Release Control

Atlassian JIRA Software Plugin Quick Reference

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Serena. Serena assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

License and copyright information for 3rd party software included in this release can be found on the product's news page at http://support.serena.com/ProductNews/default.aspx and may also be found as part of the software download available at http://support.serena.com.

Trademarks

Serena, Dimensions, ChangeMan, Comparex, and StarTool are registered trademarks of Serena Software, Inc. The Serena logo, PVCS, TeamTrack, License Manager and Composer are trademarks of Serena Software, Inc. All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 2345 NW Amberbrook Drive, Suite 200, Hillsboro, OR 97006.

Part number: 3.0

Publication date: 2017-06-12

Table of Contents

Preparing the Plugins

The integration between each integrating product and Release Control is implemented through the corresponding plugin.

Documentation References

- Information on installing, configuring, and using this plugin is included in this document.
- For overview and getting started information on Release Control, see the *Release Control Getting Started Guide*.

For information on preparing this plugin for use, see the following:

- About Plugins [page 4]
- Installing Plugins [page 5]
- Release Control Administration Overview [page 5]
- Adding Plugins [page 6]
- Adding Base Configurations [page 7]
- Adding Plugin Configurations [page 8]
- Updating Plugin Configuration Details [page 9]
- Adding Plugin Configuration Tags [page 10]
- Updating Collections to Use Different Tags [page 11]
- Overriding Configuration Settings [page 11]

About Plugins

Plugins implement integrations with external products to provide core functionality for Release Control. After applying the Release Control configuration in SBM Configurator, all default plugins are already installed for you. You can access Release Control Administration from SBM Configurator to add plugins and one or more unique configurations for each plugin.

Default Plugins

Release Control installs the following plugins as part of the default installation:

- Atlassian JIRA
- CA Nolio
- ChangeMan ZMF *
- Deployment Automation *
- Dimensions CM *

- Jenkins
- SBM *
- ServiceNow

* Example configurations for these are included in the Release Control installation.

Provider Types

Plugins implement object and execution provider types. In the default plugins, object provider type is used for requests and deployment units and execution provider type is used for deployment tasks. Each plugin can implement any number of variations of provider types as needed, so only one plugin is needed per integrating product.

For information on creating custom provider types, see the *Release Control SDK Reference*.

Installing Plugins

Default plugins are installed by the Release Control installer.

If you have created a custom plugin, are using a plugin version that is not installed by default, or are restoring a plugin, you will need to install it.

If the plugin (or version of the plugin) you want to use is already installed, continue with Adding Plugins [page 6].

For information on upgrading plugins, see the latest Plugin Release Notes on Documentation Center.

To install a plugin:

1. Download the plugin jar files that you want to install from the Support website, under the latest version of Release Control.

   The latest plugin file name is as follows:

   ```
   rlc-provider-jira-3.0.jar
   ```

2. For Release Control version 6.2 or later only, copy the jar files to the Release Control plugins installation directory. For example:

   ```
   C:\Program Files\Micro Focus\Release Control\Plugins
   ```

   Important: Plugins version 3.0 and later are supported only for Release Control version 6.2 and later. For installation instructions for earlier plugin versions with earlier versions of Release Control, see the plugin Quick References for those versions.

Release Control Administration Overview

Use Release Control Administration to prepare your plugins for use.

There are several ways to navigate to the same place within the user interface. The procedures in this documentation show one of the ways to navigate. Use the included examples to become familiar with the navigation.

To access Release Control Administration, in your Web browser, enter the URL for Release Control Administration. For example:

```
http://serverName/workcenter/tmtrack.dll?StdPage&Template=rlc/admin
```

Release Control Administration includes the following global menu options:

- **Plugins:** Add plugins that are installed on your system.
- **Configurations:** Add base configurations and configurations created from them.
  - **Base Configurations:**
    - A set of shared properties that enables you to quickly create and update configurations of different types using that information.
    - A base must be created for each set of configurations.
    - Multiple versions of configurations can be added for the same base configuration.
    - You can add multiple execution and object type configurations (called actions and item configurations in the examples) within one base configuration.
  - **Configurations:**
    - A configuration is a specific set of information needed to connect to and integrate with external products using a particular plugin.
    - Configurations are selected from collections to add items such as Requests, Deployment Units, and Deployment Tasks.
    - The first configuration is created at the same time you create the base and you can create as many configurations as you need based on each base.
    - Many fields can be overridden to give more flexibility in creating configurations for different groups or slightly different purposes.
    - If the base is updated and fields are overridden, the updates in the base are ignored.
- **Tags:**
  - Unique names you can assign to plugin configurations in addition to the plugin-defined Type Name.
  - These are referenced in collection widgets to categorize the objects, such as requests, deployment units, and deployment tasks.
  - Several plugin-defined tags are available for the default implementation of Release Control, such as Request, Deployment Unit, and Deployment task tag.

