Application Help for Production Process Designer

Generated on: 2024-05-09 14:29:56 GMT+0000

SAP Digital Manufacturing | latest

PUBLIC

Original content: https://help.sap.com/docs/SAP_DIGITAL_MANUFACTURING/f1949e1d1fd24d7e998e5f70fecc1835?locale=en-US&state=PRODUCTION&version=latest

Warning

About Production Process Designer

The Production Process Designer helps you model production processes and get transparency when you create the layout of your shop floor. Production processes define the interaction between machines or rules, actions, and workflows that control the execution on the shop floor.

Advantages of Production Process Designer

With the Production Process Designer, you can:

- Model production processes using a visual designer
- Focus on business logic instead of on programming skills
- Manage configurations centrally instead of programming each piece of equipment individually
- Manage different versions of configurations
- Adapt configurations easily if the environment changes

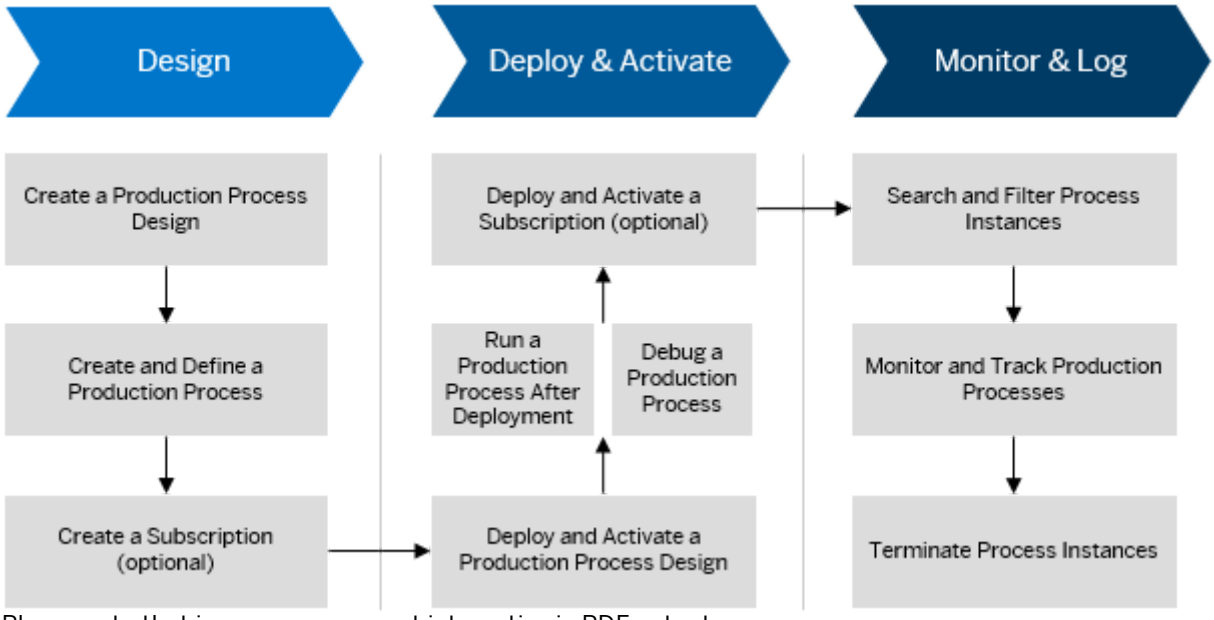
Related Information

Production Process Administration
Design Production Processes

Workflow Overview

The following diagram shows the general workflow to use production process:

This image is interactive. Hover over each area for a description. Click highlighted areas for more information.



Please note that image maps are not interactive in PDF output.

Scenarios

With the production process designer, you can design production processes that fit for different scenarios. For each scenario, you need to prepare some master data beforehand. The following sections explain what master data must be created for each scenario.

Scenario 1: Machine Automation

To control machines without human intervention, design an automation sequence. The machine-to-machine or machine-to-system communication is orchestrated by an Production Connector / SAP Plant Connectivity system.

For this scenario, you need to prepare the following master data:

- A Production Connector / SAP Plant Connectivity system
- One or more external service providers with **Production Connector / Plant Connectivity** as the destination type, associated with the above Production Connector / SAP Plant Connectivity system

  These service providers provide shop floor services (OPC UA method) which can be triggered by the automation sequence, for example, to start a machine operation.
- One or more third-party services

  You need to connect the Production Connector / SAP Plant Connectivity web server automatically created in the **Manage Web Servers** app with other related web servers and add the web services you would like to use. For details, see Manage Web Servers.

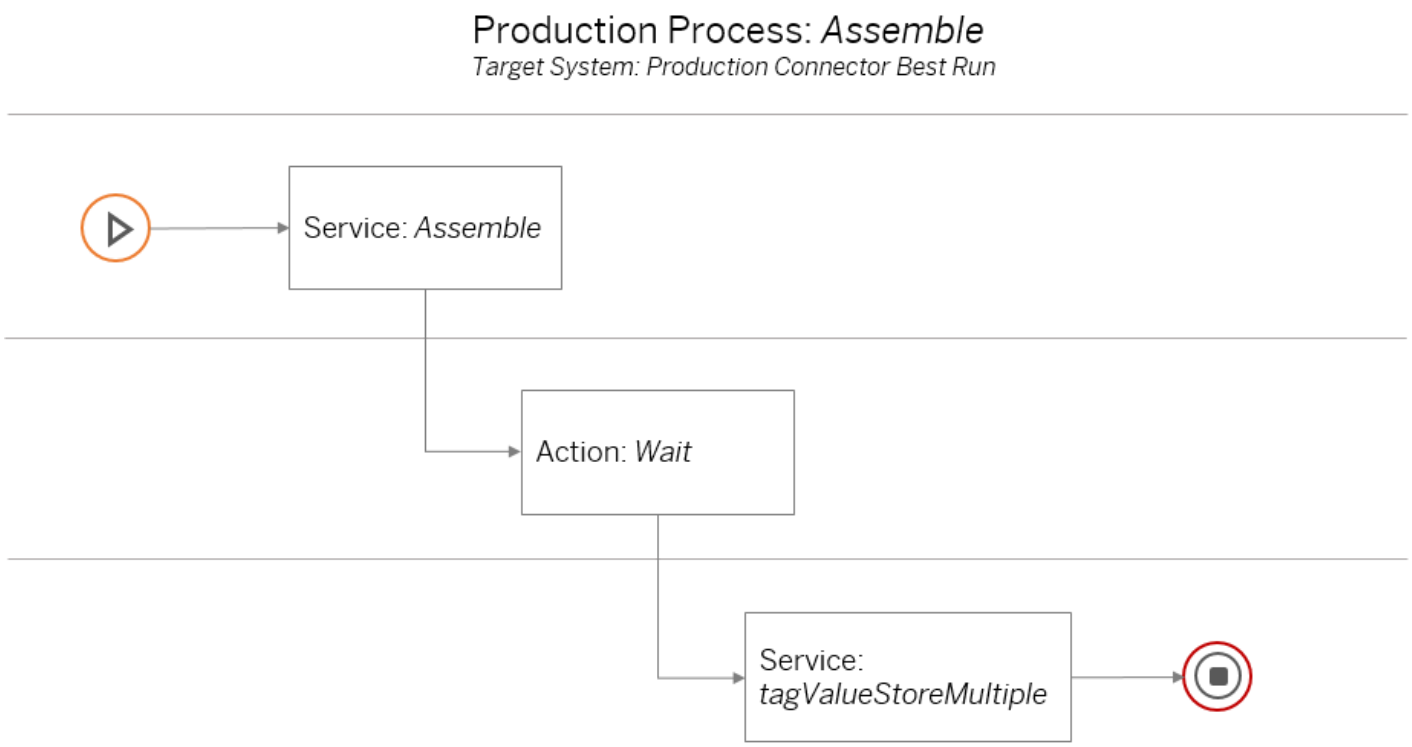
Example

Create a production process to use a robot to assemble a product. After a short pause, write values to some indicators. You can also define a conditional branch to cover error handling in the process.

When the production process is started, the shop floor service (OPC UA method) `Assemble` provided by the external service provider `Robot A` is called. This service performs some assembly activities, for example, assembles a bike.

After one second, the production connectivity model service `tagValueStoreMultiple` provided by the Production Connector / SAP Plant Connectivity web server is called and the output values of the `Assemble` service are stored in some tags.

The interval of one second is defined by the `@Wait` function.



Scenario 2: Manufacturing Execution Automation

To manage and control manufacturing and shop floor operations, design a cloud process. By enabling this kind of production process, you can automate activities (such as start an SFC) or enforce activities (such as data collection) on the shop floor.

For this scenario, you need to prepare the following master data:

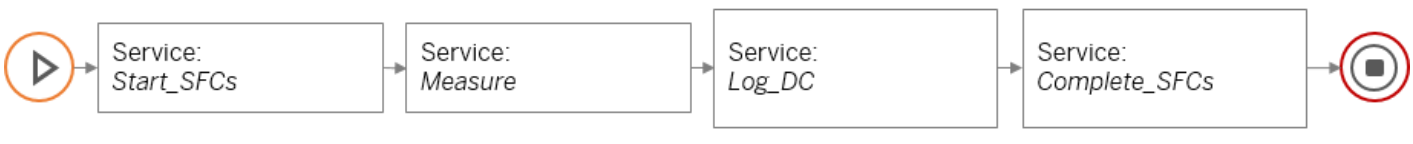
- One or more web services

  You need to connect the DMC cloud web server automatically created in the **Manage Web Servers** app with other related web servers and add the web services you would like to use. For details, see Manage Web Servers.

Example

- Start an SFC at an operation activity on a particular resource (service: `Start_SFCs`)
- Measure object (service: `Measure`)
- Collect data from the shop floor (service: `Log_DC`)
- Complete the SFC (service: `Complete_SFCs`)

For more information about these services, refer to Available Services and Subprocesses.



Scenario 3: Convergence of Manufacturing Execution and Machine Automation

In the third scenario, you combine scenarios 1 and 2 to embed an automation sequence inside a cloud process.

Create a production process that is run in the cloud and another production process (automation sequence) that is run by a Production Connector / SAP Plant Connectivity system. Then embed the second in the first.

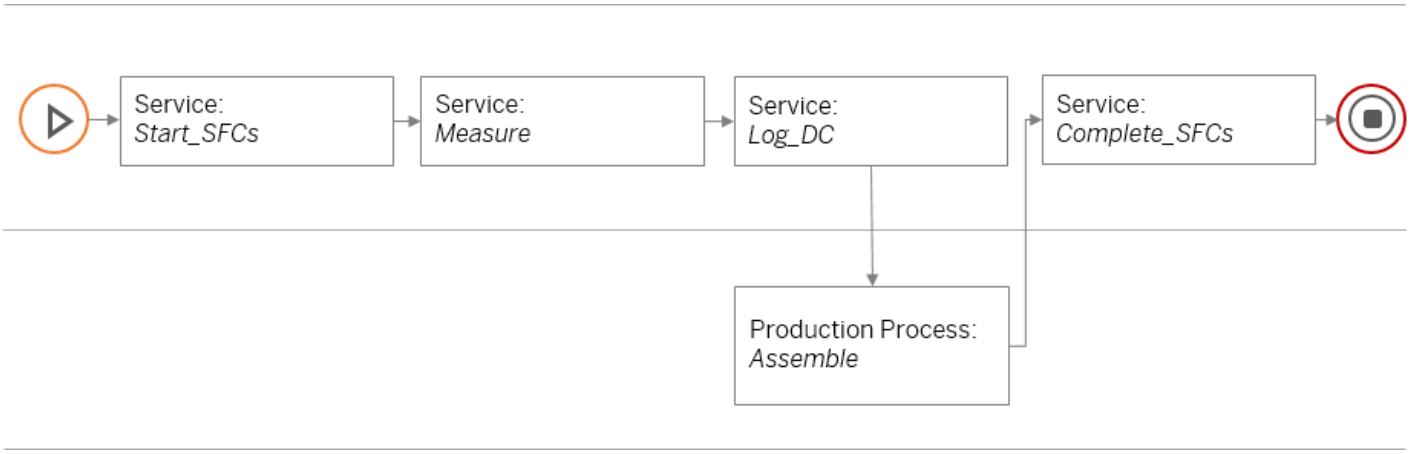
ℹ Note

To embed a production process in another production process, both production processes must be included in the same production process design.

For this scenario, you need to prepare the master data required for both scenario 1 and scenario 2.

Example

Scenario 3: Compared with the example for scenario 2, after starting an SFC, a robot assembles the product.



Similarly, you can also embed a cloud process inside an automation sequence.

Prerequisites

There are some prerequisites to use the Production Process Designer.

Create a Destination for SAP Digital Manufacturing

To call SAP Digital Manufacturing services or cloud processes defined in the production process designer, create a destination for SAP Digital Manufacturing in the SAP BTP cockpit.

Prerequisites

- You are a member of the SAP Business Technology Platform global account.
- You have obtained the service key information of an SAP Digital Manufacturing service instance. For more information, see Prepare for API Integration.

Context

DMC cloud services are API services provided by SAP Digital Manufacturing. For details, see SAP Business Accelerator Hub.

The actual services available may be more than those listed under the public API packages in SAP Business Accelerator Hub. There are some internal services as well. You can check all services under DMC group in the **Manage Service Registry** app.

