



# **Red Hat Process Automation Manager 7.6**

**Designing and building cases for case  
management**



# Red Hat Process Automation Manager 7.6 Designing and building cases for case management

---

Red Hat Customer Content Services  
brms-docs@redhat.com

## Legal Notice

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This document explains case management concepts in Red Hat Process Automation Manager 7.6.

## Table of Contents

<b>PREFACE</b> .....	<b>3</b>
<b>CHAPTER 1. CASE MANAGEMENT</b> .....	<b>4</b>
<b>CHAPTER 2. CASE MANAGEMENT MODEL AND NOTATION</b> .....	<b>5</b>
<b>CHAPTER 3. CASE FILES</b> .....	<b>6</b>
<b>CHAPTER 4. SUBCASES</b> .....	<b>7</b>
<b>CHAPTER 5. AD HOC AND DYNAMIC TASKS</b> .....	<b>10</b>
<b>CHAPTER 6. ADDING DYNAMIC TASKS AND PROCESSES TO A CASE USING THE API</b> .....	<b>11</b>
6.1. CREATING A DYNAMIC USER TASK USING THE REST API	12
6.2. CREATING A DYNAMIC SERVICE TASK USING THE REST API	13
6.3. CREATING A DYNAMIC SUBPROCESS USING THE REST API	15
<b>CHAPTER 7. CASE ROLES</b> .....	<b>17</b>
7.1. CREATING CASE ROLES	18
7.2. ROLE AUTHORIZATION	19
7.3. ASSIGNING A TASK TO A ROLE	20
7.4. MODIFYING CASE ROLE ASSIGNMENTS DURING RUN TIME USING SHOWCASE	22
7.5. MODIFYING CASE ROLE ASSIGNMENTS DURING RUN TIME USING REST API	24
<b>CHAPTER 8. STAGES</b> .....	<b>27</b>
8.1. DEFINING A STAGE	27
8.2. CONFIGURING STAGE ACTIVATION AND COMPLETION CONDITIONS	28
8.3. ADDING A DYNAMIC TASK TO A STAGE	29
<b>CHAPTER 9. MILESTONES</b> .....	<b>31</b>
9.1. MILESTONE CONFIGURATION AND TRIGGERING	31
<b>CHAPTER 10. CASE EVENT LISTENER</b> .....	<b>34</b>
<b>CHAPTER 11. RULES IN CASE MANAGEMENT</b> .....	<b>36</b>
11.1. USING RULES TO DRIVE CASES	36
<b>CHAPTER 12. CASE MANAGEMENT SECURITY</b> .....	<b>40</b>
12.1. CONFIGURING SECURITY FOR CASE MANAGEMENT	40
<b>CHAPTER 13. CLOSING CASES</b> .....	<b>42</b>
13.1. CLOSING A CASE USING THE PROCESS SERVER REST API	42
13.2. CLOSING A CASE IN THE SHOWCASE APPLICATION	42
<b>CHAPTER 14. CANCELING OR DESTROYING A CASE</b> .....	<b>44</b>
<b>CHAPTER 15. ADDITIONAL RESOURCES</b> .....	<b>45</b>
<b>APPENDIX A. VERSIONING INFORMATION</b> .....	<b>46</b>



---

# PREFACE

As a developer, you can use Business Central to configure Red Hat Process Automation Manager assets for case management.

Case management differs from Business Process Management (BPM). It focuses more on the actual data being handled throughout the case rather than on the sequence of steps taken to complete a goal. Case data is the most important piece of information in automated case handling, while business context and decision-making are in the hands of the human case worker.

Red Hat Process Automation Manager includes the **IT\_Orders** sample project in Business Central. This document refers to the sample project to explain case management concepts and provide examples.

The [Getting started with case management](#) tutorial describes how to create and test a new IT\_Orders project in Business Central. After reviewing the concepts in this guide, follow the procedures in the tutorial to ensure that you are able to successfully create, deploy, and test your own case project.

## Prerequisites

- Red Hat JBoss Enterprise Application Platform 7.2 is installed. For information about installing Red Hat JBoss Enterprise Application Platform 7.2, see [Red Hat JBoss Enterprise Application Platform 7.2 Installation Guide](#).
- Red Hat Process Automation Manager is installed. For information about installing Red Hat Process Automation Manager, see [Planning a Red Hat Process Automation Manager installation](#).
- Red Hat Process Automation Manager is running and you can log in to Business Central with the **user** role. For information about users and permissions, see [Planning a Red Hat Process Automation Manager installation](#).
- The Showcase application is deployed. For information about how to install and log in to the Showcase application, see [Using the Showcase application for case management](#).

## CHAPTER 1. CASE MANAGEMENT

Case management is an extension of Business Process Management (BPM) that enables you to manage adaptable business processes.

BPM is a management practice used to automate tasks that are repeatable and have a common pattern, with a focus on optimization by perfecting a process. Business processes are usually modeled with clearly defined paths leading to a business goal. This requires a lot of predictability, usually based on mass-production principles. However, many real-world applications cannot be described completely from start to finish (including all possible paths, deviations, and exceptions). Using a process-oriented approach in certain cases can lead to complex solutions that are hard to maintain.

Case management provides problem resolution for non-repeatable, unpredictable processes as opposed to the efficiency-oriented approach of BPM for routine, predictable tasks. It manages one-off situations when the process cannot be predicted in advance. A case definition usually consists of loosely coupled process fragments that can be connected directly or indirectly to lead to certain milestones and ultimately a business goal, while the process is managed dynamically in response to changes that occur during run time.

In Red Hat Process Automation Manager, case management includes the following core process engine features:

- Case file instance
- A per case runtime strategy
- Case comments
- Milestones
- Stages
- Ad hoc fragments
- Dynamic tasks and processes
- Case identifier (correlation key)
- Case lifecycle (close, reopen, cancel, destroy)

A case definition is always an ad hoc process definition and does not require an explicit start node. The case definition is the main entry point for the business use case.

A process definition is introduced as a supporting construct of the case and can be invoked either as defined in the case definition or dynamically to bring in additional processing when required. A case definition defines the following new objects:

- Activities (required)
- Case file (required)
- Milestones
- Roles
- Stages

## CHAPTER 2. CASE MANAGEMENT MODEL AND NOTATION

You can use Business Central to import, view, and modify the content of Case Management Model and Notation (CMMN) files. When authoring a project, you can import your case management model and then select it from the asset list to view or modify it in a standard XML editor.

The following CMMN constructs are currently available:

- Tasks (human task, process task, decision task, case task)
- Discretionary tasks (same as above)
- Stages
- Milestones
- Case file items
- Sentries (entry and exit)

The following tasks are not supported:

- Required
- Repeat
- Manual activation

Sentries for individual tasks are limited to entry criteria while entry and exit criteria are supported for stages and milestones. Decision tasks map by default to a DMN decision. Event listeners are not supported.

Red Hat Process Automation Manager does not provide any modeling capabilities for CMMN and focuses solely on the execution of the model.

## CHAPTER 3. CASE FILES

A case instance is a single instance of a case definition and encapsulates the business context. All case instance data is stored in the case file, which is accessible to all process instances that might participate in the particular case instance. Each case instance and its case file are completely isolated from the other cases. Only case instance participants can access the case file.

A case file is used in case management as a repository of data for the entire case instance. It contains all roles, data objects, the data map, and any other data. The case can be closed and reopened at a later date with the same case file attached. A case instance can be closed at any time and does not require a specific resolution to be completed.

The case file can also include embedded documentation, references, PDF attachments, web links, and other options.

The **Case ID prefix** is an optional configurable prefix that enables you to easily distinguish different types of cases. The prefix is followed by a generated ID in the format **ID-XXXXXXXXXX**, where **XXXXXXXXXX** is a generated number that provides a unique ID for the case instance. If a prefix is not provided, the default prefix is **CASE** with the following identifiers:

**CASE-0000000001**

**CASE-0000000002**

**CASE-0000000003**

You can specify any prefix. For example, if you specify the prefix **IT**, the following identifiers are generated:

**IT-0000000001**

**IT-0000000002**

**IT-0000000003**

---

## CHAPTER 4. SUBCASES

Subcases provide the flexibility to compose complex cases that consist of other cases. This means that you can split large and complex cases into multiple layers of abstraction and even multiple case projects. This is similar to splitting a process into multiple subprocesses.

A subcase is another case definition that is invoked from within another case instance or a regular process instance. It has all of the capabilities of a regular case instance:

- It has a dedicated case file.
- It is isolated from any other case instance.
- It has its own set of case roles.
- It has its own case prefix.

You can use the process designer to add subcases to your case definition. A subcase is a case within your case project, similar to having a subprocess within your process. Subcases can also be added to a regular business process. Doing this enables you to start a case from within a process instance.

For more information about adding a subcase to your case definition, see [Getting started with case management](#).

The **Sub Case Data I/O** window supports the following set of input parameters that enable you to configure and start the subcase:

## Sub Case Data I/O



## Data Inputs and Assignments

[+ Add](#)

Name	Data Type	Source	
UserRole_	String ▼	▼	
Independent	String ▼	▼	
GroupRole_	String ▼	▼	
DestroyOnAbort	String ▼	▼	
DataAccess_	String ▼	▼	
DeploymentId	String ▼	▼	
Data_	String ▼	▼	
CaseDefinitionId	String ▼	▼	

## Data Outputs and Assignments

[+ Add](#)

Name	Data Type	Target	
CaseId	String ▼	▼	

Cancel [Save](#)**Independent**

Optional indicator that tells the process engine whether or not the case instance is independent. If it is independent, the main case instance does not wait for its completion. The value of this property is **false** by default.