Adding Plugins

Several of the default plugins have been added for you. If you don't see the default plugin you want to use, you can add it. If you have just installed a non-default or custom plugin, you will need to add it.

If the plugin (or version of the plugin) you want to use is already added, continue with Adding Plugin Configurations [page 8].

Important: The Release Control Administration options can be performed only by users with Managed Administrator privileges.

CAUTION:

If you are using Internet Explorer as your web browser, ensure Compatibility View is not set for the Release Control site. Otherwise, you will receive the error "Your browser is not supported."

To add a plugin:

1. In your Web browser, enter the URL for Release Control Administration. For example:

   ```
   http://serverName/workcenter/tmtrack.dll?StdPage&Template=rlc/admin
   ```

   Tip: The link for your installation is in SBM Configurator on the **Release Control** page.

2. In the **Plugins** tab, click the **Add Plugin** icon.

3. In the **Available Plugins** list, select the Atlassian JIRA plugin you want to use .

   Note: If you have multiple versions of the same plugin installed, you can choose an the version to add. By default, the latest one is selected.

Adding Base Configurations

A base plugin configuration must be created for each set of configurations. It stores connection information and other shared properties and enables you to quickly create configurations of different types using that information.

Important: The Release Control Administration options can be performed only by users with Managed Administrator privileges.

To add a base configuration:

1. In your Web browser, enter the URL for Release Control Administration. For example:

   ```
   http://serverName/workcenter/tmtrack.dll?StdPage&Template=rlc/admin
   ```

   Tip: The link for your installation is in SBM Configurator on the **Release Control** page.

2. Select the **Configurations** tab.

3. To create a base with a new set of connection details and shared properties:

   a. Click the **Add Configuration** icon (⚙).

b. Give a unique title and description for the first configuration for the base.

Tip: The configuration title is the name that appears in the drop-down selection when adding requests, deployment units, and deployment tasks. Consider including the server name if you have configurations for multiple servers, integrating product version, plugin version, or any other distinguishing information.

c. In the **Plugin Instance** field, select the Atlassian JIRA plugin .

d. In the **Type Name** field, select the specific implementation of the provider type that you want to use.

e. In the **Base Configuration** field, select **New**.

f. In **Base Configuration Title**, give a unique title for the base configuration that encompasses the scope of the configurations that will inherit its settings.

Tip: Consider including the server name if you have configurations for multiple servers, integrating product version, plugin version, or any other distinguishing information.

g. Fill out the remaining fields presented in the user interface. See JIRA Request Item Configuration Details [page 19] .

Adding Plugin Configurations

Plugin configurations provide connection information, product-specific details, and filters telling the plugin what data to obtain from the integrating product. You may use multiple configurations to point to different systems or projects, to use different versions of an integrating product or plugin, or to use different filters.

Before you can use a plugin, you must at minimum update it with your specific connection information for the configuration. Each of the plugin configurations in the default implementation have example values, but you'll need to update these according to your own implementation.

If you are using one of the plugins with example configurations, you can update the example configurations with your connection information and other details or you can add new configurations.

Important: The Release Control Administration options can be performed only by users with Managed Administrator privileges.

To add a plugin configuration:

1. In your Web browser, enter the URL for Release Control Administration. For example:

```
http://serverName/workcenter/tmtrack.dll?StdPage&Template=rlc/admin
```

Tip: The link for your installation is in SBM Configurator on the **Release Control** page.

2. Select the **Configurations** tab.

3. To create a configuration from an existing base:

   a. Click the + beside the base that you want to use to create the configuration.

   b. Give a unique title and description for the configuration.

      Tip: The configuration title is the name that appears in the drop-down selection when adding requests, deployment units, and deployment tasks. Consider including the server name if you have configurations for multiple servers. You may also want to include integrating product version, plugin version, or other distinguishing information.

   c. In the **Plugin Instance** field, select the Atlassian JIRA plugin .

   d. In the **Type Name** field, select the specific implementation of the provider type that you want to use.

   e. In the **Base Configuration** field, select the base that you want to use for this configuration.

      Tip: Select **New** to create a new base configuration as you create the configuration instead of using an existing configuration. See Adding Base Configurations [page 7].

   f. Fill out the fields presented in the user interface. See JIRA Plugin Configuration Overview [page 19].

   g. To enter different information in any grayed out fields, select **Override**.

Updating Plugin Configuration Details

Some settings can be changed after you have created a configuration.