Procedure

1. Go to your subaccount.
    a. Log on to the SAP BTP cockpit and select a region.
    b. Select a global account.
    c. Select a subaccount that you're working on for production process.
2. Choose **Connectivity** > **Destinations**.
3. Choose **Create Destination**.
4. Specify the destination information as follows:

Property	Value
Name	`SAP_DMC_DEFAULT_SERVICE_KEY`
Type	`HTTP`
Description	Not required
URL	Service key: `public-api-endpoint` Note that this URL does not refer to the authentication endpoint. A typical URL, for example, should look like `https://api.xxxx.dmc.cloud.sap`.
Proxy Type	`Internet`
Authentication	`OAuth2ClientCredentials`
Client ID	Service key: `clientid`
Client Secret	Service key: `clientsecret`
Token Service URL	Enter the `url` in the service key and add `/oauth/token` to the end. A typical token service URL, for example, should look like `https://{$Subdomain}.authentication.xxxx.hana.ondemand.com/oauth/token`
Token Service User	Not required.
Token Service Password	Not required.

Note

You can find the `clientid`, `clientsecret` and token service `url` in the service key of service instance by choosing **Instances and Subscriptions** → Select an instance → **Service Keys**. See "Create service key" section in Prepare for API Integration for more information.

The services can have updates upon new releases. If you come across errors when calling the services, you have to re-create the service key and destination.

5. Save the destination.

Results

The SAP Digital Manufacturing cloud runtime web server (`DMC_Cloud`) is automatically created in **Manage Web Servers** app and has SAP Digital Manufacturing cloud services under it. The destination you created in SAP BTP cockpit is used automatically as the default destination for the SAP Digital Manufacturing cloud runtime web server. For details, see Manage Web Servers.

Authentication for Automation Sequence and Subscription

Authentication is required if you want to use SAP Digital Manufacturing service in automation sequence or subscribe to an indicator to invoke SAP Digital Manufacturing service according to predefined conditions when the value of the indicator is changed.

1. Create service instance and service key in SAP Business Technology Platform. Note down `clientid` and `clientsecret`. For details, see Prepare for API Integration.
2. Set up the default destination with the `clientid` and `clientsecret`. For details, see Create a Destination for SAP Digital Manufacturing.
3. Download SAML metadata.

   > **Note**
   >
   > The SAML OAuth authentication is available for SAP Plant Connectivity versions 15.5 (SP00) or lower.
   >
   > As of SAP Plant Connectivity 15.5 SP01 onward, the Production Connector / SAP Plant Connectivity to SAP Digital Manufacturing authentication process uses the X.509 certificate authentication. For details, see Certificates. As you don't need to add trust configuration in the tenant subaccount any more, skip to step 7.

   a. Add SAP Plant Connectivity system in Production Connectivity Model using the **Configure Production Connectivity** app in the **Manufacturing Automation** section on launchpad. For more information, see Configure Production Connectivity.

   b. In the **Certificates** section, download the SAML metadata file using the **Generate New Certificate** option or **Select Certificate** from the list.
4. Add a new trust configuration for this SAP Plant Connectivity in the tenant subaccount by uploading the SAML metadata XML.

   > **Note**
   >
   > SAP Plant Connectivity works as an identity provider in this case. For more information, see Establish Trust and Federation with UAA Using Any SAML Identity Provider.
5. Create a new role collection and assign the role `Automation_Technical_User` from the service instance, with an application identifier starting with `dmc-services-<randomString>`, to the role collection. This application gets created when you create a service instance of SAP Digital Manufacturing (`digital-manufacturing-services`) in step 1. For more information, see Configure Authorization by Creating Role Collections.
6. Assign the role collection to a technical user: `pco_integration_user`.

   > **Note**
   >
   > For more information, see Create Users.

   a. Create a new user with the following information:

   - User name: `pco_integration_user`
   - Email: Any email would work.

   b. Assign the role collection created in the step above.
7. Set up connection from Production Connector / SAP Plant Connectivity web server to SAP Digital Manufacturing Cloud web server. For details, see Setting up Connections Between Web Servers.
8. Add and deploy services. For details, see Adding and Deploying Services from a Connected Web Server.

Related Information

Available Services and Subprocesses
Manage Subscriptions

Configure Shop Floor Services (OPC UA Methods)

Shop floor services (OPC UA methods) are services provided by external service providers that are created on the selected runtime Production Connector / SAP Plant Connectivity system.

Context

You need to configure these services in the **Manage Service Providers** app first to use them in the automation sequence and subscription. To do so, follow the steps below:

1. Open the **Manage Service Provider** app and choose *(Create New Service Provider)*.
2. Input the name and description of the service provider.
3. Choose **Production Connector / Plant Connectivity** as the destination type.
4. Choose **External** as the service provider type.
5. Select the Production Connector / SAP Plant Connectivity system.

   Production Connector / SAP Plant Connectivity systems configured in using the Configure Production Connectivity app are available for selection.
6. Choose **OPCUAServer** as the Production Connector / SAP Plant Connectivity usage type.
7. Configure the maximum number of retry attempts and retry interval.
8. Configure the **Data Source Properties** and **Security** section.

   See Step 6 in Configuring Service Provider for Services-Based Communication for details.
9. Choose **Save**.
10. Configure services in the service provider.

    See Configuring Services in Service Provider for details.

A client proxy contains the authentication details required when the service receives a call from Production Connector / SAP Plant Connectivity. A client proxy is created as a destination in Production Connector / SAP Plant Connectivity.

To create a client proxy:

1. In the **Manage Service Providers** app, select the service provider you have created.
2. Choose the **Service** tab.
3. Select the service group and choose a service.
4. Choose **Client Proxy Configuration**.
5. Choose *(Add)* to add a client proxy.
6. Input the name and description of the client proxy.
7. Choose **Submit for Deployment** if you want to add the client proxy to a deployment group.

For details, see Configuring Services in Service Provider.

> **Note**
>
> You can edit or delete a client proxy only if it is in the **Draft** status. If you want to edit a client proxy that is in the **Deployed** status, you must first remove the client proxy from the deployment group. For more information on deployment groups, see Deploy Shop Floor Elements.

Configure Client Proxy for Third-Party Service

To use a third-party service in an automation sequence or subscription, the service must have a client proxy configured for it. When defining the automation sequence or subscription, you either need to create a new client proxy or select an existing client proxy for this service.

Prerequisites

- You have defined the service metadata in the **Manage Service Registry** app. For details, see Create an API Service (RESTful and OData).
- You have created a web server in the **Manage Web Servers** app. For details, see Manage Web Servers.
- You have assigned the services with web server in the **Manage Service Registry** app. For details, see Create an API Service (RESTful and OData).

The third-party service, called by an automation sequence or subscription, supports the following authentication methods:

1. No authentication (anonymous)
2. Basic authentication
3. OAuth 2.0 Client Credentials authentication
4. OAuth 2.0 SAML Bearer Assertion authentication (cloud services only)
5. X.509 certificate authentication

- For authentication methods 2, 3, 4: You have created a destination in SAP BTP cockpit first and then added the destination, connected the Production Connector / SAP Plant Connectivity runtime web server with the web server you have created, and added and deployed services in the **Manage Web Servers** app. For details, see Managing Destinations and Manage Web Servers.
- For authentication method 5: You have connected the Production Connector / SAP Plant Connectivity runtime web server with the web server you have created, selected the service certificate, and added and deployed services in the **Manage Web Servers** app. For details, see Manage Web Servers and Adding and Deploying Services from a Connected Web Server.

Context

For any client proxy for a third-party service, some configurations are predefined in the **Manage Service Registry** app for the service. For example, the endpoint URL and headers.

Procedure

1. Edit a production process.
    a. Open the **Design Production Processes** app.

b. Open a production process design.

c. Select a production process with a Production Connector or SAP Plant Connectivity runtime environment.

2. In the production process editor, drag and drop a third-party service to the canvas.
3. On the right panel, choose the link next to **Client Proxy**.
4. In the **Select a Client Proxy** window, select an existing client proxy or create a new client proxy.

   Choose **From Destination** to a create client proxy from the destination that you added to the web server.

   Choose **From Certificate** to create a client proxy from the certificate that you added to the web server.

   Choose **Anonymously** to create a client proxy if the service has no authentication (anonymous).

   > **Note**
   > To use a DMC cloud service or registered cloud process in an automation sequence or subscription, choose the link next to **Client Proxy** and select `SAP_DMC_DEFAULT_SERVICE_KEY` as the client proxy destination.

5. Save your entries.

Prerequisites for Using Third-Party Service in Cloud Process

There are some prerequisites for using a third-party service in a cloud process, timer, or business rule.

- You have defined the service metadata in the **Manage Service Registry** app. For details, see Create an API Service (RESTful and OData).
- You have created a web server in the **Manage Web Servers** app. For details, see Manage Web Servers.
- You have assigned the services with web server in the **Manage Service Registry** app. For details, see Create an API Service (RESTful and OData).

The third-party service, called by a cloud process, timer, or business rule, supports the following authentication methods:

1. No authentication (anonymous)
2. Basic authentication
3. OAuth 2.0 Client Credentials authentication (cloud services only)

- For authentication methods 2, 3: You have created a destination in SAP BTP cockpit first and then added the destination, connected the runtime web server with the web server you have created in the **Manage Web Servers** app. For details, see Managing Destinations and Manage Web Servers.

Restrictions for Cloud Processes

See the related section in 3379404 for details.

Production Process Administration

Use the **Production Process Administration** app to centrally perform administrative tasks that are relevant to production processes.

> **Note**
> Only deployed and active cloud processes are displayed on the list page.

Key Features

- Manage work groups
- Change log level for cloud processes and automation sequences
- Configure debug mode settings for cloud processes
- Configure booster mode settings for cloud processes
- Download archived process instance logs as a backup
- Delete automatically or manually cloud process or automation sequence instances that are not in “running” status

Related Information

Manage Work Groups

Manage Work Groups

A work group is a group of users that work collectively as a unit. With work groups, you can manage users in production process designs or deployment groups more easily.

You can delete a work group as long as it's not being used in any production process designs.

Related Information

Manage Users in a Production Process Design
Manage Users in a Deployment Group

Superuser for Managing Users and Work Groups

A user with the role of `Manufacturing_Admin` can manage users for production process designs, deployment groups, and work groups even though the user himself or herself is not assigned to them. This avoids the situation where the only authorized user is blocked from accessing the system for whatever reason (for example, the user has left the company) and the relevant objects in the system become unusable or user management for these objects become impossible.

The so-called superuser is not allowed to edit the relevant objects, such as updating a production process or adding shop floor elements to a deployment group. The role of `Manufacturing_Admin` is only for managing users in case of above-mentioned situations.

The special permissions for the role `Manufacturing_Admin` are as follows:

App	User Management Actions
Design Production Processes	• Add users and work groups to a production process design • Remove users and work groups from a production process design

App	User Management Actions
Production Process Administration	• Add users to a work group • Remove users from a work group • Delete work groups that are not assigned to any production process designs • Add more work groups, if necessary
Deploy Shop Floor Elements	• Add users to a deployment group • Remove users from a deployment group

Create a Work Group

You use this procedure to create work groups in the Production Process Administration app.

Prerequisites

You have the role of `Manufacturing_Admin`.

Procedure

1. Open the Production Process Administration app.
2. Choose Manage Work Groups > Create.
3. Enter a name and description for the work group.
4. Choose Create.

Manage Users in a Work Group

You can perform different operations on a work group such as adding and removing users.

Prerequisites

- You have the role of `Manufacturing_Admin`.
- You are assigned to the work group.

Context

As the creator of a work group, you are automatically assigned to it.

A work group must contain at least one user. If there are other users in the work group, you can remove yourself from the work group.

Procedure

1. Open the Production Process Administration app.

2. Choose **Manage Work Groups**.
3. To add more users, choose *(Add)*.
4. To remove users from the work group, select the users and then choose *(Delete)*.

Log Level

You can change the log level of both cloud processes and automation sequences with the **Production Process Administration** app.

`Production_Engineer` or `Production_Supervisor` can change the log level of cloud processes while `Automation_Engineer` can do so for automation sequences.

The log level decides the type of information recorded and displayed in the **Monitor Production Processes** app. There are four log levels:

Log Level Type	Information Displayed in the Monitor Production Processes app
Error	all the error information, including error information of the error end and error catch controls
Info	all the error information the start information, end information of each step the start information of the conditional gateway and which sequence flow is taken the start information of the parallel gateway
Debug	all the Info information the input parameter, output parameter of each step the variable information the request and response information
None	no information displayed

You can edit the log level of cloud processes in the dropdown list on the list page.

To edit the log level of automation sequences (only for Production Connector, SAP Plant Connectivity version 15.5.4 or above), choose **Change Automation Sequence Log Level**. As the automation sequence log level follows the log level of its Production Connector / SAP Plant Connectivity system, choose the Production Connector / SAP Plant Connectivity system of your preferred automation sequence and edit the log level.

i Note

You can designate the log level before and during execution. If a new log level is defined during execution, the process instance that is currently running will not be affected. This log level will only take effect in the new process instance.

If no log level is set, the default option will use the **Error** log level.

You can quickly navigate to the corresponding production process page and process instance list page by choosing **Design Production Processes** or **Monitor Production Processes** from the popover of the cloud process name link on the list page.

