation or modification of artifacts in another collection.

What can Tasktop do to artifacts in a gateway collection?

Action	Permissable
Create artifacts in collection	✗
Update artifacts in collection	✗
Detect additions or updates to artifacts in collection in order to create or update artifacts in another collection	✓

Model



When integrating data from multiple collections, there are three factors that are critical to success:

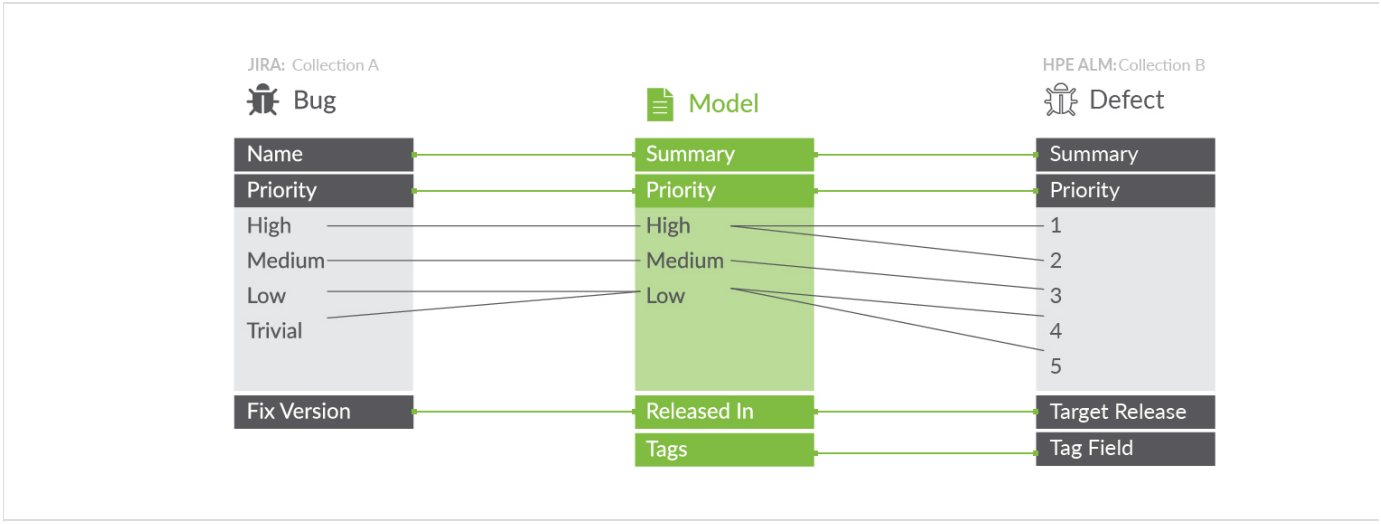
1. The ability to normalize disparate definitions of artifacts between different collections
2. The ability to scale the integrations to support many collections with hundreds or even thousands of projects and artifacts.
3. Efficient flow of data – meaning, only flow information that is necessary between collections

These three critical success factors are met with our usage of "*models*". In very basic terms, a model is simply a list of fields or attributes that define a certain artifact that you want to integrate. For example, below is a very basic defect model:

Defect Model	
Field	Field Type
Description	String
Priority	Single Select: • High • Medium • Low
Status	Single Select: • New • In Progress • Complete

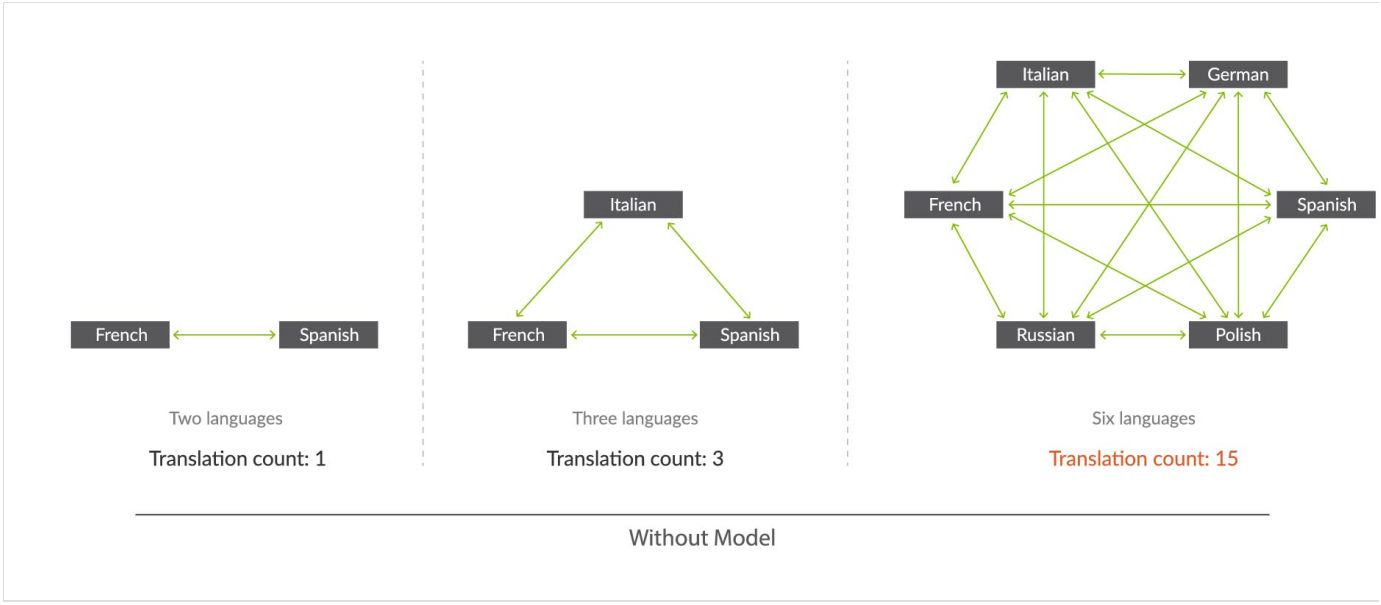
Let's talk about the first critical success factor – the ability to normalize disparate definitions of artifacts between different collections. Or, another way of thinking of it, the classic *"you say tomato, I say tomahto" conundrum*. In the diagram below it is apparent that the JIRA bug is similar, but not the same, as the HPE

ALM defect. The solution to this problem is to be able to “map” each defect to a common definition of a defect and “normalize” the fields and field values. Then, when you are communicating about “defects”, everyone is speaking the same language via the “*model*” definition. Like this:

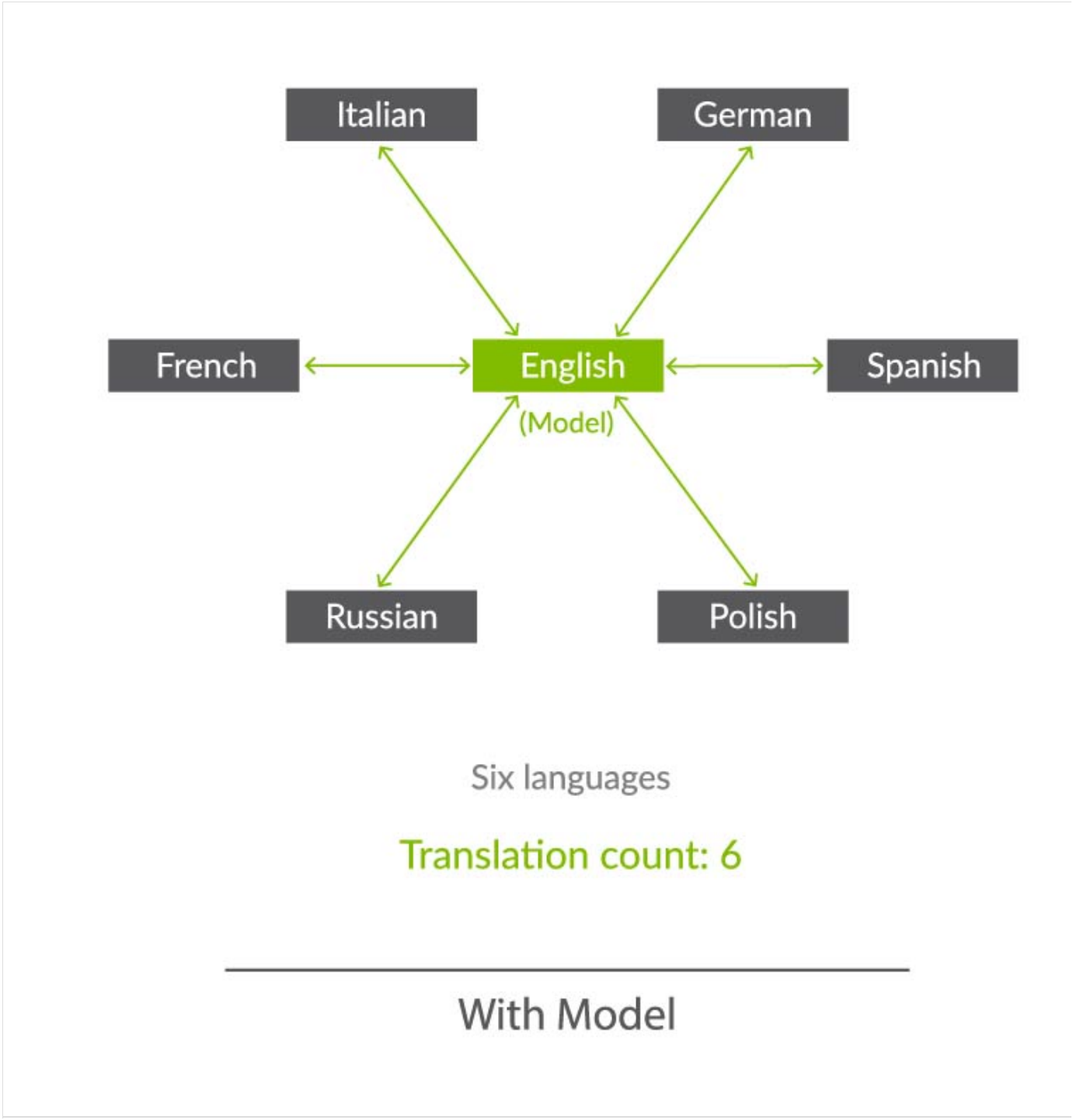


A good analogy to help understand why models are so important is the act of translating between people who speak different languages. If you have two people that speak two different languages, you need to translate only between those two points. If, however, you have three different languages, you have three points of disconnect in communication that need to be translated. But, as you add more and more languages, the number of disconnects blocking communication does not grow linearly – even if you have just 6 languages, you have 15 points of disconnect to translate between! And if you have 10 languages you will have 45! As you can see, resolving these point-to-point disconnects individually quickly becomes unsustainable given the sheer number of them that can arise. It is in this way that models save the day, acting as a “universal translator,” overcoming all of the communication disconnects that are present by translating between all of the points at once.Now that we have the ability to solve the *“you say tomato, I say tomahto" problem*, the second critical success factor comes into play, which is the desire to *scale your integration landscape* to support many collections with hundreds or even thousands of projects and artifacts.

Integrating Without Models



Integrating With Models



Now that we've solved the first two critical success factors, there is one more that might not seem as obvious but is actually quite important to your overall success. When flowing large volumes of data, you need *efficient flow of data*, not the 'drink from the firehose' approach where all fields of all artifacts are flowing everywhere. There is no business value in that and, worse, you will end up with significant performance issues. Instead, by using *models*, you can limit, or target, the exact data that you need to flow between collections – nothing more, and nothing less, than what is necessary.

In summary, models solve the critical three success factors for large scale integration landscapes – giving users the ultimate in flexibility, scalability, and consistency at the same time.

You can learn how to create a model here.

Flow Specification and Templates

Now that we have introduced the concepts of *artifacts*, *collections*, and *models*, we can come back to the concept of an *integration* and discuss a bit more detail about how Tasktop thinks of integrations. As discussed earlier, the basic concept of an integration is the flow of information between two or more collections.

You can learn how to configure your integration here.

The last two concepts to introduce relate to integrations as a whole. First, the *flow specification*. This is probably the trickiest aspect of an integration, which is why we also have introduced another concept, called *templates*, to help. Defining the details of how you want the flow of data between collections to occur has a lot of nuances and details. For instance, do you want to create new artifacts, or modify already existing artifacts? Would you like artifacts and information to flow in both directions or just one direction? What types of collections (and how many of them) would you like to integrate? Once an artifact has flowed to another repository, which project should it go in?

While we provide the ability to handle all of these aspects of "flow specification", we simplify it for users by having templates. Picking a template jump-starts your integration, bundling many of the flow configuration elements to facilitate quicker configuration.

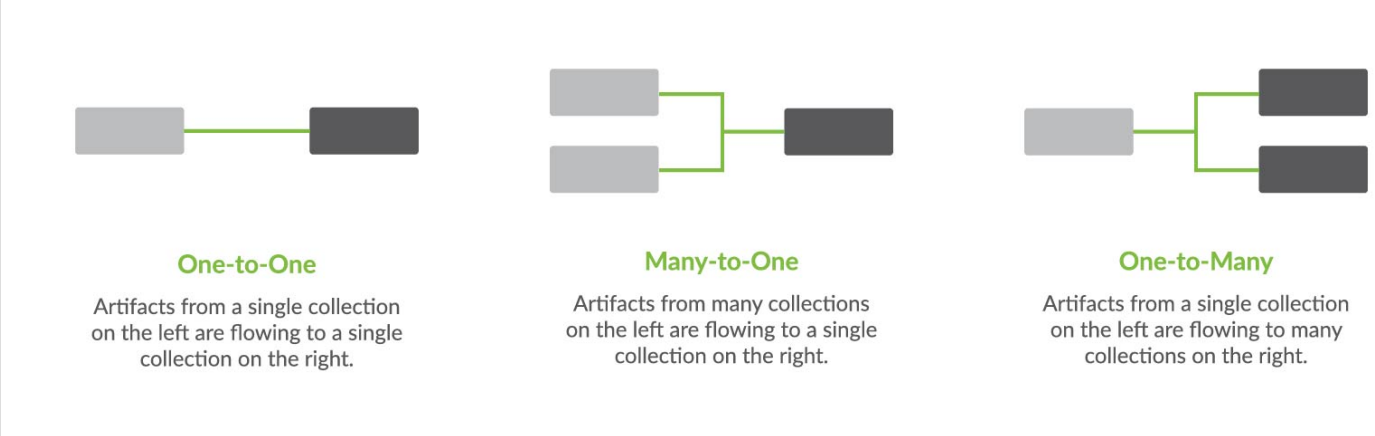
You can learn about each integration template here.

Integration Style

Each *template* is based on an underlying style that defines whether you want to *create new artifacts* in collections or *modify already existing artifacts* in collections.

Canvas Layout

Each template follows a certain canvas layout, determining the quantity and types of collections that can be added to the canvas. The canvas will either follow a many-to-one, one-to-many, or one-to-one layout.



By picking a given template, you are, in essence, also picking the style of integration and canvas layout, which in turn influences other configuration options such as the artifact flow directionality, field flow directionality, and routing directionality, making the act of integrating your collections quick and painless.

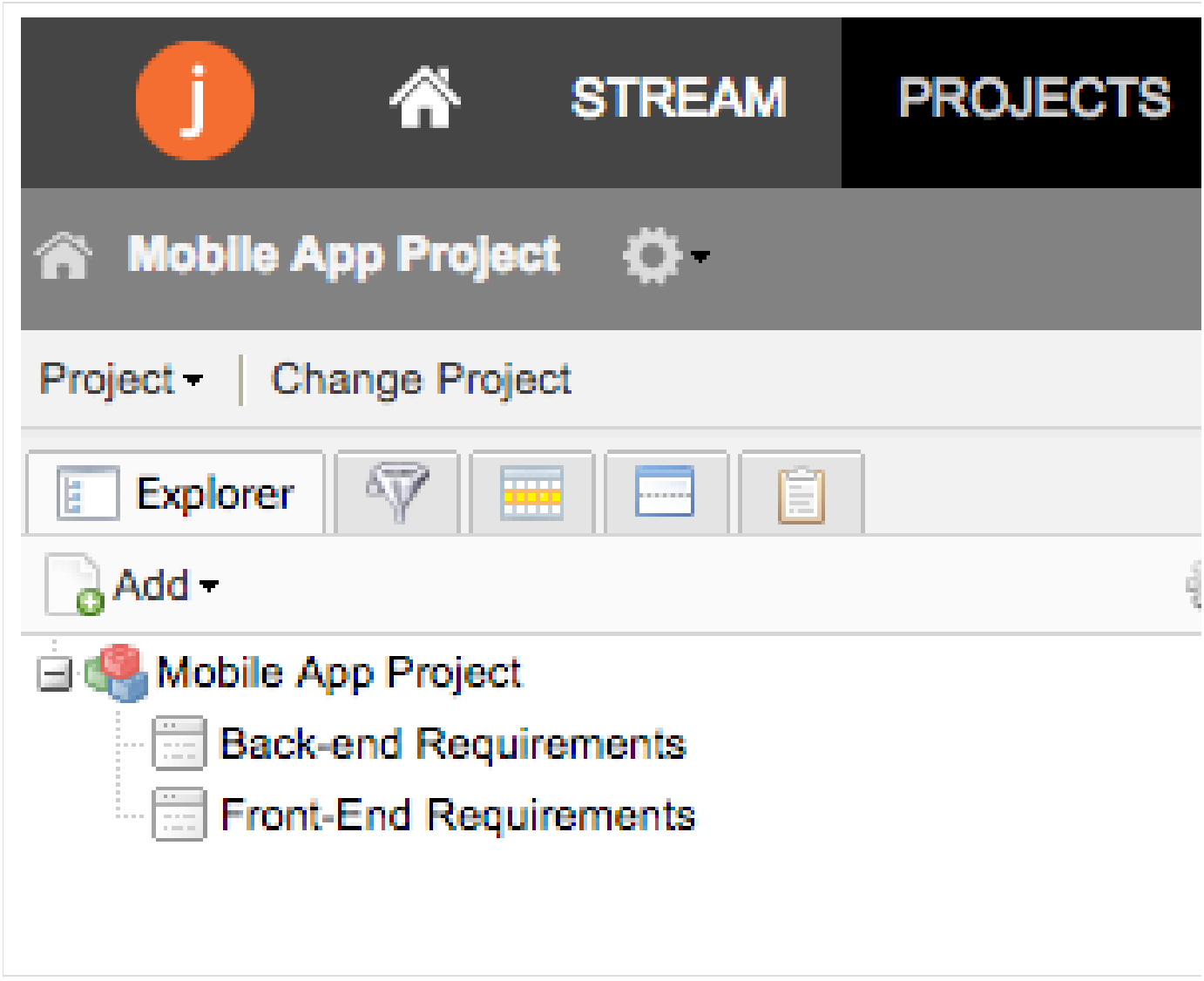
Containment

A container is a tool used to group artifacts. Examples of containers include Projects, Workspaces, Folders, and Sets (in Jama), to name a few. Some repositories contain *high level containers*, such as workspaces, which are then broken into *low level containers*, such as projects.

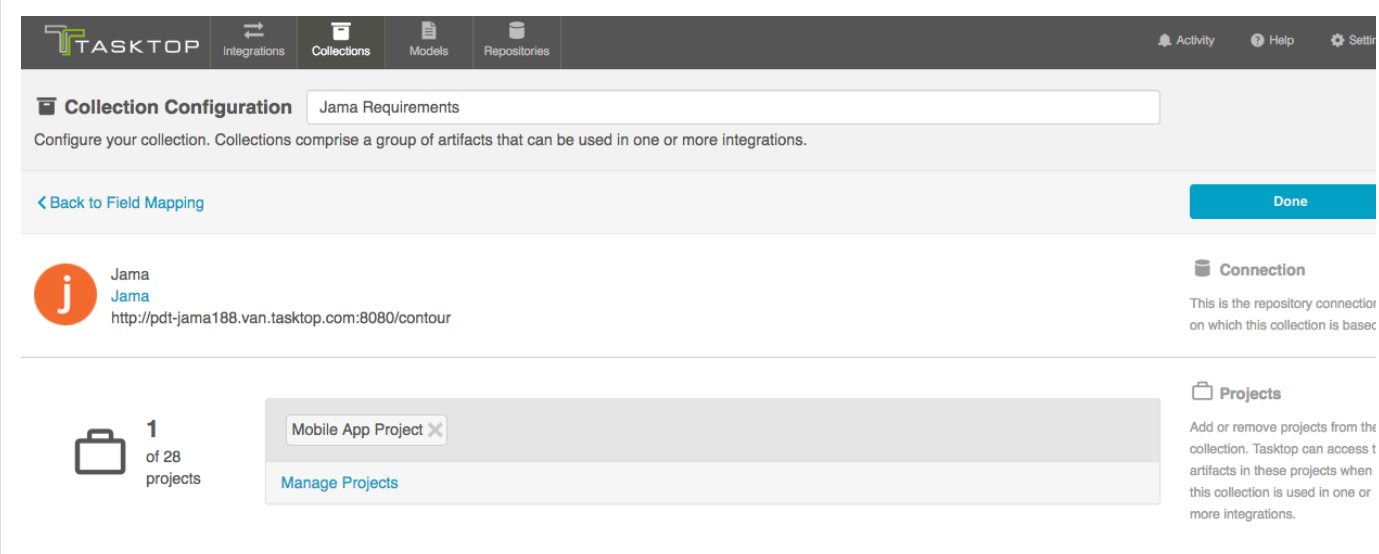
Containers are a key component of creating your collection, as each collection is defined by its artifact type (i.e. defect, requirement, test case, etc), by the model it is mapped to, and by the *high level containers* it includes. In this way, containers are essential for how you define which artifacts can flow as part of your integration.

Your containers also become important during the Artifact Routing stage of configuring your integration. On the Artifact Routing Screen, you are able to determine how artifacts should flow from one collection's containers to the other's. Some repositories allow you to route at only the *low level container* level, some allow you to route at the *high level container* level, and others allow a mixed approach.

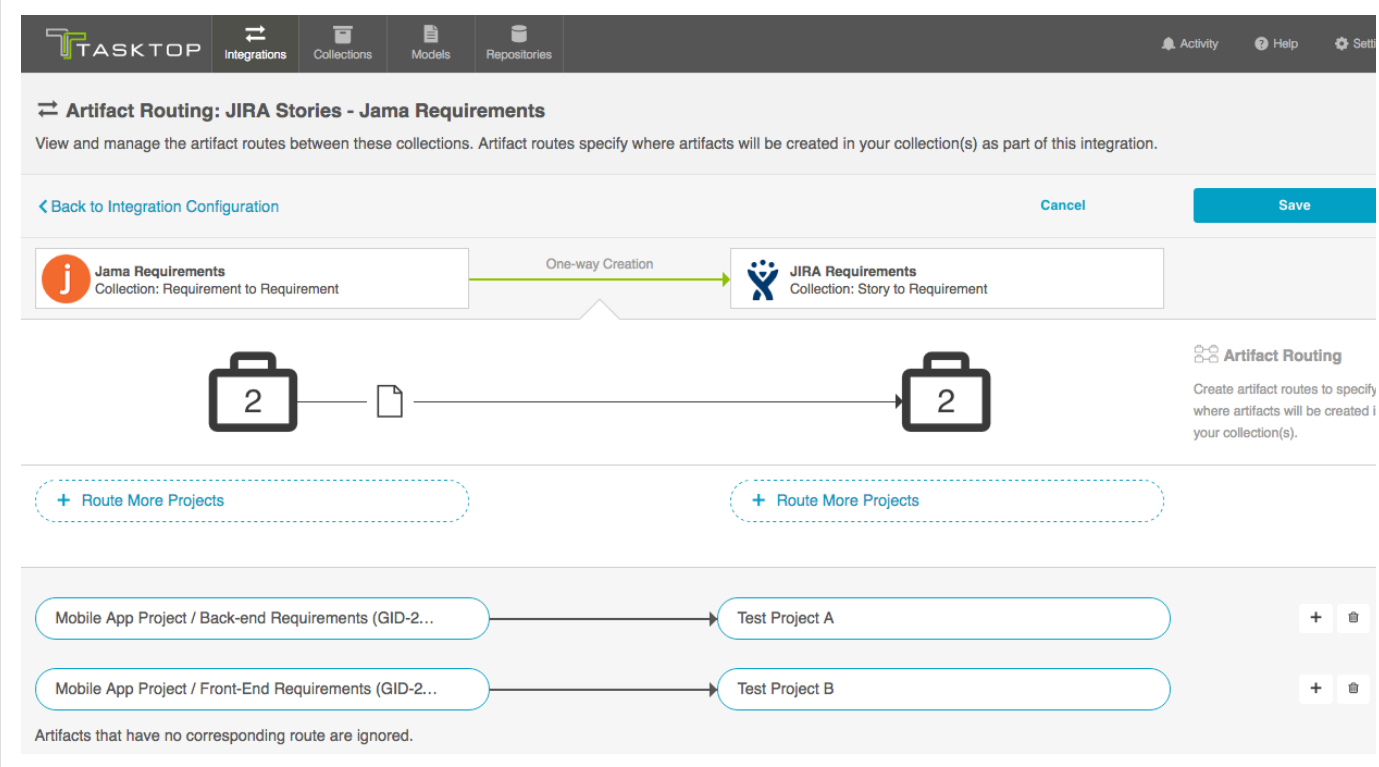
To understand this better, let's look at an example in Jama. Jama contains *high-level containers* (projects) which are then divided into several *low-level containers* (sets). Here, our *high-level container* is the Mobile App Project, which is then divided into two *low-level containers:* the Back-End Requirements set and the Front-End Requirements set.



When we configure our Jama collection, we will define that collection at the *high-level container* level: this means that we can define the collection based on projects. Here, we have selected the Mobile App Project for use in our collection.



However, when routing artifacts, we will utilize *low-level containers* (sets) to determine which container Jama artifacts will flow to in our other repository. In the example below, the Back-end Requirements set in Jama will flow to Test Project A in JIRA, and the Front-End Requirements set in Jama will flow to Test Project B in JIRA. Both the Front End Requirements set and the Back End Requirements set are contained within the high level Mobile App Project.

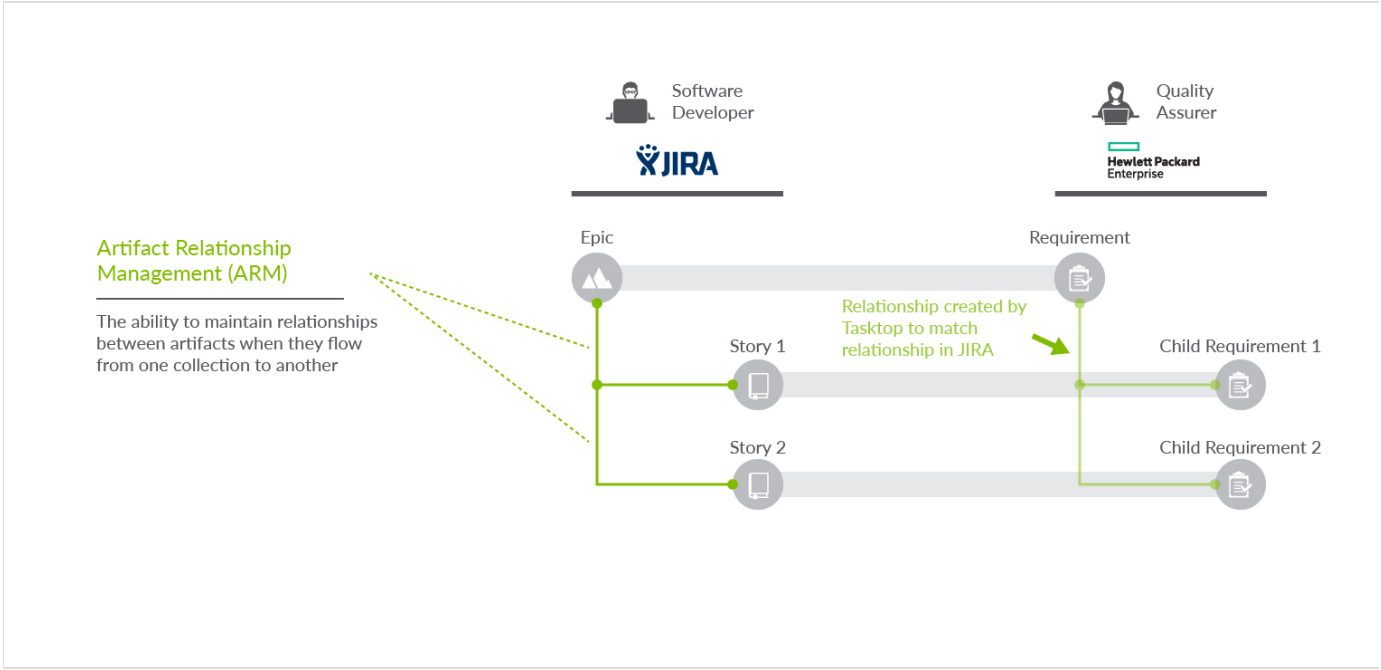


Artifact Relationship Management (ARM)

Artifact Relationship Management refers to the ability to maintain relationships between artifacts when they flow from one collection to another. By utilizing the Relationship Specification Screen when configuring your collection, you can ensure that relationships are preserved between your artifacts. You'll learn more about how to configure Artifact Relationship Management in the Quick Start Guide.

In the example below, you can see an example of an Integration from JIRA to HPE which utilizes Artifact Relationship Management (ARM) to do the following:

- Flow JIRA Epics to HPE Requirements
- Flow JIRA Stories to HPE Child Requirement
- Utilizes Artifact Relationship Management to preserve the relationships between the artifacts in each repository



Quick Start Guide

Tasktop: Apex Release (17.1)

Overview

Setting up a new integration takes four simple steps.

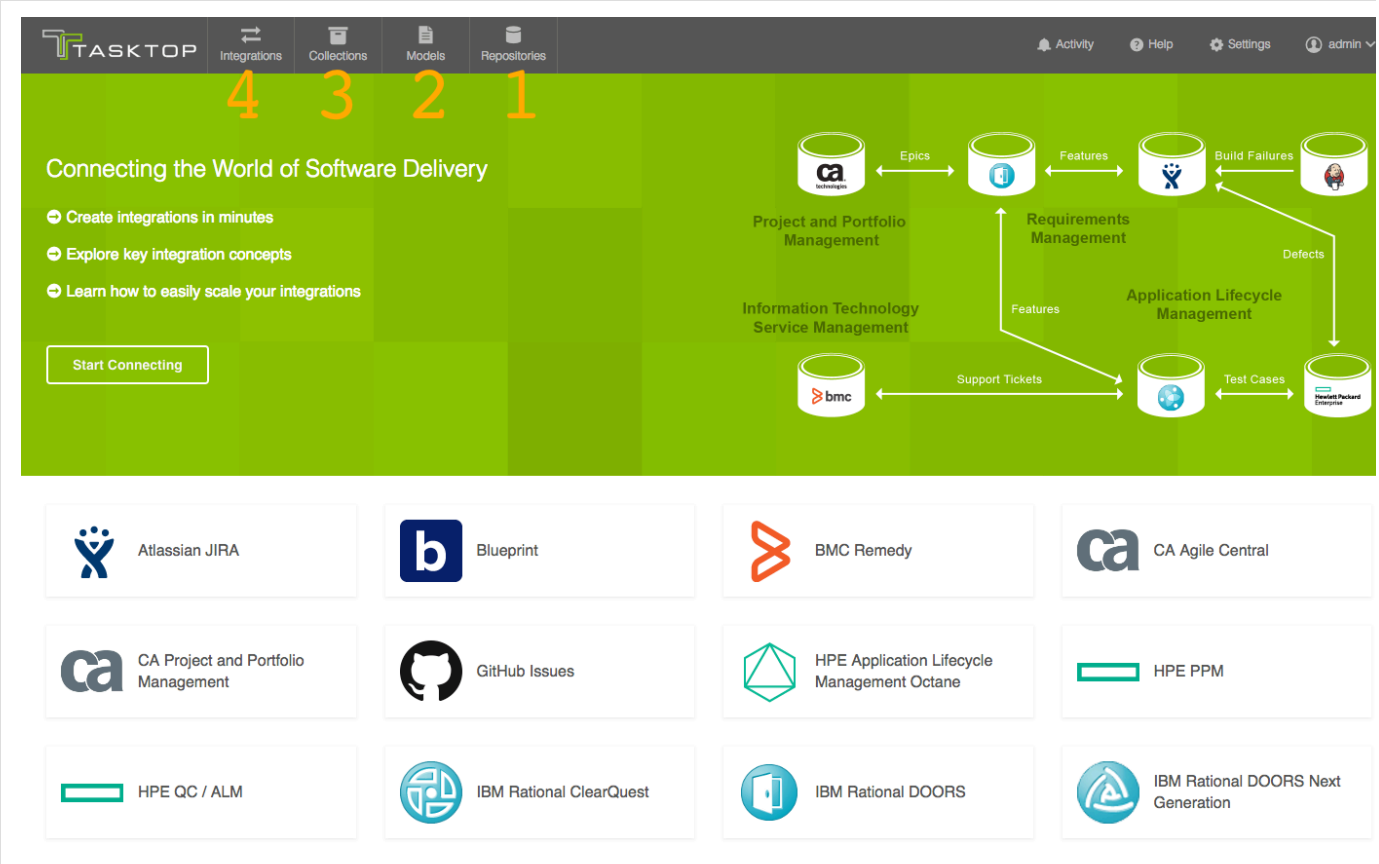
1. Connect to your repository
2. Create a new model or use an existing model
3. Create your collection(s) (which includes mapping your collection to the model you've picked)
4. Configure the integration using one of our out of the box templates

Finally, once you've configured your integration, you can easily expand or modify your Integration.

You can find details on each of these steps within the subpages linked above. You can also see each step

identified in the interface of Tasktop Integration Hub, as shown in the image, below.

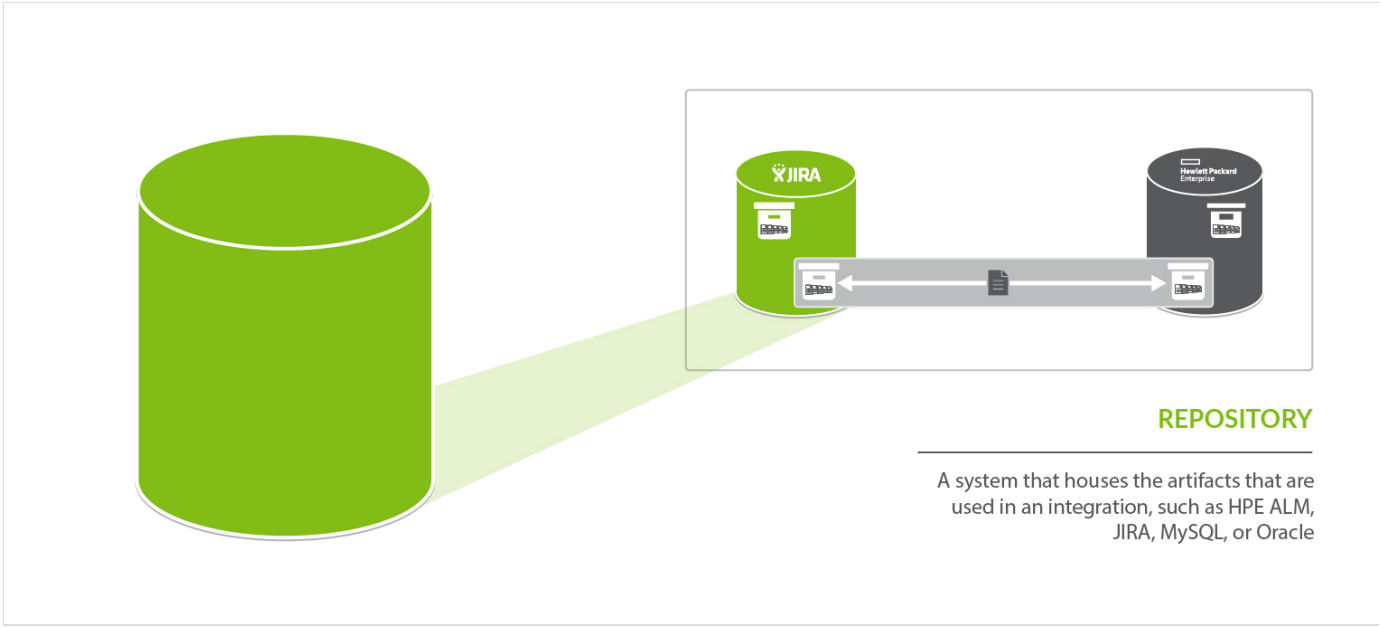
To learn more about each specific integration template (Synchronize, Create via Gateway, Modify via Gateway, and Enterprise Reporting), take a look at the Integration Template section.



Step 1: Connect to Your Repository

Tasktop: Apex Release

What is a Repository?



A *repository* is any system that houses the artifacts that can be used in an integration. Repositories can be systems used as part of the software delivery process, like *HPE ALM, CA Agile Central, JIRA,* etc., or repositories can be more generic databases, like *MySQL* or *Oracle*.

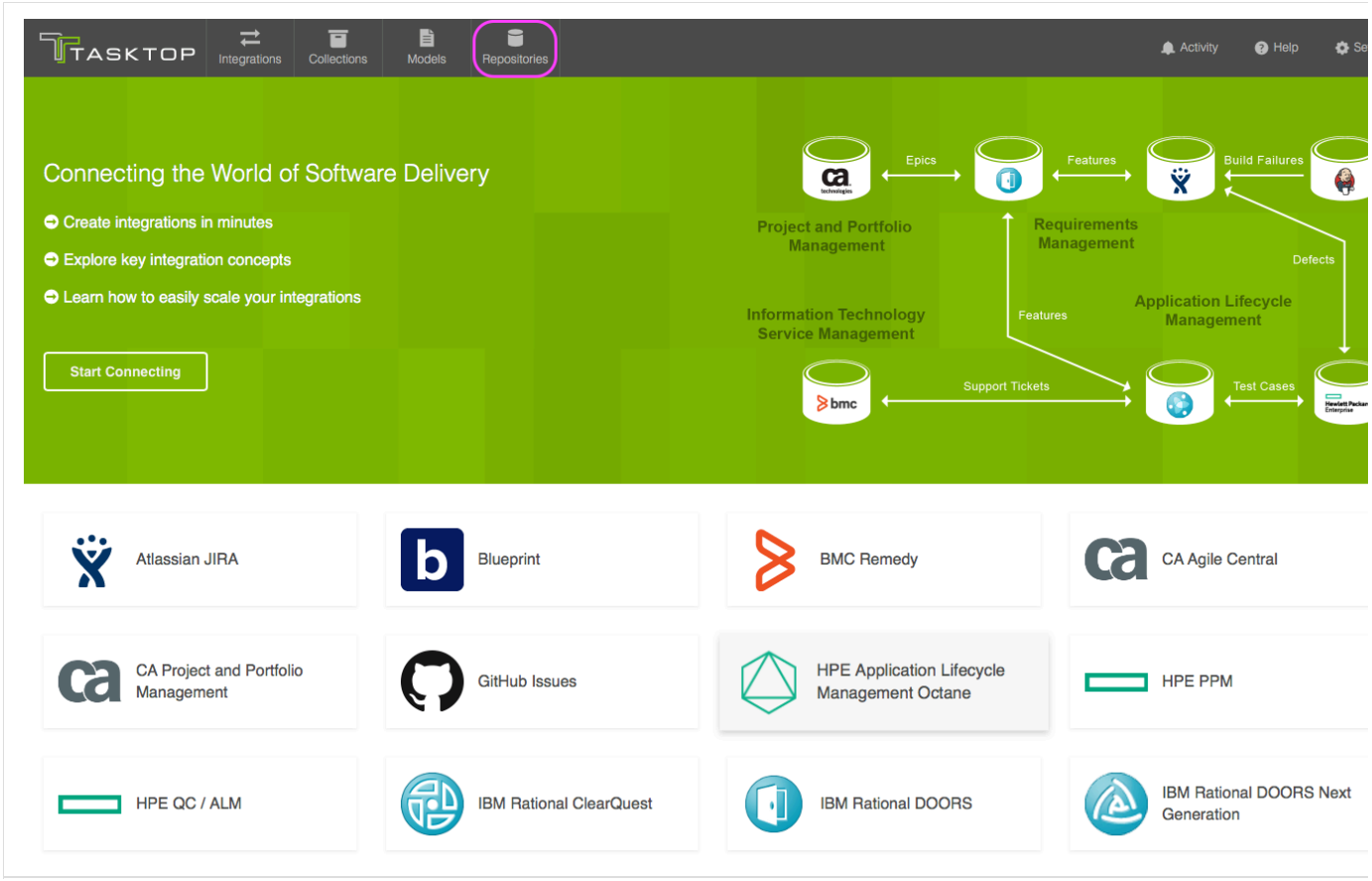
A *repository connection* is a connection to a specific instance of a given repository that permits Tasktop to communicate with that repository. To configure a *repository connection*, users will need to provide base credentials such as a server URL, a username, and a password.

Step 1: Connect to Your Repository

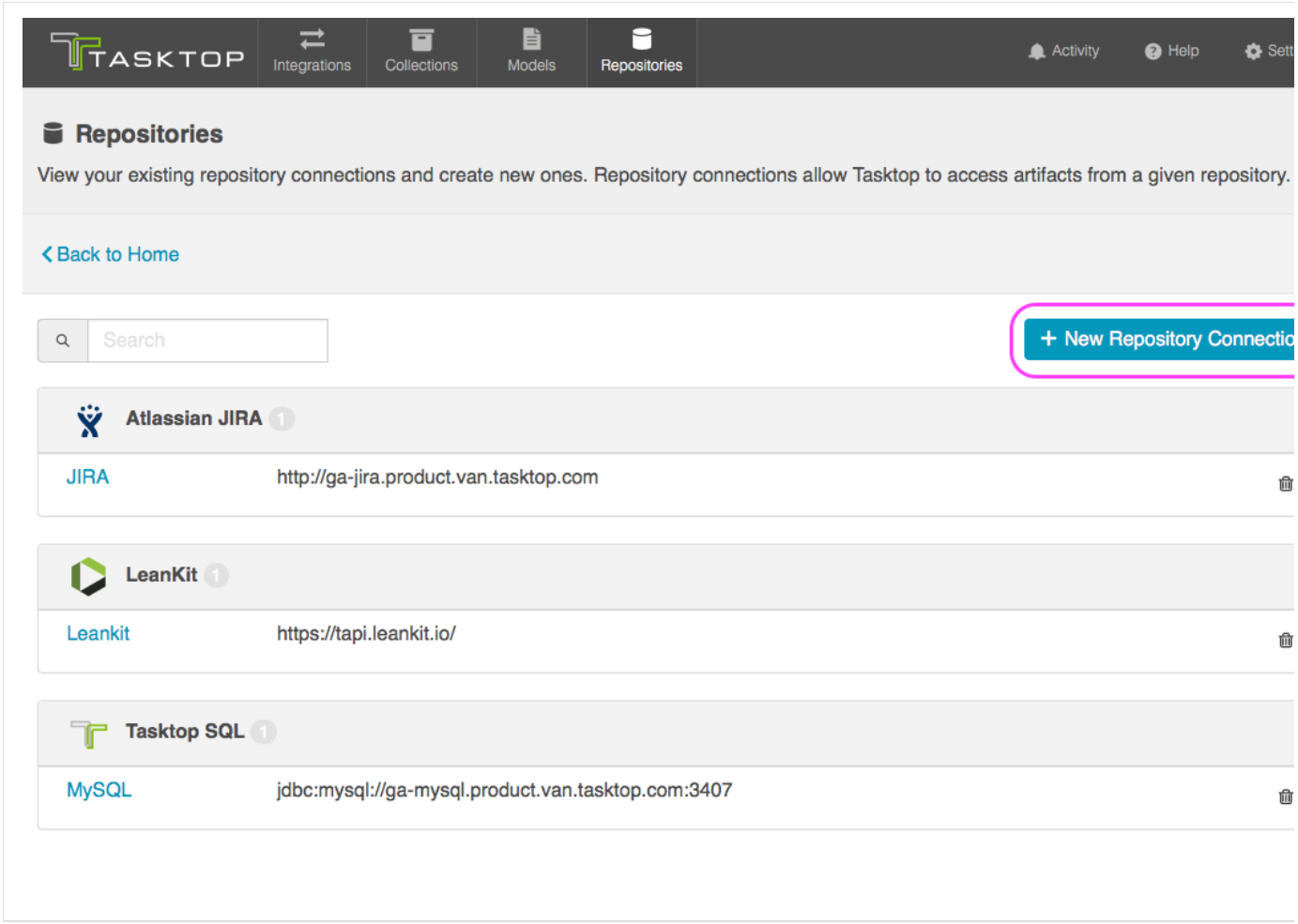
Check out the video below to learn how to create a new repository connection:

Basic Repository Connection

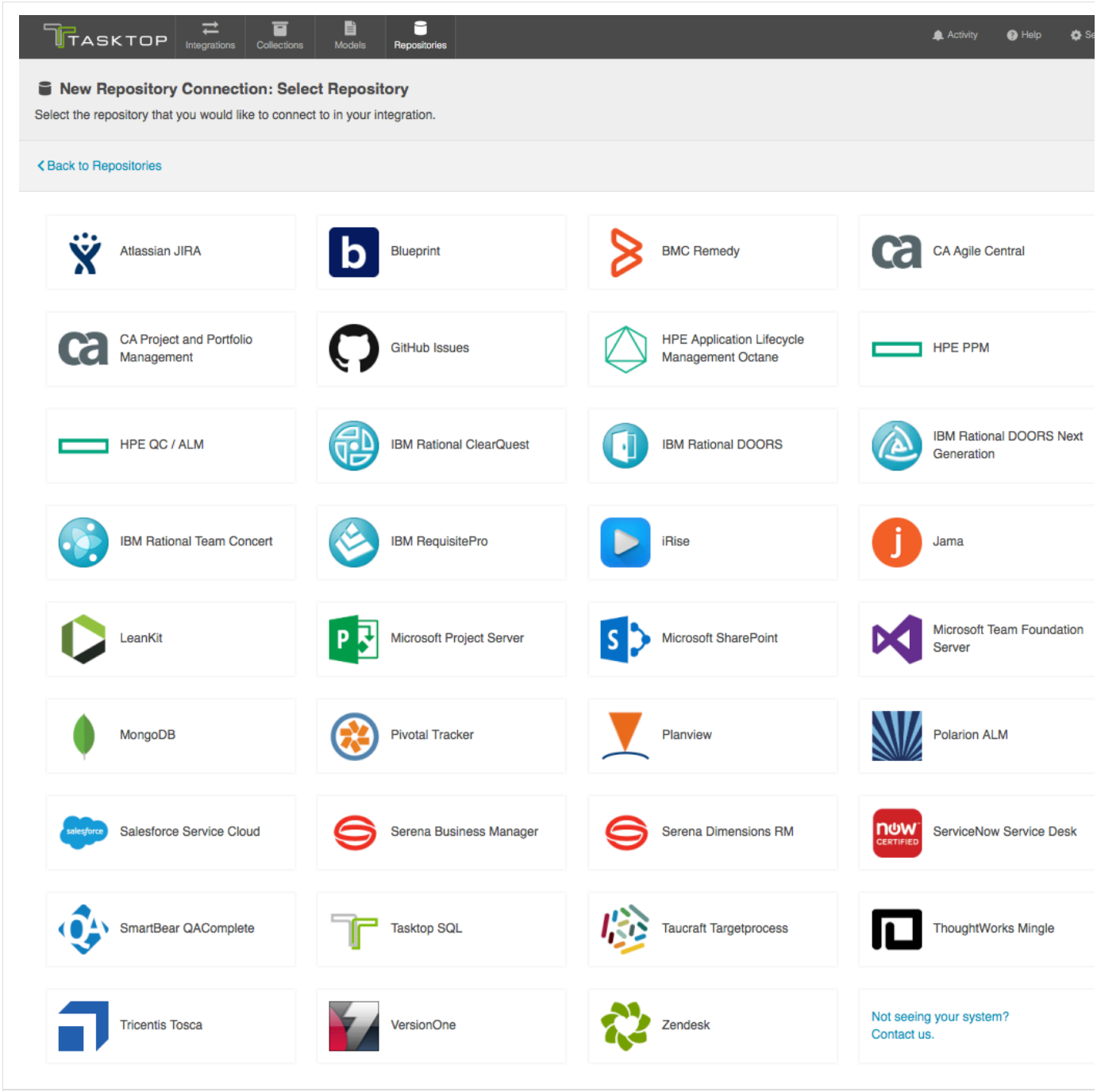
To create a repository connection, select 'Repositories' at the top of the screen



Click the '+ New Repository Connection' button



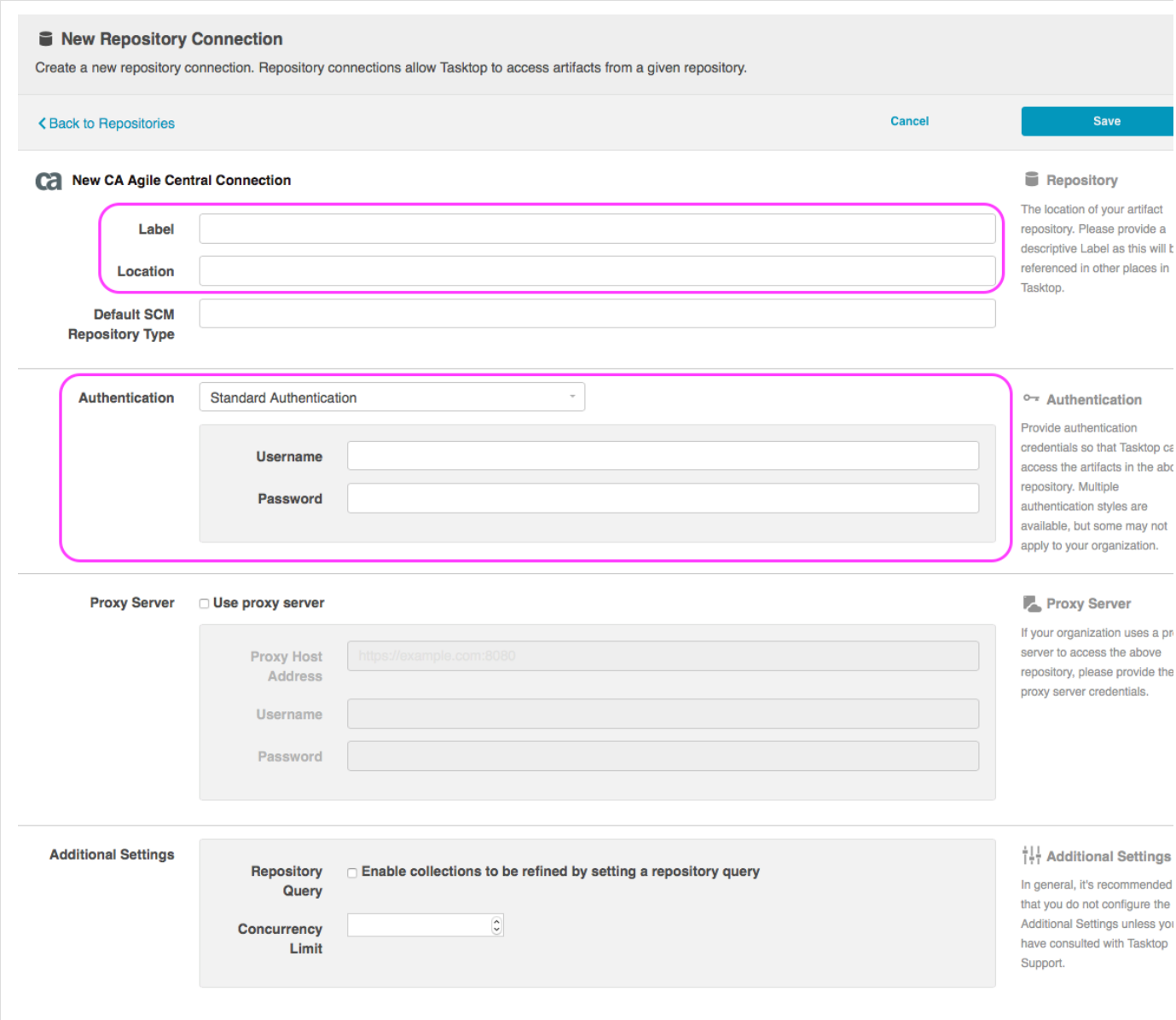
Click the logo of the repository you would like to connect to:



This will lead you to the New Repository Screen.

To connect to a repository, you must populate the following fields:

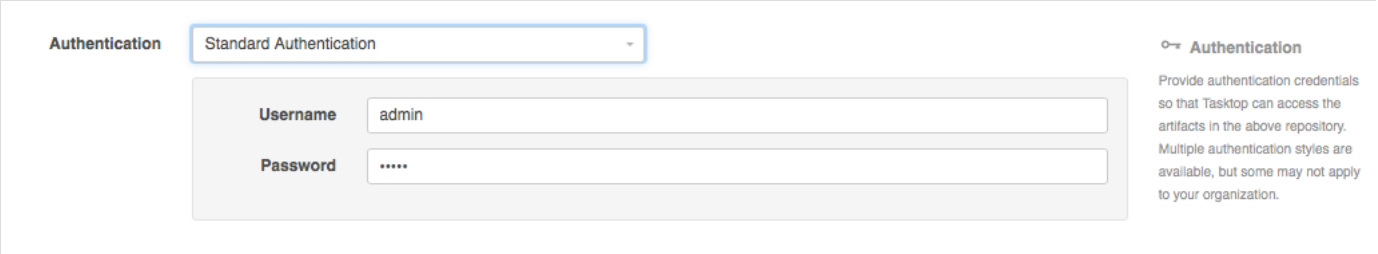
- Label: This is the name you will give to your Repository Connection. This is how it will be referenced throughout the Tasktop Application
- Location: This is the URL used to access the repository.
- Authentication Details (see authentication section below for more details):



Authentication

Standard Authentication

For most scenarios, you will select 'Standard' Authentication.' This is where you will enter the username and password used to access the repository. We recommend creating login credentials specifically for Tasktop to access your repository.



SSO Authentication

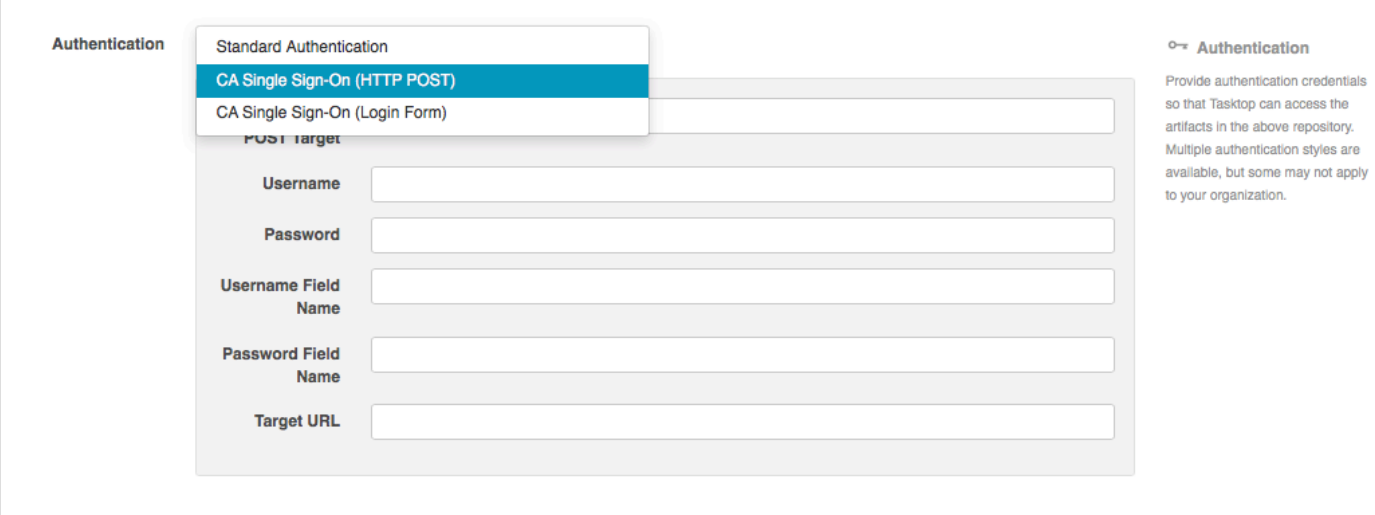
If you connect to a repository utilizing CA SSO authentication, you can select "Single Sign-On (HTTP POST)" or "Single Sign-On (Login Form)" to enter your authentication credentials.

Tasktop currently supports the following SSO implementations:

- CA Siteminder/CA Single Sign-On Form-based
- CA Siteminder/CA Single Sign-On Value-based

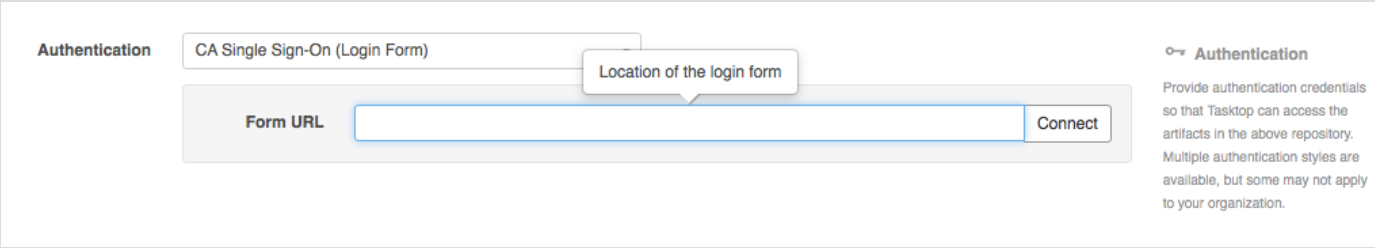
HTTP POST:

The HTTP Post option, pictured below, will generate the authentication form for you to fill in. Only the first 3 fields are required.

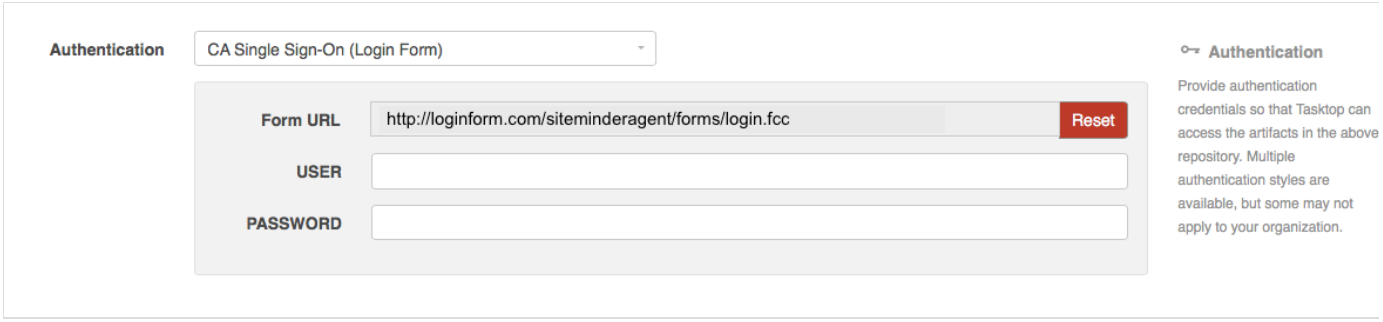


Login Form

The 'Single Sign-On (Login Form)' option, pictured below, will allow you to enter the URL for your SSO log-in form.



Once the URL is entered, Tasktop will auto-generate the fields that must be populated to connect to the repository.



Proxy Server

If Tasktop is installed behind a firewall, you may need to connect to external ALM repositories (e.g. hosted or cloud ALM repositories) through a proxy. To create a connection to such external ALM 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:

Additional Settings

⚠️ In general, it is recommended that you do not configure the Additional Settings unless you have consulted with Tasktop Support.

Repository Query

If you plan to utilize a repository query, select the checkbox here.

⚠️ Repository Queries are advanced functionality, and should only be used when you are truly unable to filter as desired using the built-in Tasktop functionality of Repositories, Collections, and Artifact Filtering. You can learn more about artifact filtering here.

Concurrency Limit

In general, we recommend leaving the Concurrency Limit field blank. However, in cases where there is concern regarding high Tasktop load on a repository, a value can be set to limit how much work Tasktop can do in parallel on that repository.

⚠️ Caution should be used when setting this value. Setting the value too low when there is a large number of projects configured in collections and a low "Change Detection Polling Interval" setting can potentially cause Tasktop to be unable to process artifact changes. Please consult with Tasktop Support before setting a value here.

Additional Steps for Database Connection

In order to configure a reporting integration, you must first connect to the database that will be used by that integration. Creating a new database connection is similar to creating a new repository connection, with a few extra considerations. To create a new database connection:

Step 1: Download the SQL Driver

MS SQL Server

The JDBC driver for MS SQL Server can be downloaded from the Microsoft support site. The SQL Driver Location should reference the directory containing the sqljdbc42.jar file. This file should be the only .jar file in that directory, or you may end up with errors upon configuring your collection.

MySQL

The MySQL Connector/J driver can be downloaded from the MySQL download site. The SQL Driver Location should reference the directory containing the mysql-connector-java-<version>-bin.jar file.

Oracle

The JDBC driver for Oracle can be downloaded from the Oracle support site. Note that it is best if the Oracle JDBC driver that is used matches the version of the Oracle server that you are connecting to. Additionally, the ojdbc6.jar file is the only file that should be in the directory that is used for the SQL Driver Location or you may end up with errors upon configuring your collection.

Step 2: Upload the JDBC driver

The SQL driver files must be put on the file system of the same server where Tasktop is installed. When setting up a connection to your database with the SQL connector, the SQL Driver Location field should reference the location of the SQL driver files on the server.

MS SQL Server

The SQL Driver Location should reference the directory containing the sqljdbc42.jar file. This file should be the only .jar file in that directory, or you may end up with errors upon configuring your collection.

MySQL

The SQL Driver Location should reference the directory containing the

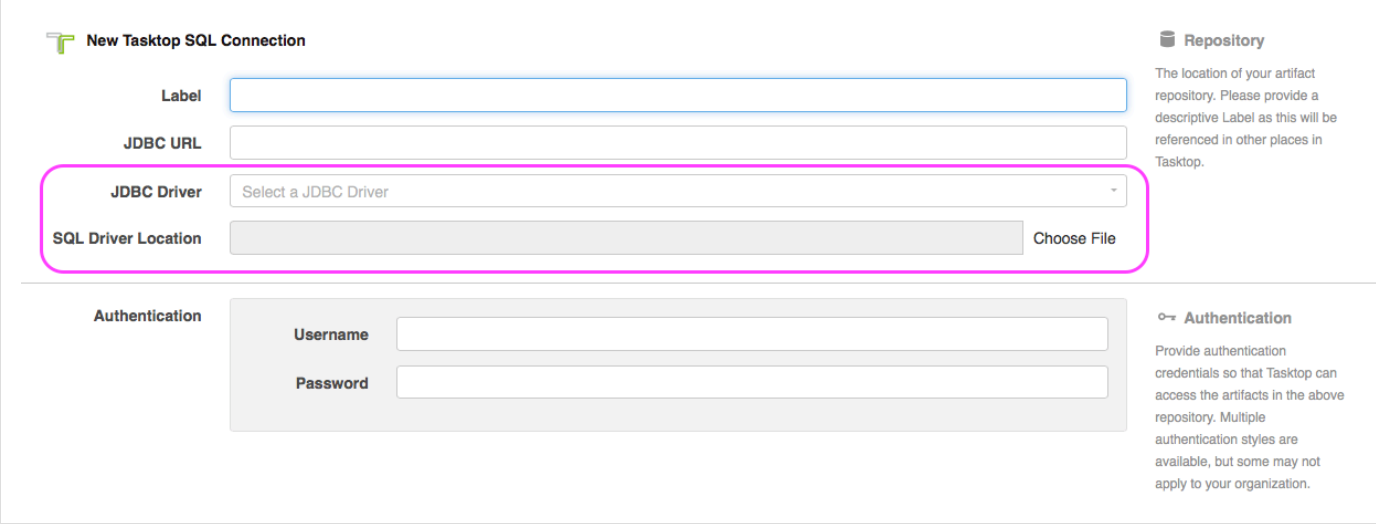
mysql-connector-java-<version>-bin.jar file.

Oracle

The SQL Driver Location should reference the directory containing the ojdbc6.jar file. The ojdbc6.jar file should be the only file in that directory, or you may end up with errors upon configuring your collection. Note that it is best if the Oracle JDBC driver that is used matches the version of the Oracle server that you are connecting to.

Step 3: Connect to your Database

1. In Tasktop, go to the Repositories tab from the top menu, and click New Repository Connection
2. Select Tasktop SQL as the repository type
3. Enter a label for your connection
4. Enter the URL of your database. The protocol should be "jdbc:sqlserver://" for a MS SQL database, "jdbc:mysql://" for a MySQL database or "jdbc:oracle://" for an Oracle database
5. Enter a username and password for your database
6. Select the appropriate JDBC driver (SQL Server, MySQL or Oracle)
7. Enter the SQL driver location, which is the location of the SQL driver files on the Tasktop server. See steps 1 and 2 above for more information on the SQL driver files.
8. Click Done to save the connection

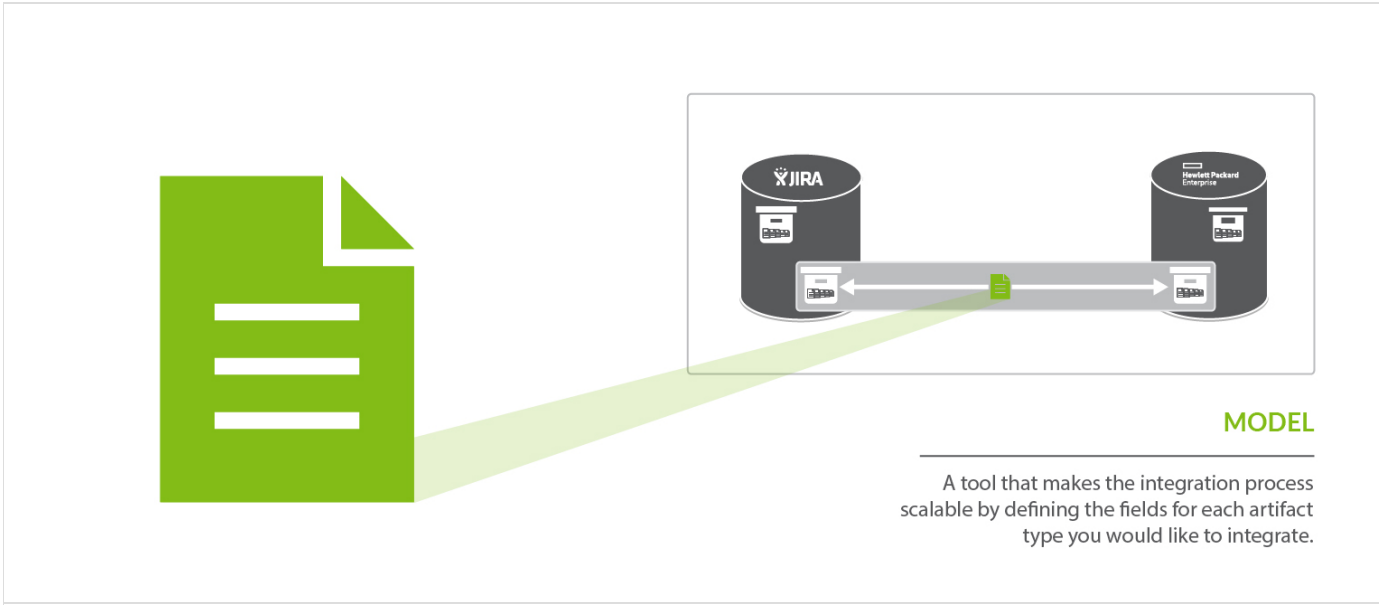


Step 2: Create or Reuse a Model

Tasktop: Apex Release (17.1)

- What is a Model?
- How to Create or Reuse a Model

What is a Model?



A model is a tool that makes the integration process scalable by defining the fields for each artifact type you would like to integrate. By mapping collections to the same model, you will be able to easily add new repositories and new projects within those repositories to your integration landscape. You can learn more models in the Key Concepts.

How to Create or Reuse a Model

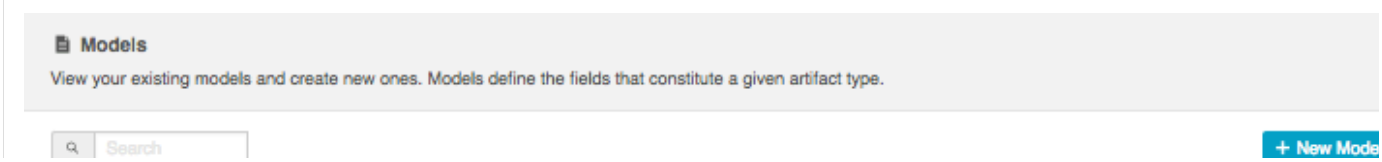
Check out the video below to learn how to create a new model:

Out of the Box Models

Tasktop comes pre-packaged with several out-of-the-box models that are ready for you to use!

However, if you'd like to configure a new model, or edit an existing model, simply follow the steps outlined below.

⚠️ *Note: The ability to create custom models is available only for Enterprise and Ultimate Editions. Custom models are currently available for Pro users for a limited time, in preview mode.*



Artifact
Formatted ID
Project
Type
Created
Modified
Severity
Status
Priority
Release
Assignee

ChangeSet
Formatted Id
Web Url

Defect
Formatted ID
Summary
Description
Severity
Priority
Status
Resolution
Release Detected
Sprint Detected
Release Targeted
Sprint Targeted
Created By
Modified By
Owner
Created
Modified
URL
Blocks
Related

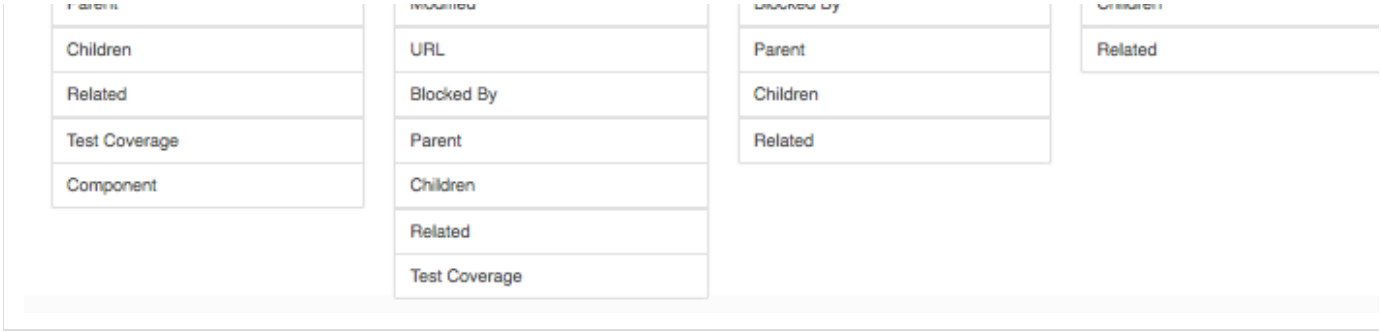
Feature/Epic
Formatted ID
Summary
Description
Severity
Priority
Status
Release
Sprint
Estimate
% Complete
Created By
Modified By
Owner
Created
Modified
URL
Blocked By
Parent
Children
Related
Test Coverage

Requirement
Formatted ID
Summary
Description
Severity
Priority
Status
Testing Status
Release
Sprint
Created By
Modified By
Owner
Created
Modified
URL
Blocked By
Parent

Story
Formatted ID
Summary
Description
Severity
Priority
Status
Testing Status
Release
Sprint
Story Points
Estimate
Remaining
Created By
Modified By
Owner
Created
Modified

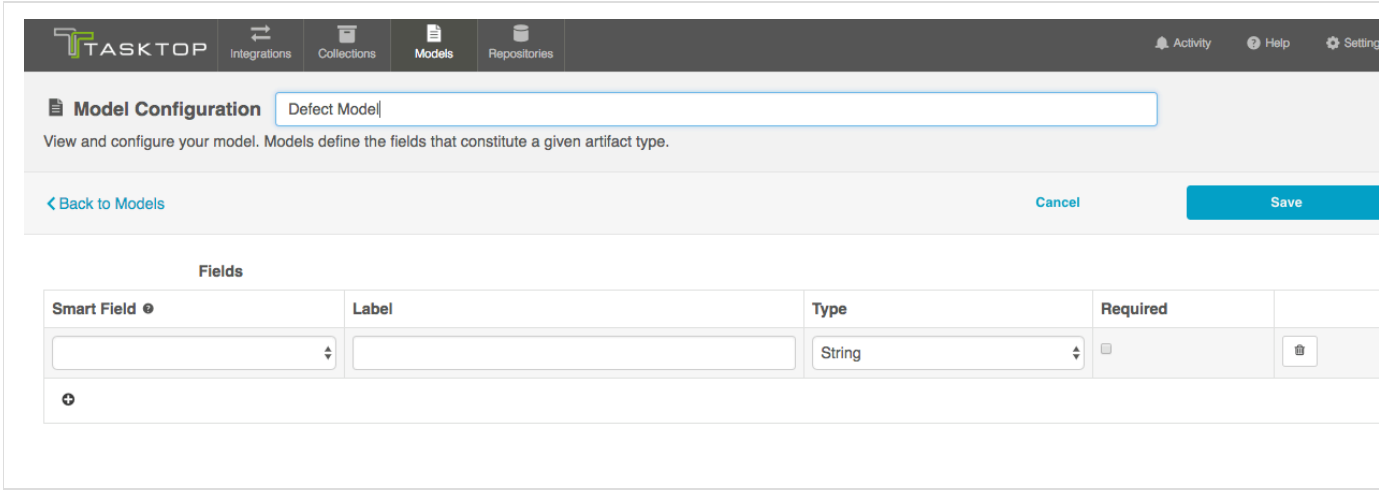
Task
Formatted ID
Summary
Description
Severity
Priority
Status
Release
Sprint
Estimate
Remaining
Created By
Modified By
Owner
Created
Modified
URL
Blocked By

Ticket
Formatted ID
Summary
Description
Severity
Priority
Status
Impact
Urgency
Created By
Modified By
Owner
Created
Modified
URL
Blocked By
Parent



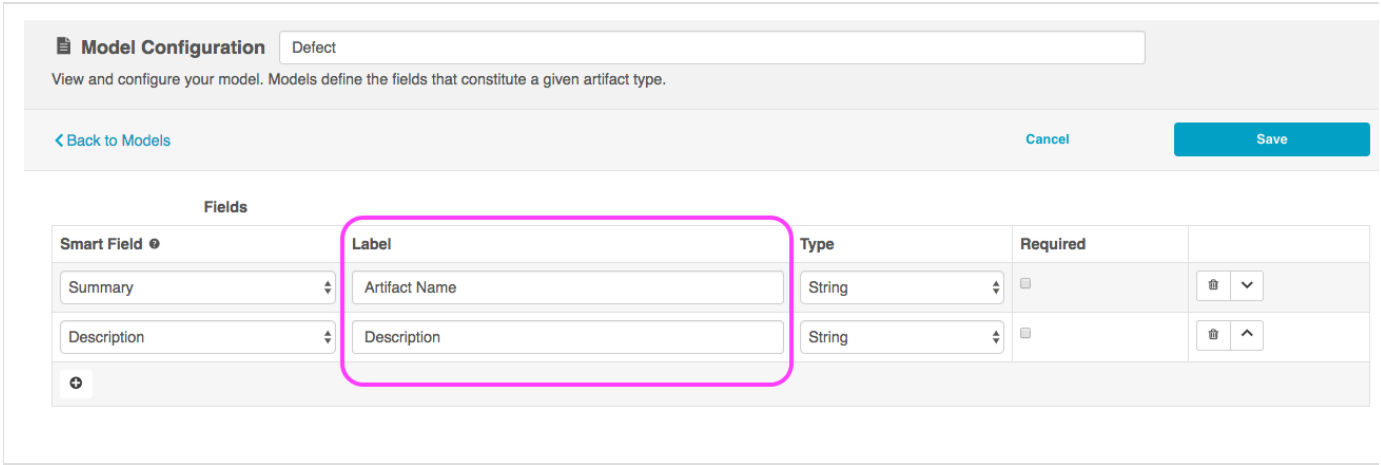
Add Fields to Your Model

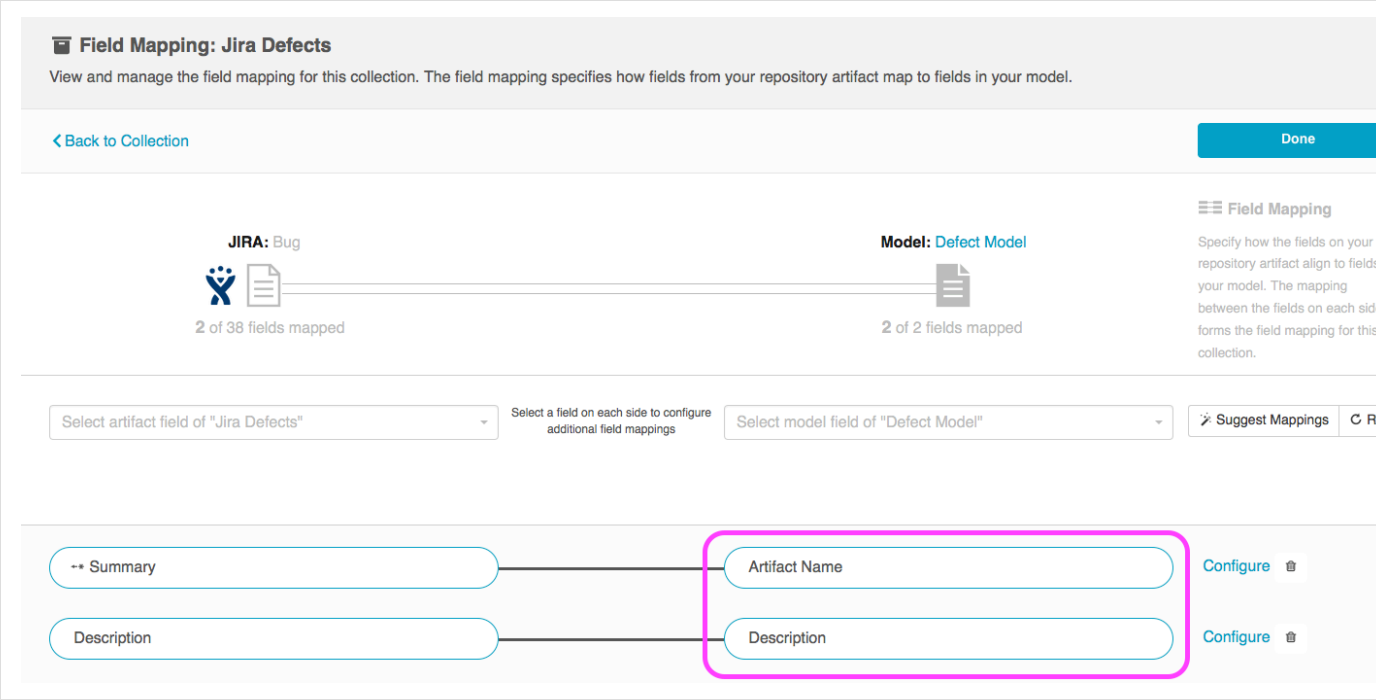
You can start configuring your first model field immediately–just name it and start entering metadata into the first line (to add additional fields to your model, simply click on the plus sign at the bottom left of the model box).



Field Label

This is the name of the field in your model that you will see throughout the application, from the collection to model mapping page in a collection to the field flow page in an integration.





Smart Field Designation

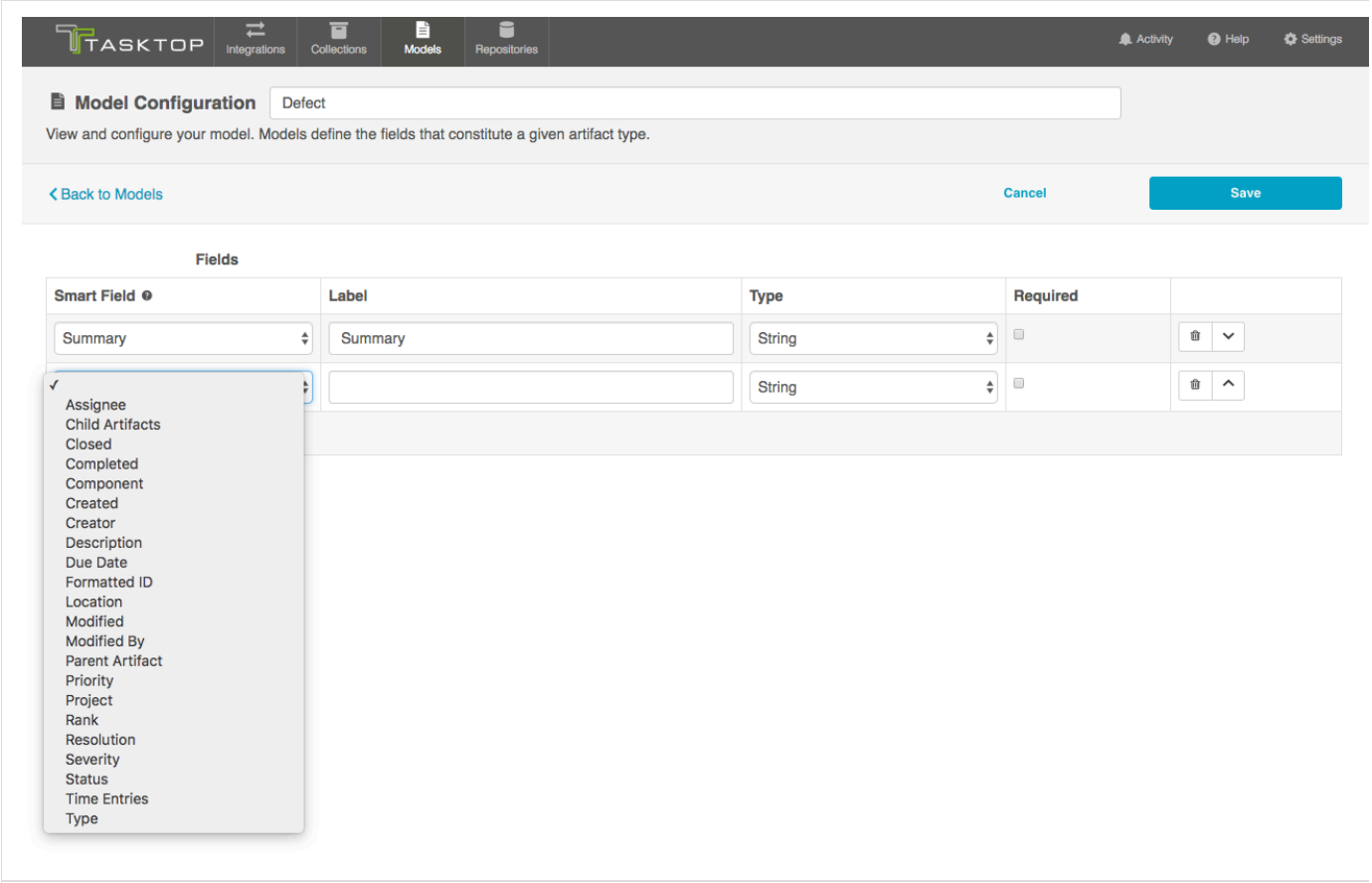
For each field you create, you can optionally identify its corresponding smart field type. Smart fields are a set of fields commonly available in the connectors for all of the repositories Tasktop connects to. By designating a smart field to your model field, Tasktop will be able to more easily match fields from your repositories to your models while you are creating and editing collections.