To update a plugin configuration:

- Edit the base configuration to change shared information for all its subordinate configurations.
- Edit the configurations to change non-base settings.
- Override base settings at the configuration level.
  - You can change a base field, which is shown in gray, only if there is an override option beside it. Select **Override** to make the field editable. If there is no override option, you can create a new configuration.
  - If you override at the configuration level and then update the base configuration, the configuration retains the override rather than inheriting the new base configuration setting.

Note: You can also override fields at the item or project level to filter selections available in collections. See Overriding Configuration Settings [page 11].

For field descriptions, see JIRA Plugin Configuration Overview [page 19]

Adding Plugin Configuration Tags

Tags are unique names you can assign to plugin configurations in addition to the system-assigned Type Name. These tags are used in collection widgets to show or not show the objects, such as requests and deployment units.

Tags are provided for the default plugin configurations and are pre-selected in the example settings. However, you may add any type of object needed for your custom implementations of Release Control and the plugins.

Warning: Do not remove the default tags unless you are customizing the system. Object configurations must have tags assigned to them; otherwise, items cannot be linked to deployment tasks, because the configuration for items does not appear and the selection says "No configurations available".

Note: When you create tags or assign tags to a configuration, those tags are not used by existing items. The tags will be used only for items added after the tags are created or assigned.

Important: The Release Control Administration options can be performed only by users with Managed Administrator privileges.

To add tags to a specific configuration:

1. In your Web browser, enter the URL for the Release Control Administration. For example:

   ```
   http://serverName/workcenter/tmtrack.dll?StdPage&Template=rlc/admin
   ```

   Tip: The link for your installation is in SBM Configurator on the **Release Control** page.

2. In the **Configurations** tab, select a plugin configuration.

   Note: If you add a tag at the configuration level, that tag is available to be selected by other configurations and is listed in the global **Tags** list.

3. Click the **Tags** tab.
4. Click **Edit**.
5. To use existing tags, select one or more tags, use the arrows to move them to the **Selected Tags** box.
6. To create a new tag, click **Create New** below the selection box and give the tag a name and description. This creates the tag and automatically adds it to the **Selected Tags** box.

Tip: You can view tags for all configurations in the global **Tags** tab. From there you can create tags, edit existing tag names and descriptions, and copy tags.

Updating Collections to Use Different Tags

Tags defined in Release Control Administration are used in collections to determine what plugin information to show.

If you are using the default implementation, you do not need to make any changes. However, if you've added tags to use for custom collections or to combine types of information shown in the default collections, you need to modify the collections to use the tags you want.

To update the collections to use different tags:

1. In SBM Composer, open the process app that you want to update, such as *RLC - Release Package* or *RLC - Release Train*.
2. Select **Visual Design** in the App Explorer.
3. Select the form to update, such as **RT Deployment State Form**.
4. In the form, select the tab for the collection you want to update.
5. Select the collection widget.
6. If you are updating an object collection, such as **Requests** or **Deployment Units**, in the Property Editor Parameters **Tags** field, update the list of tags. In execution collections, such as Deployment Tasks, update the **View Tags** field.
7. If you are changing the name of the collection, you should update the tab name to match.
8. Deploy the process app.

For details on updating and deploying process apps, see the SBM Composer documentation.

Overriding Configuration Settings

Configuration field values can be overridden in several ways, depending on your needs. Overrides are available as follows:

- Within a configuration, you can override base settings for fields that have an Override checkbox beside them. See Adding Plugin Configurations [page 8].
  - These overrides are used to limit value selections when adding objects in collections, such as requests, deployment units, and deployment tasks.
  - These overrides are also used by the system for validation and execution.
- Using entries in the *Configuration Overrides* table, you can pre-define sets of field values for specific purposes, such as for a project, an application, or a release.
  - These overrides are used to limit value selections when adding objects in collections, such as requests, deployment units, and deployment tasks.
  - You can define sets of overrides for all configurations or limit them by configuration UUID or by a specified tag.

- You can use dynamic values in the *Configuration Overrides* table to access the exact values for fields set within an item, such as a release package or deployable release train. To dynamically override the field, it must be used on a form.
- These overrides are not used when validating and executing deployment tasks.

CAUTION:

It is not recommended to override connection information this way unless you are doing it only to restrict the selections for objects such as requests. If a user in an override has permission to make selections but the user in the configuration or its internal override does not have enough permissions to execute the deployment, the deployment may fail. The user in the configuration or its internal override must have the permissions necessary for validation and execution.