Related Information

Configure Debug Mode Settings

You can configure the debug mode settings for cloud processes in the **Production Process Administration** app.

Prerequisites

You have the role of `Production_Engineer`.

Procedure

1. Open the **Production Process Administration** app.
2. Under the **Debug Mode** column of the list page, switch on the toggle of your preferred cloud process.
3. In the warning message box, choose **OK**.

   The cloud process will then be executed in debug mode once it is triggered.

> **Note**
>
> Choose **Debug** in the process instance detail page of the **Monitor Production Processes** app to enter the debug session. You cannot access the debug session in the **Design Production Processes** app when there is another one in progress. Terminate the current session and enter a new one if you want to do so. Debug mode is automatically set back to “Off” once the debug session is triggered.

Related Information

Debug Deployed Production Processes
Monitor and Track Production Processes

Booster Mode

The production process execution performance is optimized with the booster mode.

You can configure the booster mode of cloud processes directly on the list page of the **Production Process Administration** app. By default, the **Booster Mode** toggle is set as “on” by the system.

You must have the role of `Production_Engineer` or `Production_Supervisor` to perform the action.

> **Note**
>
> You still need to turn on the booster mode toggle manually for existing processes before 2308 (including their updated versions), as the updated production process inherits its booster mode status from the old one.

	Booster Mode - On	**Booster Mode - Off**
Runtime performance	Reduced execution time	Normal duration
Process instance list view	Displayed	Displayed
Process instance detailed page view	Displayed	Displayed

	Booster Mode - On	Booster Mode - Off
Parallel gateway execution	Execution time can be further reduced for processes with service elements.	Normal parallel execution time **Note** You are not recommended to use the parallel element if the execution time of your parallel branch is less than 6 seconds, as the time saved is unnoticeable. Instead, arrange your flows in sequential order.
Automatic retry	Normal	Normal

Download Archived Logs

SAP Digital Manufacturing archives process instance logs from the previous day in the background and saves them in folders by day. You can download logs of the past three months and back up them locally.

Prerequisites

You have the role of `Production_Supervisor`, `Production_Engineer`, `Automation_Engineer` or `Plant_Manager`.

Context

To avoid performance issues, back up process instance logs by downloading archived data files and delete regularly process instance data.

Procedure

1. Open the **Production Process Administration** app.
2. Choose **Back Up and Delete** > **Download Archived Logs** in the upper right corner.
3. In the **Download Archived Logs** window, Choose **Download** for logs that you want to download.

   Note

   You can also search by keywords.

   The archive download setting uses UTC±00:00 as standard time.

Delete Process Instances

You can define a period to delete automatically or manually cloud process or automation sequence instances that are not in “running” status with the **Production Process Administration** app. After deletion, you won’t be able to check these process instances in the app anymore.

Prerequisites

You have the role of `Production_Supervisor` or `Production_Engineer`.

Context

To avoid performance issues, process instance data is recommended to be deleted regularly.

> **Note**
>
> The detail view of “running” status process instances in the **Monitor Production Processes** app might be affected if the delete setting is enabled.

You can view the activity logs of process instance deletion settings by choosing **View Change History** in both the manual and automatic deletion settings window. The system logs the activity different users have performed.

Manual Deletion

Procedure

1. Choose **Back Up and Delete** > **Instant Deletion** in the upper right corner.
2. In the **Instant Deletion** window, specify the period of data that you want to delete.

   > **Note**
   >
   > The system uses UTC±00:00 as standard time in the settings.

3. Choose **Delete**.

   The instances within the period will be deleted immediately.

Automatic Deletion

Procedure

1. Choose **Back Up and Delete** > **Automatic Deletion Settings** in the upper right corner.
2. In the **Automatic Deletion Settings** window, define the period of automatic deletion.

   > **Note**
   >
   > The system uses UTC±00:00 as standard time in the settings.
   >
   > The default is two weeks (or 14 days). You can change it from 2 to 28 days (or 1 to 4 weeks).
   >
   > If the automatic deletion was set to over 4 weeks in release 2208, then 4 weeks will replace the previous setting as the current deletion period.
   >
   > Process instances with failed status are kept four weeks longer than the period you have set.

3. Choose **Save**.

Design Production Processes

Use the **Design Production Processes** app to create production processes in different production process designs.

With the Production Process Designer, you can model production processes and gain transparency when creating the layout of your shop floor. A production process can define the interaction between machines, define rules, actions, and workflows that control the execution on the shop floor, or a mixture of both.

When you deploy and activate production process designs,the system translates to processes into configuration and stores them in the corresponding runtime environments.

A production process design can have more than one version. Any changes to the configurations require a new version. A user must be assigned to a production process design to edit, deploy, or delete it. The production process design can be fully edited in Draft or Modified status, but is partially editable in Failed status. It is validated when being deployed. Once deployed, it can no longer be changed or deployed again. To update a production process design that has been deployed, create a new version.

Production process designs can have the following statuses:

- **Draft**: The production process design has just been created or edited before being submitted to deployment group.
- **Modified**: A new version of a deployed production process design has been saved.
- **Awaiting Deployment**: The production process design has been submitted to the deployment group but not deployed.
- **Deployed**: The production process design has been sent to the runtime environment.
- **Awaiting Un-deployment**: The deployed production process design has been submitted to a new deployment group for deletion.
- **Archived**: The production process design has been deleted from the runtime environment.
- **Failed**: Deployment failed and you need to try again to deploy it.

Production processes can have the following statuses:

- **Editable**: The production process has just been created or edited before being submitted to deployment group or it has been submitted to the deployment group but failed to be sent to the runtime environment.
- **Deployed**: The production process has been sent to the runtime environment.
- **Archived**: The production process has been deleted from the runtime environment.

ℹ Note

You can save your search with filters by applying a smart variant. Smart variant is supported by the **Design Production Processes**, **Deploy Shop Floor Elements**, **Manage Automatic Triggers**, **Monitor Production Processes**, **Production Process Administration**, **Recover Production Processes**, and **Manage Service Registry** apps. To apply a smart variant, choose the *(Select View)* next to the **Standard** text in the header of the list page and then choose **Save As**. Add a name for the variant.

For an example of creating a production process in the **Design Production Processes** app, see Creating a Printing Production Process.

ℹ Note

The Production Process Designer uses the time zone of your selected plant as the standard time. If no plant is selected, the time of your current browser will be displayed instead.

Concepts

A few concepts and background information for the Production Process Designer before use.

Available Services and Subprocesses

In the **Design Production Processes** app in SAP Digital Manufacturing (DMC), different types of production processes use different services. Additionally, a process can embed other processes as subprocesses for more flexible reuse.

ℹ Note

In the service library, services with web server assigned are grouped as root node while services without web server assigned are put under one independent folder. If the service is not assigned with any web server, it cannot be used in the process.

To use snippet, see Create and Use Snippets.

A parent process can never nest in a subprocess. For example, `Process A` → `Process B` → `Process A` is not allowed.

Cloud runtime environment

Process Runtime Web Server	Web Server to Connect	Object Included	Note
DMC cloud web server	(not required)	DMC cloud services	[1]
		read/write indicators	[2], [15]
		read/write indicators locally	[3]
		read/write indicators flexibly	[19]
		POD plugins	[4]
		registered cloud processes	[5]
	user-defined web server (cloud)	third-party services (RESTful, OData or SOAP)	[6], [7], [17]
	user-defined web server (on-premise)		
	(not required)	cloud processes (subprocess)	[20]
	Production Connector on Windows web server / SAP Plant Connectivity on Windows web server	automation sequences (subprocess)	[8], [15]

Production Connector / SAP Plant Connectivity runtime environment

You should have configured Production Connectivity Model with the shop floor system added first before creating automation sequences. For details, see About Production Connectivity Model.

Process Runtime Web Server	Web Server to Connect	Object Included	Note
Production Connector on Windows web server / SAP Plant Connectivity on Windows web server	(not required)	functions	[9]
		shop floor services (OPC UA methods)	[10]
		write indicators	[2]

Process Runtime Web Server	Web Server to Connect	Object Included	Note
	DMC cloud web server	DMC cloud services	[1], [11], [12], [16]
		registered cloud processes	[5], [11], [12], [16]
	user-defined web server (cloud)	third-party services (RESTful)	[6], [7], [12]
	user-defined web server (on-premise)	third-party services (RESTful)	
	(not required)	automation sequences (subprocess)	[14]
	DMC cloud web server	cloud processes (subprocess)	[18], [13]

ℹ Note

1. For details, see Create a Destination for SAP Digital Manufacturing.
2. Read or write values from indicators on asset/equipment.

   ### → Tip

   Applicability of the Terms Asset and Equipment in this Documentation

   Asset or equipment refers to a digital representation of a physical asset, a machine, or a tool used in production.

   Please refer to the applicability of the terms "asset" and "equipment" in the context of this documentation:

   - The term asset is applicable for those tenants that are onboarded to the Asset Model.
   - The term equipment is applicable for those tenants that are onboarded to the Asset Central Model.

   There are two types of indicators available: 1) indicator mapped to a primitive data type tag; 2) indicator mapped to a component (primitive data type) of a structured data type tag. Note that only production processes and automation sequences of Production Connector runtime support the second indicator type. In such cases reading or writing indicator value only acts on the value of the component mapping to the indicator. Other component values under the same tag are not affected. See Mapping Indicators to Tags for details.

   You can check the service details in the **Manage Service Registry** app.

3. Read or write values from indicators locally. Without bothering to assign a shop floor system to the asset, you can save the value temporarily to a local indicator (write) and reuse (read) it later.

   Under the Asset Central Model, the equipment must be onboarded to the Production Connectivity Model and assigned to a plant for the indicators to be used in the Production Process Designer. Only unmapped indicators whose equipment is not assigned to any shop floor system are supported for reading or writing indicators locally.

   However, under the Asset Model, you don't need to onboard the asset first before using indicators in the Production Process Designer. Both mapped and unmapped indicators are supported for reading or writing indicators locally.

   You can check the service details in the **Manage Service Registry** app.

4. For details, see Controls.
5. Cloud processes published from the Production Process Designer to the **Manage Service Registry** app.

6. Any services that you manually register with SAP Digital Manufacturing in the **Manage Service Registry** app. Not all RESTful, OData and SOAP services are supported. For details, see Registering Services in the SaaS Tenant of SAP Digital Manufacturing.

7. Connect the runtime web server with the web server you created to use the services. For details, see Manage Web Servers.

8. Check **Visible to Cloud Runtime** to make the automation sequence you created visible to the current runtime.

9. For details, see Predefined Functions in the Multiple Call Destination System.

10. These services are provided by external service providers that are created on the selected runtime Production Connector / SAP Plant Connectivity system. Besides, a client proxy must be configured for each service. For details, see Configure Shop Floor Services (OPC UA Methods). You need to select the client proxy when using the service or production process.

11. Connect the runtime web server with the DMC cloud web server. Then deploy services you want to use from the connected web server. For details, see Manage Web Servers.

12. A client proxy must be configured for each service or production process. For details, see Configure Client Proxy for Third-Party Service. You need to select the client proxy when using the service or production process.

13. Check **Visible to Production Connector / Plant Connectivity Runtime** to make the cloud process you created visible to Production Connector / SAP Plant Connectivity runtime.

14. The parent and child (embedded) processes must be run by the same Production Connector / SAP Plant Connectivity system.

15. To use indicator services, create an automation sequence or a subscription, you should have configured the Production Connectivity Model with the shop floor system added first. For details, see About Production Connectivity Model.

16. To enable communication between Production Connector / SAP Plant Connectivity and SAP Digital Manufacturing systems, configure Production Connector / SAP Plant Connectivity as a system in SAP Digital Manufacturing for Insights and add the trust configurations. For details, see Authentication for Automation Sequence and Subscription.

17. For details, see Prerequisites for Using Third-Party Service in Cloud Process.

18. Connect the runtime web server with the DMC cloud web server. For details, see Manage Web Servers.

19. Read or write values from indicators you decide through string constant values, variables, parameters, or expressions. Designate the value of the asset name, reference path, and reference name in the value help for the system to identify the indicator. Support for those tenants that are onboarded to the Asset Model only. For details on asset, structure (reference path), and reference name, see Managing Assets.

    For example, for “indicator1” you created under a multi-level structure of “TopGroup1/SubGroup1” in the asset “DemoAsset1” in the **Manage Assets** app, the input value of this indicator can be:

    - assetName: `"DemoAsset1"`
    - referencePath: `"TopGroup1/SubGroup1"`
    - referenceName: `"indicator1"`

    You can check the service details in the **Manage Service Registry** app.

20. If a cloud process is both a subprocess created in the current production process design and a process published to service registry, you should use the subprocess one under the “Production Processes” folder in the left panel instead of the one registered in the service registry (under the “DMC_Cloud/Production Process” folder).

Configure Service Library

Services that are available for use depends on the runtime environment of the particular process. You can always configure the service library to hide and show services to facilitate your design work.

Procedure

1. Open a production process in the editor.
2. On the left panel, under **Services and Processes**, choose **Select Services**.
3. In the **Select Services** window, select to show and deselect to hide a service.

   > **Note**
   >
   > Instead of selecting the top group, you need to select the immediate parent group of the service for it to be available in the left panel.