Selecting a Smart Field will also give Tasktop the power to suggest the proper field type for your model field.

You do not have to select a smart field for all model fields. If you cannot find a smart field that corresponds to a model field, just leave the smart field drop down empty for that field.

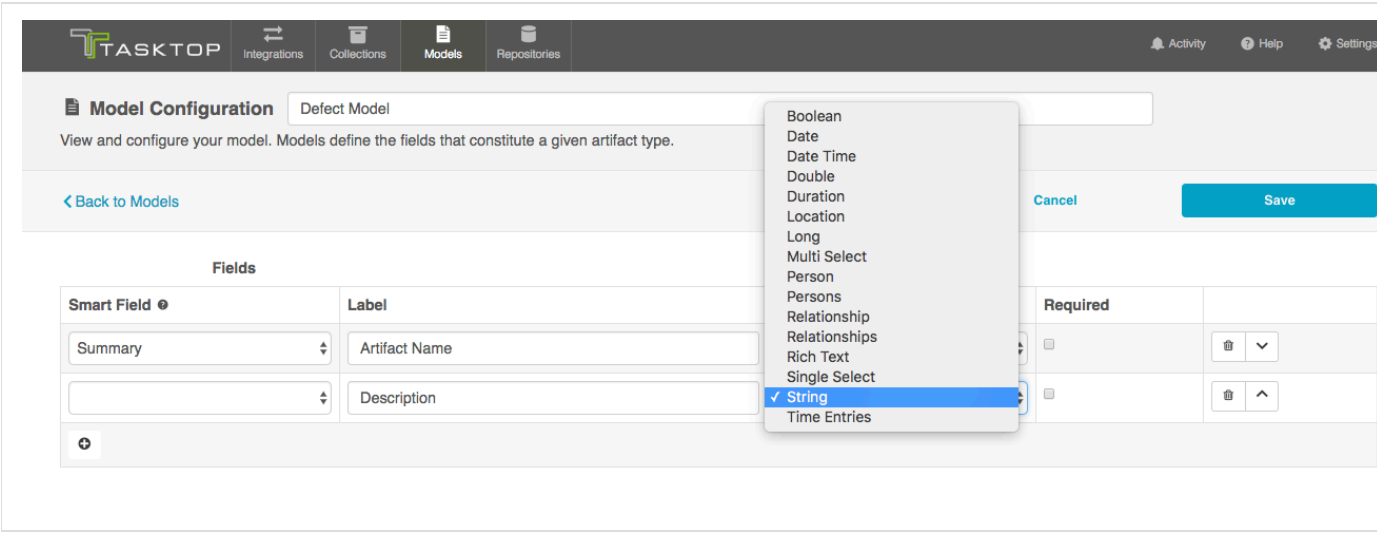
Some examples of smart fields are:

- Formatted ID - the human-readable ID of an artifact
- Location - the field that holds the URL of an artifact
- Modified - a date-time field showing when changes were last made to an artifact



Field Type

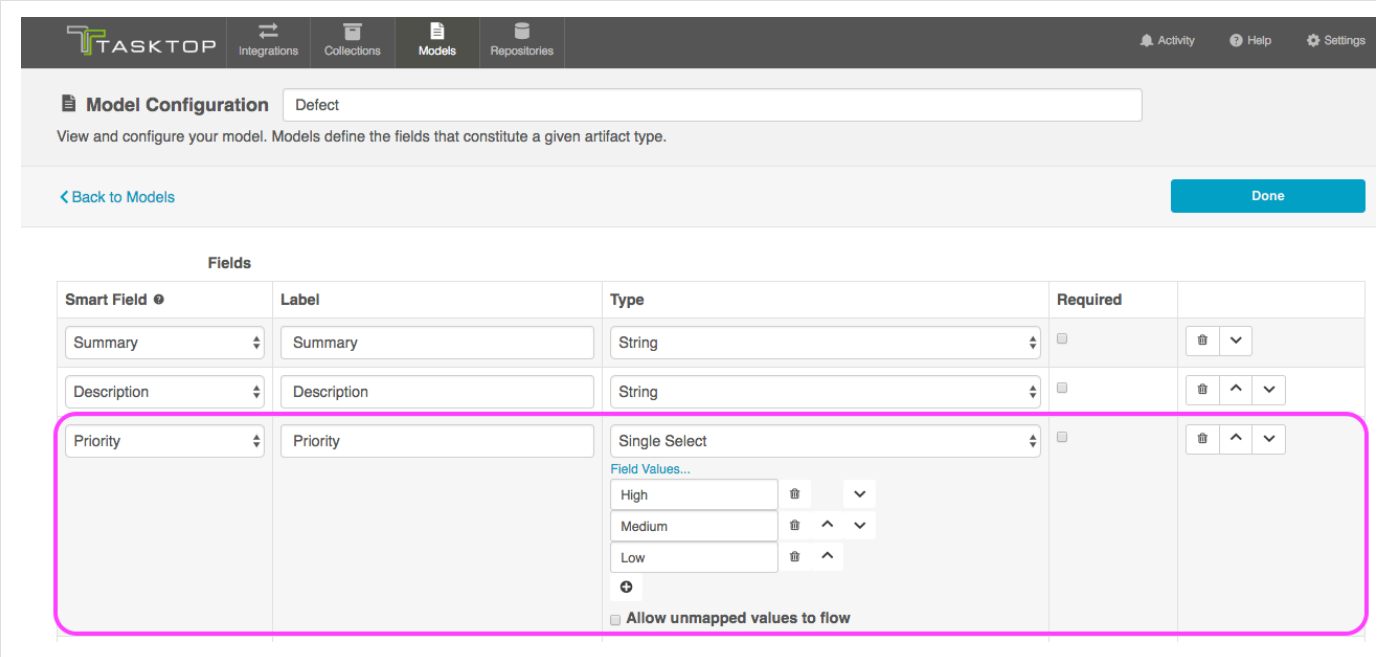
Tasktop supports a number of field types in models. Identify the field type that most closely aligns with the type of information you expect to flow through this model field.



Some field types require extra configuration. Single-selects and multi-selects, for example, give you the option to configure specific values for your field and/or to allow unmapped values to flow.

If you do not allow unmapped values to flow (the default setting), then the server will reject any value that is not specified in the model. In general, this is the recommended approach. If you select this approach, you will need to map all possible values for the repository field to the specific values for the model field.

If you do allow unmapped values to flow, field values not specified in the model will be able to flow while the integration is running. This can make sense in a reporting integration or in single select to string transforms, where there are many options available and you don't desire any normalization of the data flowing through.

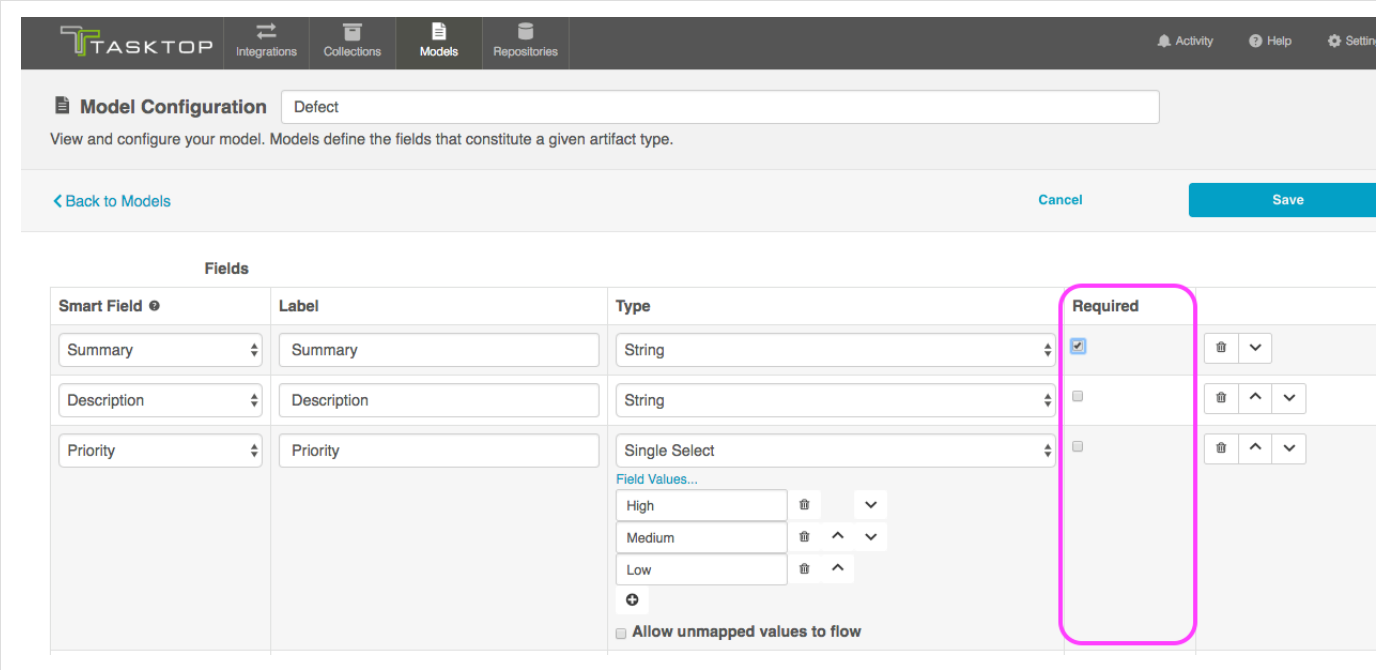


In the image above, you have added 3 specific values for the field "Priority" but have not allowed unmapped values to flow, meaning that any field values sent from the collection will need to be mapped to these 3 model values in order for your artifact to flow successfully.

If you'd like to add additional field values, you can use the '+' button to do so.

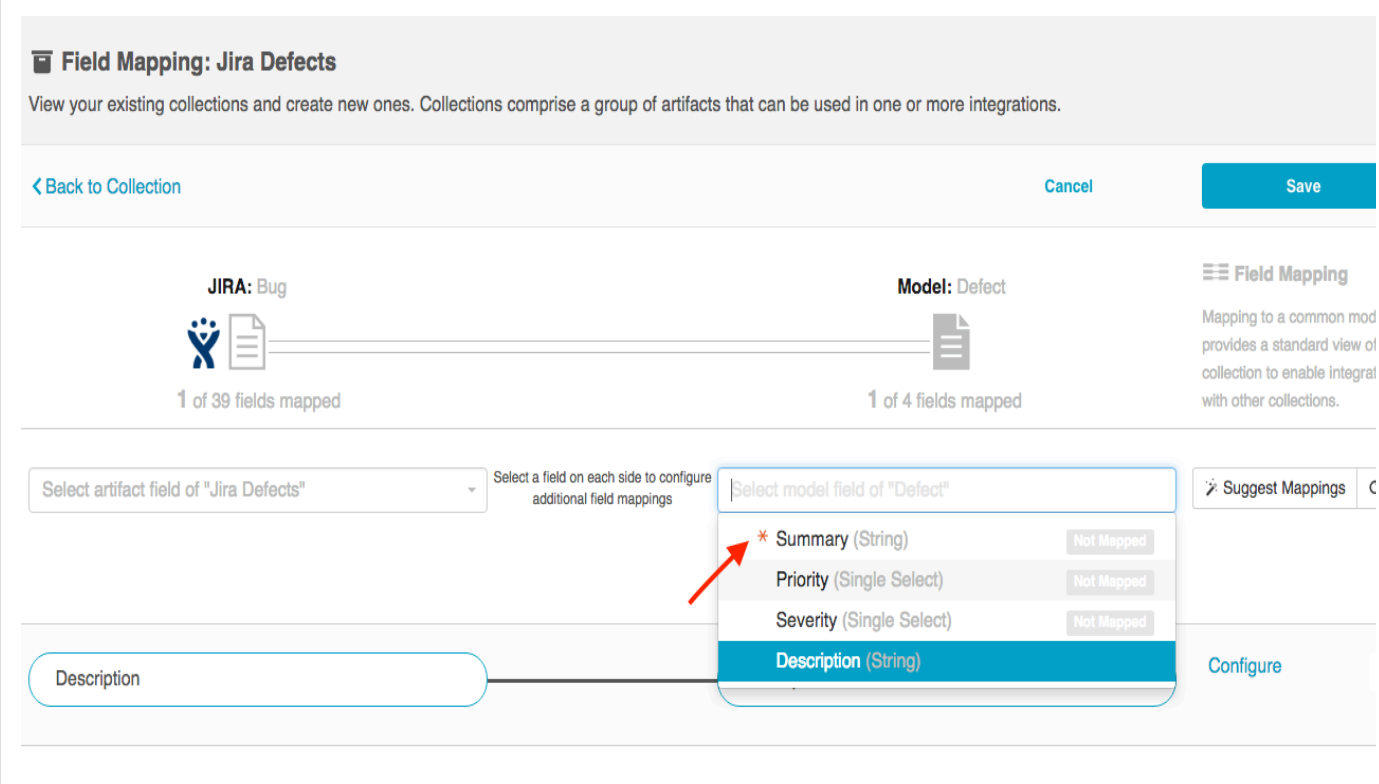
Required Designation

For each field, you can configure whether or not that field requires a value.

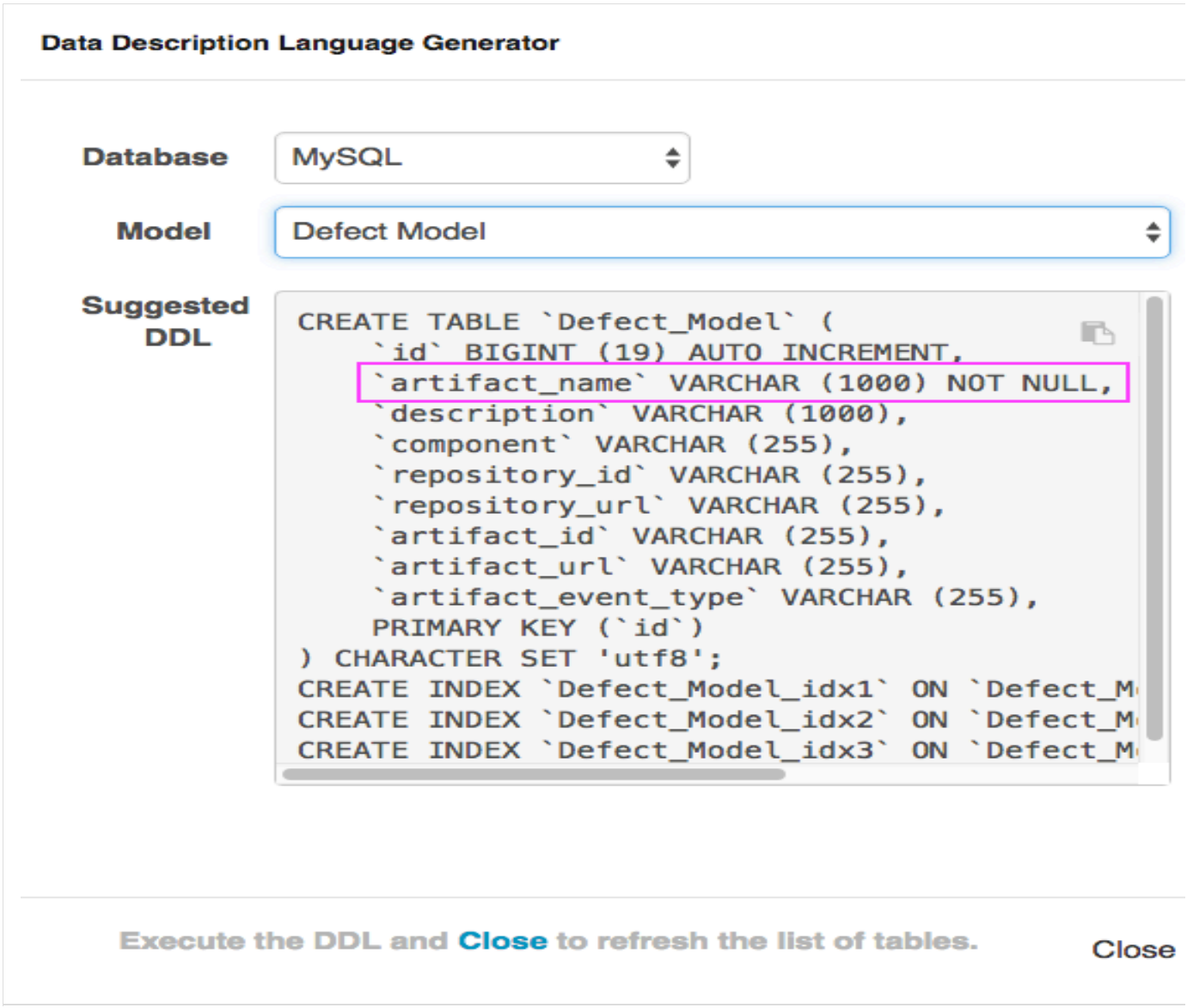


Marking a field as required has implications for all collection types:

- For repository collections, any required model field will be shown with a red asterisk in the collection to model mapping:



- For database collections, the suggested DDL will mark the field as required; this means that if you use that suggested DDL to create your database tables, the field will be required by your database table to create a new record about an artifact:



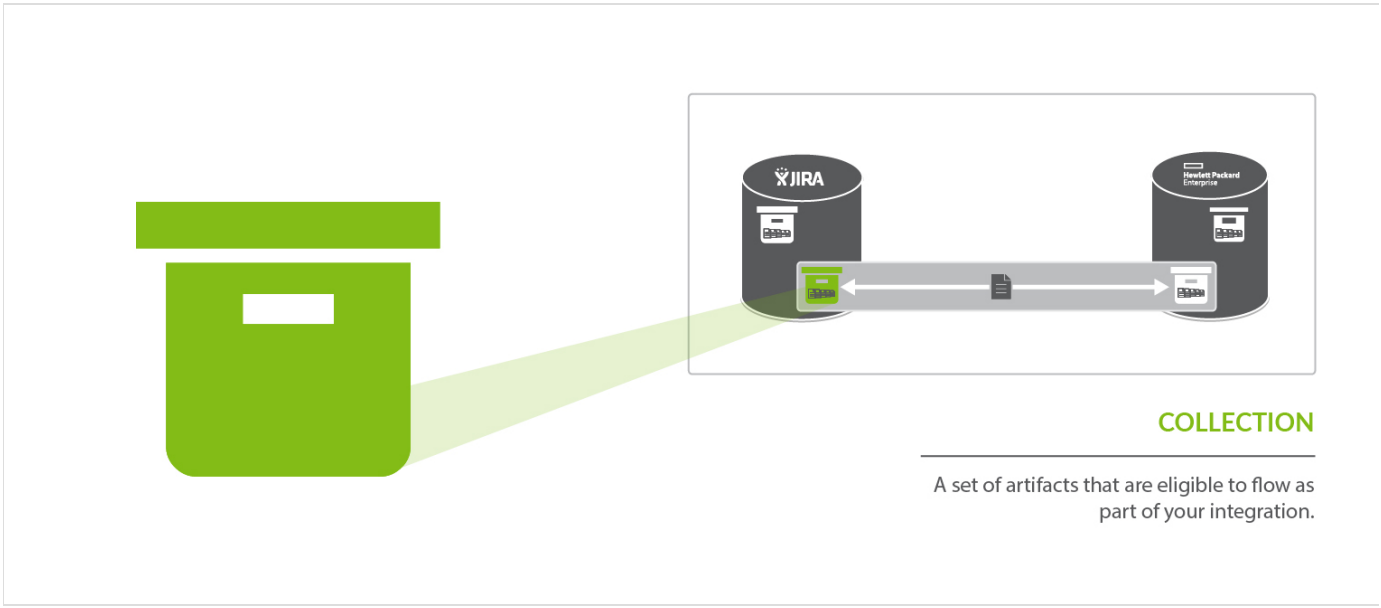
- For gateway collections, you will need to pass in a value in the payload for any required field in order for Tasktop to accept the payload.

Step 3: Create Your Collection(s)

Tasktop: Apex Release (17.1)

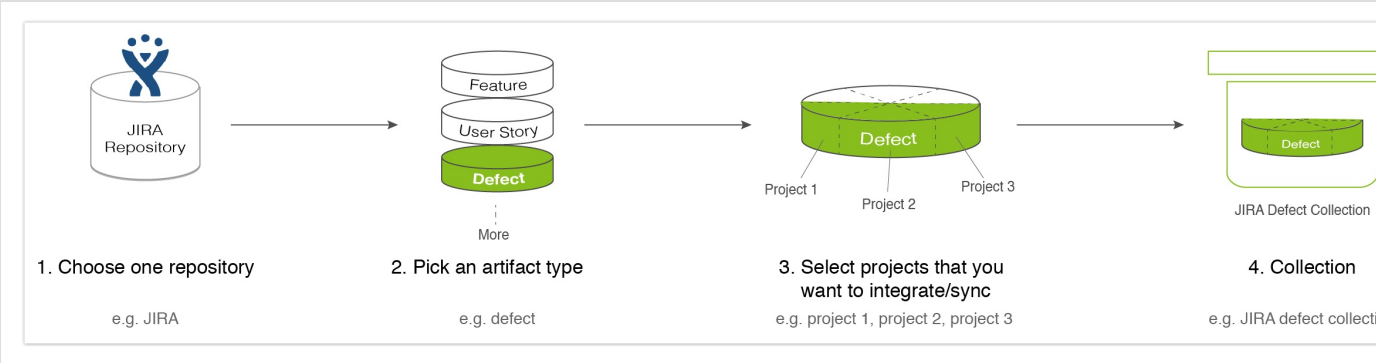
- What is a Collection?
  - Create a Repository Collection
    - Create a Standard Repository Collection
      - Identify Collection Type
      - Name Collection
      - Select Repository Connection
      - Add Projects to Collection
      - Identify the Collection Artifact Type and Model

What is a Collection?



You can think of a *collection* as the set of artifacts that are eligible to flow as part of your integration. The process of creating a collection consists of a few steps which whittle down your repository into a smaller subset of artifacts. To create your collection, you will specify:

1. The repository the artifacts live in
    a. Each collection can only come from *one* repository
2. The artifact type (i.e. defect, requirement, test case, etc)
    a. Each collection can only contain *one* artifact type
3. The projects within the repository those artifacts live in
    a. Each collection can contain one or multiple projects
4. The model you would like your collection to be mapped to (not pictured)
    a. Each collection can be mapped to one and only one model



Step 3: Create Your Collection(s)

You can learn more about collections in the Key Concepts.

Create a Repository Collection

There are two types of Repository Collections: Standard Repository Collections, which connect to repositories like *JIRA, HPE ALM,* and *ServiceNow*, and Database Repository Collections, which connect to databases, such as *MySQL.*

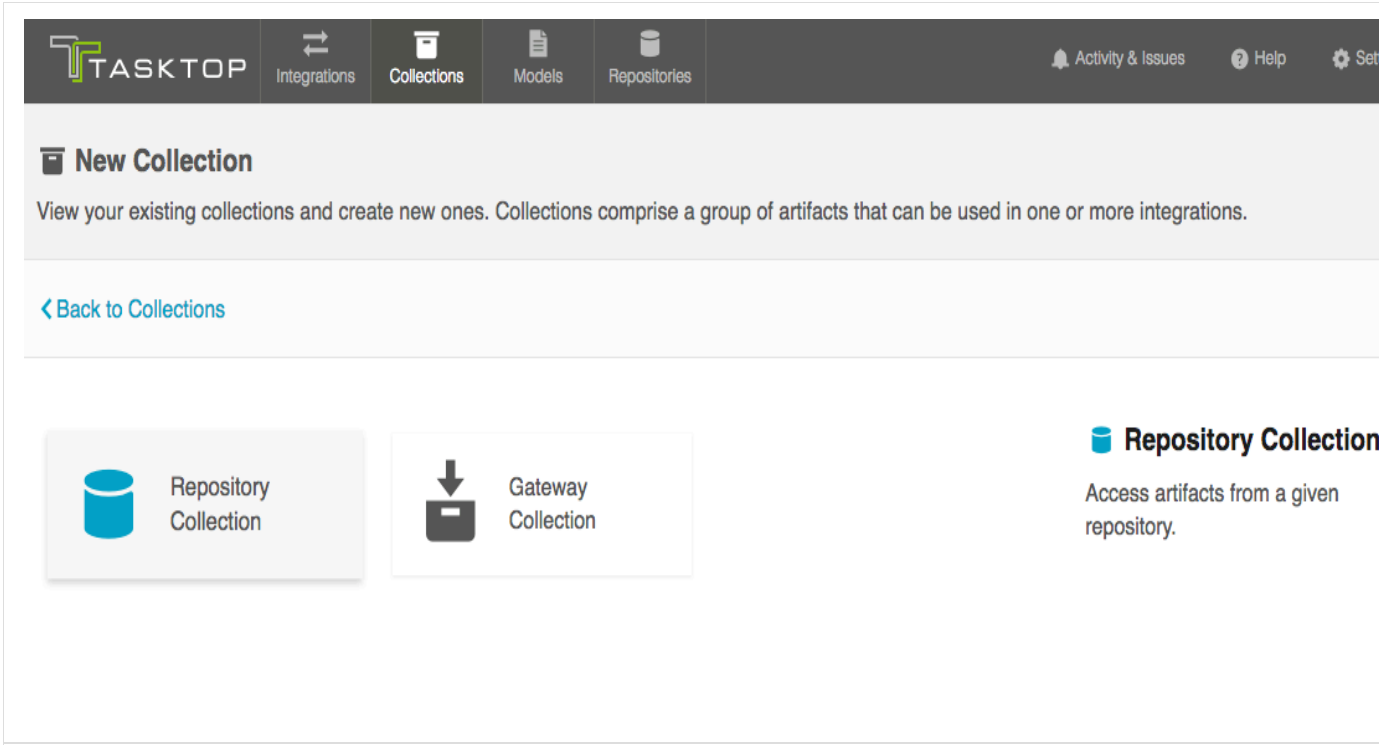
Create a Standard Repository Collection

Check out the video below to learn how to create a new repository collection:

Identify Collection Type

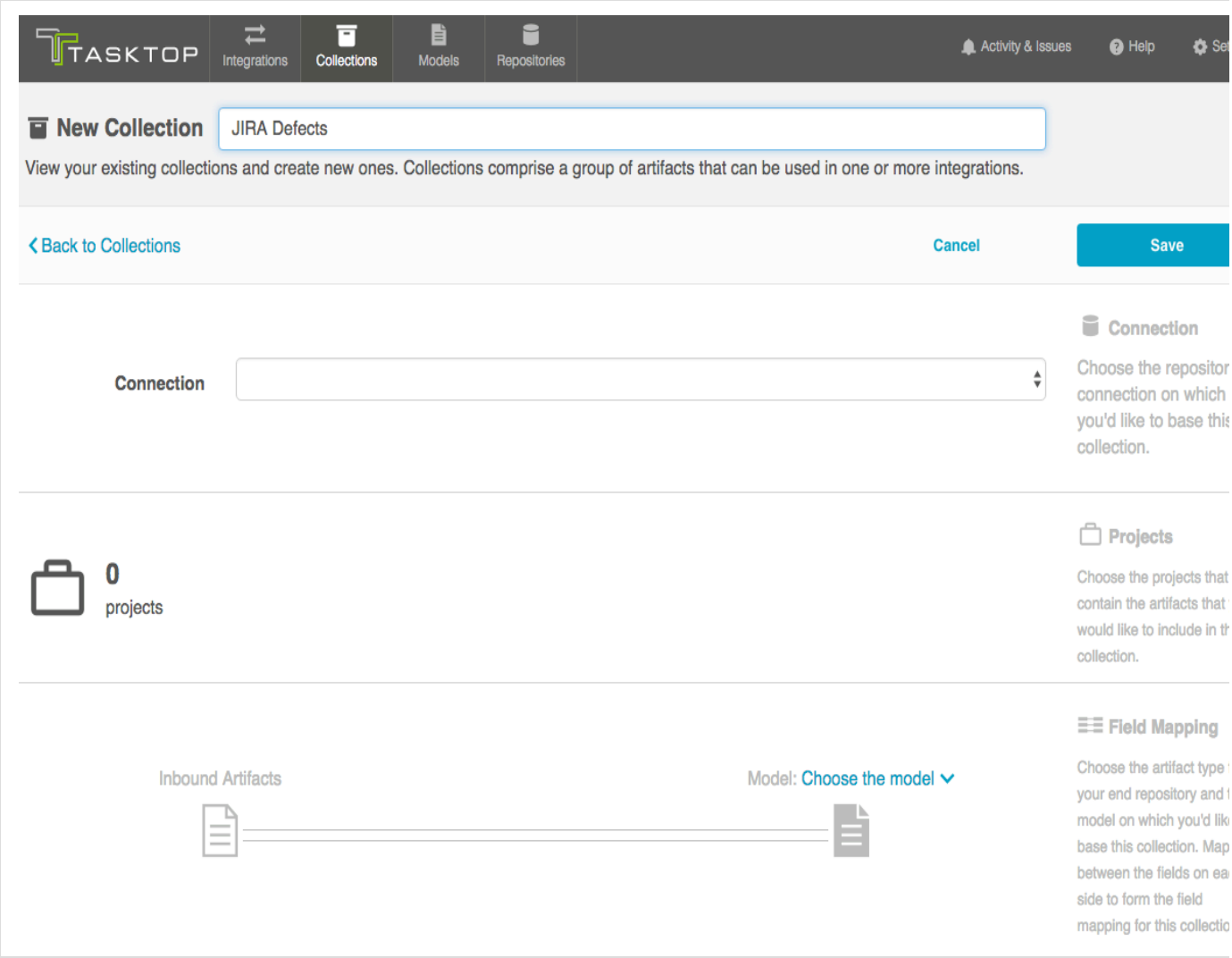
To create a repository collection, do the following steps after you go to "New Collection":

Select "Repository Collection" as the collection type.



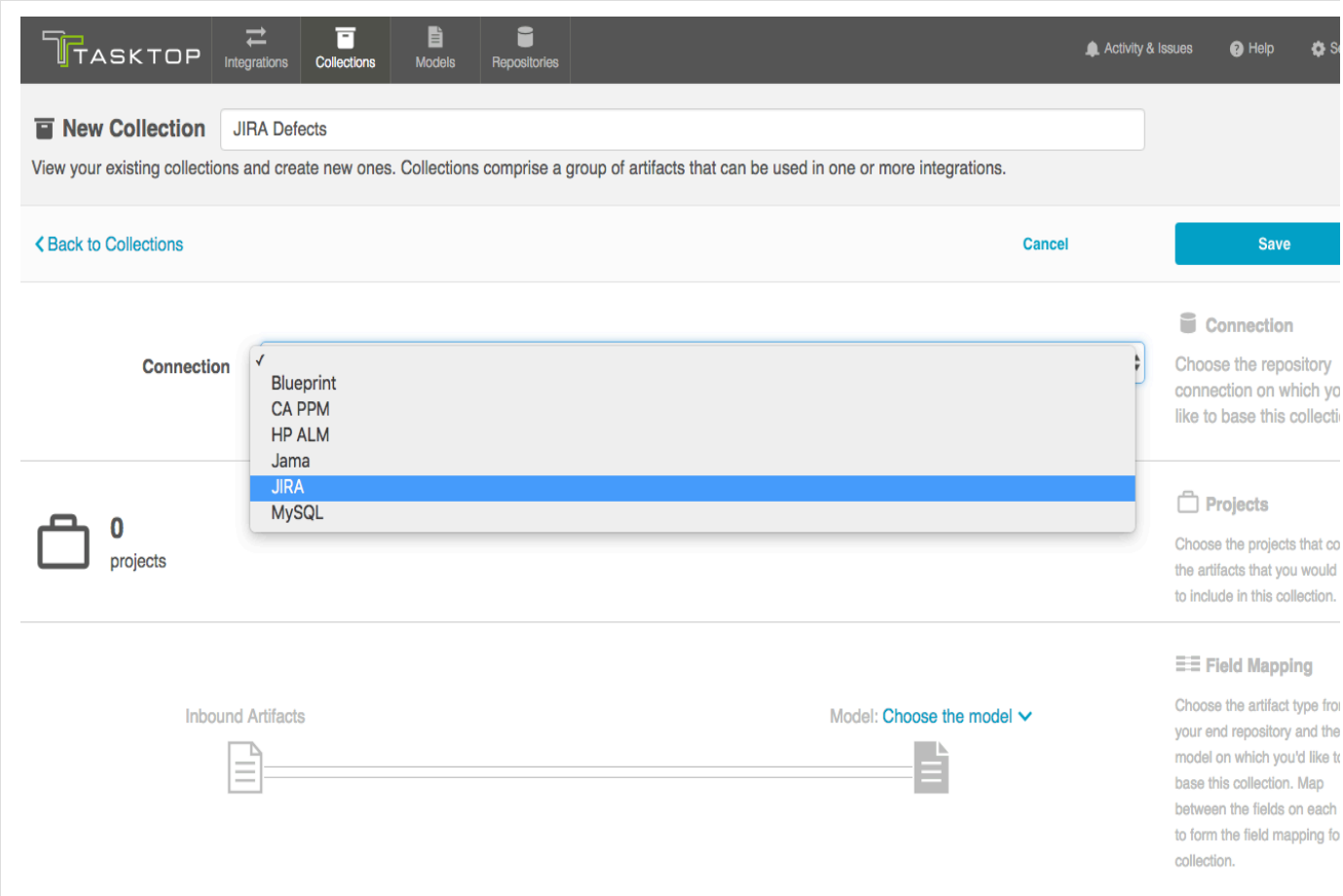
Name Collection

Enter a name for your collection



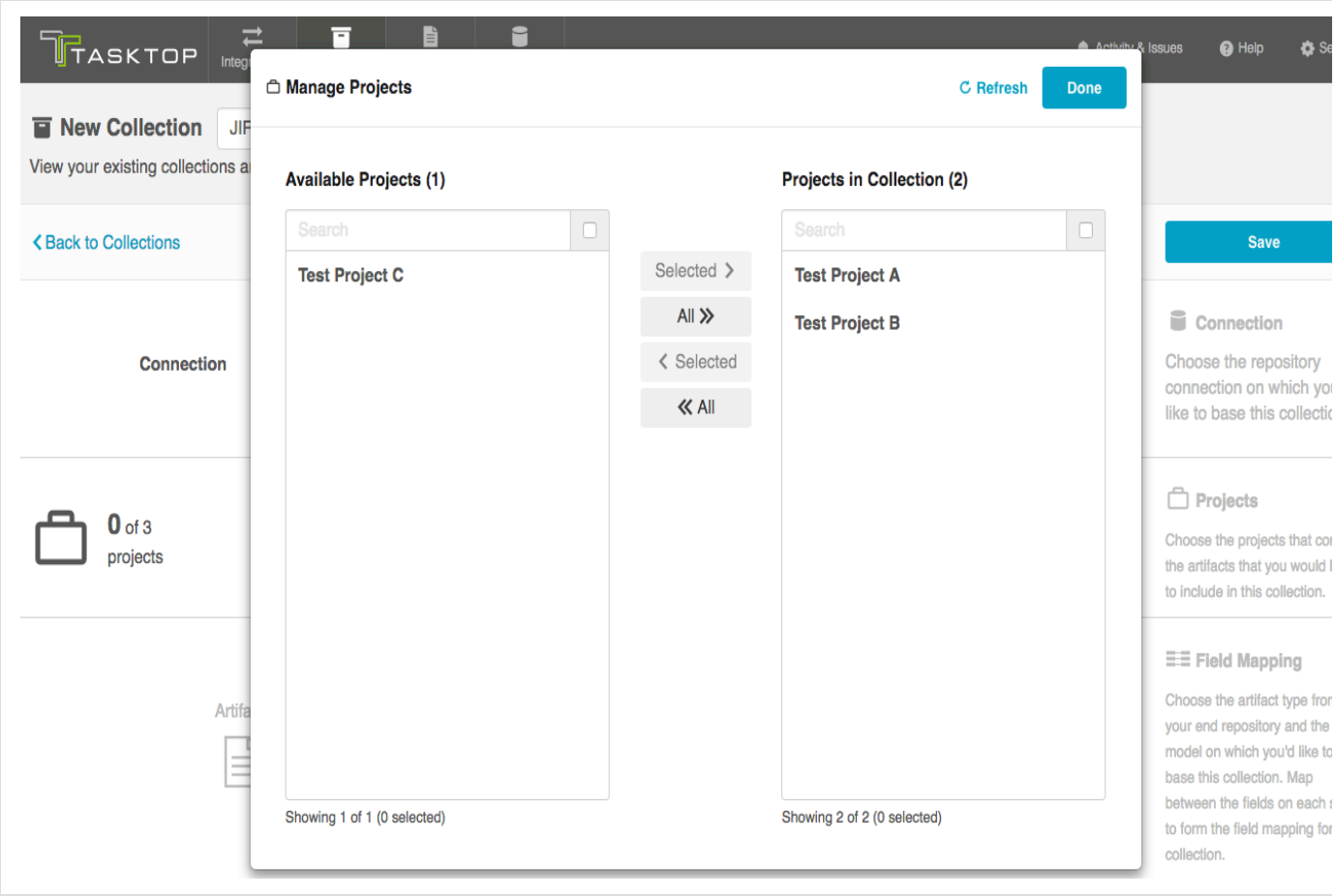
Select Repository Connection

Select the Repository Connection on which you'd like to base this collection. The collection will include artifacts from the repository collection you have selected.



Add Projects to Collection

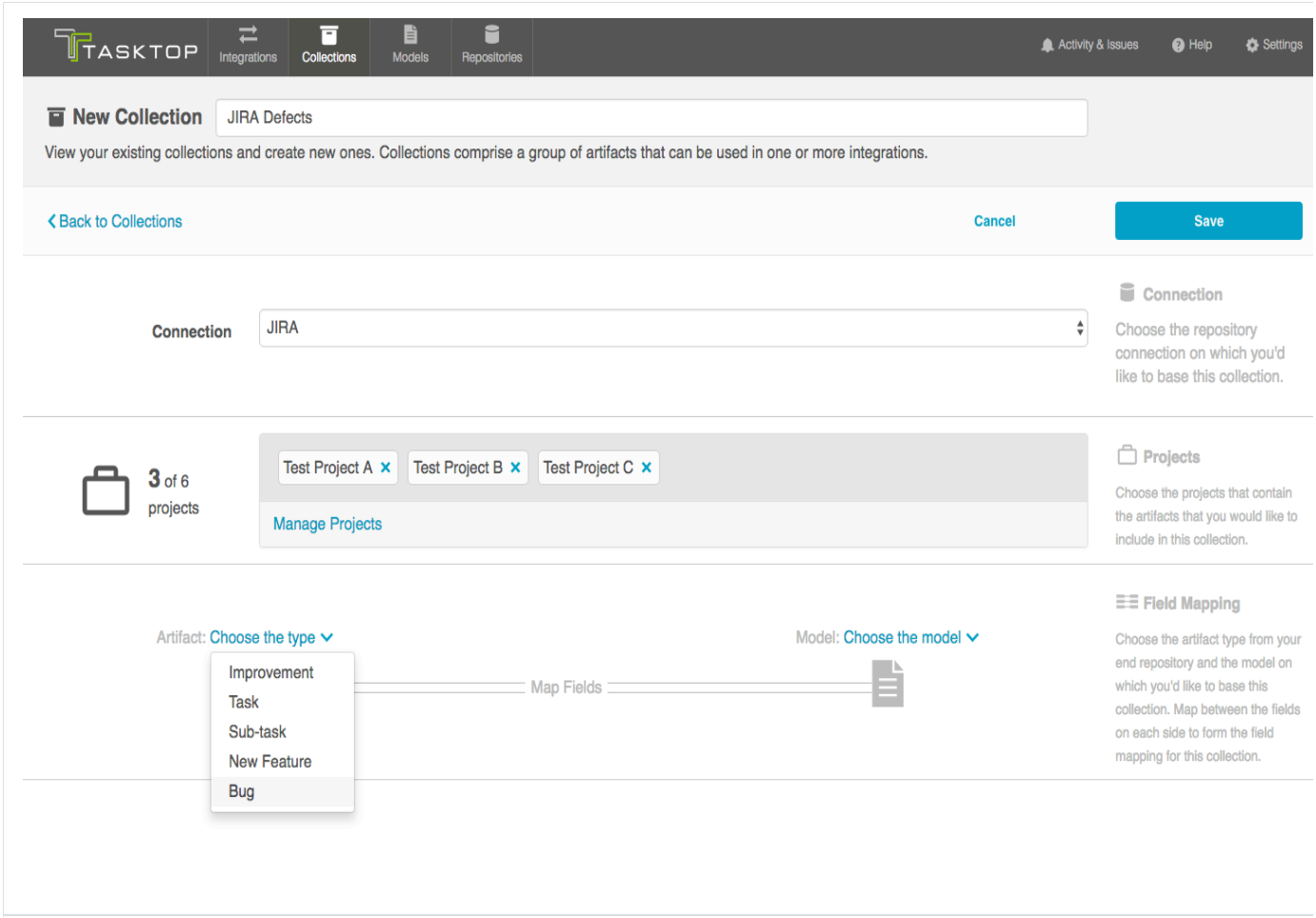
Add projects to you collection by selecting 'Manage Projects'. These are the projects from which Tasktop will be able to create, retrieve, and update artifacts.

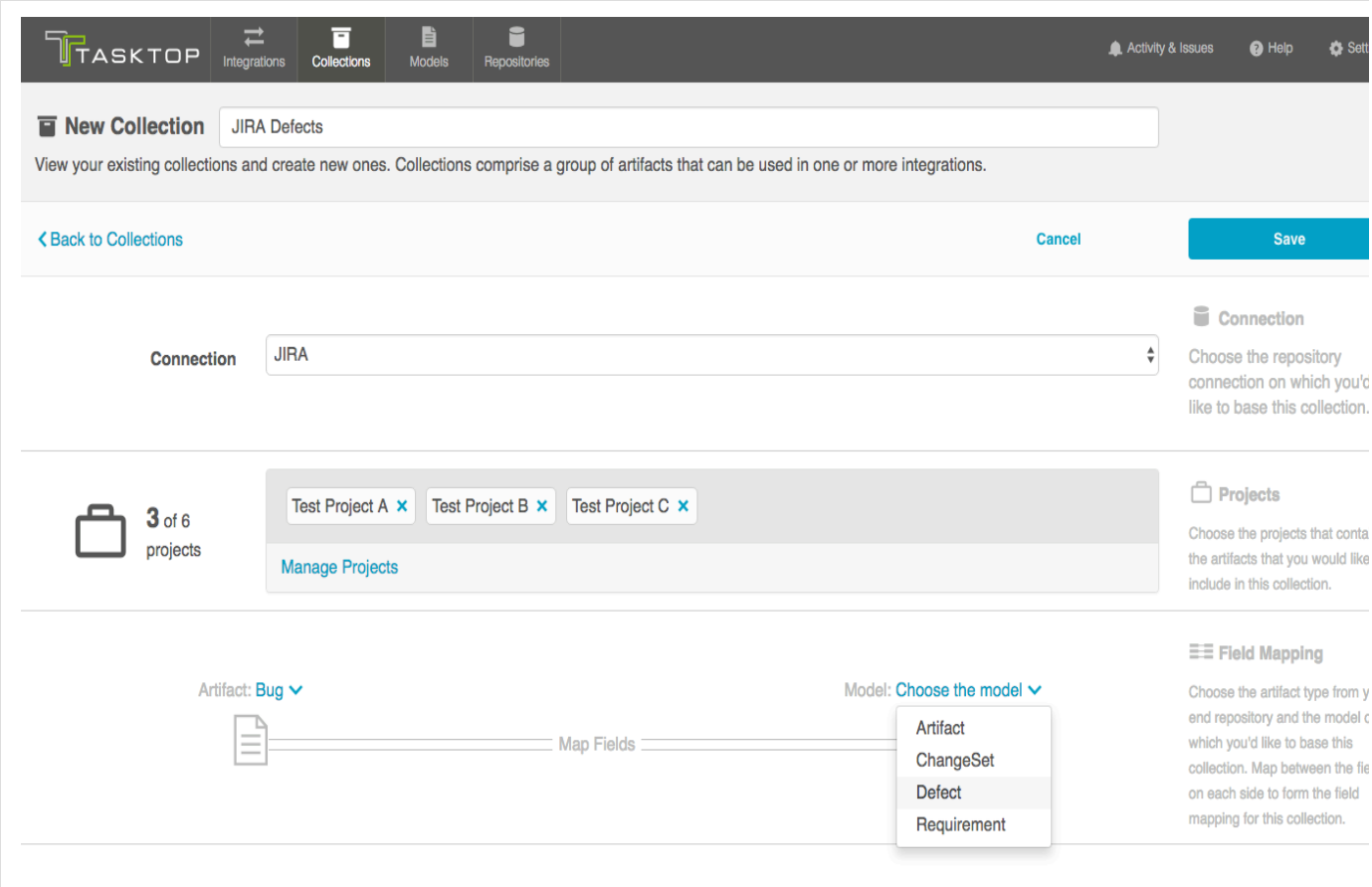


Identify the Collection Artifact Type and Model

Select the artifact type from the end repository that you would like to include in this collection. Remember, a single collection can only contain artifacts of a single type.

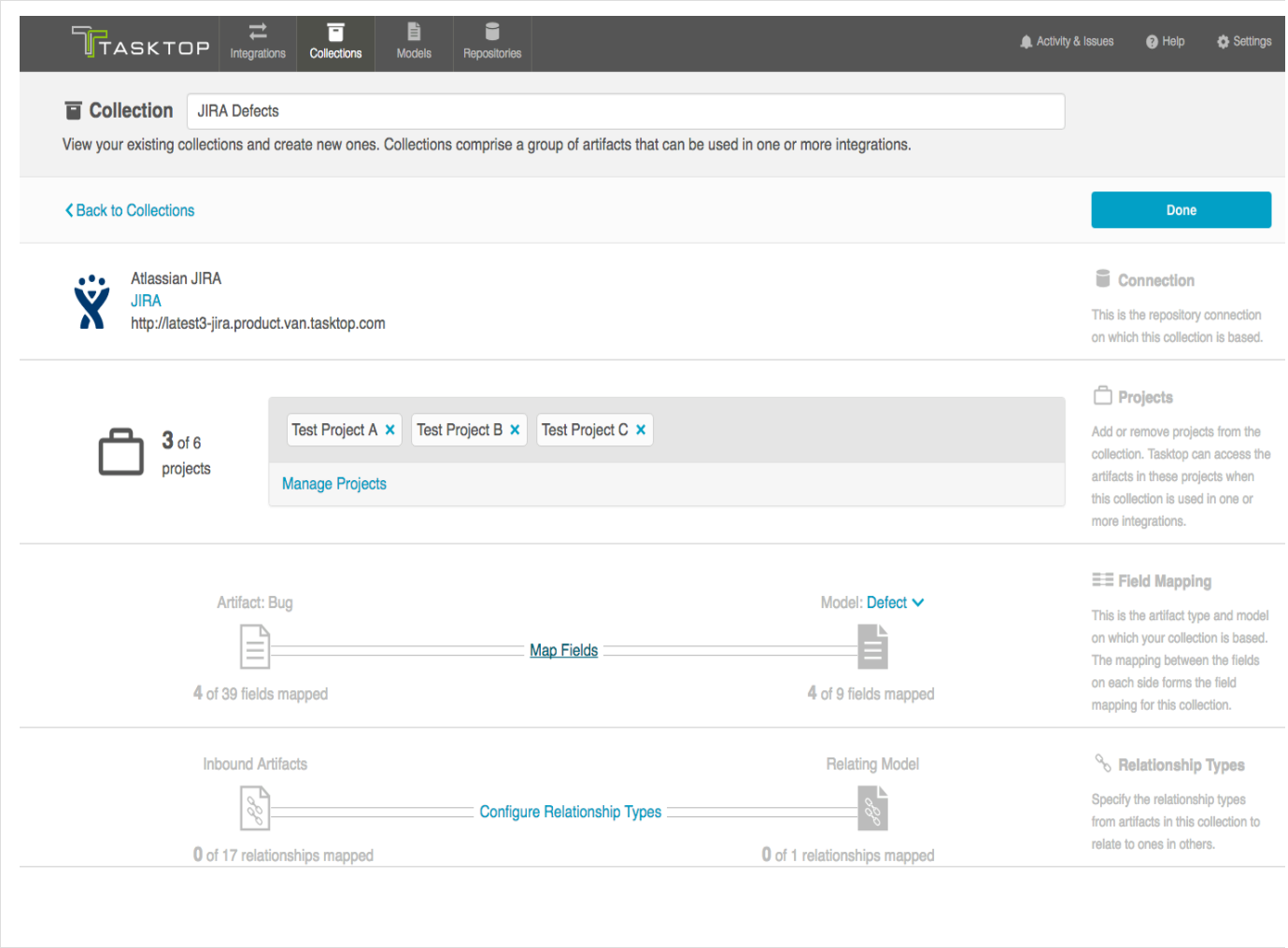
Select the Model on which you'd like to base this collection.


TASKTOP
Integrations
Collections
Models
Repositories
Activity & Issues
Help
Settings
New Collection
JIRA Defects
View your existing collections and create new ones. Collections comprise a group of artifacts that can be used in one or more integrations.
Back to Collections
Cancel
Save
Connection
JIRA
Connection
Choose the repository connection on which you'd like to base this collection.
3 of 6 projects
Test Project A
Test Project B
Test Project C
Manage Projects
Projects
Choose the projects that contain the artifacts that you would like to include in this collection.
Artifact: Choose the type
Improvement
Task
Sub-task
New Feature
Bug
Map Fields
Model: Choose the model
Field Mapping
Choose the artifact type from your end repository and the model on which you'd like to base this collection. Map between the fields on each side to form the field mapping for this collection.



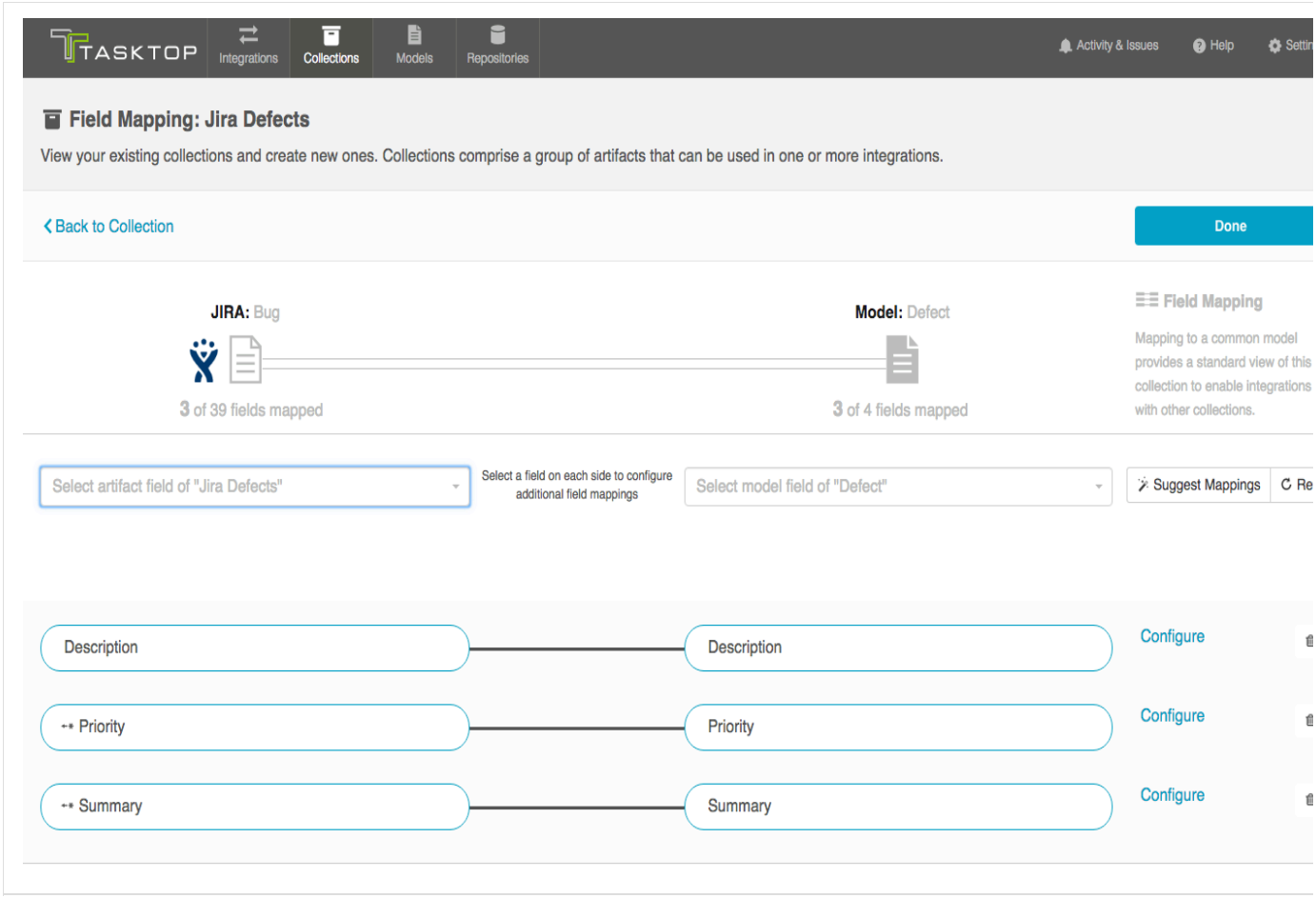
Verify/Complete the Collection to Model Mapping

Now that you have identified the collection artifact type and model, you can complete the collection to model field mapping by going into the "Map Fields" link. The link will become active once you save the collection.

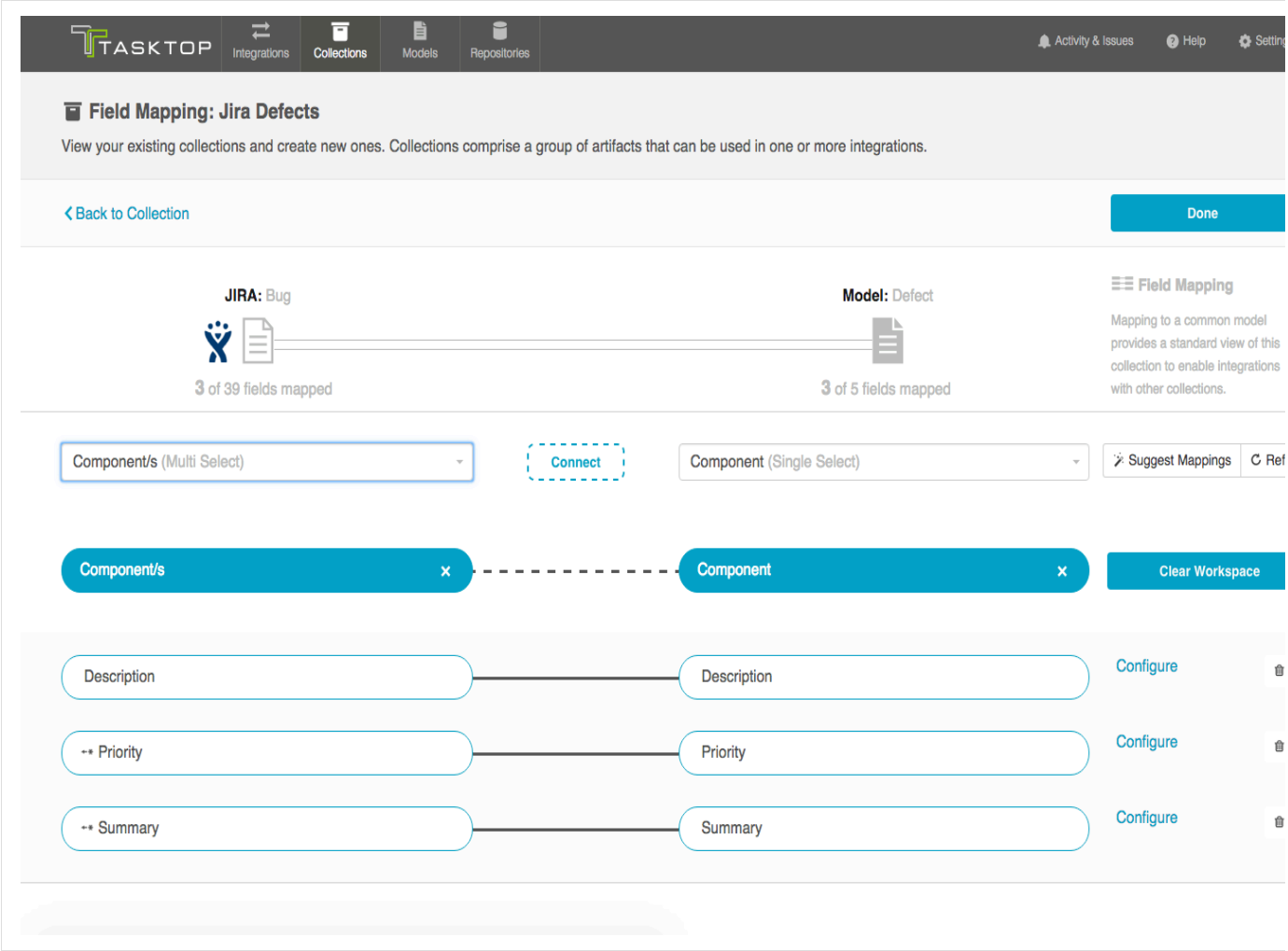


Doing so will take you to a drill in page where you can specify how the fields in your model will map to the fields available on your artifacts in the repository. Tasktop will auto-map fields when possible, based on the names of fields and the standard field designations that have been set in a given model.

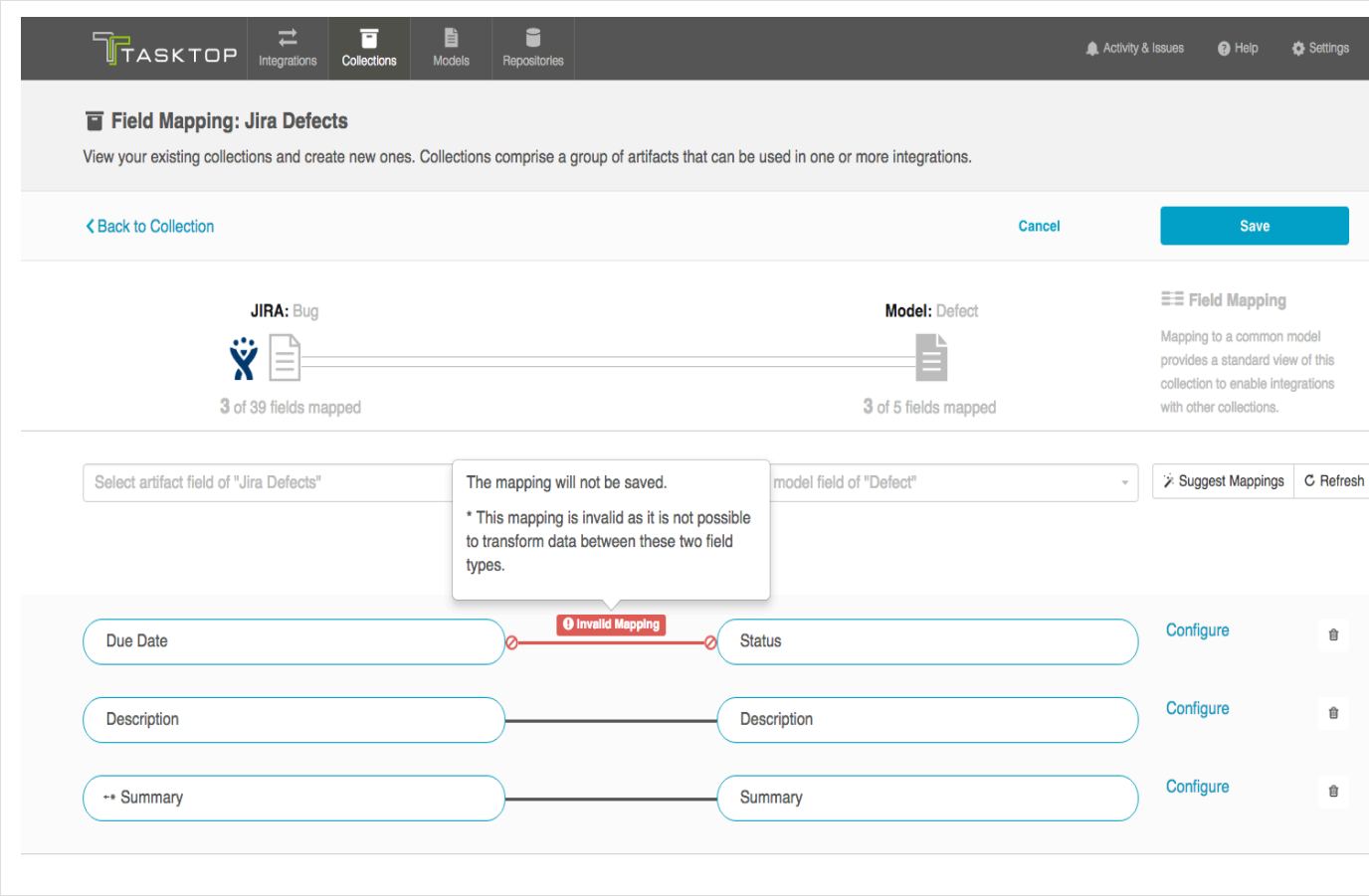
💡 Tip: If you need to refresh the fields available for the collection, use the 'refresh' button to the right of 'Suggest Mappings,' rather than your browser's 'refresh' button.



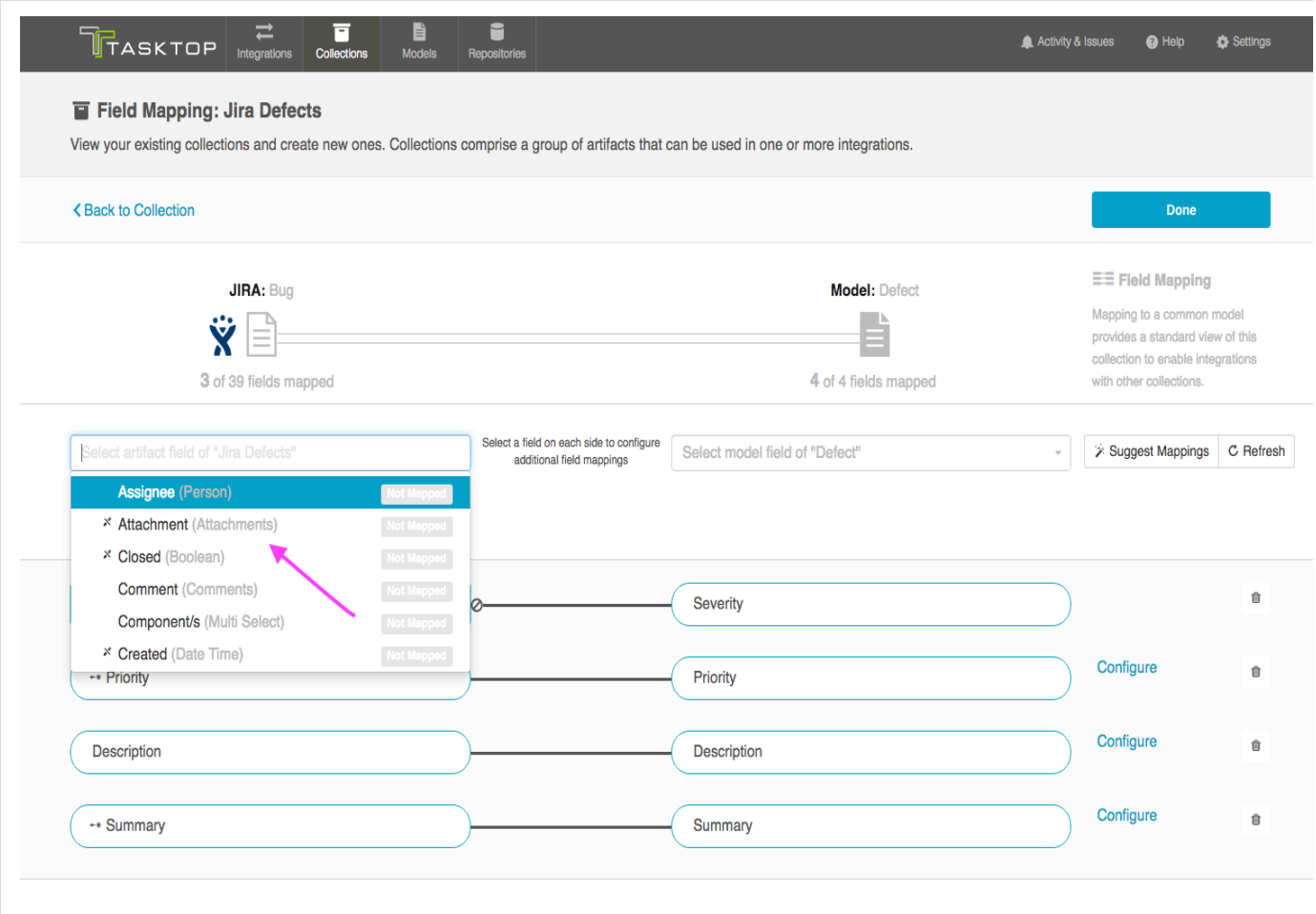
You can map additional fields by using the two drop down boxes:



⚠️ Note: If you attempt to map fields that do not have a valid transform between one another (example: if you map 'due date,' a date field, to 'status,' a single-select field), you will get an 'invalid mapping' warning, and the mapping will not be saved.



To help troubleshoot, you can view the field type when selecting each value from the drop down menu. This will enable you to ensure that the transforms between the two field types will make sense.



Field Mapping Icons

On the Collection-to-Model Mapping screen, you will see a number of icons which will help you understand any special properties or requirements for each field. If you hover your mouse over an icon, you will see a pop-up explaining what the icon means. You can also review their meanings in the legend below:

Icon	Meaning
C	A constant value will be sent. Note that: • If the icon is on the side of the collection, this means that a constant value will be sent to your model. This means that any time this collection is integrated with another collection, the *other* collection will receive this constant value for the field in question. • If the icon is on the side of the model, this means a constant value will be sent to your collection. This means that any time this collection is integrated with another collection, that *this* collection will receive this constant value for the field in question.
	Collection field is read-only and cannot receive data.
←✱ ✱→	To create artifacts in your collection, this field must be mapped to your model.

✳	This is a required field in your model; it must be mapped to your collection.
⊘	This field will not be updated as part of your integration because the mapping would be invalid. You do not have the option of changing this.

💡 NOTE: Some fields will have both the 'read only' and the 'required for artifact creation' icons (

). This means that while the field is required for artifact creation, it is also read-only due to the fact that the repository will generate the value for that field upon creation. In these cases, the value **does not need to be mapped** on the Collection-to-Model Field Mapping screen. For example, the field 'Date Created' may be required for artifact creation. However, this field is generated by the repository at the time of creation, and therefore is read-only. It cannot be modified based on fields sent from the model, and therefore does not need to be mapped on the Collection-to-Model Field Mapping page.

Constant Value Mapping

In some scenarios, either the collection artifact or the model might require that a value be provided for a given field. This value is usually provided by mapping it to the equivalent field. However, sometimes your collection artifact has a field that needs a value that doesn't align with any fields in your model, and sometimes your model might have a required field that doesn't have an equivalent field from the collection artifact. In these cases, you can set a constant value. By doing so, you'll specify the value that you would like to provide for that field.

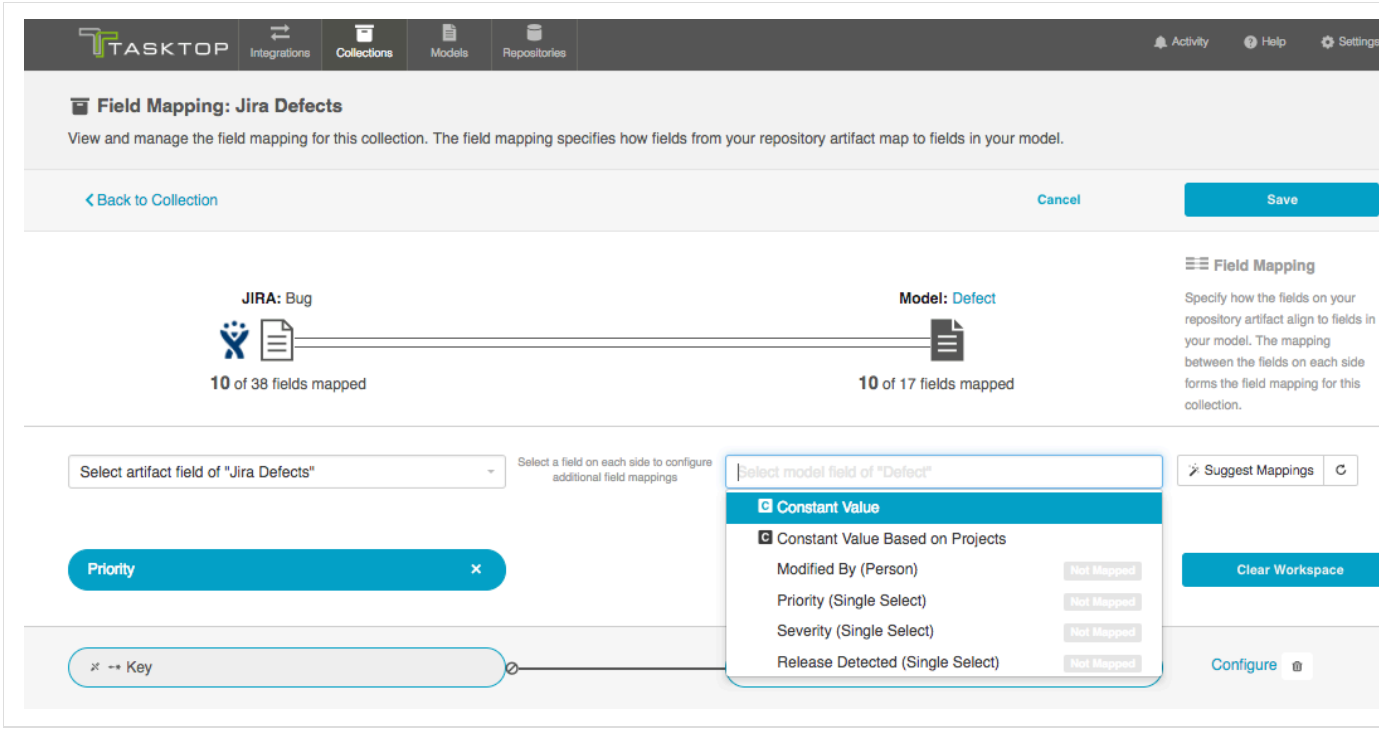
Constant values can be set for the following fields types:

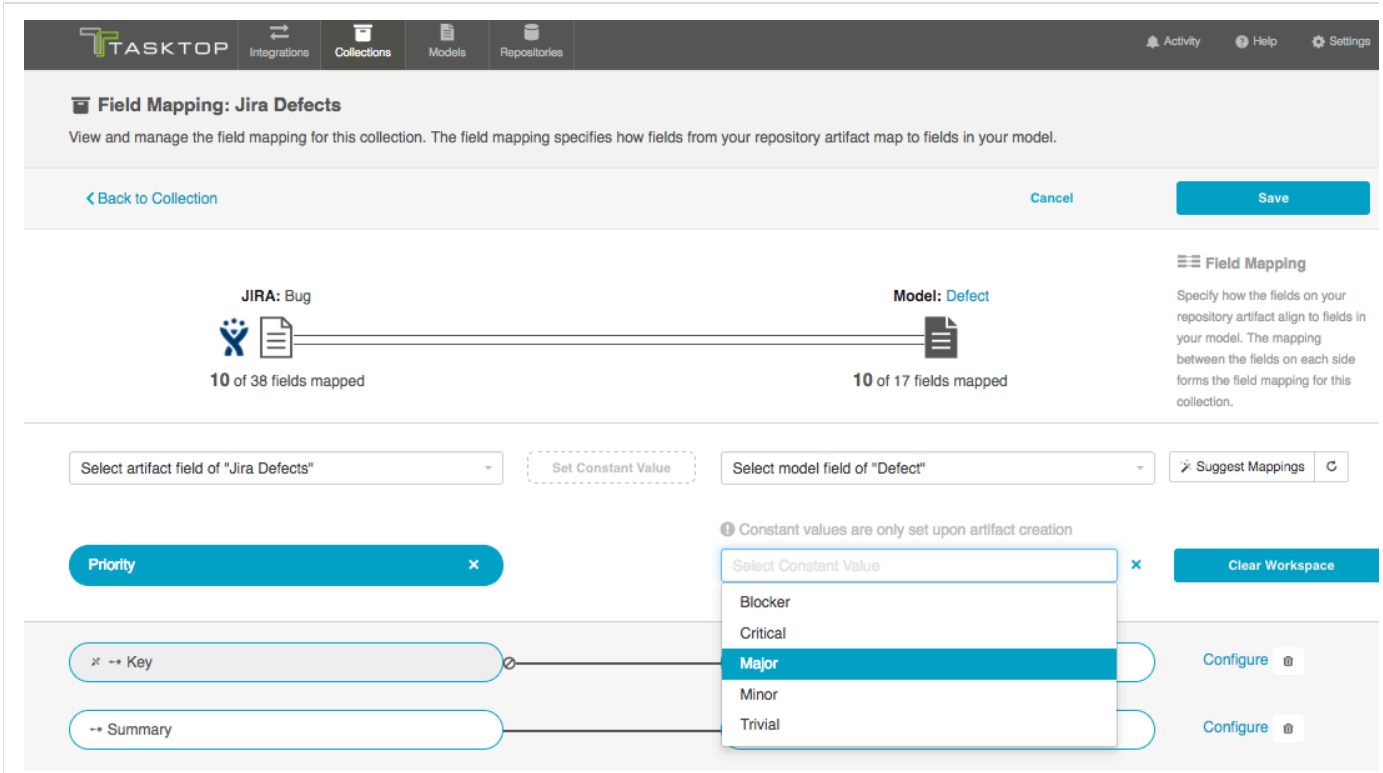
- Single Select
- String
- Date/DateTime
- Person
- Rich Text
- Double
- Long
- Boolean
- Location

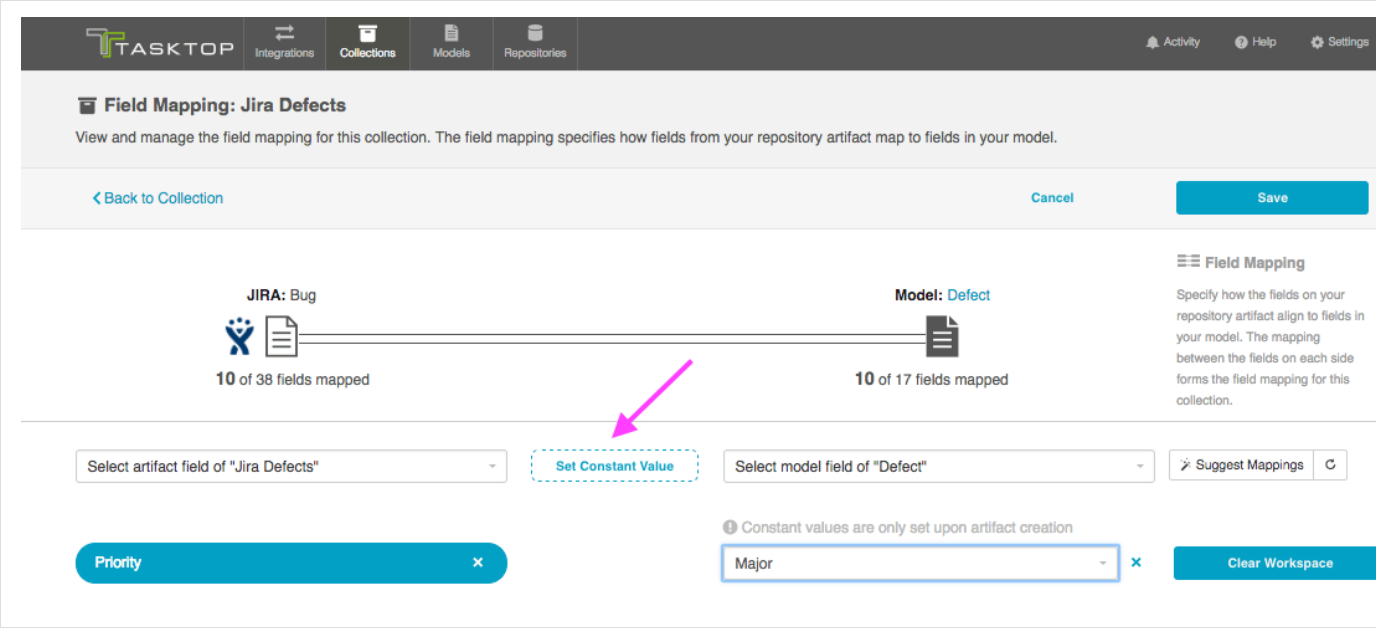
Scenario 1: If your repository requires a field for artifact creation, but that field is *not* a part of your model:

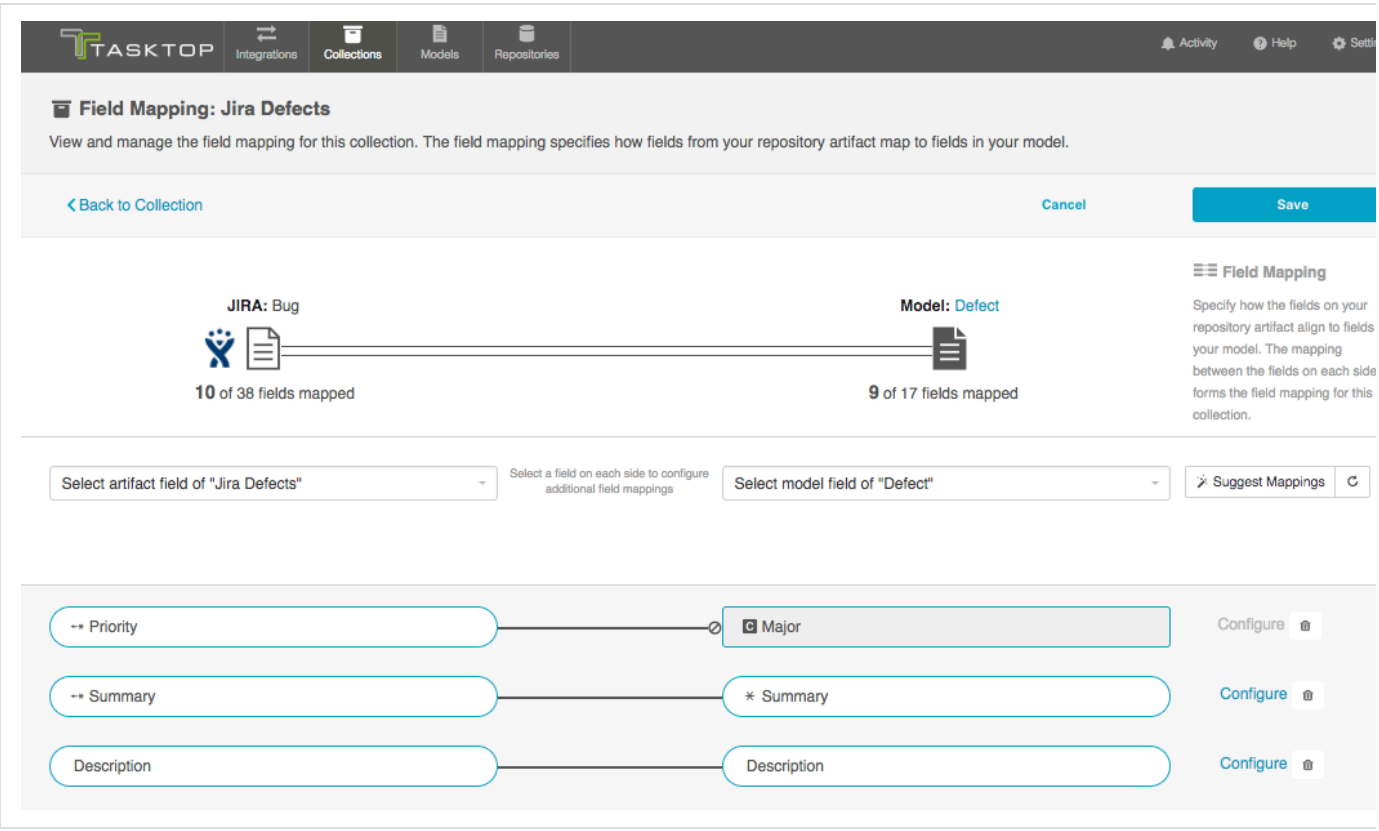
Solution: Set a constant value on the side of the model, to send to your collection.