To override configuration settings:

1. Select the fields for which overriding is allowed. See Making Fields Eligible for Override [page 12].
2. Add entries to the *Configuration Overrides* table. See Creating Configuration Overrides [page 13].
3. Select an override entry in one of the following ways:
   - In a release package or deployable release train, select an entry in the **Configuration Override** field. See *Release Control Getting Started Guide* or online Help, "Creating Release Packages" or "Creating Deployable Release Trains".
   - In SBM Composer, process app designers can select the override entry for the project and hide the field in the process app. This way the overrides are set automatically for each new release package and deployable release train in that project.

For more details, see the following:

- Making Fields Eligible for Override [page 12]
- Creating Configuration Overrides [page 13]

Making Fields Eligible for Override

To override plugin configuration field values in an item or project using a *Configuration Override* auxiliary table entry, the fields must be eligible for override.

Important: The Release Control Administration options can be performed only by users with Managed Administrator privileges.

To select overrides for a configuration:

1. In your Web browser, enter the URL for Release Control Administration. For example:

```
http://serverName/workcenter/tmtrack.dll?StdPage&Template=rlc/admin
```

Tip: The link for your installation is in SBM Configurator on the **Release Control** page.

2. In the **Configurations** tab, select a plugin configuration at the level you want to allow overrides, either base or configuration level.

   If you allow configuration overrides on the base level, the child actions or items configurations inherit these settings. You can also allow overrides on the child level even if they were already set in the base configuration. If you remove an override selection from the base in this case, it will still be allowed on the child level.

3. In the **Details** tab, select **Overrides** from the drop-down.

4. Click **Edit**.

5. Select the fields you want to allow for override and click **Save**.

Note: If you allow overrides at the configuration for an actions configuration but not at the item configuration level, there may be situations where your deployment task cannot be overridden completely due to the associations of items. In this case, it is better to override at the base level so that it encompasses the related actions and item configurations.

Creating Configuration Overrides

Configuration overrides are auxiliary items that can be set up to enable overriding plugin configuration field values at the item or project level. Each configuration override entry can have multiple property value rows, so that a set of field overrides is contained in a single table entry. Once created, the set of configuration overrides can be selected in options for deployable release trains and release packages.

To create a configuration override entry:

1. In Work Center, click the user icon in the upper right of the page, and then select **Manage Data**.

2. In the **Table** field, select *Configuration Override*.

3. Click **Submit**.

   Tip: If you want to see if an override already exists, select any filters needed to limit results and click **Search** to search for existing rules.

4. Enter a title and description.

5. Click **Add Row**.

6. Enter **Property Name**, which is the internal name of the field you want to override.

   Tip: In the plugin configuration **Overrides** view, click the copy icon next to the field name to copy the field's internal name to the clipboard.

7. Enter **Property Value**, which is the value for the field that you want to use instead of the one defined in Release Control Administration.

   Note: For selection values, use *true* for selected and *false* for deselected.

8. Select **Scope** as follows:

   - **Global**: The override is applied to all same-named properties for all configurations.

   - **By UUID**: The override is applied only to a specific configuration. For this, you must specify the UUID of the configuration.

   - **By Tags**: The override is applied only to configurations with a specified tag. You must specify the internal name of the tag.

     Tip: In Release Control Administration in a **Tags** page, click the copy icon beside the tag definition to copy the field internal name to the clipboard.

9. Enter **Scope Value**. For example, if **UUID** is selected for **Scope**, enter the UUID from the configuration to which you want this override to apply.

   Tip: In Release Control Administration in the configuration view, click the copy icon beside the UUID field to copy the UUID to the clipboard. The UUID is not shown on the configuration edit page.

10. Click **Submit**.

Custom Columns

Custom columns enable you to specify which fields from integrating products that you want to reference in Release Control. These are typically used to display information in collection lists, such as requests and deployment units.

Default collection columns and their related custom column table entries are pre-configured in the default installation, so you do not have to do any extra configuration to have a working implementation. However, you can customize them for your implementation.

For information on configuring custom column fields, see the following:

- Updating Custom Column Items [page 14]
- Adding Custom Column Items [page 15]
- Selecting Custom Column Entries [page 16]
- General Fields [page 17]

Updating Custom Column Items

To update custom column auxiliary table entries:

1. In SBM Application Administrator, click **Auxiliary Data**.

2. Select the *RLC Custom Columns* auxiliary table.

3. Select a row and click **Details**.

4. Click **Update**.

5. Update existing or add new field information in **Display Names** and matching **Internal Names** using the built-in fields.

   Tip:

   - Use the general fields for columns that are not provider-specific. See General Fields [page 17].

   - Use the Help on the right side of the update form as a reference for the field names for some of the plugins, such as DA, SBM, Dimensions CM, and ChangeMan ZMF. Copy and paste using the provided **Copy** buttons.