4. Save your changes.

Controls

Controls are visualizations of different BPMN event elements. Some controls are applicable only to the cloud process; some controls have different usages in the different categories of production processes.

Control	Cloud Process / Automation Sequence
 Start	Cloud Process, Automation Sequence
 End	Cloud Process, Automation Sequence
 Condition	Cloud Process, Automation Sequence
 Parallel	Cloud Process
 Error Catch	Cloud Process
 Error End	Cloud Process
 Wait	Cloud Process

Control	Cloud Process / Automation Sequence
 Script Task	Cloud Process
 POD Message	Cloud Process

Start

A start element represents the first step in a process. Each process has only one start element.

Use the start element to define process input parameters.

ℹ Note

Start element supports variables only.

End

An end element represents the last step in a process. Each process has only one end element.

Use the end element to define process output parameters.

Condition

Use a condition element to create multiple flows in a process but only one flow is executed under a given condition.

- For a cloud process, you can create two or more flows behind the condition element.

  For each flow, define an evaluation expression. The expressions are evaluated **sequentially** following the given order. In other words, an expression is only evaluated when it is the first expression or the previous expressions are all evaluated to `false`.

  When an expression is evaluated and the result is `true`, the corresponding flow is executed. The other flows are all ignored.

  #### ℹ Note

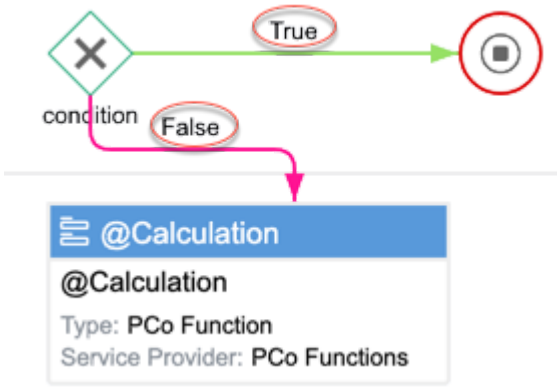
  You can drag and drop the evaluation expressions to adjust the evaluation order.

  #### → Recommendation

  Define the last evaluation expression as the ELSE clause so as to ensure that there is always one flow to execute.

- For an automation sequence, you can create only two flows behind the condition element.

  Adjust the evaluation order of the two flows by double-clicking either `True` or `False` on the lines. And then change `True` to `False`, or vice versa.



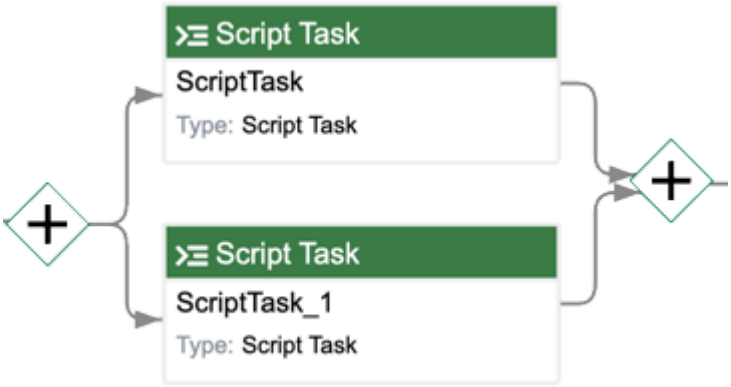
Parallel

Use a parallel element to create multiple flows that are executed simultaneously to save the overall execution time of production process.

Use the parallel elements in pairs. All parallel branches should be started from one parallel element and joined together to another common parallel element at the end.

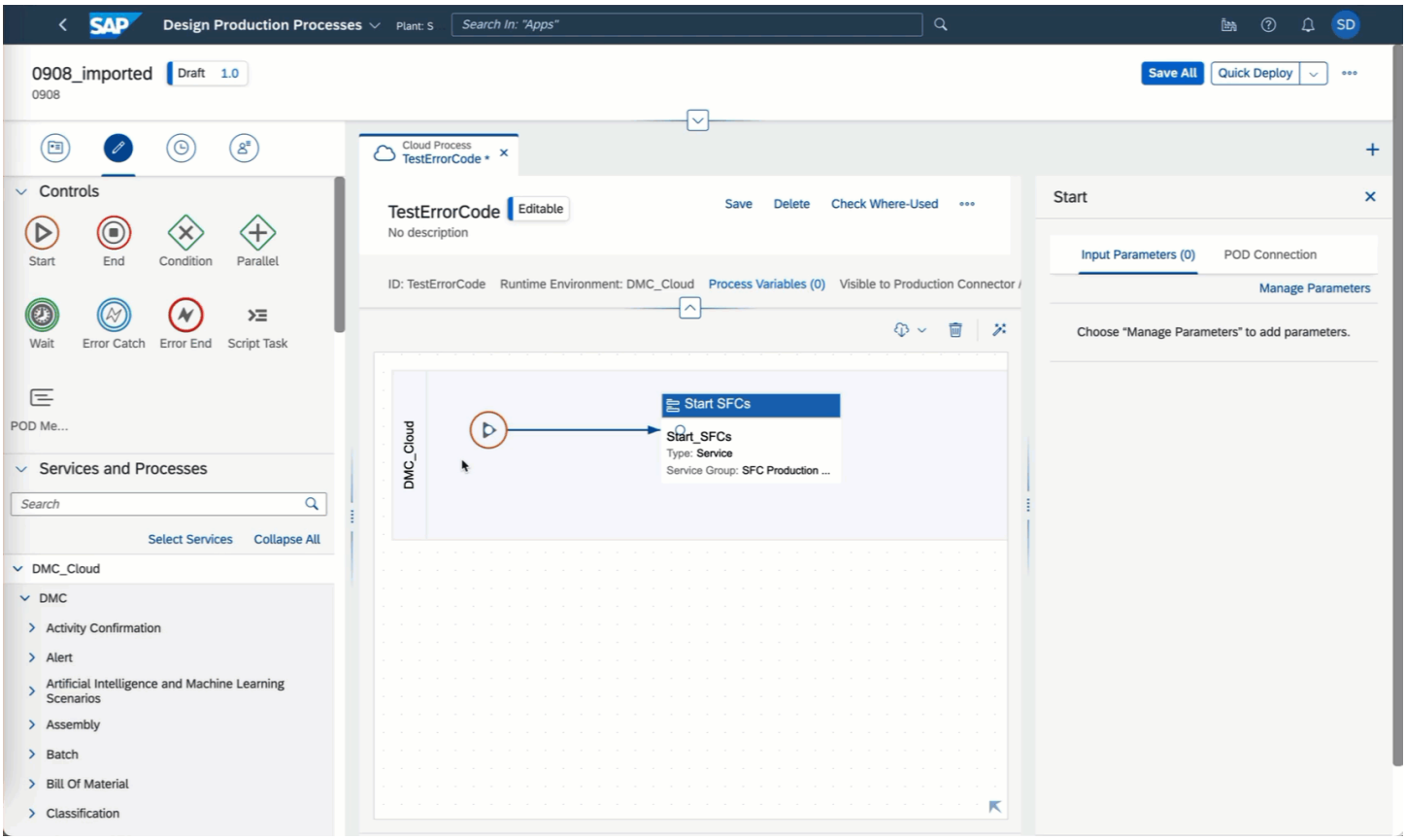
ℹ Note

When the booster mode is off, you are not recommended to use the parallel element if the execution time of your parallel branch is less than 6 seconds, as the time saved is unnoticeable. Instead, arrange your flows in sequential order.



Error Catch

An error catch element catches errors codes thrown by a service/process on which it is defined. Use the error catch element to create alternative flows that handle possible error states the service/process might run into.



→ Tip

1. Drag the error catch element from the left panel.
2. Put the error catch element on the border area of the Start SFCs service element on the canvas, with the service highlighted. Release the error catch element and you will see it is attached to the service element.
3. Select the service error codes you want to catch on the right panel of the error catch settings.
4. Drag and drop the error end element on the canvas. Create a flow from the error catch to the error end element. On the right panel of the error end settings, select the error code you have defined.

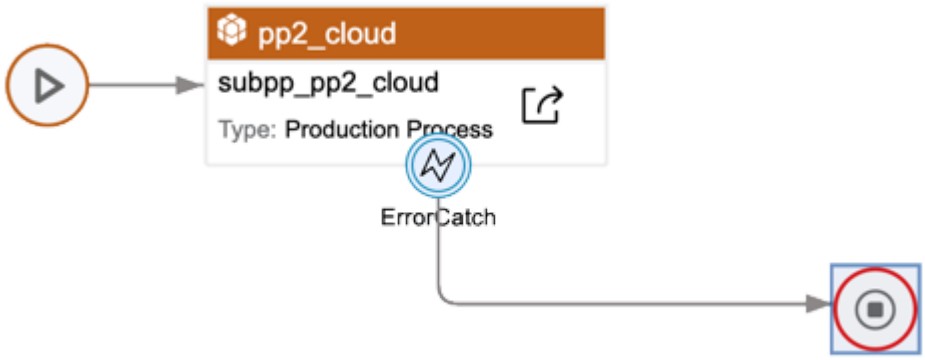
Object	Error code caught by the error catch element
DMC cloud services	HTTP status codes
third-party services	HTTP status codes
cloud processes	error codes defined in the Manage Error Codes settings
registered cloud processes	HTTP status codes, error codes defined in the Manage Error Codes settings

The error code caught by the error catch element can be used as a variable afterwards in these types of flow tasks in the alternative flow:

- Service: input parameter
- Script task element: input parameter

- Subprocess: input parameter
- End element: output parameter

> **Note**



The execution result status for a process catching an error and successfully running to the end is **Completed** in the **Monitor Production Processes** app. See Monitor and Track Production Processes for details.

- Condition element: condition expression

For example, when editing an expression in the condition element, you can input directly `errorCode` and choose the error code from the dropdown list.

- If you want to catch errors thrown by a DMC cloud service, follow the steps below:

  1. Attach an error catch element to the service and define which HTTP status code (401, for example) this error catch element catches.
  2. In the alternative flow leading from the error catch element, add a condition element and define one of the evaluation expressions with the error code variable in the expression editor as `'errorCode' == "HTTP401"`. To do so,

     a. Choose **Functions and Variables** > **Variables** > **'errorCode'**.

     b. Input `" == "`.

     c. Choose **Functions and Variables** > **References**.

     d. Choose **HTTP401** from the list error codes you have defined in Step 1.

- If you want to catch errors thrown by a cloud process (subprocess), follow the steps below:

  1. Define error codes in the **Manage Error Codes** settings. For example, add “StartSfcFailed” and “Start SFC failed.” as the **Key** and **Message**.
  2. In the subprocess, add one or more error end elements and choose in the right panel from the list of error codes you have defined in Step 1.
  3. In the parent process, attach an error catch element to the subprocess and define which error codes (StartSfcFailed, for example) this error catch element catches.
  4. In the alternative flow leading from the error catch element, add a condition element and define one of the evaluation expressions with the error code variable in the expression editor as `'errorCode' == "StartSfcFailed"`. To do so,

     a. Choose **Functions and Variables** > **Variables** > **'errorCode'**.

     b. Input `" == "`.

c. Choose Functions and Variables > References.

d. Choose StartSfcFailed from the list error codes you have defined in Step 3.

For more information, see Define Error Codes for Error End Events.

You can also attach an error catch element to POD plugin, or POD message with Require Confirmation checked. Select from the following error codes:

- INTERNAL: The POD has encountered a fatal internal error that is technical.
- POD_CONFIGURATION: A POD configuration problem is preventing the user from proceeding.

Error End

Use the error end element to display an error message when a process runs into an error.

When a process runs into an error end, the process ends and throws an error message.

You have the option to re-launch asynchronously-run cloud processes terminated with error in a central place later. To do so, switch on the Show Process in Recovery List toggle. Choose from the following termination modes so that the production process will appear in the recovery list based on different reasons:

Termination Mode:

- Failure: A condition on input parameters that became true (e.g. parameter out of range)

  Note

  You can specify the input parameters causing the failure in the dropdown list.

- In Queue: Due to an external context that makes the process waiting (e.g. resource/equipment blocked)

You can see these production process instances in the Recover Production Processes app. See Recover Production Processes for further actions.

Wait

Use a wait element as a placeholder to delay the following process step from execution for a specified duration. For each wait element, provide a name and input a constant value, an expression, or a variable to specify the wait time from 1 to 3600 seconds in the attribute panel.

Note

The wait element is treated as a usual step and cannot be used at the beginning or end of the whole process. When the process is executing the wait step, the process status in the Monitor Production Processes app is Running.

To avoid timeout issues, if the wait time is set to over 30 seconds, you are recommended to run the process asynchronously.

Beware that a system delay within 10 seconds might add to the actual wait time in the runtime environment.

Script Task

A script task is an activity/service defined by users in JavaScript to enhance some simple logics in production process designer for cloud runtime. When a production process execution arrives at the script task, the corresponding script is executed.

Note

There are reserved keywords that cannot be used as variable names assigned to parameters of script task: **out**, **out:print**, **lang:import**, **context**, and **elcontext**.

For each script task, provide a name and description (optional) in the attribute panel. Define input parameters and output parameters. Enter your script code in the code editor. Choose **Insert Parameter** to quickly insert predefined parameter names into the code. Choose *(format)* to apply automatic indentation.