To set a constant value for a field, select 'Constant Value' from the drop down menu on the model side. Enter the value, and then click the 'Set Constant Value' box.


TASKTOP
Integrations
Collections
Models
Repositories
Activity
Help
Settings
Field Mapping: Jira Defects
View and manage the field mapping for this collection. The field mapping specifies how fields from your repository artifact map to fields in your model.
Back to Collection
Cancel
Save
JIRA: Bug
Model: Defect
10 of 38 fields mapped
10 of 17 fields mapped
Field Mapping
Specify how the fields on your repository artifact align to fields in your model. The mapping between the fields on each side forms the field mapping for this collection.
Select artifact field of "Jira Defects"
Select a field on each side to configure additional field mappings
Select model field of "Defect"
Suggest Mappings
Constant Value
Constant Value Based on Projects
Modified By (Person)
Priority (Single Select)
Severity (Single Select)
Release Detected (Single Select)
Not Mapped
Priority
Clear Workspace
Key
Configure


TASKTOP
Integrations
Collections
Models
Repositories
Activity
Help
Settings
Field Mapping: Jira Defects
View and manage the field mapping for this collection. The field mapping specifies how fields from your repository artifact map to fields in your model.
Back to Collection
Cancel
Save
JIRA: Bug
Model: Defect
10 of 38 fields mapped
10 of 17 fields mapped
Field Mapping
Specify how the fields on your repository artifact align to fields in your model. The mapping between the fields on each side forms the field mapping for this collection.
Select artifact field of "Jira Defects"
Set Constant Value
Select model field of "Defect"
Suggest Mappings
Constant values are only set upon artifact creation
Priority
Select Constant Value
Blocker
Critical
Major
Minor
Trivial
Clear Workspace
Key
Summary
Configure
Configure





Once the constant value is set, you will notice a few things:

- The pill will be rectangular and grey: this denotes that a constant value has been set.
- The Constant Value icon will be displayed inside the pill.
- The 'prohibited' icon will appear next to the pill. This indicates that no values can be sent to the Constant Value field. The constant value is essentially a dead end, and cannot be linked to a repository or model on the other side.

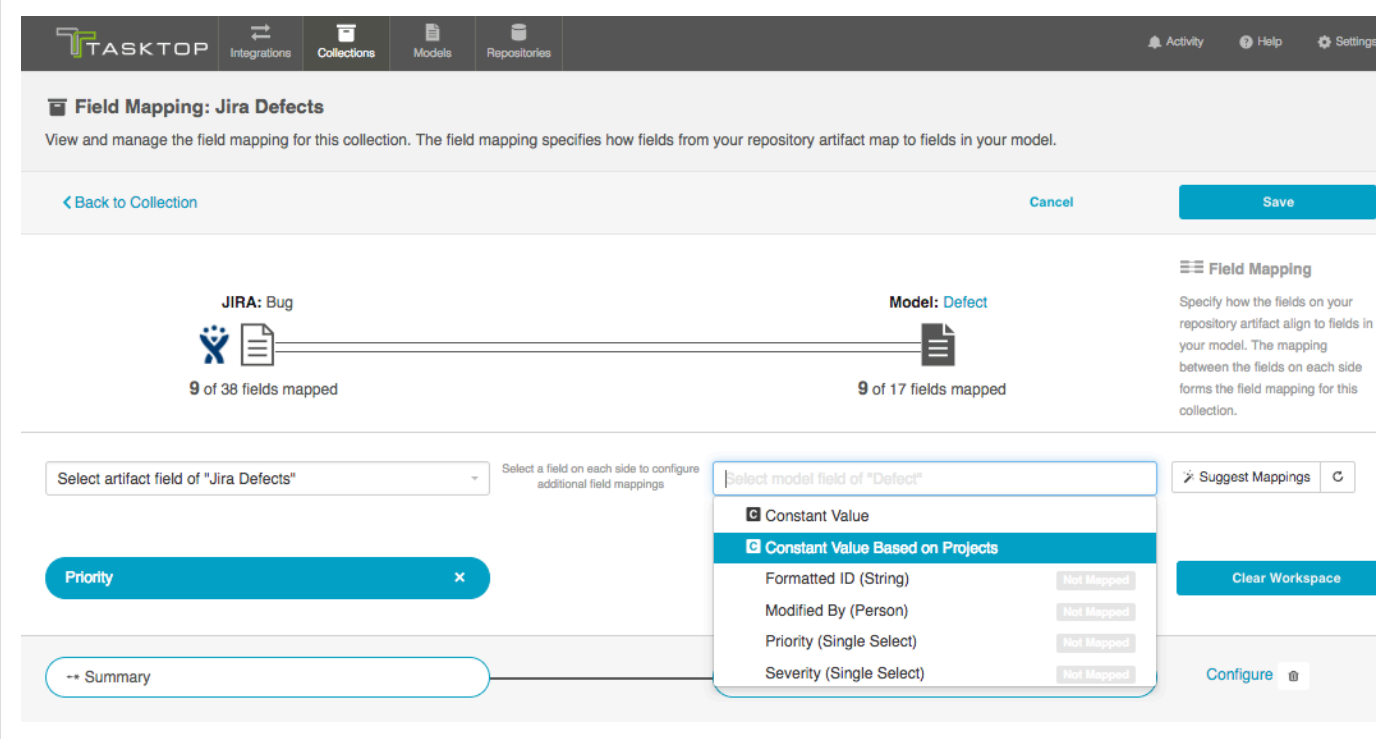
In the scenario above, any time a new defect is created in JIRA, the priority will be set to 'Major.' JIRA will not send 'priority' data to any other collections, as 'priority' does not exist in the model.

If desired, you can also set constant values per project:

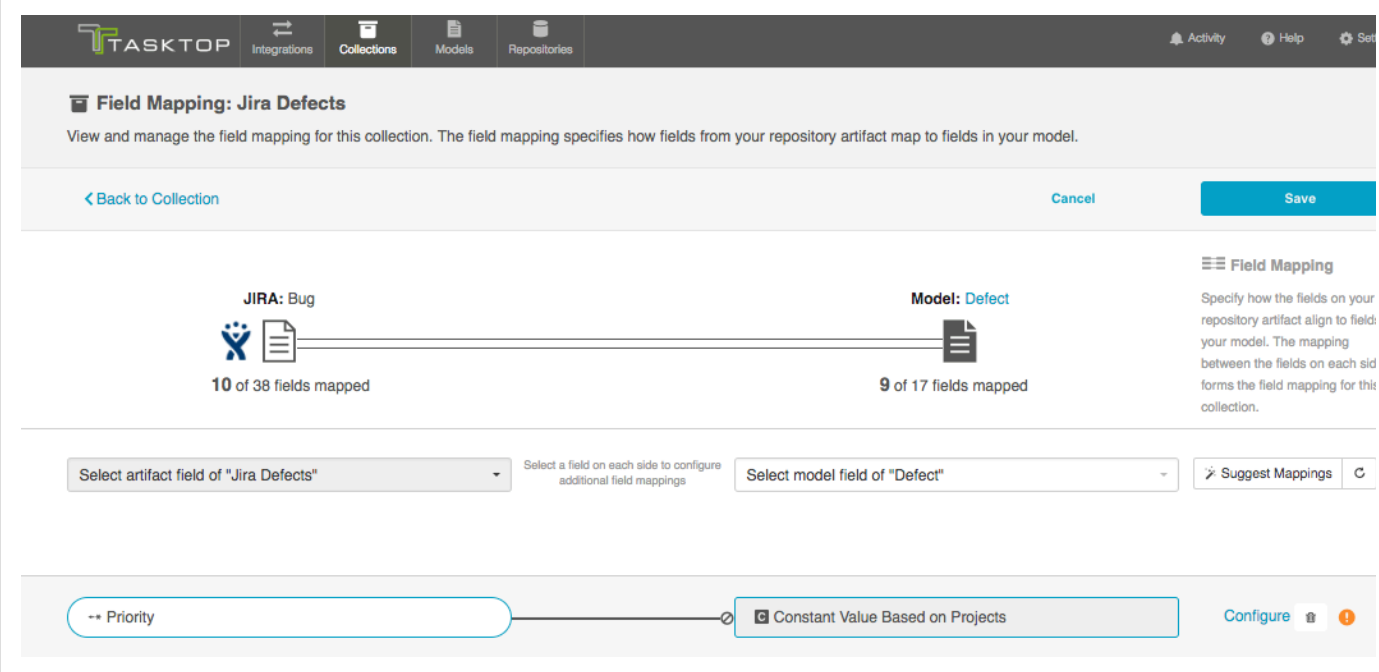
You may wish to set a constant value based on project in the following scenarios:

- In order to set a unique value for a specific field, such as release or iteration, depending on the project
- If the values for a single-select field vary across projects

To do this, select 'Constant Value Based on Projects':

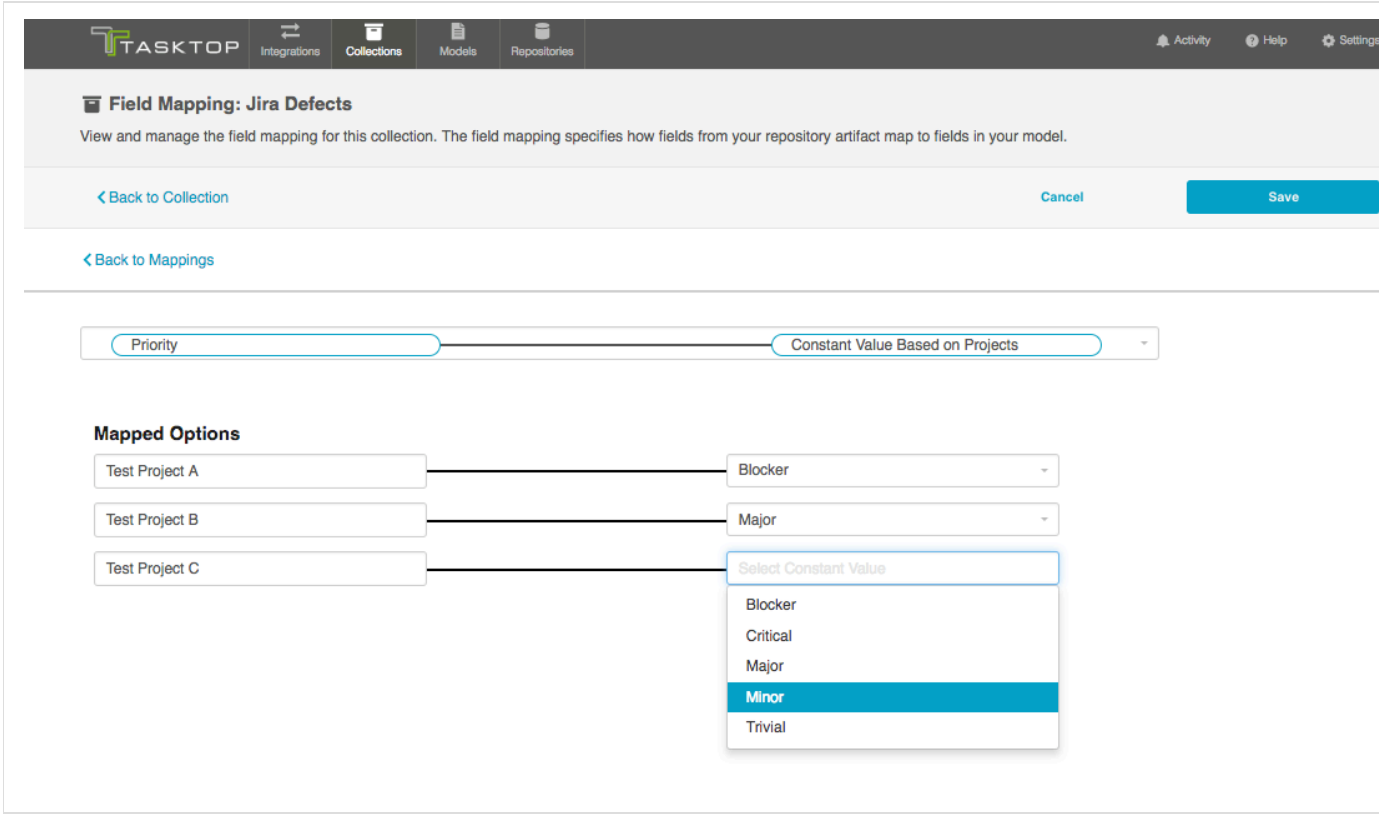


Once selected, you will see an orange exclamation point appear next to the 'Configure' link:

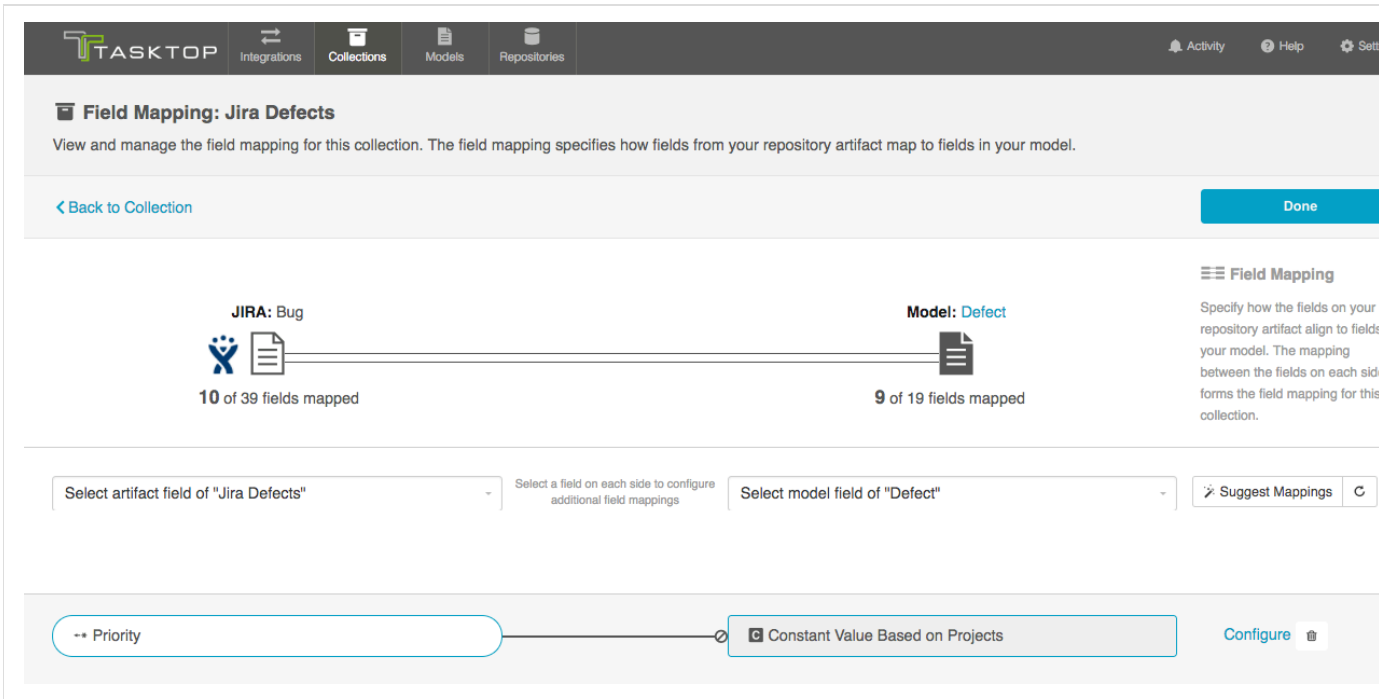


Click "Configure" to get to the Configuration Screen. On this screen, you will be able to set a distinct

constant value for each project in your collection:



In the screenshot above, an artifact created in Test Project A would be assigned the value of "Blocker" for the Priority field, while an artifact created in Test Project B would be assigned the value of "Major" for the Priority field. Finally, an artifact created in Test Project C would be assigned the value "Minor" for the Priority field.



Once the constant value is set, you will notice a few things:

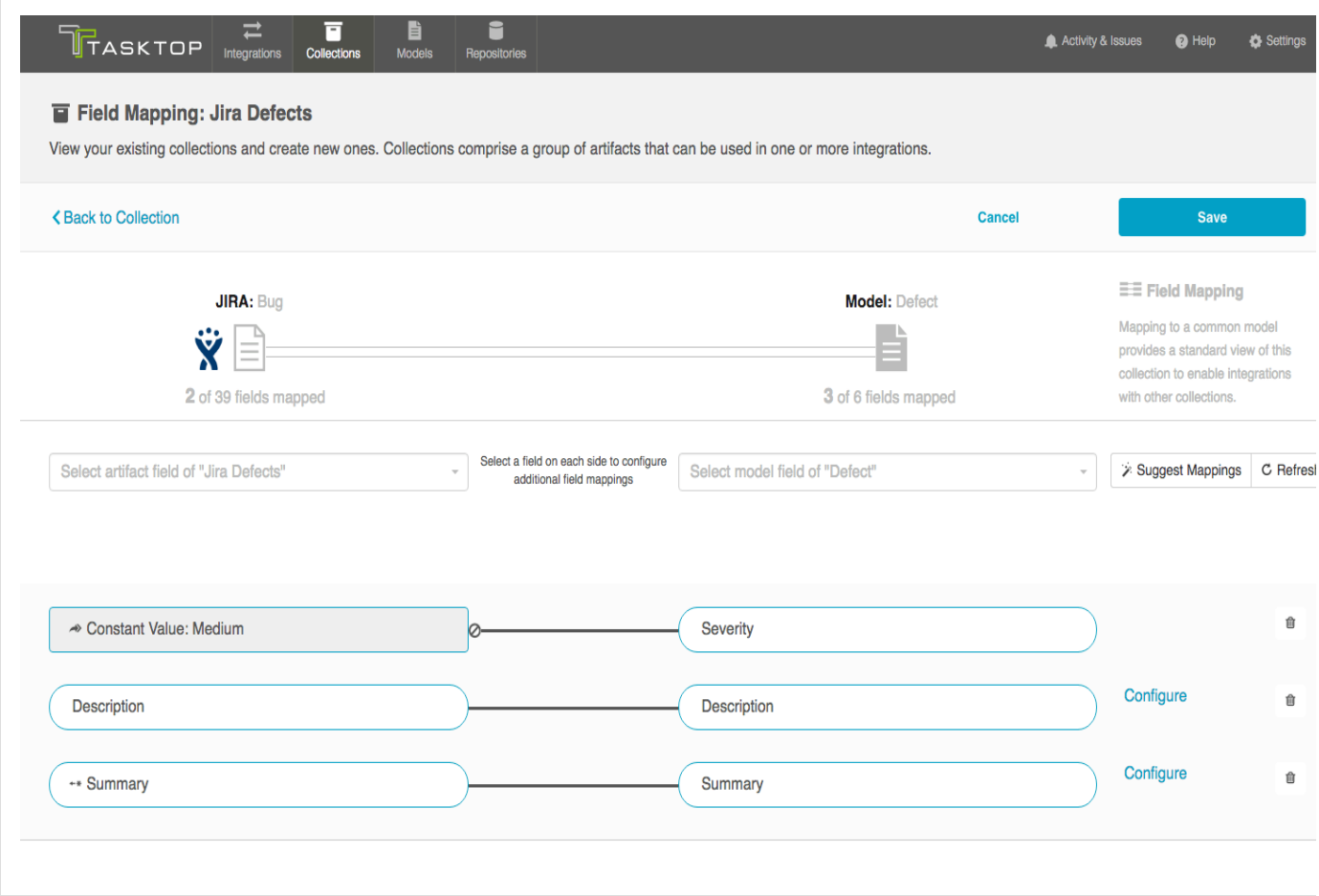
- The pill will be rectangular and grey: this denotes that a constant value has been set.
- The Constant Value icon will be displayed inside the pill.
- The 'prohibited' icon will appear next to the pill. This indicates that no values can be sent to the Constant Value field. The constant value is essentially a dead end, and cannot be linked to a repository or model on the other side.

Note: Sometimes, a single-select field in your collection will not return any values that you can select in the UI. In cases when this is true, and when the artifact will accept new values for that field, you will see a text input in which you can configure a constant value (instead of the traditional drop-down list for a single select).

Scenario 2: If your model requires a field, but the end repository utilized in your collection does not have that field:

Solution: Set a constant value on the collection side to send to your model. This means that any time this collection creates a corresponding artifact in another collection, the 'severity' field (in the example below) will be set to the constant value, in that other repository.

To set a constant value for a field, select 'Constant Value' from the drop down menu on the collection side. Enter the value, and then click the 'Set Constant Value' box.



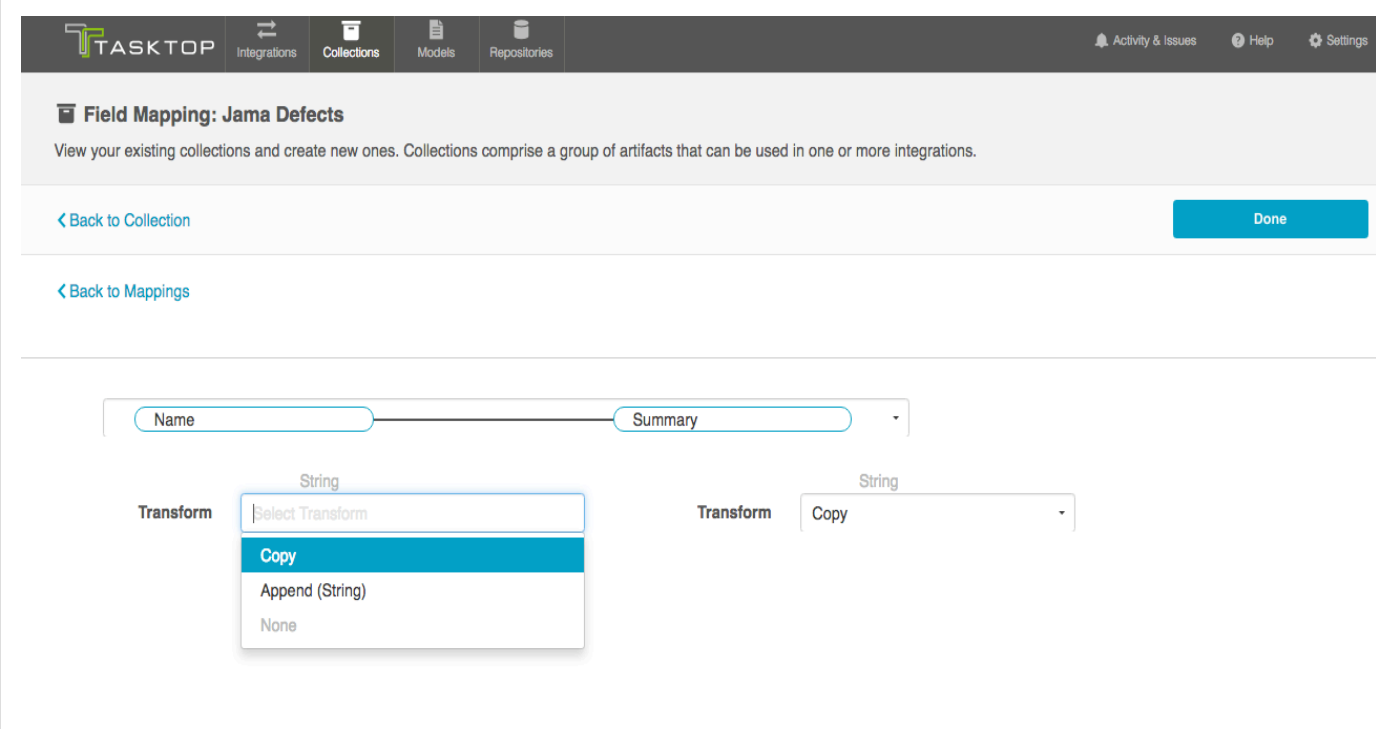
Once the constant value is set, you will notice a couple of things:

- The pill will be rectangular and grey: this denotes that a constant value has been set.
- The 'prohibited' icon will appear next to the pill. This indicates that no values can be sent to the Constant Value field. This makes sense, because in this example your repository did not have a

'severity' field to begin with.

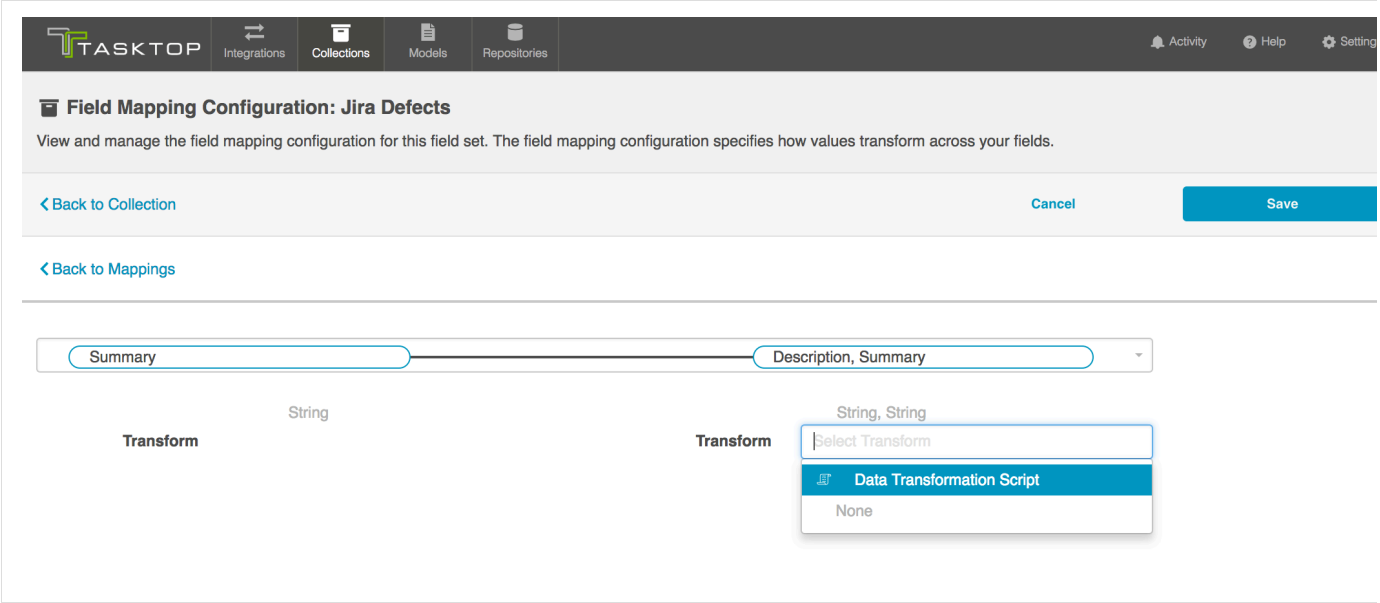
Transforms

On the Collection-to-Model field mapping screen, once you click 'Configure' next to a mapping pair, you will see the available transforms outlined. Similar attributes on different repositories often come in different formats, resulting in the need for values to be transformed to the proper format for a given repository. This screen allows you to configure how different types of fields will translate from one to the other.



If you have configured a Custom Data Transformation script, you can apply it here.

⚠️ *Note: Custom Data Transformations are available only for Enterprise and Ultimate Editions. Custom Data Transformations are currently available for Pro users for a limited time, in preview mode.*

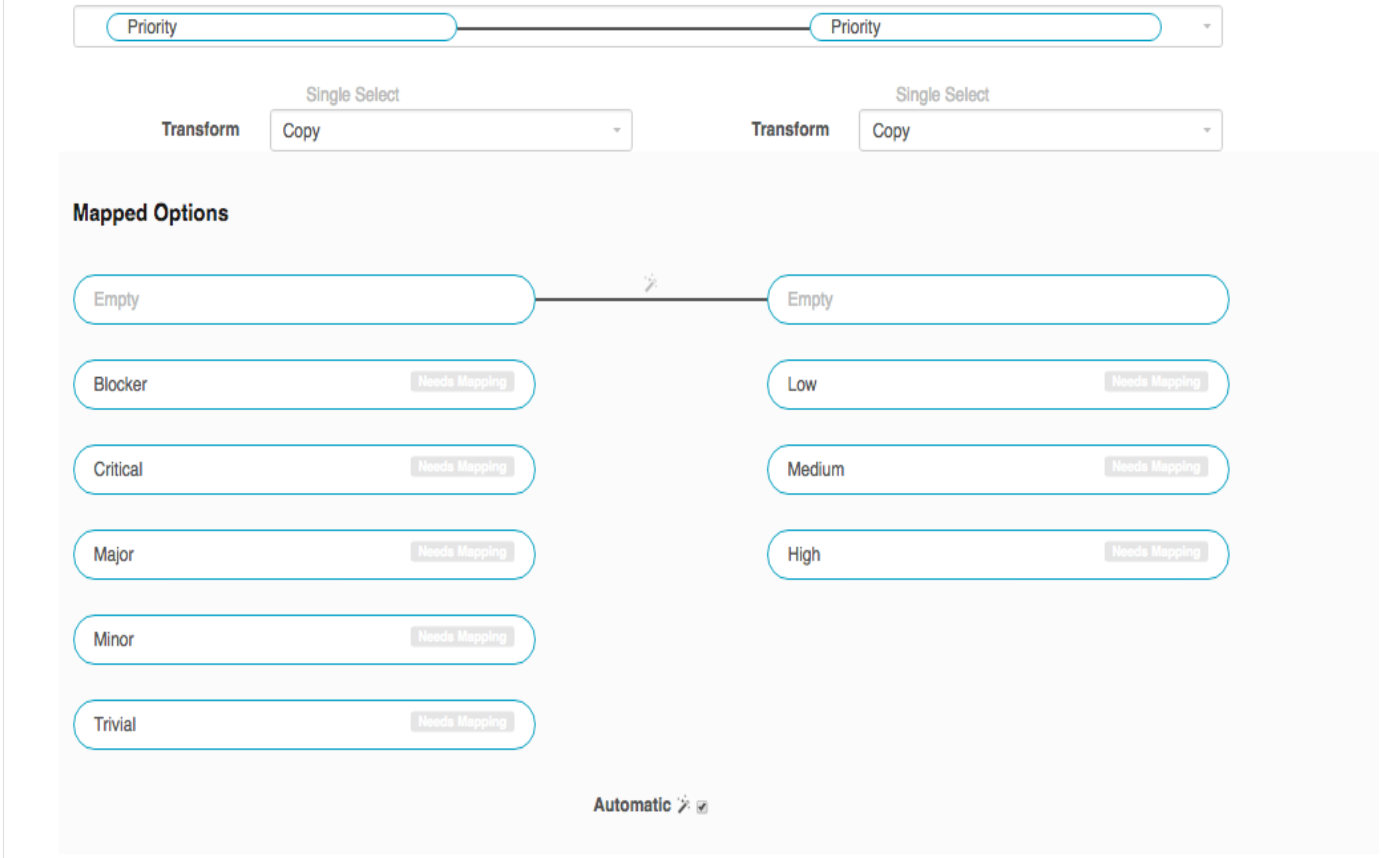


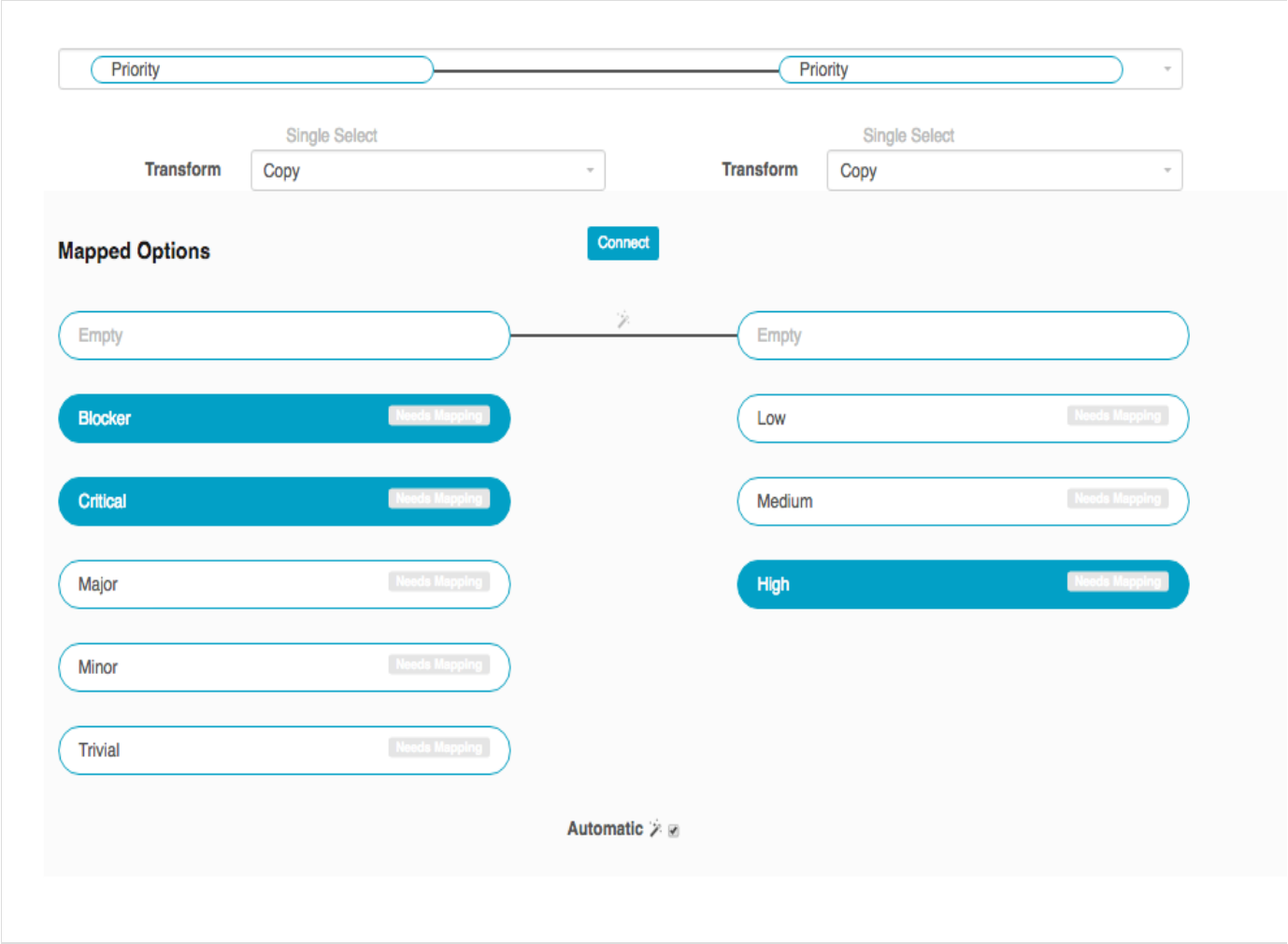
Single Select Fields

When flowing single-select fields, it is important to click the 'configure' button in order to map your field options to the model.

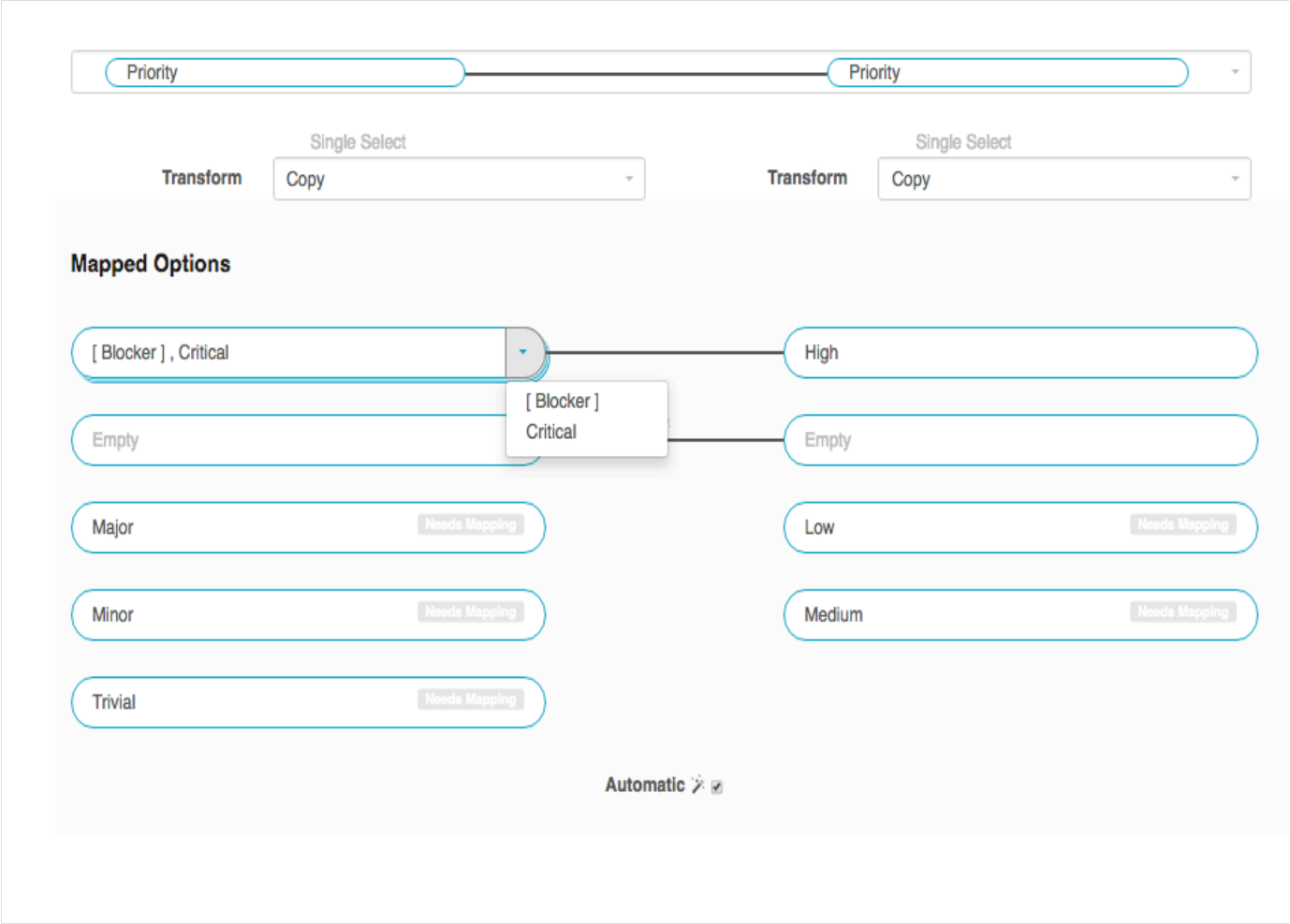
You will configure your transform as '*copy*' on both the collection and on the model side. This means that the model will pass an identical copy of its value to the collection, and vice versa.

To complete the field mapping, select the values in the collection and in the model that you would like to map to one another, and then click 'connect.' This process enables to the model to act as a 'translator' between two different collections which may have different sets of options for a single select field.


Priority
Priority
Single Select
Transform
Copy
Single Select
Transform
Copy
Mapped Options
Empty
Empty
Blocker
Needs Mapping
Low
Needs Mapping
Critical
Needs Mapping
Medium
Needs Mapping
Major
Needs Mapping
High
Needs Mapping
Minor
Needs Mapping
Trivial
Needs Mapping
Automatic



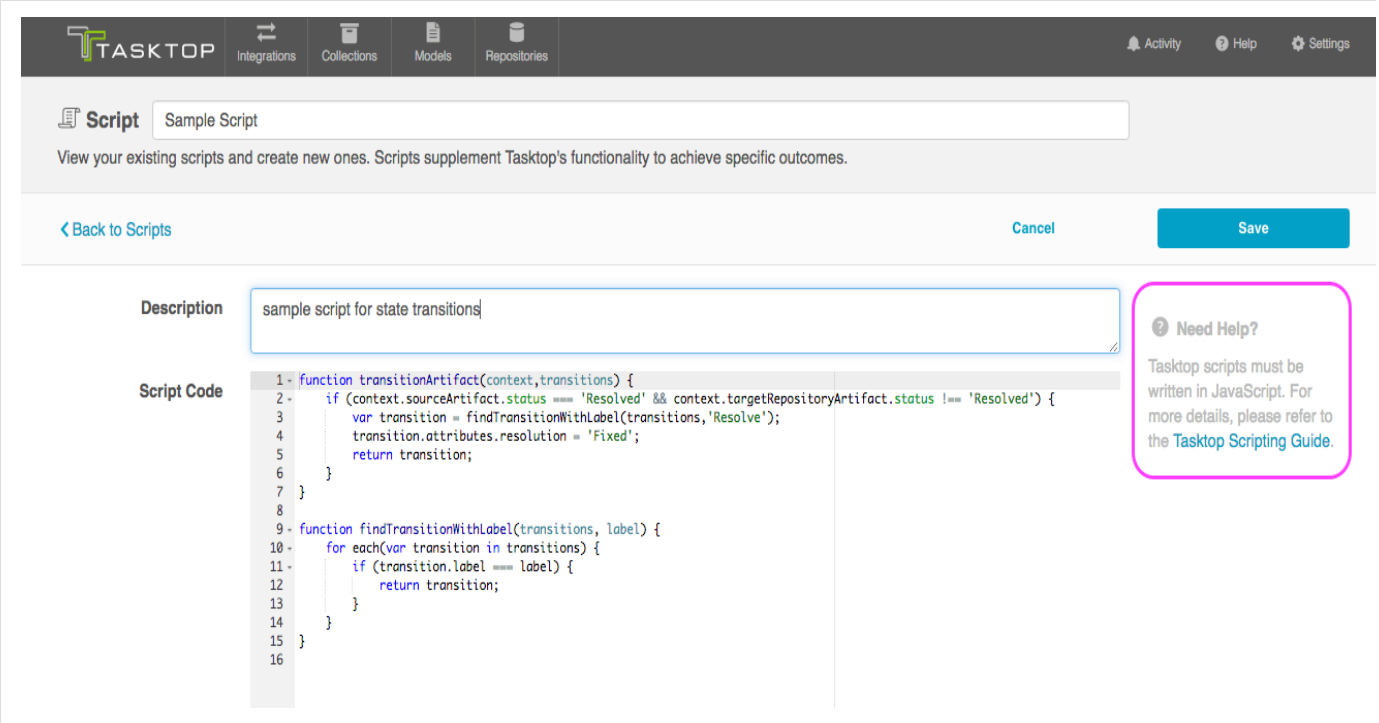
💡 When you map multiple collection values to a single model value, you will find that one value on the collection side is listed in brackets. This indicates which value will be set when the mapped model value is passed in. In the scenario below, if the model passes a 'high' priority value to your collection, that artifact will default to a priority status of 'blocker,' rather than 'critical.' You can modify the default value by clicking the arrow icon on the collection field pill.



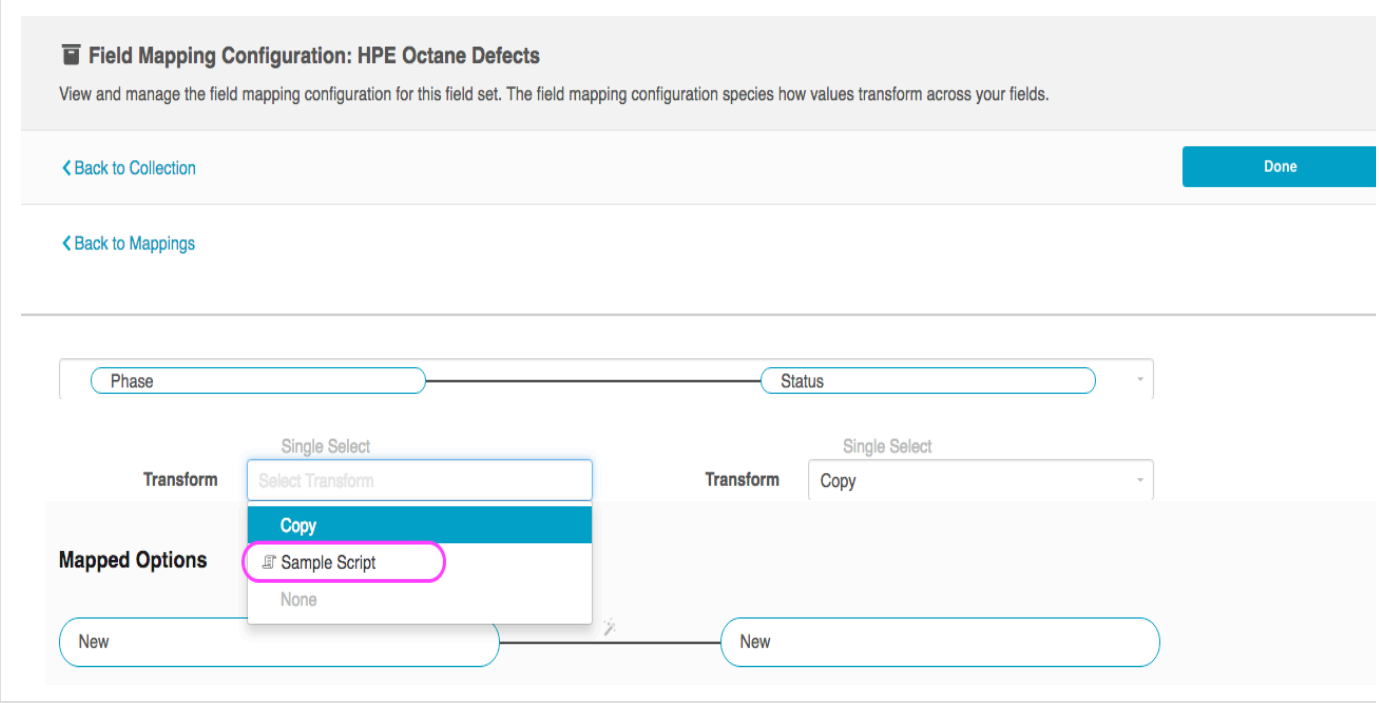
You can also map many model fields to a single collection field. The brackets on the model side similarly indicate which value will be set in the model when either of the mapped collection values are passed in.

State Transitions Scripting

To perform state transitions, a script can be used. Add a script from the Scripts page accessible from Settings. Once added, the script can be referenced as a transform from the corresponding repository collection Field Mapping Configuration page. To use a script, select Configure for the status field of a Field Mapping, and select the script as a transform on the status field. You can find more information on State Transitions Scripting in the Scripting Guide, linked on the New Script page.



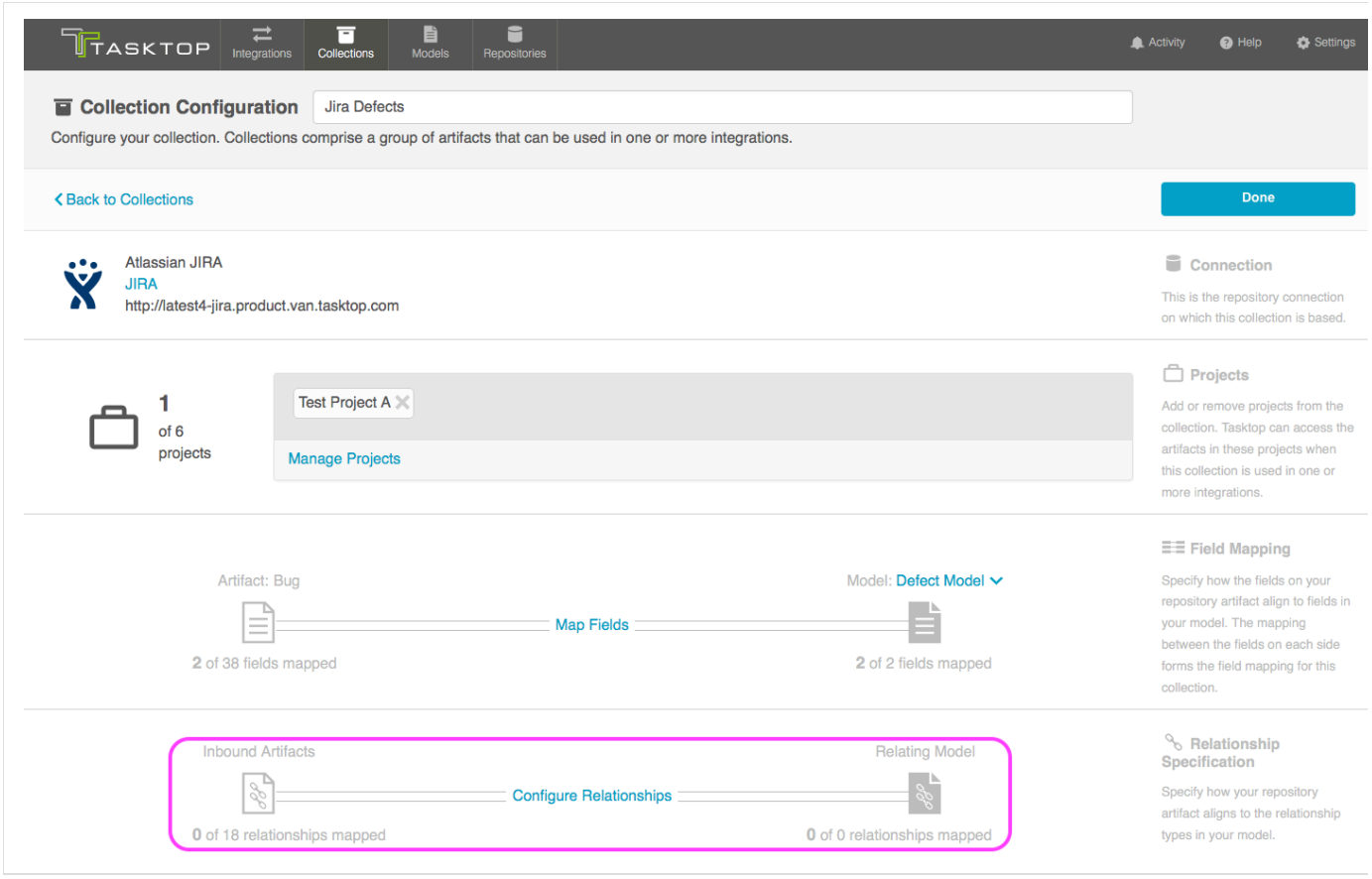
Below, you can see that the new script will appear as a 'transform' option on the Field Mapping Configuration page:

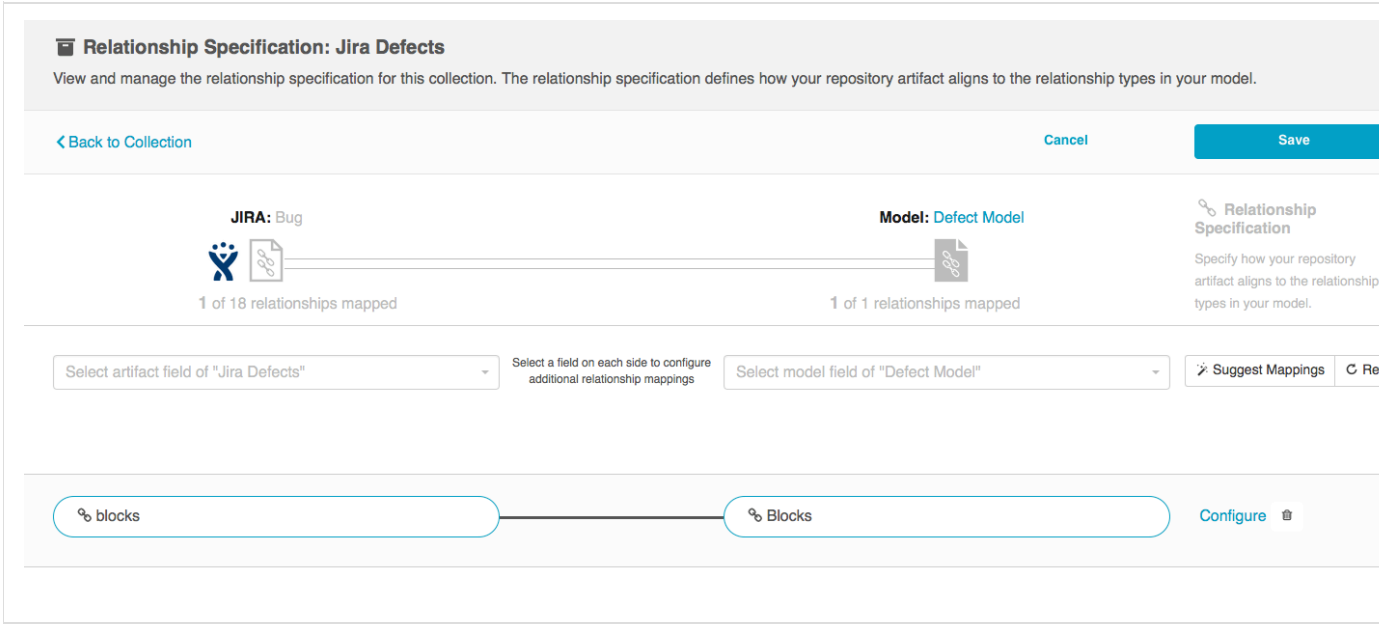


Configure Collection to Model Relationship Mapping

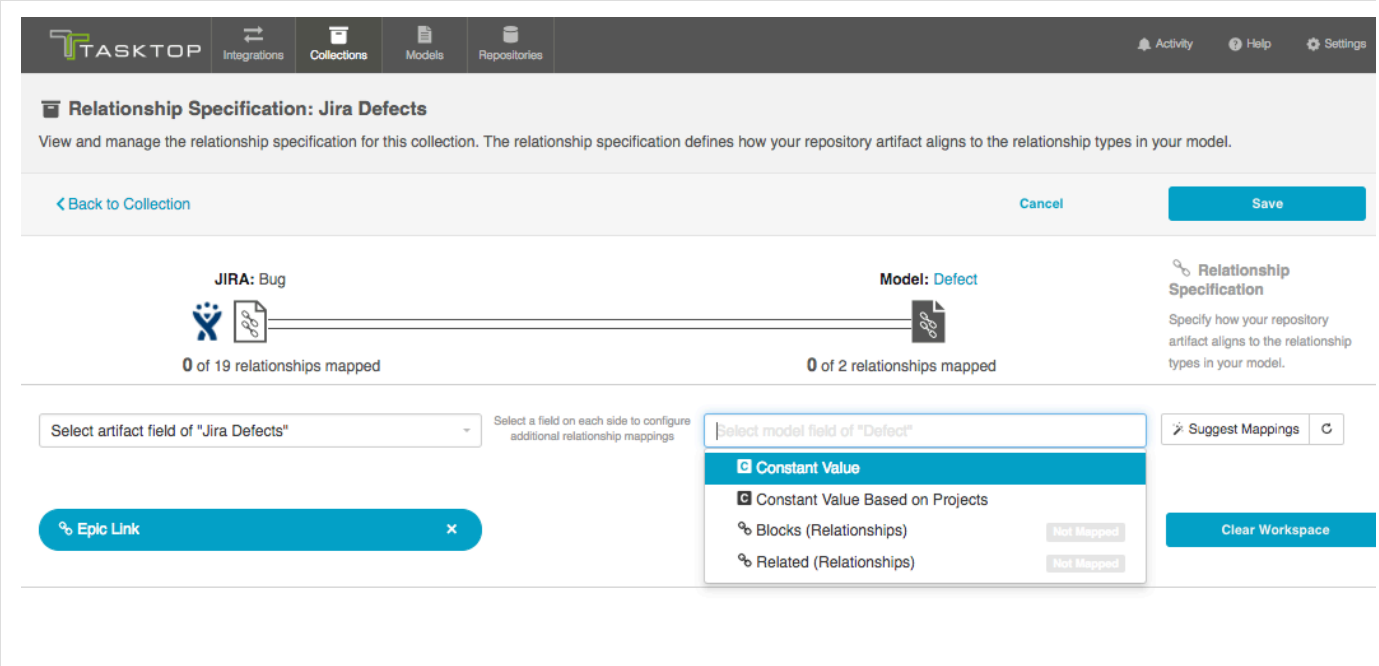
⚠️ *Note: Unlimited relationships are available only for Enterprise and Ultimate Editions. The Pro Edition currently has access to unlimited relationships for a limited time, in preview mode. Once the preview period ends, Pro users will revert to a limit of one relationship type.*

If you have any relationship(s) fields in your model, you can map those by clicking the "Configure Relationships" link.



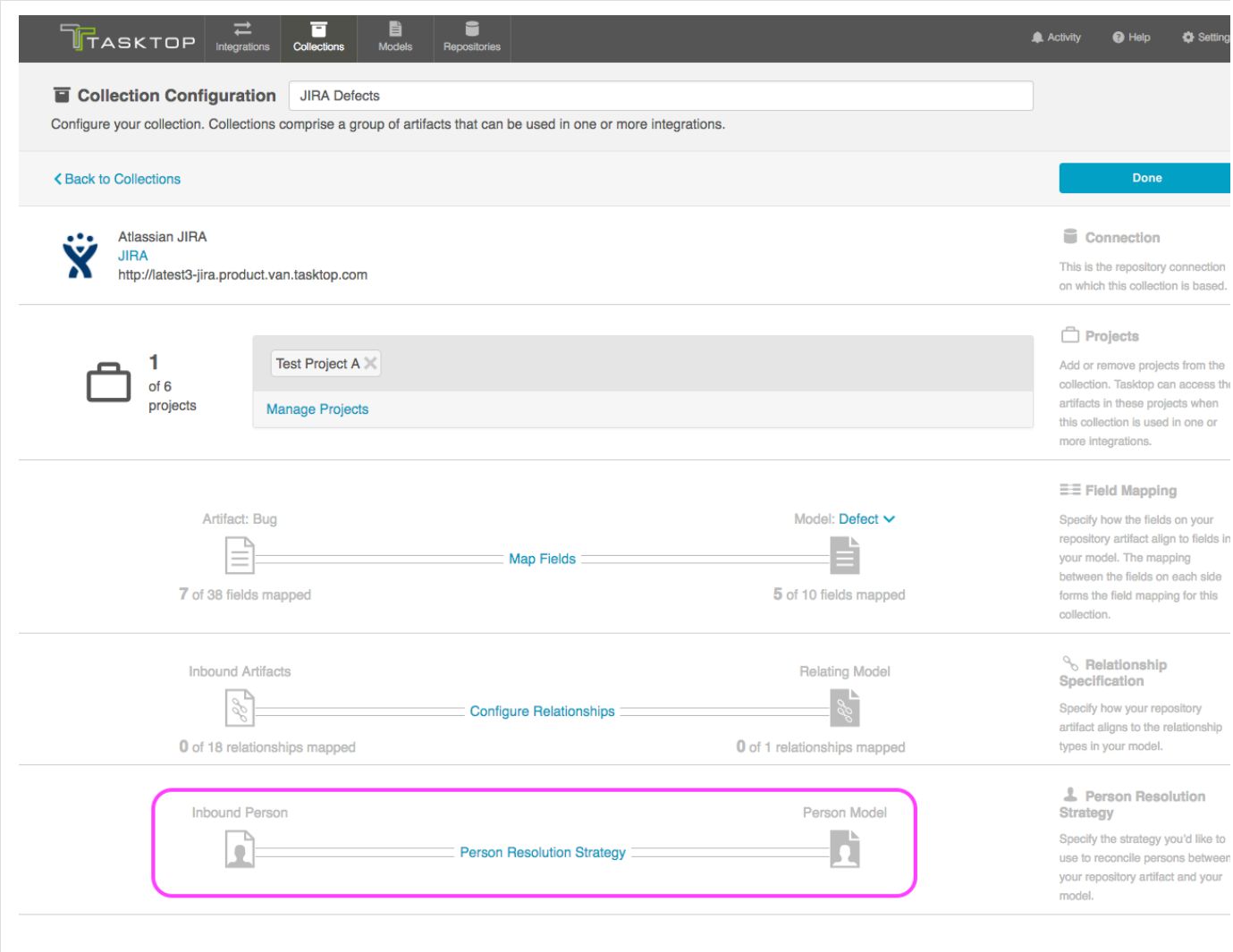


For 'relationship' type fields, you have the option of configuring constant values. To learn how to configure constant values, please reference the constant value section above.



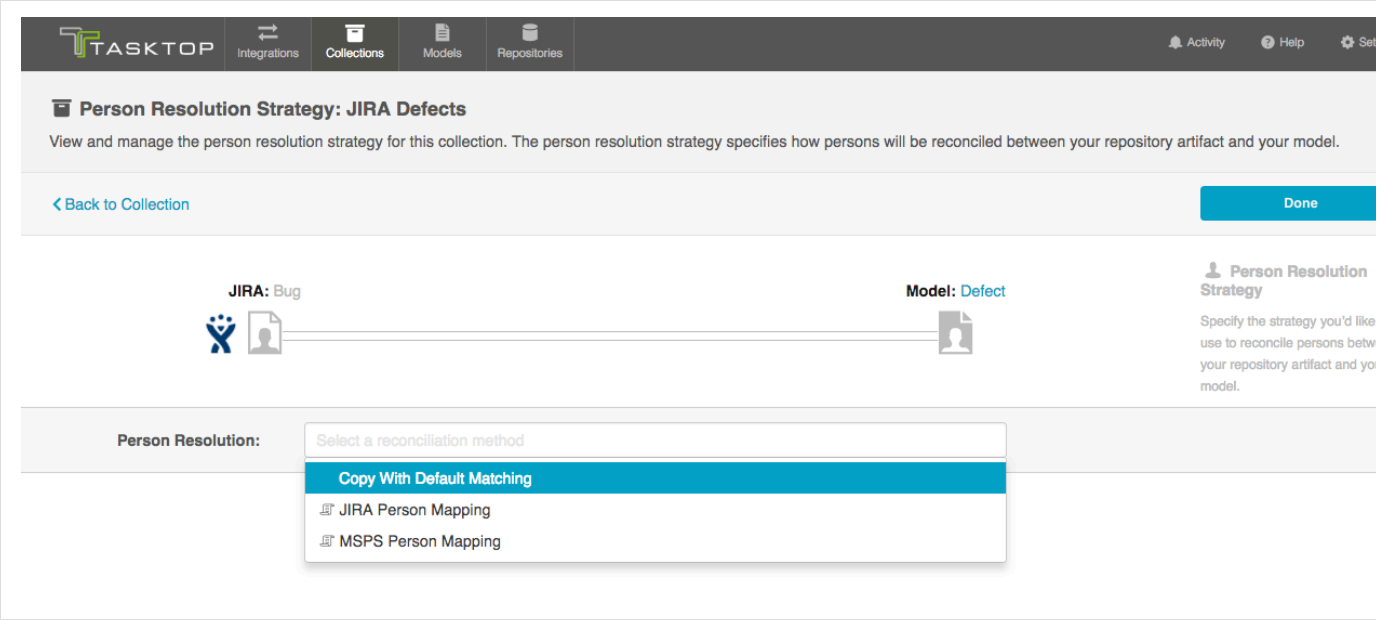
Configure Person Resolution Strategy

To configure your Person Resolution Strategy, click the 'Person Resolution Strategy' link.



If you have configured a Person Mapping script on the settings page, you will be able to select that script here, or to choose our default person resolution strategy ("Copy with Default Matching"). Our default algorithm will match based on name, ID, and/or e-mail.

⚠️ *Note: Custom Person Mapping Scripting is available only for Enterprise and Ultimate Editions. Custom Person Mapping Scripting is currently available for Pro users for a limited time, in preview mode.*



Optional: Set a Repository Query

You can learn more about Repository Queries here.

Create a Database Collection

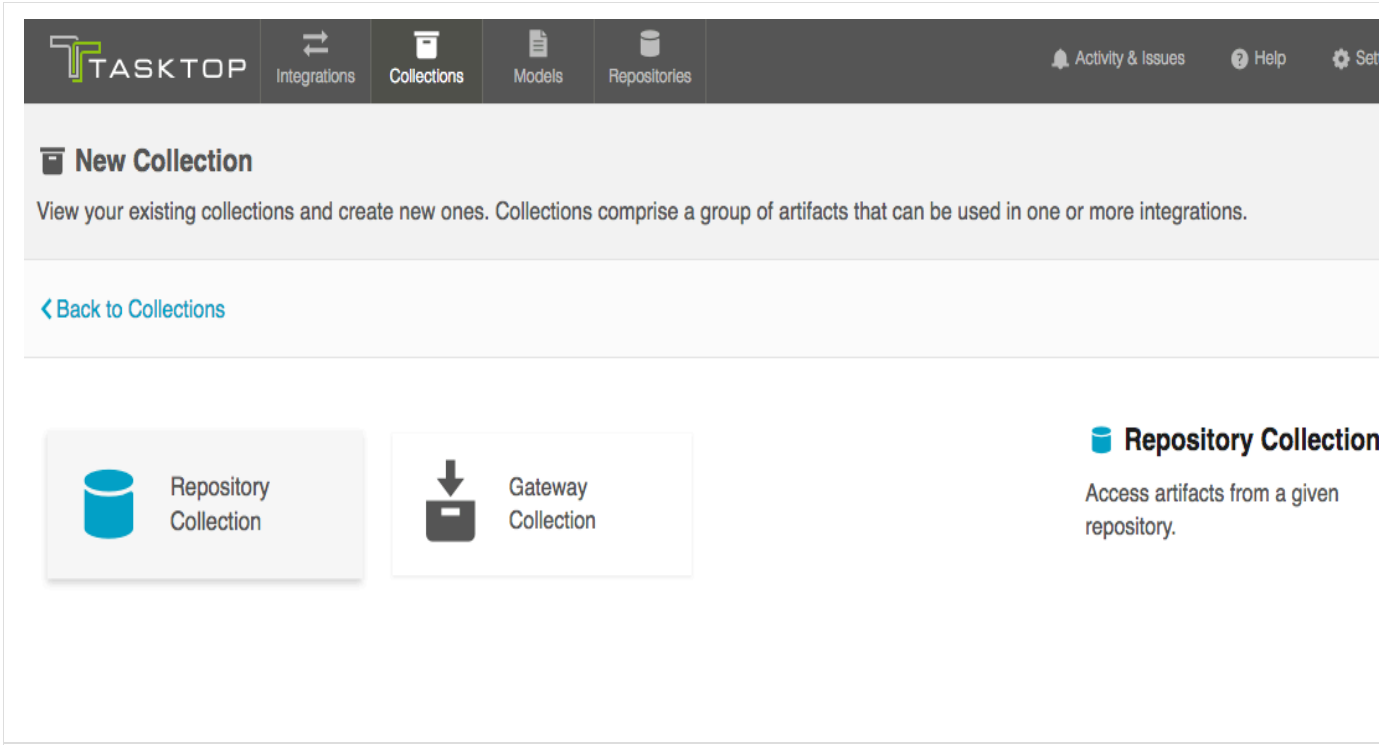
Check out the video below to learn how to create a new collection for your database repository:

💡 Remember that a Database Collection is a type of Repository Collection

Identify Collection Type

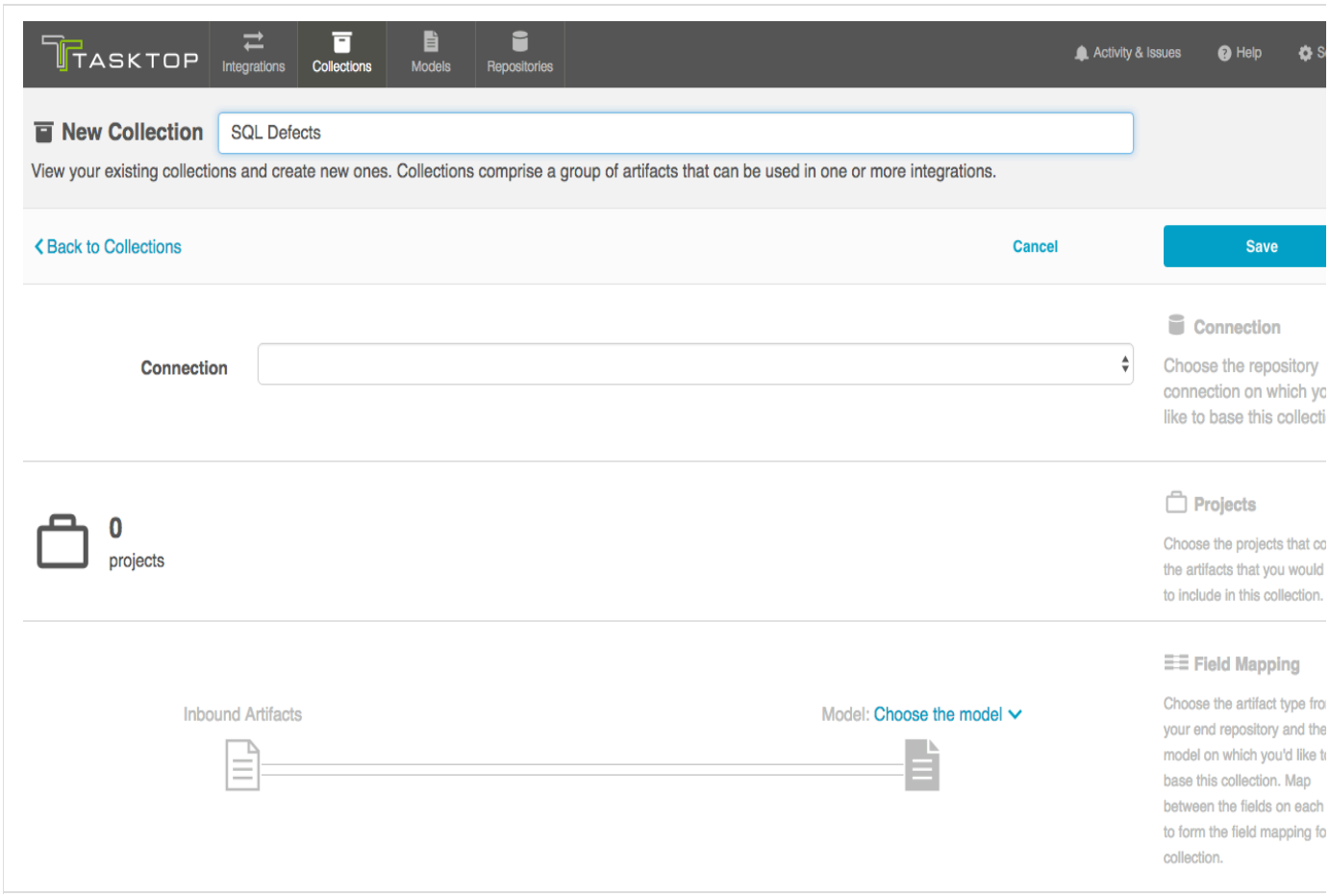
To create a database collection, do the following steps after you go to "New Collection":

Select "Repository Collection" as the collection type.



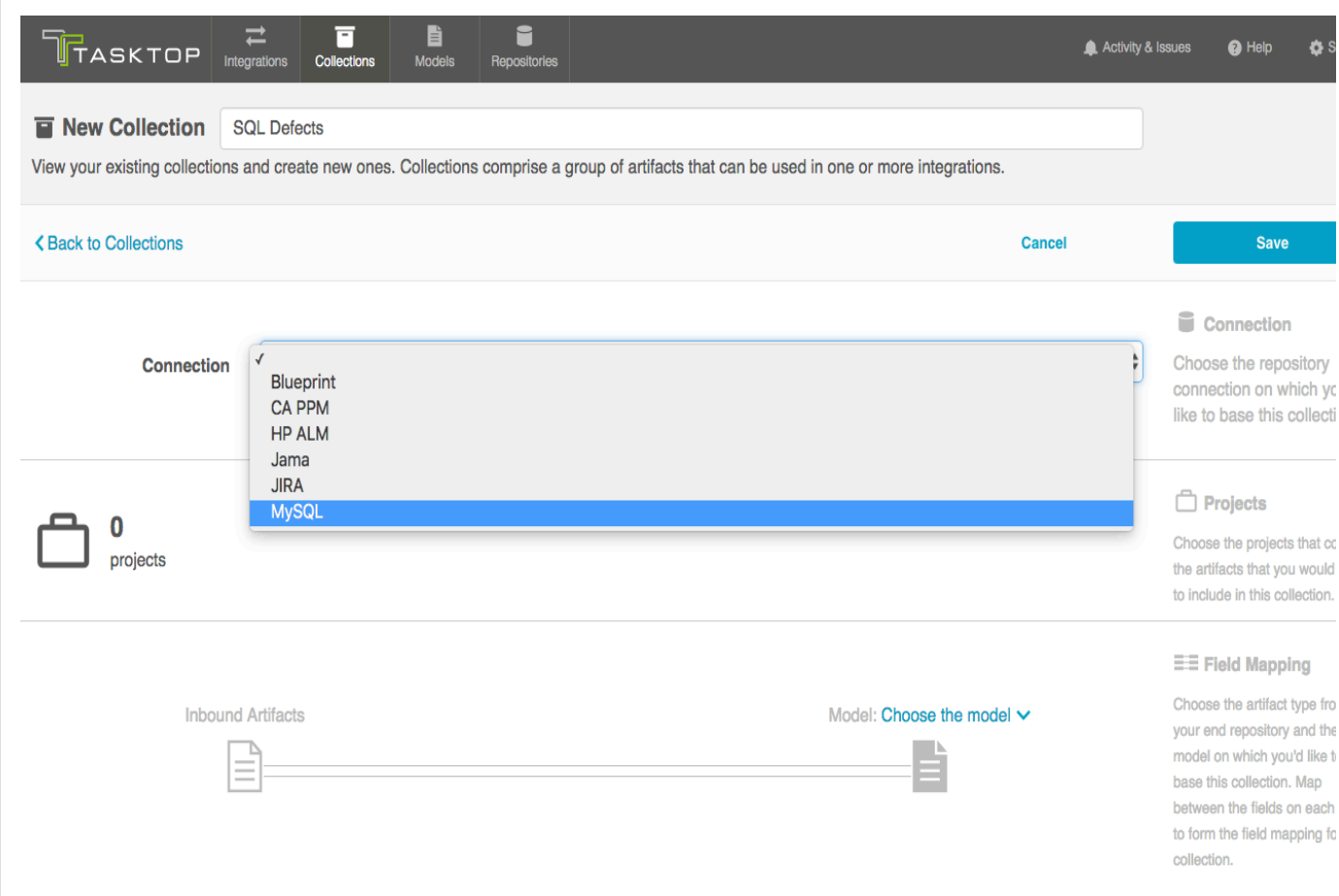
Name Collection

Enter a name for your collection



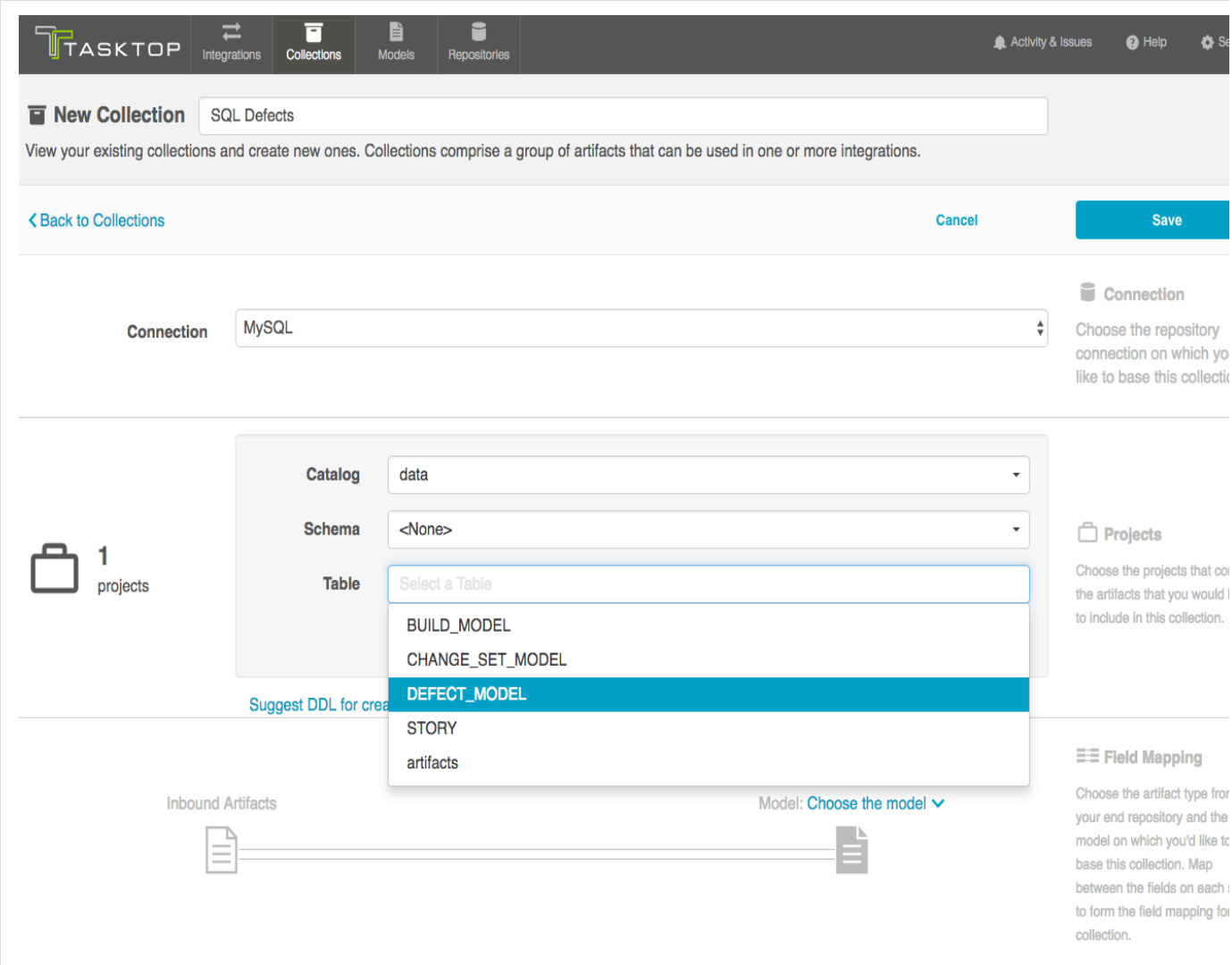
Select Repository Connection

Select the Repository Connection on which you'd like to base this collection. The collection will include artifacts from the repository collection you have selected.

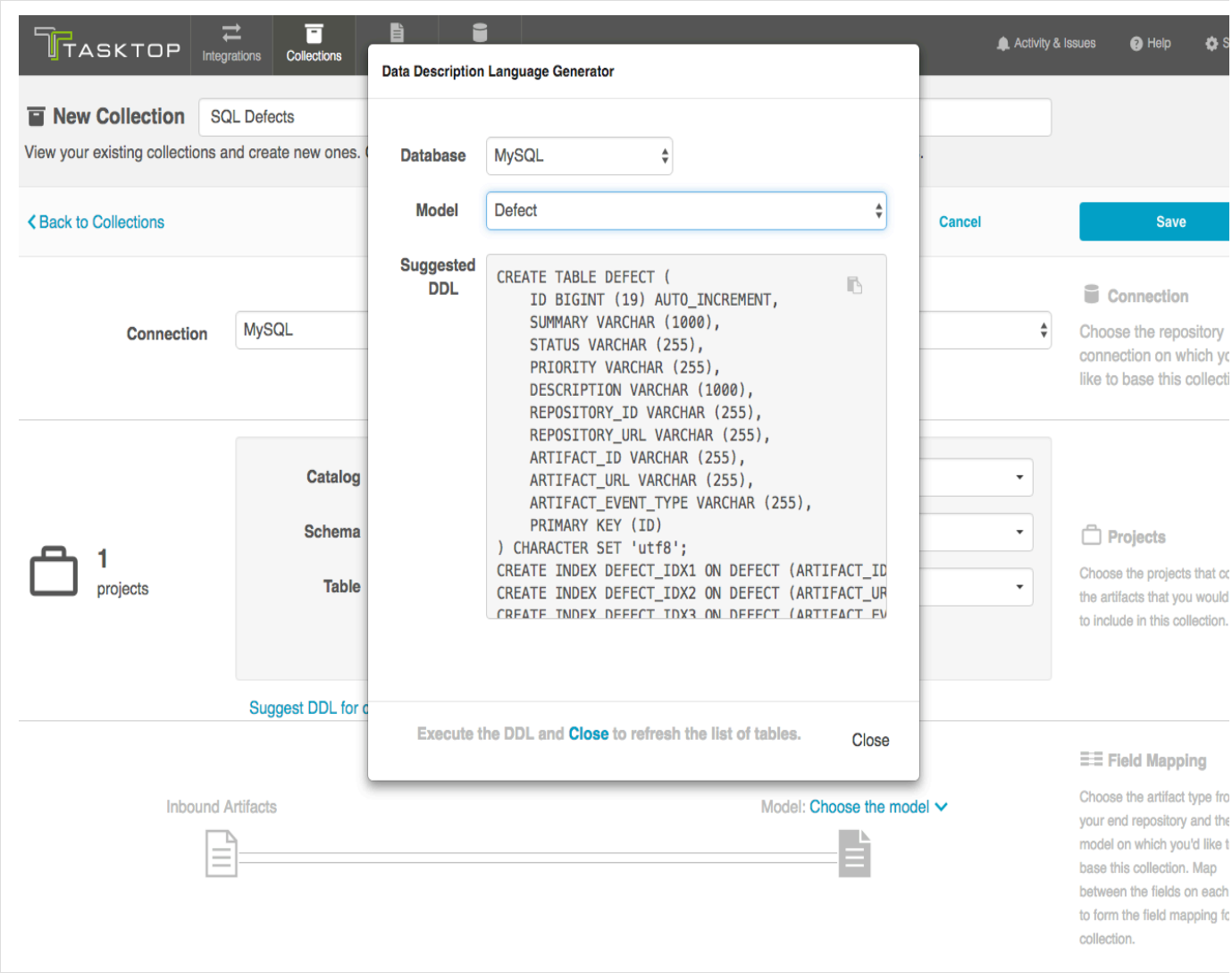


Identify Database Table

Choose the database table that will receive artifacts pointed at this collection.