   - For more information on fields specific to your plugin, see the corresponding section in the documentation for your plugin.

   - Use the provided aliases if you are using more than one provider for deployment units or requests and the use similar column data, such as create date. See Aliases for Similar Provider-Specific Fields [page 18].

   - Use extended or custom fields in the plugin configurations to specify any other fields you may want to reference.

   - Specify a width for a column by appending a colon and the width to the Internal name. For example, description:300.

Adding Custom Column Items

To add custom column entries:

1. In SBM Application Administrator, click **Auxiliary Data**.

2. Select the *RLC Custom Columns* auxiliary table.

3. Click **New**.

4. Enter a title and description.

5. Enter **Display Names** and matching **Internal Names** using the built-in fields.

   Tip:

   - Use the general fields for columns that are not provider-specific. See General Fields [page 17].
   - Use the Help on the right side of the update form as a reference for the field names for some of the plugins, such as DA, SBM, Dimensions CM, and ChangeMan ZMF. Copy and paste using the provided **Copy** buttons.
   - For more information on fields specific to your plugin, see the corresponding section in the documentation for your plugin.
   - Use the provided aliases if you are using more than one provider for deployment units or requests and the use similar column data, such as create date. See Aliases for Similar Provider-Specific Fields [page 18].

Selecting Custom Column Entries

The release package item contains the custom column entries that are used to populate its Deployment Unit or Request Collection widget columns.

An administrator should set a default value for this field in the Release Packages project, and typically, non-administrative users should not be able to change the selection. However, should you need to select the custom column entries to use for a particular release package, use the following procedure.

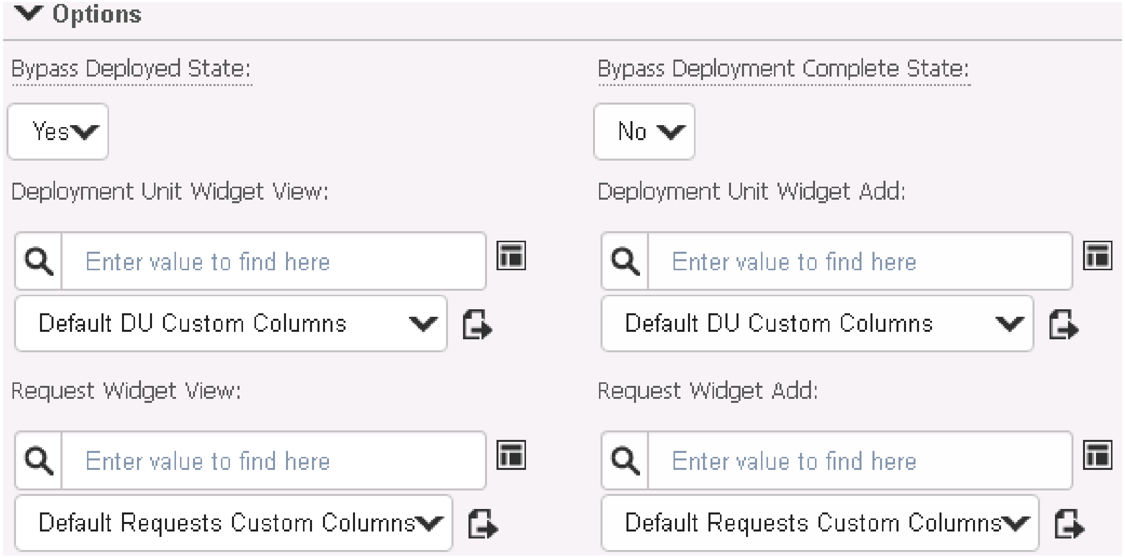
Note: The RLC Custom Column auxiliary table implementation is provided so that you don't have to manually update columns on the forms. However, if you don't want to use this implementation and instead choose to manually enter custom columns as parameters into the widget, remember to update all occurrences of the widget on state forms and transition forms.

To select custom column entries:

1. In Work Center in Release Control, create or update a release package or other item that includes the Deployment Unit or Request Collection widget.
2. In the **Options** section of the submit or edit form you are filling out, find and select the custom column entry you want to use for each of the following fields:

   - Deployment Unit Widget View
   - Deployment Unit Widget Add
   - Request Widget View
   - Request Widget Add

   The default selections are shown in the following figure:



Options

Bypass Deployed State: Yes

Bypass Deployment Complete State: No

Deployment Unit Widget View: Default DU Custom Columns

Deployment Unit Widget Add: Default DU Custom Columns

Request Widget View: Default Requests Custom Columns

Request Widget Add: Default Requests Custom Columns

General Fields

Following are the general custom column fields, which are common across providers:

```
id
artifact_id
request_id
description
name
provider_instance_uuid
provider_instance_name
syncattempteddatetimestring
syncsucceededdatetimestring
syncmessage
syncstatus
title
type
url
```