**GroupRole\_XXX**

Optional group to case role mapping. The role names belonging to this case instance can be referenced here, meaning that participants of the main case can be mapped to participants of the subcase. This means that the group assigned to the main case is automatically assigned to the subcase, where **XXX** is the role name and the value of the property is the value of the group role assignment.

**DataAccess\_XXX**

Optional data access restrictions where **XXX** is the name of the data item and the value of the property is the access restrictions.

**DestroyOnAbort**

Optional indicator that tells the process engine whether to cancel or destroy the subcase when the subcase activity is aborted. The default value is **true**.

**UserRole\_XXX**

Optional user to case role mapping. You can reference the case instance role names here, meaning that an owner of the main case can be mapped to an owner of the subcase. The person assigned to the main case is automatically assigned to the subcase, where **XXX** is the role name and the value of the property is the value of the user role assignment.

**Data\_XXX**

Optional data mapping from this case instance or business process to a subcase, where **XXX** is the name of the data in the subcase being targeted. This parameter can be provided as many times as needed.

**DeploymentId**

Optional deployment ID (or container ID in the context of Process Server) that indicates where the targeted case definition is located.

**CaseDefinitionId**

The mandatory case definition ID to be started.

**CaseId**

The case instance ID of the subcase after it is started.

## CHAPTER 5. AD HOC AND DYNAMIC TASKS

You can use case management to carry out tasks ad hoc, rather than following a strict end-to-end process. You can also add tasks to a case dynamically during run time.

Ad hoc tasks are defined in the case modeling phase. Ad hoc tasks that are not configured as **Adhoc autostart** are optional and might not be used during a case. Therefore, they must be triggered by a signal event or by a Java API.

Dynamic tasks are defined during the case execution and are not present in the case definition model. Dynamic tasks address specific needs that arise during the case. They can be added to the case and worked on at any time using a case application, as demonstrated in the Red Hat Process Automation Manager Showcase application. Dynamic tasks can also be added by Java and Remote API calls.

Dynamic tasks can be user or service activities, while ad hoc tasks can be any type of task. For more information about task types, see "BPMN2 tasks in process designer" in [Designing business processes in Business Central](#).

Dynamic processes are any reusable sub-process from a case project.

Ad hoc nodes with no incoming connections are configured in the node's **Adhoc autostart** property and are triggered automatically when the case instance is started.

Ad hoc tasks are optional tasks that are configured in a case definition. Because they are ad hoc, they must be triggered in some way, usually by a signal event or Java API call.

## CHAPTER 6. ADDING DYNAMIC TASKS AND PROCESSES TO A CASE USING THE API

You can add dynamic tasks and processes to a case during run time to address unforeseen changes that can occur during the lifecycle of a case. Dynamic activities are not defined in the case definition and therefore they cannot be signaled the way that a defined ad hoc task or process can.

You can add the following dynamic activities to a case:

- User tasks
- Service tasks (any type that is implemented as a work item)
- Reusable subprocesses

Dynamic user and service tasks are added to a case instance and immediately executed. Depending on the nature of a dynamic task, it might start and wait for completion (user task) or directly complete after execution (service task). For dynamic subprocesses, the process engine requires a KJAR containing the process definition for that dynamic process to locate the process by its ID and execute it. This subprocess belongs to the case and has access to all of the data in the case file.

You can use the Swagger REST API application to create dynamic tasks and subprocesses.

### Prerequisites

- You are logged in to Business Central and a case instance has been started using the Showcase application. For more information about using Showcase, see [Using the Showcase application for case management](#).

### Procedure

1. In a web browser, open the following URL:  
**/http://localhost:8080/kie-server/docs**
2. Open the list of available endpoints under **Case instances :: Case Management**
3. Locate the **POST** method endpoints for creating dynamic activities.  
**POST /server/containers/{id}/cases/instances/{caseId}/tasks**

Adds a dynamic task (user or service depending on the payload) to case instance.

**POST /server/containers/{id}/cases/instances/{caseId}/stages/{caseStageId}/tasks**

Adds a dynamic task (user or service depending on the payload) to specific stage within the case instance.

**POST /server/containers/{id}/cases/instances/{caseId}/processes/{pId}**

Adds a dynamic subprocess identified by the process ID to case instance.

**POST**

**/server/containers/{id}/cases/instances/{caseId}/stages/{caseStageId}/processes/{pId}**

Adds a dynamic subprocess identified by process ID to stage within a case instance.

<b>POST</b>	<code>/server/containers/{id}/cases/instances/{caseId}/tasks</code> Adds dynamic task (user or service depending on the payload) to case instance
<b>POST</b>	<code>/server/containers/{id}/cases/instances/{caseId}/stages/{caseStageId}/tasks</code> Adds dynamic task (user or service depending on the payload) to given stage within case instance
<b>POST</b>	<code>/server/containers/{id}/cases/instances/{caseId}/processes/{pId}</code> Adds dynamic subprocess identified by process id to case instance
<b>POST</b>	<code>/server/containers/{id}/cases/instances/{caseId}/stages/{caseStageId}/processes/{pId}</code> Adds dynamic subprocess identified by process id to stage within case instance

- To open the documentation, click the REST endpoint required to create the dynamic task or process.
- Click **Try it out** and enter the parameters and body required to create the dynamic activity.
- Click **Execute** to create the dynamic task or subprocess using the REST API.

## 6.1. CREATING A DYNAMIC USER TASK USING THE REST API

You can create a dynamic user task during case run time using the REST API. To create a dynamic user task, you must provide the following information:

- Task name
- Task subject (optional, but recommended)
- Actors or groups (or both)
- Input data

Use the following procedure to create a dynamic user task for the **IT\_Orders** sample project available in Business Central using the Swagger REST API tool. The same endpoint can be used for REST API without Swagger.

### Prerequisites

- You are logged in to Business Central and an IT Orders case instance has been started using the Showcase application. For more information about using Showcase, see [Using the Showcase application for case management](#).

### Procedure

- In a web browser, open the following URL:  
**`/http://localhost:8080/kie-server/docs`**.
- Open the list of available endpoints under **Case instances :: Case Management**
- Click the following **POST** method endpoint to open the details:  
**`/server/containers/{id}/cases/instances/{caseId}/tasks`**
- Click **Try it out** and then input the following parameters:

**Table 6.1. Parameters**

Name	Description
<b>id</b>	<b>itorders</b>

Name	Description
<b>caseld</b>	<b>IT-0000000001</b>

body

```
{
  "name" : "RequestManagerApproval",
  "data" : {
    "reason" : "Fixed hardware spec",
    "caseFile_hwSpec" : "#{caseFile_hwSpec}"
  },
  "subject" : "Ask for manager approval again",
  "actors" : "manager",
  "groups" : ""
}
```

- In the Swagger application, click **Execute** to create the dynamic task.

This procedure creates a new user task associated with case **IT-000000001**. The task is assigned to the person assigned to the **manager** case role. This task has two input variables:

- **reason**
- **caseFile\_hwSpec**: defined as an expression to allow run time capturing of a process or case data.

Some tasks include a form that provides a user-friendly UI for the task, which you can locate by task name. In the IT Orders case, the **RequestManagerApproval** task includes the form **RequestManagerApproval-taskform.form** in its KJAR.

After it is created, the task appears in the assignee's **Task Inbox** in Business Central.

## 6.2. CREATING A DYNAMIC SERVICE TASK USING THE REST API

Service tasks are usually less complex than user tasks, although they might need more data to execute properly. Service tasks require the following information:

- **name**: The name of the activity
- **nodeType**: The type of node that will be used to find the work item handler
- **data**: The map of the data to properly deal with execution

During case run time, you can create a dynamic service task with the same endpoint as a user task, but with a different body payload.

Use the following procedure using the Swagger REST API to create a dynamic service task for the **IT\_Orders** sample project available in Business Central. You can use the same endpoint for REST API without Swagger.

### Prerequisites

- You are logged in to Business Central and an IT Orders case instance has been started using the Showcase application. For more information about using Showcase, see [Using the Showcase application for case management](#).

## Procedure

- In a web browser, open the following URL:  
**/http://localhost:8080/kie-server/docs**
- Open the list of available endpoints under **Case instances :: Case Management**
- Click the following **POST** method endpoint to open the details:  
**/server/containers/{id}/cases/instances/{caseId}/stages/{caseStageId}/tasks**
- Click **Try it out** and then enter the following parameters:

Table 6.2. Parameters

Name	Description
<b>id</b>	<b>itorders</b>
<b>caseId</b>	<b>IT-0000000001</b>

### body

```
{
  "name": "InvokeService",
  "data": {
    "Parameter": "Fixed hardware spec",
    "Interface": "org.jbpm.demo.itorders.services.ITOrderService",
    "Operation": "printMessage",
    "ParameterType": "java.lang.String"
  },
  "nodeType": "Service Task"
}
```

- In the Swagger application, click **Execute** to create the dynamic task.