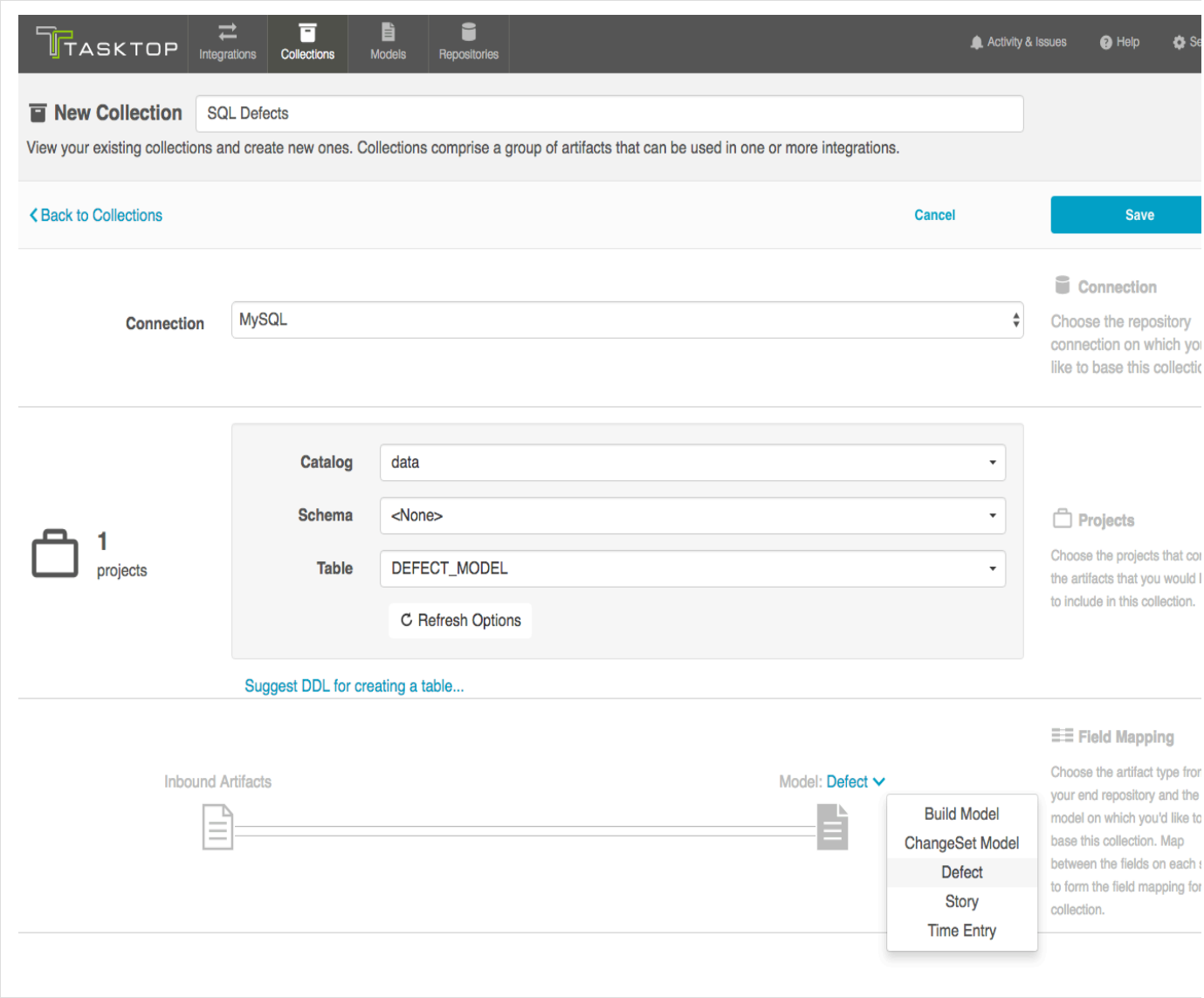


Note–if your table is not listed, you can use the "Suggest DDL" tool to generate a SQL command that can help you create a table that aligns with the model on which you'd like to base this collection.



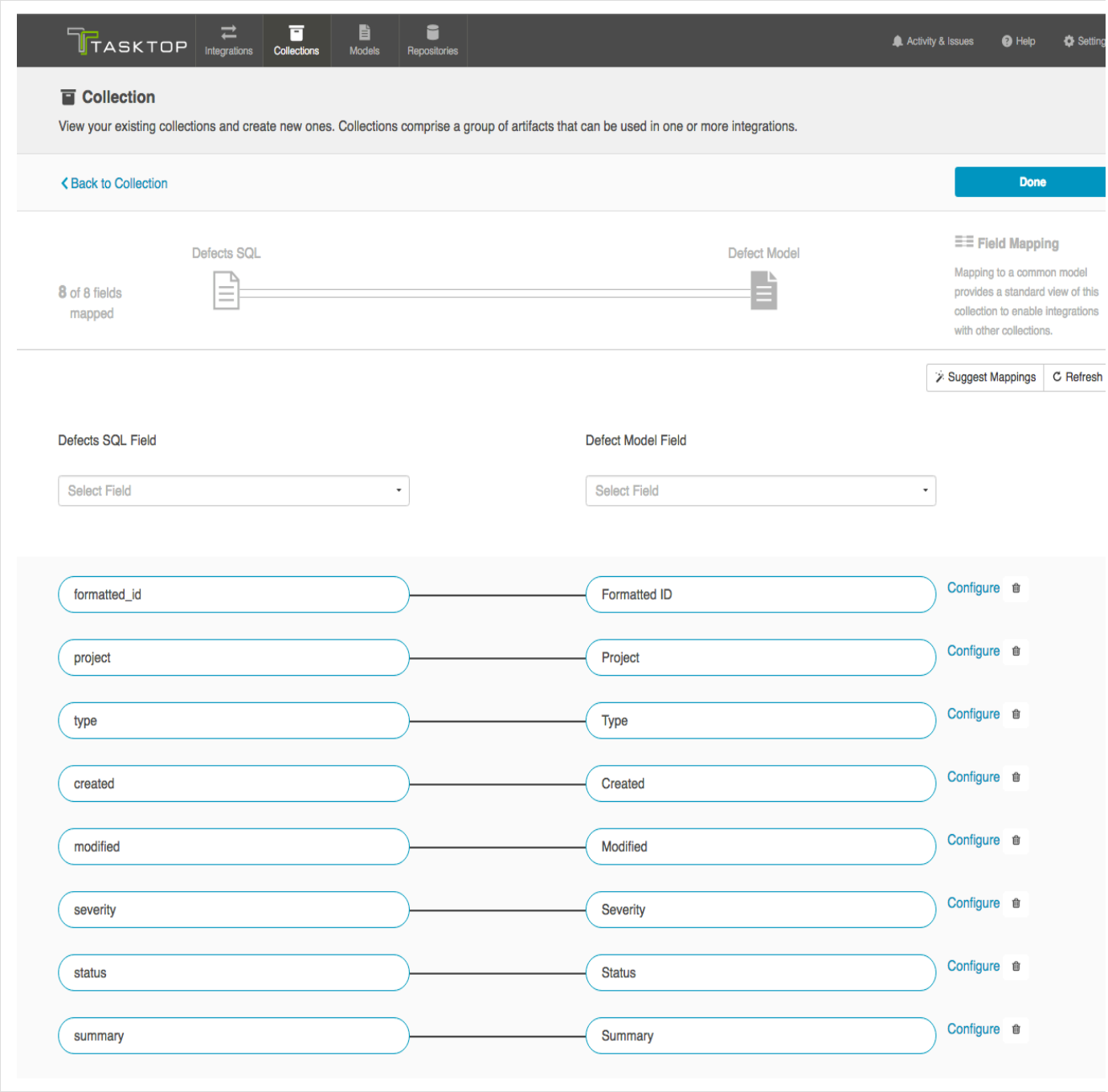
Identify the Collection Model

Select the Model on which you'd like to base this collection.



Verify/Complete the Collection to Model Mapping

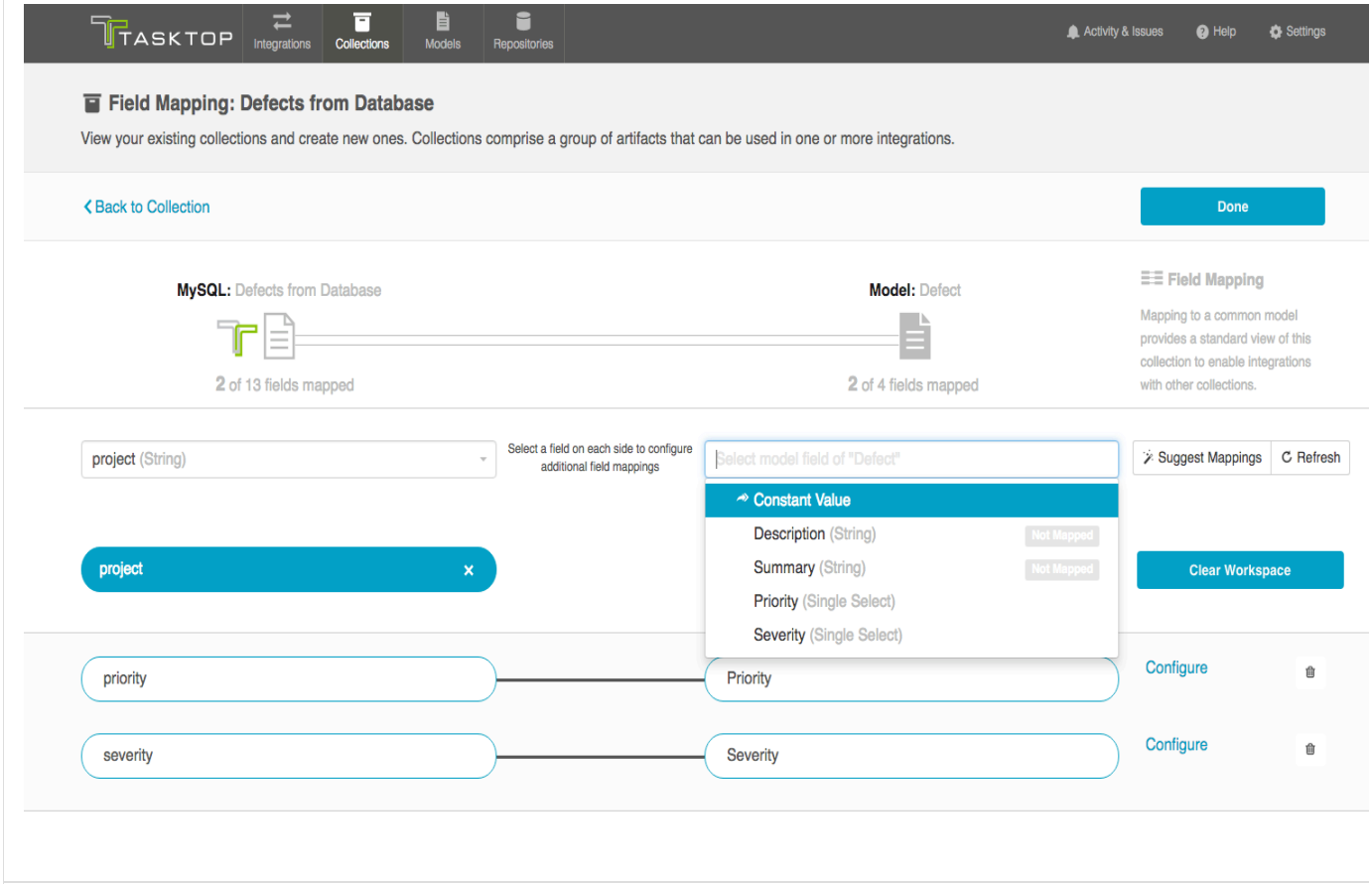
Now that you have identified the collection artifact type and model, you can complete the collection to model field mapping by going into the "Map Fields" link. Note-- if you used the Suggest DDL tool to create your database table, the mapping will be done automatically.

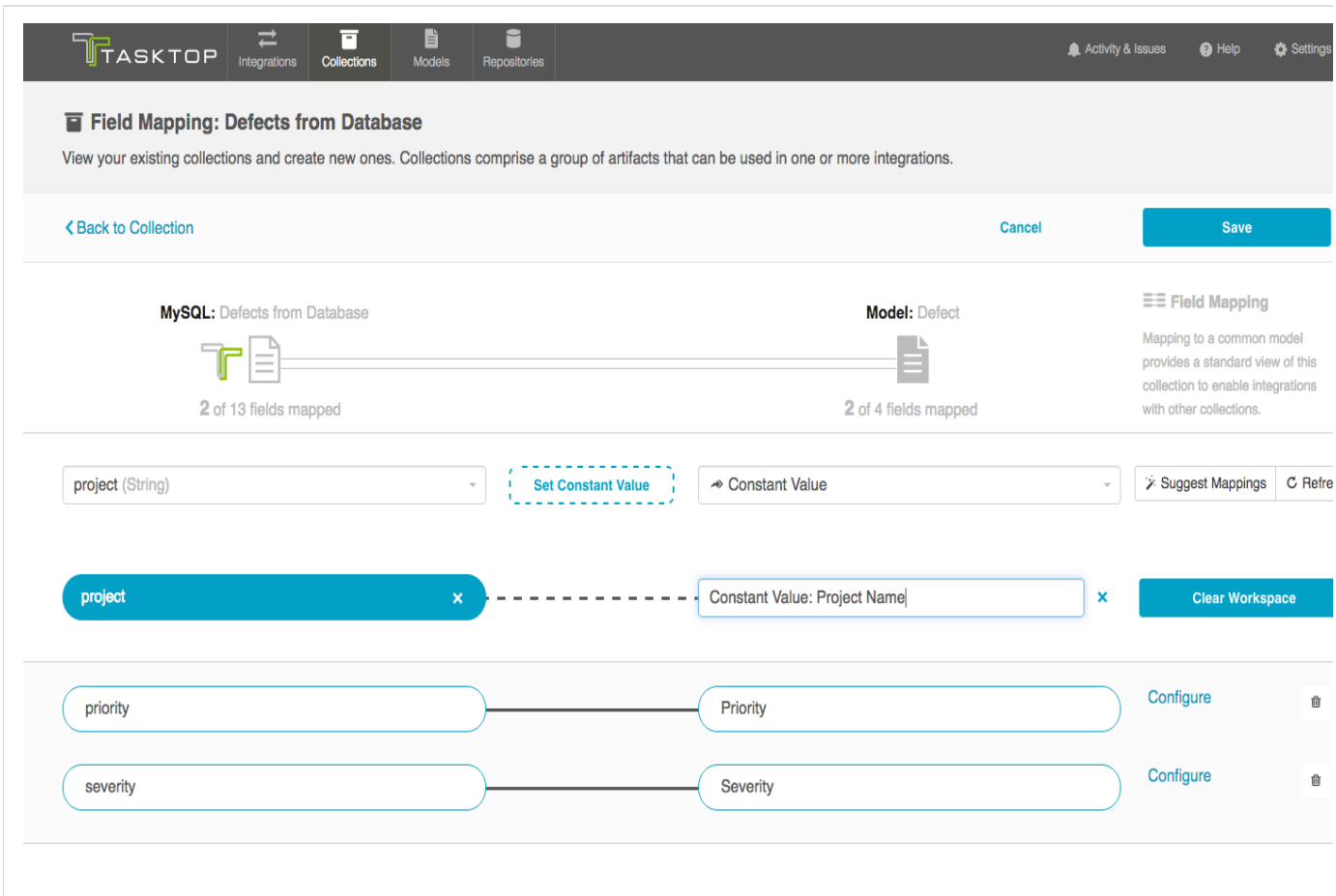


Constant Value Mapping

In some scenarios, the database might require that some of its columns/fields always have a value. This value is usually provided by mapping it to the equivalent model field. When there is no equivalent field in the model that can provide a value, you can set a constant value into your end-database column/field. The value you configure will then always get written out.

To set a constant value for a field (either from the collection artifact or from the model), select the 'Constant Value' option from the drop down menu. Enter the value, and then click the 'Set Constant Value' box.


TASKTOP
Integrations
Collections
Models
Repositories
Activity & Issues
Help
Settings
Field Mapping: Defects from Database
View your existing collections and create new ones. Collections comprise a group of artifacts that can be used in one or more integrations.
Back to Collection
Done
MySQL: Defects from Database
2 of 13 fields mapped
Model: Defect
2 of 4 fields mapped
Field Mapping
Mapping to a common model provides a standard view of this collection to enable integrations with other collections.
project (String)
Select a field on each side to configure additional field mappings
Select model field of "Defect"
Suggest Mappings
Refresh
Constant Value
Description (String)
Not Mapped
Summary (String)
Not Mapped
Priority (Single Select)
Severity (Single Select)
project
Clear Workspace
priority
Priority
Configure
severity
Severity
Configure


TASKTOP
Integrations
Collections
Models
Repositories
Activity & Issues
Help
Settings
Field Mapping: Defects from Database
View your existing collections and create new ones. Collections comprise a group of artifacts that can be used in one or more integrations.
Back to Collection
Cancel
Save
MySQL: Defects from Database
2 of 13 fields mapped
Model: Defect
2 of 4 fields mapped
Field Mapping
Mapping to a common model provides a standard view of this collection to enable integrations with other collections.
project (String)
Set Constant Value
Constant Value
Suggest Mappings
Refre
project
Constant Value: Project Name
Clear Workspace
priority
Priority
Configure
severity
Severity
Configure

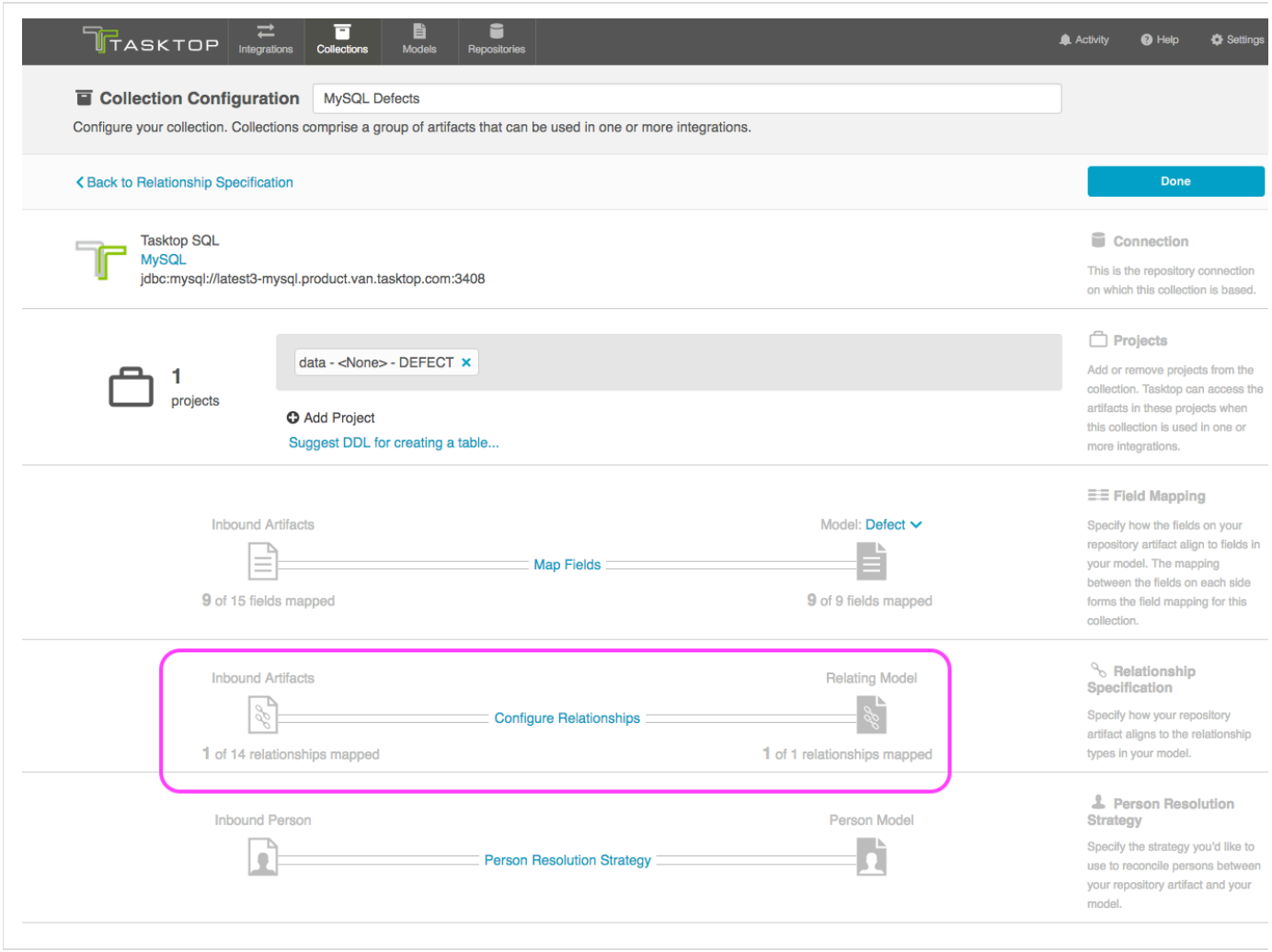
Note: Constant values can be set for the following fields types:

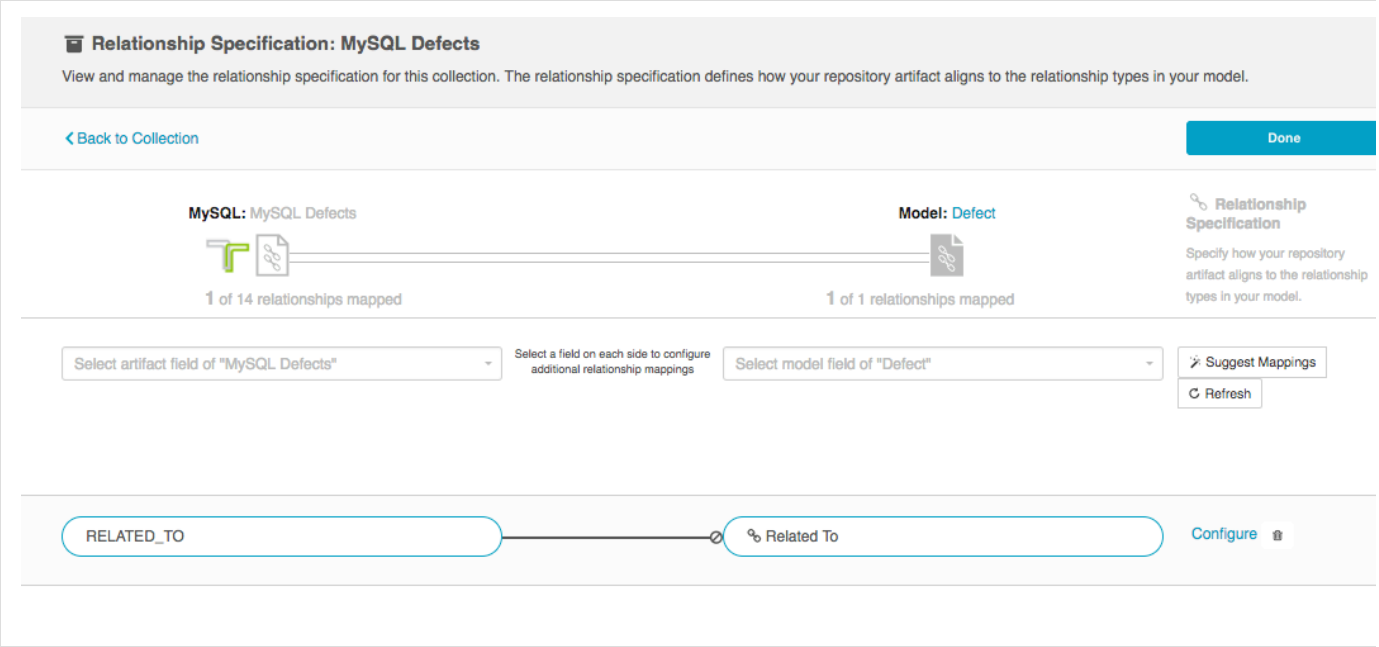
- Single Select
- String
- Date/DateTime
- Person
- Rich Text
- Double
- Long
- Boolean
- Location

Only some of these types are relevant for your database collection, however, given the field types that can be configured in the database itself.

Configure Collection to Model Relationship Mapping

If you have any relationship(s) fields in your model, you can map those on the "Configure Relationship Types" screen of a given collection. Note-- if you used the Suggested DDL tool to create your database table, the mapping should be done generally.





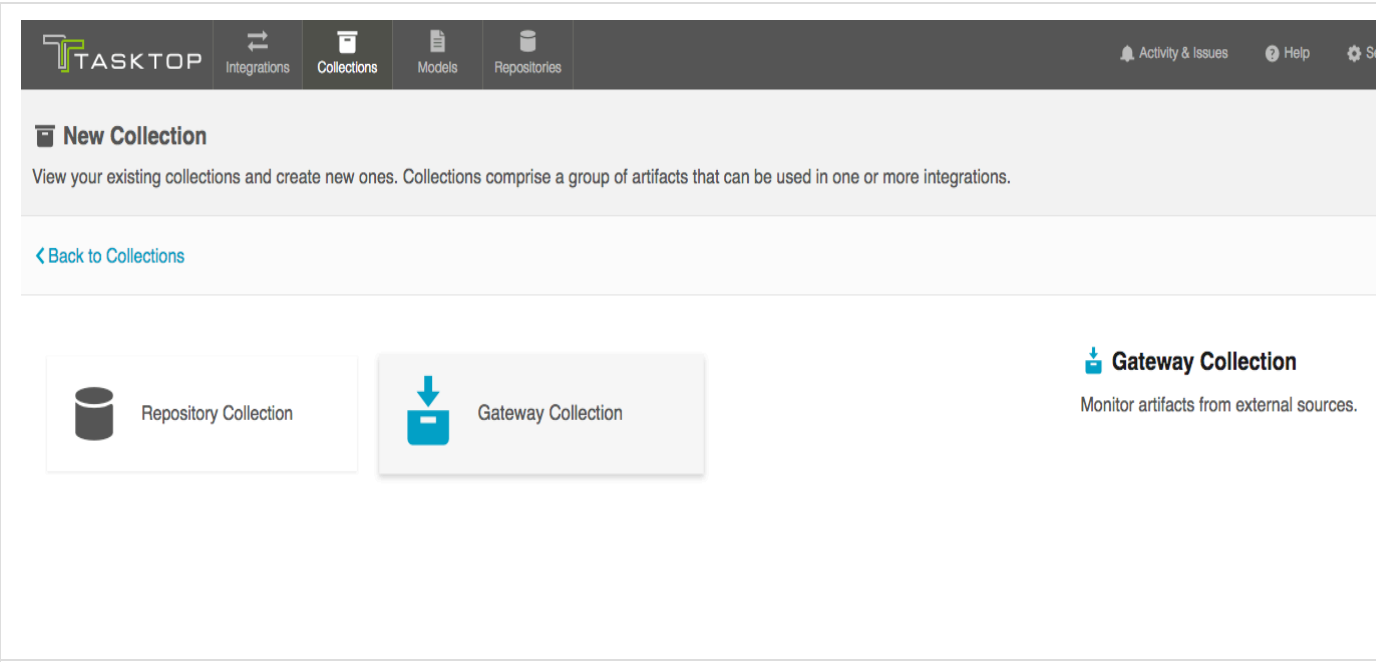
Create a Gateway Collection

Check out the video below to learn how to create a new gateway collection:

Identify Collection Type

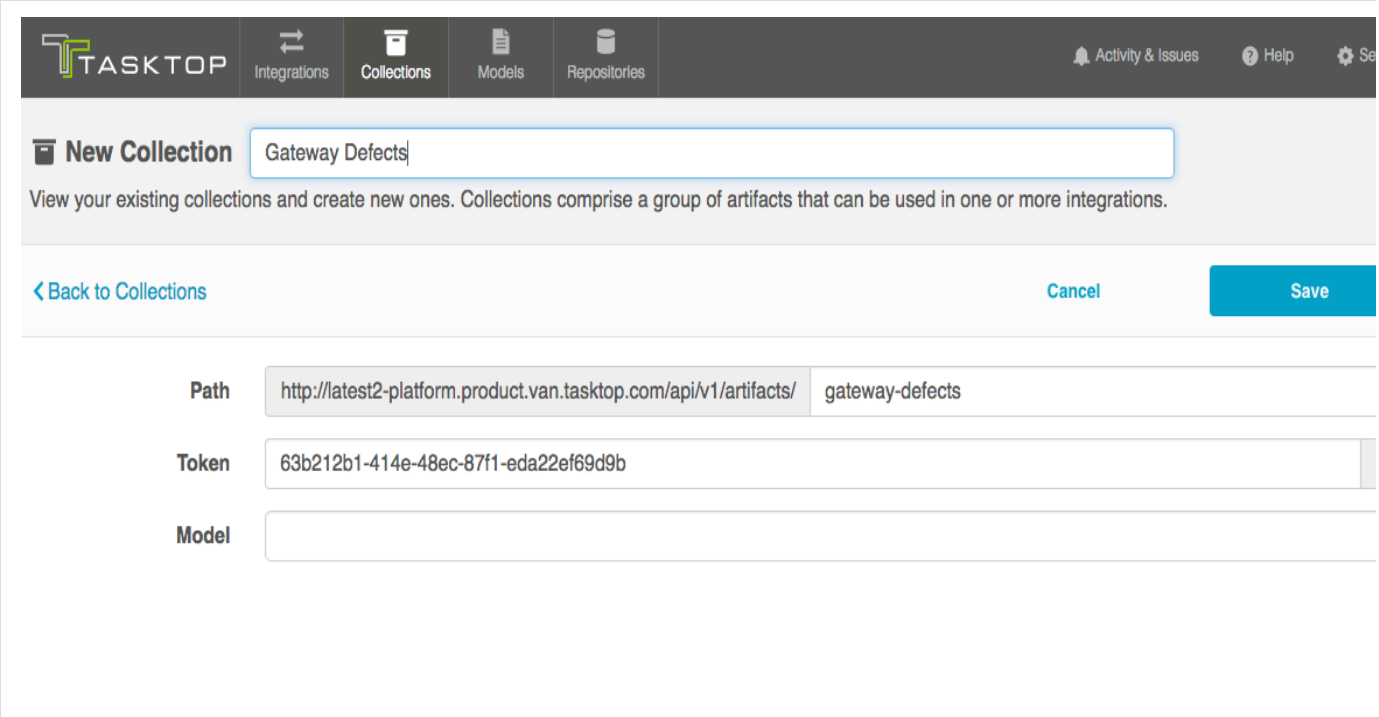
To create a gateway collection, do the following steps after you go to "New Collection":

Select "Gateway Collection" as the collection type.



Name Collection

Enter a name for your collection.



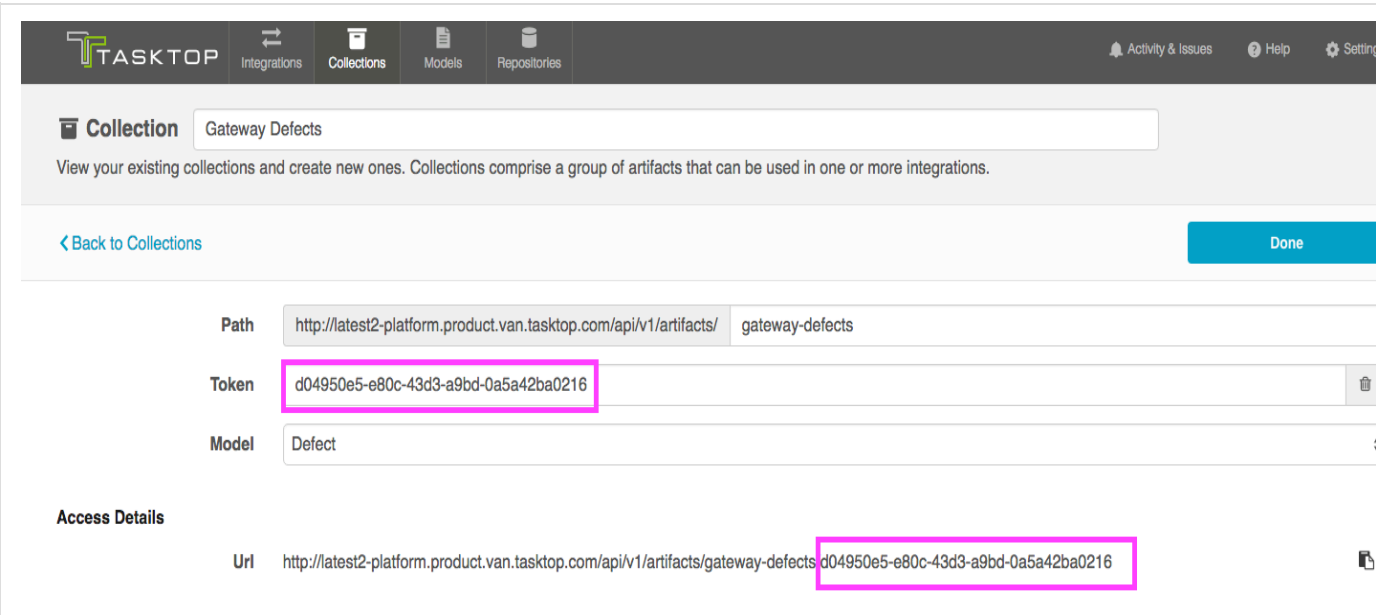
Specify the Path for Collection

Specify the Path for your collection. These characters will form the REST endpoint to which you can send artifacts to Tasktop via this gateway collection.

💡 Upon first creating your Gateway collection, Tasktop will populate Path with the name that you have given to your collection. You can change this if desired.

Secure Your Gateway Collection

To secure your gateway collection, Tasktop automatically appends a token (a universally unique identifier) t o the path of a gateway collection. This token will be incorporated into your gateway URL and help ensure that only users that know the full path with its token can access your gateway collection.

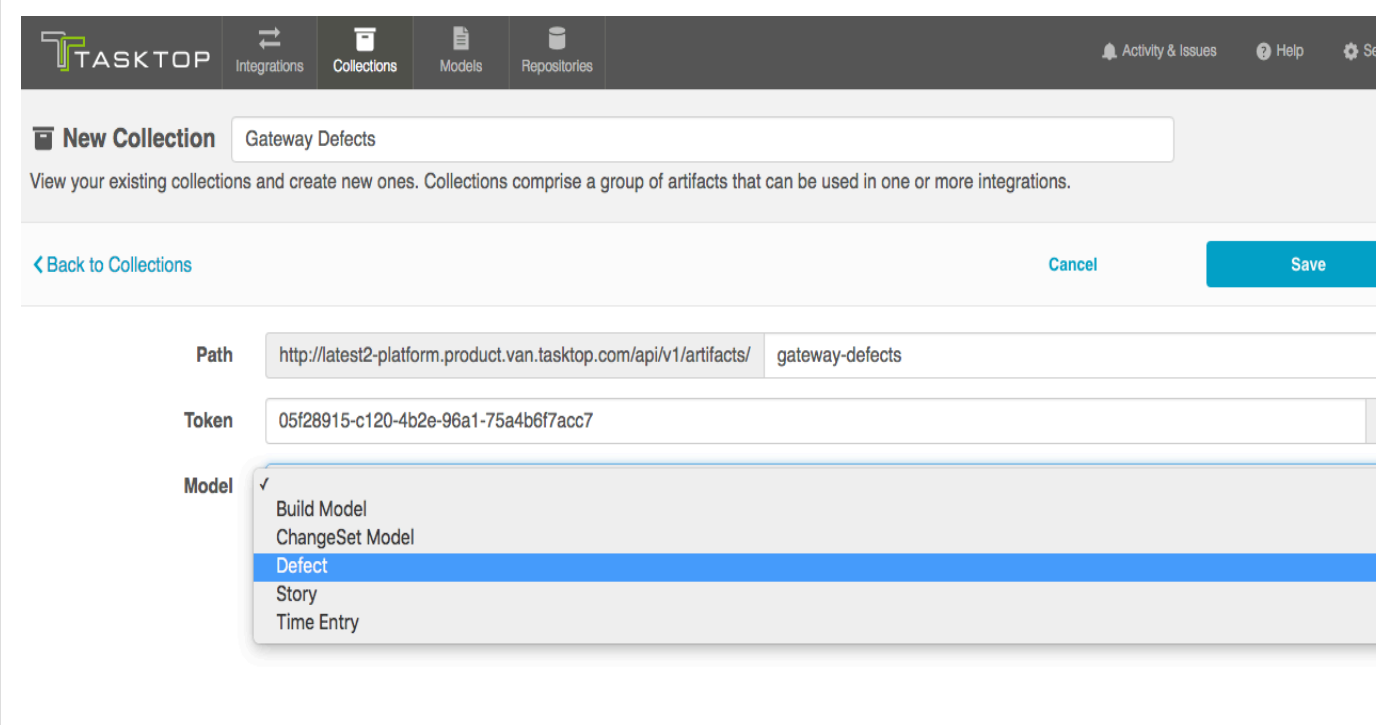


You can remove the token by clicking the trash can icon to the right, and refresh it by hitting the magic

wand icon that appears in its place. Once refreshed, click 'save,' and the URL will be updated.

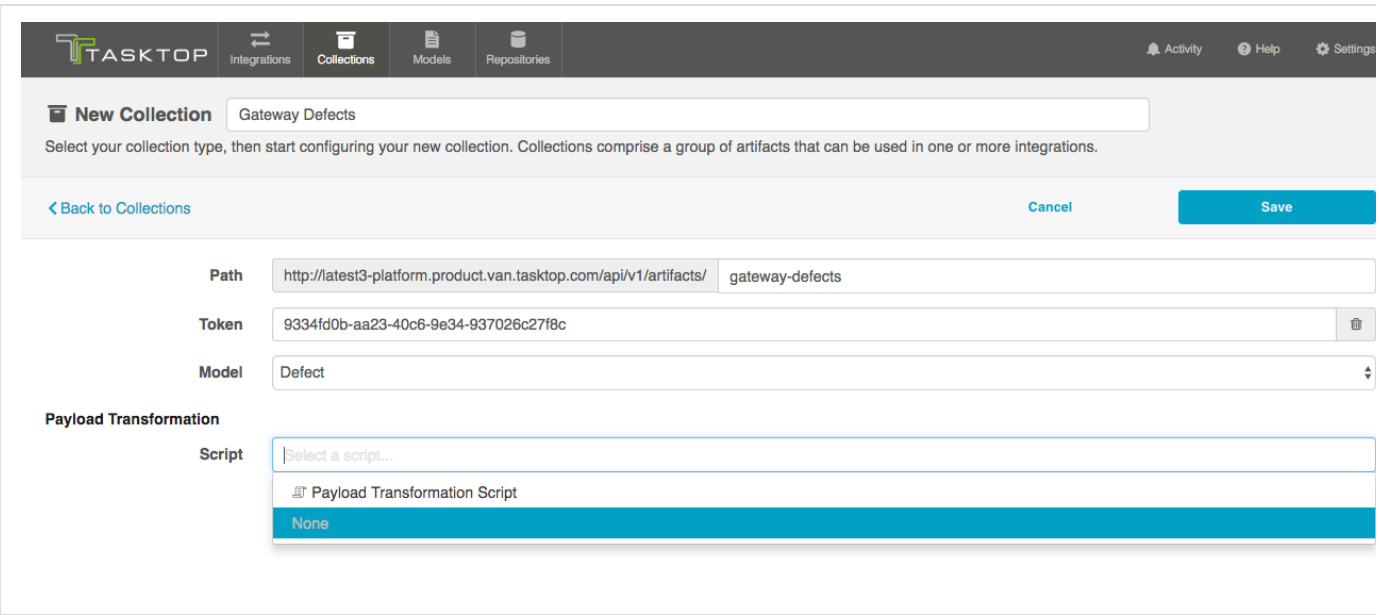
Identify the Collection Model

Select the Model on which you'd like to base this collection.



Optional: Apply a Script

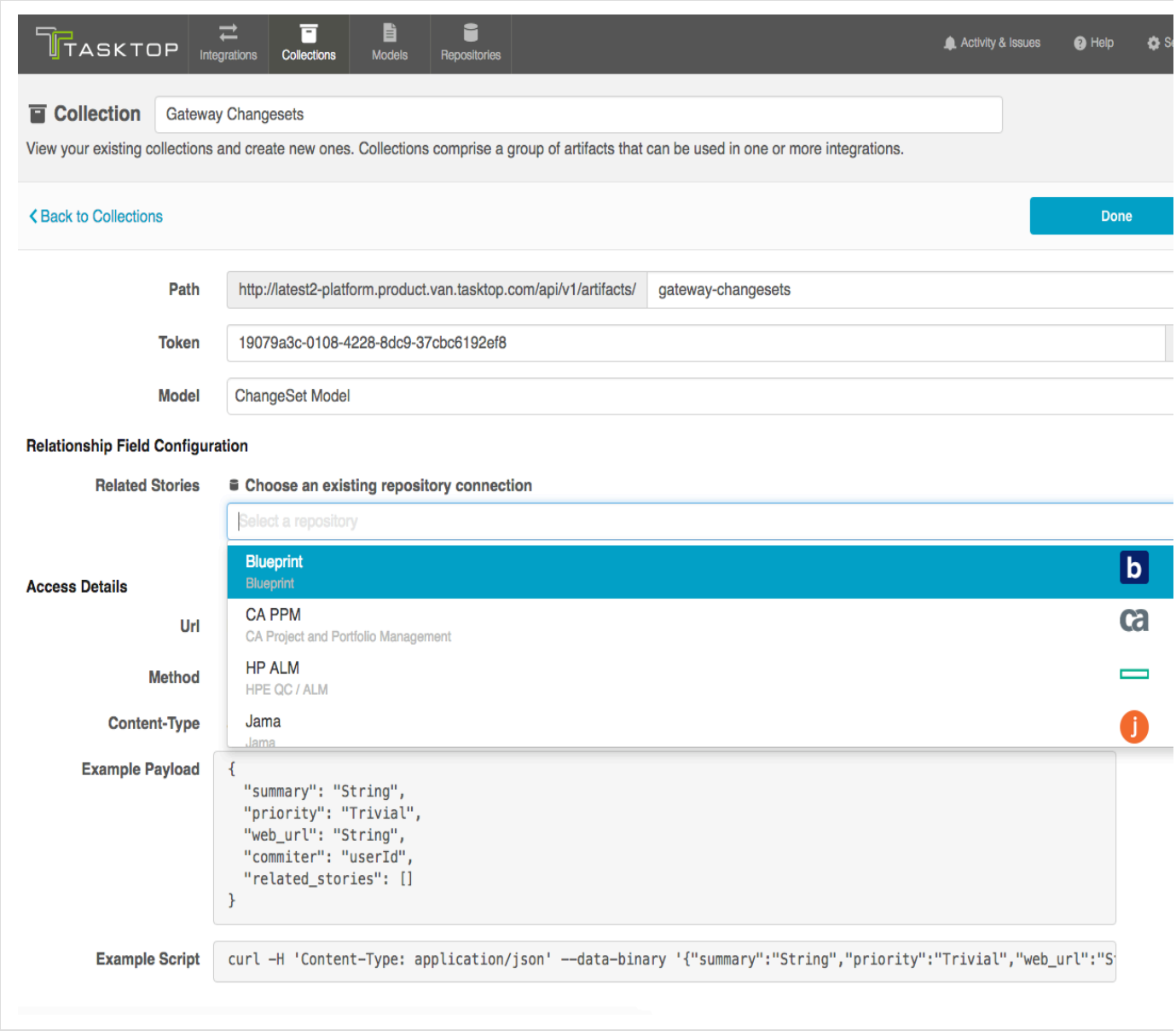
If you have configured a payload transformation script for your Gateway collection on the Settings screen, you can select it here.



Optional: Identify Target Repository for Model Relationship(s) Field(s)

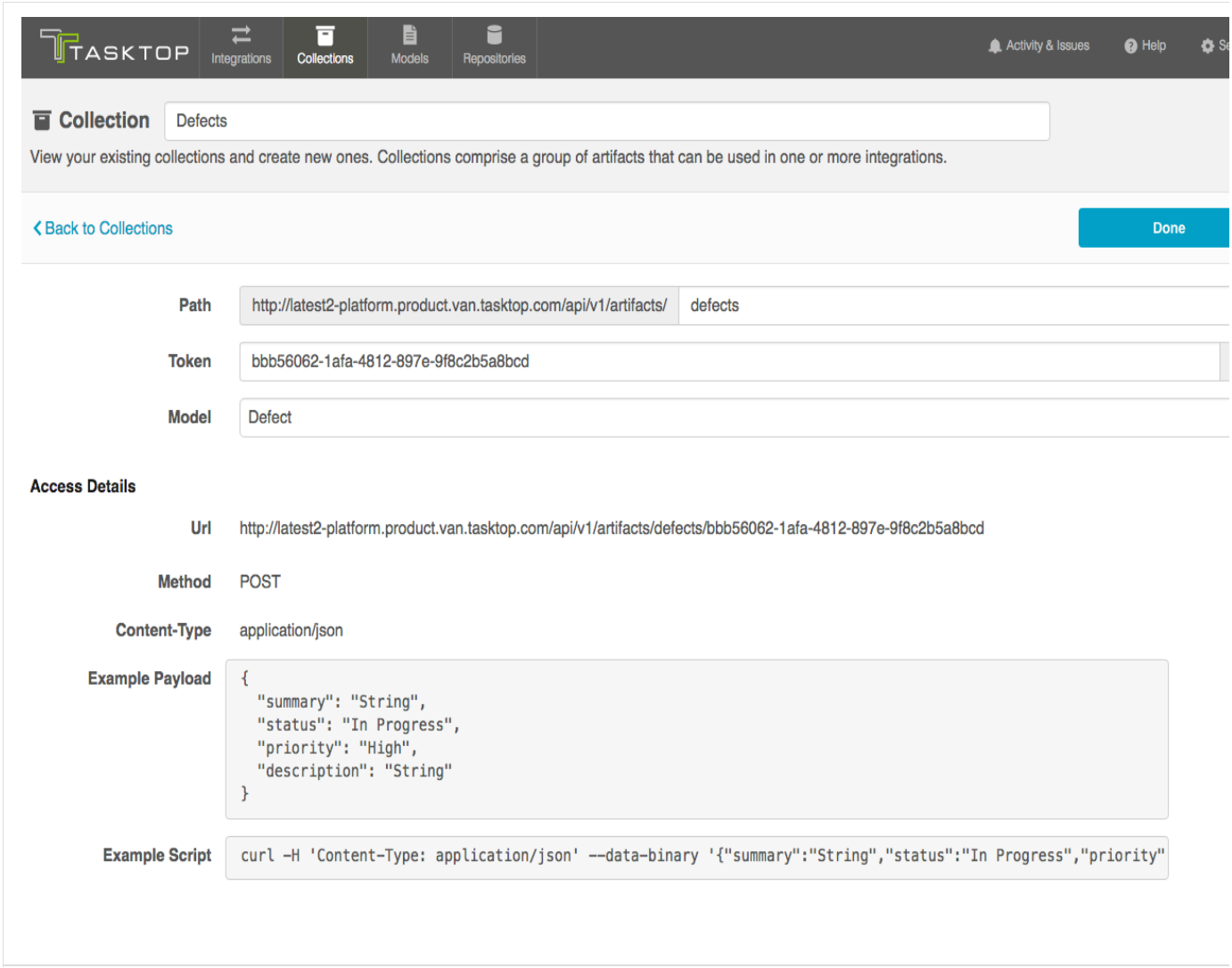
If you have any relationship(s) fields in your model, you'll need to identify a target repository for each. This will ensure that enough information is being sent in via the Gateway to uniquely locate the artifact you'd

like to relate to.



Observe Access Details for Collection

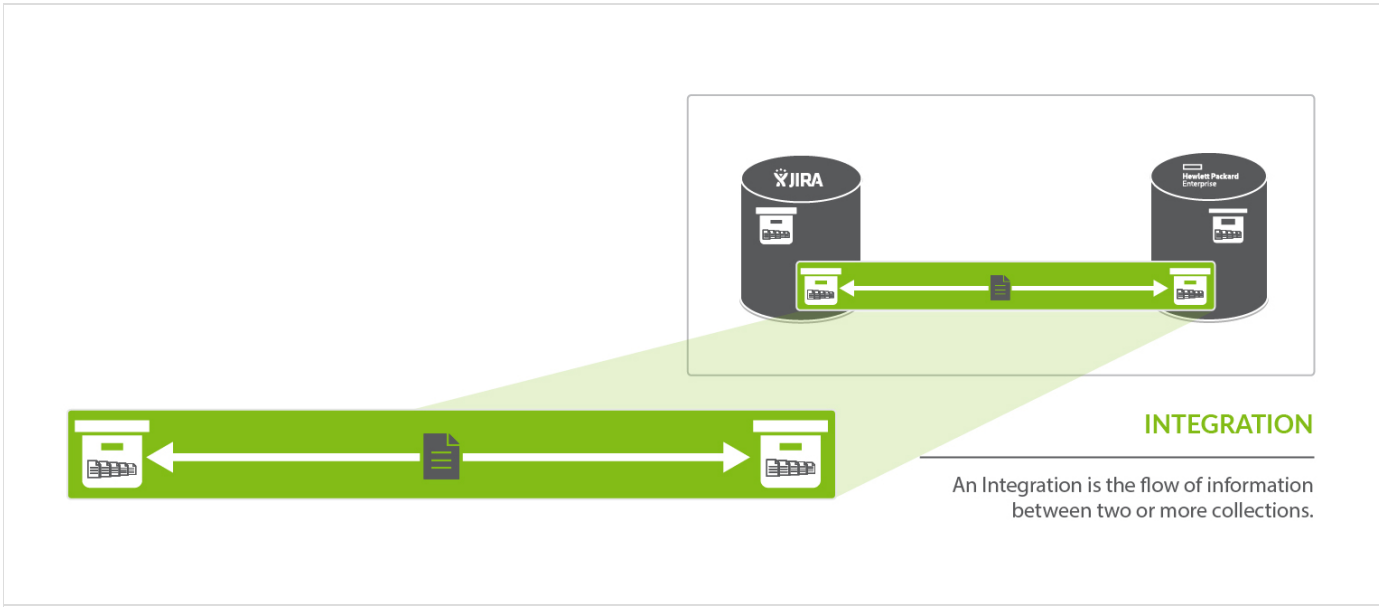
Observe the access details given for this gateway collection.



Step 4: Configure your Integration

Tasktop: Apex Release (17.1)

What is an Integration?



An *integration* is quite simply the flow of information between two or more collections. When you configure your integration, you can customize the field flow, artifact routing, artifact filtering, as well as enable or disable comment flow or attachment flow.

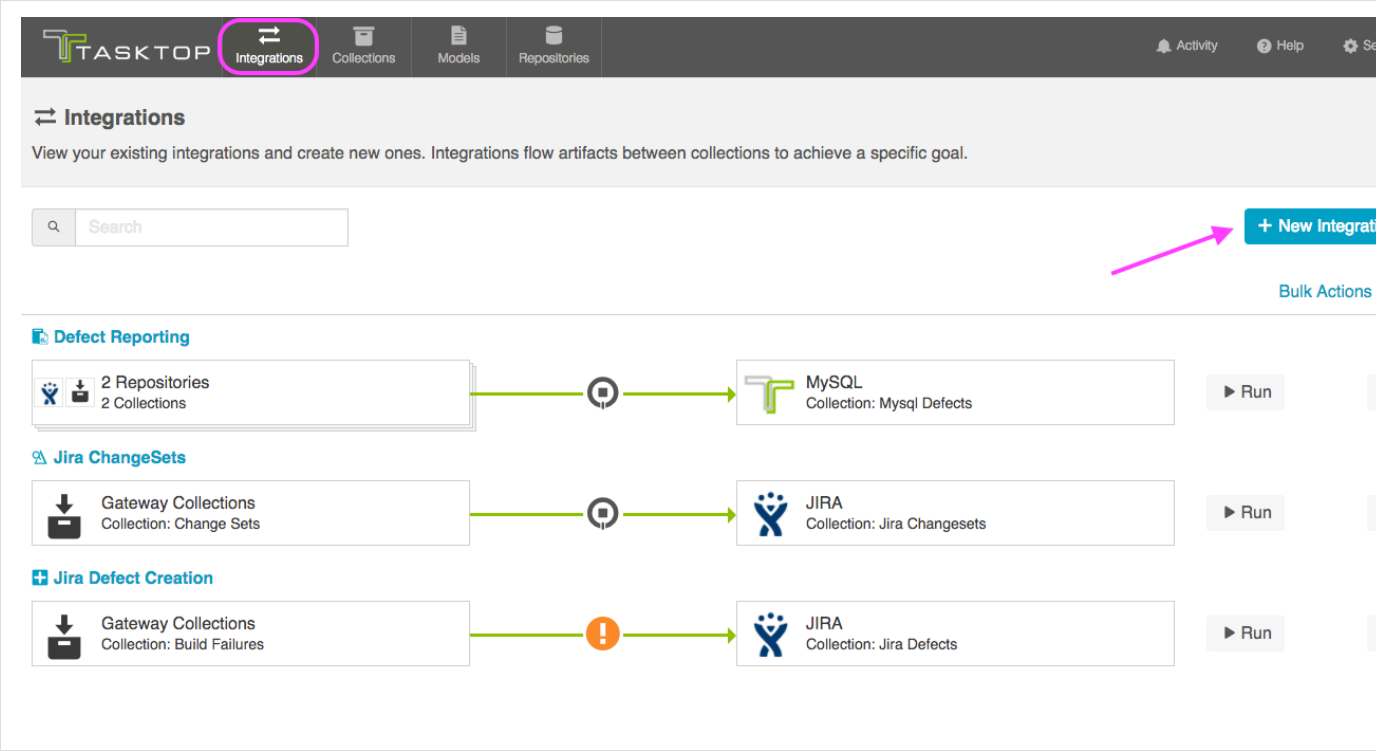
Step 4: Configure your Integration

Getting Started

Now that you have all of your base components set up, you can configure integrations to connect the artifacts in your collections.

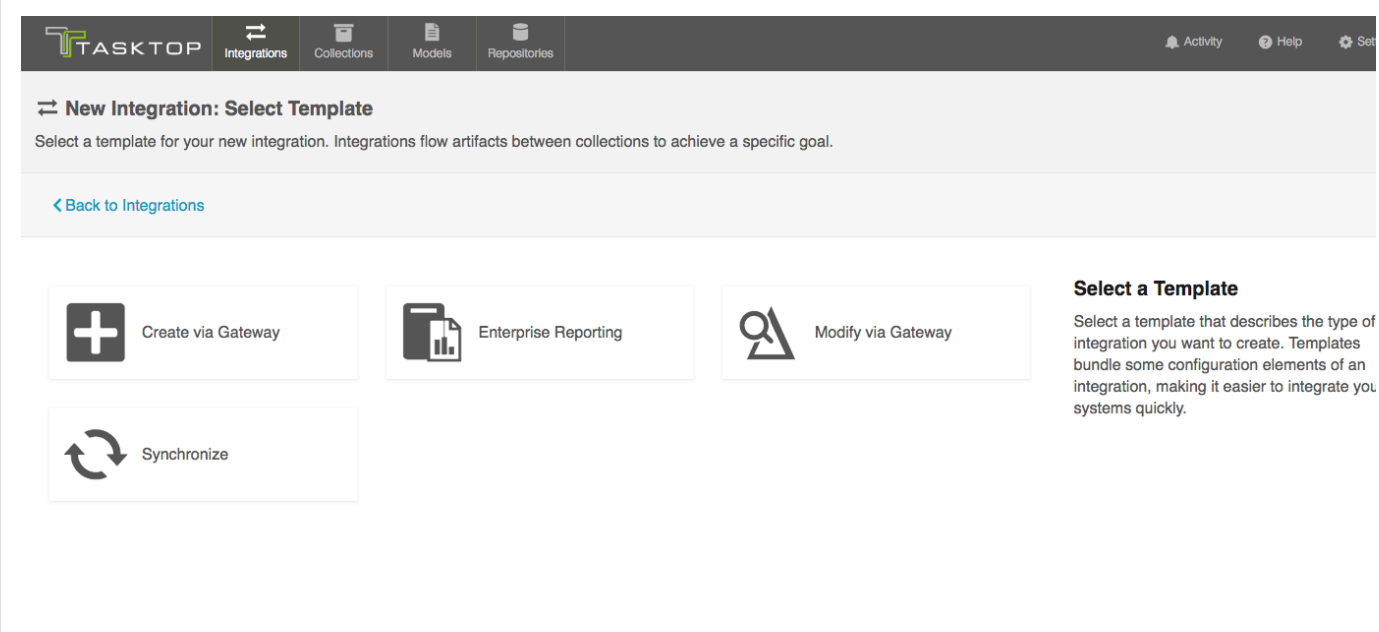
⚠️ Note that this section outlines the general components required to configure an integration. Please consult the Integration Template pages to learn more about the template-specific settings (such as comment and attachment flow).

To configure your integration, select 'Integrations' at the top of the screen, then click 'New Integration.'

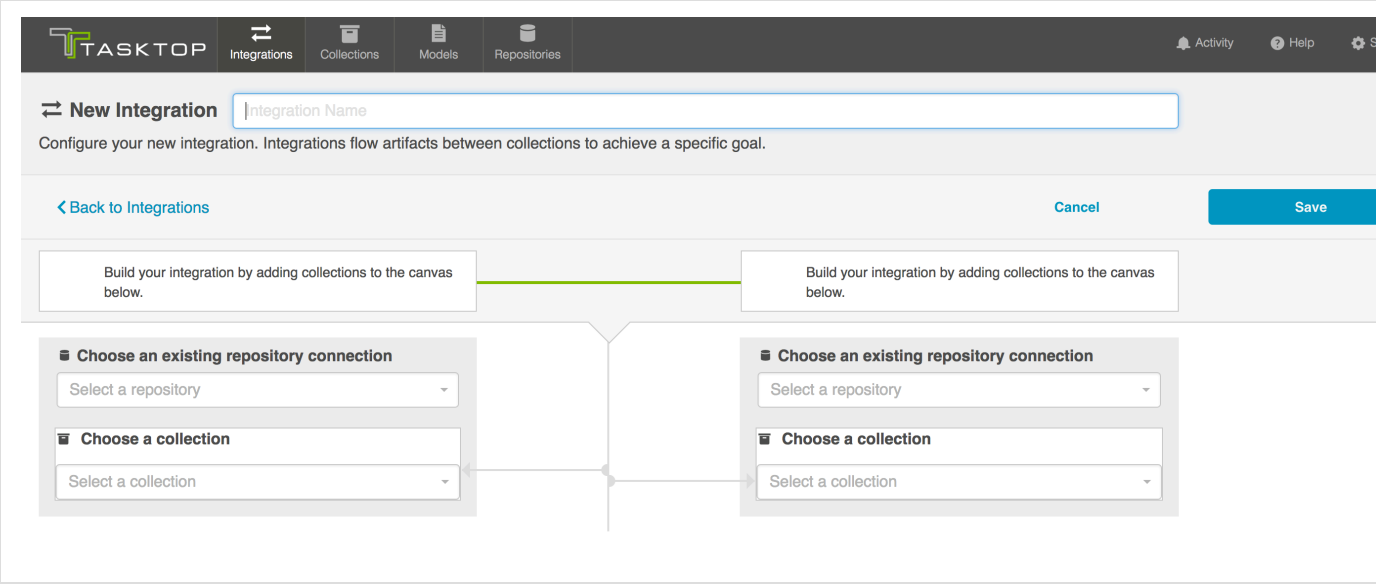


Select your desired integration template from the options available.

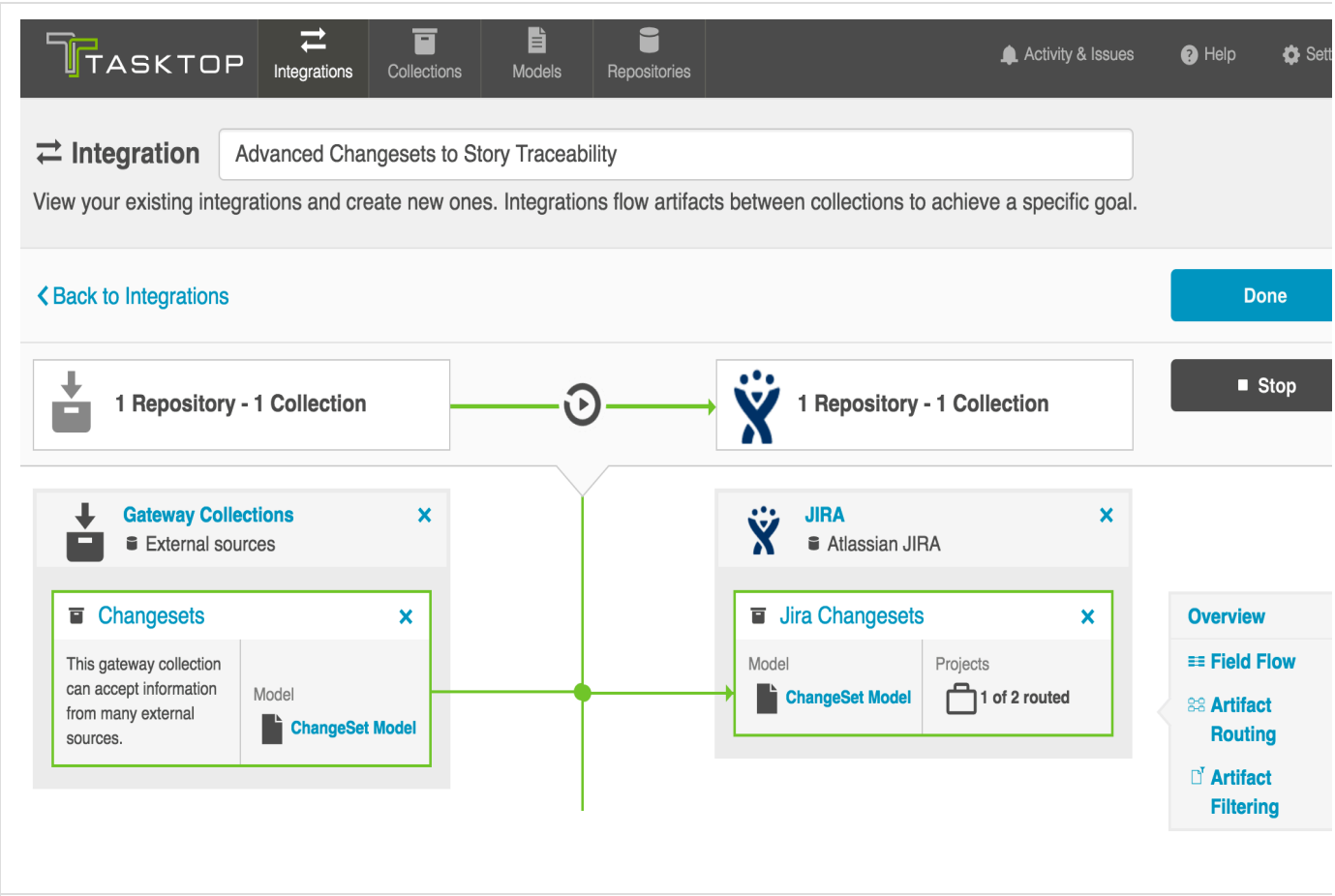
Depending on the edition of Tasktop you are utilizing, you may not have all options available.



This will bring you to the New Integration Screen:

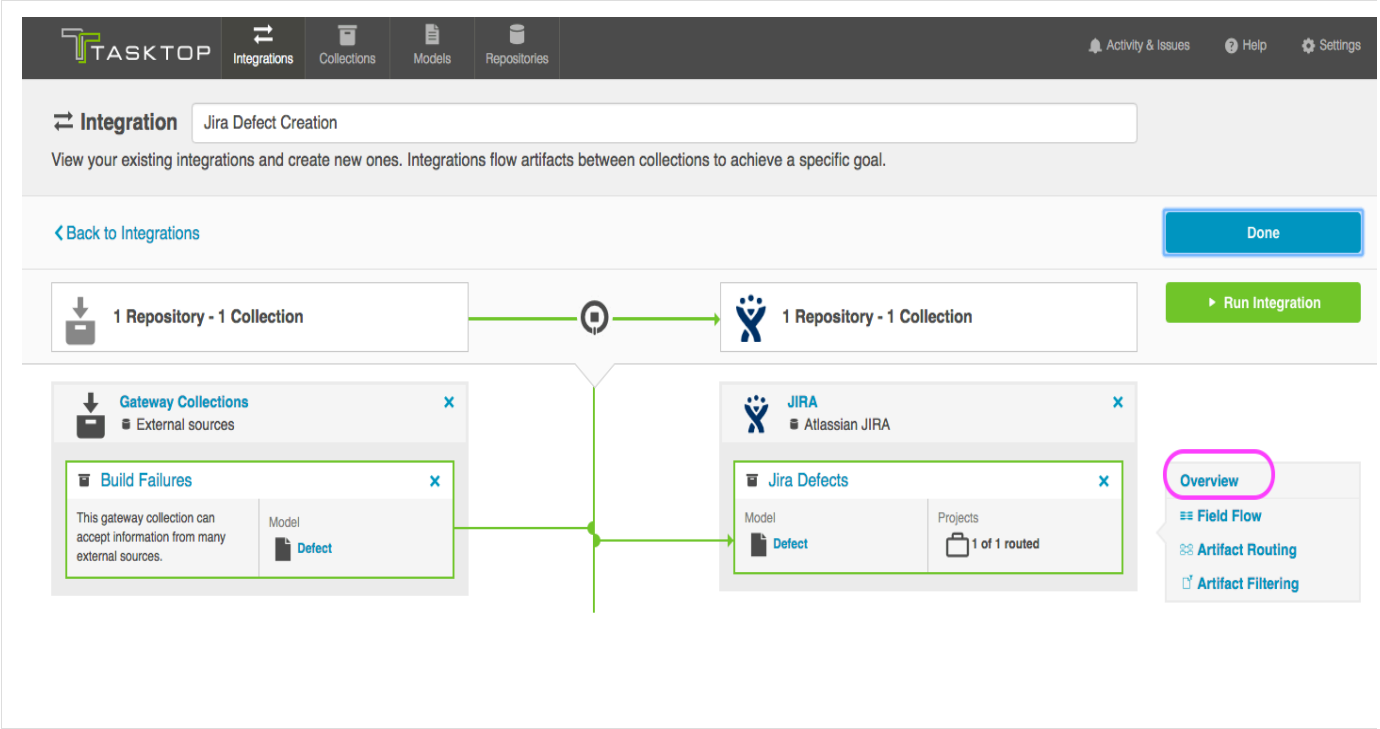


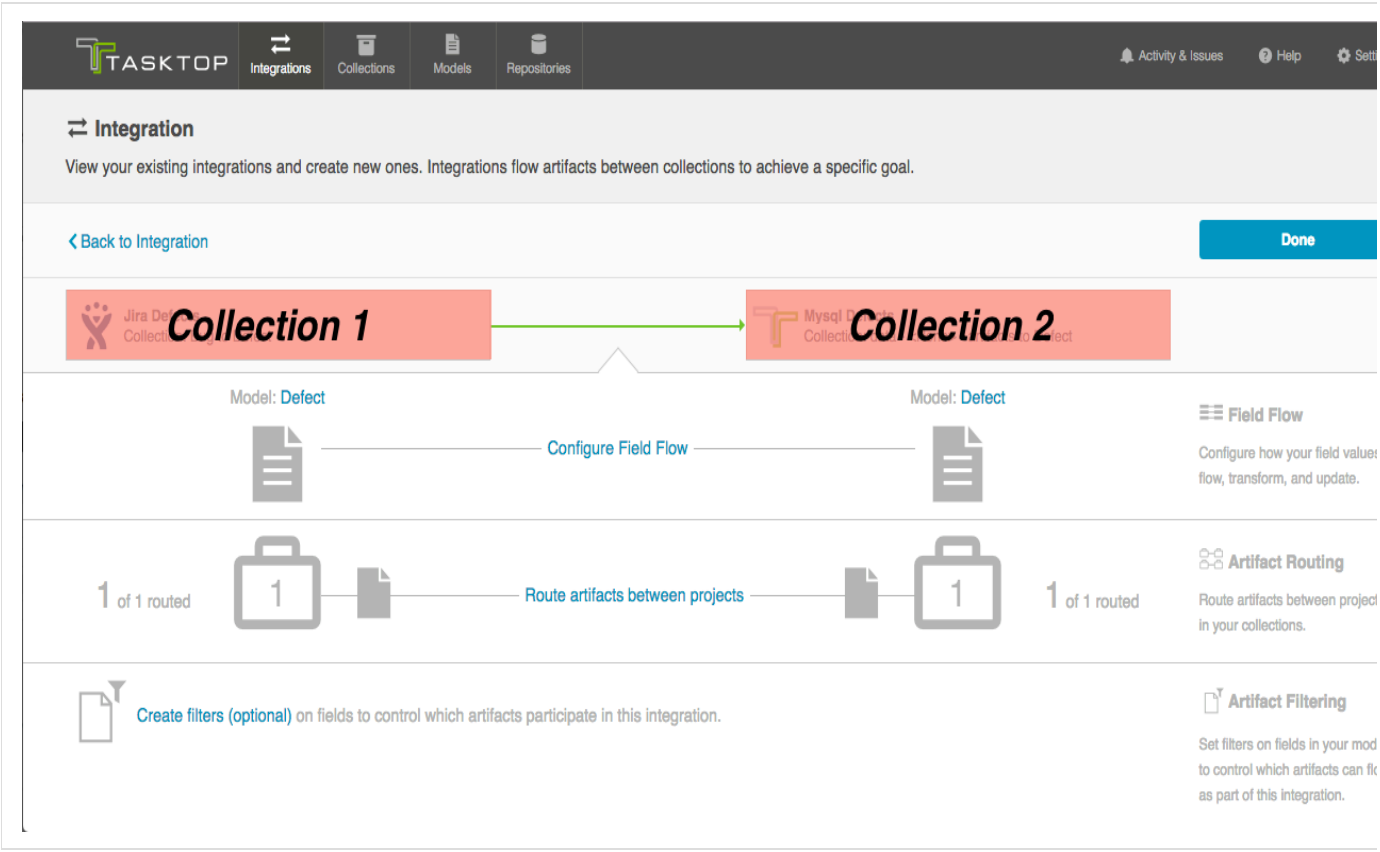
Name your integration and select your repositories and collections:



While each template might have some special steps and affordances (which are detailed in the help section for a given integration template), the general configuration components of an integration are described below.

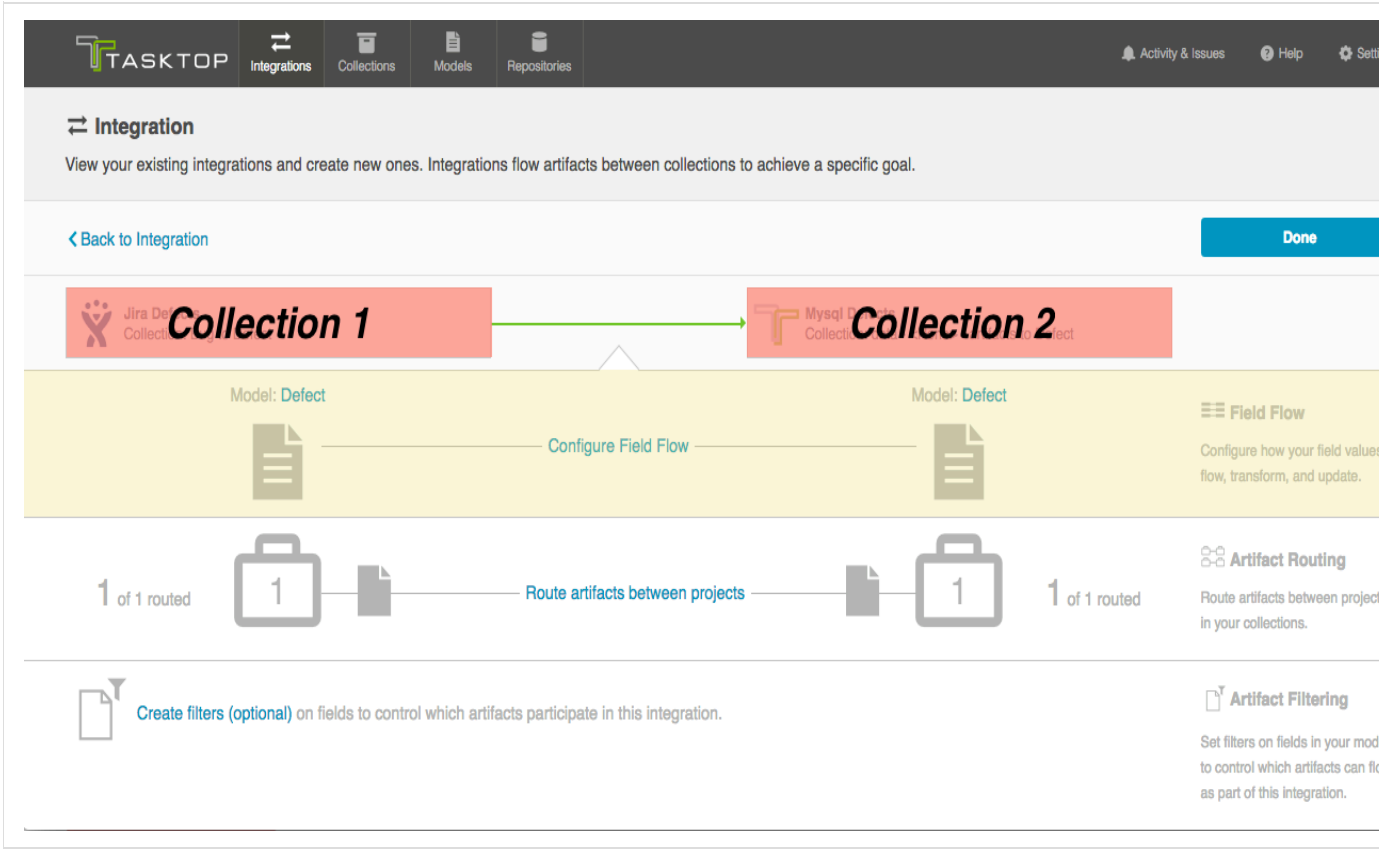
You can click the 'Overview' link on the right side of the Integration Page to get to the main display page (shown in the second screen shot).

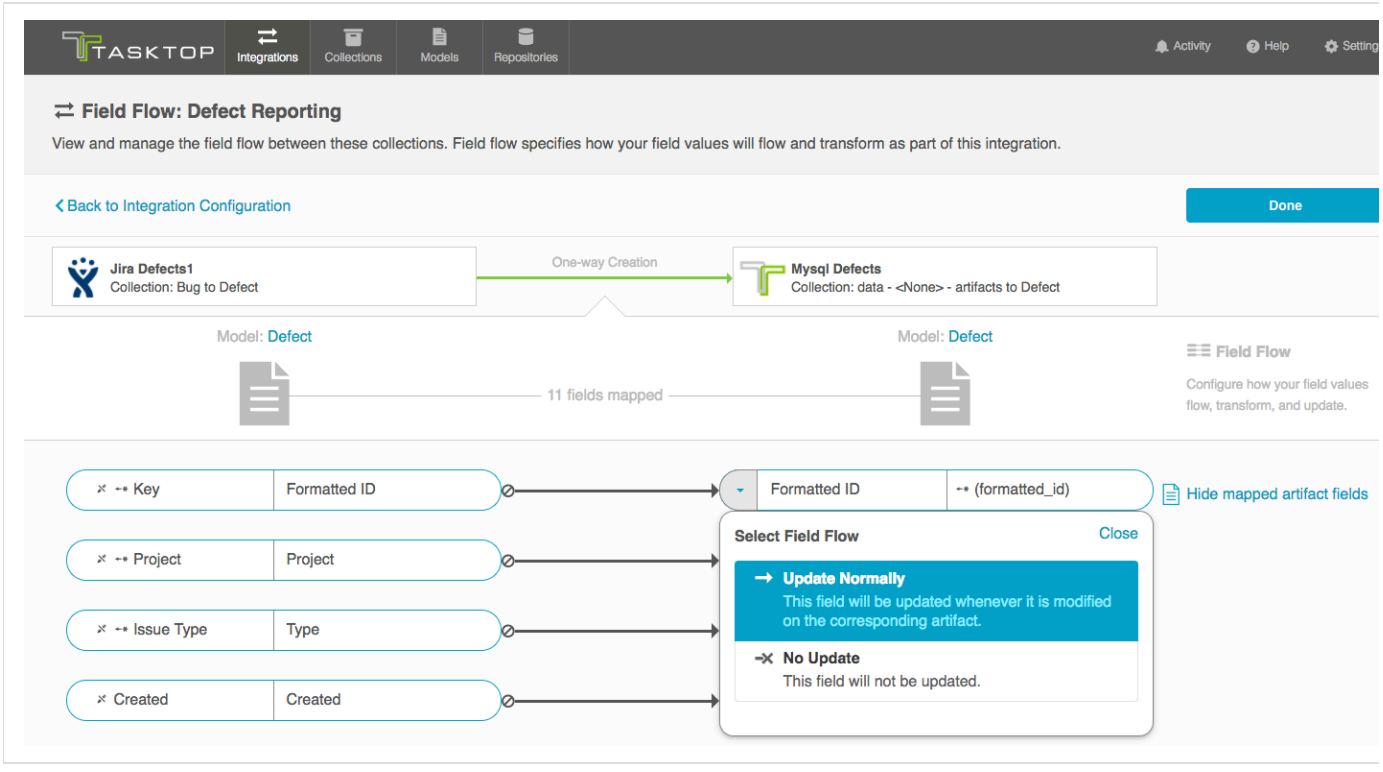




Field Flow

The field flow configured for a given integration specifies which fields should flow in that integration. For Gateway and Data integrations, you can choose to flow a given field (Update Normally) or to not flow a given field (No Update). For Synchronize Integrations, you can also specify the field update frequency (Update Normally, Always Update, Upon Artifact Creation, or No Update).





You can see the names of the mapped artifact fields for each collection on the far left and far right, with the model fields displayed in the middle. To hide the mapped artifact fields, select 'Hide mapped artifact fields' on the right.

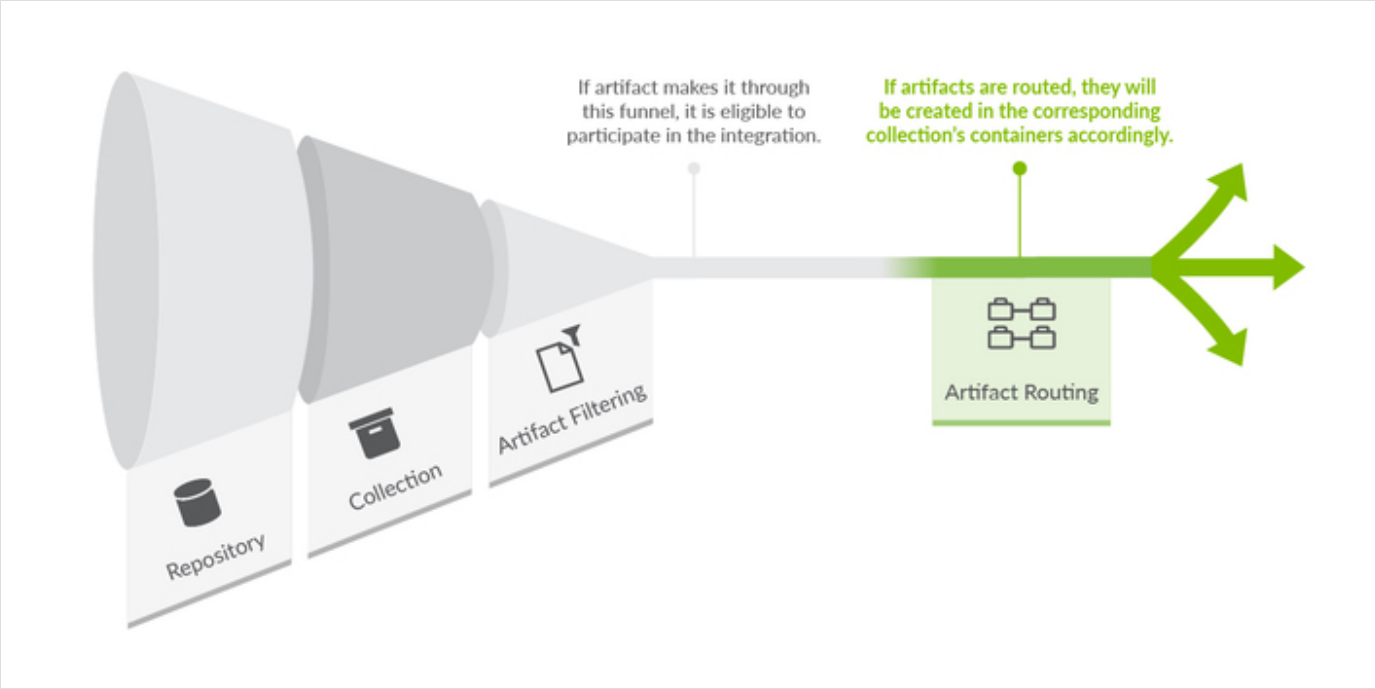
⚠️ Note: The field flow behavior is somewhat unique when using the Enterprise Reporting Template. See that section for more details.