General Fields for View Widgets Only

Some of the general fields can be shown only in the request and deployment unit *view* widgets, because they are populated on the Release Control side rather than on the provider side. If you use these fields in the request or deployment unit *add* widget, you should create a separate *RLC Custom Column* table entry for the add widget rather than sharing the column entry for the view and add widgets; otherwise, these fields will be empty. These fields are as follows:

```
artifact_id                    request_id
provider_instance_uuid         provider_instance_name
syncattempteddatetimestring    syncsucceededdatetimestring
syncmessage                    syncstatus
```

Synchronization Fields for Reload Data Transitions

Synchronization data appears in the columns for a request or deployment unit widget only if **Reload Request Data** or **Reload Deployment Unit Data** transitions are executed. (By default, this is done in release packages on the **Requests** or **Deployment Unit** tabs, respectively.) The sync fields are populated as follows:

- `syncattempteddatetimestring`: shows the last time a reload was done
- `syncmessage`: if the reload failed, shows the error received from the provider
- `syncstatus`: either FAILED or SUCCEEDED
- `syncsucceededdatetimestring`: shows the last time a reload was successful

For information on the reload transitions, see "Reloading Request Data" and "Reloading Deployment Unit Data" in the *Release Control Getting Started Guide*.

Aliases for Similar Provider-Specific Fields

Aliases are a type of general field that should be used if you are using multiple providers for deployment units or requests. Alias fields should be used instead of the related provider-specific field to avoid having the same type of data appear in separate columns. Following are the aliases and some of the provider-specific fields that they include. See the documentation for your plugin for more information.

Internal Names	Display Names	Included Provider-Specific Fields
createdBy	Created By	ZMF/SDA component version creator, SDA snapshot user, Dimensions CM originator, SBM Request createdBy, ServiceNow sys_created_by
createdOn	Created On	ZMF createDate, SDA created, Dimensions CM creationDate, SBM Request createdDate, ServiceNow sys_created_on
owner	Owner	SBM owner, Jira assignee, ServiceNow assigned_to
updatedBy	Updated By	Dimensions CM lastUpdatedBy, SBM request modifiedBy, ServiceNow sys_updated_by
updatedOn	Updated On	Dimensions CM updateDate, SBM request modifiedDate, ServiceNow sys_updated_on
type	Type	SBM activeinactive, Jira issuetype

Plugin Custom or Extended Fields

Some plugins include custom fields, extended fields, or custom attribute fields that enable you to specify fields that you want to reference. The typical use for this is to specify them in custom column entries in addition to all of the other types of fields available.

JIRA Plugin Configuration Overview

Through the Atlassian JIRA plugin, Release Control requests are associated with JIRA items, such as requests, and Release Control deployment tasks execute JIRA actions, such as checking the status of an existing JIRA item.

Before you can use the JIRA plugin, administrators must do the following:

- Install and add the plugin. See Preparing the Plugins [page 4].
- Add one or more configurations of the plugin. See Adding Plugin Configurations [page 8] and the configuration details topic for the configuration you plan to use.
- Configure custom column fields to display appropriate data. See Fields for Atlassian JIRA Software [page 20].

When associating JIRA items, you have to option to create them.

See the following for more details.

- JIRA Request Item Configuration Details [page 19]
- JIRA Request Actions Configuration Details [page 20]
- Fields for Atlassian JIRA Software [page 20]

JIRA Request Item Configuration Details

A JIRA Request Item configuration enables you to associate items in JIRA projects with Release Control requests.

Field Descriptions

The field descriptions are included in the UI. While creating or editing a base or configuration, point to a field name to view its description.

A few fields that may need additional explanation are described in the following table.

Field	Description	Example Values
JIRA URL	URL for the JIRA server	http://JiraHost:8080/
Project Filter	Filter used to limit the list of projects available to choose from on the search form. If empty, all projects will be returned; otherwise, only projects that match the filter will be available to search. The filter accepts text with a single character wild card (?) and a multiple character wild card (*).	Micro Focus R*
Issue Type Filter	Comma-delimited list of issue types to be displayed on the search form	Improvement, Task, New Feature, Bug, Epic

Field	Description	Example Values
Result Limit	Maximum number of results to be returned in a search	300

JIRA Request Actions Configuration Details

A JIRA Request Actions configuration enables you to execute JIRA tasks, such as checking item status, through Release Control deployment tasks.