Note

The JavaScript engine (Nashorn) of script task follows ECMAScript 5.1.

Name of the variable in script and input parameter or process variable should be different. Otherwise, it may lead to execution errors in runtime.

Only the following types or objects are supported in script task output parameter.

Supported Type / Object	Example
Null Type	`var a = null`
Boolean Type	`var a = true` `var b = false`
String Type	`var a = "abc"`
Number Type	`var a = 1` `var a = 1.1`
JSON Object	`var a = {"FullName": "Full Name"}`
Array Object	`var a = ["a","b","c"]` `var b = [1,2,3]`

You can use any supported types or objects in the script task code internally, but need to convert them to the above types or objects before sending to script task output parameter. For example, before sending to script task output parameter, use `var a = (new Date()).toISOString()` to get current date time and convert to string.

Choose **Test Run** to do code validation before deployment.

Note

Test run button is only enabled for the creator of the production process.

You can also use codes, for example, `var localDateTimeString = fn.localDateTime("2022-02-15T04:26:45.123Z","Europe/Berlin")`, to convert to your preferred execution timezone. Note that this feature is only applicable to deployed production processes. Test run is not supported.

Note

The execution of script tasks is subject to resource limits, for example, with respect to processing time or memory usage. The limits enforced by the workflow service have the purpose of freeing up resources as early as possible for other tasks. The limits protect against excessive usage, for example, caused by in-efficient programming or unexpected input sizes. If the limits are exceeded, the corresponding workflow instance is put into the ERRONEOUS state. The error is written to the error logs of the workflow instance. You can retrieve the error logs using the REST API. If your scripts reach the resource limits,

analyze the reasons, for example, large input data. Try to reduce the input size or the complexity of the transformations executed on it. For the specific limits that apply to script tasks, see the related section in 3379404 for details.

POD Plugin and POD Message

You can add POD plugin into cloud production process. The production process will call the POD to take action when the POD plugin is executed.

Use POD Message in cloud production process to define message shown in a specified type of POD. Process with steps asking for user's action from POD will wait upon user's further action in POD to proceed with its following steps when it is triggered. You can also view its status as “Running (waiting)” (counted in “Running” KPI carousel) in the **Monitor Production Processes** app. For details, see Monitor and Track Production Processes.

Note

This function is only visible to SAP Digital Manufacturing for execution license users.

1. Configure settings for POD plugin/message in the **POD Connection** tab of **Start** control.

   ### Note

   This step is a prerequisite for using POD plugin/message in the process later.

   At the same time, you need to do some configurations on the POD side:

   - Select **Production Process** in the event section of POD Notifications in the **POD Designer** app. For details, see POD Notification Options.
   - Check the connectivity status to make sure connection is established between POD and process.
   - Use the Process Overview plugin on the POD to check the execution status of the process.

2. Choose **Add**.
3. Select a POD type (Work Center, Operation Activity, Order) in which the POD plugin/message takes effect.
4. Designate corresponding metadata in **Conditions to Select POD** so that the production process knows which POD to call when the POD plugin/message is executed.

   ### Note

   The metadata that you designate in **Conditions to Select POD** must match the **Subscription** options that you select when configuring POD Notifications in the POD Designer. For more information, see POD Notification Options.

   For example, when configuring **POD Notifications** in the **POD Designer**, you select **Work Center** and **Resource** in the **Subscription** section and ensure that **Production Process** is one of the events selected in the **Event** section. Then, when configuring **Conditions to Select for POD** in the **Design Production Processes** app, you ensure that the production process sends processes to the POD that are listening for both the same **Work Center** and **Resource** by designating the same **Work Center** and **Resource** that you selected when configuring **POD Notifications** in the **POD Designer**.

   ### Note

   Both input parameters of the **Start** control or constant values in the value helper are supported.

5. Drag and drop a POD plugin in the service library to the canvas.

6. For each plugin, give it a name and select an action type in the attribute panel. There are two action types:

    - **Launch and Wait**: Execute the POD plugin and wait for input from operator to continue. In the POD, choose **Continue** to finish POD action and continue to next step of process.
    - **Launch and Continue**: Execute the POD plugin and continue to the next step without further input.

    When POD plugin is executed, selected plugin will appear automatically on the selected POD.

    > **Note**
    >
    > It is recommended to use asynchronous call if POD plugins are run with "Launch and Wait", otherwise it may cause timeout.

7. You can also drag and drop a POD Message control to the canvas.
8. Give the POD message a name and input a message in the attribute panel.

    > **Note**
    >
    > Choose **Insert** to add the input parameter of the start element, process variable, or output parameter of the previous element in the POD message. Only primitive data type and primary node of the structured date type (the node with primitive data type that has not any sub-node, for example: `person[0].name`) is supported. Apply escape character (`\" \"`) if you want to use double quotes (`""`) in your message. You can also add your own expression.
    >
    > Message translation is not supported when you change the language setting.

9. Check the **Require Confirmation** checkbox to ask for operator's confirmation in a dialog in POD. Otherwise, the message will appear as message toast on the selected POD and the process is continued without confirmation from POD.

Define Error Codes for Error End Events

In a production process design, you can define error codes and assign them to error end events in various cloud processes. Each error end event can throw errors defined by one particular error code.

Prerequisites

You have the role of `Automation_Engineer` or `Production_Engineer`.

Procedure

1. Open a production process design.
2. Choose **Manage Error Codes** in the upper right corner of the header section.
3. Create one pair of key and message for each error code.

    > **Example**
    >
    > - **Key**: `sfcStartInsufficientQty`
    > - **Message**:
    >
    >   `SFC could not be started because there was no sufficient quantity for the material`
    >
    >   Note that variables are not supported for the message. For example, `SFC 'sfcV' not started` is **not** supported.

Related Information

Controls

Process Variables

Variables defined within a process are used to map parameters between elements within a process. They are not inherited by subprocesses or detectable by the parent process.

Note

Apart from process variables created manually, you can also use error codes as variables. For more information, see Error Catch.

Create a variable for the following purposes:

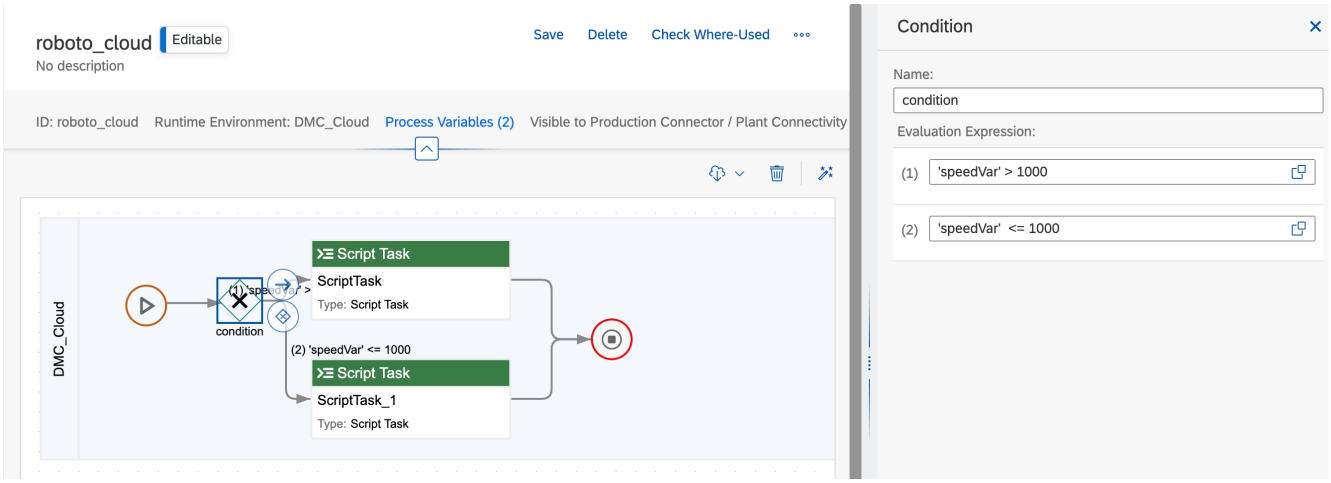
- Assign a process input parameter to a variable
- Assign an output parameter to a variable

The variables can be used to do the following:

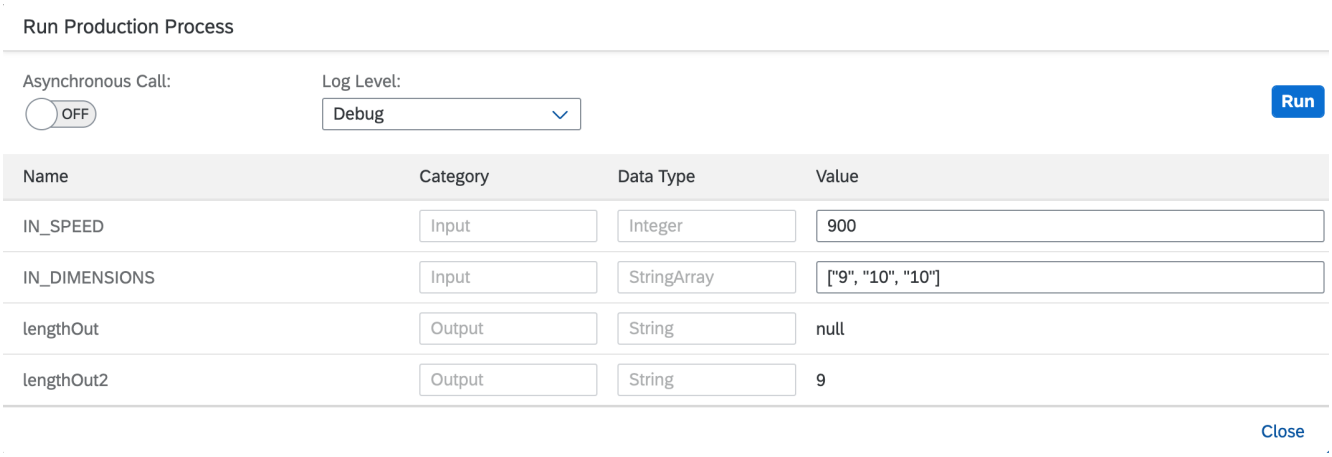
- Define expressions for input parameters
- Define Boolean expressions for conditions that evaluate to either `true` or `false`
- Define values in arrays

Example

1. Create an integer variable `speedVar` and a string array variable `dimensionsVar`.
2. Create two process input parameters on the start element:
    - Integer parameter `IN_SPEED`, assigned to variable `speedVar`
    - String array parameter `IN_DIMENSIONS`, assigned to variable `dimensionsVar`
3. Add two script tasks with the following settings:
    - Flow A: Define both the input and output parameters as `length` (data type: string) with the input parameter value as `'dimensionsVar[0]'`. Define the script as `$output.length = $input.length`.
    - Flow B: Define both the input and output parameters as `length2` (data type: string) with the input parameter value as `'dimensionsVar[0]'`. Define the script as `$output.length2 = $input.length2`.
4. Add a condition element and define the conditions as follows:
    - If `'speedVar' > 1000`, flow A is executed.
    - If `'speedVar' =< 1000`, flow B is executed.



5. Create two process output parameters on the end element:
    - String parameter `lengthOut`, assigned to parameter `'ScriptTask#length'`.
    - String parameter `lengthOut2`, assigned to parameter `'ScriptTask#length2'`.
6. At runtime, the input values for `IN_SPEED` and `IN_DIMENSIONS` are `900` and `["9", "10", "10"]`, respectively.
7. Flow B is executed.
8. The input parameter `length2` in step 3 receives "9" as its value.



Data Types and Data Type Conversion

When defining input or output parameter using expressions with variables or assigning parameters to variables, you must ensure the data types of the parameters and variables either match each other or can be converted.

String

To define a string parameter value, enclose the value with double quotation marks. For example, `"name"`.

Boolean

Only `true` and `false` (case-sensitive) are supported for Boolean values.

Array

To define an array parameter with a fixed array value instead of assigning a variable directly to it, enclose the value using square brackets. And if you want to use a value in an array variable, define it following this pattern: `'variable[index]'`.

Example

Define a string variable as `sVar` and a string array variable as `arrVar`. And then define values for the following parameters:

- A string array parameter: `['sVar', "red", "green"]`
- A string array parameter: `['arrVar[0]']`
- A string parameter: `'arrVar[0]'`

Data Type Conversion

The following table describes the convertibility between different data types.

- A: Type can be cast or converted without any errors or precision loss.
- B: Type can can be converted but sign loss may occur during conversion for negative values.
- C: Type can be converted but precision loss may occur during conversion.
- D: Type can be converted but it may overflow during conversion.
- E: Type can be converted but it depends on the validity of the input string.
- F: Type can't be converted.