In this example, a Java-based service is executed. It consists of an interface with the public class **org.jbpm.demo.itorders.services.ITOrderService** and the public **printMessage** method with a single **String** argument. When executed, the parameter value is passed to the method for execution.

Numbers, names, and other types of data given to create service tasks depend on the implementation of a service task's handler. In the example provided, the **org.jbpm.process.workitem.bpmn2.ServiceTaskHandler** handler is used.



## NOTE

For any custom service tasks, ensure the handler is registered in the deployment descriptor in the **Work Item Handlers** section, where the name is the same as the **nodeType** used for creating a dynamic service task.

## 6.3. CREATING A DYNAMIC SUBPROCESS USING THE REST API

When creating a dynamic subprocess, only optional data is provided. There are no special parameters as there are when creating dynamic tasks.

The following procedure describes how to use the Swagger REST API to create a dynamic subprocess task for the **IT\_Orders** sample project available in Business Central. The same endpoint can be used for REST API without Swagger.

### Prerequisites

- You are logged in to Business Central and an IT Orders case instance has been started using the Showcase application. For more information about using Showcase, see [Using the Showcase application for case management](#).

### Procedure

- In a web browser, open the following URL:  
**/http://localhost:8080/kie-server/docs.**
- Open the list of available endpoints under **Case instances :: Case Management**
- Click the following **POST** method endpoint to open the details:  
**/server/containers/{id}/cases/instances/{caseId}/processes/{pld}**
- Click **Try it out** and enter the following parameters:

**Table 6.3. Parameters**

Name	Description
<b>id</b>	<b>itorders</b>
<b>caseId</b>	<b>IT-0000000001</b>
<b>pld</b>	<b>itorders-data.place-order</b>

The **pld** is the process ID of the subprocess to be created.

**body**

```
{
  "placedOrder" : "Manually"
}
```

- In the Swagger application, click **Execute** to start the dynamic subprocess.

In this example, the **place-order** subprocess has been started in the IT Orders case with the case ID **IT-0000000001**. You can see this process in Business Central under **Menu → Manage → Process Instances**.

If the described example has executed correctly, the **place-order** process appears in the list of process instances. Open the details of the process and note that the correlation key for the process includes the

IT Orders case instance ID, and the **Process Variables** list includes the variable **placedOrder** with the value **Manually**, as delivered in the REST API body.

## CHAPTER 7. CASE ROLES

Case roles provide an additional layer of abstraction for user participation in case handling. Roles, users, and groups are used for different purposes in case management.

### Roles

Roles drive the authorization for a case instance and are used for user activity assignments. A user or one or more groups can be assigned to the owner role. The owner is whoever the case belongs to. Roles are not restricted to a single set of people or groups as part of a case definition. Use roles to specify task assignments instead of assigning a specific user or group to a task assignment to ensure that the case remains dynamic.

### Groups

A group is a collection of users who are able to carry out a particular task or have a set of specified responsibilities. You can assign any number of people to a group and assign any group to a role. You can add or change members of a group at any time. Do not hard code a group to a particular task.

### Users

A user is an individual who can be given a particular task when you assign them a role or add them to a group.



#### NOTE

Do not create a user called **unknown** in process engine or Process Server. The **unknown** user account is a reserved system name with superuser access. The **unknown** user account performs tasks related to the SLA violation listener when there are no users logged in.

The following example illustrates how the preceding case management concepts apply to a hotel reservation with:

- Role = **Guest**
- Group = **Receptionist, Maid**
- User = **Marilyn**

The **Guest** role assignment affects the specific work of the associated case and is unique to all case instances. The number of users or groups that can be assigned to a role is limited by the **Case Cardinality**, which is set during role creation in the process designer and case definition. For example, the hotel reservation case has only one guest while the **IT\_Orders** sample project has two suppliers of IT hardware.

When roles are defined, ensure that roles are not hard-coded to a single set of people or groups as part of case definition and that they can differ for each case instance. This is why case role assignments are important.

Role assignments can be assigned or removed when a case starts or at any time when a case is active. Although roles are optional, use roles in case definitions to maintain an organized workflow.



## IMPORTANT

Always use roles for task assignments instead of actual user or group names. This ensures that the case remains dynamic and actual user or group assignments can be made as late as required.

Roles are assigned to users or groups and authorized to perform tasks when a case instance is started.

## 7.1. CREATING CASE ROLES

You can create and define case roles in the case definition when you design the case in the process designer. Case roles are configured on the case definition level to keep them separate from the actors involved in handling the case instance. Roles can be assigned to user tasks or used as contact references throughout the case lifecycle, but they are not defined in the case as a specific user or group of users.

Case instances include the individuals that are actually handling the case work. Assign roles when starting a new case instance. In order to keep cases flexible, you can modify case role assignment during case run time, although doing this has no effect on tasks already created based on the previous role assignment. The actor assigned to a role is flexible but the role itself remains the same for each case.

### Prerequisites

- A case project that has a case definition exists in Business Central.
- The case definition asset is open in the process designer.

### Procedure

1. To define the roles involved in the case, click on an empty space in the editor's canvas, and click  to open the **Properties** menu.
2. Expand **Case Management** to add a case role.  
The case role requires a name for the role and a case cardinality. Case cardinality is the number of actors that are assigned to the role in any case instance. For example, the **IT\_Orders** sample case management project includes the following roles:

Figure 7.1. ITOrders Case Roles

The screenshot shows a web interface for Case Management. At the top, there is a dropdown menu labeled 'Case Management'. Below it, there is a section for 'Case ID Prefix' with a text input field containing 'IT'. Underneath, there is a section for 'Case Roles' which contains a table with three columns: 'Name', 'Cardinality', and a set of control icons (a plus sign and a trash can). The table lists three roles: 'owner' with a cardinality of 1, 'manager' with a cardinality of 1, and 'supplier' with a cardinality of 2.

Name	Cardinality	
owner	1	+ 🗑️
manager	1	+ 🗑️
supplier	2	+ 🗑️

In this example, you can assign only one actor (a user or a group) as the case **owner** and assign only one actor to the **manager** role. The **supplier** role can have two actors assigned. Depending on the case, you can assign any number of actors to a particular role based on the configured case cardinality of the role.

## 7.2. ROLE AUTHORIZATION

Roles are authorized to perform specific case management tasks when starting a new case instance using the Showcase application or the REST API.

Use the following procedure to start a new IT Orders case using the REST API.

### Prerequisites

- The **IT\_Orders** sample project has been imported in Business Central and deployed to the Process Server.

### Procedure

1. Create a **POST** REST API call with the following endpoint:  
<http://host:port/kie-server/services/rest/server/containers/itorders/cases/itorders.orderhardware/instances>
  - **itorders**: The container alias that has been deployed to the Process Server.
  - **itorders.orderhardware**: The name of the case definition.
2. Provide the following role configuration in the request body:

```
{
  "case-data" : { },
  "case-user-assignments" : {
    "owner" : "cami",
    "manager" : "cami"
  }
}
```

```

    },
    "case-group-assignments" : {
      "supplier" : "IT"
    }
  }
}

```

This starts a new case with defined roles, as well as autostart activities, which are started and ready to be worked on. Two of the roles are user assignments (**owner** and **manager**) and the third is a group assignment (**supplier**).

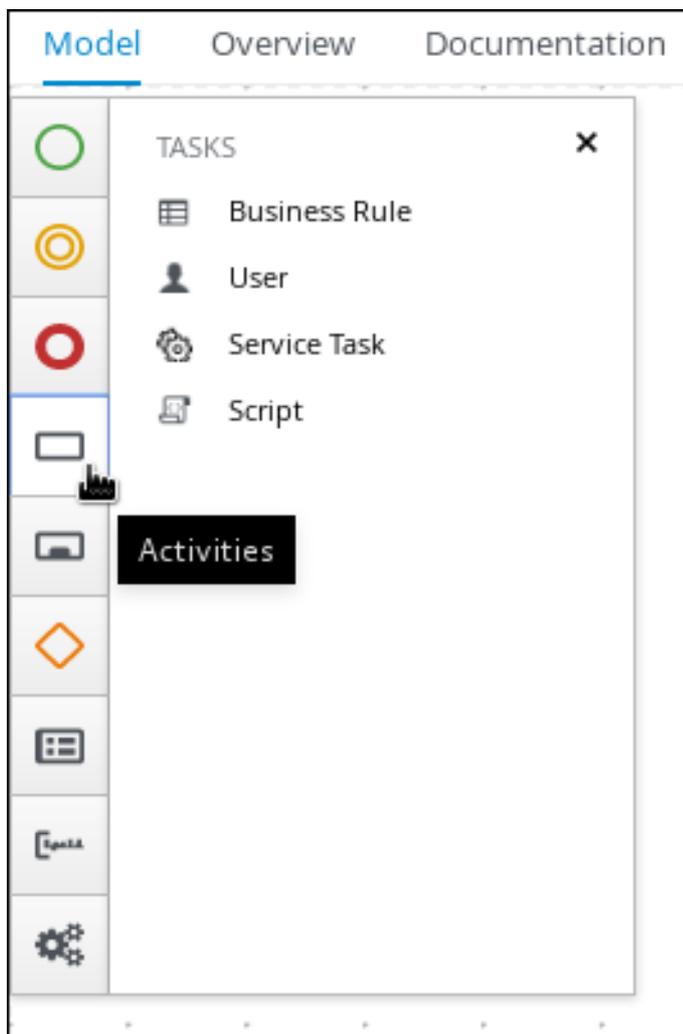
After the case instance is successfully started, the case instance returns the **IT-0000000001** case ID.