Field Flow Icons

On the Integration Field Flow page, you will see a number of icons, which will help you understand any special properties or requirements for each field. If you hover your mouse over an icon, you will see a pop-up explaining what the icon means. You can also review their meanings in the legend below:

Icon	Meaning
C	A constant value will be sent. Note that: • If the icon is on the side of the collection, this means that a constant value will be sent to your model. This means that any time this collection is integrated with another collection, the *other* collection will receive this constant value for the field in question. • If the icon is on the side of the model, this means a constant value will be sent to your collection. This means that any time this collection is integrated with another collection, that *this* collection will receive this constant value for the field in question.
	Collection field is read-only and cannot receive data
	To create artifacts in your collection, this field must be mapped to your model.
	This is a required field in your model; it must be mapped to your collection.
	This field will not be updated as part of your integration, due to how you have configured it. This field flow configuration can be changed if you'd like.
	This field will not be updated as part of your integration because the mapping would be invalid. You do not have the option of changing this.
	This field will update normally as part of your synchronize integration; this means it will be updated whenever it is modified on the corresponding artifact.
	This field will always update as part of your synchronize integration; this means that it will be updated whenever *any* fields are modified on the corresponding artifact.
	This field will only be updated upon initial artifact creation.

Artifact Routing

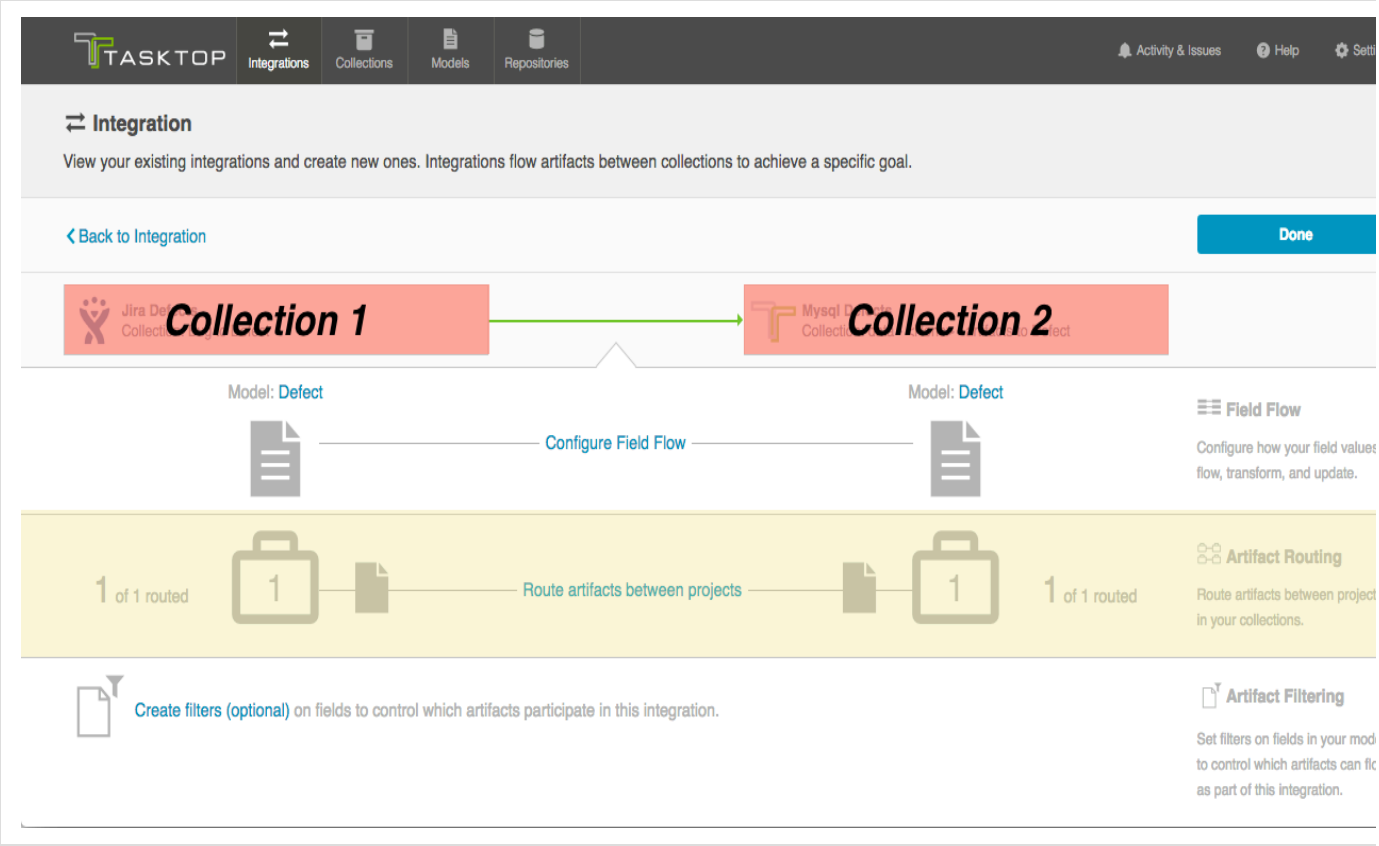


Artifact Routing is needed when artifacts are being created as part of an integration. In addition to knowing the repository in which artifacts should be created, Tasktop also needs to know which container (i.e. project, module, folder, etc) a given artifact should be created in. Specifying the artifact routing does this. If your integration does not entail artifact creation, you will not see or need to configure artifact routing.

💡 Initially, the aritfact routing will determine where an artifact gets created. Over time, if an artifact on either side moves, we will move the artifact to the corresponding container of the new route, if this is allowed in your repository. if you are moving between lower-level containers, such as sets or folders, this is generally possible. However, we will not do so if the move on one side crosses the bounds of the top-level container (generally the high-level container, added at the collection level).

Check out the video below to learn how to configure Artifact Routing:

To configure Artifact Routing, select 'Route artifacts between projects' on the Integration Overview screen, or 'Artifact Routing' on the right pane of the Integration Configuraiton screen.

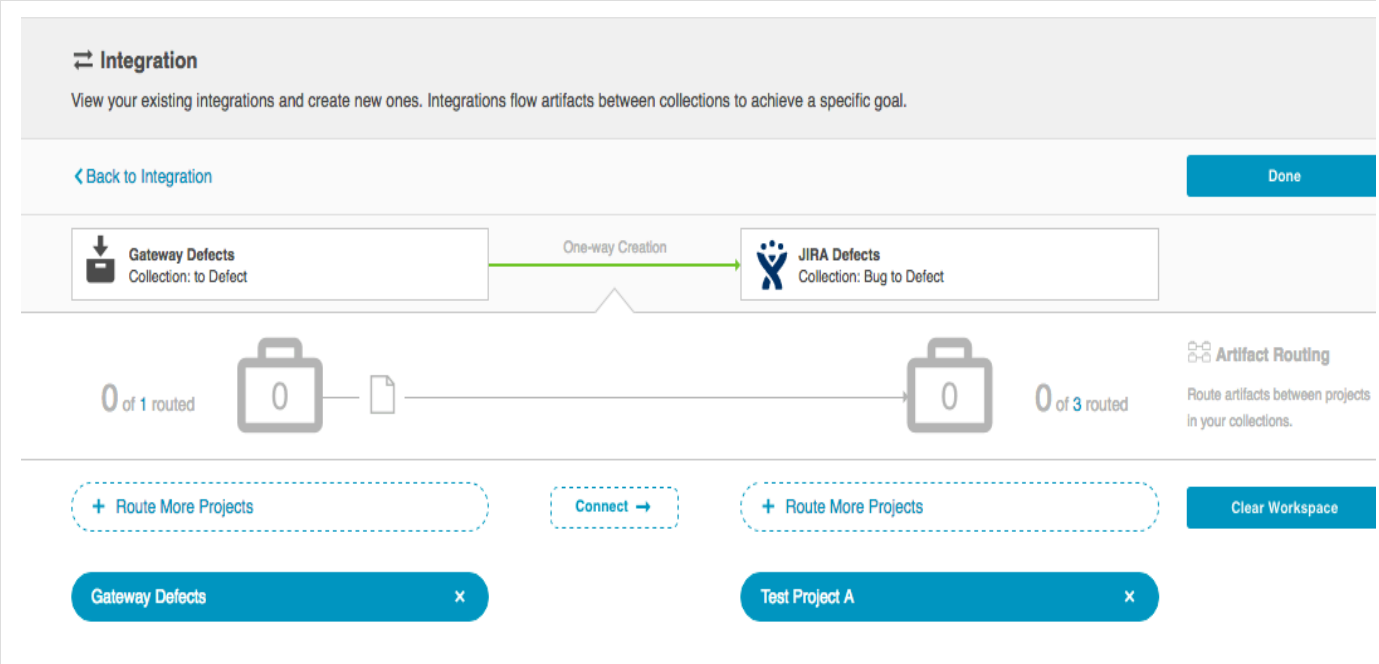


Static Artifact Routing

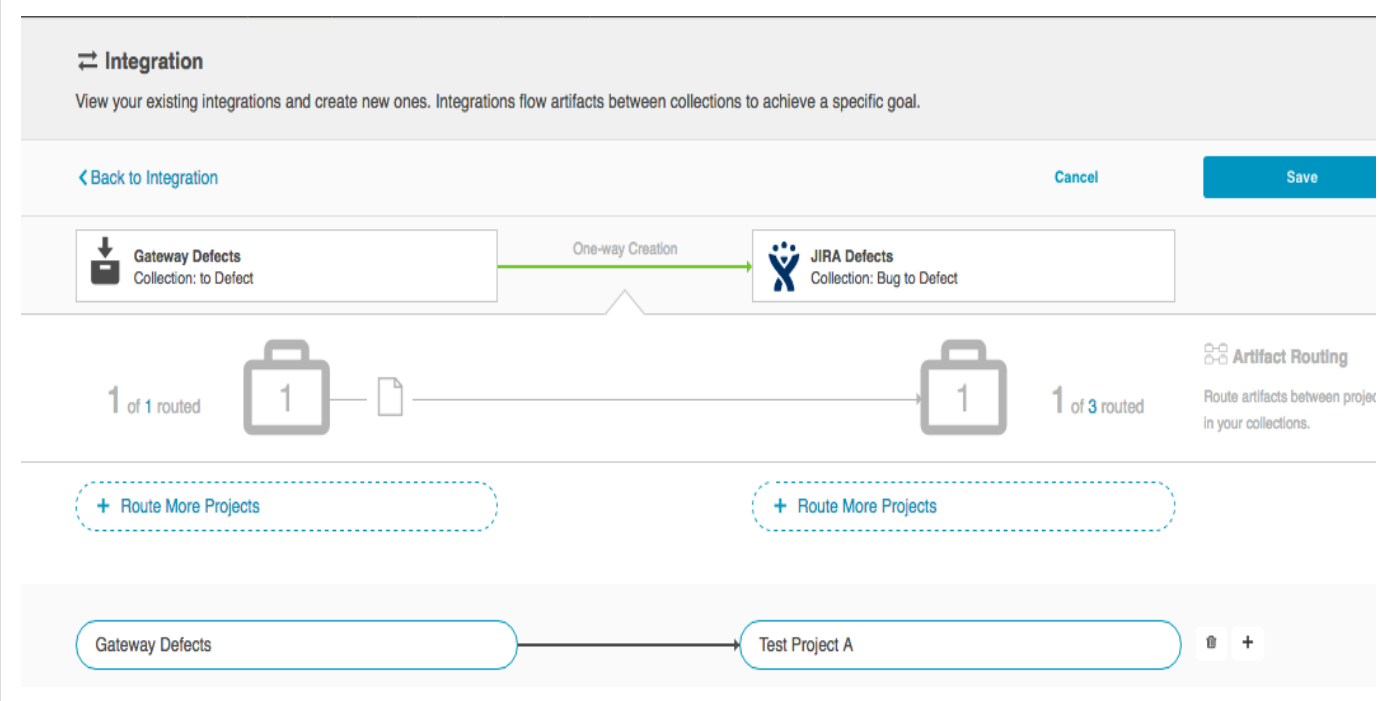
In some cases, the project an artifact is in on one side can sufficiently determine which project an artifact should be created in in the corresponding collection. In these instances, you can configure what is known as 'static artifact routing' (also known as 'explicit artifact routing').

To configure a static artifact route, use the "Route More Projects" buttons to add projects from your collections to your working space and connect them using the "Connect" button.

Note: Static artifact routes can have one or more source projects, but only a single target project.



In the example shown below, artifacts from Gateway Defects will be created in Test Project A in JIRA.



Conditional Artifact Routing

⚠ *Note: Conditional Artifact Routing is available only for Enterprise and Ultimate Editions. Conditional Artifact Routing is currently available for Pro users for a limited time, in preview mode.*

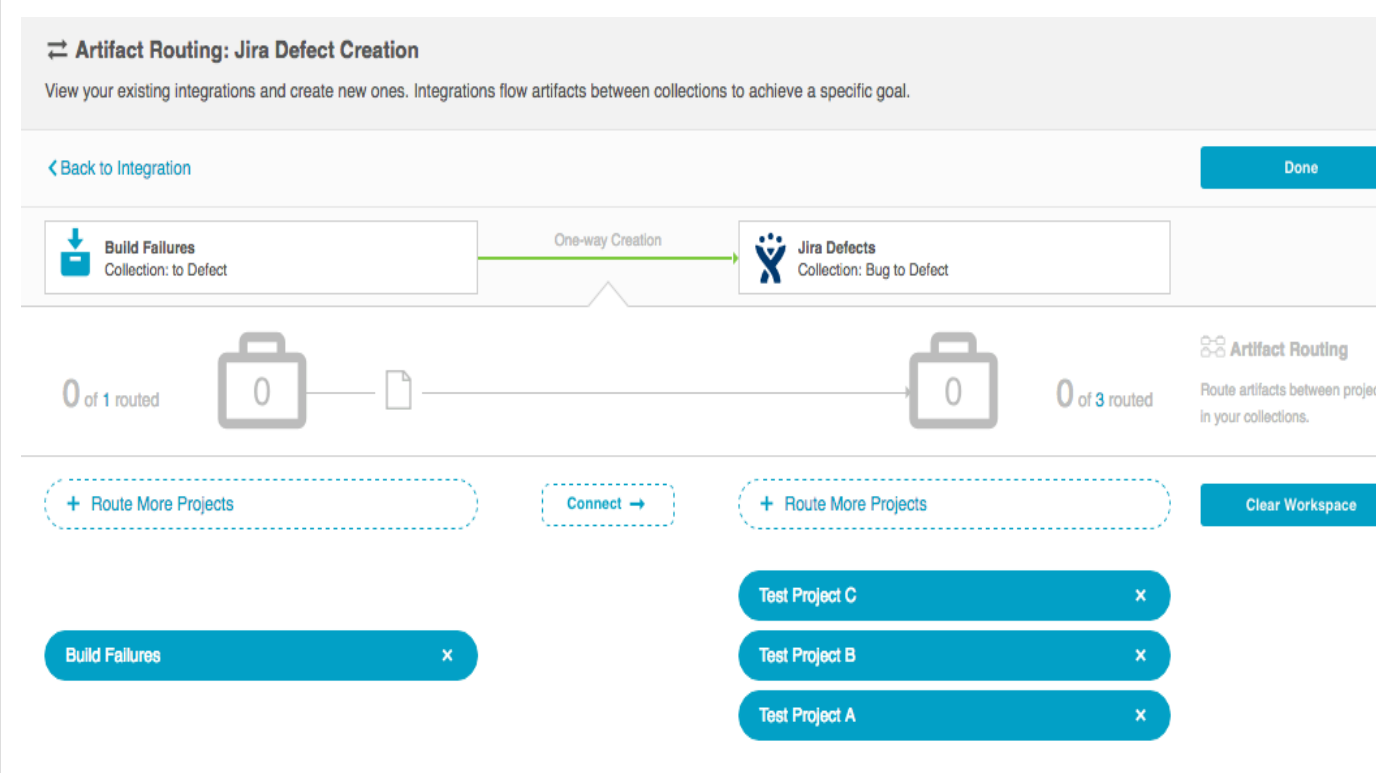
In other cases, the project an artifact is in on one side does not provide enough information to determine which project an artifact should be created in in the corresponding collection. Often times, in fact, some unique characteristic of an artifact is the factor that should be used to determine where an artifact should be created on the other side.

In these instances, artifacts are routed between projects across collections conditionally. Conditional

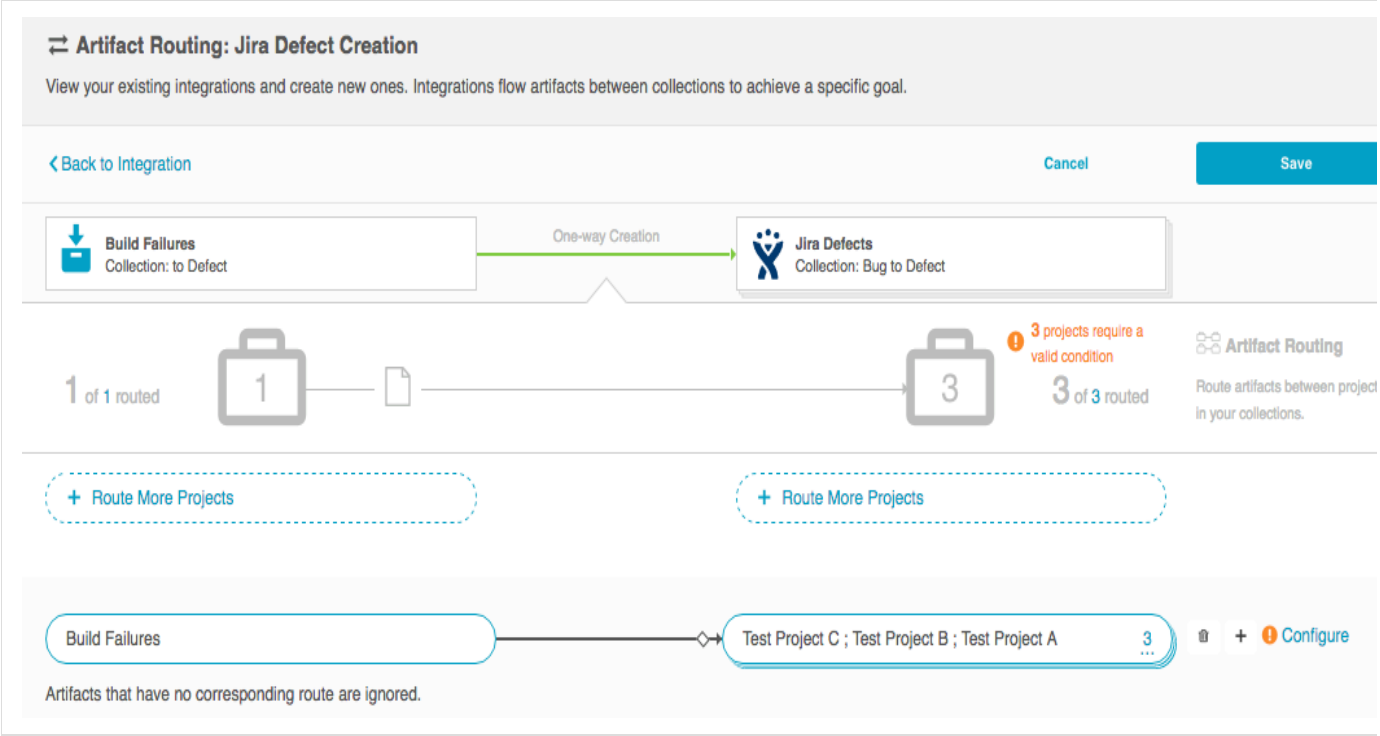
artifact routing (also known as 'dynamic artifact routing') can be used to inspect a certain field of an artifact and, depending on its value for that field, to route that artifact to be created in the appropriate project in the other collection.

To create a conditional artifact route, use the "Route More Projects" buttons to add projects from your collections to your working space and connect them using the "Connect" button.

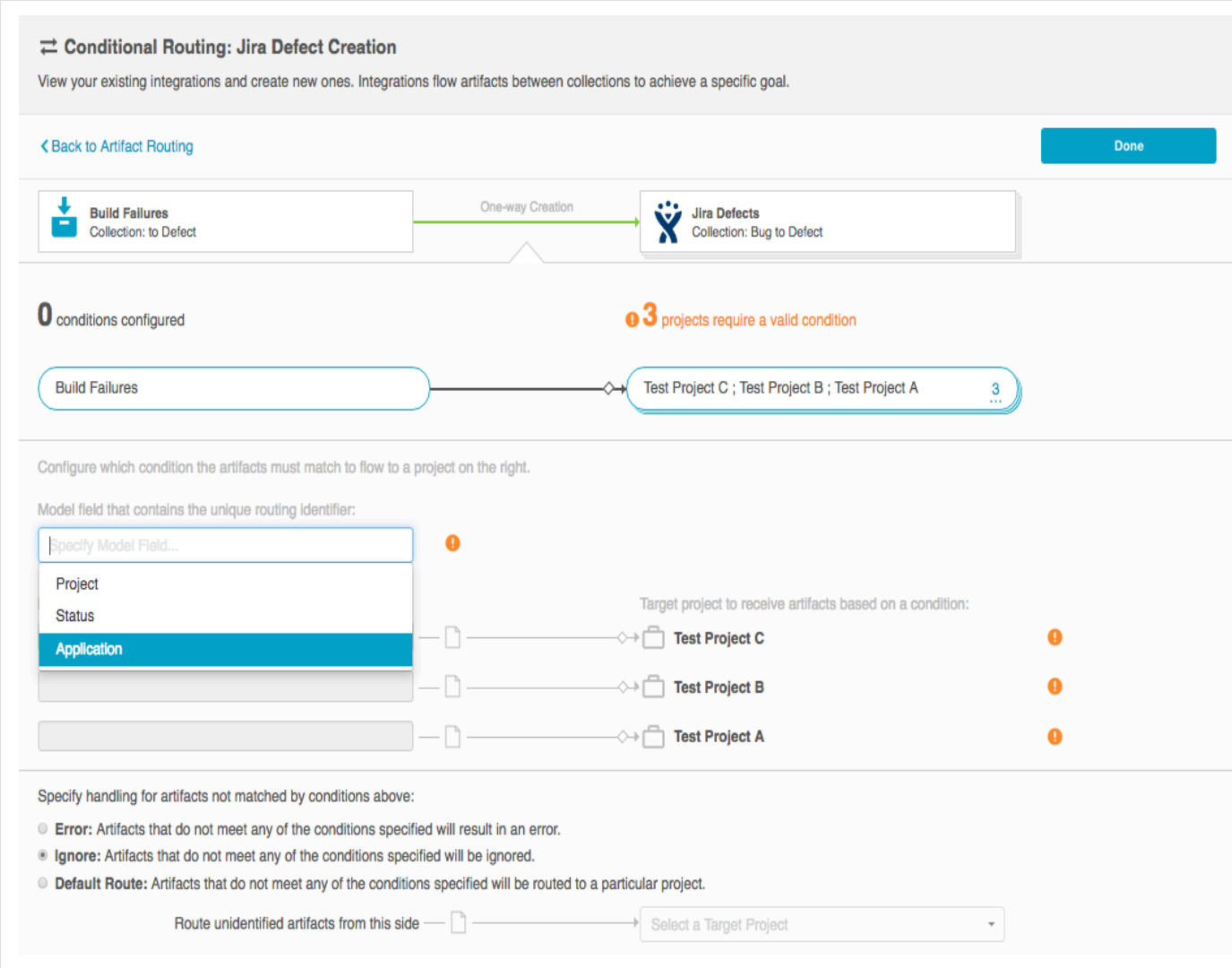
Note: Conditional artifact routes can have one or more source projects, and always have multiple target projects.



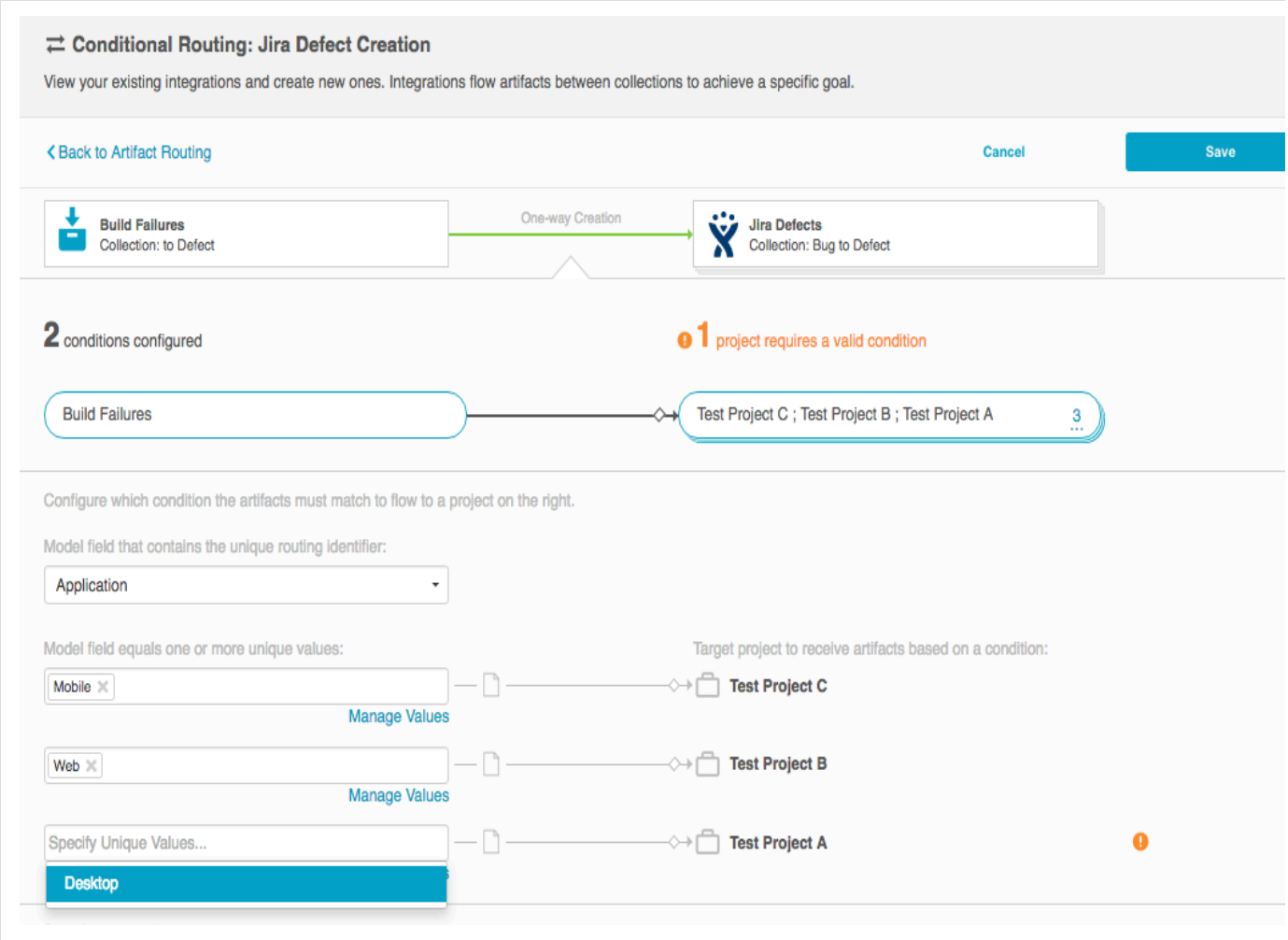
Notice that after you've created your conditional artifact routing group, you'll be prompted to set the conditions that will define that route.



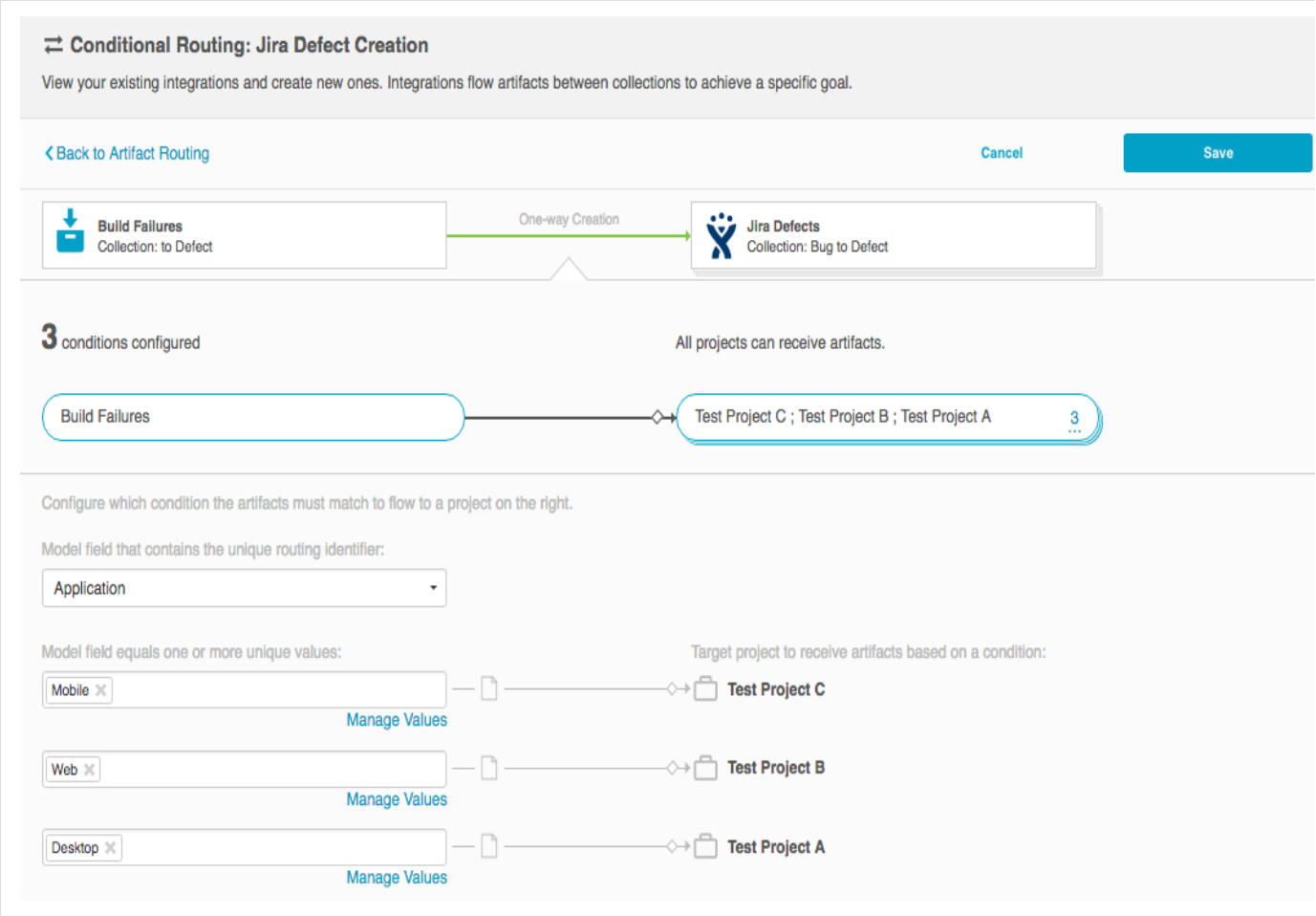
Upon clicking "Configure", you'll see the conditional artifact routing screen. Here you'll start by selecting the model field of an artifact that has the unique routing identifier. In the example below, the field "Application" contains the unique values that should determine the project an artifact should be created in in JIRA.



After you select the model field, you can identify one or more value to correspond to each target project. You can also use the 'Manage Values' link to select from a list of values.



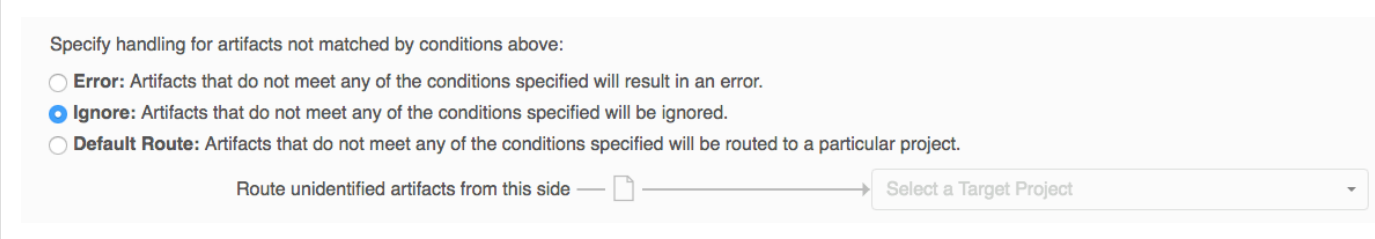
Once you've done this, you'll see your full conditional artifact routing group:



In this example:

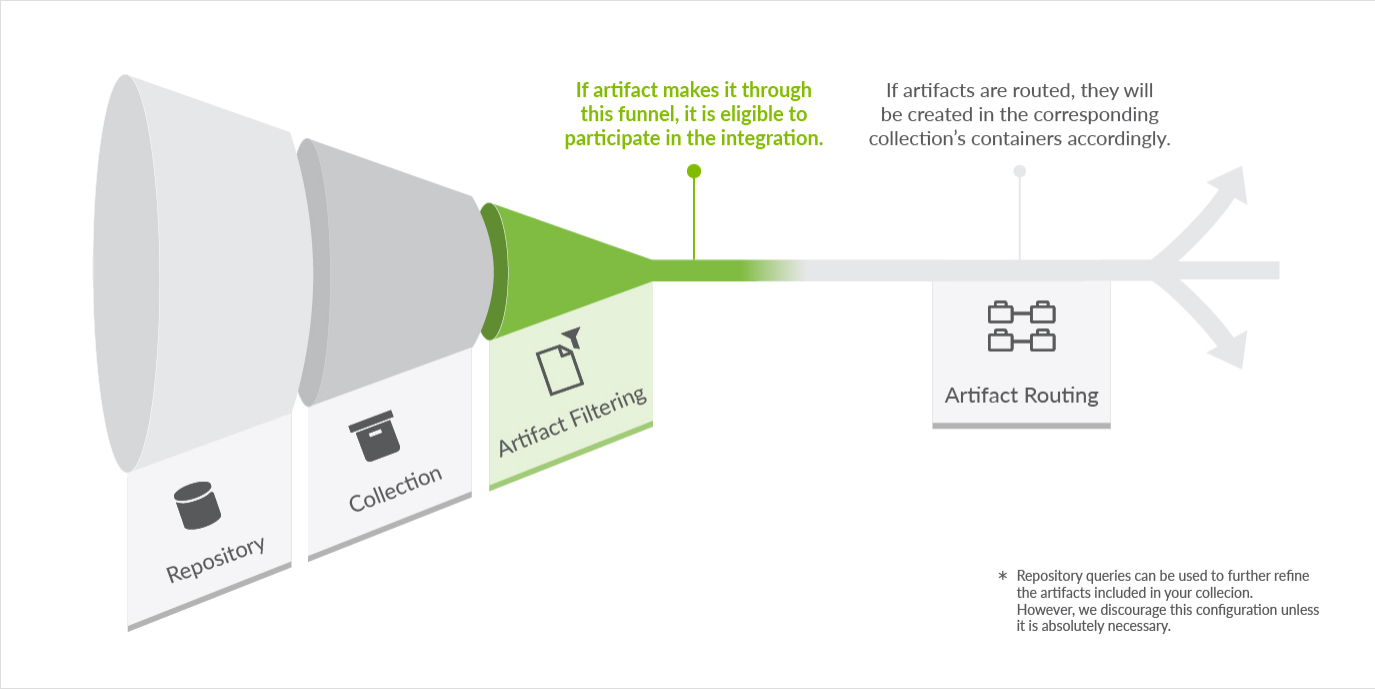
- defects from Gateway Defects with a value of "Mobile" in the Application field will cause the creation of a corresponding defect in Test Project C in JIRA,
- defects from Gateway Defects with a value of "Web" in the Application field will cause the creation of a corresponding defect in Test Project B in JIRA, and
- defects from Gateway Defects with a value of "Desktop" in the Application field will cause the creation of a corresponding defect in Test Project A in JIRA.

You can specify how you'd like to handle defects from Gateway Defects that do not meet any of the conditions specified (for instance, its value for "Application" is "Other",) by selecting one of the options provided at the bottom of the screen:



Artifact Filtering

When configuring your integration, you have several options available to refine which artifacts are eligible to flow. The final mechanism available is *artifact filtering*, which is configured at the Integration level.

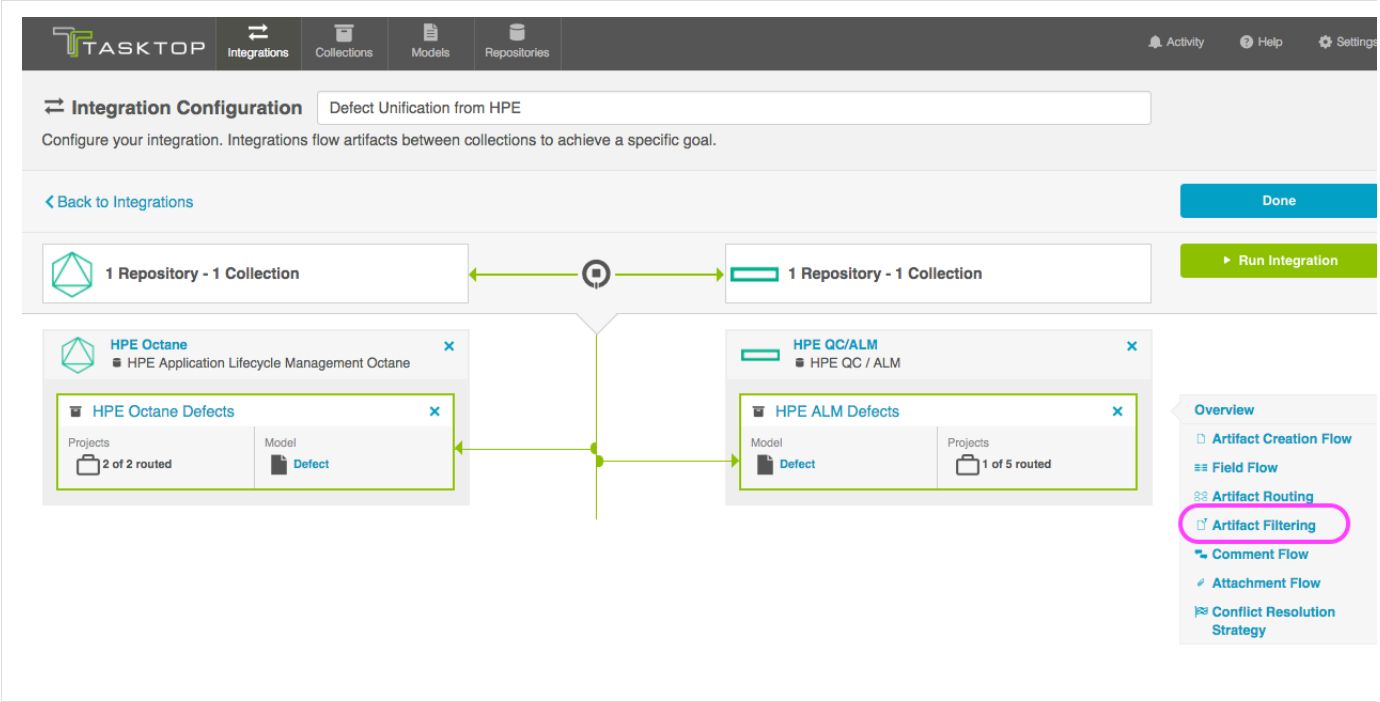


To configure *Artifact Filtering,* select 'Create filters (optional)' from the Integration Configuration Overview screen, or select 'Artifact Filtering' from the right pane of the Integration Configuration screen.

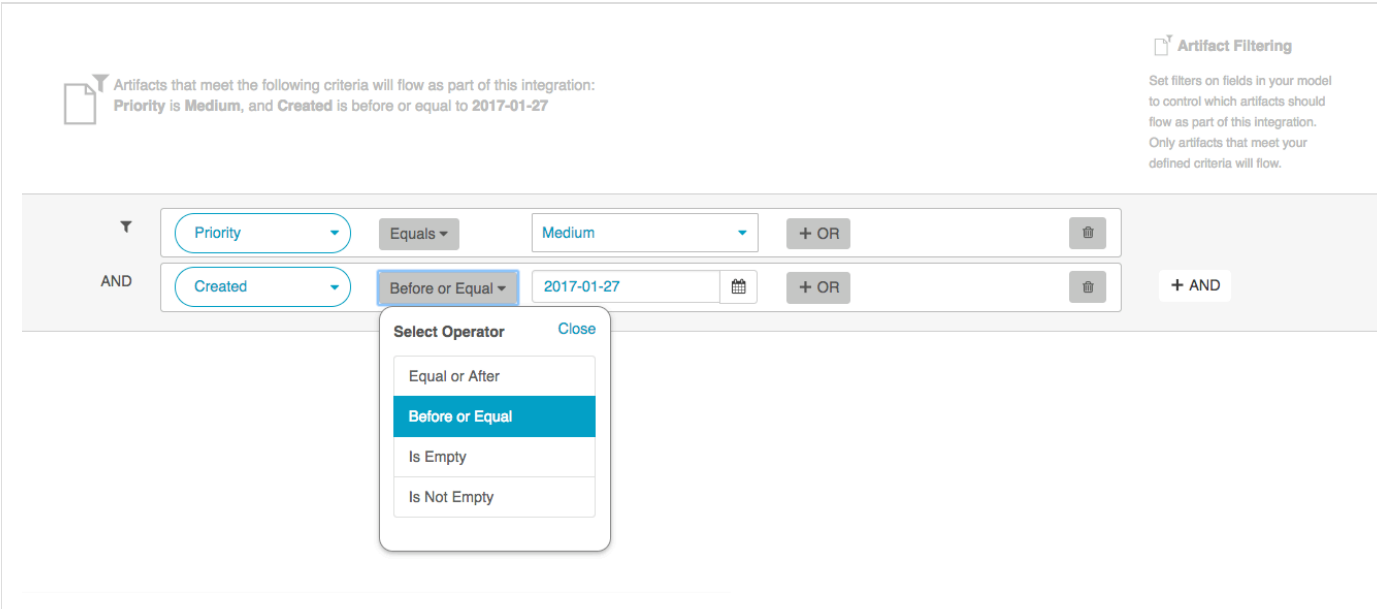
Artifact Filtering enables you to set filters on an integration in order to limit which artifacts are eligible to flow in your integration.

To use a field for artifact filtering, it must:

- Be a part of your model, and be mapped to the collection you are filtering from
- Be one of the following field types:
  - Single Select
    - Note that in cases where 'allow unmapped values to flow' is enabled in the model, only f ields that are already a part of the model will be considered for artifact filtering
  - Multi-Select
    - Note that in cases where 'allow unmapped values to flow' is enabled in the model, only f ields that are already a part of the model will be considered for artifact filtering
  - Date
  - Date/Time
  - String



This will lead you to the Artifact Filtering Configuration screen, where you can configure one or more criteria for artifact filtering.



In the example above, only defects that have a priority of "Medium" and a Created date on or before 1/27/17 will be created or modified in the integration.

💡 The filtering behavior is somewhat unique when using the Enterprise Reporting Template. See that section for more details.

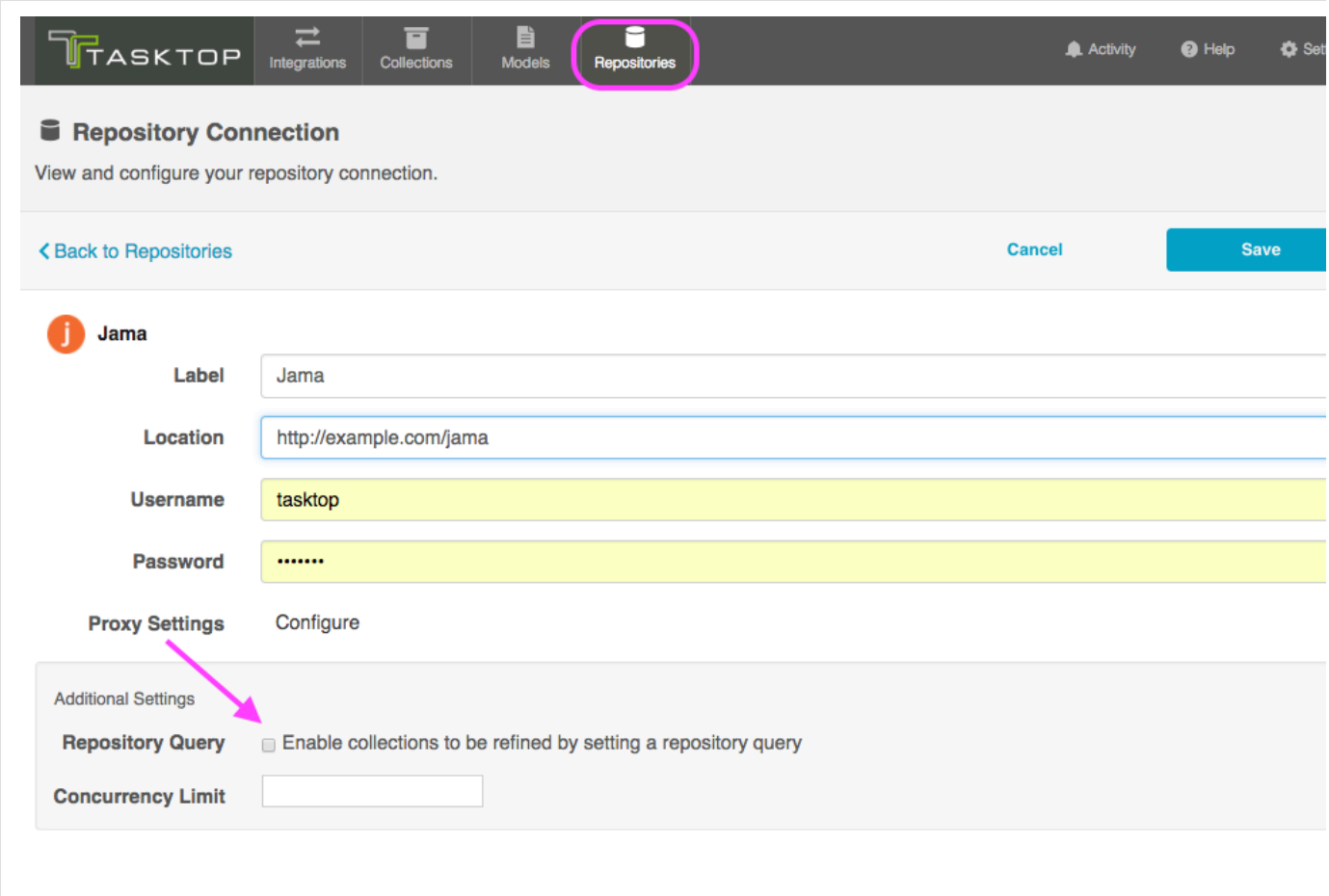
Filtering via Repository Queries

In rare cases, you may find that the best option to restrict the artifacts eligible to flow is by setting a query within the repository itself.

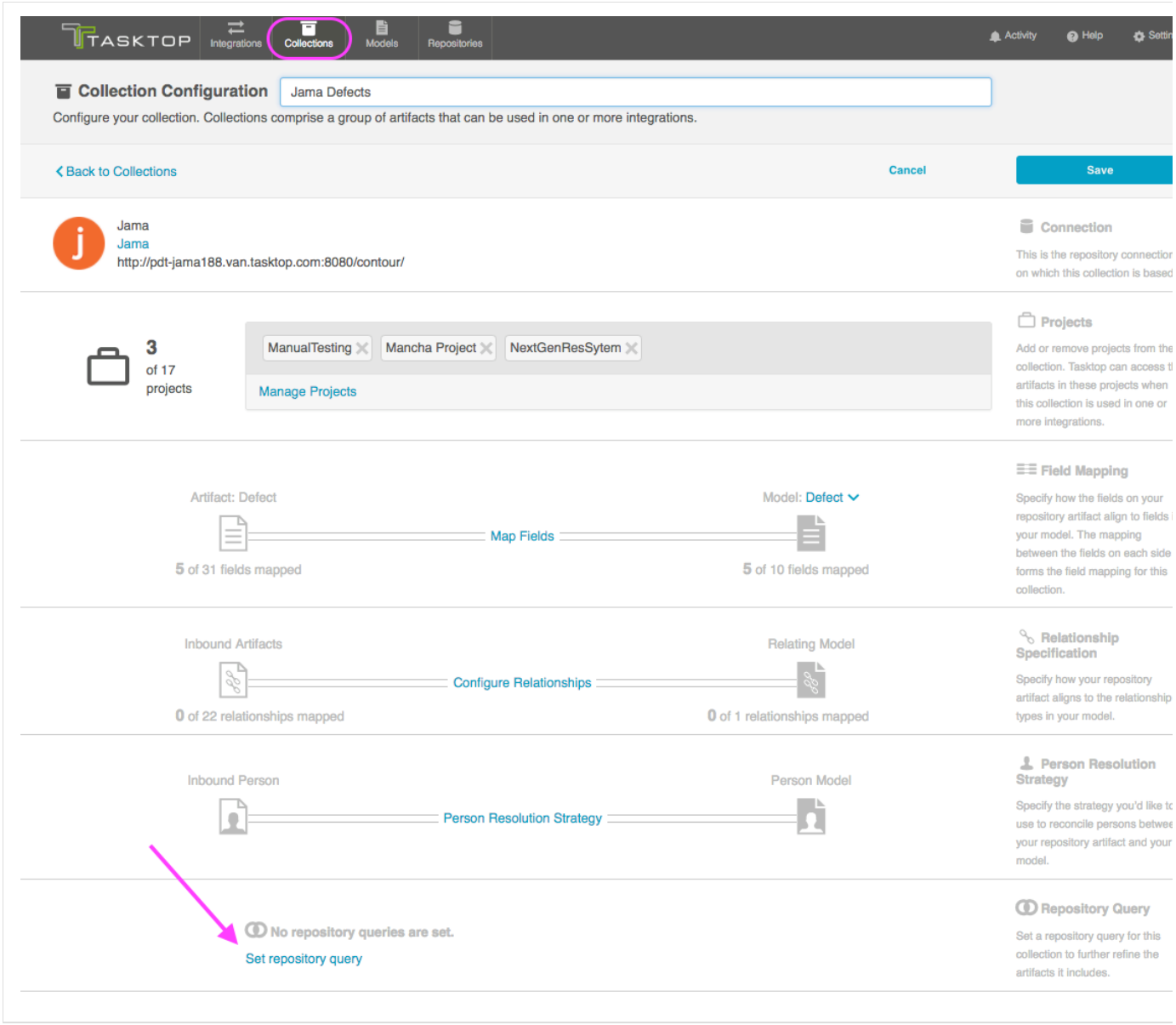
⚠️ Repository Queries are advanced functionality, and should only be used when you are truly unable to

filter as desired using the built-in Tasktop functionality of Repositories, Collections, and Artifact Filtering.

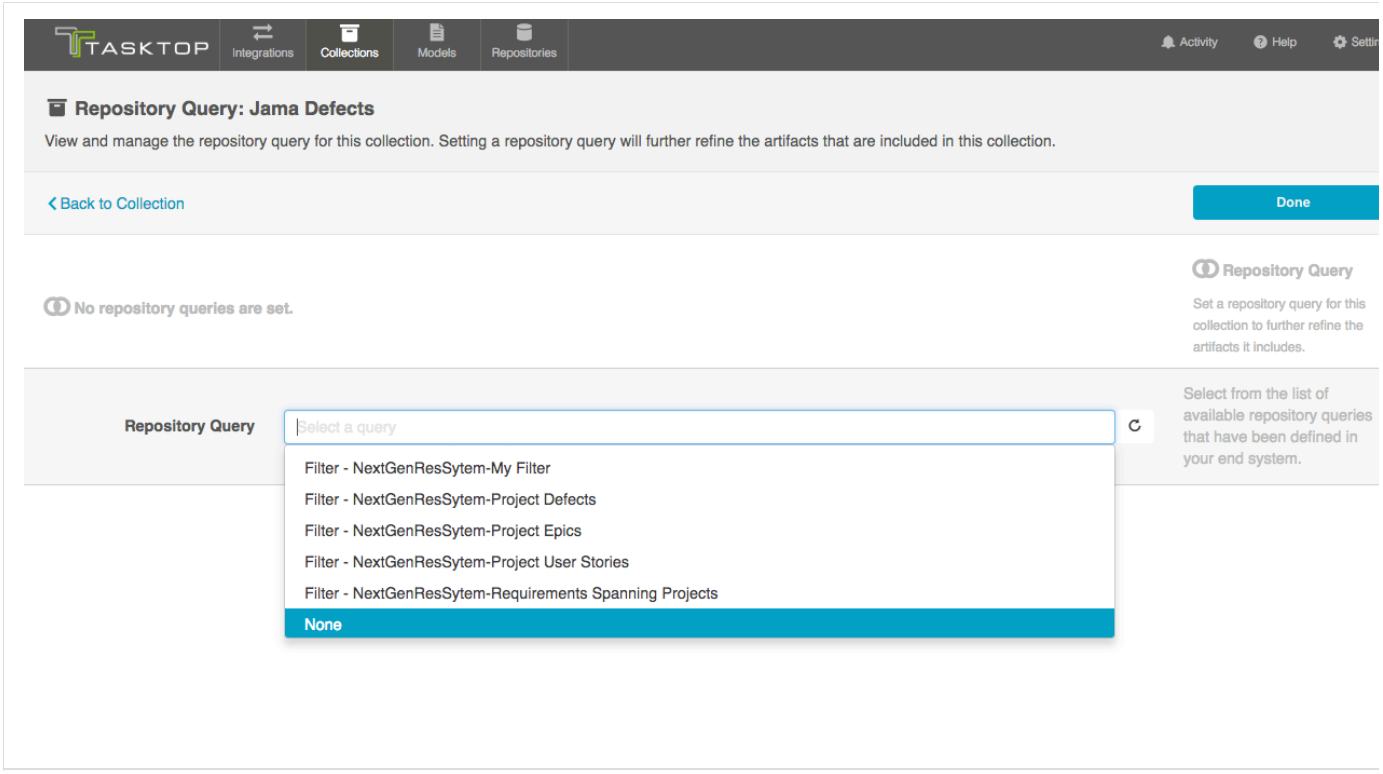
If you plan to utilize repository queries, check the box next to 'Enable collections to be refined by setting a repository query,' on the Repository Connection screen.



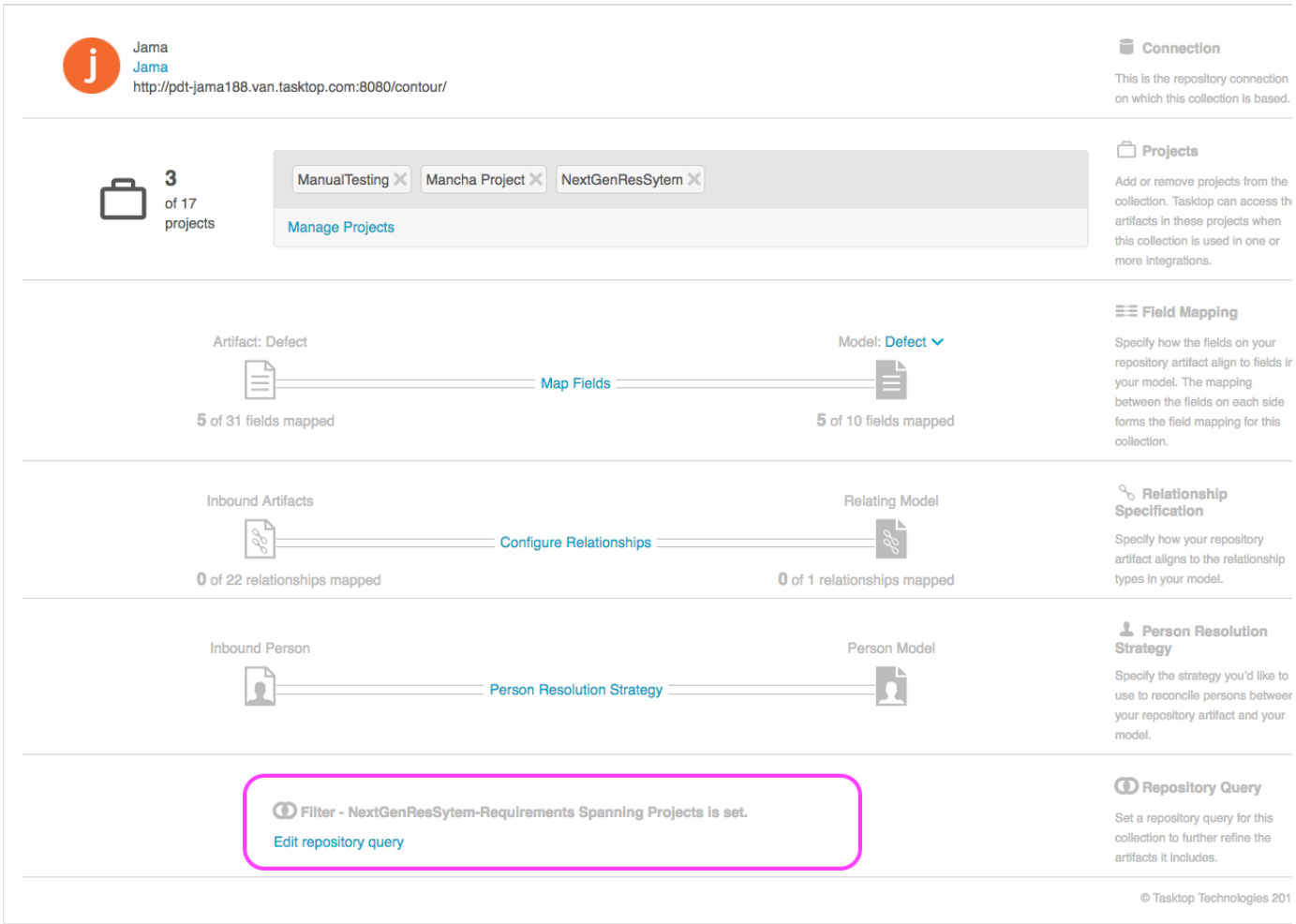
Once this is selected, you will be able to select a repository query at the Collection level for any collections utilizing this repository.



On the drill-in page, you'll see a list of available repository queries. Select the query you'd like to use, and click 'Save,' and then 'Done.'



You will then see the selected repository query on the Collection Configuration screen:



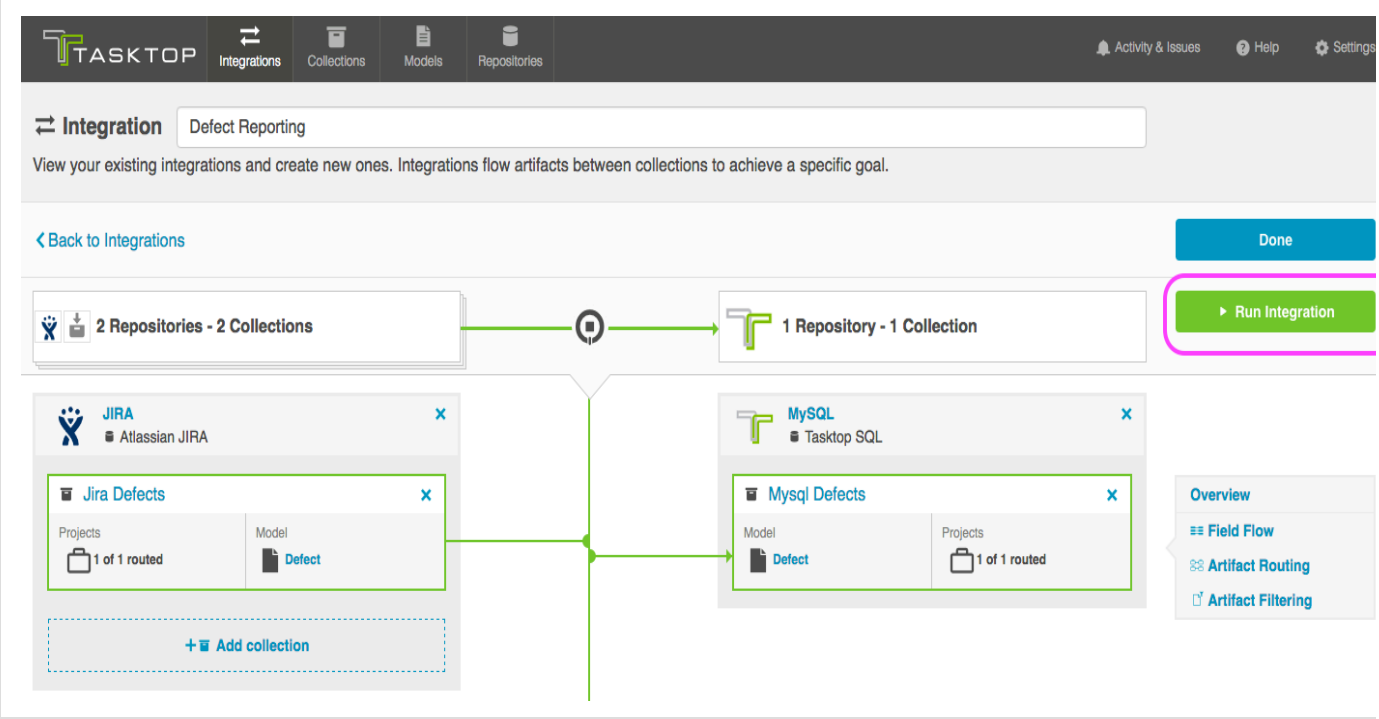
💡 Remember, applying a repository query to a collection will only further refine the artifacts included in that collection. If you select a query that encompasses artifacts in projects not in your collection, these artifacts will not be added to the collection unless you also add those projects to your collection as you normally would.

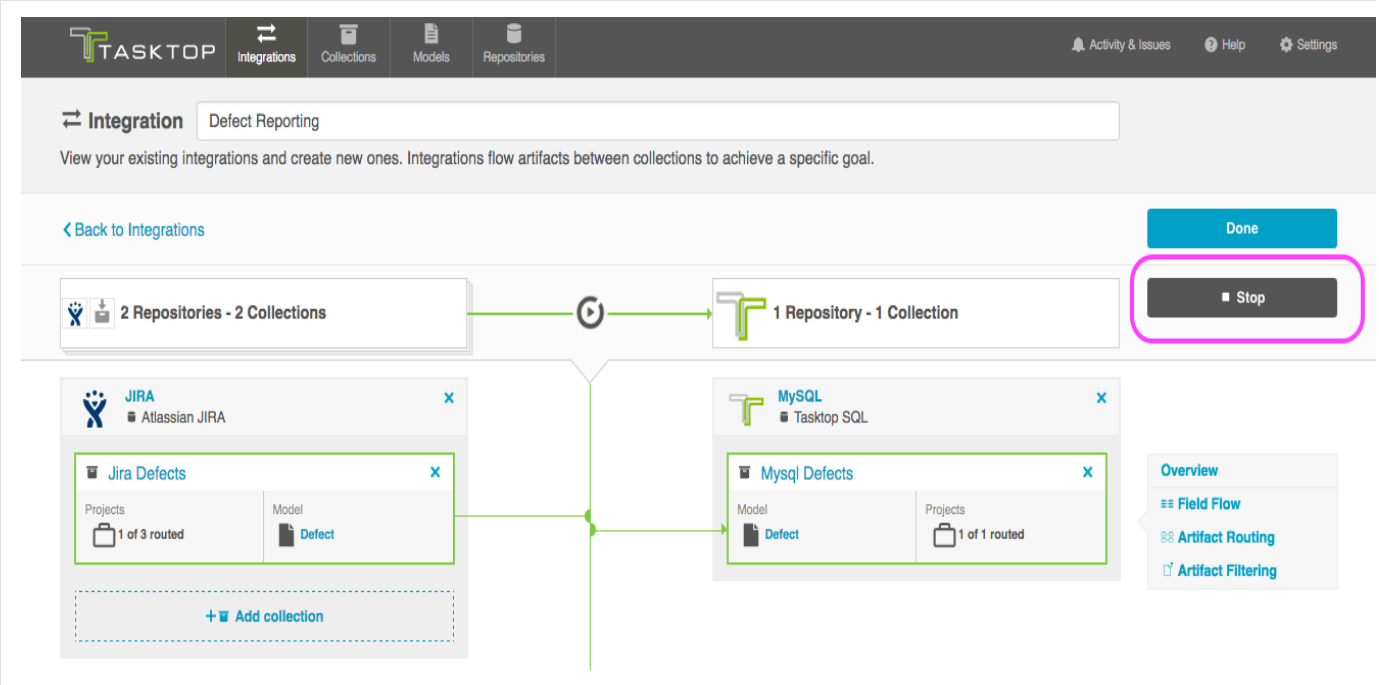
Running your Integration

There are two ways to start or stop your integration:

From the Integration Configuration Screen

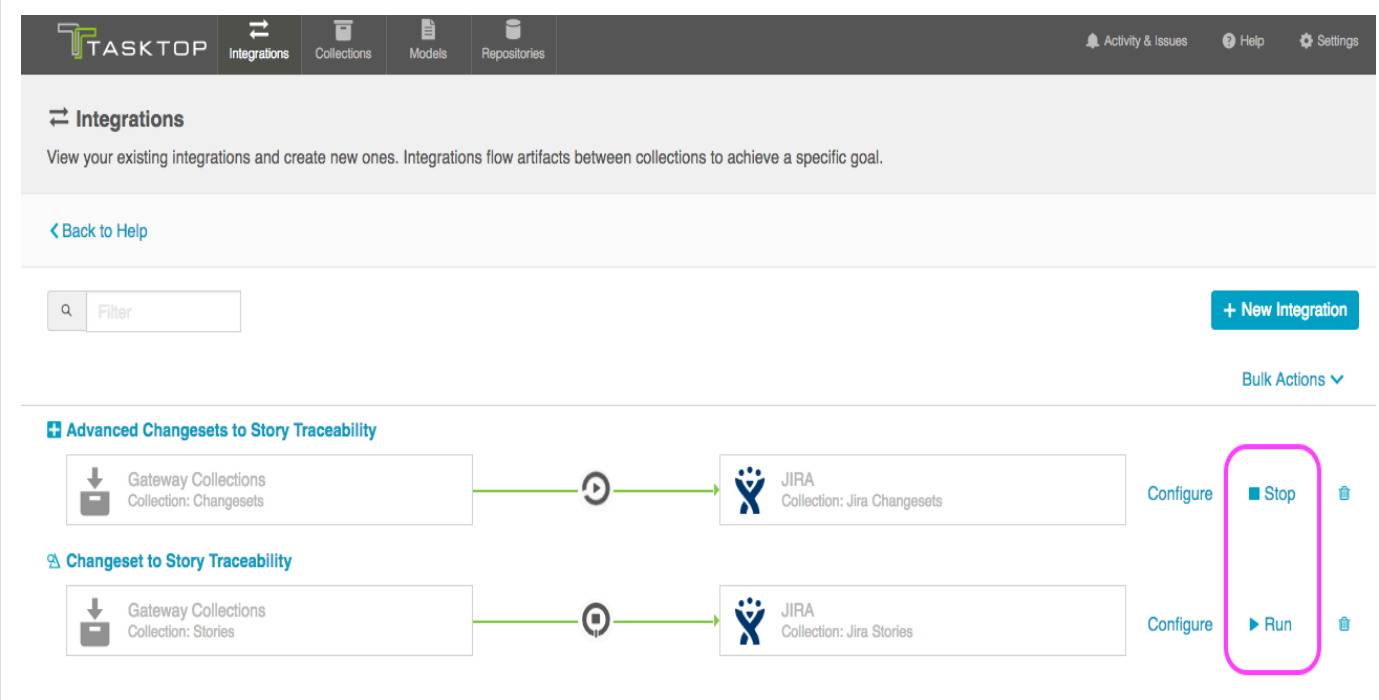
Simply click the 'Run Integration' to run the integration, and the 'Stop' button to stop the integration.

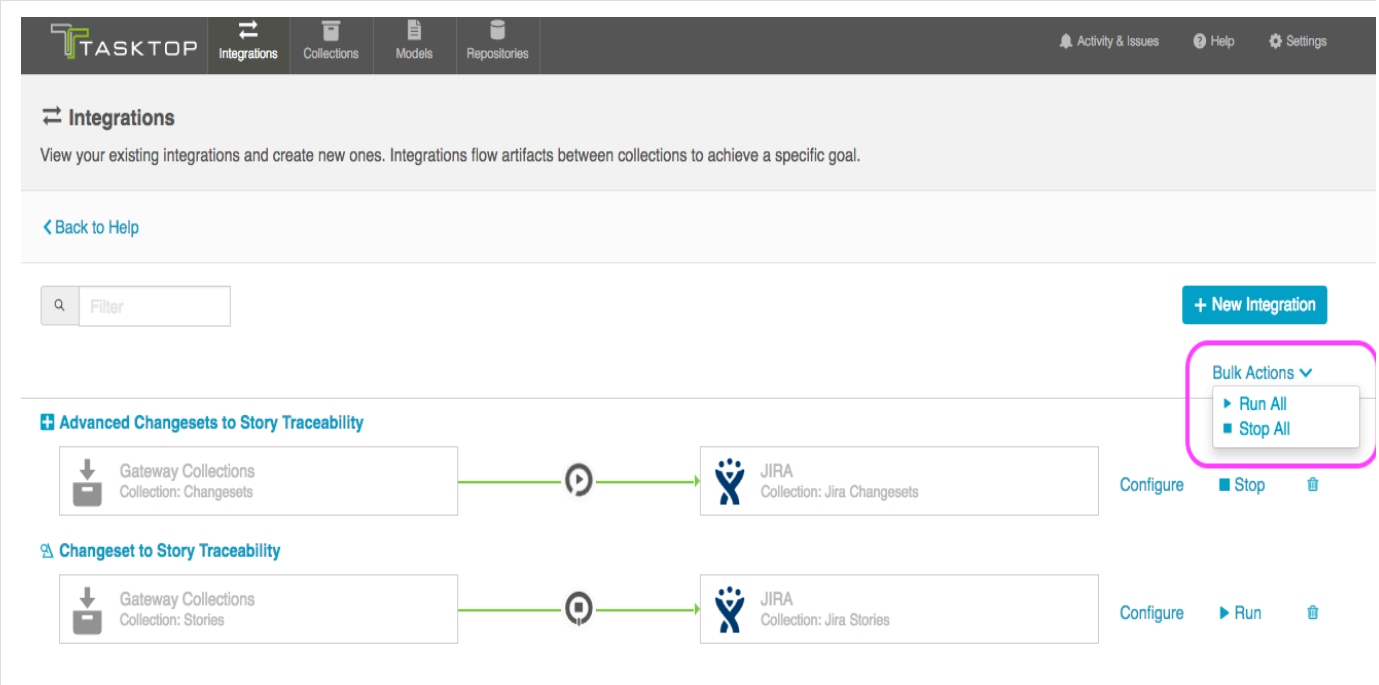




From the Integrations Page

Click 'Run' or 'Stop' next to each integration you would like to update. You can also use the 'Bulk Actions' button to run or stop all integrations.





Step 5: Expand or Modify your Integration

Tasktop: Apex Release (17.1)

You've already configured your integration, and it's running great! Now you'd like to increase the scale by adding additional projects from each of your end-repositories to your integration landscape, or by adding additional fields to your mapping. No problem - you can make these updates in just a few clicks!

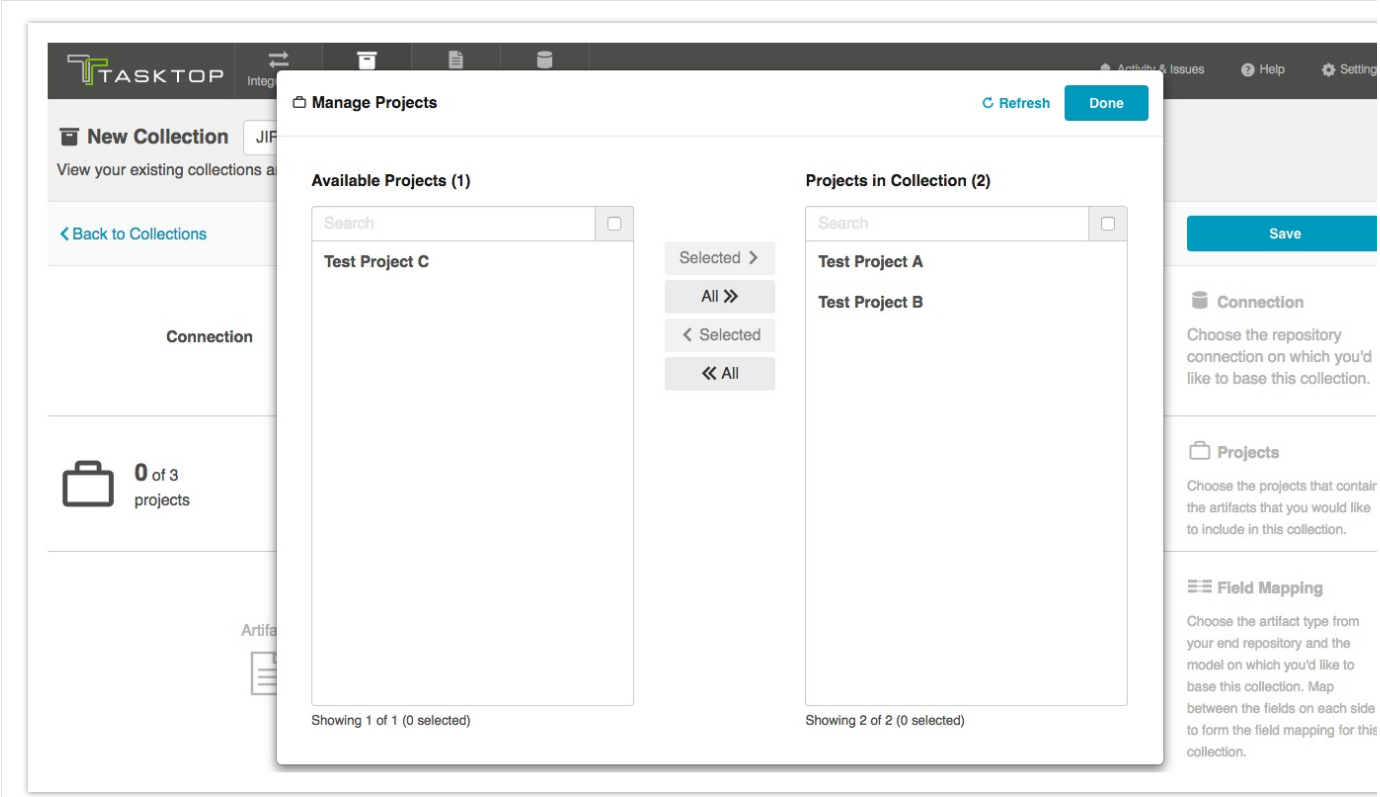
Below, we've included some tips and tricks on how to effectively scale your integration, as well as information on what to expect when you make modifications to your integration configuration after the integration has been activated.

Adding Projects

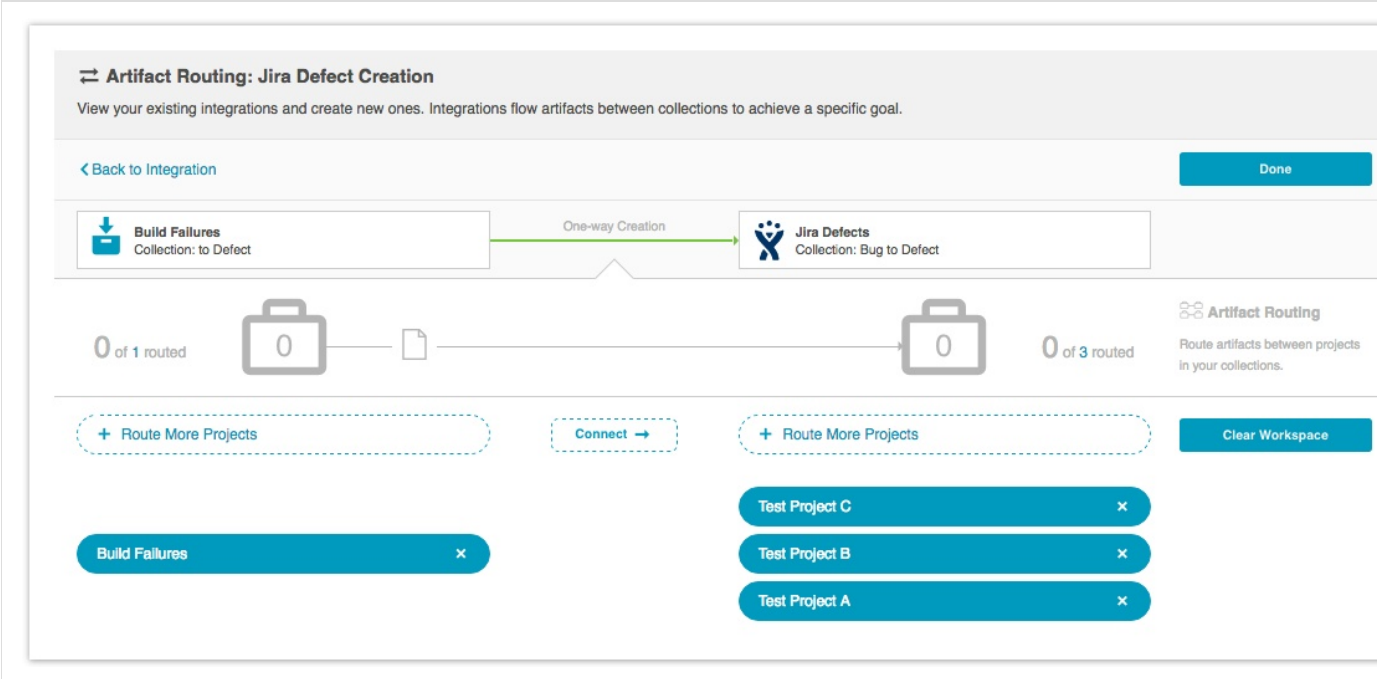
In order to add additional projects from one or more of your end repositories to your integration landscape, simply navigate to each collection, and add additional projects as desired. Once that's saved, navigate to the integration, click on 'Artifact Routing' and route the projects appropriately. You're all set - your integration will pull in the additional projects and flow data according to the configuration that you have already set.

Once the new projects have been added and routed, Tasktop will detect the artifacts contained within the new project(s) at the change detection interval (configured on the Settings page).

On the Collection Configuration Screen:



On the Artifact Routing Screen (in the Integrations section):

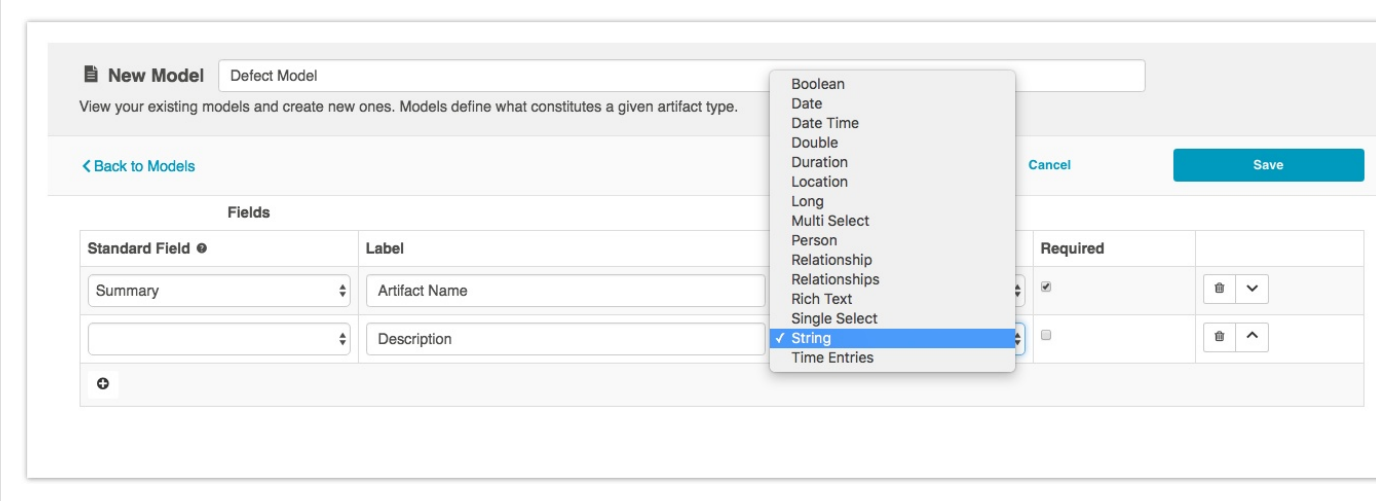


Adding or Editing Fields

If you'd like to add, remove, or edit a field in your model, Tasktop allows you to do so even after the Integration has begun to run. Once the field has been added to your model, navigate to your relevant collections and map that field as needed.

If you add or edit a field in your model, for newly created artifacts, the field will be synced automatically. Tasktop will detect these changes according to the change detection interval.

⚠️ Note that if you add or edit a new field mapping on an integration that has already begun running, Tasktop will not apply those new field mappings to artifacts that had already been synced and were created before that mapping had been added unless/until that field specifically changes on the artifacts.