Field Descriptions

The field descriptions are included in the UI. While creating or editing a base or configuration, point to a field name to view its description.

A few fields that may need additional explanation are described in the following table.

Field	Description	Example Values
JIRA URL	URL for the JIRA server	http://JiraHost:8080/
Project Filter	Filter used to limit the list of projects available to choose from on the search form. If empty, all projects will be returned; otherwise, only projects that match the filter will be available to search. The filter accepts text with a single character wild card (?) and a multiple character wild card (*).	Micro Focus R*
Issue Type Filter	Comma delimited list of issue types to be displayed on the search form	Improvement, Task, New Feature, Bug, Epic
Result Limit	Maximum number of results to be returned in a search	300

Fields for Atlassian JIRA Software

If you are using the JIRA plugin for requests, following are the custom column fields specific to this provider. For information on configuring them, see Custom Columns [page 14].

Display Names	Internal Names
Project, Assignee, Status, Priority, Issue Type, Reporter, Created, Updated, Due Date	project, assignee, status, priority, issuetype, reporter, created, updated, duedate

Aliases

Following are the aliases that can be used for similar fields when you are using multiple plugins for requests. If you are using only JIRA for requests, you do not need to use these.

JIRA Plugin Field Names	Alias Field Names
Assignee / assignee	Owner / owner
Type / issuetype	Type / type
Reporter / reporter	Created By / createdby
Created / created	Created On / createdOn
Updated / updated	Updated On / updatedOn

Using the Atlassian JIRA Software Plugin

After configuration is complete, you can do the following in Release Control:

- Add JIRA requests to release packages.
  - Projects displayed on the form are limited to those that match the project filter.
  - Status filters displayed are those configured on the configuration.
  - Date/time fields use your SBM locale.
  - Searches are limited to your search criteria.
- View JIRA requests that have been added.
  - View requests in the **Requests** tab.
  - Open the item in JIRA by clicking the name link.
- Synchronize request data from JIRA using the **Reload Request Data** button.

For more information, see the following:

- Associating JIRA Items with Requests [page 21]
- Creating JIRA Requests [page 23]
- Configuring Tasks to Assign JIRA Requests [page 24]
- Configuring Tasks to Transition JIRA Requests [page 25]

Associating JIRA Items with Requests

After a JIRA Request Item configuration has been added or updated as needed, you can associate Atlassian JIRA items, as requests, with release packages.

To associate JIRA items with requests:

1. In a deployable release train or release package, select the **Requests** tab.
2. Click **Edit Requests**.
3. Click the **+** icon to add a request.
4. Select the JIRA Item configuration.
5. Select **Search for items**.

Enter search criteria and then click **Find Items** to find and select the Atlassian JIRA items to associate with the request.

The fields that are displayed when searching are described in the following table.

Field	**Description**	**Example Values**
Project	Limits the results to the selected projects; items in the list should match the Project Filter configuration option	My JIRA Project
Status Filter	Limits the results to issues that are in one of the selected statuses	To Do, In Progress, In Review, Done
Title Filter	Limits the results to issues that have a summary that matches the search criteria entered. For incomplete words, you must enter wildcard characters using * for multiple-character matches and ? for single-character matches.	
Issue Type Filter	Select a JIRA Issue Type filter	
Assignee Filter	Select a JIRA Assignee filter	
Original Estimate	The original estimate of how much work is involved in resolving this issue. For example: 2, 3w 4d 12h 5m	
Remaining Estimate	An estimate of how much work remains until this issue will be resolved. For example: 2, 3w 4d 12h 5m	

Creating JIRA Requests

If a JIRA request that you want to associate with a Release Control request or configure an action for does not yet exist, you can create it while adding or linking the request.

Note: You can also create a request from the **Item** link option while adding or editing a deployment task.

To create JIRA a request while adding or editing a Release Control request:

1. In a deployable release train or release package, select the **Requests** tab.
2. Click **Edit Requests**.
3. Click the **+** icon to add a request.
4. Select the JIRA Request Item configuration.
5. Select **Create item**.
6. Fill out the resulting fields.
7. Click **Create & Add**.

The fields that are displayed when creating JIRA requests are described in the following table.

Field	Description	Example Values
Project	Select the JIRA project in which you want to create the request.	My JIRA Project
Issue Type	Select from the available issue types. These are limited by those included in the configuration. See JIRA Request Item Configuration Details [page 19]. Additional fields appear based on this selection.	Improvement
Linked Issues	Select the relationship a linked issue has to this issue. For example: • is blocked by • blocks • is duplicated by • relates to	is blocked by
Issue	Use the search icon to search for the issue to link to the request, select it in the resulting page, and click **Select** to return to the Create item page.	