For information about how to start a new case instance using the Showcase application, see [Using the Showcase application for case management](#).

### 7.3. ASSIGNING A TASK TO A ROLE

Case management processes need to be as flexible as possible to accommodate changes that can happen dynamically during run time. This includes changing user assignments for new case instances or for active cases. For this reason, ensure that you do not hard code roles to a single set of users or groups in the case definition. Instead, role assignments can be defined on the task nodes in the case definition, with users or groups assigned to the roles on case creation.

Red Hat Process Automation Manager contains a predefined selection of node types to simplify business process creation. The predefined node panel is located on the left side of the diagram editor.



## Prerequisites

- A case definition has been created with case roles configured at the case definition level. For more information about creating case roles, see [Creating case roles](#).

## Procedure

1. Open the **Tasks** list and drag the user or service task you want to add to your case definition on to the process design palette.
2. With the task node selected, click  to open the **Properties** panel on the right side of the designer.
3. Expand **Implementation/Execution**, click **Add** below the **Actors** property and either select or type the name of the role to which the task will be assigned. You can use the **Groups** property in the same way for group assignments.  
For example, in the **IT\_Orders** sample project, the **Manager approval** user task is assigned to the **manager** role:

The screenshot displays the Red Hat Process Automation Manager interface. On the left, a case flow diagram is visible on a grid background. It consists of several tasks and milestones: 'Prepare hardware spec' (a user task icon) is connected to 'Manager approval' (a user task icon). Below this, 'Milestone 1: Order placed' (a flag icon) is connected to 'Notify requestor' (a document icon). 'Notify requestor' is connected to 'Milestone 2: Order shipped' (a flag icon). The 'Manager approval' task is highlighted with a blue border. On the right, the 'Properties' panel is open, showing details for the selected task.

**Properties Panel:**

- General:**
  - Name: Manager approval
  - Documentation: (empty text area)
- Implementation/Execution:**
  - Task Name: ManagerApproval
  - Subject: (empty text field)
  - Actors:
    - Name: manager (dropdown menu)
    - + Add
  - Groups:
    - Name: (empty text field)
    - + Add
  - Assignments: 3 data inputs, 2 data outputs
  - Reassignments: 0 reassignments
  - Notifications: 0 notifications
  - Is Async:

In this example, after the **Prepare hardware spec** user task has been completed, the user assigned to the **manager** role will receive the **Manager approval** task in their **Task Inbox** in Business Central.

The user assigned to the role can be changed during the case run time, but the task itself continues to have the same role assignment. For example, the person originally assigned to the **manager** role might need to take time off (if they become ill, for example), or they might unexpectedly leave the company. To respond to this change in circumstances, you can edit the **manager** role assignment so that someone else can be assigned the tasks associated with that role.

For information about how to change role assignments during case run time, see [Modifying case role assignments during run time using Showcase](#) or [Modifying case role assignments during run time using REST API](#).

## 7.4. MODIFYING CASE ROLE ASSIGNMENTS DURING RUN TIME USING SHOWCASE

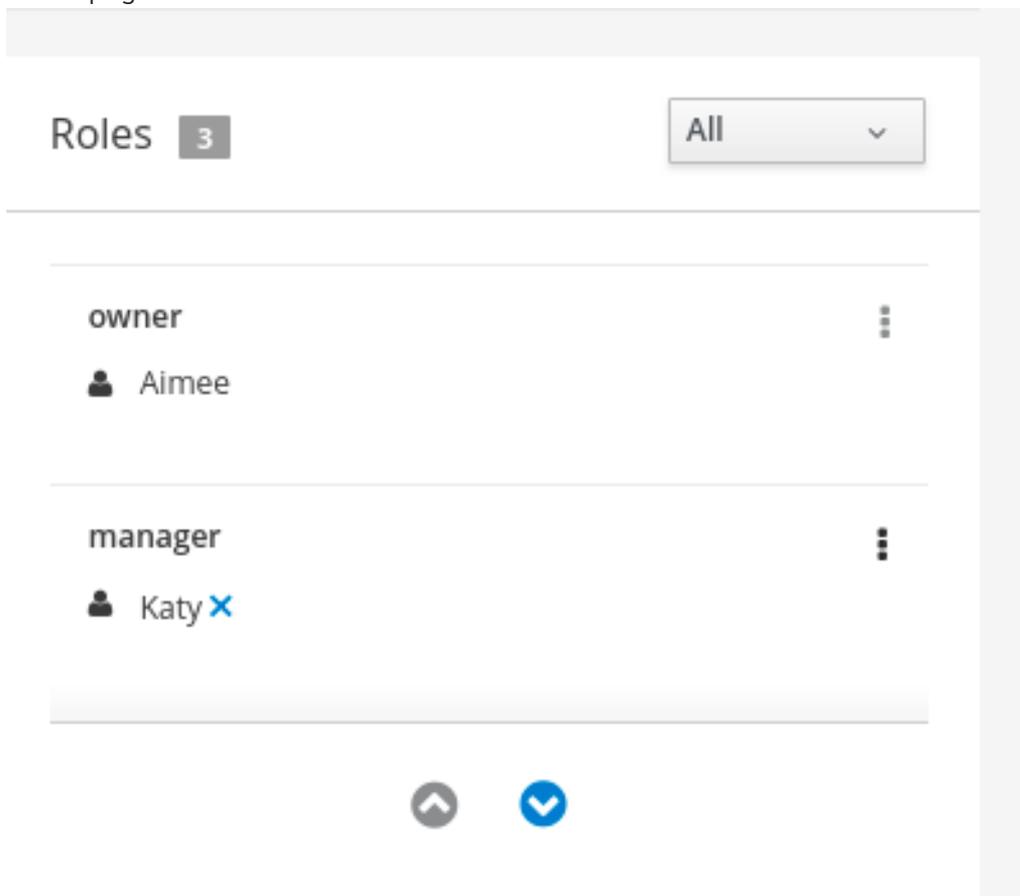
You can change case instance role assignments during case run time using the Showcase application. Roles are defined in the case definition and assigned to tasks in the case lifecycle. Roles cannot change during run time because they are predefined, but you can change the actors assigned to the roles to change who is responsible for carrying out case tasks.

### Prerequisites

- There is an active case instance with users or groups already assigned to at least one case role.

### Procedure

1. In the Showcase application, click the case you want to work on in the **Case list** to open the case overview.
2. Locate the role assignment that you want to change in the **Roles** box in the lower-right corner of the page.



3. To remove a single user or group from the role assignment, click the  next to the assignment. In the confirmation window, click **Remove** to remove the user or group from the role.
4. To remove all role assignments from a role, click the  next to the role and select the **Remove all assignments** option. In the confirmation window, click **Remove** to remove all user and group assignments from the role.
5. To change the role assignment from one user or group to another, click the  next to the role and select the **Edit** option.

6. In the **Edit role assignment** window, delete the name of the assignee that you want to remove from the role assignment. Type the name of the user you want to assign to the role into the **User** field or the group you want to assign in the **Group** field.  
At least one user or group must be assigned when editing a role assignment.
7. Click **Assign** to complete the role assignment.

## 7.5. MODIFYING CASE ROLE ASSIGNMENTS DURING RUN TIME USING REST API

You can change case instance role assignments during case run time using the REST API or Swagger application. Roles are defined in the case definition and assigned to tasks in the case life cycle. Roles cannot change during run time because they are predefined, but you can change the actors assigned to the roles to change who is responsible for carrying out case tasks.

The following procedure includes examples based on the **IT\_Orders** sample project. You can use the same REST API endpoints in the Swagger application or any other REST API client, or using Curl.

### Prerequisites

- An IT Orders case instance has been started with **owner**, **manager**, and **supplier** roles already assigned to actors.

### Procedure

1. Retrieve the list of current role assignments using a **GET** request on the following endpoint:  
**/http://localhost:8080/kie-server/services/rest/server/containers/{id}/cases/instances/{caseId}/roles**

Table 7.1. Parameters

Name	Description
<b>id</b>	<b>itorders</b>
<b>caseId</b>	<b>IT-0000000001</b>

This returns the following response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<case-role-assignment-list>
  <role-assignments>
    <name>owner</name>
    <users>Aimee</users>
  </role-assignments>
  <role-assignments>
    <name>manager</name>
    <users>Katy</users>
  </role-assignments>
  <role-assignments>
    <name>supplier</name>
```

```

    <groups>Lenovo</groups>
  </role-assignments>
</case-role-assignment-list>

```

- To change the user assigned to the **manager** role, you must first remove the role assignment from the user **Katy** using **DELETE**.

**/server/containers/{id}/cases/instances/{caseId}/roles/{caseRoleName}**

Include the following information in the Swagger client request:

Table 7.2. Parameters

Name	Description
<b>id</b>	<b>itorders</b>
<b>caseId</b>	<b>IT-0000000001</b>
<b>caseRoleName</b>	<b>manager</b>
<b>user</b>	<b>Katy</b>

Click **Execute**.

- Execute the **GET** request from the first step again to check that the **manager** role no longer has a user assigned:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<case-role-assignment-list>
  <role-assignments>
    <name>owner</name>
    <users>Aimee</users>
  </role-assignments>
  <role-assignments>
    <name>manager</name>
  </role-assignments>
  <role-assignments>
    <name>supplier</name>
    <groups>Lenovo</groups>
  </role-assignments>
</case-role-assignment-list>

```

- Assign the user **Cami** to the **manager** role using a **PUT** request on the following endpoint:  
**/server/containers/{id}/cases/instances/{caseId}/roles/{caseRoleName}**

Include the following information in the Swagger client request:

Table 7.3. Parameters

Name	Description
<b>id</b>	<b>itorders</b>

Name	Description
<b>caseId</b>	<b>IT-0000000001</b>
<b>caseRoleName</b>	<b>manager</b>
<b>user</b>	<b>Cami</b>

Click **Execute**.