Convert From \ Convert To	Boolean	UByte	Byte	Char	UShort	Short	UInteger	Integer	Single (Float)	ULong	Long	Double	DateTime
Boolean	A	A	A	A	A	A	A	A	A	A	A	A	F
UByte	C	A	A	A	A	A	A	A	A	A	A	A	F
Byte	C	B	A	B	B	A	B	A	A	B	A	A	F
Char	C	A	A	A	C	A	B	A	A	B	A	A	F
UShort	C	A	A	A	A	A	A	A	A	A	A	A	F
Short	D	D	D	D	D	A	B	A	A	B	A	A	F
UInteger	D	D	D	D	D	C	A	A	A	A	A	A	F
Integer	D	D	D	D	D	D	D	A	A	B	A	A	F
Single (Float)	D	D	D	D	D	D	D	C	A	C	C	A	F
ULong	D	D	D	D	D	D	D	D	C	A	A	A	F
Long	D	D	D	D	D	D	D	D	D	D	A	D	F
Double	D	D	D	D	D	D	D	D	D	D	C	A	F
DateTime	F	F	F	F	F	F	F	F	F	F	F	F	A
String	E	E	E	E	E	E	E	E	E	E	E	E	E

Note

Arrays follow the same rules. For example, a byte array can also be converted to a string array but cannot be converted to a date time array.

Supported Operators in Expressions

You can use expressions to define parameters and conditions. Refer to the following table for the logical and mathematical operators that are supported in the expressions.

Operator	Description
(	Left parenthesis
)	Right parenthesis
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to
&&	Logical conjunction (And)
||	Logical disjunction (Or) This operator is used to enter an 'or' condition. **Example** Create this expression that evaluates to `true` or `false`: `'TAG_TEMP'<57 ||'TAG_TEMP'>63`.
+	Plus
–	Minus
*	Multiply
/	Divide
%	Modulo operation
^	Power **Note** This is supported for automation sequences only.

Operator	Description
&	Concatenation You can use this operator to concatenate two string values. **Example** `'Temperature' & " degrees Celsius"` `Temperature` is the name of an integer variable, enclosed in single quotes. The string value that is to be appended to the variable value is enclosed in double quotes. At the runtime, if the value for `Temperature` is `44`, it is first converted into a string value and then combined with the string value as `44 degrees Celsius`. **Note** For the cloud runtime environment, you can also use + for the same purpose.
!	Logical negation (Not)

Supported Functions in Expressions in Cloud Processes

For a cloud process, you can use some functions in the expressions to define parameters or conditions. Refer to the following table for the supported functions.

Function	Description
`var:lte(varName, value)`	Lower than or equals
`var:lt(varName, value)`	Lower than
`var:gte(varName, value)`	Greater than or equals
`var:gt(varName, value)`	Greater than
`var:get(varName)`	Retrieve the value of a variable, using this function won't throw an exception when the variable doesn't exist.
`var:getOrDefault(varName, defaultValue)`	Providing a default value which is returned when the variable isn't set or the value is null.
`var:notEmpty(varName)`	Checks if the variable value is `notEmpty`.
`var:empty(varName)`	The reverse operation of `isEmpty`
`var:eq(varName, value)`	Check if a variable equals to a given value.
`var:ne(varName, value)`	The reverse operation of `equals`
`var:contains(varName, value1, value2, …)`	Check if all values provided are contained within a variable.
`var:containsAny(varName, value1, value2, …)`	Similar to the `contains` function, but `True` will be returned if any (and not all) passed values are contained in the variable.

Function	Description
`var:base64(varName)`	Convert a binary or string variable in `Base64` string.
`var:exists(varName)`	Return `True` if the variable has a non-null value.

Supported Functions in Expressions in Automation Sequences

For an automation sequence or a subscription, you can use some functions in the expressions to define parameters or conditions.

Refer to this link for the supported functions.

Manage Users in a Production Process Design

Invite a user to work on a production process design by assigning the user or a work group that contains the user to the production process design.

Prerequisites

- You have the role of `Automation_Engineer` or `Production_Engineer`.
- You're assigned to this production process design.

  By default, the creator of a production process design is automatically assigned to it.

Procedure

1. Open the **Design Production Processes** app.
2. Open a production process design.
3. On the **Users** tab, add or remove users and work groups.

Related Information

Manage Work Groups

Activity Logs in a Production Process Design

You can search and view the activity logs of the design time.

The system logs the activity you have performed in the design time of the production process design on the **History** tab. The log covers:

- object name
- object type
- ID
- status change
- date and time of the activity

> **Note**
>
> The system uses the time zone of your selected plant as the standard time. If no plant is selected, the time of your current browser will be displayed instead.

- the user performing the activity

Create a Production Process Design

A production process design is a group of production processes (for example, for one particular production line). Before creating production processes, create a production process design first.

Prerequisites

You have the role of `Automation_Engineer` or `Production_Engineer`.

Context

> **Note**
>
> The name and the version are both case-sensitive.

Procedure

1. Open the **Design Production Processes** app.
2. Choose **Create**.
3. In the **Create Production Process Design** window, specify the required information.
    a. **Name**: For your own memory purposes, enter a name. With characters of any language and a maximal length of 100, the name can be changed at any time for the production process design in draft or modified status. Except for archived production process designs, there must not be any production process designs with duplicated names in a single landscape.
    b. **ID**: An ID is a unique identifier with limited western characters (A–Z, a–z, 0–9, _) for technical use. ID is automatically generated as you type the name. You can also define the ID by yourself.
    c. **Description**
    d. **Version**
    e. **Label**: Add a label with limited western characters (A–Z, a–z, 0–9, _) and a maximal length of 128 to categorize the production process design in this version. All production processes under this design inherit this label as well. Choose from an existing label or create a new one. You can also edit the label later for production process design in **Draft** or **Modified** status. Label information is retained when you copy, export, or import the production process design. You can easily filter out the production process design on the list page with the label you have added. You can also filter out process instances with this label in the **Monitor Production Processes** app. See Search and Filter Process Instances for details.
4. Choose the **Create** button.

> **Note**
>
> Choose **Edit Header** next to the production process design name on the detailed page to edit related information.
>
> Choose **Validate** to check any dependency issues that can block the deployment.

Next Steps

Update a Production Process Design

For each production process design, only one draft version is allowed. After a production process design is deployed or deleted, you can create a new version of it. Note that you do not have to be assigned to a production process design to save it as a new version.

Prerequisites

- You have the role of `Automation_Engineer` or `Production_Engineer`.
- The production process design is in any of the following statuses:
    - **Deployed**
    - **Archived**

Context

All production processes are reserved in the new version. You can modify or delete the production processes.

All the users assigned to the previous version are still assigned to the new version. If you were not assigned to the previous version, you are now assigned to the new version.

You can select different versions of production process designs in the **Version** dropdown list inside the production process design detailed page.

Procedure

1. Open the **Design Production Processes** app.
2. Open a production process design.
3. Choose **Edit**.
4. Add your own name or leave it as is.
5. Use the suggested version or enter a new version.
6. Choose **Save as New Version**.

Results

The status of the new version is **Modified**.

When updating a production process design (in **Archived** or **Deployed** status) with changes, as its production process may have already been used in other places, such as another production process or automatic trigger, you will be reminded to check the where-used list before deployment. You can either give up your changes or continue with them.

ℹ Note

If you change the input or output parameters of a production process used by a subscription and then deploy its production process design, the subscription will get updated with this change. If the subscription is already deployed, a modified version will be auto-generated by the system. You need to check the subscription and reassign the parameter value accordingly.

When you deploy a new version of a production process design, the runtime configurations of the previous version will be updated automatically to that of the new version in the runtime environment. Meanwhile, the status of the previous version becomes **Archived**.

> **Note**
>
> You cannot submit modified production process design to deployment group if its previous version is in **Awaiting Un-deployment** status. You need to remove the previous version from the deployment group created for un-deployment to proceed.

Copy a Production Process Design

You can create a new production process design by copying an existing one. Note that you do not have to be assigned to the production process design to copy it.

Prerequisites

You have the role of `Production_Engineer`.

Context

All production processes are reserved in the new production process design. You can modify or delete the production processes or create new ones.

> **Note**
>
> The name and ID for the new production process design must be different from the old one.

All the users assigned to the copied production process design are still assigned to the new one. If you were not assigned to the copied production process design, you are now assigned to the new one.

Procedure

1. Open the **Design Production Processes** app.
2. Open a production process design.
3. In the additional options dropdown menu, choose **Copy** .
4. Enter a new name and ID that is also unique.
5. Optionally, change the description, type or version.
6. Choose **Copy**.

Delete Production Process Designs

When a production process design is in the **Draft**, **Delete**, **Modified**, **Archived** status, you can always delete it. After deploying the production process design from the cloud to the runtime environment, you must first delete the design from the runtime environment (un-deploy) and then from the cloud.

Prerequisites

- You have the role of `Automation_Engineer` or `Production_Engineer`.
- You're assigned to this production process design.

Delete a Deployed Production Process Design from the Runtime Environment

Procedure

1. Open the **Design Production Processes** app.
2. Open a production process design in the status of **Deployed**.
3. Choose **Select Deployment Group** in the dropdown menu of **Quick Delete**.

   > **Note**
   >
   > For a fast deletion, you can choose **Quick Delete** to complete the following steps all at once (while still need to delete it from cloud afterwards).

4. In the **Add to Deployment Group** window, confirm the default deployment group automatically created and choose **Submit for Deployment**. The status of the production process design becomes **Awaiting Un-deployment**.

   Alternatively, instead of using the default deployment group, you can select other existing deployment groups by choosing **Select Another Group**. In the **Select Deployment Group** window, you can also choose *(add)* to create a new deployment group. If you create a new deployment group, you become its administrator automatically.

5. In the Deployment Group window, adjust the shop floor elements that you want to un-deploy.
6. Choose **Deploy and Activate**.

   > **Note**
   >
   > You can still go to the **Deploy Shop Floor Elements** app to un-deploy the production process design afterwards. See Deploy and Activate Shop Floor Elements for details.

Results

- All runtime configurations are deleted.
- The status of the production process design becomes **Archived**.

Delete from the Cloud

Procedure

1. Open the **Design Production Processes** app.
2. Choose *(Filter)* and select **Archived**.
3. For a production process design in the status of **Archived**, choose *(Delete)*.
4. Confirm the deletion.

Delete in Batch

Procedure

1. Open the **Design Production Processes** app.
2. Select the production process designs that you want to delete in batch.
3. Choose **Delete**.

> **Note**
>
> Designs in the **Draft**, **Archived**, **Modified** status will be deleted directly. Designs in the **Awaiting Un-deployment**, **Awaiting Deployment** or **Failed** status will be ignored. Deployed designs will be submitted to a deployment group for your further action.

Export and Import Production Process Designs

You can export production process designs with cloud processes and relevant services (user-defined services and MII services) to a file, and upload to other systems.

Prerequisites

You need to have the role of `Production Engineer` or `Manufacturing Admin` to export and import, or have the role of `Automation_Engineer` to export.

Export

Procedure

1. Open the **Design Production Processes** app.
2. Select the production process designs you want to export.
3. Choose **Export**.
4. Set a password.

   Only production process designs with cloud processes and relevant services can be exported. You can choose **Continue Anyway** or **Close** for production process designs that do not meet the criteria.
5. Choose **OK**.

   You can see the export status in the message strip on the list page. Once the export is finished, you can choose **Download** to download the file. The status of production process designs will be changed to draft after the export.

   You can choose **Check Details** for detailed information about the export.

   - Choose **Skip and Download** to skip exporting failed production process designs and continue exporting other ones.
   - Choose **Retry All** to redo the export for all designs you have selected.
   - Choose **Close** to stop the export.

> **Note**
>
> The detailed information is kept for seven days and will not be available afterwards.

Import

Procedure

1. Open the **Design Production Processes** app.
2. Choose **Import**.
3. Choose the local file that you have downloaded from the source system.
4. Enter the password you have set.
5. Choose **Continue**.

   You can see the import status in the message strip on the list page and choose **Check Details** to open the import manager for detailed information about the import.

   Production process designs with import errors are displayed in the **Wait for Action** tab of the import manager in the following three categories:

   - **Duplicated Design**: There is already the same design in the target system. Choose **Import a Copy** to import it as a separate one and add a new name. Meanwhile, a new ID will be auto-generated. You can also **Overwrite** the existing design with the new one. **Skip** will ignore the selected design.

     > **Note**
     >
     > Existing design in deployed status will be overwritten by a new modified version. If the existing design is in draft or modified status, it will be overwritten in the current version.

   - **Existent Name**: There is already a design with the same name in the target system. Choose **Import a Copy** to rename the design and import it as a separate one. Meanwhile, a new ID will be auto-generated if the imported ID is the same as the current one. **Skip** will ignore the selected design.
   - **Other Error**: All designs with other errors except the above ones are displayed here. Choose **Show Detail** for further error descriptions. You can choose **Skip** to ignore the selected design.

   Production process designs without import errors are displayed in the **In Progress** tab, with their import status information included.

   You can view all production process designs imported by you in the **My Imports** tab on the list page.

Create and Define a Production Process

Create a production process by defining a sequence of activities based on rules. The activities are driven by various kinds of services that are either available in the cloud (including those provided by SAP Digital Manufacturing itself) or provided by an on-premise system.

Prerequisites

- To create, run, update and delete a cloud process, you must have the `Production_Engineer` role.
- To create, run, update and delete an automation sequence, you must have the `Automation_Engineer` role.
- To define error end events and corresponding error catch events for cloud processes, define suitable error codes.

  For more information, see Define Error Codes for Error End Events.
- See Available Services and Subprocesses for other prerequisites in detail.

Context

Depending on the runtime environment, the production processes are divided into two categories: cloud processes that are run in the cloud and automation sequences that are run by a Production Connector / SAP Plant Connectivity system. When creating a production process, you must first specify a runtime environment.

The different categories of production processes have different controls and services available for use.