Configuring Tasks to Assign JIRA Requests

After a JIRA Request Actions configuration has been added or updated as needed, you can configure deployment tasks to assign JIRA requests.

To assign JIRA requests:

1. In a deployable release train or release package, select the **Deployment Tasks** tab.
2. Click **Edit Deployment Tasks**.
3. Click the **+** icon to add a request.
4. Select the Atlassian JIRA request provider name.
5. Select **Assign**.
6. Fill out the resulting fields.

The fields that are displayed when configuring a deployment task to assign JIRA requests are described in the following table.

Field	Description	Example Values
Item	Click the link icon to select a request. From here, you can: **Use available items**: Select requests that have already been added. **Use available configurations**: Search for existing requests or create new ones. This uses the JIRA Item configuration. See Associating JIRA Items with Requests [page 21] and Creating JIRA Requests [page 23].	**Note:** If you select more than one item in the search results, only the first one will be associated with the deployment task.
Environment	Select each environment in which you want to execute the task. These are the environments configured in the deployment path. For example, Dev, Test, and Prod.	
Assignee	Select an assignee.	
Comment	Enter a comment.	

Field	Description	Example Values
Schedule Options	Select the schedule option and fill out the additional fields that open based on your selection. Options include the following: • None • Specific Time • Opportunity Window • Time Window • Delay • Hold Until Released	

Configuring Tasks to Transition JIRA Requests

After a JIRA Request Actions configuration has been added or updated as needed, you can configure deployment tasks to transition JIRA requests.

To transition JIRA requests:

1. In a deployable release train or release package, select the **Deployment Tasks** tab.
2. Click **Edit Deployment Tasks**.
3. Click the **+** icon to add a request.
4. Select the Atlassian JIRA request provider name.
5. Select **Do transition**.
6. Fill out the resulting fields.

The fields that are displayed when configuring a deployment task to assign JIRA requests are described in the following table.

Field	Description	Example Values
Item	Click the link icon to select a request. From here, you can: **Use available items**: Select requests that have already been added. **Use available configurations**: Search for existing requests or create new ones. This uses JIRA Item configuration. See Associating JIRA Items with Requests [page 21] and Creating JIRA Requests [page 23].	

Field	Description	Example Values
Environment	Select the environment in which you want to schedule this action.	
Transition	Select a transition. For example: • To Do • In Progress • Done • Closed	
Schedule Options	Select the schedule option and fill out the additional fields that open based on your selection. Options include the following: • None • Specific Time • Opportunity Window • Time Window • Delay • Hold Until Released	

Plugins Glossary

Artifacts

Files or other software elements. In Deployment Automation, these are represented by component versions.

Base Configurations

A base plugin configuration must be created for each set of configurations. It stores connection information and other shared properties and enables you to quickly create configurations of different types using that information. Many fields can be overridden to give more flexibility in creating configurations for different groups or slightly different purposes.

Baselines

In Dimensions CM, containers for release-ready components or artifacts. For more details, refer to the Dimensions CM documentation.

Change Packages

In ChangeMan ZMF, containers for release-ready components. For more details, see the ChangeMan ZMF documentation.

Component Versions

Instances of a Deployment Automation component, which contain the artifacts to be deployed or otherwise configured.

Configuration Overrides

You can override field values when creating or updating a configuration. You can also override the configuration settings at the SBM item level using auxiliary table override values. See Overriding Configuration Settings [page 11].

Configurations

Provide connection information, product-specific details, and filters telling the plugin what data to obtain from the integrating product.

Plugin Instances

Variations of a plugin that may represent different versions of the plugin or instances owned by different business units who want to maintain an independent maintenance schedule.

Plugins

Plugins are used to implement the integrations between Release Control and other products.

Projects

In SBM, containers for release items, such as application releases, release trains, and environments; SBM projects.

Single Sign-on (SSO)

Serena-installed software that enables a user to log in to a Web-based component of SBM and be recognized on subsequent accesses to that component or other Web-based components of SBM. This software also provides the ability for security tokens to be used in an orchestration, allowing Web services to be called without requiring the user to provide credentials at inconvenient times.

Snapshots

A Deployment Automation snapshot captures an application's current state, and as the application moves through different environments, the snapshot ensures that proper component versions are used.

Tags

Tags are unique names you can assign to plugin configurations in addition to the system-assigned Type Name. These are referenced in collection widgets to categorize the objects, such as requests, deployment units, and deployment tasks.

Type Name

Type indicates the type of configuration based whether the configuration is a base configuration or one of the provider types, integration entity (item) or execution (action). Type Name gives the specific instance name of the base configuration or provider type.