- Execute the **GET** request from the first step again to check that the **manager** role is now assigned to **Cami**:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<case-role-assignment-list>
  <role-assignments>
    <name>owner</name>
    <users>Aimee</users>
  </role-assignments>
  <role-assignments>
    <name>manager</name>
    <users>Cami</users>
  </role-assignments>
  <role-assignments>
    <name>supplier</name>
    <groups>Lenovo</groups>
  </role-assignments>
</case-role-assignment-list>
```

## CHAPTER 8. STAGES

Case management stages are a collection of tasks. A stage is an ad hoc subprocess that can be defined using the process designer and may include other case management nodes, such as a milestone. A milestone can also be configured as completed when a stage or a number of stages are completed. Therefore, a milestone may be activated or achieved by the completion of a stage, and a stage may include a milestone or a number of milestones.

For example, in a patient triage case, the first stage may consist of observing and noting any obvious physical symptoms or a description from the patient of what their symptoms are, followed by a second stage for tests, and a third for diagnosis and treatment.

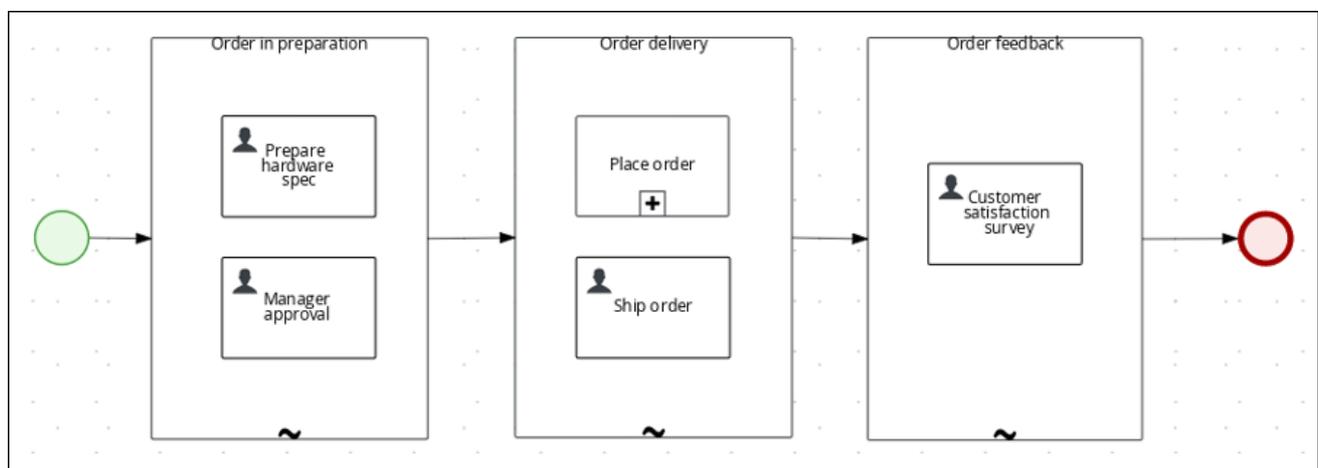
There are three ways to complete a stage:

- By completion condition.
- By terminal end event.
- By setting the **Completion Condition** to **autocomplete**, which will automatically complete the stage when there are no active tasks left in the stage.

### 8.1. DEFINING A STAGE

A stage can be modeled in BPMN2 using the process designer. Stages are a way of grouping related tasks in a way that clearly defines activities that, if the stage is activated, must complete before the next stage of the case commences. For example, the **IT\_Orders** case definition can also be defined using stages in the following way:

**Figure 8.1. IT\_Orders project stages example**



#### Procedure

1. From the predefined node panel located on the left side of the diagram editor, drag and drop an **Ad-Hoc** node onto the design canvas and provide a name for the stage node.
2. Define how the stage is activated:
  - If the stage is being activated by an incoming node, connect the stage with a sequence flow line from the incoming node.
  - If the stage is instead being activated by a signal event, configure the **SignalRef** on the signal node with the name of the stage that you configured in the first step.

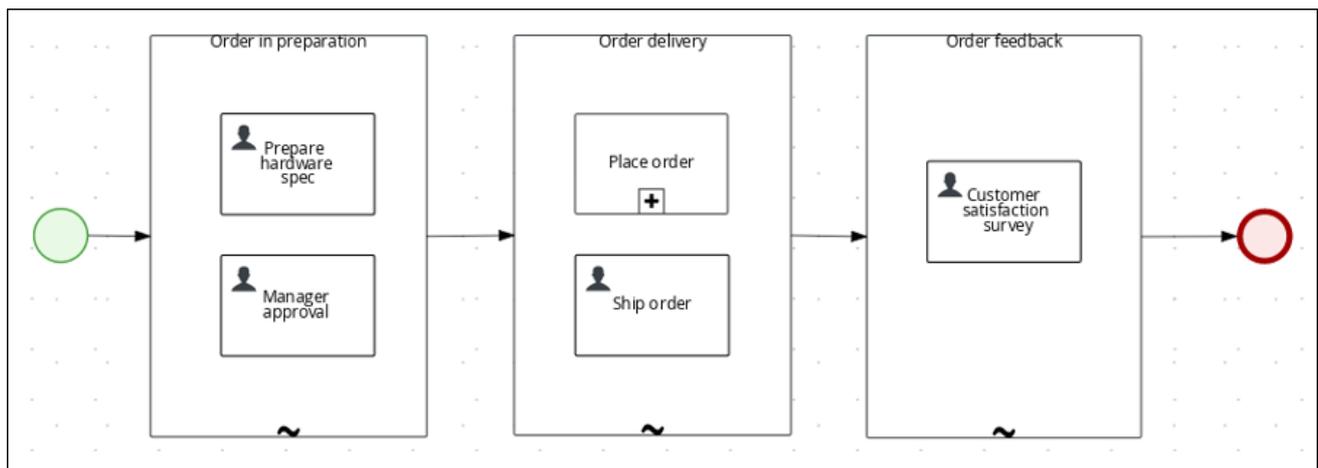
- Alternatively, configure the **AdHocActivationCondition** property to activate the stage when the condition has been met.
3. Re-size the node as required to provide room to add the task nodes for the stage.
  4. Add the relevant tasks to the stage and configure them as required.
  5. (Optional) Configure a completion condition for the stage. As an ad hoc subprocess, stages are configured as **autocomplete** by default, which means that the stage will automatically complete and trigger the next activity in the case definition once all instances in the stage are no longer active.  
To change the completion condition, select the stage node and open the **Properties** panel on the right, expand **Implementation/Execution**, and modify the **AdHocCompletionCondition** property field with a free-form Drools expression for the completion condition you require. For more information about stage completion conditions, see [Section 8.2, "Configuring stage activation and completion conditions"](#).
  6. Once the stage has been configured, connect it to the next activity in the case definition using a sequence flow line.

## 8.2. CONFIGURING STAGE ACTIVATION AND COMPLETION CONDITIONS

Stages can be triggered by a start node, intermediate node, or manually using an API call.

You can configure stages with both activation and completion conditions using free-form Drools rules, the same way that milestone completion conditions are configured. For example, in the **IT\_Orders** sample project, the **Milestone 2: Order shipped** completion condition (`org.kie.api.runtime.process.CaseData(data.get("shipped")) == true`) can also be used as the completion condition for the **Order delivery** stage represented here:

Figure 8.2. IT\_Orders project stages example



Activation conditions can also be configured using a free-form Drools rule to configure the **AdHocActivationCondition** property to activate a stage.

### Prerequisites

- You have created a case definition in the Business Central process designer.
- You have added an ad hoc subprocess to the case definition that is to be used as a stage.

## Procedure

1. With the stage selected, click  to open the **Properties** panel on the right side of the designer.
2. Expand **Implementation/Execution** and in the **AdHocActivationCondition** property editor define an activation condition for the start node. For example, set **autostart: true** to make the stage automatically activated when a new case instance is started.
3. The **AdHocCompletionCondition** is set to **autocomplete** by default. To change this, input a completion condition using a free-form Drools expression. For example, set **org.kie.api.runtime.process.CaseData(data.get("ordered") == true)** to activate the second stage in the example shown previously.

For more examples and information about the conditions used in the **IT\_Orders** sample project, see [Getting started with case management](#).

## 8.3. ADDING A DYNAMIC TASK TO A STAGE

Dynamic tasks can be added to a case stage during run time using a REST API request. This is similar to adding a dynamic task to a case instance, but you must also define the **caseStageId** of the stage to which the task is added.

Use the following procedure to add a dynamic task to a stage in the **IT\_Orders** sample project available in Business Central using the Swagger REST API tool. The same endpoint can be used for the REST API without Swagger.

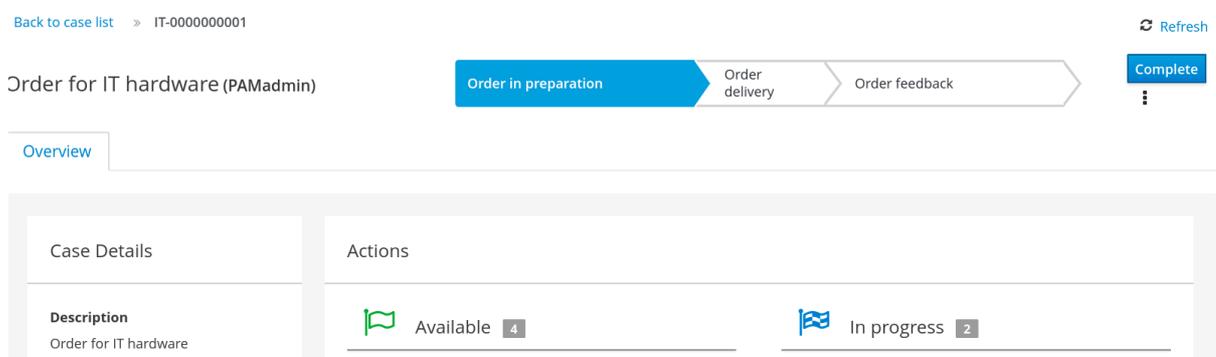
### Prerequisites

- The **IT\_Orders** sample project BPMN2 case definition has been reconfigured to use stages instead of milestones, as demonstrated in the provided example. For information about configuring stages for case management, see [Defining a stage](#).