Integration Templates

Tasktop: Apex Release (17.1)

Tasktop offers a range of Integration Templates to enable you achieve a diverse set of goals:

- Synchronize Integration: This integration connects teams working in different tools as they fulfill their roles in the software development lifecycle. As part of this integration, artifacts will flow between disparate repository collections. You can choose to have either one-way or two-way artifact creation. Artifacts created in repository collection A can create corresponding artifacts in repository collection B, artifacts in repository B can create corresponding artifacts in repository collection A, or both can occur in the same integration. You'll also configure the direction in which each field on those artifacts should be updated.

- Create via Gateway: This integration creates traceability between artifacts across the software development lifecycle. New artifacts will be created in a repository collection when artifacts are sent to Tasktop via a Gateway collection.

- Modify via Gateway: This integration creates traceability between artifacts across the software development lifecycle. Already existing artifacts in a repository collection will be located and modified in a specified way when artifacts are sent to Tasktop via a Gateway collection.

- Enterprise Reporting: This integration simplifies enterprise reporting by unlocking software lifecycle data from its application tool silos and providing a rich data repository for near real-time analytics. Records will be created in a single database when artifacts from one or more collections are created or changed.

Synchronize Integration Template

Tasktop: Apex Release (17.1)

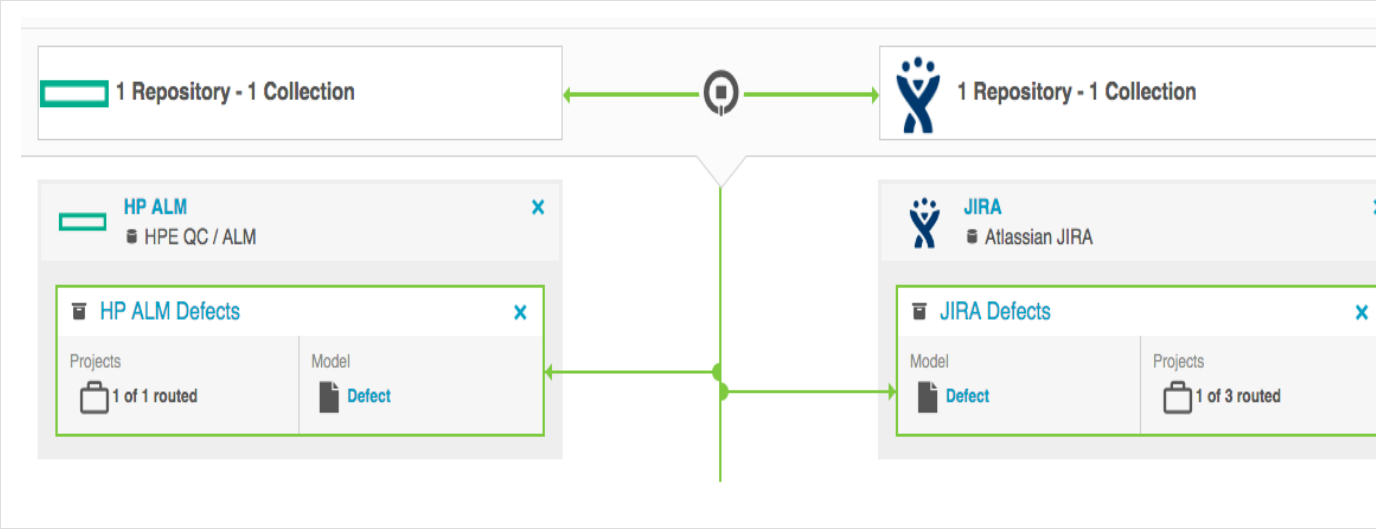
Check out the video below to learn how to configure a Synchronize Integration.

⚠️ This video assumes that you have already configured your repositories, models, and collections as outlined in the Quick Start Guide.

Use Case and Business Value

This integration connects teams working in different tools as they fulfill their roles in the software development lifecycle.

As part of this integration, artifacts will flow between disparate repository collections. You can choose to have either one-way or two-way artifact creation. Artifacts created in repository collection A can create corresponding artifacts in repository collection B, artifacts in repository B can create corresponding artifacts in repository collection A, or both can occur in the same integration. You'll also configure the direction in which each field on those artifacts should be updated.



Template Affordances

The Synchronize Integration Template allows you to flow artifacts between two repository collections.

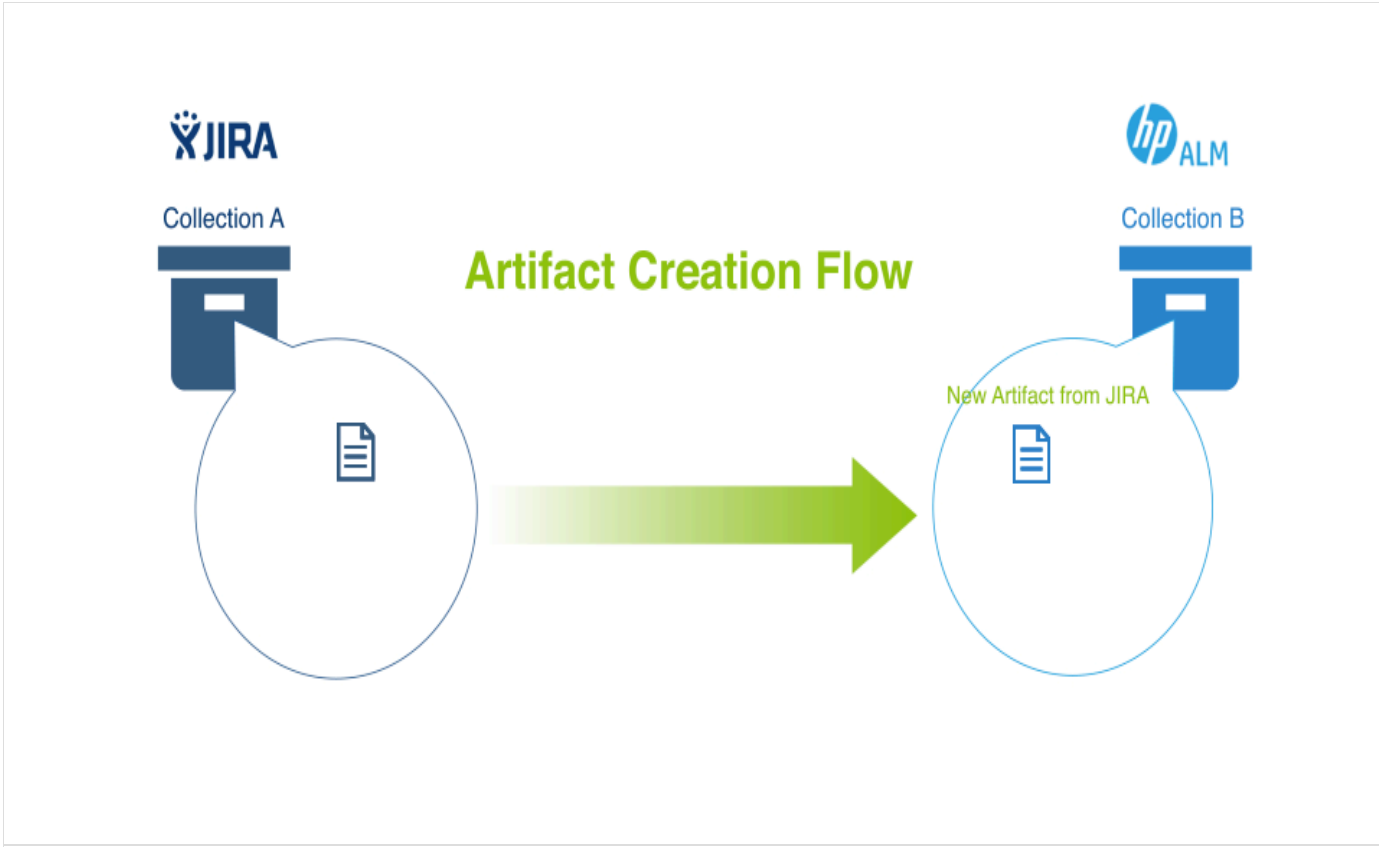
Template-Specific Configuration and Information

To create this integration, you can follow the general 4 steps described in the Quick Start Guide. However, there are a few additional steps which are unique to the Synchronize Integration Template:

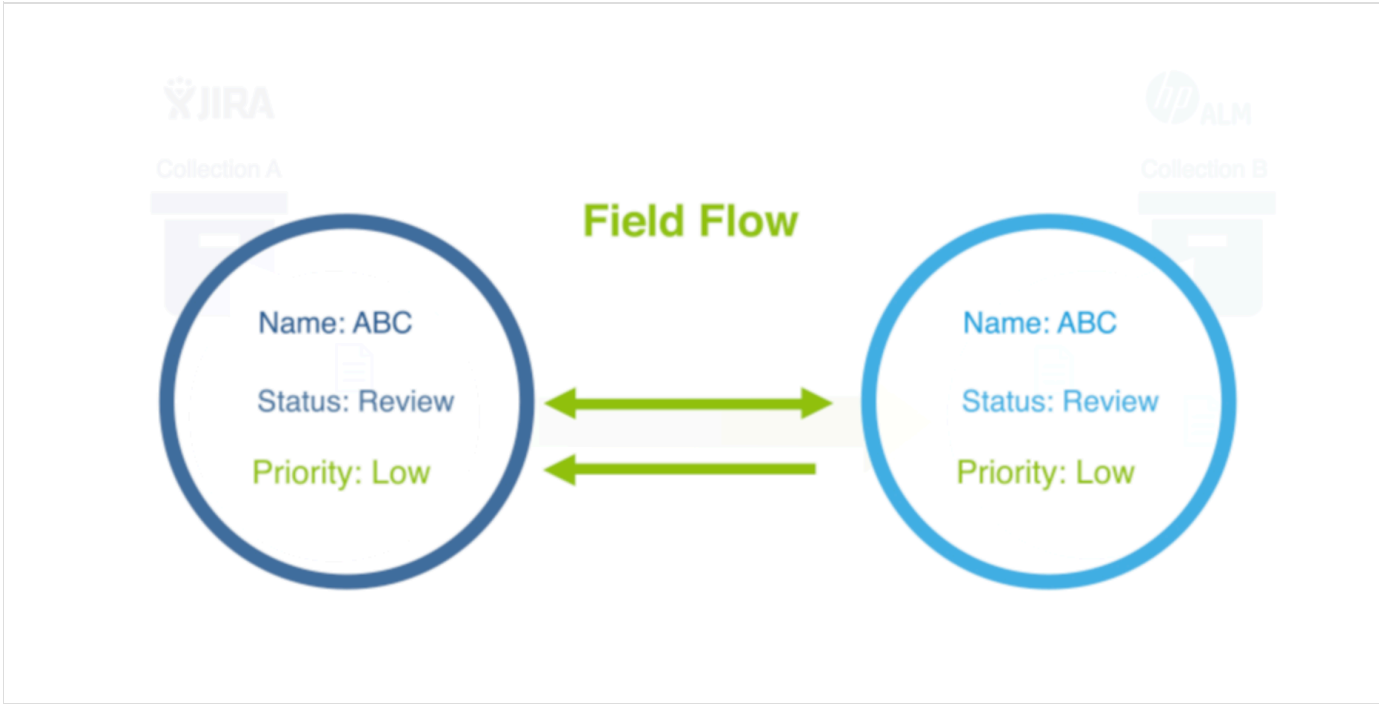
Artifact Creation Flow

The Synchronize Integration is unique in that it allows the user to determine whether to create artifacts in both repositories, or to create artifacts in just one of the two repositories.

Note that this setting relates only to the creation of artifacts (as opposed to the modification of fields on those artifacts). So for example, if I chose to set up one-way artifact creation from JIRA to HPE ALM, this means that when the integration is run, new or existing artifacts from JIRA would send to HPE ALM, but new or existing artifacts from HPE ALM would not send to JIRA. However, once a JIRA artifact sends to HPE ALM, if any modifications are made to that artifact in HPE ALM, that modification could flow back over to JIRA, based on the integration's Field Flow configuration. So while the integration is not creating n ew artifacts in JIRA, it can modify existing artifacts based on corresponding changes made in HPE ALM.



Note that in the image above, artifact creation flow is configured such that artifacts will only be created i n HPE ALM, and not in JIRA.

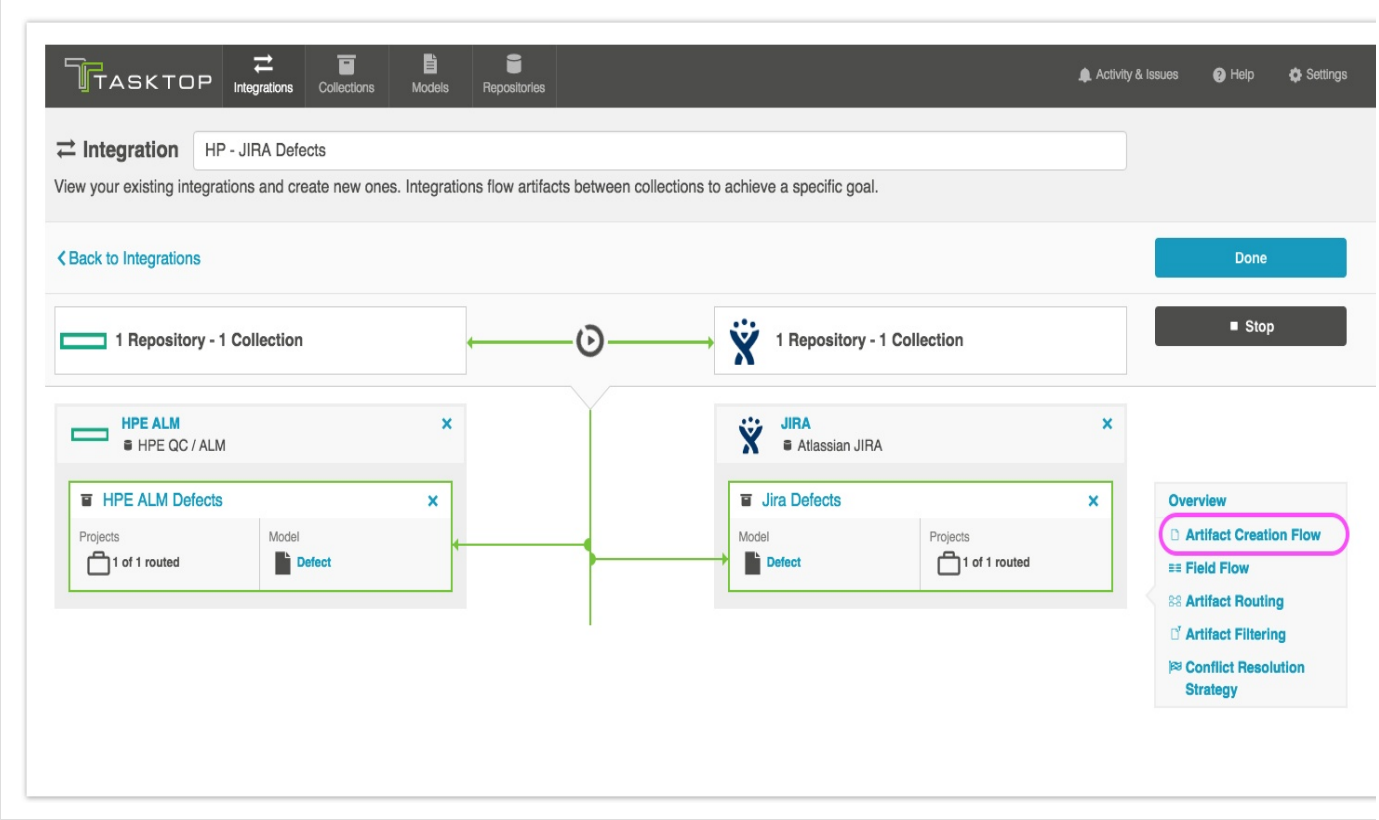


However, the same integration can have field flow configured bidirectionally. This means that even though artifact creation flow is only one-way (from JIRA to HPE ALM), if an artifact is modified in HPE ALM (i.e. to have status updated to 'done'), that modificaiton can still flow back to its corresponding

artifact in JIRA, based on the integration's field flow configuration.

Here's how to configure the Artifact Creation Flow:

From the Integration page, select 'Artifact Creation Flow'



This will lead you to the Artifact Creation Flow page, where you will be able to select from Two-way Creation (artifacts will be created in both collections to correspond to new and existing artifacts in the other collection), or One-way Creation (only one of the two repositories will have new artifacts created to correspond to new and existing artifacts in the other collection).

Field Flow

As you saw in the Quick Start Guide, you can configure field flow to specify which fields should flow in an integration. In the Synchronize Integration Template, field flow will function the same as in the other templates, with two simple additions:

1. You will also be able to specify which direction the fields will flow in. You can choose to have your fields flow bi-directionally between repositories, to flow in one direction, or not to flow at all.

2. You can also specify the frequency the fields will flow in:

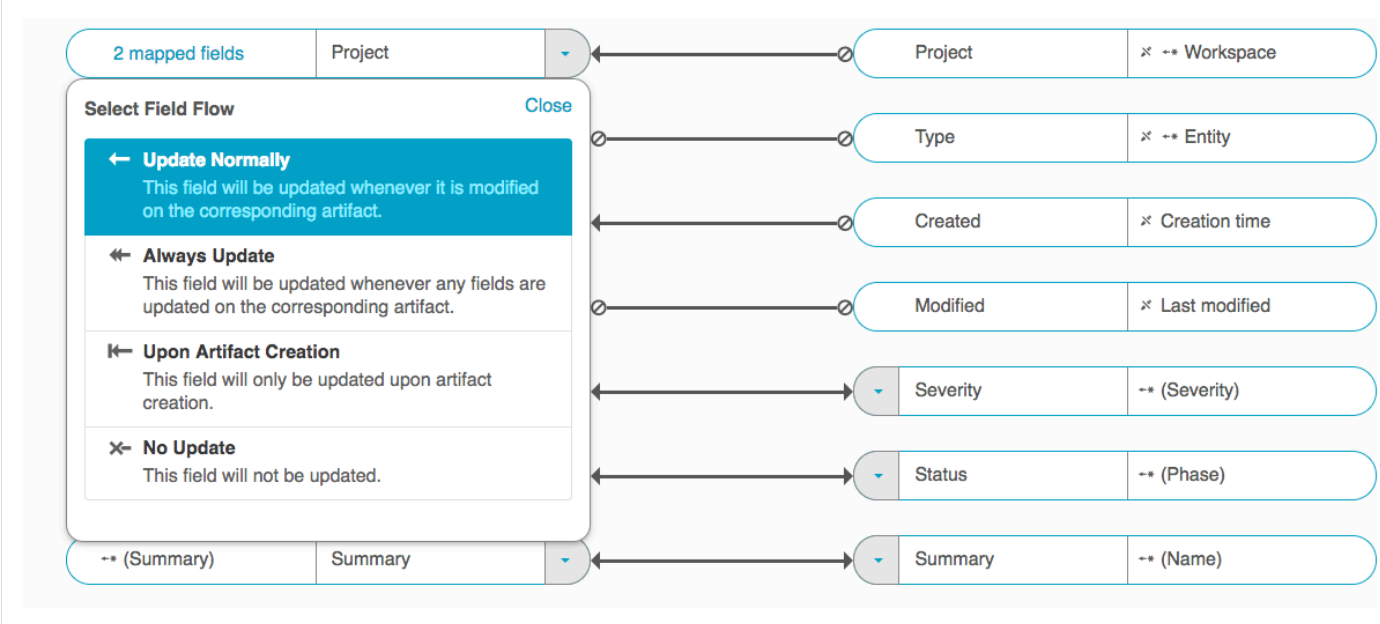
Update Normally: This field will be updated whenever it is modified on the corresponding artifact

Always Update: This field will be updated whenever any fields are updated on the corresponding artifact

Upon Artifact Creation: This field will only be updated upon artifact creation

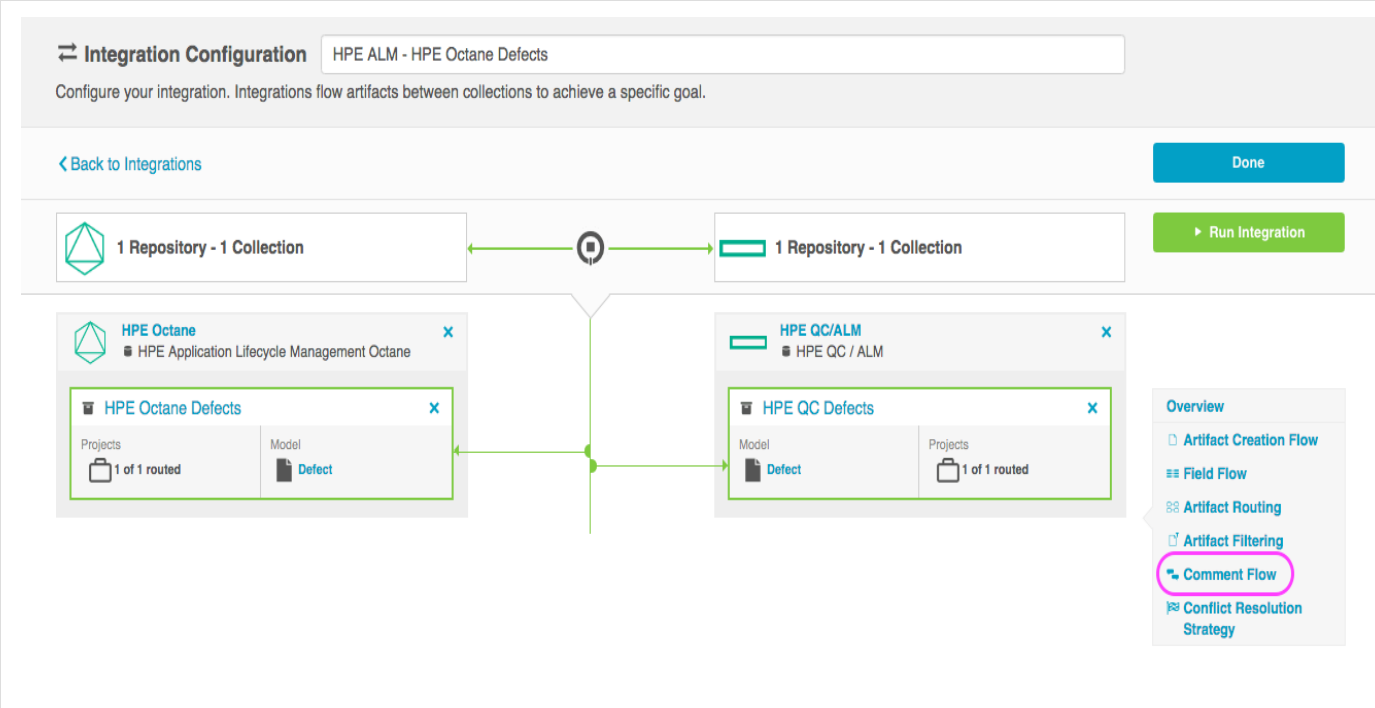
No Update: This field will not be updated

⚠️ Note: The field flow settings behave a bit differently for Constant Values. This is because constant values exist as part of your Tasktop configuration, and not on the artifact itself. Therefore, changes in constant values are not detected in the same way that updates made on the actual artifact are detected. If you change the constant value that is linked to your model, your integration will not automatically detect this update and sync it over. The value will only update if another field on that artifact is updated. Because of this, for constant values, "update normally" and "always update" will behave identically: meaning that the constant value will update whenever any other field is updated on that artifact.

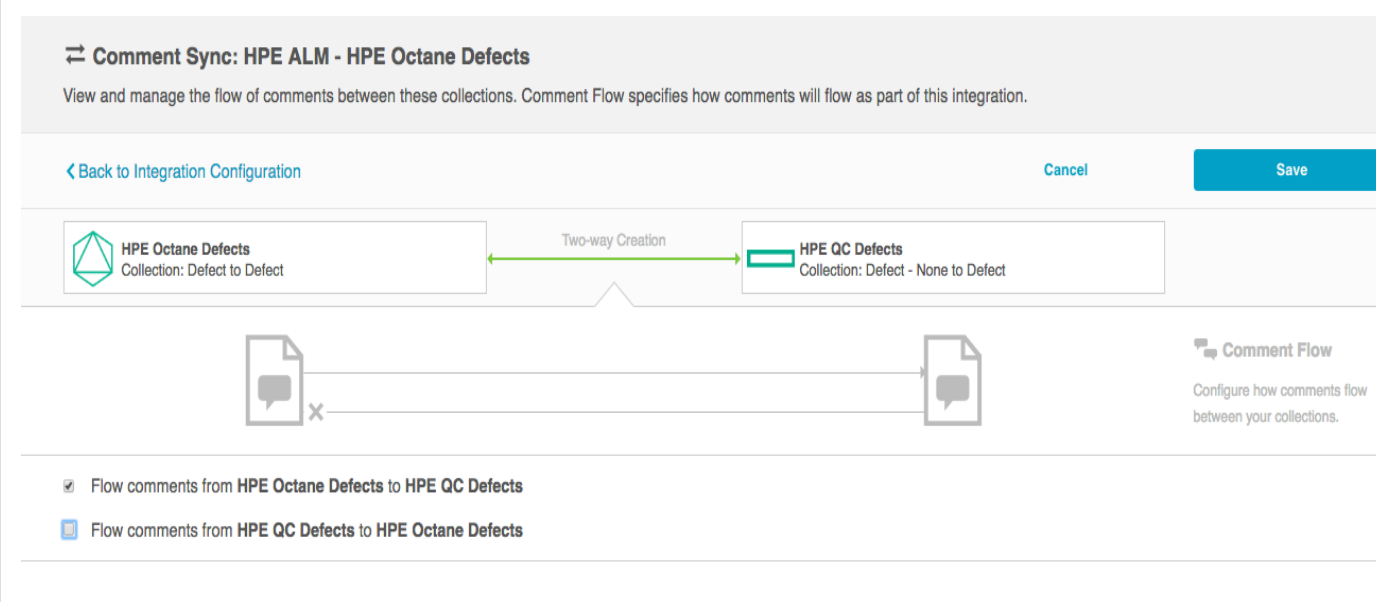


Comment Flow

When configuring a synchronize integration, you have the option of deciding to flow comments between collections. To enable and configure Comment Flow, click 'Comment Flow' on the Integration Configuration screen.



This will bring you to the Comment Flow page:

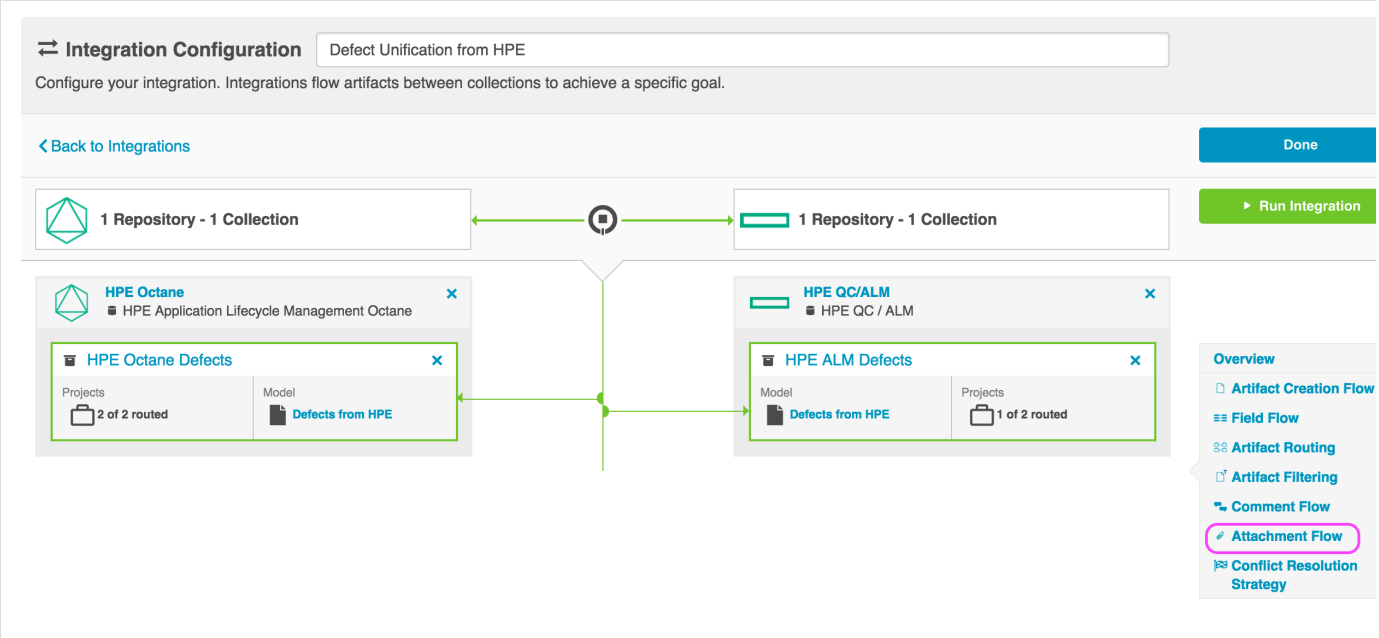


If your collection enables comment flow, you will be able to use the check-boxes to flow, or not flow, comments as part of your integration. You can choose to flow comments bi-directionally or in a single direction.

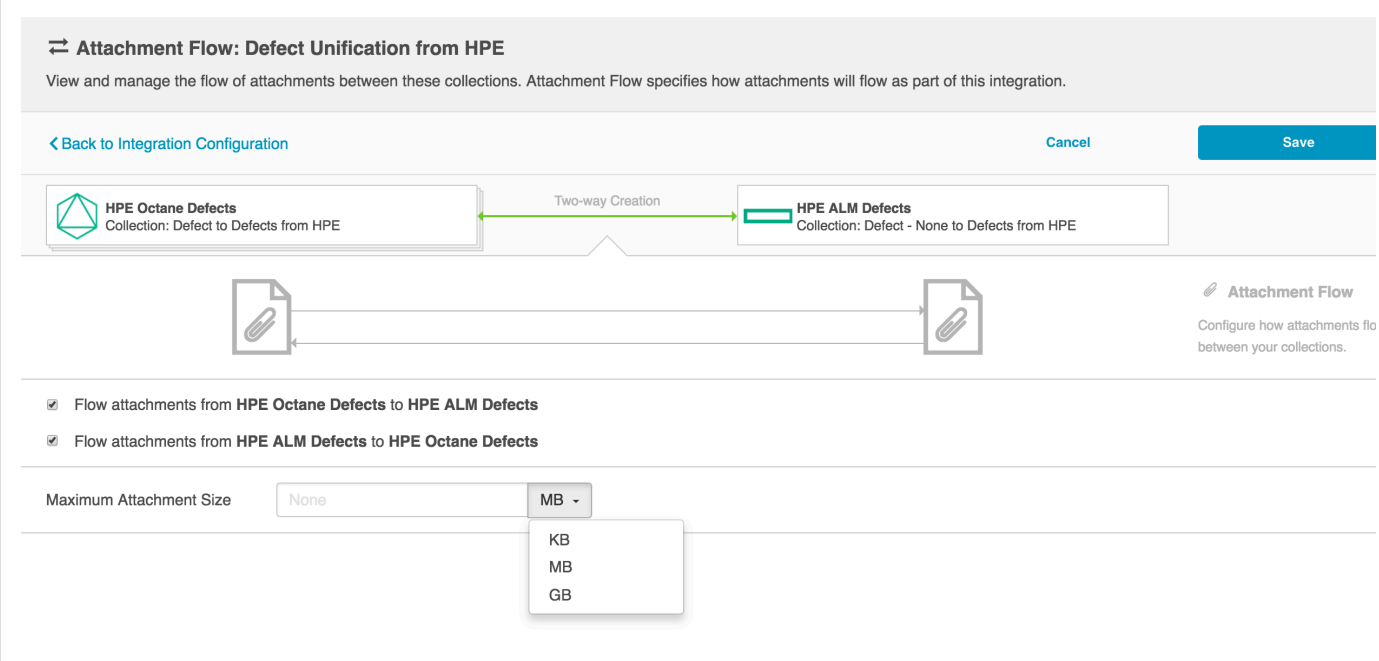
When a given repository supports impersonation (the ability for Tasktop to assign a specific user to a given artifact or artifact entity), Tasktop will assign the comment to the proper user given we can locate the user with the information provided from the artifact in the other repository. If a given repository supports impersonation but Tasktop cannot locate the person with the information provided from the artifact in the other repository, Tasktop will write out the comment attributed to the default Tasktop user and will also record the comment author from the other repository at the beginning of the comment. When a given repository does not support impersonation, Tasktop will write out the comment author from the other repository at the beginning of the comment.

Attachment Flow

When configuring a synchronize integration, you have the option of deciding to flow attachments between collections. To enable and configure Attachment Flow, click 'Attachment Flow' on the Integration Configuration screen.



This will bring you to the Attachment Flow screen:



If your collection enables attachment flow, you will be able to use the check-boxes to flow, or not flow, attachments as part of your integration. You can also configure the maximum attachment size. If attachments are larger than this size, they will be ignored by your integration.

💡 If you are unsure of the maximum attachment size allowed in your repository or if you leave this field blank and it turns out that the attachment is, in fact, larger than the maxiumum size the repository allows,

you will see an error message in Tasktop for that attachment. You can then deduce, based on the error message in Tasktop, what the maximum size is, and use that data to populate the field on the Attachment Flow screen.

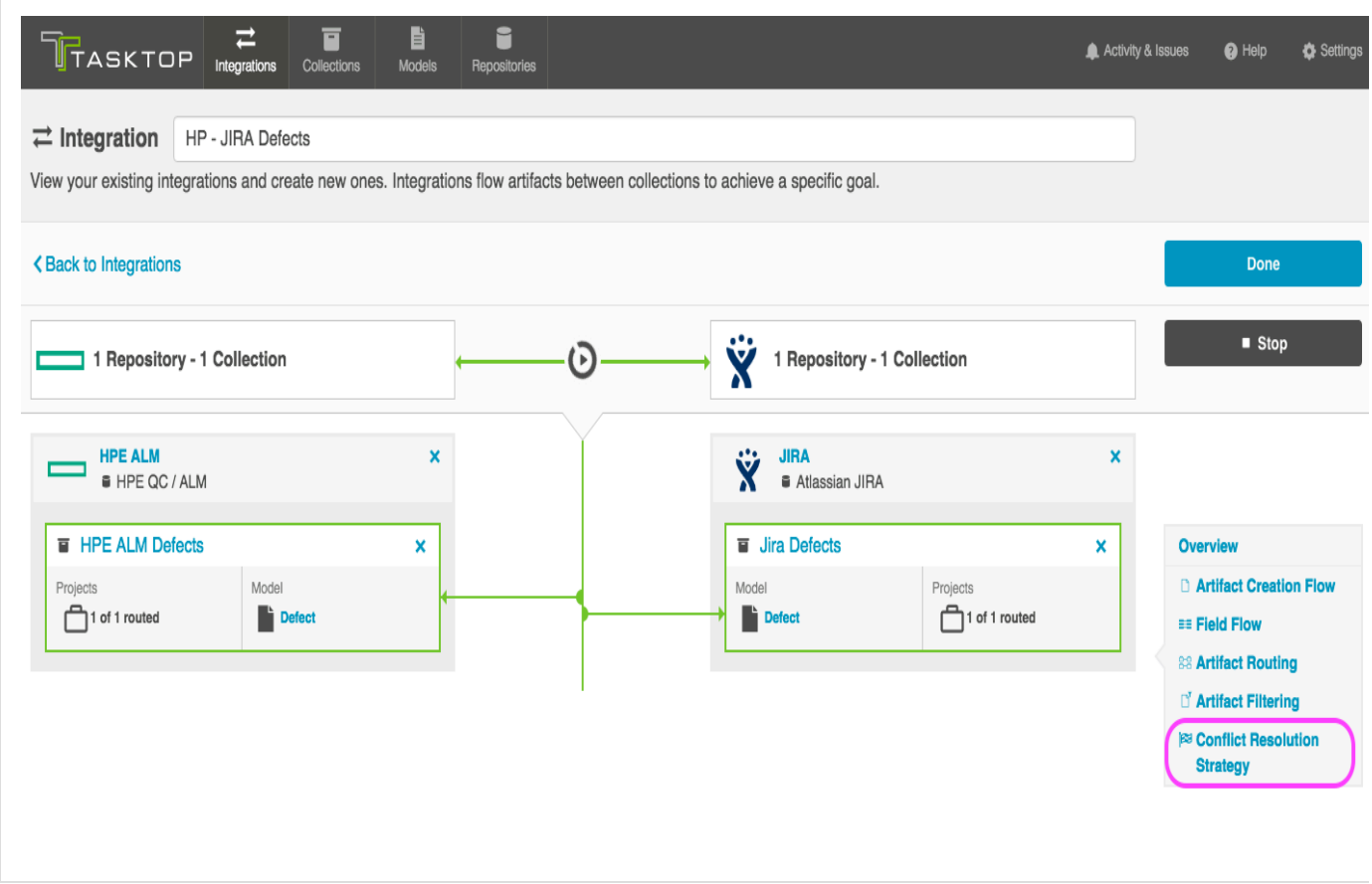
When a given repository supports impersonation (the ability for Tasktop to assign a specific user to a given artifact or artifact entity), Tasktop will assign the attachment to the proper user given we can locate the user with the information provided from the artifact in the other repository. If a given repository supports impersonation but Tasktop cannot locate the person with the information provided from the artifact in the other repository, Tasktop will write out the attachment attributed to the default Tasktop user. When a given repository does not support impersonation, Tasktop will similarly write out the attachment attributed to the default Tasktop user.

Conflict Resolution Strategy

Another unique byproduct of the Sync Integration is the Conflict Resolution Strategy, which allows users to control how a data conflict will be resolved. A data conflict occurs when mapped fields of an end repository get updated with different values in each repository before the synchronization scan runs, if bidirectional field flow is enabled for the Synchronize integration.

To illustrate this, let's imagine a scenario where you have been running a sync integration for defects between HPE ALM and JIRA with bidirectional field flow (the default setting) for some time. If HPE ALM's Defect #123 is modified to have one value (say, status = done), and its pair artifact in JIRA is modified to have a conflicting value (say, status = in progress) before the next synchronization search, this data conflict will need to be resolved in some way before the two artifacts can be synchronized.

To select your Conflict Resolution Strategy, click the 'Conflict Resolution Strategy' link on the right side of the Integration page.

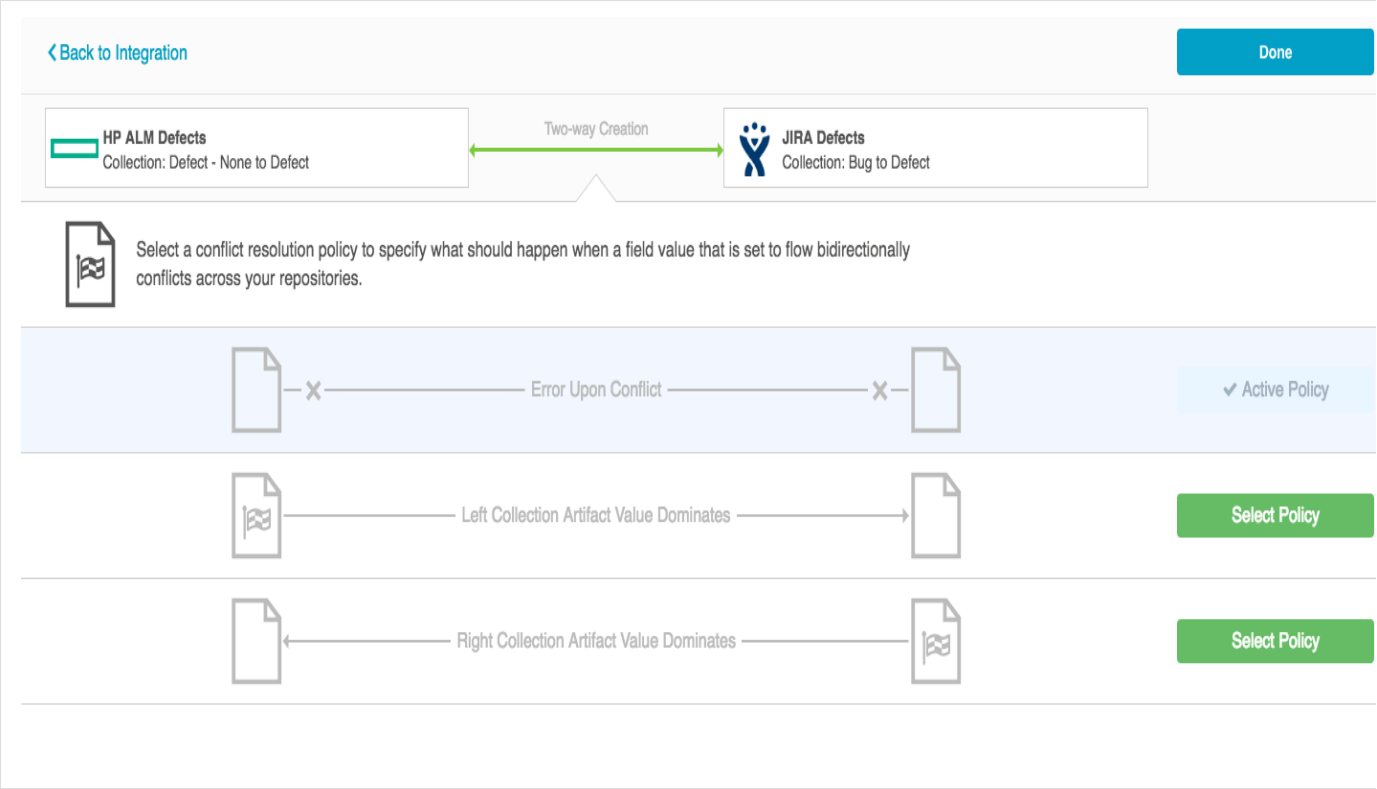


You will have three options for your Conflict Resolution Strategy:

1) Error upon Conflict: An error will be generated, and no updates will be made for the conflicted field, or any other fields on the artifact. The error message will notify you that the conflict occurred and will provide steps on how to resolve the conflict. Note that once a conflict is detected, no subsequent updates will be made to the artifact pair until the conflict is resolved.

2) Left Collection Artifact Value Dominates: Values from the artifact in the left collection will over-write the values in the right collection.

3) Right Collection Artifact Value Dominates: Values from the artifact in the right collection will over-write the values in the left collection.



Synchronizing Internal Relationships

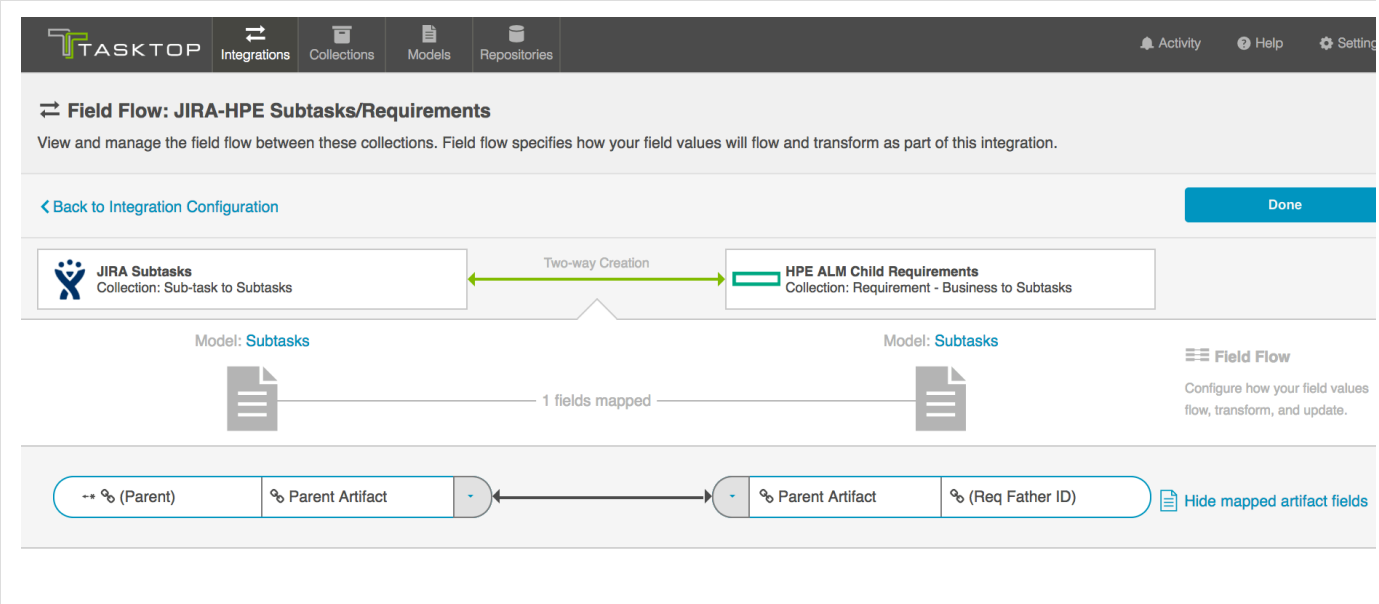
Tasktop affords you the ability to not only flow various artifacts between your collections, but also to mirror the relationships between those artifacts.

Below, we'll outline how to configure parent-child relationships in a synchronize integration, though the same process can be followed for a multitude of relationship types:

1. In Collection A, we have parent-tasks liked to child-subtasks. In Collection B, we have parent-requirements linked to child-sub-requirements. To flow these artifacts along with their relationships, we will need to configure two integrations:
    a. Task-Requirement Synchronize Integration
    b. Subtask - Sub-Requirement Synchronize Integration, with 'parent' relationship field
2. First, configure your Story-Story Synchronize Integration as you typically would
3. Next, configure your Subtask - Task Synchronize Integration
    a. Ensure that your model includes the 'Parent' Smart Field.
    b. On your Collection pages, click 'configure relationship types,' and map the 'parent' field appropriately

c. On your Integration Field Flow page, you will see the two relationship types mapped to one another.

4. Run both integrations. You sill see your parent artifacts, your child artifacts, as well as *their relationships to one another* successfully flow as part of your integration.

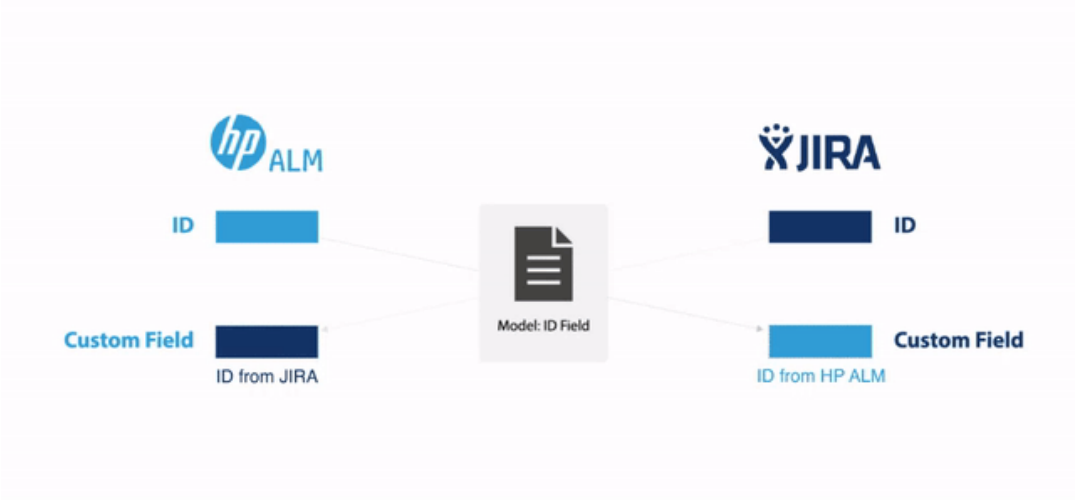


Synchronizing an Artifact ID or URL Reference

Imagine this scenario: You are syncing defects between two end repositories: JIRA and HPE ALM. You've created a defect, "Defect A" in JIRA and synchronized it over to HPE. Now, you'd like to have a field in JIRA which will tell you the defect ID or URL for the corresponding defect that has been created in HPE. You'd similarly like to have a field in HP to reference its corresponding JIRA artifact. This will allow you to easily trace the corresponding artifacts that have been created in your other end repositories.

To set this up, you will need to configure two different field mappings in each collection:

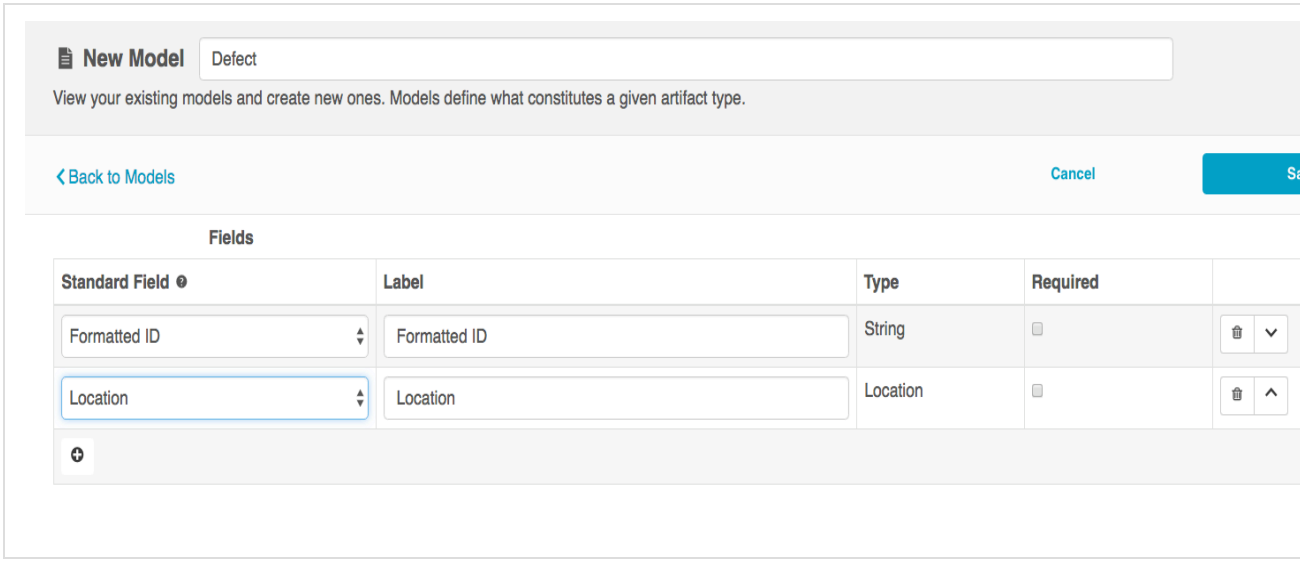
- You will need to specify which field to pull the ID (or URL) from
- You will need to specify which field to use to store the ID (or URL) of the artifact in the other repository



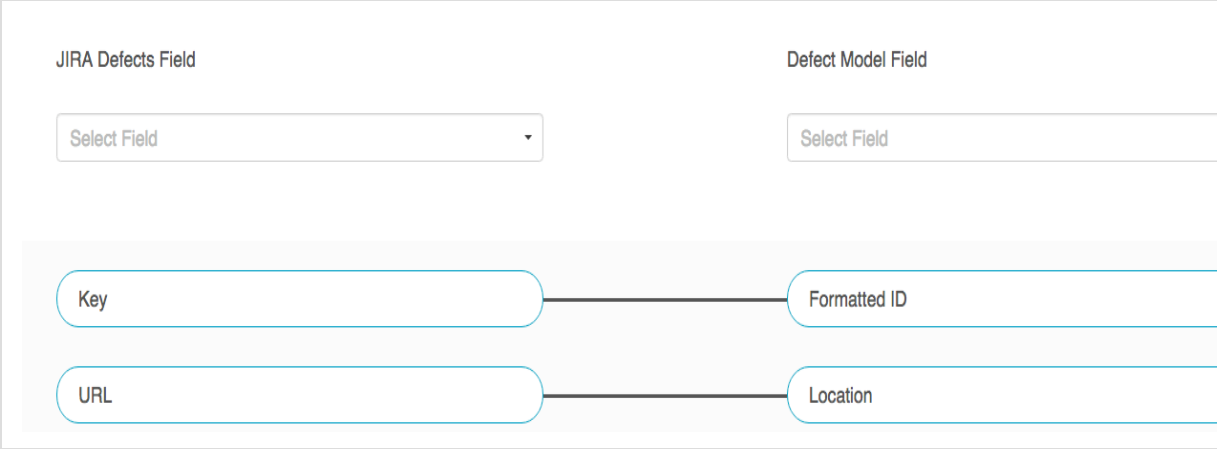
In the diagram above, you can see that HPE is sending its ID field to a custom field in JIRA, and that JIRA is sending its ID field to a custom field in HPE. In order to set up this integration, you will need to configure your model to accept that ID field. We'll walk through how to do that below.

The instructions below will walk you through how to set up this configuration for the ID field, but the same instructions will also apply for location/URL:

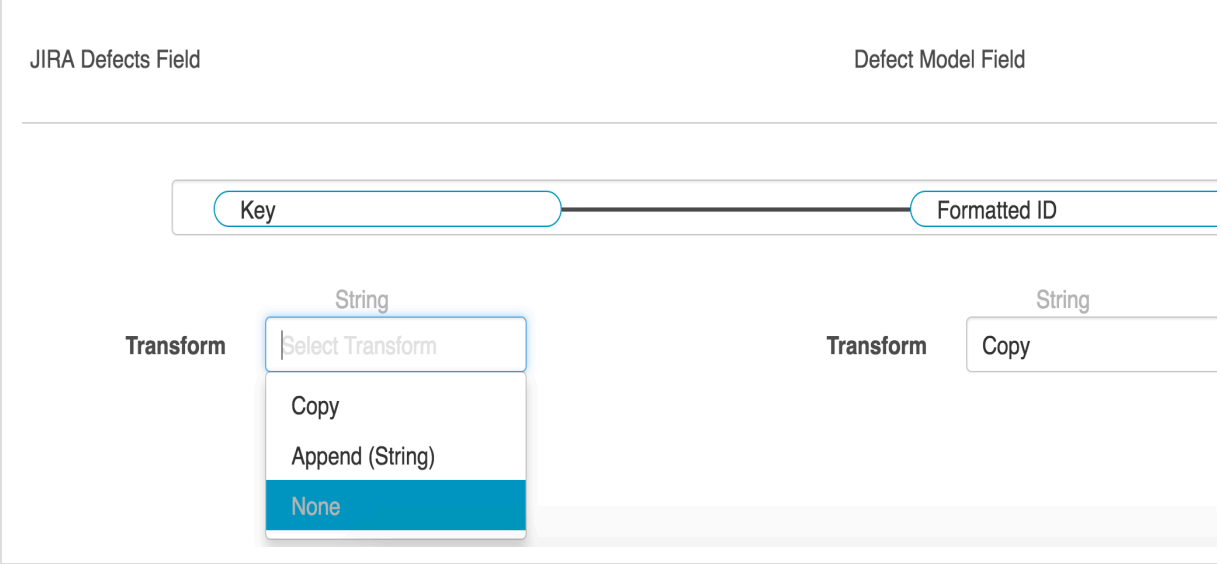
1. Go to the Model that you are utilizing in the integration. Ensure that your model includes the Formatted ID field. We've also shown the 'Location' field below, for reference.



2. Go to the Collections page for *each* of your end repositories, and set up mapping to tell the integration where to pull the ID from:
    a. Map the Formatted ID model field to the corresponding field in your end repository. This is the field that the collection will take the ID data from. Note that Formatted ID is called 'Key' in JIRA, but may be referred to using a different name in another end repository (i.e. 'issue ID').

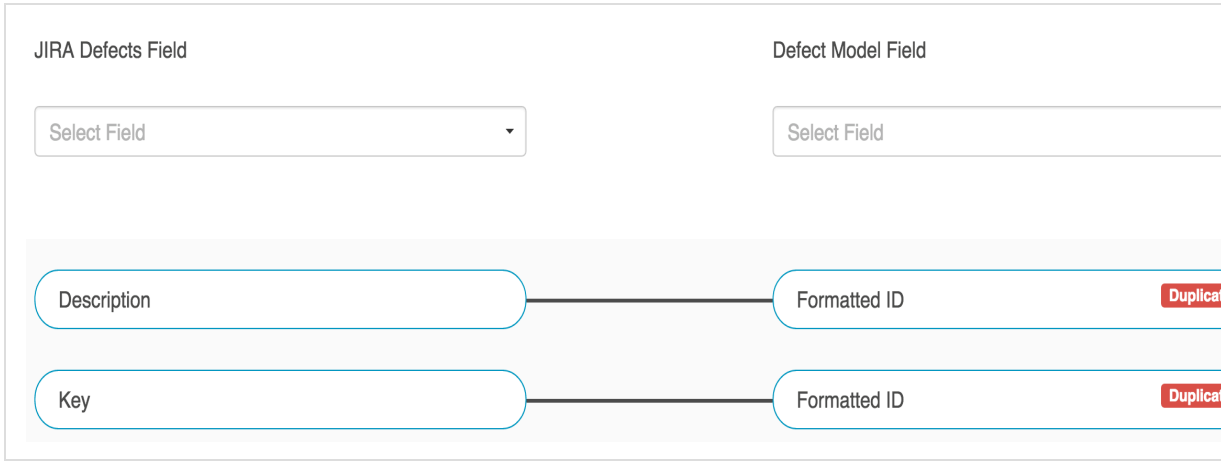


b. Click 'Configure' next to your mapping, and update the Transform values as outlined below. This will tell the collection to *send* data from the Key to the model, but not vice versa.

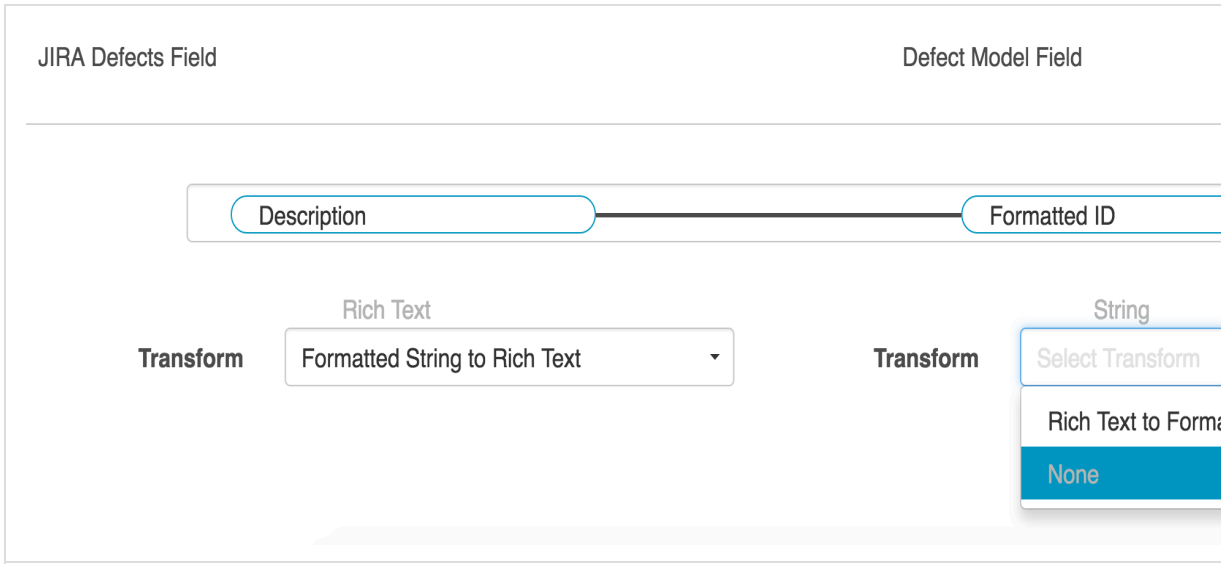


c. Repeat these steps in your other end repository.

3. Now that our model is able to acquire ID data from each end repository, let's tell it where to store that data in the other end-repository, once it's received from the model. To do this, you will set up an additional mapping in each Collection:
   a. Navigate to one of you Collections
   b. Map Formatted ID in your collection once more, this time to determine where you would like to store this data in your end repository. The field mapping page will tell you that this is a 'duplicate,' but that is ok!

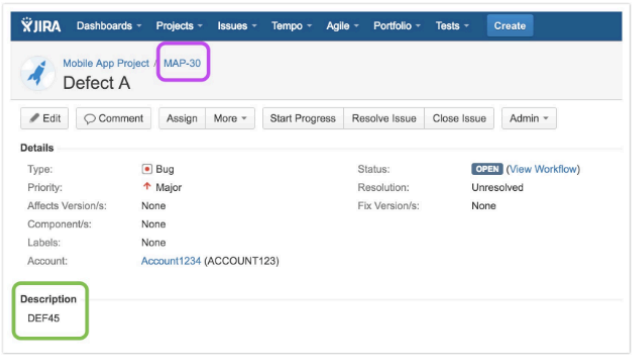


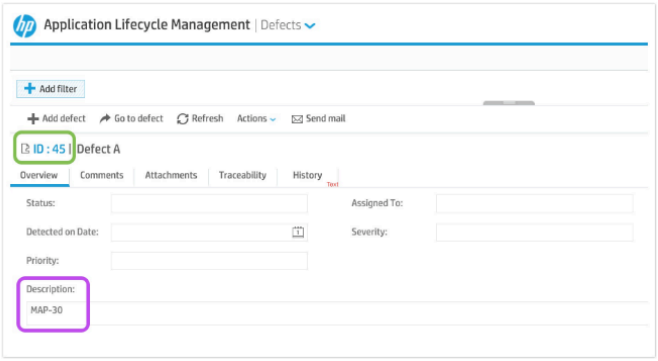
c. Click 'Configure' on the new mapping, and configure as shown below. This will tell the collection to take data from the model and send it to the 'Description' field, but not vice versa.



d. Save your mapping and collection.
e. Repeat these steps on your other collection

4. When you run the integration, the ID of the corresponding artifact will now flow to the original artifact (and vice versa), as specified in your field mapping:





Defect in JIRA: HPE ID is listed under 'Description'
Corresponding defect in HPE: JIRA ID is listed under 'Description'

Create via Gateway Integration Template

Tasktop: Apex Release (17.1)

- Use Case and Business Value
- Template Affordances
- Template-Specific Configuration and Information

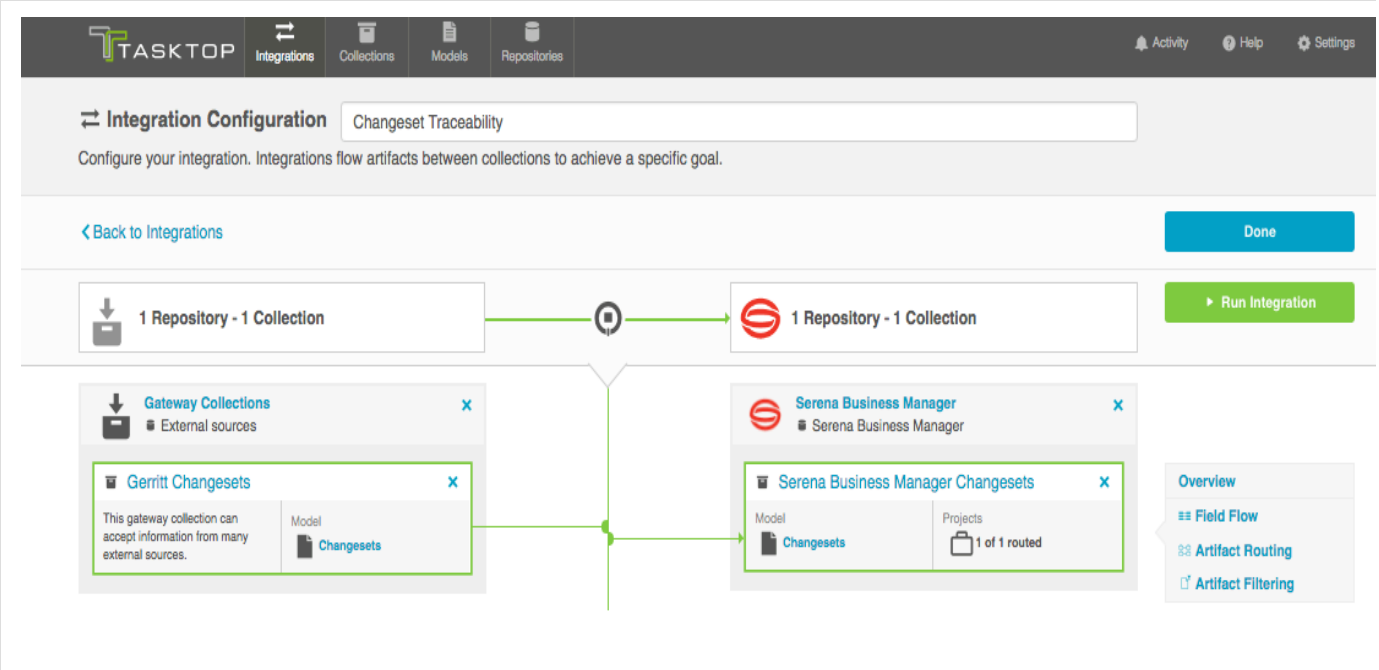
Check out the video below to learn how to configure the Create via Gateway Integration Template.

⚠️ This video assumes that you have already configured your repositories, models, and collections as outlined in the Quick Start Guide .

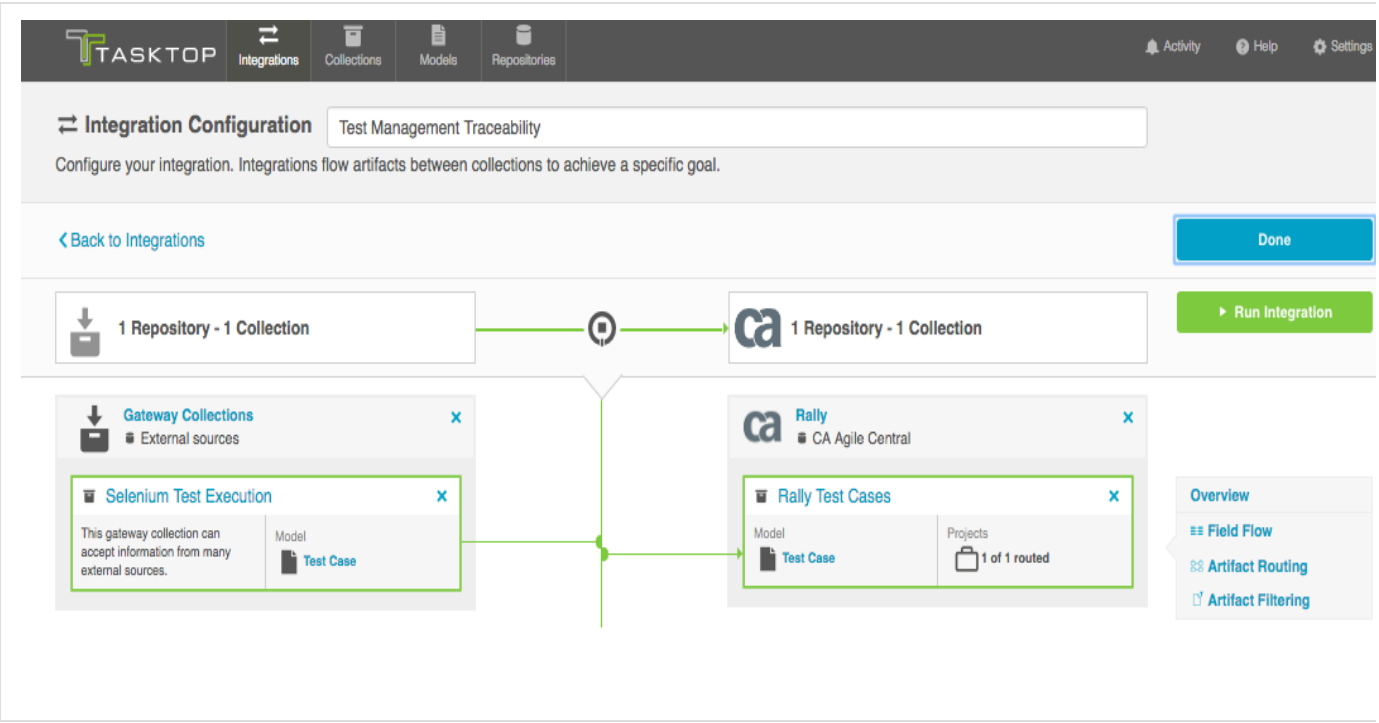
Use Case and Business Value

This integration creates traceability between artifacts across the software development lifecycle. New artifacts will be created in a repository collection when artifacts are sent to Tasktop via a Gateway collection. Optionally, these newly-created artifacts can be related to already-existing artifacts in the same repository.

For example, if your development team uses Gerrit for source code management and Serena Business Manager (SBM) for its agile story management but would like traceability between changesets in Gerrit and stories in SBM, you could set up an integration that would trigger the creation of changesets in SBM when changesets were created in Gerrit. And if the changesets in Gerrit identify the stories in SBM to which they pertain, Tasktop would find the already existing story in SBM and create a relationship between the two artifacts.



Additionally, if your QA team uses a tool like Selenium for test execution but Rally for test management, you can set up an integration that would trigger the creation test results in Rally when test results are created in Selenium. And if the test results from Selenium identify the tests in Rally which they cover, Tasktop would find the already-existing test and create a relationship between the two artifacts.



Template Affordances

The Create via Gateway Integration Template allows you to flow artifacts from a single gateway collection into a single repository collection. When a new artifact is sent to Tasktop via our REST API, an artifact will be created in the target repository collection

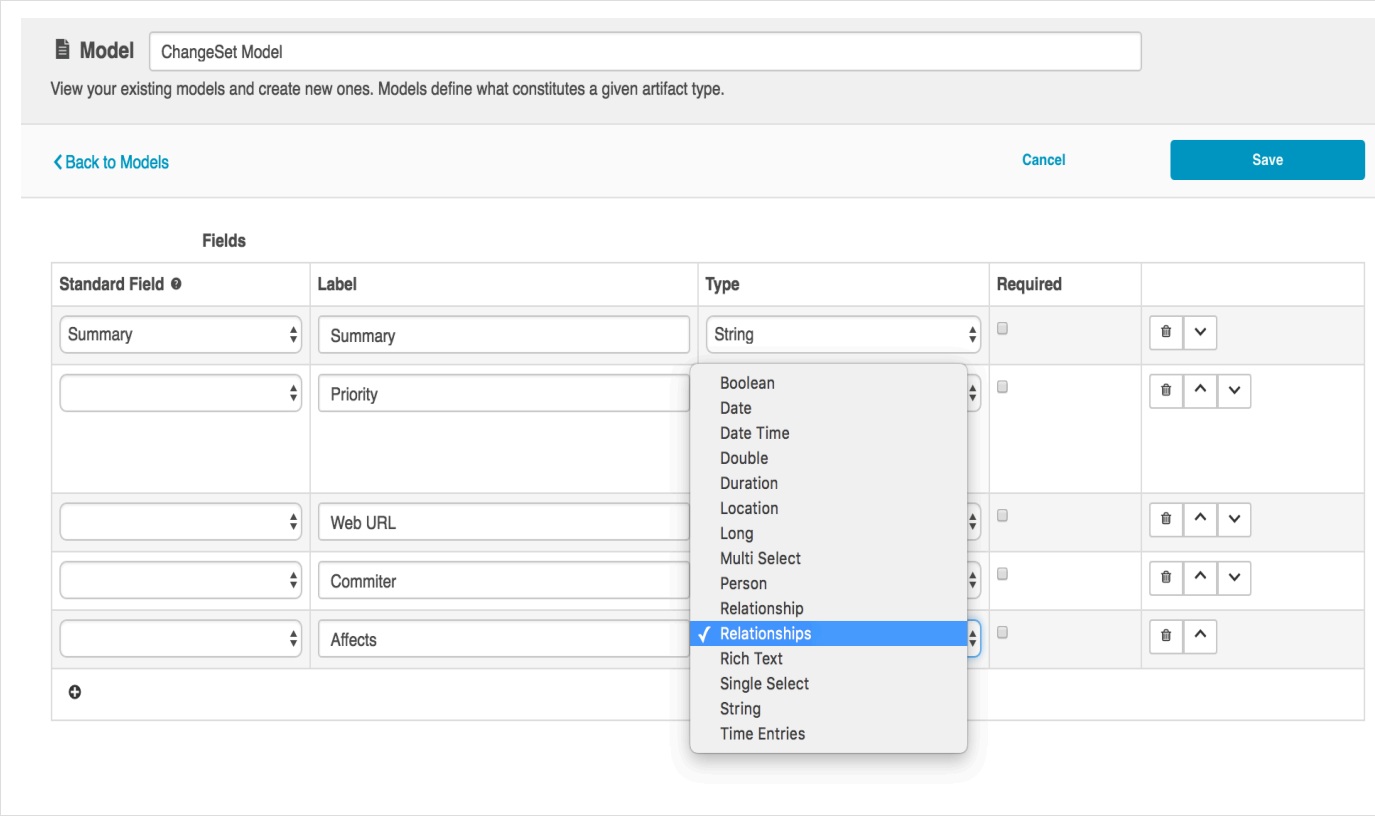
Template-Specific Configuration and Information

To create this integration, you can follow the general 4 steps described in the Quick Start Guide. However, there are some additional things you'll need to configure if you'd like to enable the optional part of this integration, the secondary act of creating a relationship between the new artifact created by Tasktop and an already-existing artifact in the same repository.

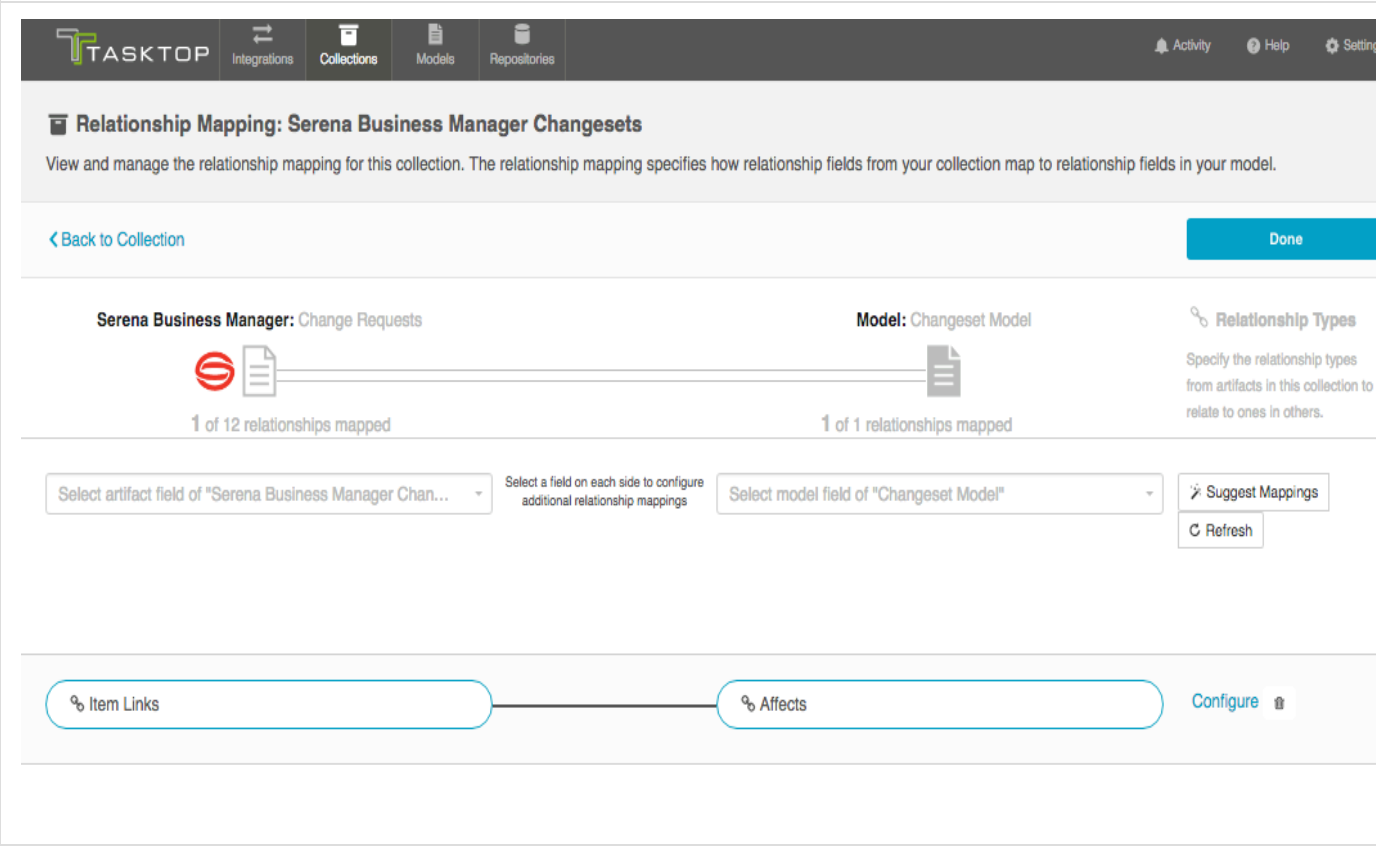
Model: When creating your model, you can create a field that is of type "relationship" or "relationships". You should use "Relationship" when the newly-created artifact can only relate to one other artifact and "relationships" when the newly-created artifact can relate to multiple artifacts.

One example of this is a relationship field type called "Parent"--this should generally be singular, as most artifacts usually only have a single parent. However, if the relationship field type is called "Blocks", it can likely be plural, as one artifact can block many artifacts.

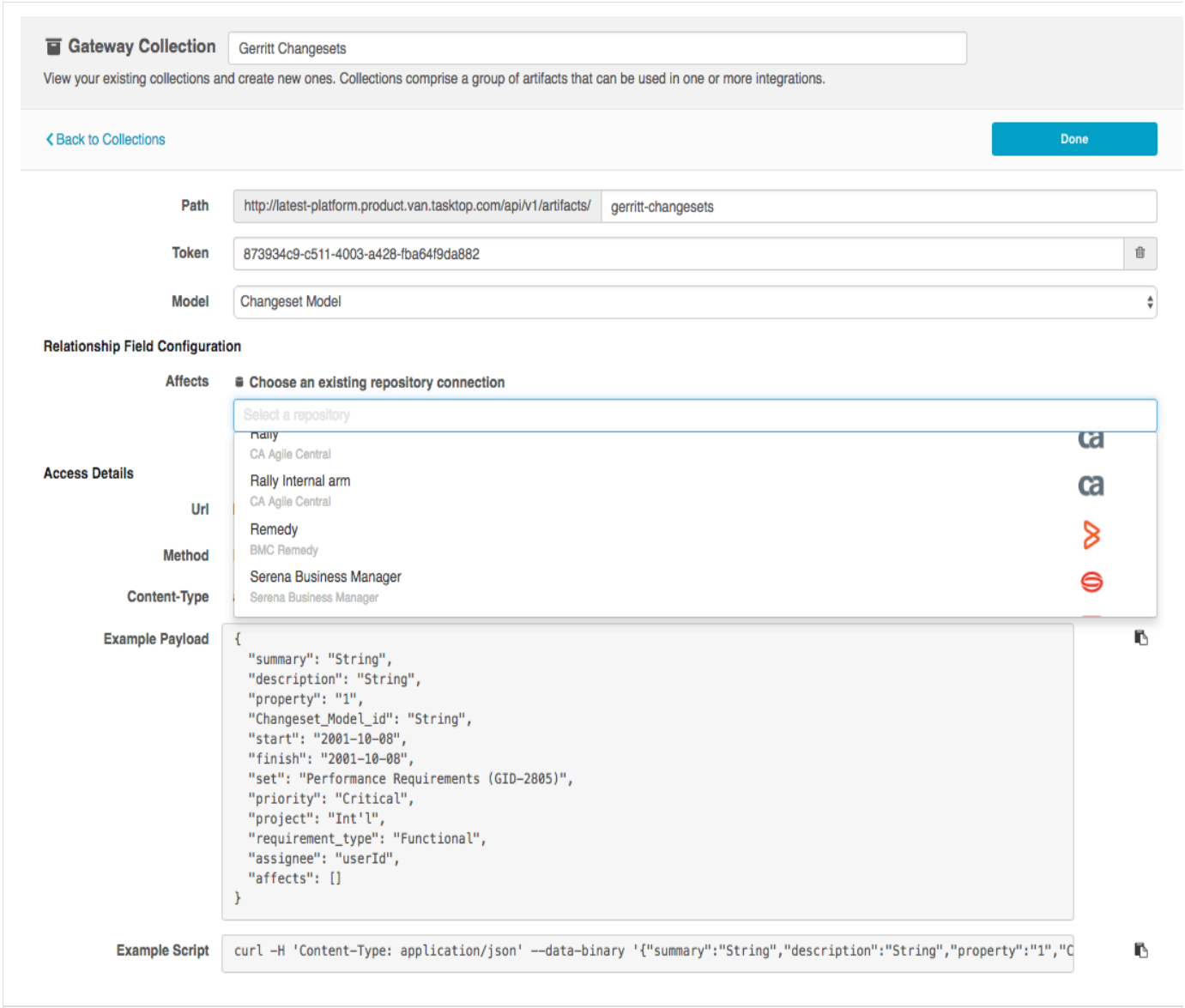
In the first use case example described above, I want the relationship to be "Affects" because any incoming changeset can affect many stories. So I'd configure a relationships field, plural.