A production process can embed other production processes for more flexible reuse. There is one limitation for the automation sequences: the parent and child (embedded) processes must be run by the same Production Connector / SAP Plant Connectivity system.

Related Information

Controls
Available Services and Subprocesses
Process Variables
Supported Operators in Expressions
Data Types and Data Type Conversion

Create a Production Process

Procedure

1. Open the **Design Production Processes** app.
2. Open a production process design.
3. Choose *(Create)*.
4. In the **Create Production Process** window, enter the required information.
    - **Name**: For your own memory purposes, enter a name. While names can be duplicated with characters of any language and a maximal length of 100 across production process designs, there must not be any production processes with duplicated names in a single production process design.
    - **ID**: An ID is a unique identifier with limited western characters (A–Z, a–z, 0–9, _) for technical use. ID is automatically generated as you type the name. You can also define the ID by yourself.

      Note

      ID cannot be changed once the production process is created, but the name can be changed at any time for the production process in draft or modified status.

    - **Runtime Type**: Select either **Cloud**, **Production Connector**, **Plant Connectivity**.
    - **Runtime Environment**: Choose the runtime web server in which the production process will be deployed.
    - **Visible to Production Connector / Plant Connectivity Runtime, Visible to Cloud Runtime**: Check to make the production process you created visible to other runtime. Alternatively, you can configure it later by editing header. See Available Services and Subprocesses for details.
5. Choose **Create**.

Note

Choose **Edit Header** next to the production process name on the detailed page to edit related information.

Define a Production Process

Procedure

You may not follow the steps below in a strictly sequential order. They are documented in such a way that you can have an overview of all the steps required for defining a production process.

1. Open a production process in the editor view.

   If the process is not already open, click the process in the repository view () and then go to the editor view ().

2. Drag and drop required elements - controls, services and processes - to the canvas, within a swimlane.

   Use Shift to select more than one element of the process.

   When you embed a process into another process, the former process becomes a subprocess of the latter process.

   > **Note**
   >
   > Some services are by default hidden. Edit the service library to display them, if required. For available services to different runtimes, see Available Services and Subprocesses.

3. Select a destination from web server for third-party services.
4. Connect the elements with lines.
5. If required, define evaluation expressions for the condition elements.

   > **Note**
   >
   > If it is a cloud process, you can drag the expressions up and down to change the order of evaluation.

6. If required, create process variables for parameter mapping.

   > **Note**
   >
   > Parameter data type of **Start** element should match the variable data type.

7. If required, create process input parameters on the **Start** control and define the parameter values.

   > **Note**
   >
   > The parameter or variable name can only contain A–Z, a–z, 0–9 and underscores. It must start with a letter and end with either a letter or a number. Double underscore is not allowed. Reserved words (`mod`, `debugger`, `lt`, `errorCode`, `do`, `while`, `empty`, `div`, `else`, `continue`, `function`, `catch`, `if`, `case`, `new`, `void`, `in`, `var`, `finally`, `this`, `variableContainer`, `eq`, `task`, `null`, `ne`, `true`, `subdomain`, `try`, `tokenType`, `for`, `delete`, `switch`, `not`, `default`, `and`, `ge`, `typeof`, `execution`, `or`, `break`, `false`, `gt`, `instanceof`, `with`, `throw`, `tenantId`, `le`, `authenticatedUserId`, `return`, `MOD`, `DEBUGGER`, `LT`, `ERRORCODE`, `DO`, `WHILE`, `EMPTY`, `DIV`, `ELSE`, `CONTINUE`, `FUNCTION`, `CATCH`, `IF`, `CASE`, `NEW`, `VOID`, `IN`, `VAR`, `FINALLY`, `THIS`, `VARIABLECONTAINER`, `EQ`, `TASK`, `NULL`, `NE`, `TRUE`, `SUBDOMAIN`, `TRY`, `TOKENTYPE`, `FOR`, `DELETE`, `SWITCH`, `NOT`, `DEFAULT`, `AND`, `GE`, `TYPEOF`, `EXECUTION`, `OR`, `BREAK`, `FALSE`, `GT`, `INSTANCEOF`, `WITH`, `THROW`, `TENANTID`, `LE`, `AUTHENTICATEDUSERID`, `RETURN`, `Undefined`, `Null`, `Boolean`, `String`, `Number`, `Object`, `class`, `enum`, `extends`, `super`, `const`, `export`, `import`) are also not permitted.
   >
   > Choose **Manage Parameters** for batch editing of parameters. You can check the **Required** box to set the value of parameter as mandatory for cloud processes.

You can use the "Execute Asynchronously" option of cloud processes (run as subprocesses) to prevent the execution result from affecting the main process. This asynchronous execution of subprocesses is displayed separately in the **Monitor Production Processes** app.

The Production Process Designer supports both primitive and complex data types as variable or parameter value.

- **primitive data types**: a set of basic data types, including `Boolean`, `String`, `Byte`, `DateTime`, `Integer`, `Short`, `Long`, `Double`, `Single (Float)`

  Note

  `UByte`, `UInteger`, `UShort`, `ULong`, `Char` are also supported in an automation sequence and subscription.

- **primitive array data types**: array of primitive data types, including `BooleanArray`, `StringArray`, `ByteArray`, `DateTimeArray`, `IntegerArray`, `ShortArray`, `LongArray`, `DoubleArray`, `SingleArray (FloatArray)`

  Note

  `UByteArray`, `UIntegerArray`, `UShortArray`, `ULongArray`, `CharArray` are also supported in an automation sequence and subscription.

- **complex data types**: a composite of primitive, primitive array, or other complex data types, including `Structure` and `StructureArray`. The complex data types in the Production Process Designer are defined as schema under the **Schemas** tab of the **Manage Service Registry** app. See Schemas for details.

To designate complex data type (schema defined in service registry) to variable or input/output parameter value of cloud production process and automation sequence, script task and service, choose `Structure` or `StructureArray` as type during creation. For details on adding parameter or variable values, see Add Value for Parameter or Variable.

8. If required, create process output parameters on the end element and define the parameter values.
9. Define the input and output parameter values of all services and subprocesses.

Note

For third-party services coming from service registry, only `String` enum of parameter or `String` item enum of array parameter can be consumed in Production Process Designer.

When the service responses multiple codes, only response code 200 response schema will be shown in the design side.

Making changes to a service or schema may affect its usage in the production process. A compatible change won't cause runtime errors while an incompatible one will do. For example, changing the name and description, adding optional parameters and response code, are compatible service changes, while changing the API path, adding mandatory parameters, and changing HTTP method, web server, or response code, are incompatible service changes.

You will see an *(error)* next to the service or subprocess if it has been deleted or disabled. You can remove it or replace it with another one. In other cases, you will see a *(warning)* if the service or subprocess has a compatible change. Choose **Synchronize** in the message strip or the *(synchronize)* in canvas to refresh it. If the change is incompatible, you will see an *(error)*. Choose *(synchronize)* to refresh it and reassign the values. If the change is about a general information change, such as URL, path, HTTP method, or protocol, you will also see an *(error)*. These changes are refreshed automatically when you modify the production process.

For schema (complex data type) used in services, script tasks, the input or output parameters of the production processes, or variables in production processes, its compatible changes, for example, can cover changing property description, or enum values, while adding mandatory properties, changing the property name and data type are incompatible changes. If the schema has a compatible change under the **Schemas** tab of the **Manage Service**

Registry app, you will see a *(warning)*. If the change is incompatible, you will see an *(error)*. Redo the value mapping to make sure that the problem is solved. Choose **Done** and **Complete All to Proceed**.

You can review all the above errors or warnings in an issue report in the production process header.

In a production process of cloud runtime, if a user-defined service has multiple content types defined for request body and response, you can choose **application/json** or **text/plain** in **INPUT/OUTPUT** to define the input/output parameter in JSON, or plain text. You also have the option to add variables or parameters to the input parameter area of the text/plain type service with the **Insert** button. Note that if text/plain is defined as content type in response in the **Manage Service Registry** app, only string data type is allowed in the Production Process Designer.

In the **Automatic Retry Settings**, define the time and interval for the service (RESTful and OData) to be retried if there is a failure during execution. From the dropdown list, select the error codes by which you expect the retry to be triggered. Add an interval range from 5 to 60 seconds. Note that a system delay within 10 seconds might add to the actual interval time in the runtime environment. Add a retry limit no more than 30 times. Execution performance will be affected if retry limit is over 10 times.

Besides the automatic retry settings you have configured, the system performs automatic retry by default in the following situations:

API Method	Service (RESTful and OData)	Registered Production Process / Read Indicators, Read Indicators Flexibly, Read Indicators Locally
GET	a. error codes: 500 or above b. all connection or timeout issues without error codes c. error messages: i. recvAddress (..) failed: Connection reset by peer ii. Failed to resolve	/
POST, PUT, PATCH, DELETE, or methods other than GET request	a. error messages: i. readAddress (..) failed: Connection reset by peer ii. Connection refused iii. Connection timed out iv. writeAddress (..) failed: Connection reset by peer v. recvAddress (..) failed: Connection reset by peer vi. Failed to resolve b. error code: 503 (service unavailable)	a. error codes: above 500 b. all connection or timeout issues without error codes c. error messages: i. recvAddress(..) failed: Connection reset by peer ii. Failed to resolve

10. You can directly pass input parameters of the main production process or output parameters of previous service/subprocess to the value of input parameters of late service/subprocess, or output parameters of the main

production process. The output parameter value of services in cloud processes can also be written back to process variables.

> **Note**
>
> Output parameters of subprocesses with "Execute Asynchronously" checked will not be passed to the value of parameters of late elements.
>
> To pass complex data type values, you can configure the mapping relations when adding the input parameter values of the late service.

11. You can publish deployed cloud processes to service registry by choosing **Edit Header** and switching on the **Publish to Service Registry** toggle. The process you publish will be displayed in service library for easy reuse in other apps.

    > **Note**
    >
    > You can configure **Publish to Service Registry** during the design process, but the option takes effect only after the production process design is deployed.
    >
    > Production process published to service registry is executed asynchronously by default if it is called by the main process in the Production Process Designer. Alternatively, you can define it in the parameter settings of the right panel.

12. The Production Process Designer provides you the option to use a production process as a subprocess in the other runtime in the same production process design. For automation engineer, you can grant production engineer permission to call automation sequence on Production Connector / SAP Plant Connectivity. For production engineer, you can grant automation engineer permission to call production process on cloud. To do so, make sure DMC cloud web server has already connected to the Production Connector / SAP Plant Connectivity web server and open the production process first. Then, choose **Edit Header** and switch on the **Visible to Cloud Runtime** or **Visible to Production Connector / Plant Connectivity Runtime** toggle. Save it and now you can see them displayed under the repository list of the left panel in the corresponding runtime.
13. When you edit or delete a production process that has been used in other places, the system will detect it automatically and you can check in the warning message box where this production process is used. You can continue with your update or cancel. Alternatively, you can also choose **Check Where-Used** in the production process detail page to check where the production process has been used. Refresh the list to see the latest changes if the production process gets updated.

    Besides the general information about the consumers provided in the where-used list, the usage status indicates if the production process is used in an up-to-date manner by its consumers. You should pay attention and take certain actions to those production processes with “outdated” usage status. Possible usage status:

    - **Normal**: The production process is being used normally by the consumer.
    - **Outdated (compatible)**: The production process is outdated in the consumer due to compatible changes. Recommend checking further.
    - **Outdated (incompatible)**: The production process is outdated in the consumer due to incompatible changes. You need to synchronize the production process to keep using it.

    Navigate directly to each consumer through its link and check the production process accordingly.

Add Value for Parameter or Variable

Parameter or variable value in the **Design Production Processes** and **Manage Automatic Triggers** app supports primitive, primitive array and complex data types.

There are six ways to add values:

1. **Direct input**: Input directly constant value, variable, or another parameter with the same data type, structure, or expression.

2. **Expression editor**: Choose Open expression editor. You can use the expression editor to input constant values and expressions. Note that in the cloud process, the input parameters of script task and output parameters of end element do not support expression.

3. **Dropdown**: Choose one of the variables or other parameters that are of the same primitive or primitive array data type as the current parameter. In other cases, choose one of the complex data type variables with the same structure as the current parameter.

4. **Select structure property**: Choose the value help icon and in the Select Structure Property window, choose from complex data type one property whose data type is compatible with the current parameter. In other cases, choose from primitive array data type one item whose data type is compatible with the current parameter. Then choose Save.

5. **Input value**: Choose the value help icon and in the Input Value window, choose one of the complex / primitive array data type variables or other parameters with the same structure / item as the current parameter. The system will map all sub-properties / items automatically. If your selection is partially matched to the current parameter, the matched sub-properties / items will be mapped automatically while a warning icon will be displayed in the Compatibility Check column for incompatible ones. Meanwhile, the parent property / primitive array data type you have originally chosen will be cleaned. The system supports two editing modes: form view and code view. Switch between modes by choosing Switch to Code or Switch to Form.