### Procedure

1. Start a new case using the Showcase application. For more information about using Showcase, see [Using the Showcase application for case management](#).

Because this case is designed using stages, the case details page shows stage tracking:



Back to case list > IT-0000000001 Refresh

Order for IT hardware (PAMadmin) Order in preparation Order delivery Order feedback Complete

Overview

Case Details	Actions
<b>Description</b> Order for IT hardware	<span style="color: green;">🚩</span> Available <span style="background-color: gray; color: white; padding: 2px;">4</span> <span style="margin-left: 100px;"><span style="color: blue;">🚩</span> In progress <span style="background-color: gray; color: white; padding: 2px;">2</span></span>

The first stage starts automatically when the case instance is created.

2. As a **manager** user, approve the hardware specification in Business Central under **Menu → Track → Task Inbox**, then check the progress of the case.

- a. In Business Central, click **Menu** → **Manage** → **Process Instances** and open the active case instance **IT-0000000001**.
  - b. Click **Diagram** to see the case progress.
3. In a web browser, open the following URL:  
**/http://localhost:8080/kie-server/docs**.
  4. Open the list of available endpoints under **Case instances :: Case Management**.
  5. Click click the following **POST** method endpoint to open the details:  
**/server/containers/{id}/cases/instances/{caseId}/stages/{caseStageld}/tasks**
  6. Click **Try it out** to complete the following parameters:

Table 8.1. Parameters

Name	Description
<b>id</b>	<b>itorders</b>
<b>caseId</b>	<b>IT-0000000001</b>
<b>caseStageld</b>	<b>Order delivery</b>

The **caseStageld** is the name of the stage in the case definition where the dynamic task is to be created. This can be any dynamic or service task payload. See [Creating a dynamic subprocess using the REST API](#) or [Creating a dynamic service task using the REST API](#) for examples.

After the dynamic task has been added to the stage, it must be completed in order for the stage to complete and for the case process to move on to the next item in the case flow.

## CHAPTER 9. MILESTONES

Milestones are a special service task that can be configured in the case definition designer by adding the milestone node to the process designer palette. When creating a new case definition, a milestone configured as **Adhoc autostart** is included on the design palette by default. Newly created milestones are not set to **Adhoc autostart** by default.

Case management milestones generally occur at the end of a stage, but they can also be the result of achieving other milestones. A milestone always requires a condition to be defined in order to track progress. Milestones react to case file data when data is added to a case. A milestone represents a single point of achievement within the case instance. It can be used to flag certain events, which can be useful for Key Performance Indicator (KPI) tracking or identifying the tasks that are still to be completed.

Milestones can be in any of the following states during case execution:

- **Active** - The condition has been defined on the milestone but it has not been met.
- **Completed** - The milestone condition has been met, the milestone has been achieved, and the case can proceed to the next task.
- **Terminated** - The milestone is no longer a part of the case process and is no longer required.

While a milestone is available or completed it can be triggered manually by a signal or automatically if **Adhoc autostart** is configured when a case instance starts. Milestones can be triggered as many times as required, however, it is directly achieved when the condition is met.

### 9.1. MILESTONE CONFIGURATION AND TRIGGERING

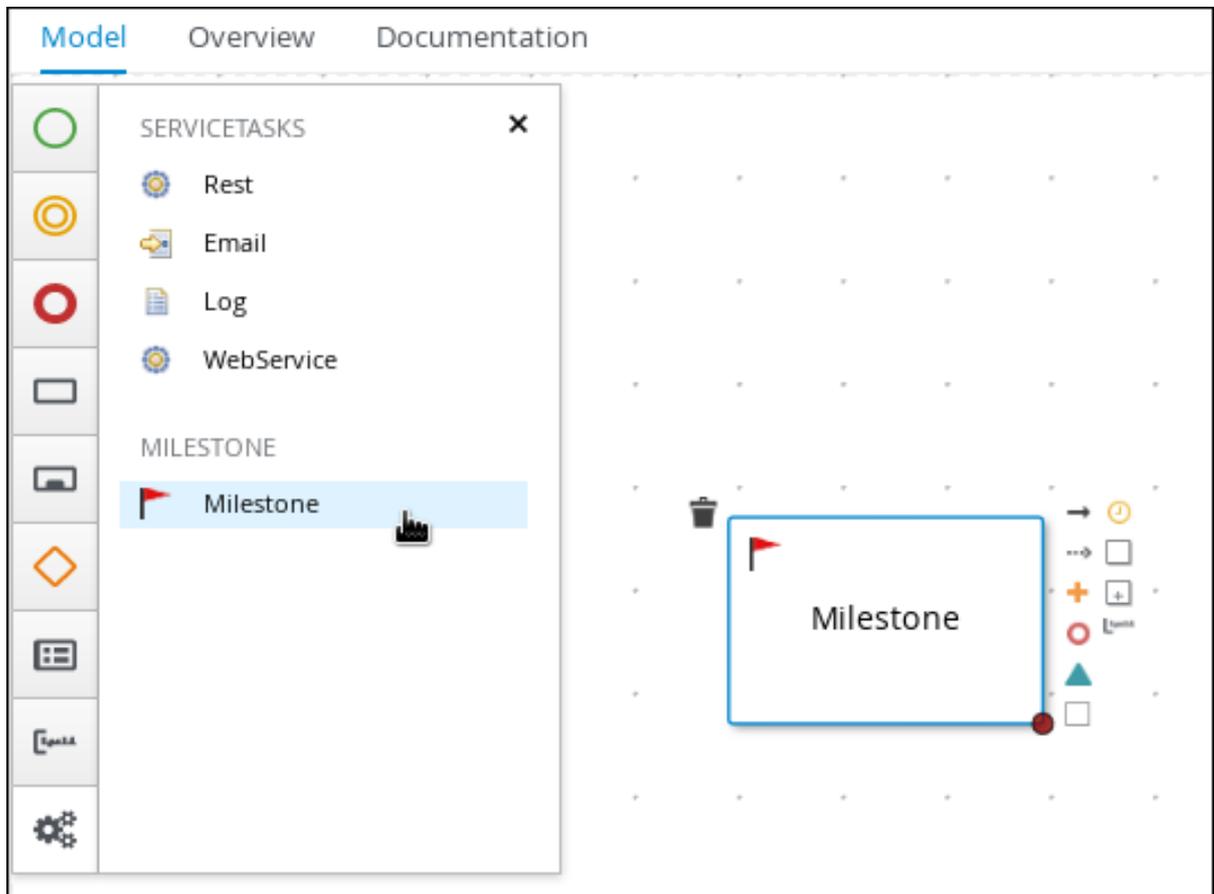
Case milestones can be configured to start automatically when a case instance starts or they can be triggered using a signal, which is configured manually during the case design.

#### Prerequisites

- A case project has been created in Business Central.
- A case definition has been created.

#### Procedure

1. From the predefined node panel located on the left side of the diagram editor, drag and drop a **Milestone** object onto the palette.



2. With the milestone selected, click  to open the **Properties** panel on the right side of the designer.
3. Expand **Data Assignments** to add a completion condition. Milestones include a **condition** parameter by default.
4. To define the completion condition for the milestone, select **Constant** from the **Source** list. The condition must be provided using the Drools syntax.
5. Expand **Implementation/Execution** to configure the **Adhoc Autostart** property.
  - Click the check box to set this property to **true** for milestones that are required to start automatically when a case instance starts.
  - Leave the check box empty to set this property to **false** for milestones that are to be triggered by a signal event.
6. **(Optional)** Configure a signal event to trigger a milestone once a case goal has been reached.
  - a. With the signal event selected in the case design palette, open the **Properties** panel on the right.
  - b. Set the **Signal Scope** property to **Process Instance**.
  - c. Open the **SignalRef** expression editor and type the name of the milestone to be triggered.

▼ Implementation/Execution

Signal ⓘ

Milestone 2: Order shipped ▼

Signal Scope

Process Instance ▼

7. Click **Save**.

## CHAPTER 10. CASE EVENT LISTENER

The CaseEventListener is used to initiate notifications for case-related events and operations that are invoked on a case instance. Implement the case event listener by overriding the methods as needed for your particular use case.

You can configure the listener using the deployment descriptors located in Business Central in **Menu → Design → \$PROJECT\_NAME → Settings → Deployments**.

When a new project is created, a **kie-deployment-descriptor.xml** file is generated with default values.

== CaseEventListener methods