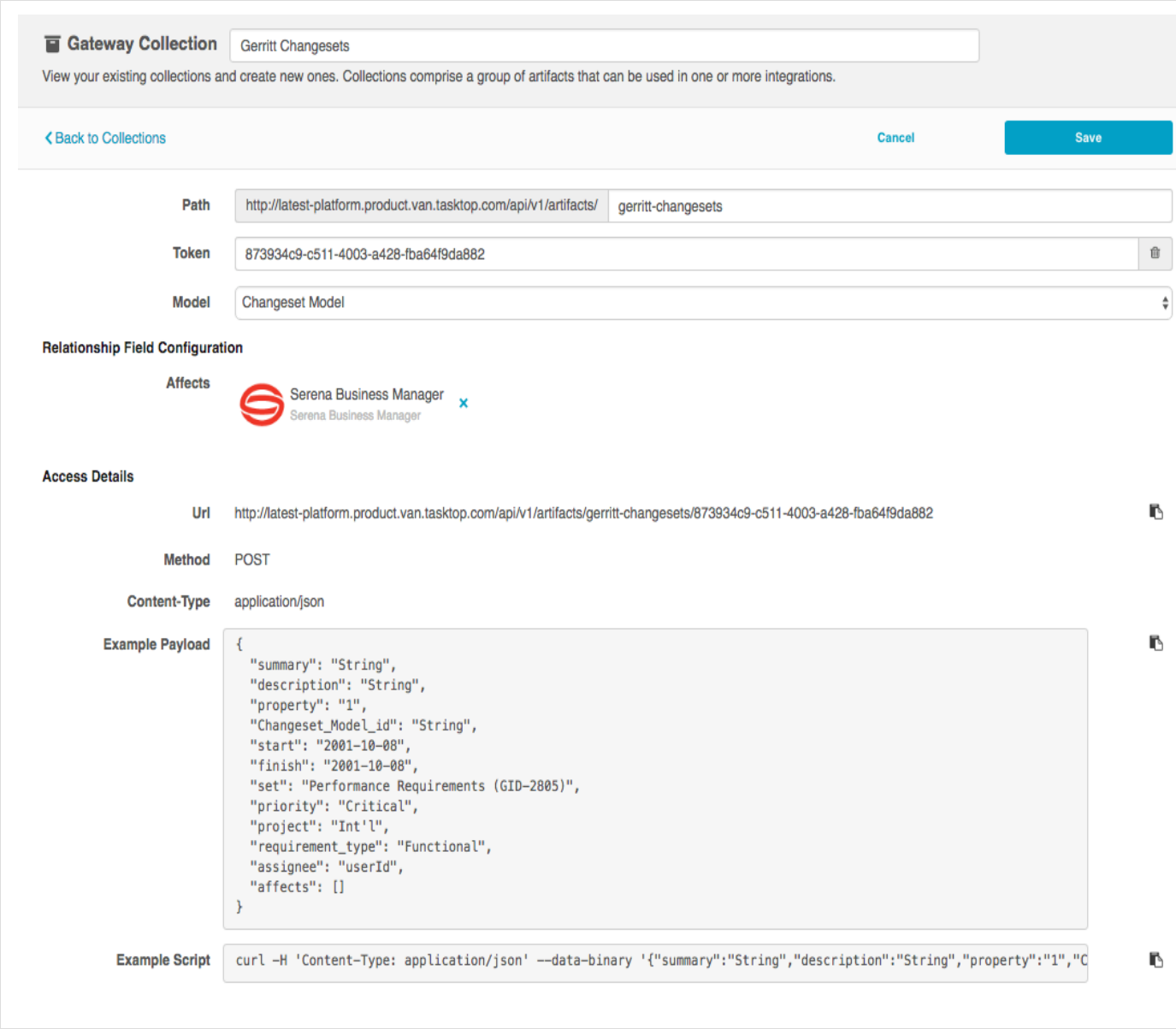
Repository Collection: When creating your repository collection, you will need to configure what relationship type available in the repository equates to the relationship field in the model. So, in the same example, if you wanted the relationship between the new changeset and the existing story to be "affects", but the relationship is actually called "items linked" in Serena, you would need to map those two fields. You'll need to do this for each relationship type configured in your model.



Gateway Collection: When creating your Gateway collection, you will see that for each model field that is a relationship you must specify the target repository that contains the related artifacts. Once this is selected, the example payload will describe what information is necessary to include in the payload so that Tasktop can find the artifact.


Gateway Collection
Gerritt Changesets
View your existing collections and create new ones. Collections comprise a group of artifacts that can be used in one or more integrations.
Back to Collections
Done
Path
http://latest-platform.product.van.tasktop.com/api/v1/artifacts/
gerritt-changesets
Token
873934c9-c511-4003-a428-fba64f9da882
Model
Changeset Model
Relationship Field Configuration
Affects
Choose an existing repository connection
Select a repository
CA Agile Central
Rally Internal arm
CA Agile Central
Remedy
BMC Remedy
Serena Business Manager
Serena Business Manager
Access Details
Url
Method
Content-Type
Example Payload
{
"summary": "String",
"description": "String",
"property": "1",
"Changeset_Model_id": "String",
"start": "2001-10-08",
"finish": "2001-10-08",
"set": "Performance Requirements (GID-2805)",
"priority": "Critical",
"project": "Int'l",
"requirement_type": "Functional",
"assignee": "userId",
"affects": []
}
Example Script
curl -H 'Content-Type: application/json' --data-binary '{"summary":"String","description":"String","property":"1","C

Selected:



Modify via Gateway Integration Template

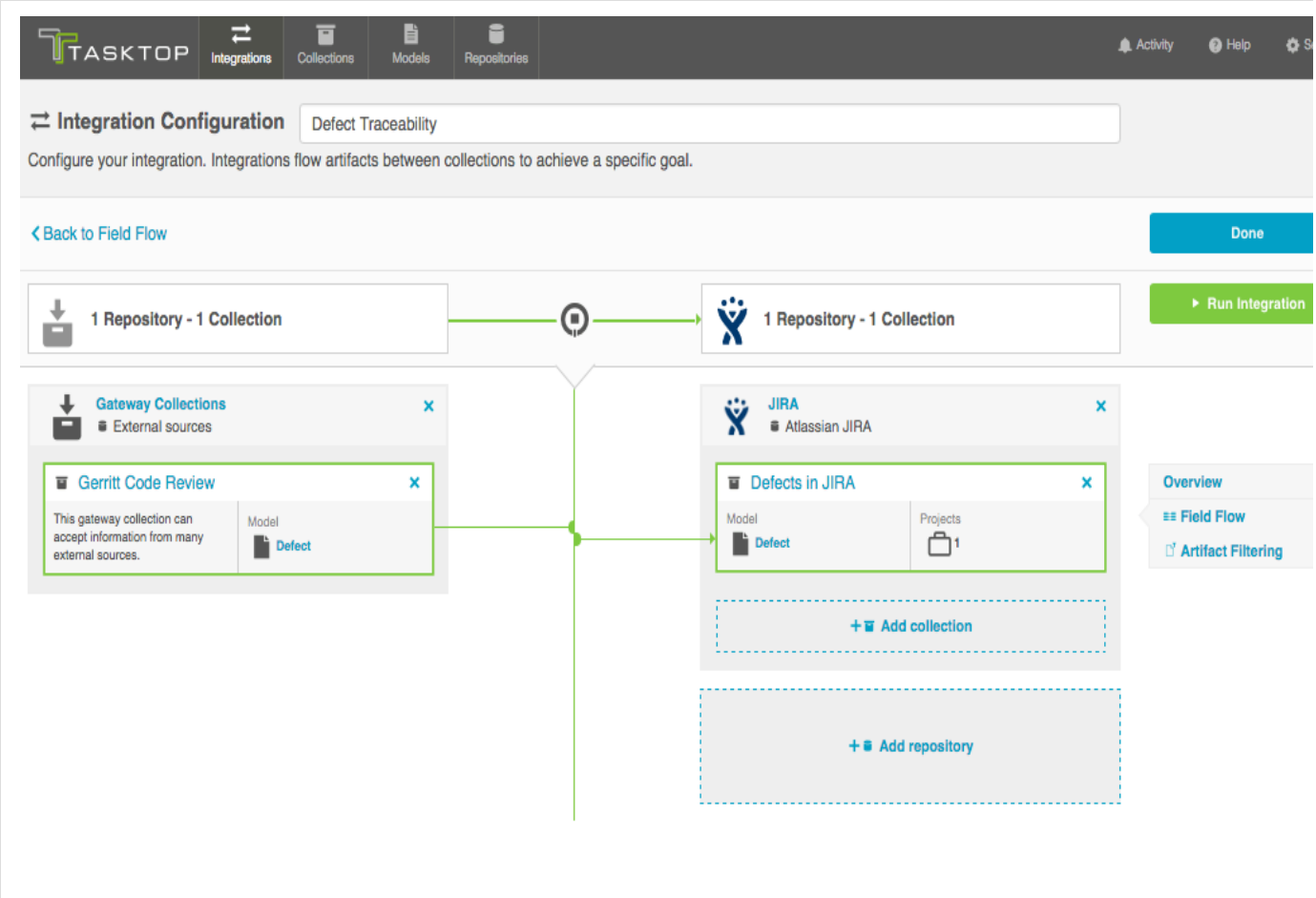
Tasktop: Apex Release (17.1)

Check out the video below to learn how to configure a Modify via Gateway Integration.

⚠️ This video assumes that you have already configured your repositories, models, and collections, as outlined in the Quick Start Guide.

Use Case and Business Value

This integration creates traceability between artifacts across the software development lifecycle. Already existing artifacts in a repository collection will be located and modified in a specified way when artifacts are sent to Tasktop via a Gateway collection.
For example,if your development team uses Gerrit for code review and uses JIRA for its agile work management but would like to know which defects in JIRA a given code review affects, or conversely which code reviews are associated with a given defect, you could set up an integration that would find an already-existing defect in JIRA anytime a code review is sent in and append one of it's fields with that code review's URL. The integration can even include updating other JIRA artifacts to which code reviews might pertain, such as stories and tech debt.



Template Affordances

The Modify via Gateway Integration Template allows you to update already-existing artifacts in target repository collections when artifacts are sent to Tasktop via a Gateway collection.

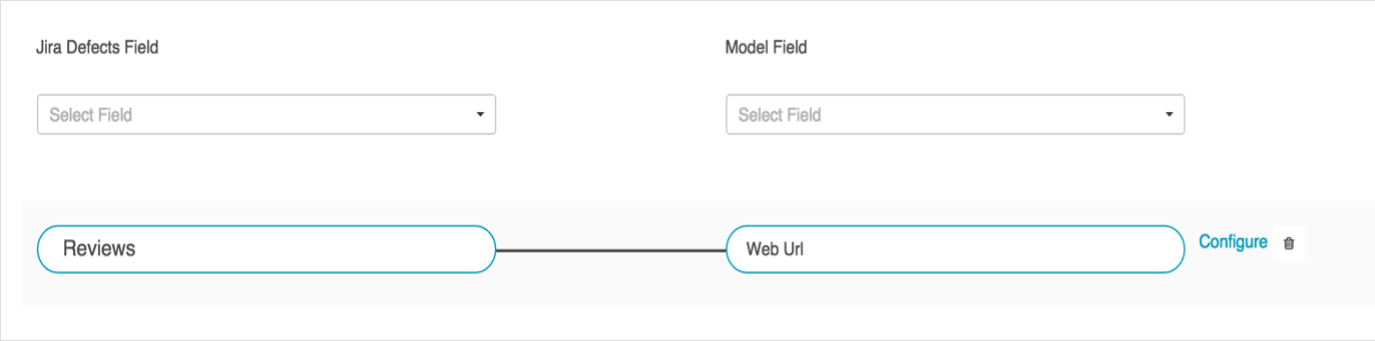
Template-Specific Configuration and Information

To create this integration, you can follow the general 4 steps described in the Quick Start Guide section. However, there are some additional steps you'll need to configure to run this integration.

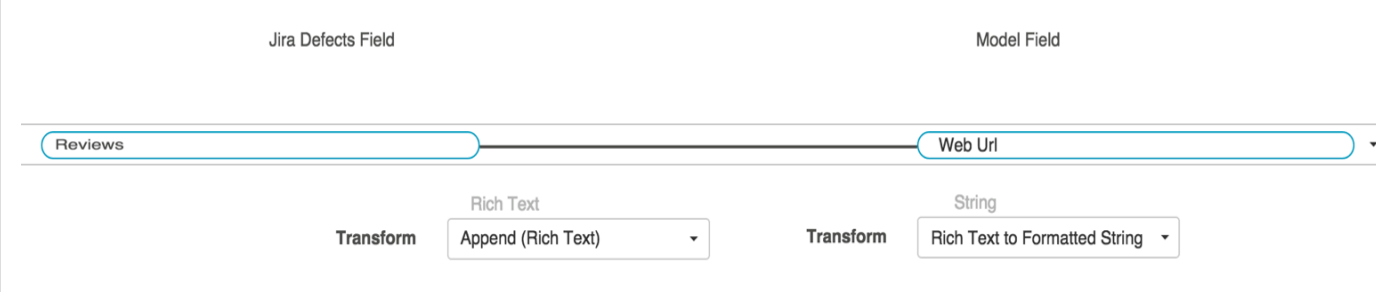
Proper Collection to Model Mapping and Transformation in Repository Collection

To specify just how you would like incoming artifacts from your gateway collection to modify already existing artifacts in your repository collection, you need to identify which field(s) on your already-existing artifacts you would like to modify and then configure how the field(s) should be changed. In the example above, the URL to any incoming code reviews from a gateway collection is being added to the review field of the JIRA defect.

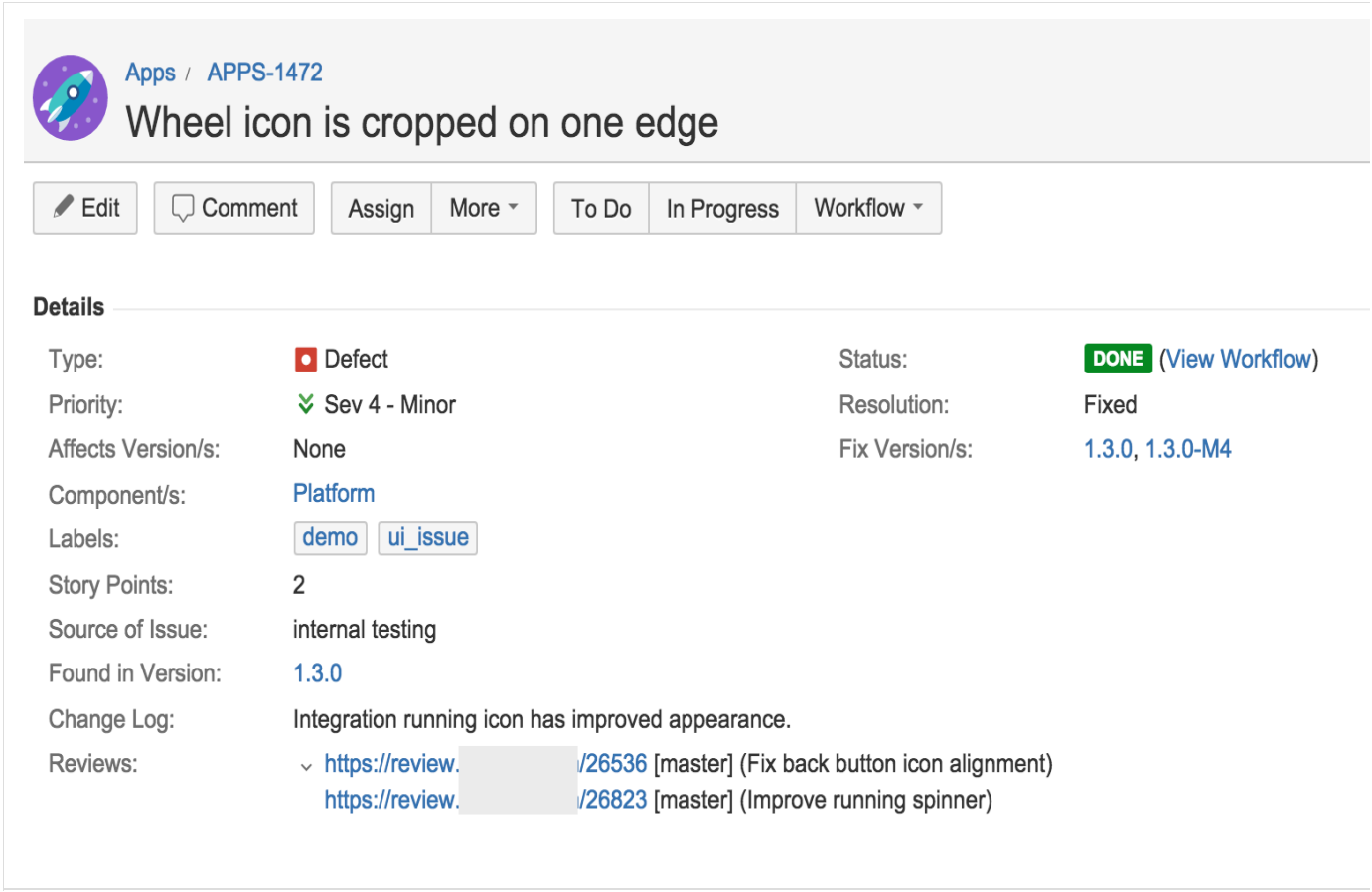
This means that the JIRA collection's collection to model mapping is configured as such. Here is the collection to model mapping:



And here are how the transformations are configured between these fields:

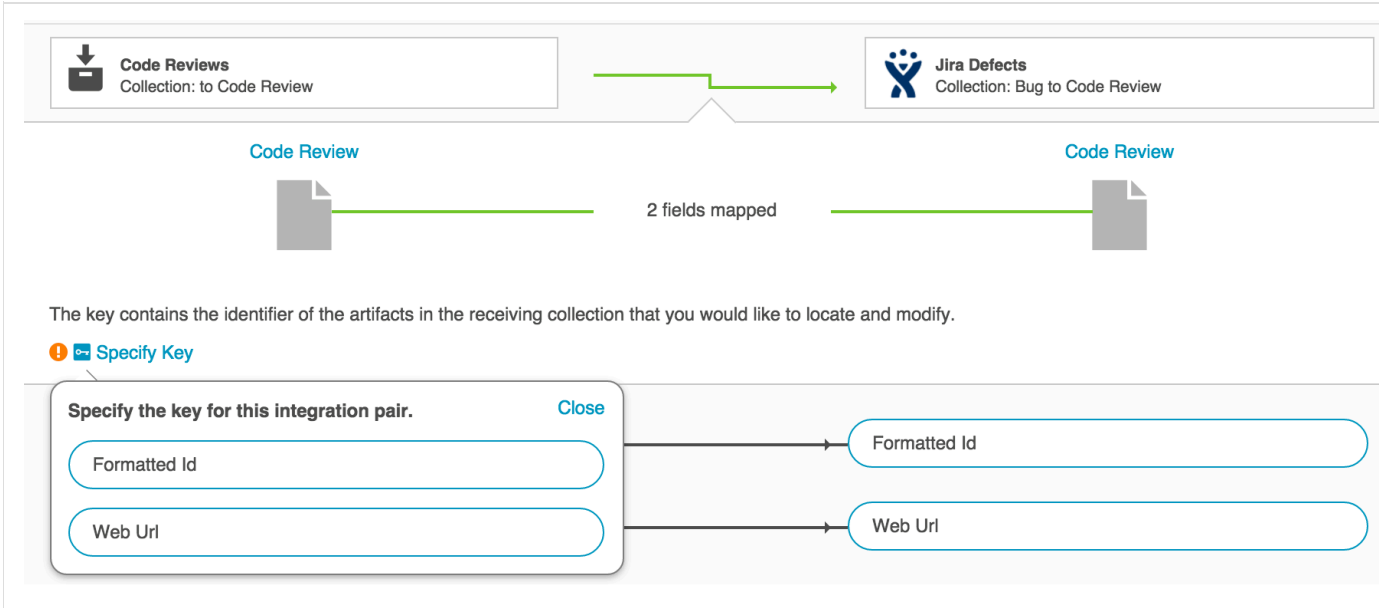


You can see that information coming in from the Gateway collection and through the model is appended to the JIRA reviews field, leaving the JIRA artifact itself looking like this:

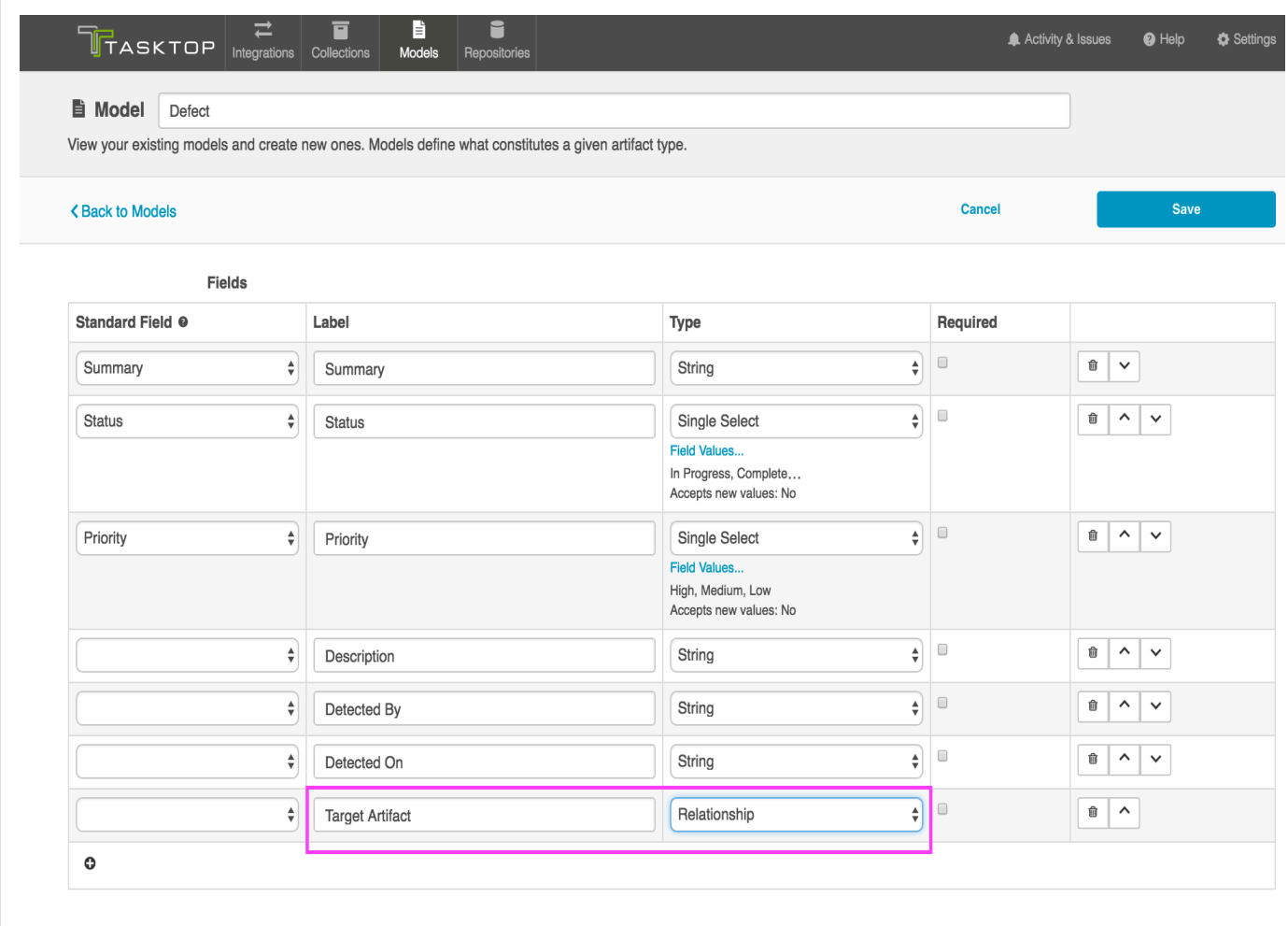


Integration Field Flow, Key ID

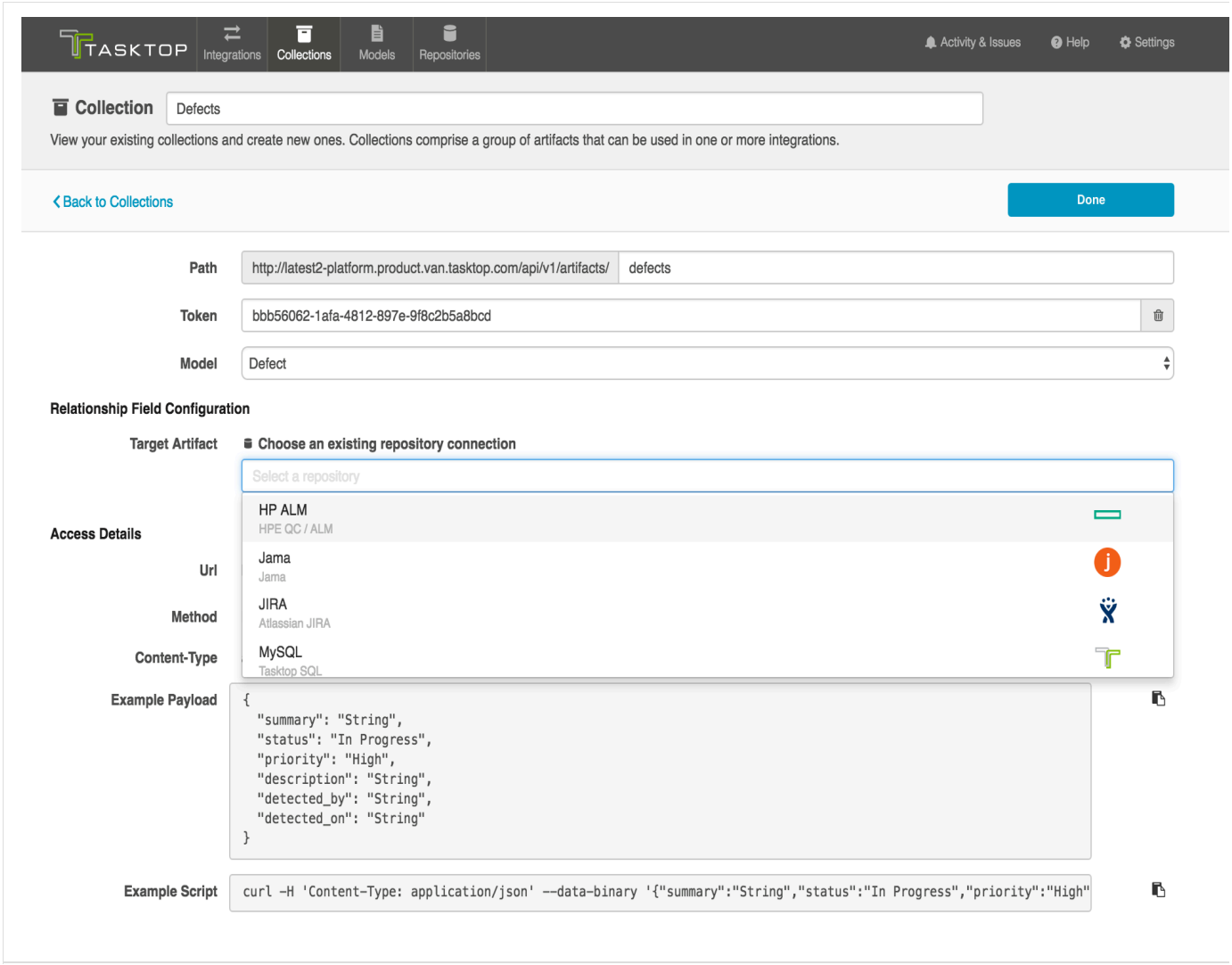
Additionally, in order for Tasktop to locate the already-existing artifact in your repository collection that will be modified as incoming artifacts are sent to Tasktop via your gateway collection, you must specify a key to locate the already-existing artifacts. The key contains the information, usually the ID of an artifact, that Tasktop needs to find the already existing artifact to relate the new artifact to. Picking the key for the integration is done on the field flow page by identifying the model field that holds the identifying information for existing artifacts.

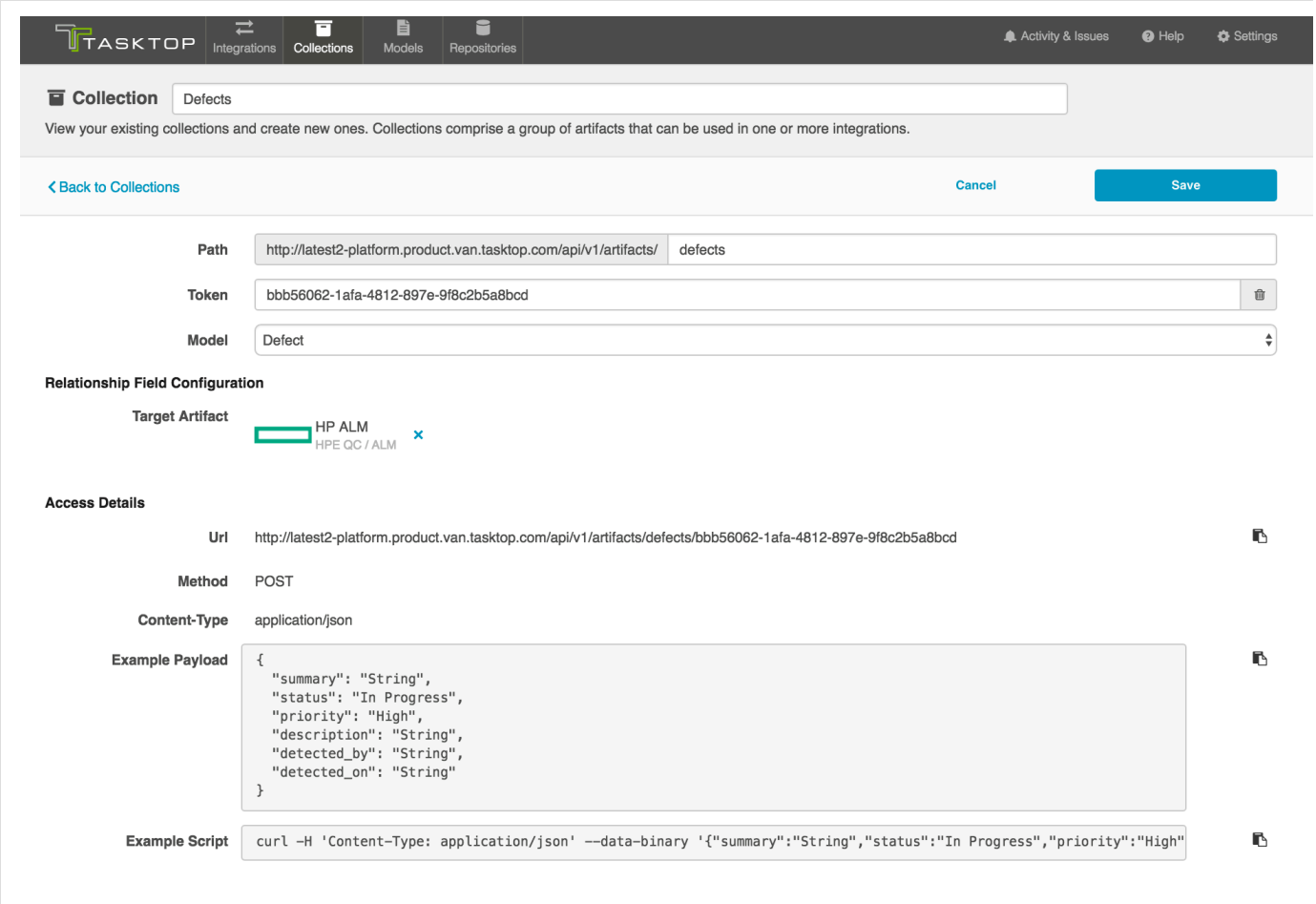


Add a field in your model of type RelationshipPlease note: some repositories require extra information in order to uniquely identify a single artifact across multiple projects. One prime example is HP. To ensure that enough information is sent in via your Gateway collection to allow Tasktop to find the specific artifact you would like to modify, please take these steps:

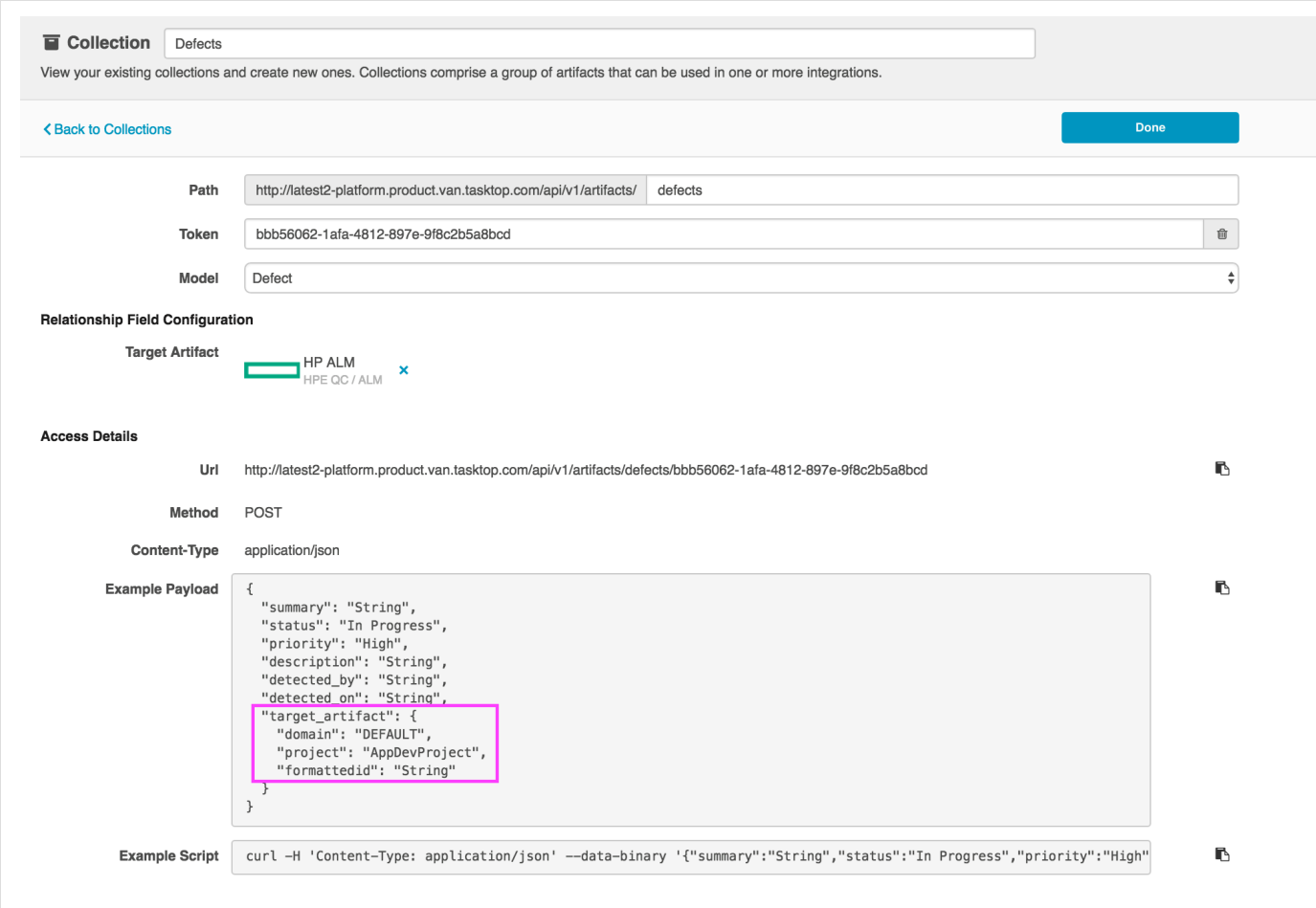


- In your Gateway collection, notice that for the new field you are prompted to pick a target repository. Select the repository you'd like to target in this Gateway integration.

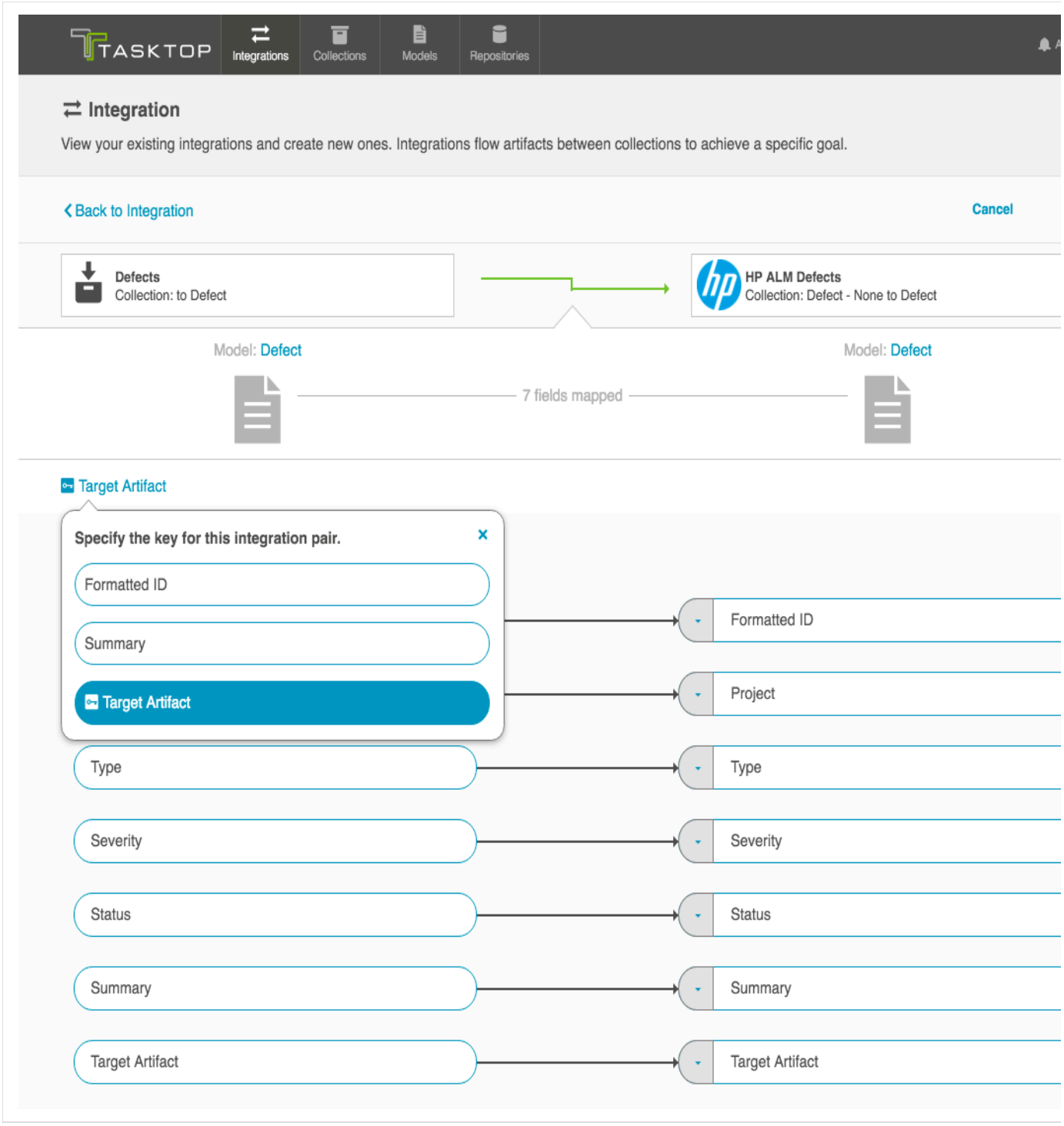

TASKTOP
Integrations
Collections
Models
Repositories
Activity & Issues
Help
Settings
Collection
Defects
View your existing collections and create new ones. Collections comprise a group of artifacts that can be used in one or more integrations.
Back to Collections
Done
Path
http://latest2-platform.product.van.tasktop.com/api/v1/artifacts/
defects
Token
bbb56062-1afa-4812-897e-9f8c2b5a8bcd
Model
Defect
Relationship Field Configuration
Target Artifact
Choose an existing repository connection
Select a repository
HP ALM
HPE QC / ALM
Jama
Jama
JIRA
Atlassian JIRA
MySQL
Tasktop SQL
Access Details
Url
Method
Content-Type
Example Payload
{
"summary": "String",
"status": "In Progress",
"priority": "High",
"description": "String",
"detected_by": "String",
"detected_on": "String"
}
Example Script
curl -H 'Content-Type: application/json' --data-binary '{"summary":"String","status":"In Progress","priority":"High"



- When you save, note that the example payload will be updated to include the pieces of information we need for that field to uniquely find artifacts

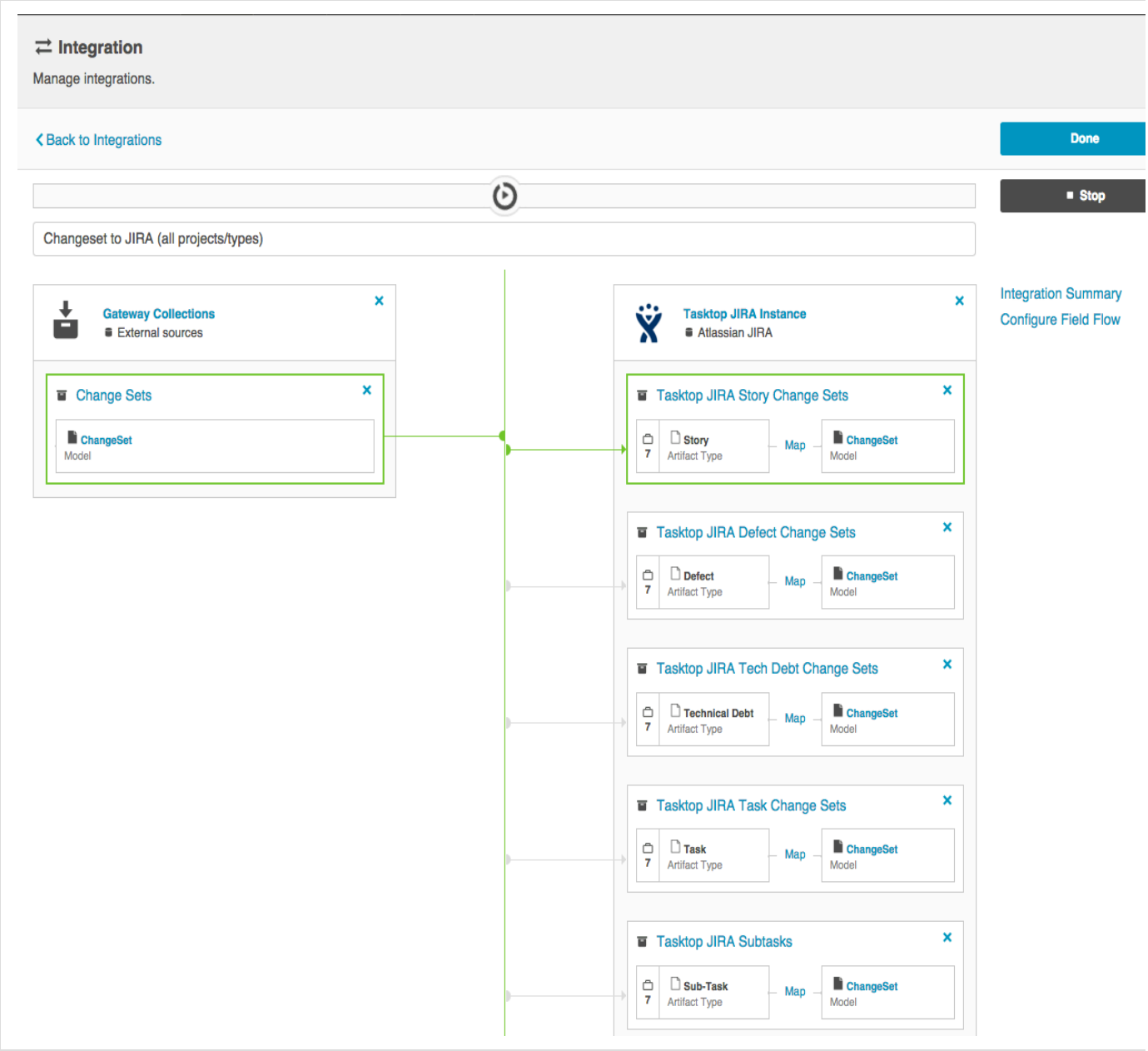


- Finally, in your integration select that field as your key on the field flow page.

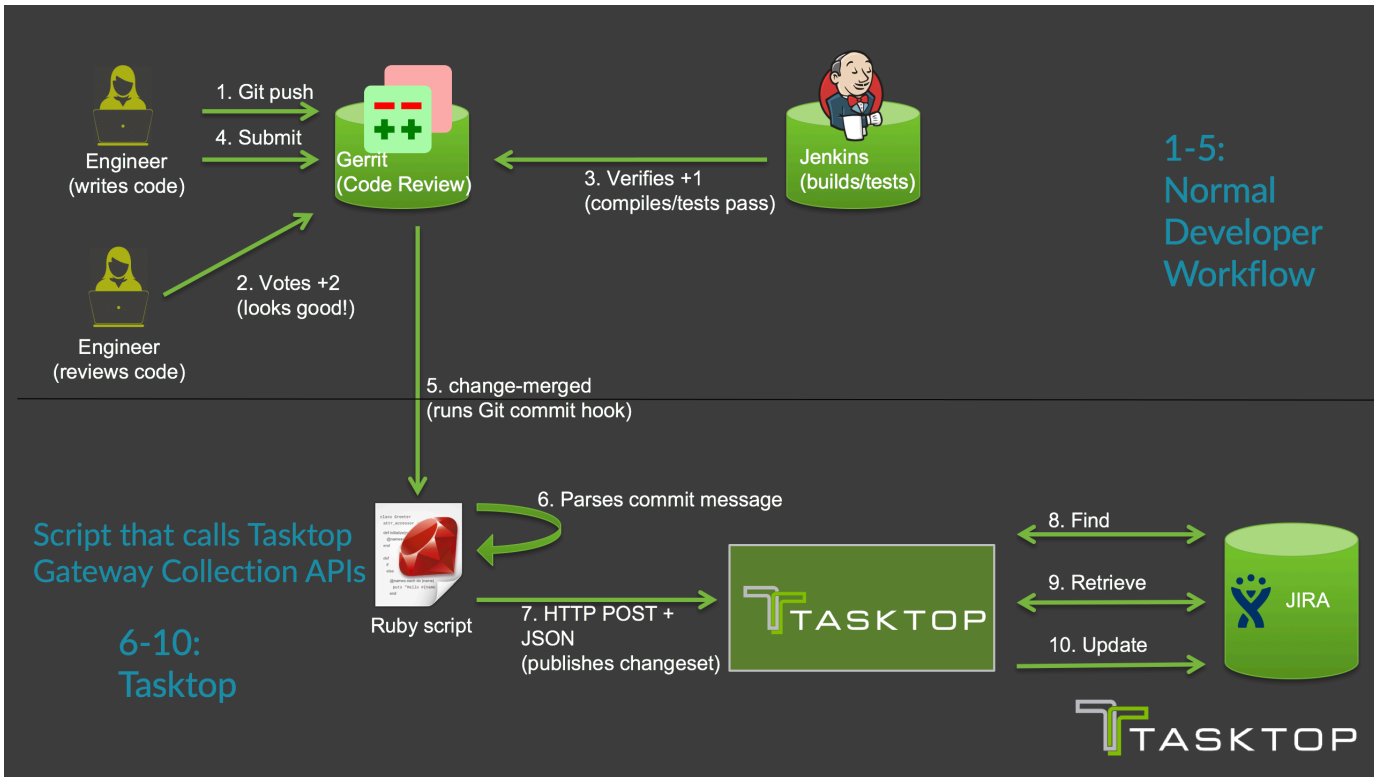


Example Use Case

This is an example of how we at Tasktop utilize the Modify via Gateway template. On the integration canvas, our integration, in which incoming changesets are modifying already-existing artifacts in JIRA, looks like this:



The image below illustrates just how the changeset is sent to Tasktop after the developers' normal workflow, at which point they then participate in the integration show above.



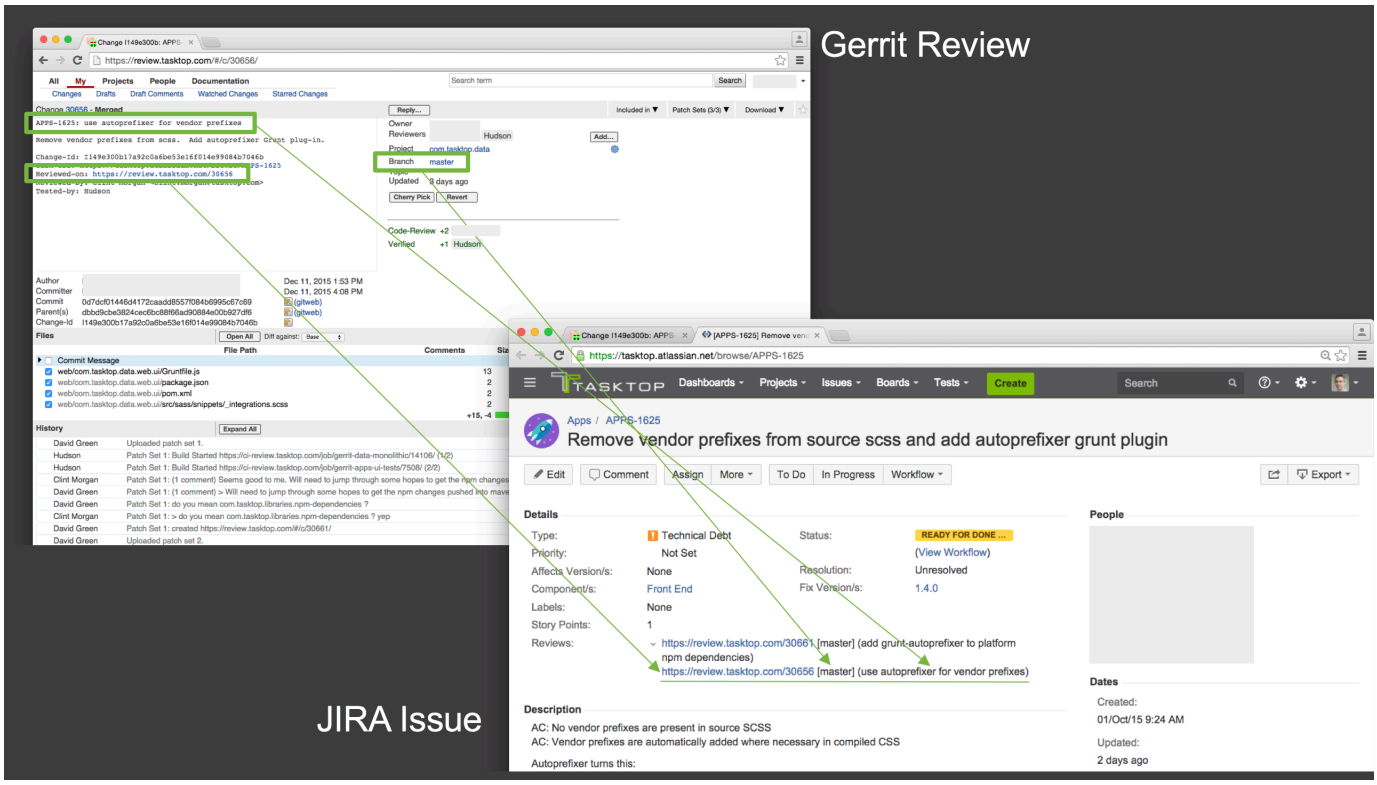
This is an example of the script that we use to automate the changesets being sent to Tasktop:

```
#!/usr/bin/ruby

require 'rubygems'
require 'logger'
require 'net/http'
require 'openssl'
require 'json'

def getOption(name)
  return ARGV[ARGV.index("--"+name)+1]
end

def sendToLink(data)
  request = Net::HTTP::Post.new(LINK_URL)
  request.body = JSON.generate(data)
  request.content_type = 'application/json'
  request.basic_auth "tasktop-platform", "tasktopSecret"
  uri = URI.parse(LINK_URL)
  response = Net::HTTP.start(uri.hostname, uri.port, :use_ssl => uri.scheme
== 'https', :verify_mode => OpenSSL::SSL::VERIFY_NONE) do |http|
      http.request(request)
  end
  if ! response.kind_of? Net::HTTPSuccess
    LOGGER.warn "Error sending to link: #{response.body}"
  end
end
```

```
LINK_URL = "https://tt-data350:8443/api/v1/artifacts/changesets"
TASK_ID_PATTERN =
/Task-Url:\s*https:\/\/tasktop.atlassian.net\/browse\/([^\s]*)/
REVIEW_URL_PATTERN = /.*Reviewed-on:\s+([^\s]*)/m
LOGGER =
Logger.new('/shared/gerrit/tasktop-site/logs/hook-change-merged.log','month
ly')
ENABLED_PROJECT_KEYS = ["APPS", "SYN", "SDK", "PLAT", "OPS", "CON", "DEV",
"QA", "RLIASE"]

project = getOption('project')
commit = getOption('commit')
branch = getOption('branch')

LOGGER.debug("Processing merge for commit #{commit} on project #{project}")

gitPath = ENV['GIT_DIR']
message = `git --git-dir #{gitPath} show -s --format=%B  #{commit}`
taskIdMatch = TASK_ID_PATTERN.match(message)
if taskIdMatch
  taskKey = taskIdMatch.captures[0]
  LOGGER.debug("Detected taskKey: #{taskKey}")
  taskKeyMatches = ENABLED_PROJECT_KEYS.any? { |project|
taskKey.start_with?(project + "-")}
  if ! taskKeyMatches
    LOGGER.info("#{taskKey} project not enabled, skipping");
    exit()
  end
  reviewUrlMatch = REVIEW_URL_PATTERN.match(message)
  webUrl = nil
  if reviewUrlMatch
    webUrl = reviewUrlMatch.captures[0]
  else
    LOGGER.error("Could not get webUrl from commit #{commit}")
    webUrl = "commit #{commit}"
  end
  firstLineOfMessage = message.lines.first.chomp
  firstLineOfMessage = firstLineOfMessage.gsub(/#{taskKey}:? /,'')
  sendToLink({"formatted_id" => taskKey, "info" => "#{webUrl} [#{branch}]
(#{firstLineOfMessage})"})
```

```
else
  LOGGER.debug("No task key found")
end
```

This image more clearly highlights how these changesets are reflected on the JIRA artifacts:



Enterprise Reporting Integration Template

Tasktop: Apex Release (17.1)

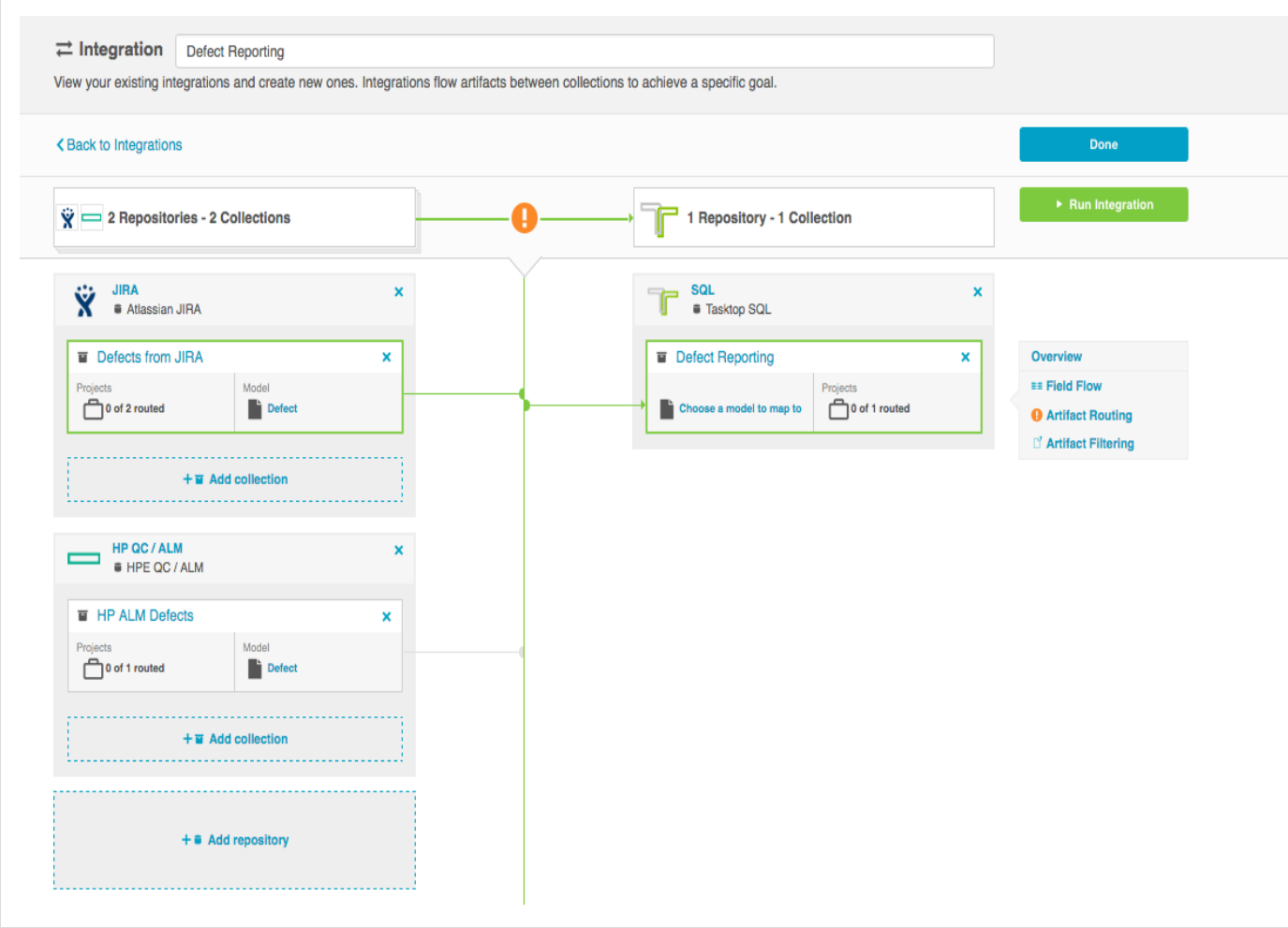
Check out the video below to learn how to configure an Enterprise Reporting Integration.

⚠️ This video assumes that you have already configured your repositories, models, and collections as outlined in the Quick Start Guide .

Use Case and Business Value

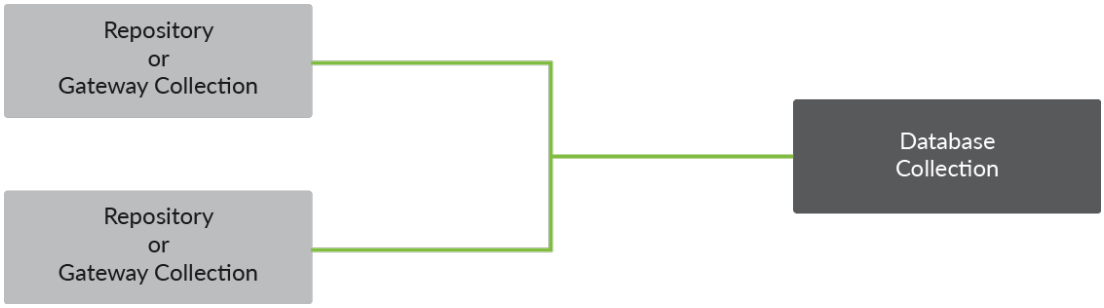
This integration simplifies enterprise reporting by unlocking software lifecycle data from its application tool silos and providing a rich data repository for near real-time analytics. Records will be created in a single database when artifacts from one or more collections are created or changed.

For example, if your organization uses multiple tools for defect discovery and resolution, such as Atlassian JIRA and HPE ALM, but would like to report on defects across both of the tools, you could set up an integration that would flow artifacts from your JIRA and HPE ALM collections into a single database table. You could then reporting directly from this aggregated table or, more likely, ETL it into your existing reporting infrastructure.



Template Affordances

The Enterprise Reporting Template allows you to flow artifacts from multiple repository collections and/or gateway collections into a single database collection.



Template-Specific Configuration and Information

Data Structures

A Reporting Integration populates a table with rows corresponding to the state of artifacts at a specific point in time. As an artifact changes, new rows are inserted corresponding to the new state of the artifact. The result is that each artifact has a series of rows corresponding to the state of the artifact at each point in time. The rows for all artifacts in a table can be thought of as an event stream.

Please note: Tasktop will examine your repositories for changes as specified in the polling interval that you have configured. This means that if you have configured the polling interval to be 1 minute, and a given artifact is changed twice in that minute, you'll only get a single record that reflects both changes.

The database table populated by the Reporting Integration has columns corresponding to fields in the artifact model, as well as some built-in fields that are designed to facilitate reporting. The following is an example of a database table corresponding to a simple Defect model:

```
CREATE TABLE `Defect` (
    `id` BIGINT (19) AUTO_INCREMENT,
    `formatted_id` VARCHAR (1000) NOT NULL,
    `project` VARCHAR (255) NOT NULL,
    `type` VARCHAR (255) NOT NULL,
    `severity` VARCHAR (255) NOT NULL,
    `status` VARCHAR (255) NOT NULL,
    `summary` VARCHAR (1000) NOT NULL,
    `repository_id` VARCHAR (255),
    `repository_url` VARCHAR (255),
    `artifact_id` VARCHAR (255),
    `artifact_url` VARCHAR (255),
    `artifact_event_type` VARCHAR (255),
    PRIMARY KEY (`id`)
);
```

Database Output

Default Information that Tasktop will Flow

The following columns represent information that will automatically be flowed to your database table.

Column	Description
`id*`	A surrogate key, can be used in reports to uniquely identify a row.
`repository_id*`	The unique identifier of the connection, can be used in reports to identify a repository connection.
`repository_url*`	The URL of the repository, can be used in reports to identify a repository.
`artifact_id*`	An id of an artifact that is globally unique, can be used in reports to uniquely identify an artifact across repositories and collections. The value of the `artifact_id` is an opaque value; assumptions should not be made about its structure or content. It should be noted that the `artifact_id` does not correspond to the id of the artifact as it is represented in the repository itself, but is useful for reporting since it is globally unique.
`artifact_url`	The URL of the artifact for browser access, can be used in reports to identify an artifact.
`artifact_event_type`	The type of event for the artifact that caused this entry. It can be used to see if the artifact has been added, changed or removed from the collection. |

*Denotes that this is a required field, meaning that your target database table will need to have a column to store this information.

💡 Note: If you use the Suggest DDL to create your table, all of the fields above will be included. If you are creating your table without that mechanism, you'll need to ensure that a column exists for the required pieces of information and, ideally, for the non-required fields as well. Your database table columns will need to be named as displayed above in either upper or lower case, but with the underscores as displayed.

Ordering of Rows

Though it may appear that rows in the table are inserted an order corresponding to the point in time that changes occurred, the order of rows in the table is not guaranteed. Reports should use a mapped field from the model (such as `modified`) to determine when a change occurred.

Artifact Event Type

In the artifact event type column of your database table, you'll see either "changed", "removed", or "filtered"

Changed

Changed indicates that either an existing artifact was changed or that a new artifact was added to your collection.

Removed

Removed indicates that a given artifact is in a project that has been removed from the collection. Here is a sample scenario to illustrate this event type:

In this reporting integration Project B and C are routed to the database table in my SQL collection at the start of an integration. Artifacts flow and records get written out:

Result Grid | Filter Rows: Search | Edit: | Export/Import:

id	formatted_id	project	type	created	modified	severity	status	summary	description	repository_id	repository_url	artifact_id	artifact_url	artifact_event
1	TPB-8	Test…	Bug	2016-…	2016-0…	Blocker	To Do	d33269d5e…	desc	c004d8cc-6…	http://ga-jira…	["com.ta…	http://ga-j…	changed
2	TPB-1	Test…	Bug	2015-…	2016-0…	Major	To Do	test bug B1	test bug B	c004d8cc-6…	http://ga-jira…	["com.ta…	http://ga-j…	changed
3	TPC-1	Test…	Bug	2015-…	2016-0…	Major	To Do	test bug C1	test bug C	c004d8cc-6…	http://ga-jira…	["com.ta…	http://ga-j…	changed

Project C is then removed from the source collection. At next full scan (one of the intervals configured on the Settings page), you'll see an event to denote that any artifacts in that collection have been removed:

100% 1:1

Result Grid | Filter Rows: Search | Edit: | Export/Import:

id	formatted_id	project	type	created	modified	severity	status	summary	description	repository_id	repository_url	artifact_id	artifact_url	artifact_event_ty…
1	TPB-8	Test…	Bug	2016-…	2016-0…	Blocker	To Do	d33269d5e…	desc	c004d8cc-6…	http://ga-jira…	["com.ta…	http://ga-j…	changed
2	TPB-1	Test…	Bug	2015-…	2016-0…	Major	To Do	test bug B1	test bug B	c004d8cc-6…	http://ga-jira…	["com.ta…	http://ga-j…	changed
3	TPC-1	Test…	Bug	2015-…	2016-0…	Major	To Do	test bug C1	test bug C	c004d8cc-6…	http://ga-jira…	["com.ta…	http://ga-j…	changed
4	TPC-1	Test…	Bug	2015-…	2016-0…	Major	To Do	test bug C1	test bug C	c004d8cc-6…	http://ga-jira…	["com.ta…	NULL	removed
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

⚠️ Note: If the project is added back to the collection and routed, records will not instantly be written out for all artifacts in that project; this will happen only when those artifacts change again.

Filtered

Though setting filters is meant to limit which artifacts flow in an integration, the impacts of setting filters on an enterprise reporting template are somewhat unique. Because it would not be ideal to have records in your database output that represent artifacts that have been filtered in an integration, given that these records would be stale and would not denote why a given artifact was not changing over time, it is the case that artifacts that are filtered of a reporting integration will still have records written out to the database but will have the "filtered' event type denoted.

Note the following:

- When you set a filter on an enterprise reporting integration, records will not automatically be written out for artifacts that do not meet filtering criteria. When artifacts that should be filtered out change, we'll then write out a record with the "filtered" event type.
- When a once filtered artifact field changes such that it now meets the filter criteria set, records will be written out right away.
- If you relax the filter and more artifacts are now in scope, the now in scope artifacts will only flow when the artifacts themselves change again.
- If an artifact is filtered out of the reporting integration, and then its project is removed from the collection, records will be written out for all artifacts in that collection at next full scan and marked as "removed", whether or not they have been filtered out of the integration (This effectively means that the "removed" designation supersedes "filtered" designation.)
    - If you add the project back to the collection and routed in the integration, changes to artifacts will create a new record with either the "changed" or "filtered" event type, depending on whether or not the artifact meets the filter criteria.

To ETL or Not To ETL?

ETL (Extract, Transform, Load) is a process where data is extracted from a database, transformed to be more suitable for reporting or analytics, and loaded into a database which is normally used for reporting.

The data structures populated directly by Tasktop are intended to be used as a source for ETL; Some kinds of reports are not easily produced without first performing an ETL process. ETL can also be beneficial for performance of reports.

Some reports are possible without first performing an ETL process. Examples of such reports include Artifact Cycle Time and Defect Count By State By Cycle Time.

Example Reports

Following are examples of some reports that can be driven directly from the database tables populated by a Reporting Integration:

Artifact Cycle Time

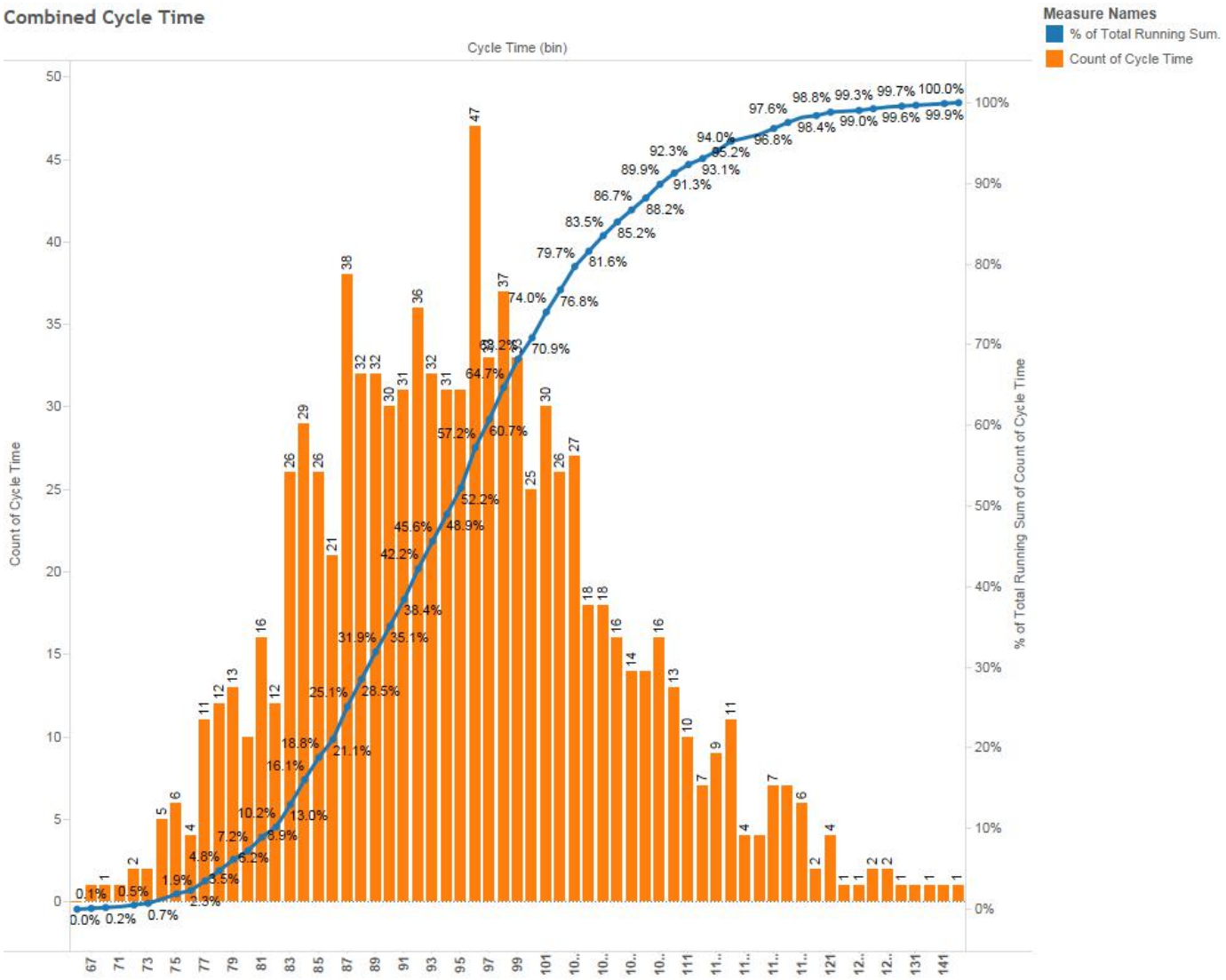
Artifact Cycle Time is often a valuable metric to measure as it can help identify areas where efficiencies can be gained and ensure “lean flow”. We have provided a model called “Artifact Cycle Time” and can be used to easily flow the necessary data to your database – enabling you to create a variety of metrics and visualizations based on the cycle time of any artifact type.

Artifact Cycle Time Model

Artifact Cycle Time
Formatted ID
Project
Type
Created
Modified
Severity
Status
Priority
Release
Assignee

If you use this model, you can easily produce visualizations such as a histogram that can identify the historical trend of cycle times.

Artifact Cycle Time Histogram



SQL

```
SELECT A.FORMATTED_ID, B.MODIFIED AS StatusOpen, C.MODIFIED AS
StatusInProgress, D.MODIFIED AS StatusReadyForTesting, E.MODIFIED AS
StatusReadyForVerification, F.MODIFIED AS StatusComplete, G.MODIFIED AS
StatusShipped, A.STATUS AS CurrentStatus FROM ARTIFACT A
  LEFT OUTER JOIN ARTIFACT B
    ON B.ARTIFACT_ID = A.ARTIFACT_ID
    AND B.STATUS = 'Open'
    AND NOT EXISTS (SELECT * FROM ARTIFACT WHERE ARTIFACT_ID =
A.ARTIFACT_ID AND (MODIFIED < B.MODIFIED OR (MODIFIED = B.MODIFIED AND
ID < B.ID)) AND STATUS = B.STATUS)
  LEFT OUTER JOIN ARTIFACT C
    ON C.ARTIFACT_ID = A.ARTIFACT_ID
    AND C.STATUS = 'In Progress'
    AND NOT EXISTS (SELECT * FROM ARTIFACT WHERE ARTIFACT_ID =
A.ARTIFACT_ID AND (MODIFIED < C.MODIFIED OR (MODIFIED = C.MODIFIED AND
ID < C.ID)) AND STATUS = C.STATUS)
  LEFT OUTER JOIN ARTIFACT D
```

```
    ON D.ARTIFACT_ID = A.ARTIFACT_ID
    AND D.STATUS = 'Ready for Testing'
    AND D.MODIFIED > (SELECT MAX(MODIFIED) FROM ARTIFACT WHERE
ARTIFACT_ID = A.ARTIFACT_ID AND STATUS IN ('Open', 'In Progress'))
    AND NOT EXISTS (SELECT * FROM ARTIFACT WHERE ARTIFACT_ID =
A.ARTIFACT_ID AND (MODIFIED < D.MODIFIED OR (MODIFIED = D.MODIFIED AND
ID < D.ID)) AND STATUS = D.STATUS
      AND MODIFIED > (SELECT MAX(MODIFIED) FROM ARTIFACT WHERE
ARTIFACT_ID = A.ARTIFACT_ID AND STATUS IN ('Open',  'In Progress')))
  LEFT OUTER JOIN ARTIFACT E
    ON E.ARTIFACT_ID = A.ARTIFACT_ID
    AND E.STATUS = 'Ready for Verification'
    AND E.MODIFIED > (SELECT MAX(MODIFIED) FROM ARTIFACT WHERE
ARTIFACT_ID = A.ARTIFACT_ID AND STATUS IN ('Open', 'In Progress', 'Ready
for Testing'))
    AND NOT EXISTS (SELECT * FROM ARTIFACT WHERE ARTIFACT_ID =
A.ARTIFACT_ID AND (MODIFIED < E.MODIFIED OR (MODIFIED = E.MODIFIED AND
ID < E.ID)) AND STATUS = E.STATUS
      AND MODIFIED > (SELECT MAX(MODIFIED) FROM ARTIFACT WHERE
ARTIFACT_ID = A.ARTIFACT_ID AND STATUS IN ('Open',  'In Progress',
'Ready for Testing')))
  LEFT OUTER JOIN ARTIFACT F
    ON F.ARTIFACT_ID = A.ARTIFACT_ID
    AND F.STATUS = 'Complete'
    AND F.MODIFIED > (SELECT MAX(MODIFIED) FROM ARTIFACT WHERE
ARTIFACT_ID = A.ARTIFACT_ID AND STATUS IN ('Open', 'Ready for Testing',
'Ready for Verification'))
    AND NOT EXISTS (SELECT * FROM ARTIFACT WHERE ARTIFACT_ID =
A.ARTIFACT_ID AND (MODIFIED < F.MODIFIED OR (MODIFIED = F.MODIFIED AND
ID < F.ID)) AND STATUS = F.STATUS
      AND MODIFIED > (SELECT MAX(MODIFIED) FROM ARTIFACT WHERE
ARTIFACT_ID = A.ARTIFACT_ID AND STATUS IN ('Open', 'Ready for Testing',
'Ready for Verification')))
  LEFT OUTER JOIN ARTIFACT G
    ON G.ARTIFACT_ID = A.ARTIFACT_ID
    AND G.STATUS = 'Shipped'
    AND G.MODIFIED > (SELECT MAX(MODIFIED) FROM ARTIFACT WHERE
ARTIFACT_ID = A.ARTIFACT_ID AND STATUS IN ('Open', 'Ready for Testing',
'Ready for Verification', 'Complete'))
    AND NOT EXISTS (SELECT * FROM ARTIFACT WHERE ARTIFACT_ID =
A.ARTIFACT_ID AND (MODIFIED < G.MODIFIED OR (MODIFIED = G.MODIFIED AND
ID < G.ID)) AND STATUS = G.STATUS
      AND MODIFIED > (SELECT MAX(MODIFIED) FROM ARTIFACT WHERE
ARTIFACT_ID = A.ARTIFACT_ID AND STATUS IN ('Open', 'Ready for Testing',
'Ready for Verification', 'Complete')))
WHERE NOT EXISTS (SELECT * FROM ARTIFACT WHERE ARTIFACT_ID =
A.ARTIFACT_ID AND (MODIFIED > A.MODIFIED OR (MODIFIED = A.MODIFIED AND
ID > A.ID)))
```

```
  AND (A.ARTIFACT_EVENT_TYPE IS NULL OR NOT A.ARTIFACT_EVENT_TYPE =
'removed')
ORDER BY A.FORMATTED_ID
```

The example above is designed to handle cases where an artifact is moved into a state more than once. For example, a defect that is moved to “Complete”, subsequently moved back into “In Progress”, then moved to “Complete” again is represented with a row having the second timestamp for the “Complete” status.

formatted_id	priority	type	severity	repository_url	StatusOpen	StatusInProgress	StatusReadyForTesting	StatusReadyForVerification	StatusComplete	StatusShipped	CurrentStatus
TCK2-Test-Project1	Low	Defect	3	HP ALM	2015-01-20 03:40:00	2015-01-24 23:10:00	2015-02-02 00:52:00	2015-04-27 03:09:00	2015-05-14 04:50:00	2015-05-23 06:37:00	Shipped
TCK2-Test-Project10	Medium	Defect	2	HP ALM	2015-02-05 01:22:00	2015-02-12 17:41:00	2015-02-24 22:13:00	2015-03-17 01:12:00	2015-03-30 17:58:00	2015-04-05 17:46:00	Shipped
TCK2-Test-Project11	High	Defect	1	HP ALM	2015-02-06 09:49:00	2015-02-13 10:00:00	2015-02-26 15:28:00	2015-03-09 08:46:00	2015-03-18 11:00:00	2015-03-27 15:58:00	Shipped
TCK2-Test-Project12	Low	Defect	3	JIRA	2015-02-06 14:09:00	2015-02-10 12:49:00	2015-02-27 12:26:00	2015-03-17 14:08:00	2015-03-29 11:39:00	2015-04-06 14:08:00	Shipped
TCK2-Test-Project13	Medium	Defect	2	HP ALM	2015-02-07 19:44:00	2015-02-11 21:33:00	2015-03-13 02:41:00	2015-04-15 19:54:00	2015-05-08 23:00:00	2015-05-17 18:06:00	Shipped
TCK2-Test-Project14	High	Defect	1	HP ALM	2015-02-07 18:43:00	2015-02-10 22:00:00	2015-02-18 22:23:00	2015-02-25 02:19:00	2015-03-07 01:31:00	2015-03-17 18:47:00	Shipped
TCK2-Test-Project15	High	Defect	1	HP ALM	2015-02-12 19:09:00	2015-03-05 18:22:00	2015-04-02 20:03:00	2015-05-01 18:26:00	2015-05-12 14:00:00	2015-05-20 19:48:00	Shipped
TCK2-Test-Project16	High	Defect	1	HP ALM	2015-02-15 19:00:00	2015-02-19 20:42:00	2015-03-14 20:20:00	2015-03-21 22:42:00	2015-04-04 17:16:00	2015-04-04 22:52:00	Shipped

Reports can be driven from the results of this SQL query, subtracting dates to produce cycle times for the desired transitions (e.g. “Open” to “Shipped”).

Status values in the SQL above correspond to the values present in the “Artifact” model; repository-specific status values can be mapped to the model values in the corresponding Collection mapping. If status values are added, removed or changed in the Artifact model, then the SQL will have to be modified accordingly.

Defect Count By State By Cycle Time

Defect Count By State By Cycle Time provides a count of defects by cycle time for each status of an artifact.

In this example, the cycle time is measured in days. Cycle time is only measured for status state transitions; Cycle time is not measured for the end state of an artifact.