```
public interface CaseEventListener extends EventListener {

    default void beforeCaseStarted(CaseStartEvent event) {
    };

    default void afterCaseStarted(CaseStartEvent event) {
    };

    default void beforeCaseClosed(CaseCloseEvent event) {
    };

    default void afterCaseClosed(CaseCloseEvent event) {
    };

    default void beforeCaseCancelled(CaseCancelEvent event) {
    };

    default void afterCaseCancelled(CaseCancelEvent event) {
    };

    default void beforeCaseDestroyed(CaseDestroyEvent event) {
    };

    default void afterCaseDestroyed(CaseDestroyEvent event) {
    };

    default void beforeCaseReopen(CaseReopenEvent event) {
    };

    default void afterCaseReopen(CaseReopenEvent event) {
    };

    default void beforeCaseCommentAdded(CaseCommentEvent event) {
    };

    default void afterCaseCommentAdded(CaseCommentEvent event) {
    };

    default void beforeCaseCommentUpdated(CaseCommentEvent event) {
    };

    default void afterCaseCommentUpdated(CaseCommentEvent event) {
```

```
};

default void beforeCaseCommentRemoved(CaseCommentEvent event) {
};

default void afterCaseCommentRemoved(CaseCommentEvent event) {
};

default void beforeCaseRoleAssignmentAdded(CaseRoleAssignmentEvent event) {
};

default void afterCaseRoleAssignmentAdded(CaseRoleAssignmentEvent event) {
};

default void beforeCaseRoleAssignmentRemoved(CaseRoleAssignmentEvent event) {
};

default void afterCaseRoleAssignmentRemoved(CaseRoleAssignmentEvent event) {
};

default void beforeCaseDataAdded(CaseDataEvent event) {
};

default void afterCaseDataAdded(CaseDataEvent event) {
};

default void beforeCaseDataRemoved(CaseDataEvent event) {
};

default void afterCaseDataRemoved(CaseDataEvent event) {
};

default void beforeDynamicTaskAdded(CaseDynamicTaskEvent event) {
};

default void afterDynamicTaskAdded(CaseDynamicTaskEvent event) {
};

default void beforeDynamicProcessAdded(CaseDynamicSubprocessEvent event) {
};

default void afterDynamicProcessAdded(CaseDynamicSubprocessEvent event) {
};
}
```

## CHAPTER 11. RULES IN CASE MANAGEMENT

Cases are data-driven, rather than following a sequential flow. The steps required to resolve a case rely on data, which is provided by people involved in the case, or the system can be configured to trigger further actions based on the data available. In the latter case, you can use business rules to decide what further actions are required for the case to continue or reach a resolution.

Data can be inserted into the case file at any point during the case. The decision engine constantly monitors case file data, meaning that rules react to data that is contained in the case file. Using rules to monitor and respond to changes in the case file data provides a level of automation that drives cases forward.

### 11.1. USING RULES TO DRIVE CASES

Refer to the case management **IT\_Orders** sample project in Business Central.

Suppose that the particular hardware specification provided by the supplier is incorrect or invalid. The supplier needs to provide a new, valid order so that the case can continue. Rather than wait for the manager to reject the invalid specification and create a new request for the supplier, you can create a business rule that will react immediately when the case data indicates that the provided specification is invalid. It can then create a new hardware specification request for the supplier.

The following procedure demonstrates how to create and use a business rule to execute this scenario.

#### Prerequisites

- The **IT\_Orders** sample project is open in Business Central, but it is not deployed to the Process Server.
- The **ServiceRegistry** is part of the **jbpm-services-api** module, and must be available on the class path.



#### NOTE

If building the project outside of Business Central, the following dependencies must be added to the project:

- **org.jbpm:jbpm-services-api**
- **org.jbpm:jbpm-case-mgmt-api**

#### Procedure

1. Create the following business rule file called **validate-document.drl**:

```
package defaultPackage;

import java.util.Map;
import java.util.HashMap;
import org.jbpm.casemgmt.api.CaseService;
import org.jbpm.casemgmt.api.model.instance.CaseFileInstance;
import org.jbpm.document.Document;
import org.jbpm.services.api.service.ServiceRegistry;

rule "Invalid document name - reupload"
```

```

when
  $caseData : CaseFileInstance()
  Document(name == "invalid.pdf") from $caseData.getData("hwSpec")

then

  System.out.println("Hardware specification is invalid");
  $caseData.remove("hwSpec");
  update($caseData);
  CaseService caseService = (CaseService)
  ServiceRegistry.get().service(ServiceRegistry.CASE_SERVICE);
  caseService.triggerAdHocFragment($caseData.getCaseld(), "Prepare hardware spec",
  null);
end

```

This business rule detects when a file named **invalid.pdf** is uploaded to the case file. It then removes the **invalid.pdf** document and creates a new instance of the **Prepare hardware spec** user task.

2. Click **Deploy** to build the **IT\_Orders** project and deploy it to a Process Server.



#### NOTE

You can also select the **Build & Install** option to build the project and publish the KJAR file to the configured Maven repository without deploying to a Process Server. In a development environment, you can click **Deploy** to deploy the built KJAR file to a Process Server without stopping any running instances (if applicable), or click **Redeploy** to deploy the built KJAR file and replace all instances. The next time you deploy or redeploy the built KJAR, the previous deployment unit (KIE container) is automatically updated in the same target Process Server. In a production environment, the **Redeploy** option is disabled and you can click **Deploy** only to deploy the built KJAR file to a new deployment unit (KIE container) on a Process Server.

To configure the Process Server environment mode, set the **org.kie.server.mode** system property to **org.kie.server.mode=development** or **org.kie.server.mode=production**. To configure the deployment behavior for a corresponding project in Business Central, go to project **Settings** → **General Settings** → **Version** and toggle the **Development Mode** option. By default, Process Server and all new projects in Business Central are in development mode. You cannot deploy a project with **Development Mode** turned on or with a manually added **SNAPSHOT** version suffix to a Process Server that is in production mode.

3. Create a file called **invalid.pdf** and save it locally.
4. Create a file called **valid-spec.pdf** and save it locally.
5. In Business Central, go to **Menu** → **Projects** → **IT\_Orders** to open the **IT\_Orders** project.
6. Click **Import Asset** in the upper-right corner of the page.
7. Upload the **validate-document.drl** file to the **default** package (**src/main/resources**) and click **Ok**.

**Create new Import Asset**
✕

**Import Asset \***

**Package**

<default>
▼

**Please select a file to upload**

validate-document.drl
📄

+ Ok
Cancel

The **validate-document.drl** rule is shown in the rule editor. Click **Save** or close to exit the rule editor.

8. Open the Showcase application by either clicking the **Apps launcher** (if it is installed), or go to **/http://localhost:8080/rhpam-case-mgmt-showcase/jbpm-cm.html**.
9. Click **Start Case** for the **IT\_Orders** project.  
In this example, Aimee is the case **owner**, Katy is the **manager**, and the supplier group is **supplier**.

**Start Case**
✕

**Case Name \***

Order for IT hardware
▼

**Case Owner \***

**Role Assignments ⓘ**

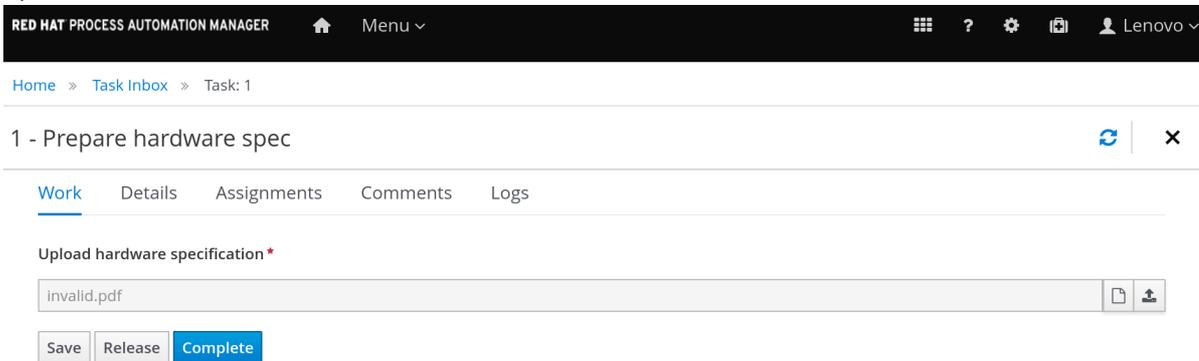
Role Name	Users	Groups
manager	<input style="width: 90%;" type="text" value="Katy"/>	<input style="width: 90%;" type="text"/>
supplier	<input style="width: 90%;" type="text"/>	<input style="width: 90%; border: 2px solid #0070c0;" type="text" value="supplier"/>

Cancel
Start

10. Log out of Business Central, and log back in as a user that belongs to the **supplier** group.

- Go to **Menu** → **Track** → **Task Inbox**.
- Open the **Prepare hardware spec** task and click **Claim**. This assigns the task to the logged in user.

- Click **Start** and click  to locate the **invalid.pdf** hardware specification file. Click  to upload the file.



RED HAT PROCESS AUTOMATION MANAGER

Home » Task Inbox » Task: 1

1 - Prepare hardware spec

Work Details Assignments Comments Logs

Upload hardware specification \*

invalid.pdf

Save Release Complete

- Click **Complete**.  
The value in the **Task Inbox** for the **Prepare hardware spec** is **Ready**.
- In Showcase, click **Refresh** in the upper-right corner. Notice that a **Prepare hardware task** message appears in the **Completed** column and another appears in the **In Progress** column.

Actions

Available <span>4</span>	In progress <span>4</span>	Completed <span>2</span>
New user task Dynamic	Milestone 1: Order placed 13/07/2018 ( Milestone )	Hardware spec ready 13/07/2018 ( Milestone )
New process task Dynamic	Manager decision 13/07/2018 ( Milestone )	Prepare hardware spec 13/07/2018 ( Human Task )
Milestone 2: Order shipped Available in: Case	Prepare hardware spec 13/07/2018 ( Human Task )	

This is because the first **Prepare hardware spec** task has been completed with the specification file **invalid.pdf**. As a result, the business rule causes the task and file to be discarded, and a new user task created.

- In the Business Central **Task Inbox**, repeat steps 12 and 13, but upload the **valid-spec.pdf** file instead of **invalid.pdf**.

## CHAPTER 12. CASE MANAGEMENT SECURITY

Cases are configured at the case definition level with case roles. These are generic participants that are involved in case handling. These roles can be assigned to user tasks or used as contact references. Roles are not hard-coded to specific users or groups to keep the case definition independent of the actual actors involved in any given case instance. You can modify case role assignments at any time as long as case instance is active, though modifying a role assignment does not affect tasks already created based on the previous role assignment.

Case instance security is enabled by default. The case definition prevents case data from being accessed by users who do not belong to the case. Unless a user has a case role assignment (either assigned as user or a group member) then they are not able to access the case instance.

Case security is one of the reasons why it is recommended that you assign case roles when starting a case instance, as this will prevent tasks being assigned to users who should not have access to the case.

### 12.1. CONFIGURING SECURITY FOR CASE MANAGEMENT

You can turn off case instance authorization by setting the following system property to **false**:

**org.jbpm.cases.auth.enabled**

This system property is just one of the security components for case instances. In addition, you can configure case operations at the execution server level using the **case-authorization.properties** file, available at the root of the class path of the execution server application (**kie-server.war/WEB-INF/classes**).

Using a simple configuration file for all possible case definitions encourages you to think about case management as domain-specific. **AuthorizationManager** for case security is pluggable, which allows you to include custom code for specific security handling.

You can restrict the following case instance operations to case roles:

- **CANCEL\_CASE**
- **DESTROY\_CASE**
- **REOPEN\_CASE**
- **ADD\_TASK\_TO\_CASE**
- **ADD\_PROCESS\_TO\_CASE**
- **ADD\_DATA**
- **REMOVE\_DATA**
- **MODIFY\_ROLE\_ASSIGNMENT**
- **MODIFY\_COMMENT**

#### Prerequisites

- The Red Hat Process Automation Manager Process Server is not running.

#### Procedure

1. Open **JBOSS\_HOME/standalone/deployments/kie-server.war/WEB-INF/classes/case-authorization.properties** file in your preferred editor.

By default, the file contains the following operation restrictions:

```
CLOSE_CASE=owner,admin
CANCEL_CASE=owner,admin
DESTROY_CASE=owner,admin
REOPEN_CASE=owner,admin
```

2. You can add or remove role permissions for these operations.
  - a. To remove permission for a role to perform an operation, remove it from the list of authorized roles for that operation in the **case-authorization.properties** file. For example, removing the **admin** role from the **CLOSE\_CASE** operation restricts permission to close a case to the case owner for all cases.
  - b. To give a role permission to perform a case operation, add it to the list of authorized roles for that operation in the **case-authorization.properties** file. For example, to allow anyone with the **manager** role to perform a **CLOSE\_CASE** operation, add it to the list of roles, separated by a comma:

```
CLOSE_CASE=owner,admin,manager
```

3. To add role restrictions to other case operations listed in the file, remove the **#** from the line and list the role names in the following format:

```
OPERATION=role1,role2,roleN
```

Operations in the file that begin with **#** have restrictions ignored and can be performed by anyone involved in the case.

4. When you have finished assigning role permissions, save and close the **case-authorization.properties** file.
5. Start the execution server.

The case authorization settings apply to all cases on the execution server.

## CHAPTER 13. CLOSING CASES

A case instance can be completed when there are no more activities to be performed and the business goal is achieved, or it can be closed prematurely. Usually the case owner closes the case when all work is completed and the case goals have been met. When you close a case, consider adding a comment about why the case instance is being closed. A closed case can be reopened later with the same case ID if required.

You can close case instances remotely using Process Server REST API requests or directly in the Showcase application.

### 13.1. CLOSING A CASE USING THE PROCESS SERVER REST API

You can use a REST API request to close a case instance. Red Hat Process Automation Manager includes the Swagger client, which includes endpoints and documentation for REST API requests. Alternatively, you can use the same endpoints to make API calls using your preferred client or Curl.

#### Prerequisites

- A case instance has been started using Showcase.
- You are able to authenticate API requests as a user with the **admin** role.

#### Procedure

1. Open the Swagger REST API client in a web browser:  
**/http://localhost:8080/kie-server/docs**
2. Under **Case Instances :: Case Management**, open the **POST** request with the following endpoint:  
**/server/containers/{id}/cases/instances/{caseId}**
3. Click **Try it out** and fill in the required parameters:

**Table 13.1. Parameters**

Name	Description
<b>id</b>	<b>itorders</b>
<b>caseId</b>	<b>IT-0000000001</b>

4. Optionally, you can include a comment to be included in the case file. To leave a comment, type it into the **body** text field as a **String**.
5. Click **Execute** to close the case.
6. To confirm the case is closed, open the Showcase application and change the case list status to **Closed**.

### 13.2. CLOSING A CASE IN THE SHOWCASE APPLICATION

A case instance is complete when no more activities need to be performed and the business goal has

been achieved. After a case is complete, you can close the case to indicate that the case is complete and that no further work is required. When you close a case, consider adding a specific comment about why you are closing the case. If needed, you can reopen the case later with the same case ID.

You can use the Showcase application to close a case instance at any time. From Showcase, you can easily view the details of the case or leave a comment before closing it.

### Prerequisites

- You are logged in to the Showcase application and are the owner or administrator for a case instance that you want to close.

### Procedure

1. In the Showcase application, locate the case instance you want to close from the list of case instances.
2. To close the case without viewing the details first, click **Close**.
3. To close the case from the case details page, click the case in the list to open it. From the case overview page you can add comments to the case and verify that you are closing the correct case based on the case information.
4. Click **Close** to close the case.
5. Click **Back to Case List** in the upper-left corner of the page to return to the Showcase case list view.
6. Click the drop-down list next to **Status** and select **Canceled** to view the list of closed and canceled cases.

## CHAPTER 14. CANCELING OR DESTROYING A CASE

Cases can be canceled if they are no longer required and do not require any case work to be performed. Cases that are canceled can be reopened later with the same case instance ID and case file data. In some cases, you might want to permanently destroy a case so that it cannot be reopened.

Cases can only be canceled or destroyed using an API request. Red Hat Process Automation Manager includes the Swagger client, which includes endpoints and documentation for REST API requests. Alternatively, you can use the same endpoints to make API calls using your preferred client or Curl.

### Prerequisites

- A case instance has been started using Showcase.
- You are able to authenticate API requests as a user with the **admin** role.

### Procedure

1. Open the Swagger REST API client in a web browser:  
**/http://localhost:8080/kie-server/docs**
2. Under **Case Instances :: Case Management**, open the **DELETE** request with the following endpoint:  
**/server/containers/{id}/cases/instances/{caseId}**

You can cancel a case using the **DELETE** request. Optionally, you can also destroy the case using the **destroy** parameter.

3. Click **Try it out** and fill in the required parameters:

**Table 14.1. Parameters**

Name	Description
<b>id</b>	<b>itorders</b>
<b>caseId</b>	<b>IT-0000000001</b>
<b>destroy</b>	<b>true</b>  (Optional. Permanently destroys the case. This parameter is <b>false</b> by default.)

4. Click **Execute** to cancel (or destroy) the case.
5. To confirm the case is canceled, open the Showcase application and change the case list status to **Canceled**. If the case has been destroyed, it will no longer appear in any case list.

## CHAPTER 15. ADDITIONAL RESOURCES

- *Getting started with case management*
- *Using the Showcase application for case management*

## APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Wednesday, June 17, 2020.