We provide a basic defect model packaged with our product:

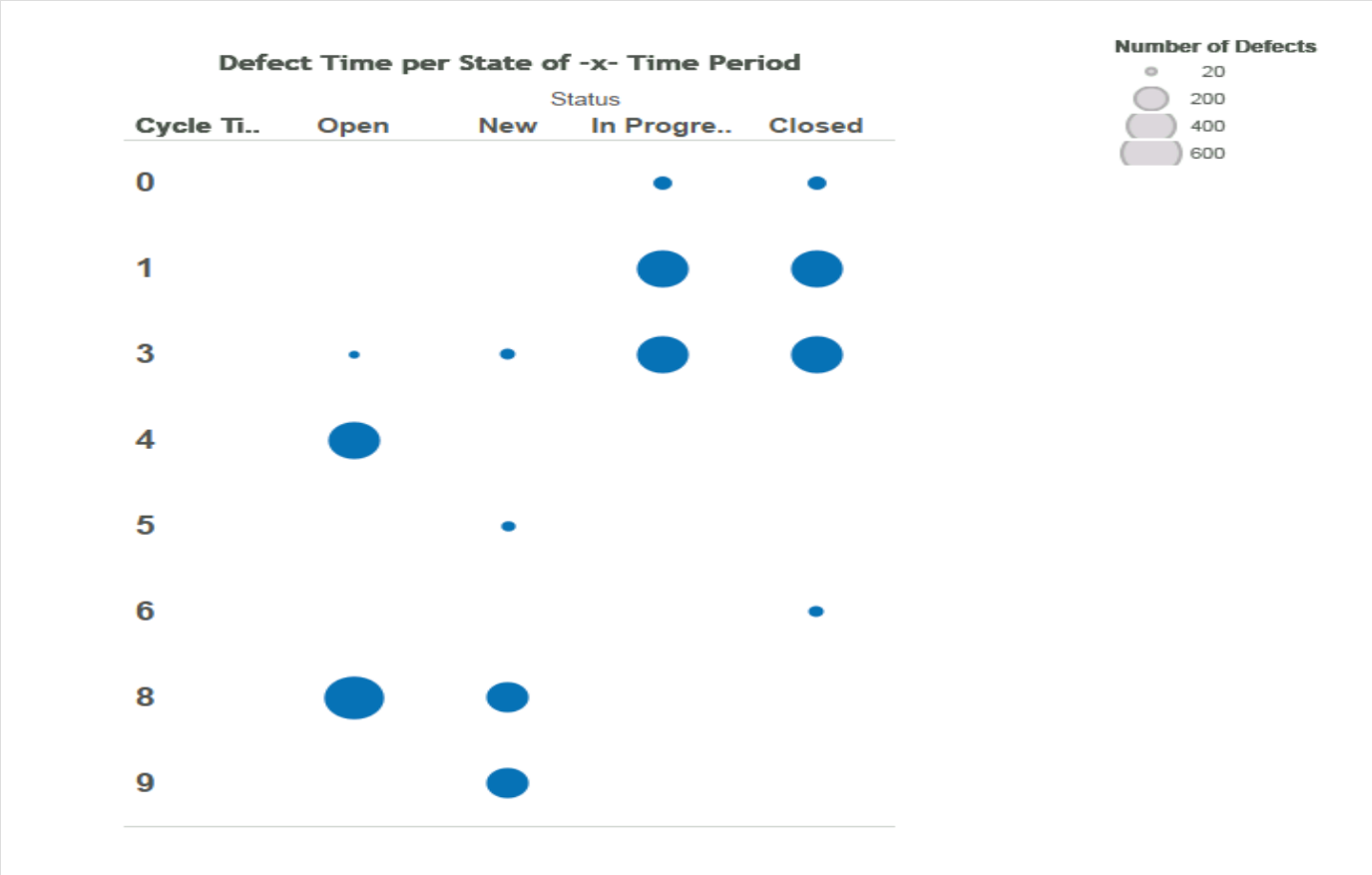
Basic Defect Model

Defect Model
Formatted ID
Project
Type
Created
Modified
Severity
Status
Summary

Summary-to-Description
Related Defects
Description

If you use this model, you can easily produce visualizations such as a bubble chart that can identify the volume of defects in each cycle time measured in days. This is simply a slightly different view into your overall cycle time.

Cycle Time Volume



SQL

```
SELECT status, COUNT(artifact_id), cycleTime FROM (
  SELECT A.ARTIFACT_ID AS artifact_id, A.STATUS AS status, SUM(
TIMESTAMPDIFF(SQL_TSI_DAY,A.MODIFIED,B.MODIFIED) ) AS cycleTime FROM
DEFECT A
    INNER JOIN DEFECT B ON A.ARTIFACT_ID = B.ARTIFACT_ID
      AND A.ID != B.ID
      AND A.STATUS != B.STATUS
      AND A.MODIFIED <= B.MODIFIED
      AND ((A.ARTIFACT_EVENT_TYPE IS NULL OR B.ARTIFACT_EVENT_TYPE IS
NULL)
        OR NOT (A.ARTIFACT_EVENT_TYPE = 'removed' OR
B.ARTIFACT_EVENT_TYPE = 'removed')
      )
    WHERE NOT EXISTS (
      SELECT * FROM DEFECT C WHERE C.ARTIFACT_ID = A.ARTIFACT_ID AND
C.ID != A.ID AND C.ID != B.ID
        AND C.MODIFIED >= A.MODIFIED AND C.MODIFIED <= B.MODIFIED
        AND ((C.STATUS = A.STATUS OR C.STATUS = B.STATUS) OR (C.STATUS
!= A.STATUS AND C.STATUS != B.STATUS))
      )
      AND NOT EXISTS (
        SELECT * FROM DEFECT D WHERE D.ARTIFACT_ID = A.ARTIFACT_ID AND
B.MODIFIED <= (
          SELECT MAX(MODIFIED) FROM DEFECT D WHERE D.ARTIFACT_ID =
A.ARTIFACT_ID AND D.ARTIFACT_EVENT_TYPE = 'removed'
        )
      )
  GROUP BY A.ARTIFACT_ID, A.STATUS
) CT GROUP BY CT.status, CT.cycleTime
ORDER BY CT.status, CT.cycleTime
```

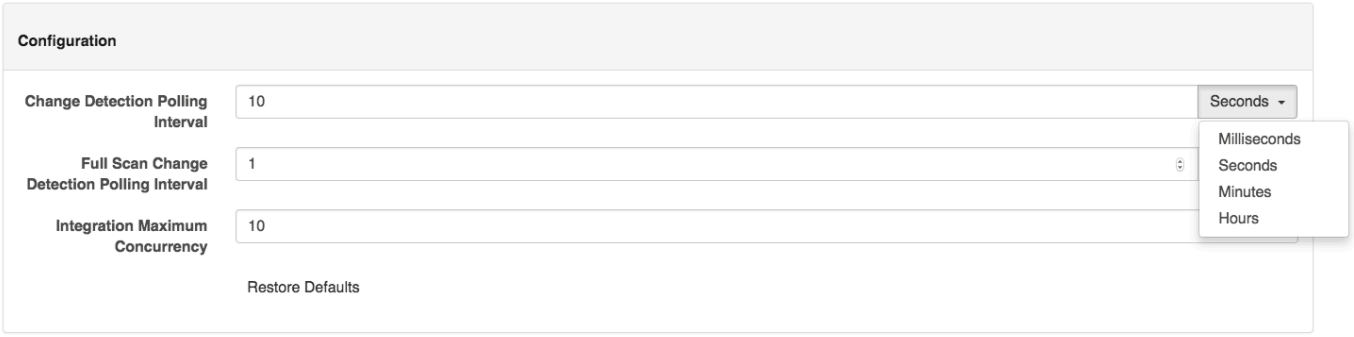
Settings

Tasktop: Apex Release (17.1)

- Polling Interval Configuration
- Logging
  - Default Logging Enabled
  - Troubleshooting Logging Enabled
  - Downloading Logs
- Scripts
  - Custom Data Transformation Script
  - Payload Transformation Script
  - Person Mapping Script
  - State Transition Script

Polling Interval Configuration

This section allows the administrator to change the polling frequency of the connected repositories.



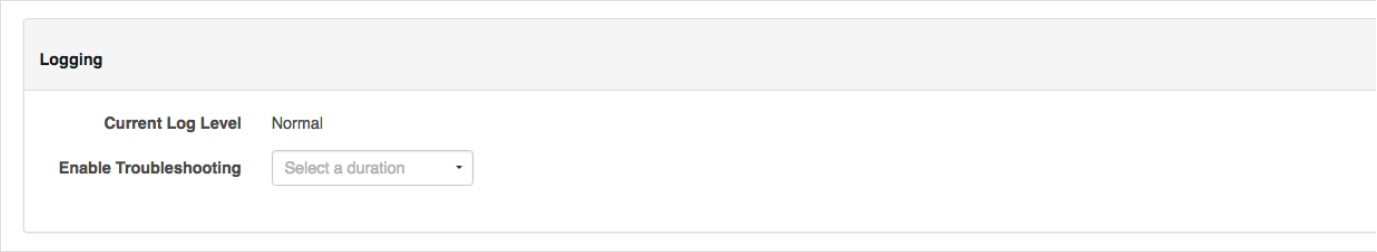
- Change Detection Polling Interval: The time between polling requests to detect *only changed artifacts*. This defaults to 1 minute, but can be customized as desired.
- Full Scan Change Detection Polling Interval: The time between polling requests to detect changed artifacts, in which *all* artifacts of a collection are scanned. This defaults to 10 hours, but can be customized as desired.
- Integration Maximum Concurrency: This limits the number of events processed concurrently by each integration. Increasing this value will enable more artifact changes to flow concurrently, whereas decreasing this value will reduce the level of concurrent changes. Changing this value has the potential to affect the load on the end-points of an integration, and may have an adverse effect on performance if set too high. The default setting (10) should be used unless advised to change by Tasktop.

Logging

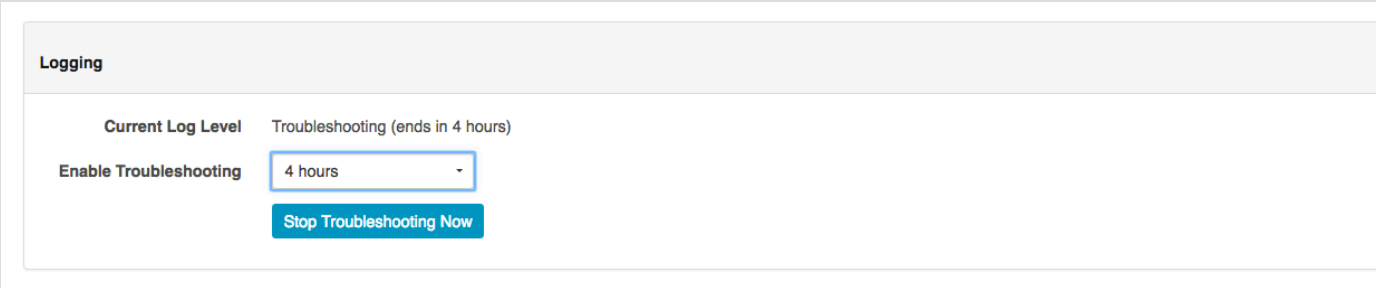
Tasktop logs various events the application performs. These can be vital for troubleshooting purposes. There are two logging levels available. The first is "Normal" and is sufficient for most scenarios. In the event that more detailed logs are required, "Enable Troubleshooting" logging level is available. Due to the large volume of logs created during Troubleshooting logging, this option has a time limit with a maximum of 24 hours. If Troubleshooting level is selected, the Normal logging level can be enabled at any time by clicking the Stop Troubleshooting Now button.

Updating the logging levels immediately changes the logging granularity and Tasktop does not need to be restarted for the change to take effect.

Default Logging Enabled



Troubleshooting Logging Enabled

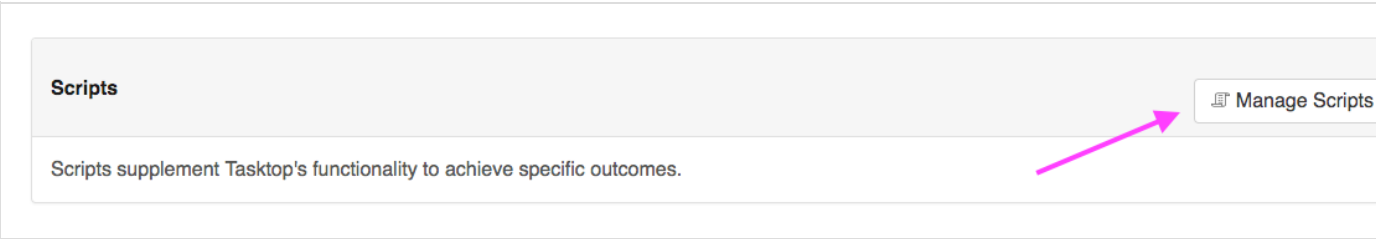


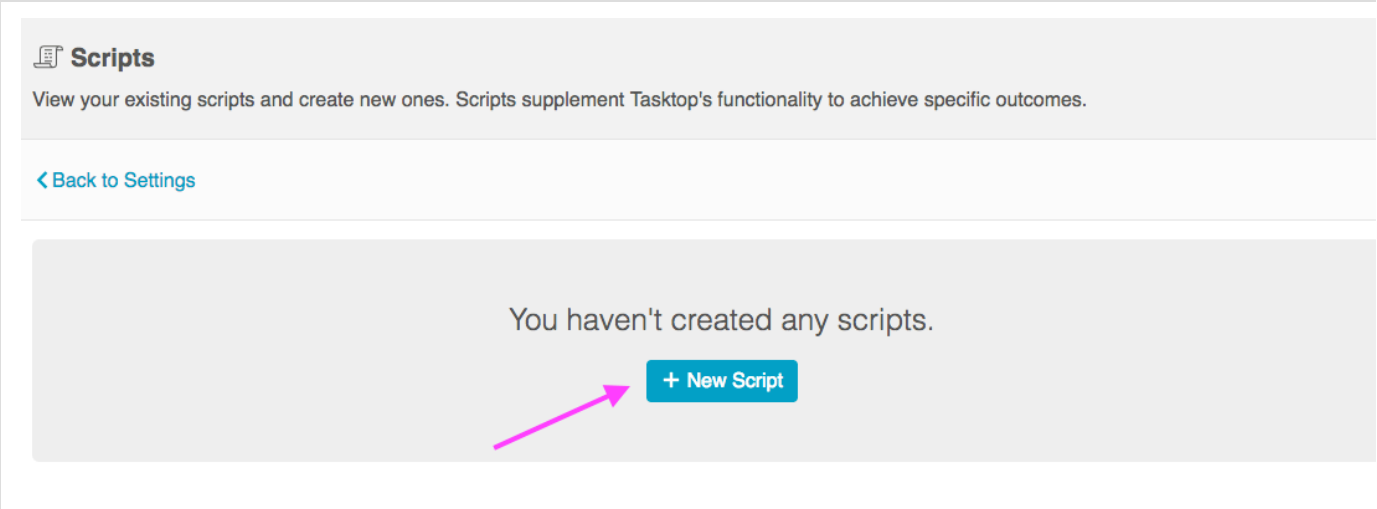
Downloading Logs

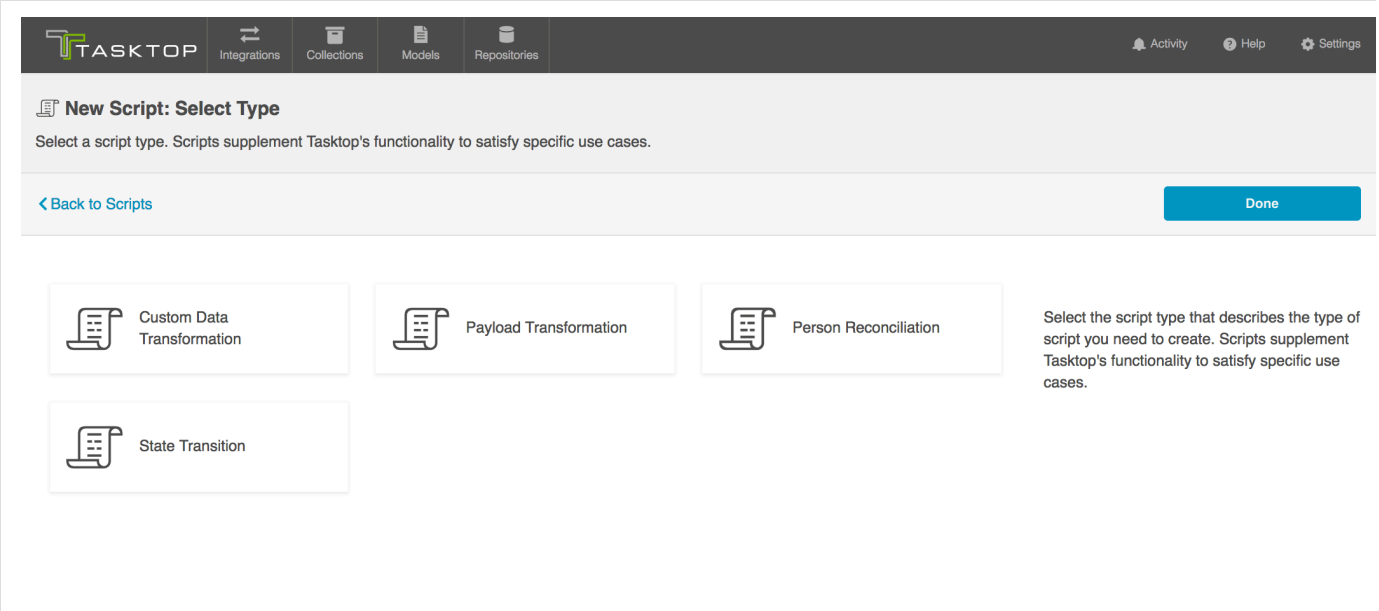
Please reference the Troubleshooting page for instructions on downloading the logs as part of the Error Report.

Scripts

You can create and save custom scripts for use in your integrations from the Settings screen. To create and edit scripts, click the 'Manage Scripts' button.







Custom Data Transformation Script

⚠ *Note: Custom Data Transformations are available only for Enterprise and Ultimate Editions. Custom Data Transformations are currently available for Pro users for a limited time, in preview mode.*

Custom Data Transformation Scripts enable you to map fields to one another which do not have out-of-the-box transforms. You can apply this script when configuring mappings from the Collection-to-Model mapping page.

Payload Transformation Script

Payload Transformation Scripts enable you to take the payload sent in by your Gateway Collection and transform it into a format that Tasktop can accept. Once you have saved your script, you can select it on the Gateway Collection screen.

Person Mapping Script

⚠ *Note: Custom Person Mapping Scripting is available only for Enterprise and Ultimate Editions. Custom Person Mapping Scripting is currently available for Pro users for a limited time, in preview mode.*

Person Mapping Scripts enable you to match 'person' fields from one repository to another. You can select the script on the Person Resolution Strategy screen during the Collection configuration process.

State Transition Script

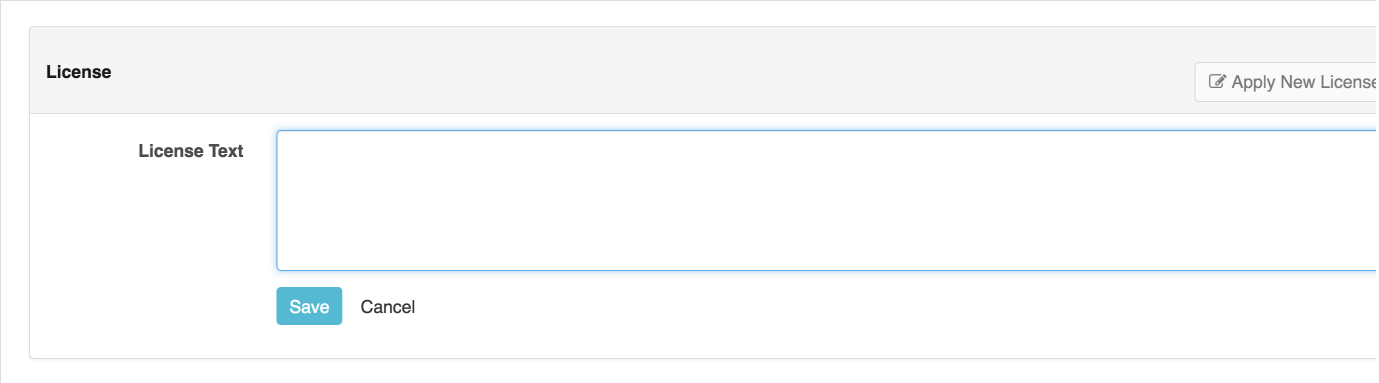
State Transition Scripts enable you to transition artifacts from one state to another according to a set workflow. The script can be applied when configuring mappings from the Collection-to-Model mapping screen.

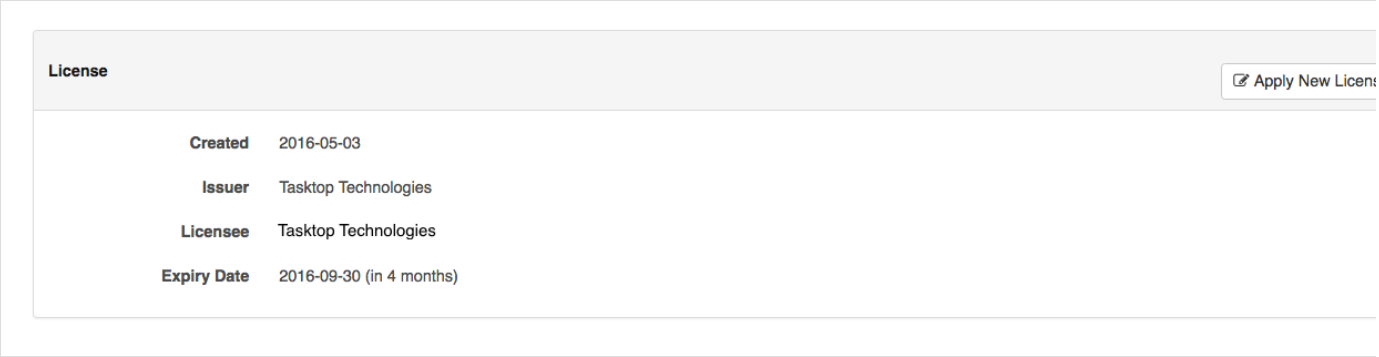
License

A license is required to run the application. Enter the license in the provided field on the Settings screen.

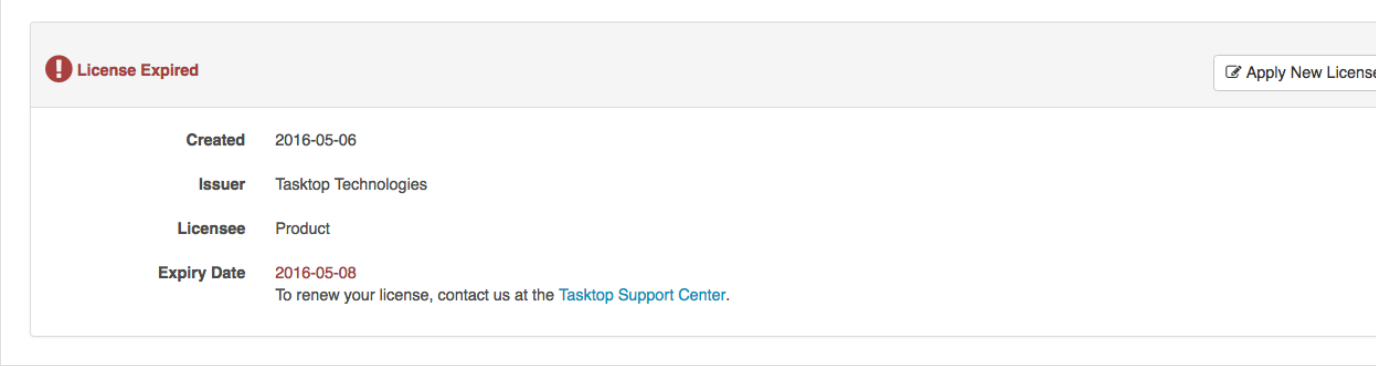
The Secure Password Storage must be set and the License must be entered before the application can be

used.

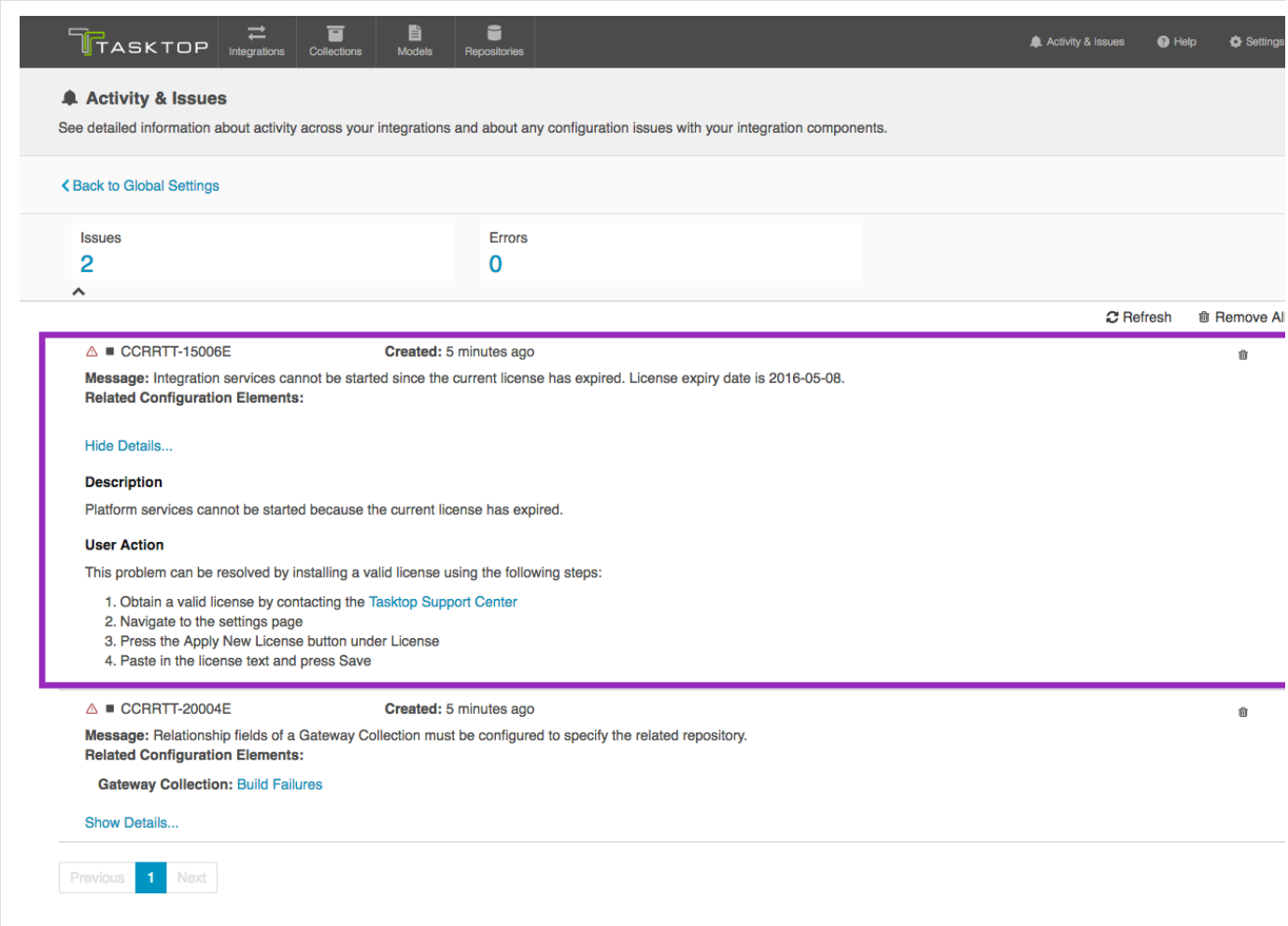




Here you'll be able to see when your license is set to expire (and will see a warning if your license is already expired):



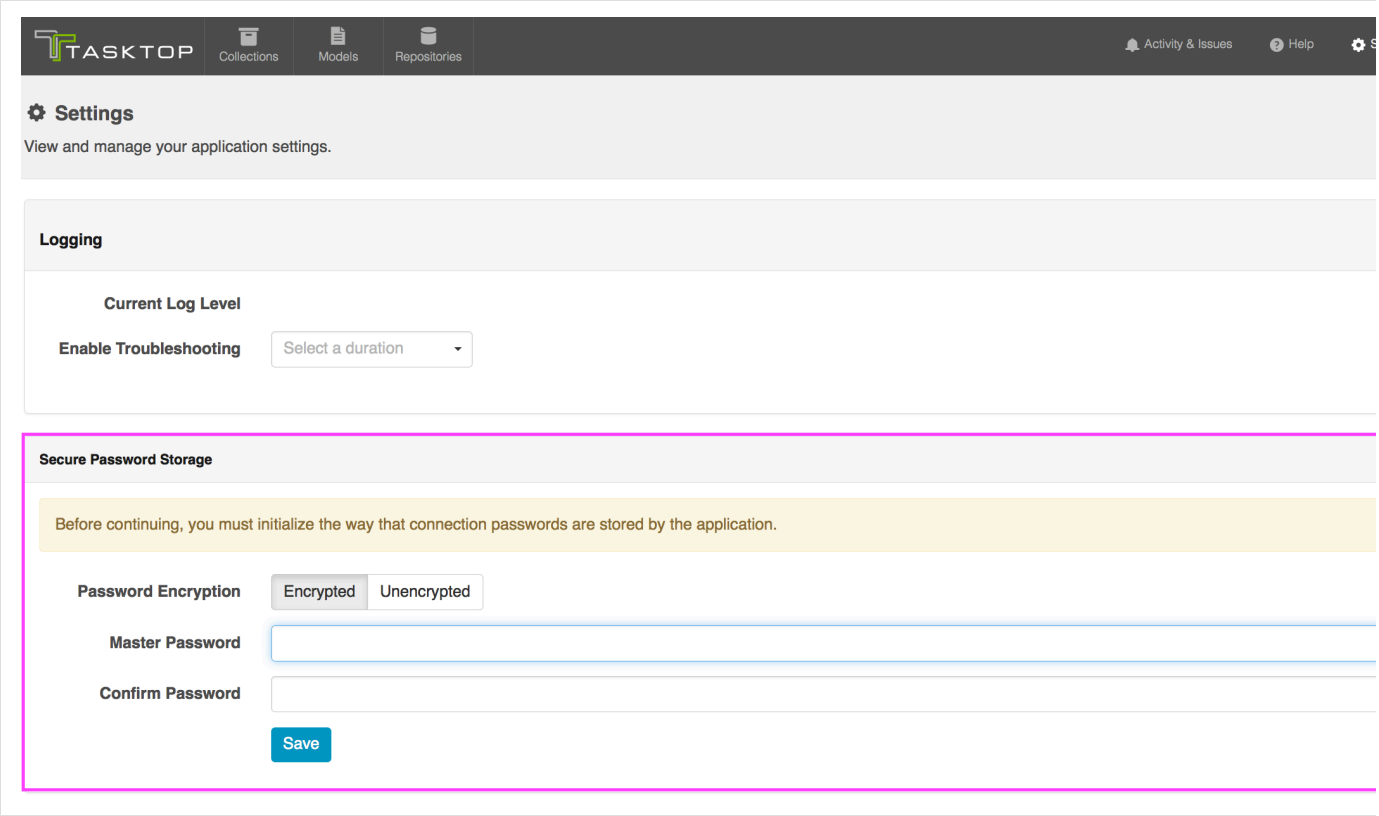
Should your license expire, in addition to seeing a warning on the Setting page, you'll also see that an issue is surfaced on the Activity screen:



When your license is expired, you'll still be able to navigate in the 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.

Secure Password Storage

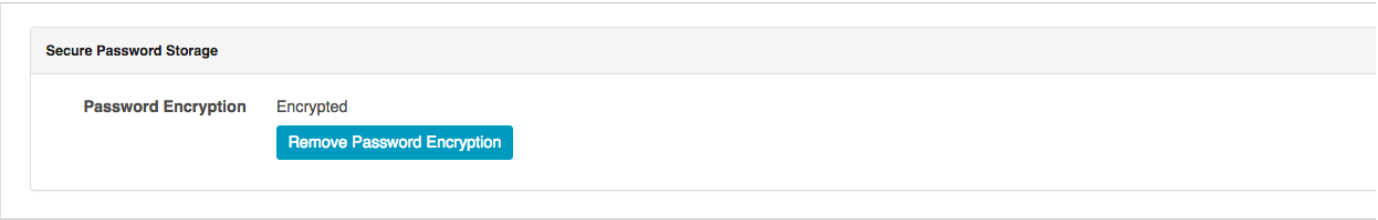
After installation, you must choose which Secure Password Storage option you would like to use: Encrypted (recommended for production environments) or Unencrypted (recommended for demo and testing environments).

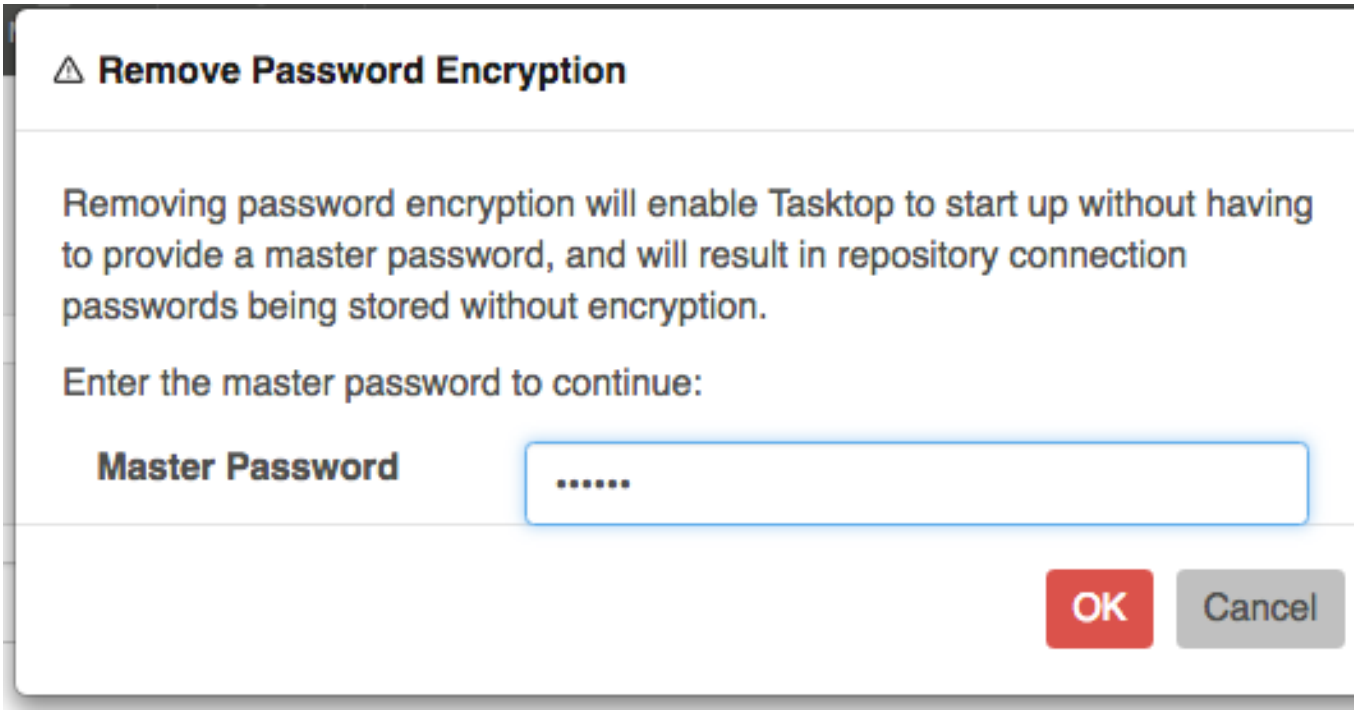


Encrypted

Selecting Encrypted security level will require you to set a Master Password. This password will be required to restart the application in the future. Selecting this security level will also result in repository connection passwords being stored with encryption.

Note: At a later point you can switch to run in unencrypted mode by providing the password originally set:

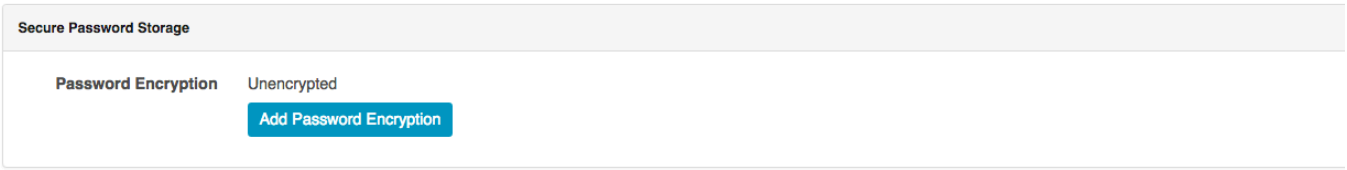


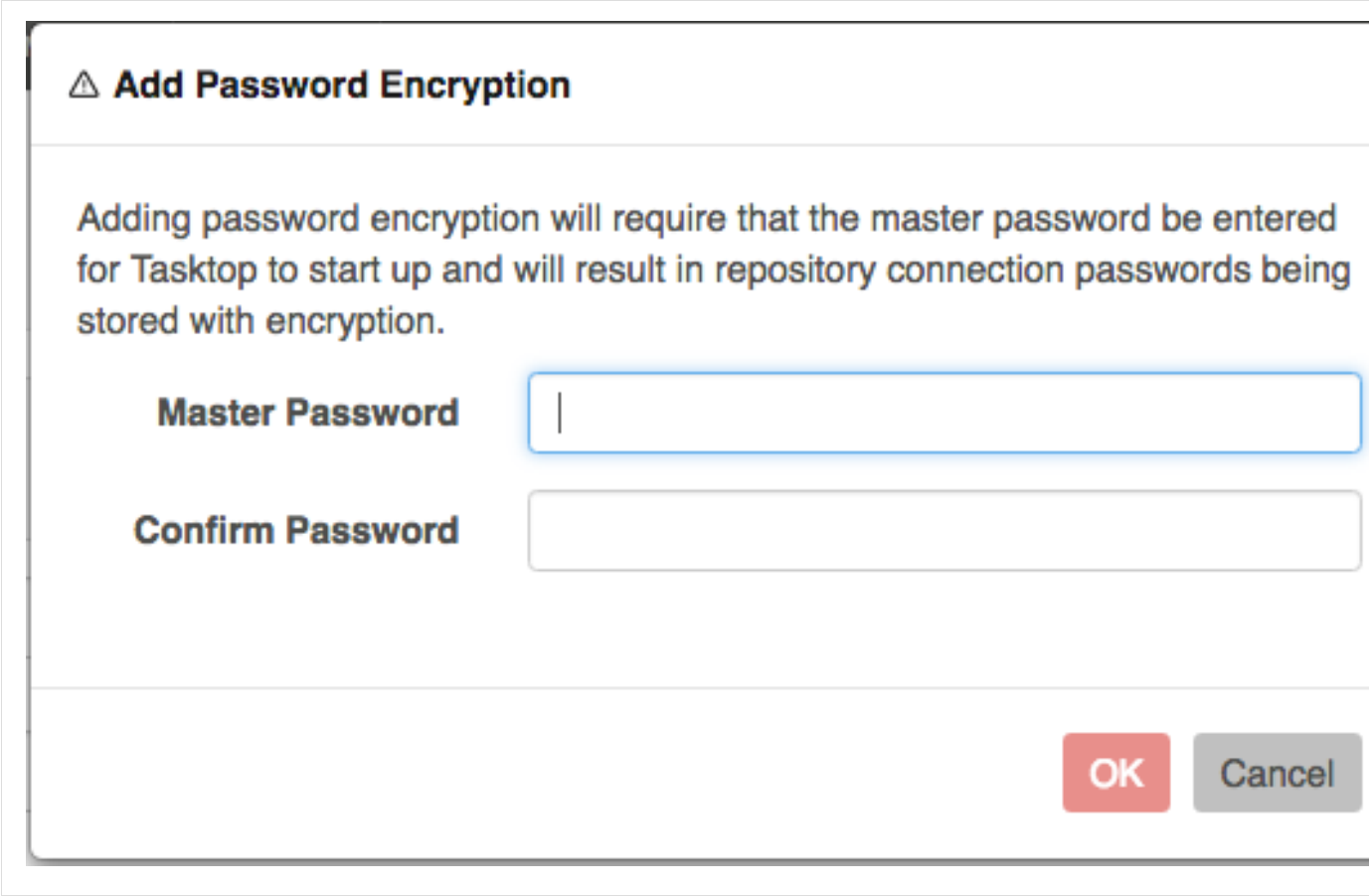


Unencrypted

Setting the security level to unencrypted does not require a password. The application can be shutdown and restarted without any password.

Note: At a later point you can switch to encrypted mode by specifying a password.





Upgrading

Tasktop: Apex Release (17.1)

- Linux
- Windows
- Recovering from an error during upgrade

Before upgrading Tasktop, be sure to shut down Tasktop and afterwards backup the internal database. The first time that Tasktop restarts after an update, 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. Also make backups of the Tomcat and Catalina configuration files that have been customized. The upgrade process will overwrite these configuration files.

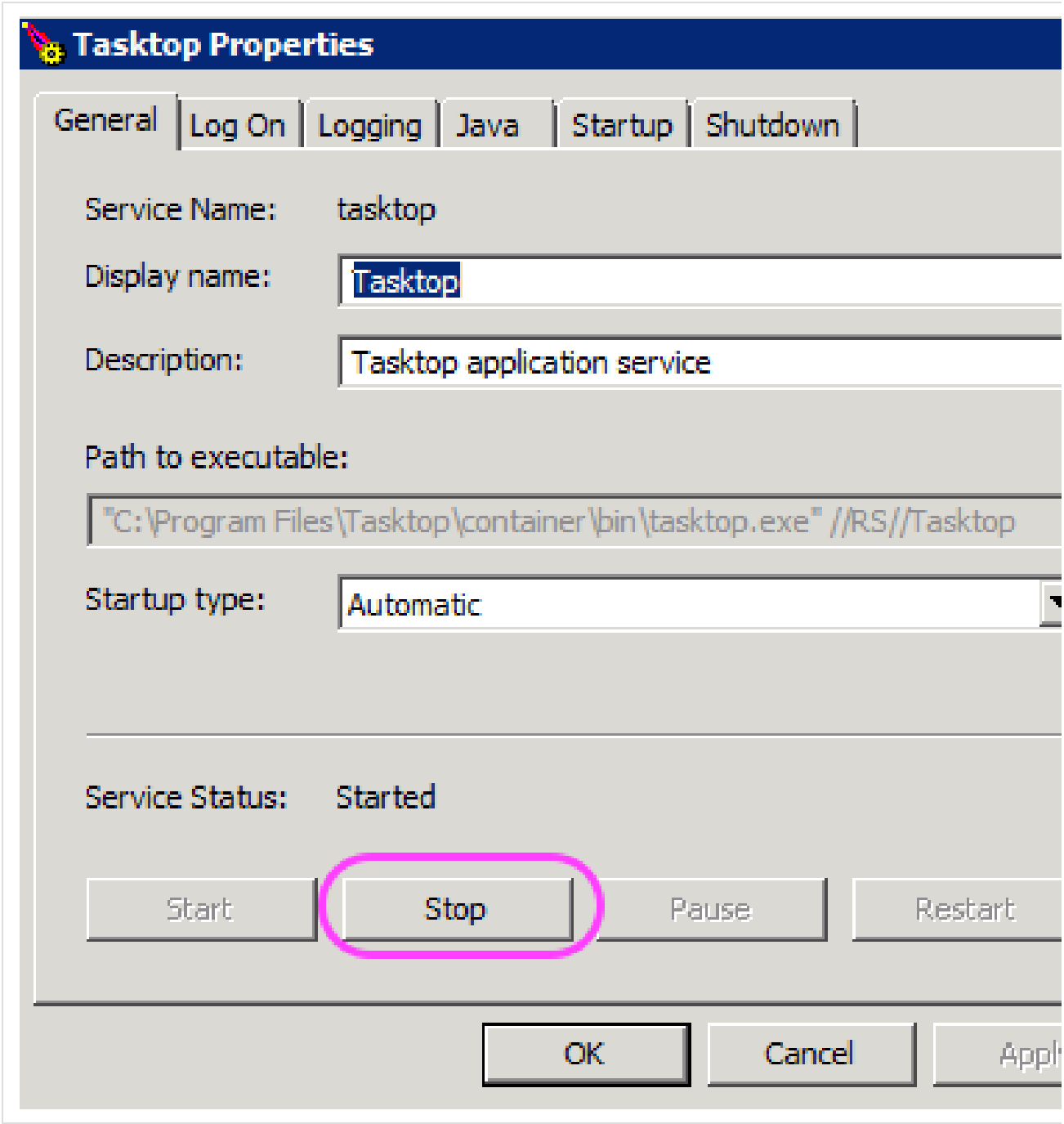
Linux

1. Ensure a copy of the old distribution archive is available in case a roll-back is required
2. Shutdown Tasktop and Keycloak
3. Back up the Keycloak configuration database (User's Tasktop data location/keycloak/standalone/data/keycloak.h2.db)
4. Move the old Tasktop installation to an archive folder
5. Unzip the new Tasktop distribution archive
6. Copy the `tasktop/db` folder from the old installation into the new installation folder `<install-location>/tasktop`
7. Re-apply any customizations to the Tomcat and Catalina configuration

8. Restore the Keycloak configuration database (User's Tasktop data location/keycloak/standalone/data/keycloak.h2.db) after installation
9. Restart Tasktop
10. Open the errors page and resolve errors related to unsatisfied connector requirements

Windows

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



3. Shutdown Tasktop and Keycloak
4. If upgrading from 17.1.0 or earlier: Back up the Keycloak configuration database (User's Tasktop data location/keycloak/standalone/data/keycloak.h2.db)

5. Backup internal database, located at `<ProgramData>\Tasktop\db`
6. Run the installer of the new version of Tasktop
7. Re-apply any customizations to the Tomcat and catalina configuration
8. If upgrading from 17.1.0 or earlier: Restore the Keycloak configuration database(User's Tasktop data location/keycloak/standalone/data/keycloak.h2.db) after installation
9. Restart Tasktop
10. Restart Keycloak
11. Open the errors page and resolve errors related to unsatisfied connector requirements

Recovering from an error during upgrade

If Tasktop fails to restart after an upgrade, or there are errors starting the integrations, then Tasktop will need to be returned to the previous version.

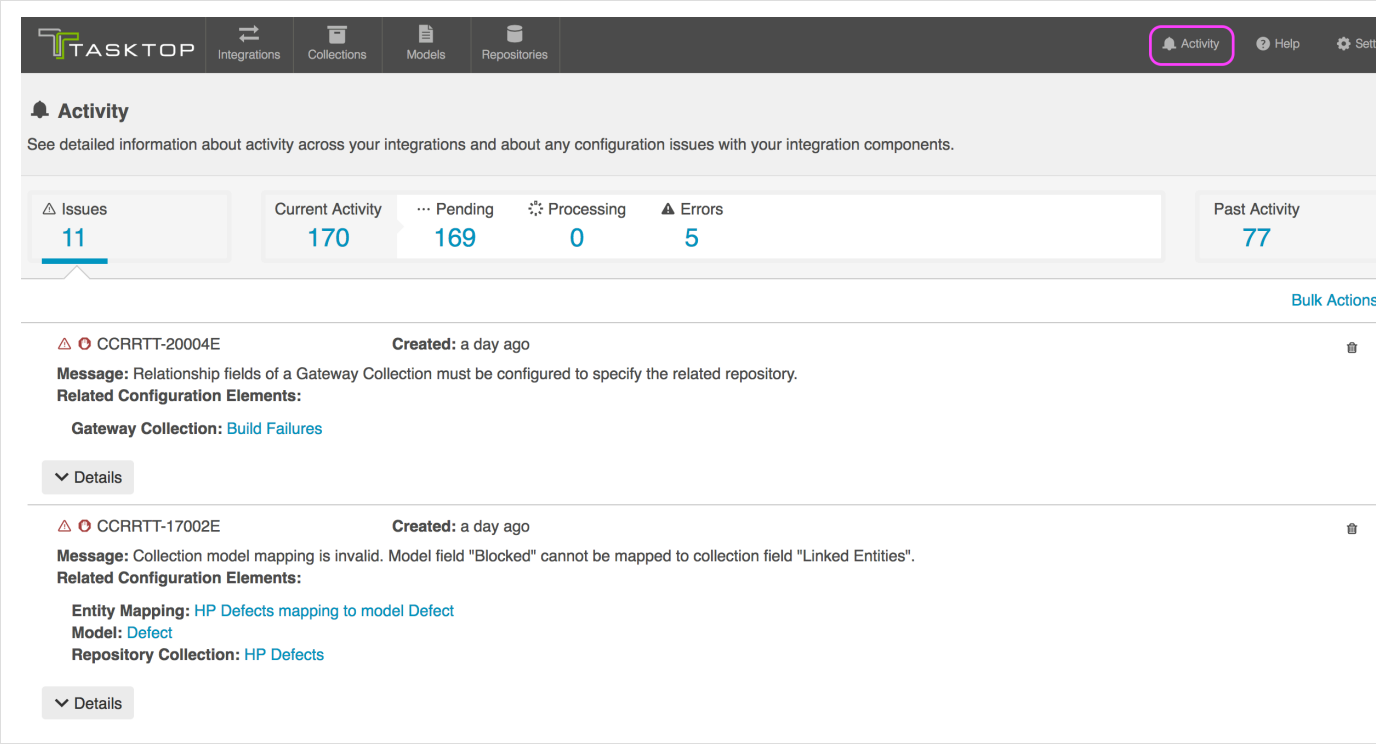
1. Shutdown Tasktop
2. Remove the new version and restore the archived version (Linux) or uninstall and run previous installer (Windows)
3. Delete the `<install-location>/tasktop/db` (Linux) or `<ProgramData>\Tasktop\db` (Windows) folder and replace it with the backed-up version
4. Restart Tasktop

Troubleshooting

Tasktop: Apex Release (17.1)

Activity Screen

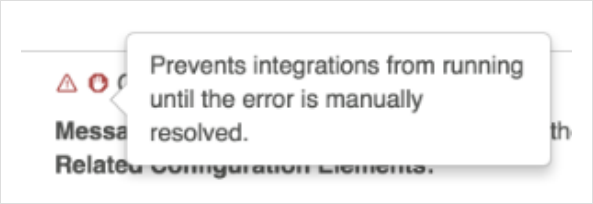
Most problems can be solved by looking at the Activity screen and following steps described on the errors displayed there. The Activity screen can be seen by clicking on “Activity” in the top right of the web application menu bar:



Issues

The *Issues* category shows issues that arise from invalid configuration of the integration components you have set up on Tasktop or from more global issues, such as having an invalid or expired license. These are things that can generally be resolved within the Tasktop application itself.

An additional warning icon appears when these issues are so fundamental that they will prevent integrations from running:



Current Activity

The *Current Activity* category shows events that are active in an integration and provides a good view to monitor what is happening in your integrations.

Current Activity encompasses the following:

- Pending: These are events that are queued up to be be processed.
- Processing: These are events that are currently processing.
- Error: These are events that Tasktop tried to process but that, for one reason or another, were not successful.

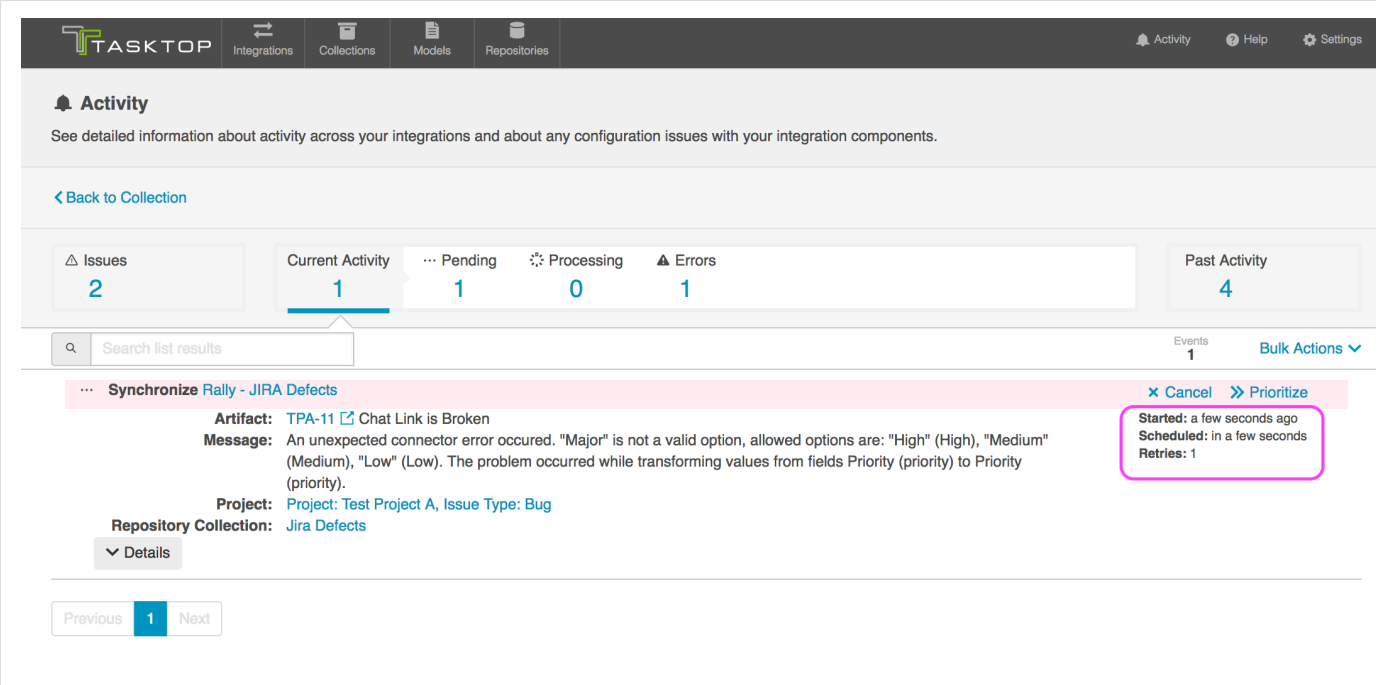
You can take different actions on the events in these different subcategories.

- Pending:
  - Prioritize: Prioritize this pending event in the queue.

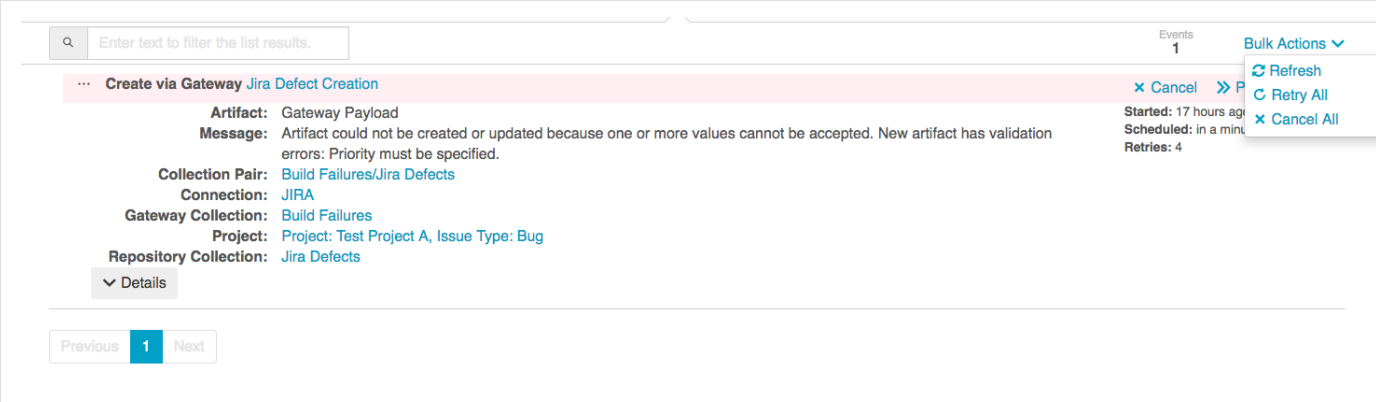
    - Cancel: Remove this event from the pending queue. It will not be processed, though subsequent changes to artifacts will trigger another event.
- Processing:
    - N/A
- Error:
    - Prioritize: Prioritize the re-try of this error in the queue. This option is especially useful if you have made changes 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 "error" and "pending" states simultaneously.
    - Re-try: Re-try this error.
        - You will see this action if the event is not already set to be re-tried.
    - Cancel: Remove this error from the list. It will not be retried, though subsequent changes to artifacts will trigger another event.

⭐ Note: Most errors will automatically be re-tried on a gradually decreasing interval (granted that Tasktop can locate the artifact that is to be changed). Re-tryable errors will be retried approximately 30 seconds after they are first encountered, and then on a gradually decreasing interval over time.

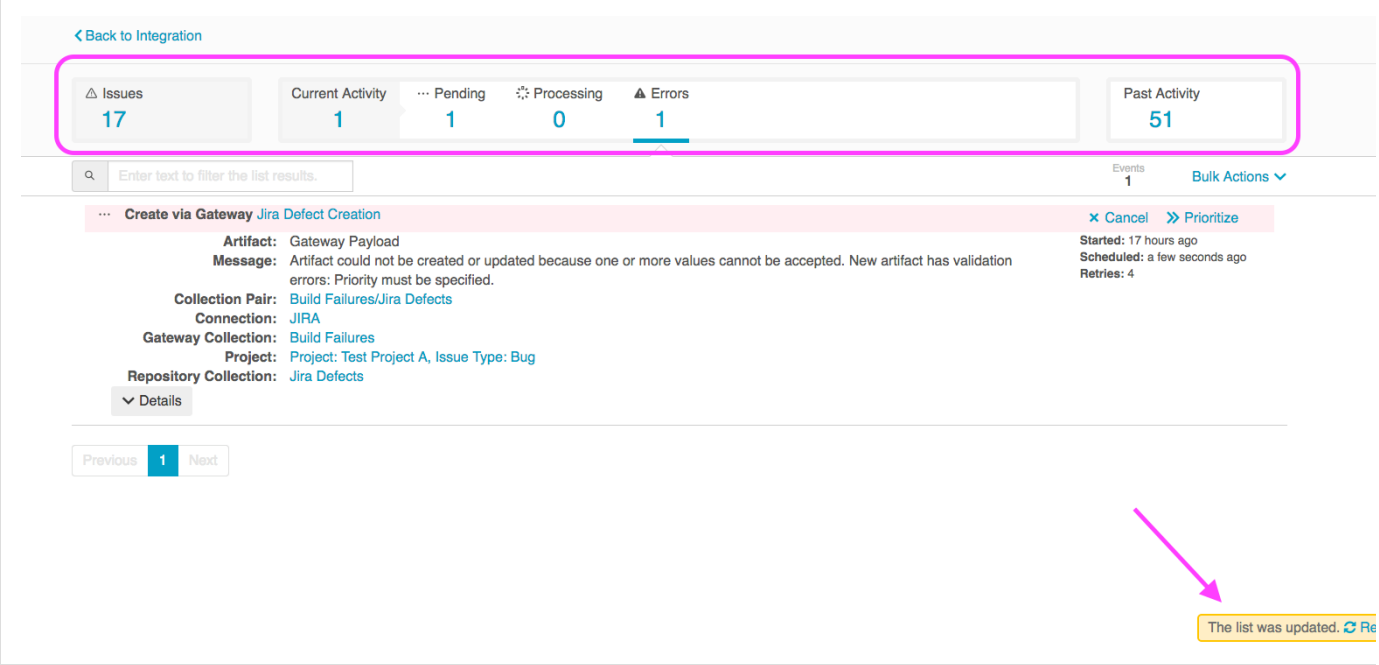
You can see information about errors being retried on the error itself. In the example below, you can see that the error has been retried 1 time. If an error will not be re-tried, this information will not be be relevant and hence will not be displayed.



You can also take bulk actions on events in these categories:



Note: 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:

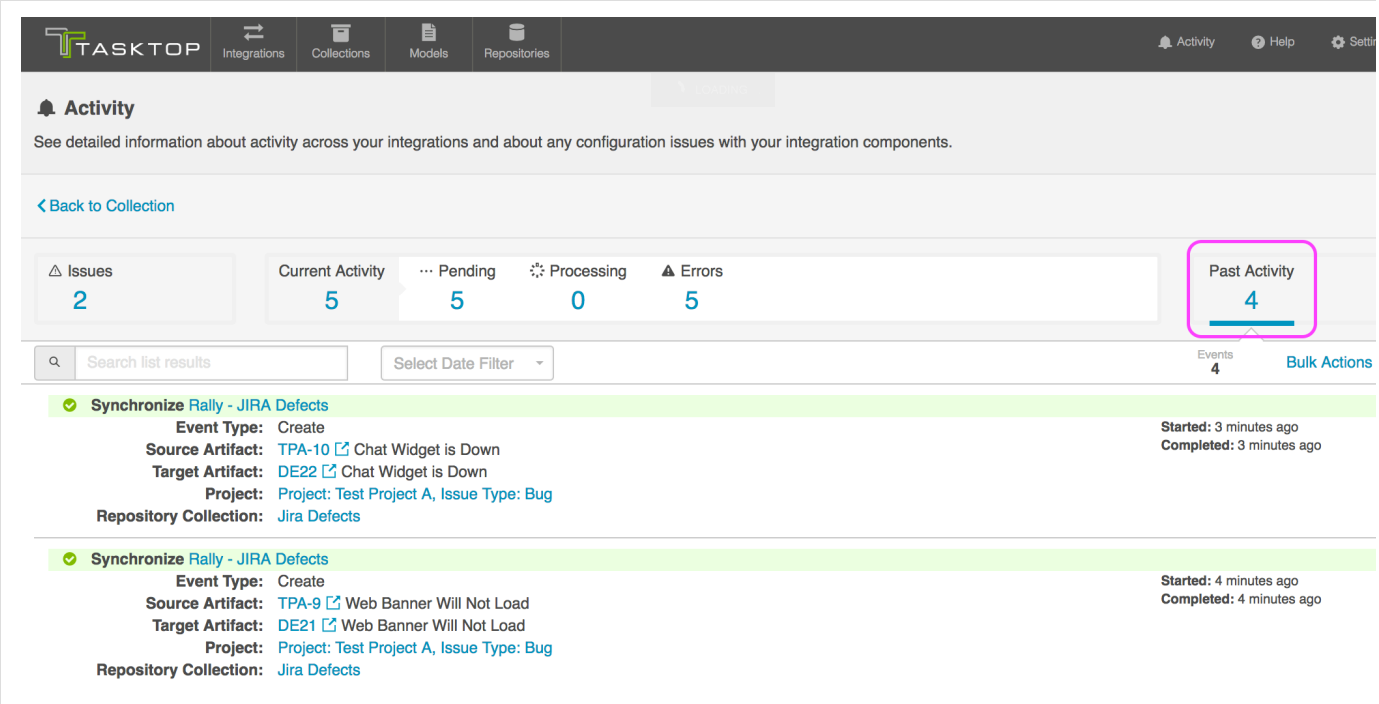


This is to avoid items unexpectedly appearing and disappearing when you might be examining them.

A complete listing of errors is available in the appendix .

Past Activity

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



Here, you can see:

- Integration Name and Type: This is displayed in the green banner at the top of each item. If your Integration has since been deleted, the name will be displayed as the Integration ID (as Tasktop does not retain name data for deleted Integrations), and you will see a "No longer available" note.
- Event Type
  - Create: An artifact was created in your collection
  - Update: An artifact was updated in your collection
- Source Artifact(s): The original artifact
- Target Artifact(s): The corresponding artifact created or updated via Tasktop in your other collection
- Message: If applicable, displays any prior errors that were encountered, before activity successfully completed
- Project: If applicable, project for the source artifact
- Collection: If applicable, collection for the source artifact
- Started: When processing started for this artifact
- Completed: When processing completed for this artifact
- Retries: If applicable, number of retries for this artifact

You can use the search box on this page to search for text within the source artifact or integration name. Additionally, you can use the date filter to search either 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 also use the Bulk Actions to refresh, or remove all past activity that meets your current search filters. If you have not entered any search filters, all past activity will be refreshed or removed.

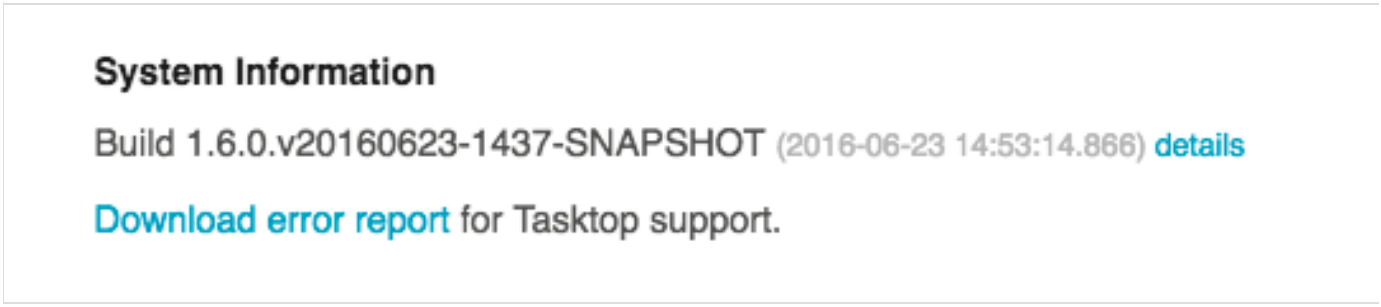
⚠️ Note that Tasktop will store up to 100,000 entries on the Past Activity screen. Once 100,000 entries are met, older entries will be deleted as new entries come in. You can also opt to clear your entries when approaching 100,000 to have better visibility into more recent past activity.

Error Report

In cases where the Activity screen is not enough to resolve a problem, an Error Report is available to

provide additional information.

The Error Report can be downloaded from Tasktop. To download, click the "Download error report" link in the System Information section on the Help page.



Error Report Contents

The downloaded Error Report file is name tasktop-state.zip. Once unzipped, there will be three folders. The folders and contents are listed below.

1. configuration
   - configuration.json
   - platform-details.json
2. logs
   - logs by day for past 14 days
3. metrics
   - metrics.json

File Name	Contents
configuration.json	Contains all the configuration of your application instance.
platform-details.json	Contains details about the specific build and license of the application
logs	A separate file is created for every day of logs. 14 days of logs are saved.
metrics.json	Contains various metrics of the application.

Configuring Logging

See the Logging section of the Settings page.

Resources

Tasktop: Apex Release (17.1)

Help and Support

The distribution comes with a README.txt file; Please refer to this document for information about

starting and stopping the application, security, and logging.

To learn more about Tasktop, see our website

For help, contact us at the Tasktop Support Center.

Feedback and Ideas

Have a suggestion or an idea for the product? Please contact us at feedback@tasktop.com.

Appendix A- Error Messages

Tasktop: Apex Release (17.1)

The following is a complete list of error messages. Error messages are displayed in the Activity view. More information about the Errors view can be found under Troubleshooting.

    - User Action
- CCRRTT-1102E – Connection HTTP proxy credentials were not accepted by the repository.
    - Description
    - User Action
- CCRRTT-1103E – Connection settings are invalid.
    - Description
    - User Action
- CCRRTT-1104W – Authentication state for repository connection has expired.
    - Description
    - User Action
- CCRRTT-1105E – Repository Collection project is invalid.
    - Description
    - User Action
- CCRRTT-1106E – API call limit on repository has been exceeded.
    - Description
    - User Action
- CCRRTT-1107E – Connection could not be established with a repository due to a failure during authentication.
    - Description
    - User Action
- CCRRTT-1401E – Integration must specify at least one route.
    - Description
    - User Action
- CCRRTT-1402E – Integration must satisfy style constraints.
    - Description
    - User Action
- CCRRTT-1403E – Integration must have all collections attached to the same model.
    - Description
    - User Action
- CCRRTT-1404E – Collection must have a mapping to a model.
    - Description
    - User Action
- CCRRTT-1405E – Integration must have a source Collection.
    - Description
    - User Action
- CCRRTT-1406E – Integration must have a target Collection.
    - Description
    - User Action
- CCRRTT-1408E – Integration failed to lookup artifact.
    - Description
    - User Action
- CCRRTT-1409E – Integration has invalid filter.
    - Description
    - User Action
- CCRRTT-1410E – Integration must specify a key identifier.
    - Description
    - User Action
- CCRRTT-1411E – All specified routes of an integration must be configured.
    - Description
    - User Action
- CCRRTT-1412E – Integration has a conditional route with invalid configuration.
    - Description

    - User Action
- CCRRTT-1413E – Collection has invalid repository query.
    - Description
    - User Action
- CCRRTT-10004E – Enterprise Reporting Integration must have exactly one target SQL Collection.
    - Description
    - User Action
- CCRRTT-10005E – Enterprise Reporting Integration must have a source Collection.
    - Description
    - User Action
- CCRRTT-10006E – Enterprise Reporting Integration target Collection must have appropriate mapping.
    - Description
    - User Action
- CCRRTT-10007E – Enterprise Reporting Integration source Collection must provide the correct model.
    - Description
    - User Action
- CCRRTT-10008E – Enterprise Reporting Integration target Collection must have exactly one project.
    - Description
    - User Action
- CCRRTT-15002E – Integration services cannot be started due to a problem with the license.
    - Description
    - User Action
- CCRRTT-15005E – Repository cannot be used due to a problem with the license.
    - Description
    - User Action
- CCRRTT-15006E – Integration services cannot be started since the current license has expired.
    - Description
    - User Action
- CCRRTT-15007E – Integration cannot be used with the configured repositories due to a restriction in the license.
    - Description
    - User Action
- CCRRTT-15008E – Collection cannot be used with the configured person mapping script due to a restriction in the license.
    - Description
    - User Action
- CCRRTT-15009E – Collection cannot be used with the configured state transition script due to a restriction in the license.
    - Description
    - User Action
- CCRRTT-15010E – Mapping cannot be used with the configured value transformation script due to a restriction in the license.
    - Description
    - User Action
- CCRRTT-16001E – Services cannot be started until Tasktop security has been initialized.
    - Description
    - User Action

- CCRRTT-16002E – Error initializing password encryption.
  - Description
  - User Action
- CCRRTT-17001E – Mapping cannot be applied since it is not valid within the current context.
  - Description
  - User Action
- CCRRTT-17002E – Collection model mapping is invalid.
  - Description
  - User Action
- CCRRTT-17003E – Artifact could not be created or updated because one or more values cannot be accepted.
  - Description
  - User Action
- CCRRTT-17004W – Artifact cannot be processed since it is currently in use.
  - Description
  - User Action
- CCRRTT-17005E – Field flow is invalid.
  - Description
  - User Action
- CCRRTT-17006E – Artifact was created but some values could not be set.
  - Description
  - User Action
- CCRRTT-17007E – Conflict resolution strategy is invalid.
  - Description
  - User Action
- CCRRTT-17008E – Artifact could not be processed as it did not meet any of the configured conditions on the Conditional Artifact Routing page.
  - Description
  - User Action
- CCRRTT-17009E – Invalid state transition.
  - Description
  - User Action
- CCRRTT-17010E – Repeated state transition.
  - Description
  - User Action
- CCRRTT-17011E – Script completed with an error.
  - Description
  - User Action
- CCRRTT-17013E – State transition scripts can only be applied with models that have a status field.
  - Description
  - User Action
- CCRRTT-20000E – No integration is listening to the Gateway Collection.
  - Description
  - User Action
- CCRRTT-20001E – Time Tracking integration model must have a field of type time entries.
  - Description
  - User Action
- CCRRTT-20002E – Time Tracking integration Collection must have a field mapping to a field of type time entries in the Model.
  - Description
  - User Action

- CCRRTT-20003W – Time Tracking integration target Collection does support impersonation of the Worker field.
  - Description
- CCRRTT-20004E – Relationship fields of a Gateway Collection must be configured to specify the related repository.
  - Description
  - User Action
- CCRRTT-20005E – Gateway collection must have a model.
  - Description
  - User Action
- CCRRTT-20006E – Gateway Collection cannot be used with the configured payload transformation script due to a restriction in the license.
  - Description
  - User Action
- CCRRTT-30000E – An unexpected error occurred.
  - Description
  - User Action
- CCRRTT-30001E – Not found.
  - Description
  - User Action
- CCRRTT-30002E – The data provided was not valid.
  - Description
  - User Action
- CCRRTT-30003E – The connector kind was not found.
  - Description
  - User Action
- CCRRTT-30004E – The request entity was not valid JSON.
  - Description
  - User Action
- CCRRTT-30005E – Secure password storage must be initialized.
  - Description
  - User Action
- CCRRTT-30006E – Error communicating with {0} repository.
  - Description
  - User Action
- CCRRTT-30007E – Error processing request MIME attachment.
  - Description
  - User Action
- CCRRTT-30008E – Tasktop is stopped, see the Issues UI and error log for more details.
  - Description
  - User Action
- CCRRTT-50001E – Unable to propagate artifact changes since the target artifact has been removed.
  - Description
  - User Action
- CCRRTT-50002E – A conflict has occurred during synchronization.
  - Description
  - User Action
- CCRRTT-50003E – Relationship values could not be resolved during synchronization.
  - Description
  - User Action
- CCRRTT-50004E – Relationship values could not be resolved during synchronization.

Appendix A – Error Messages

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.

CCRRTT-0004E – Configuration migration failed.

Description

Configuration could not be migrated to match an updated version of Tasktop due to one or more errors.

User Action

1. Investigate the cause of failure by viewing related errors under Issues on the Activities & Issues page.
2. Attempt to resolve error according to the specific error message and corresponding user actions.
3. Restart the Tasktop application.

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 and the repository.
2. Try connecting again later.

If the problem persists, contact your network administrator.

CCRRTT-1001E – 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 “Connections” page.
2. Select the connection for which the requirements were unsatisfied.
3. On the connection page, provide the required files.

CCRRTT-1002E – An unexpected connector error occured.

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-1101E – 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-1102E – 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-1103E – 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-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 – Repository Collection project is invalid.

Description

The Repository Collection project is not valid. This problem is usually caused by a project and/or type being deleted from 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. Correct the problem on the repository and then click "Refresh Projects" on the Repository Collection, or
3. Remove the referenced project from the Repository Collection

CCRRTT-1106E – API call limit on repository has been exceeded.

Description

The API limit imposed by the repository has been exceeded. This problem is usually caused during periods of heavy load.

User Action

This error will resolve itself automatically when the repository is no longer imposing a rate limit.

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-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-10004E – Enterprise Reporting Integration must have exactly one target SQL Collection.

Description

An Enterprise Reporting 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 Reporting Integration must have a source Collection.

Description

An Enterprise Reporting 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 Reporting Integration target Collection must have appropriate mapping.

Description

An Enterprise Reporting 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 Reporting Integration.

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

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

Description

An Enterprise Reporting 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 Reporting 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 Reporting Integration target Collection must have exactly one project.

Description

An Enterprise Reporting 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-15002E – Integration services cannot be started due to a problem

with the license.

Description

Tasktop 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-15006E – Integration services cannot be started since the current license has expired.

Description

Tasktop 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-15007E – 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-15008E – Collection cannot be used with the configured person mapping script due to a restriction in the license.

Description

A collection has been configured with a person mapping script, which is not permitted by the current license.

User Action

Perform one of the following:

- Delete the offending collection
- Remove the person mapping script from the offending collection

CCRRTT-15009E – Collection cannot be used with the configured state transition script due to a restriction in the license.

Description

A collection has been configured with a state transition script, which is not permitted by the current license.

User Action

Perform one of the following:

- Delete the offending collection
- Remove the state transition script from the offending collection

CCRRTT-15010E – Mapping cannot be used with the configured value

transformation script due to a restriction in the license.

Description

A mapping has been configured with a value transformation script, which is not permitted by the current license.

User Action

Perform one of the following:

- Delete the offending collection
- Remove the value transformation script from the mapping in the offending collection

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

Description

Tasktop 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-16002E – 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 includi ng a README file with installation instructions.

Alternatively, the unencrypted level of the password store maybe used.

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 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

A script provided invalid values when attempting to transition an artifact.

User Action

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

CCRRTT-17010E – Repeated state transition.

Description

A script attempted to transition an artifact with the same transition more than once.

User Action

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

CCRRTT-17011E – Script completed with an error.

Description

A script completed with an error. See the specific error message for details.

Scripts complete with errors for one of two reasons:

- the script intentionally raised an error, for example to indicate that a business rule was not satisfied
- the script 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 script to prevent the error from occurring, or
3. Modify the source or target artifact to satisfy the condition that caused the error

CCRRTT-17013E – State transition scripts can only be applied with models that have a status field.

Description

A state transition script is used in a collection mapping to a model that has no status field. A status field is required in the model in order to use a state transition script.

User Action

Either remove use of the state transition script, or ensure that the model has a status field.

To add a status field to the model:

1. navigate to the model
2. ensure that one field of the model has the Smart Field set to "Status"

To remove the transition script from the mapping:

1. navigate to the collection
2. navigate to the collection field mappings via the “Map Fields” link
3. remove the field mapping of the field using the state transition script or change it to use a different transform via the “Configure” link of the field mapping

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-20001E – 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-20002E – 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-20003W – Time Tracking integration target Collection does 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-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 script due to a restriction in the license.

Description

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

User Action

Perform one of the following:

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

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 application temporary directory.

User Action

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

CCRRTT-30008E – Tasktop is stopped, see the Issues UI and error log for more details.

Description

Tasktop 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-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

- Delete the associated artifact, or
- Move the associated artifact out of its collection such that the artifact is no longer synchronized, or
- Apply a filter to the integration such that the artifact is no longer synchronized

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-50003E – 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-50004E – 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 two separate synchronization integrations that cause B to be copied into the target repository.

User Action

- Remove the link from A to B, or
- Remove one of the two synchronization integrations

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

Supported Repository Versions

Tasktop: Apex Release (January 31, 2017)

Download as pdf: Supported Repository Versions_Tasktop Integration Hub_Apex_17.1_2017.01.31.pdf

	Repository	Supported Versions in Tasktop Apex
JIRA	Atlassian JIRA	5.0, 5.1, 5.2, 6.0, 6.0.3, 6.1, 6.2, 6.3, 6.4, 7.0, 7.1, 7.2, 7.3,7.4, Current On Demand Version
JIRA Service Desk	Atlassian JIRA - Service Desk	5.0, 5.1, 5.2, 6.0, 6.0.3, 6.1, 6.2, 6.3, 6.4
blueprint	Blueprint	6.1, 6.2, 6.3, 6.4, 7.0, 7.1, 7.2, 7.3, 7.4, Current On Demand Version
bmc	BMC Remedy	8.1.01, 8.1.02, 9.0, 9.1.00 Hotfix
ca technologies	CA Clarity PPM	14.1, 14.2, 14.3, 14.4, 15.1, Current On Demand Version
GitHub	GitHub Issues	Enterprise 11.10.343 2.0.0 and higher, Current On Demand Version
ALM Octane	HPE ALM Octane	Current On Demand Version
Hewlett Packard Enterprise	HPE PPM	9.30, 9.31, 9.32
	HPE QC / ALM	11, 11 (SP1), 11.5 (SP2), 12 (SP1), 12.2, 12.5, 12.53, Current On Demand Version
IBM	IBM Bluemix	Current On Demand Version
	IBM Rational ClearQuest	8.0.0, 8.0.1, 8.0.1.4, 9.0
	IBM Rational DOORS	9.3, 9.4, 9.5, 9.5.2, 9.6, 9.6.1

	IBM Rational DOORS Next Generation (IBM RRC)	4.0.1, 4.0.2, 4.0.3, 4.0.4, 4.0.5 Split JTS/RRC, 4.0.6, 4.0.7, 5.0, 5.0.1, 5.0.2, 6.0, 6.0.1, 6.0.2
	IBM Rational Quality Manager	4.0.1, 4.0.2, 4.0.3, 4.0.4, 4.0.5 Split JTS/RRC, 4.0.6, 4.0.7, 5.0, 5.0.1, 5.0.2, 6.0, 6.0.1, 6.0.2, 6.0.3
	IBM Rational Team Concert	4.0.1, 4.0.2, 4.0.3, 4.0.4, 4.0.5 Split JTS/RRC, 4.0.6, 4.0.7, 5.0, 5.0.1, 5.0.2, 6.0, 6.0.1, 6.0.2, 6.0.3
	IBM RequisitePro	7.1.0, 7.1.2, 7.1.3, 7.1.4
IRISE	iRise	Current On Demand Version
jama	Jama	2014.2, 2015.1, 2015.2, 2015.3, 2015.4, 2015.5, 8.0, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 8.10, 8.11, Current On Demand Version
leankit	Leankit	Current On Demand Version
Microsoft	Microsoft Project Server	2013 SP1
	Microsoft SharePoint	2013 SP1
	Microsoft Team Foundation Server	2012, 2012.1, 2012.2, 2012.3, 2012.4, 2013, 2013.2, 2013.3, 2013.4, 2015, 2015.1, 2015.2, 2015.3, 2017, Current On Demand Version
	Microsoft Test Manager	2012, 2012.1, 2012.2, 2012.3, 2012.4, 2013, 2013.2, 2013.3, 2013.4, 2015, 2015.1, 2015.2, 2015.3, 2017, Current On Demand Version
	Microsoft Visual Studio Team Services	Current On Demand Version
PivotalTracker	Pivotal Tracker	Current On Demand Version
PLANVIEW	Planview	11.3

	Polarion ALM	2014, 2014 SR1, 2014 SR2, 2014 SR3, 2015, 2015 SR1, 2015 SR2, 2015 SR3, 2016, 2016 SR1, 2016 SR2, Current On Demand Version
	Rally/CA Agile Central	2014.1, 2014.2, 2014.3, 2015.1, 2015.2, 2016.1, Current On Demand Version
	SalesForce: Sales Cloud, Service Cloud, Marketing Cloud	Current On Demand Version
	Serena Business Manager	10.1.2, 10.1.3, 10.1.4, 10.1.4.1, 10.1.5, 10.1.5.1, 10.1.5.2, 11.0, 11.0.1.1, 11.1
	Serena Dimensions RM	11.2, 11.2.1, 11.2.2, 11.2.3, 11.2.4, 12.1.0.4, 12.1.1, 12.2, 12.2.1, 12.3, 12.4
	ServiceNow: Service Desk, SDLC, PPM	Fuji, Geneva, Helsinki
	SmartBear QAComplete	11.2
	Targetprocess	Current On Demand Version
	Thoughtworks Mingle	13.1, 13.2, 13.3, 13.4, 14.1, 14.2, 15.1, 15.2, 16.1, 16.2, Current On Demand Version
	Tricentis Tosca	9.0, 9.1, 9.2, 9.3, 10.0
	VersionOne	Enterprise and Ultimate: 13.2, 13.3.3, 14.0.1, 14.0.2, 14.1.10, 14.2,14.3, 15.0, 15.0.2, 15.1, 15.2, 15.3, 16.0, 16.1, 16.2, 16.3, Current On Demand
	Whitehat	Whitehat Integration Server: Provided by Whitehat (please contact Whitehat for information on supported versions)
	Zendesk	Current On Demand Version

ZEPHYR Real-Time Test Management	Zephyr for JIRA	3.2.1 on JIRA 7.0, Current On Demand Version

End of Support Policy