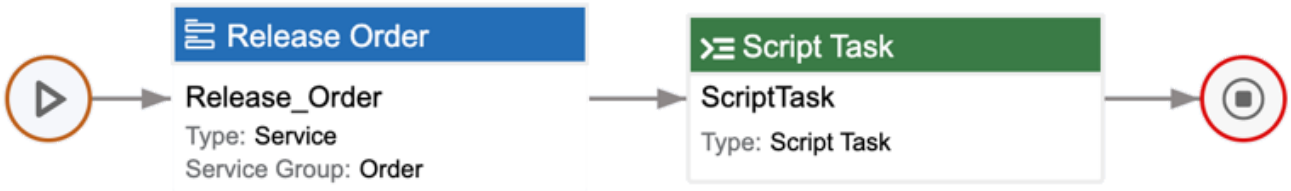
6. **Assign output** (for subscription or business rule): Choose the value help icon and in the Assign Output window, decompose the complex / primitive array data type and assign properties to variables.

ℹ Note

The Production Process Designer does not support writing the current parameter value back to a property in the complex data type variables, an item in the primitive array type variable, or partially matched complex / primitive array data type variables. However, in case you still want to do so, you can use the script task to decompose the structure of the target variable.

Example:

1. Add a script task element next to the current element whose output parameter (`Release_Order`) value you want to assign.



2. Create both input and output parameters (`startSfcRequest`) for the script task element with the very same data type as the target variable (`var_sfcStartRequest`).

3. For the script task input parameter, choose the value help icon.

4. In the Input Value window, choose output parameter `Release_Order` in the Value From column and check the mapping.

5. For the script task output parameter, choose the variable `var_sfcStartRequest`.

6. For the script, simply write a line to hand over the input parameter to the output parameter. For example, `$output.startSfcRequest = $input.startSfcRequest`.

		Primitive / Primitive Array Data Type	Complex Data Type

Runtime	Object	Primitive / Primitive Array Data Type		Complex Data Type	
		Get value from variable or parameter	Write back to variable	Get value from variable or parameter	Write back to variable

Runtime	Object	Get value from variable or parameter	Write back to variable	Get value from variable or parameter	Write back to variable
Production Process (cloud)	Web Service, Script Task, Production Process	For input parameters of service, script task, production process or automation sequence and output parameters of end element. primitive: [1], [2], [3], [4] primitive array: [1], [5]	For output parameters of service, script task or production process or automation sequence and input parameters of start element. primitive: [1], [3] primitive array: [1], [3] [Note]	For input parameters of service, script task, production process, or automation sequence (only for Production Connector) and output parameters of end element. [1], [5]	For output parameters of service, script task, production process, or automation sequence and input parameters of start element. [1], [3] [Note]
Automation Sequence (Production Connector on Windows / SAP Plant Connectivity on Windows)	Web Service, Production Process	For input parameters of service, production process, or automation sequence and output parameters of end element. primitive: [1], [2], [3], [4] primitive array: [1], [5]	For output parameters of service, production process, or automation sequence and input parameters of start element. primitive: [1], [3] primitive array: [1], [3]	For input parameters of service, production process, or automation sequence and output parameters of end element. (only for Production Connector) [1], [5]	For output parameters of service, production process, or automation sequence and input parameters of start element. [1], [3]
	Shop Floor Service, Functions	For input parameters of service and output parameters of end element. primitive: [1], [2], [3], [4] primitive array: [1], [5]	For output parameters of service and input parameters of start element. primitive: [1], [3] primitive array: [1], [3]	/	/
Subscription	Web Service, Production Process	For input parameters of service, production process, or automation sequence. primitive: [1], [2], [3], [4] primitive array: [1], [5]	For output parameters of service, production process, or automation sequence. primitive: [1], [3] primitive array: [1], [6]	For input parameters of service, production process, or automation sequence. (only for Production Connector) [1], [5]	For output parameters of service, production process, or automation sequence. [1], [6]

		Primitive / Primitive Array Data Type		Complex Data Type	
Runtime	**Object**	**Get value from variable or parameter**	**Write back to variable**	**Get value from variable or parameter**	**Write back to variable**
	Shop Floor Service	For input parameters of service. primitive: [1], [2], [3], [4] primitive array: [1], [5]	For output parameters of service. primitive: [1], [3] primitive array: [1], [6]	/	/
Timer	Web Service (including Registered Cloud Process)	For input parameters of service or production process. primitive: [1] primitive array: [1], [5]	/	For input parameters of service or production process. [1], [5]	/
Business Rule (Business Service)	Web Service (including Registered Cloud Process)	For input parameters of service or production process. primitive: [1], [2], [3], [4] primitive array: [1], [5]	For output parameters of service or production process. primitive: [1], [3] primitive array: [1], [6]	For input parameters of service or production process. [1], [5]	For output parameters of service or production process. [1], [6]
Business Rule (Event)	Registered Cloud Process	For input parameters of production process. primitive: [1], [2], [3], [4] primitive array: [1]	/	For input parameters of production process. [1], [5]	/

ℹ Note

In `String` data type parameters, enclose your input with double quotes for constant values or single quotes for variables.

When assigning an array to string data type input parameters, if the array index is a variable, you need to enclose it with single quotes. For example, `'input['a']'`. Note that for these cases the variable data type for the array index supports `Integer`, `Short`, and `Long` only.

Because there is a situation that "." is inside of the property name, the Production Process Designer supports hybrid use of "." or "[]" as complex data type property access operator to distinguish same-level variable reference from hierarchical reference. For example, `s.filter.material` refers to "material" under "filter". `s["filter.material"]` refers to "filter.material" under "s".

User-defined schema of array-of-array structure in service registry, `[[1,2,3],[4,5,6]]`, for example, is not supported in Production Process Designer.

The following special characters are supported in the string data type (including string property of structure and structure array) input/output parameter value when you call RESTful services, including DMC cloud service, read/write indicators, read/write indicators locally, read/write indicators flexibly, third-party services (RESTful), registered cloud processes, and automation sequences:

Special Character	Name	Note
#	number sign	
$	dollar sign	
/	slash	
.	period	
"	double quotes	Only supported in variable value. Using in constant value is not supported.
'	single quotes	Only supported in variable value. Using in constant value is not supported.
@	at sign	
`	backtick	
	space	

Related Information

Data Types and Data Type Conversion
Create and Define a Production Process
Create a Subscription
Create a Timer
Create a Business Rule

Import Production Processes from Other Production Process Design

Instead of creating production processes directly in the production process design, you can also import production processes from another production process design.

Prerequisites

You have either role of `Automation_Engineer` or `Production_Engineer`. Note that automation engineer can only import automation sequences from other production process designs.

You're assigned to this production process design.

Procedure

1. Open the **Design Production Processes** app.
2. Open or create a production process design.
3. Choose *(Add)*, then choose **Import Existing Process**.

4. In the **Select Production Processes** window, use the search and filter bar to filter the production process design with the relevant version, status and type.
5. To import production processes from another production process design, check the production processes you want to import and then choose **Import**.

> **Note**
>
> You can give the production process a new name when importing. Automation Engineer can only see the production processes of Production Connector / SAP Plant Connectivity runtime. Production Engineer can see production processes of cloud, Production Connector and SAP Plant Connectivity runtime.

Results

You can see production processes imported in the service library under the category of "Production Processes". Any change of the imported production process will not affect the original and vice versa.

Create and Use Snippets

You can reuse part of production process across production process designs by creating snippets.

A snippet is a small section of re-usable process components consisting of BPMN event elements and services. Snippets can contain consecutive or inconsecutive services, codes, parameters, variables. Snippets don't support subprocesses or start/end elements.

Create a Snippet

1. Open a production process.
2. Use Shift key to select more than one element of the process.
3. Choose *(save)* next to the selection area.
4. In the **Save Snippet** window, input a unique name for your snippet.
5. Choose **Save**.

Use a Snippet

1. Open a production process where you want to add a snippet.
2. Under **Services and Processes**, choose **Select Services**.

> **Note**
>
> You can see snippets created by yourself and other users under the **Snippets** group. The system only displays snippets for the current runtime. When you delete a snippet in the **Manage My Snippets** tab, it isn't deleted from production processes that contain the snippet.

3. Drag and drop a snippet onto the canvas.

Download Production Processes

You can download a production process into an image file (PNG) or a BPMN file (XML).

Prerequisites

You have the role of `Automation_Engineer` or `Production_Engineer`.

Context

To download a process, open the process in the editor and then choose the *(Download)* button.

Note

If a production process embeds one or more other production processes, the downloaded BPMN file or image does not contain the data of the embedded production processes. You can navigate into each embedded production process and download its own BPMN file or image.

Response Codes of Production Process Service

When calling a production process that is registered as a service in the **Manage Service Registry** app, you may receive different response codes depending on the execution result.

Response Code	Description
200	The synchronous call of the production process succeeded. The output is the output parameters defined for the production process, on the end event.
202	The asynchronous call of the production process succeeded. As an asynchronous call does not return the output of the production process, you will receive information of the process instance.
401	The request has not been applied because it lacks valid authentication credentials for the production process.
403	The server understood the request but refuses to authorize it due to insufficient permissions to the production process.
404	The server can't find the requested production process.
500	Internal error occurred when calling the production process. Possible errors: `errProcessDefinitionNotFound`: Process definition key and version cannot be found in tenant database. `errProcessDocParamsMissReqParam`: Required input parameters are missing. `errProcessDocParamsMissDataType`: Data type information is missing. `errProcessDocParamsUnMatch`: The data type of parameter value does not match the data type defined. For the above cases, contact your production engineer that designed the production process to find out a solution.
504	The synchronous call of the production process timed out.

You can find the list of all response codes related to each production process and the response schema in the **Manage Service Registry** app.

Deploy and Activate a Production Process Design

After completing a production process design, first assign it to a deployment group and then deploy the design from the deployment group to the runtime environment. This ensures that the deployment is in the control of fewer people and the disruption to the production environment can be minimized.

Prerequisites

- You have the role of `Automation_Engineer` or `Production_Engineer`.
- You're assigned to this production process design.
- The production process design is in the status of **Draft**.

Procedure

1. Open the **Design Production Processes** app.
2. Open a production process design.
3. Choose **Select Deployment Group** in the dropdown menu of **Quick Deploy**.

   > **Note**
   >
   > For a fast deployment, you can choose **Quick Deploy** to complete the following steps all at once.
   >
   > The system will automatically validate if all the conditions of relevant services and indicators are met for the production process design to be deployed. The validation report will show which services or indicators in which production process have issues. You can fix these issues with the information first and choose **Validate** to validate your design again.

4. In the **Add to Deployment Group** window, confirm the default deployment group automatically created and choose **Submit for Deployment**. The status of the production process design becomes **Awaiting Deployment**.

   Alternatively, instead of using the default deployment group, you can select other existing deployment groups by choosing **Select Another Group**. In the **Select Deployment Group** window, you can also choose *(add)* to create a new deployment group. If you create a new deployment group, you become its administrator automatically.

5. In the **Deployment Group** window, adjust the shop floor elements that you want to deploy.
6. Choose **Deploy and Activate**.

   > **Note**
   >
   > You can still go to the **Deploy Shop Floor Elements** app to deploy the production process design afterwards. See Deploy and Activate Shop Floor Elements for details.
   >
   > There are also cases where production process design was activated while some agents of relevant shop floor systems could not be started because remote shop floor systems are not available or servers are down under planned maintenance, for example. You can review the information in the activation exception report in the **Deploy Shop Floor Elements** app and try to restart the impacted agent instances later. To check details or to start the agent instances, choose the shop floor system and navigate to the shop floor system maintenance page.
   >
   > By default, automation engineer is allowed to submit and deploy production process design that has only Production Connector / SAP Plant Connectivity runtime production process. Otherwise, for production process design with cloud runtime production process or all three runtime production processes, only production engineer is allowed to deploy and submit.

Results

The status of the production process design becomes **Deployed**.

For production process with parameters of complex data type, you can view the details of each parameter by choosing the **Details** button even after its deployment.

Related Information

Deploy Shop Floor Elements
Deploy and Activate Shop Floor Elements

Debug Deployed Production Processes

You can debug a cloud production process after deployment in Production Process Designer.

Debugging helps you find out the problem by checking process execution step by step.

> **Note**
> Debugging supports cloud process only.

Prerequisites

You have the role of `Production_Engineer`.

Context

One process can be only debugged by one user at the same time.

You can start another debug session for the subprocess from design time.

For parallel execution, by default one branch will be selected automatically. Choose **Next Step** to execute the selected step and stop at the next step. You can switch to other branch by selecting the step in other branch.

In the production process debug session, if it takes a long time for a step to proceed, you can see that step highlighted.

Procedure

1. Open the **Design Production Processes** app.
2. Open a production process design.
3. Select a production process and choose **Debug**.
4. If required, enter input values in the debug production process window and choose **Debug**.
5. In graphic view, switch on the **Edit Breakpoint** toggle and click on the step that you want to add breakpoint. To delete all breakpoints at once, choose **Delete All Breakpoints**.
6. Choose **Next Breakpoint** to let the debugging process stop at the next breakpoint. The previous steps are executed.
7. Choose **Next Step** to let the debugging process stop at the next step. The previous step is executed.
8. Choose **Restart Debugging** to terminate the current debugging process and start another one.
9. Choose **Exit Debugging** to end the debug session and go back to the normal design time of Production Process Designer.

Note

The production process being debugged is displayed separately in the **Monitor Production Processes** app with the status "Running (debugging)". The number of debugging production process is counted under the "Running" column of the KPI carousel. Debugging does not affect the production process in normal execution.

Subprocess is executed automatically when you debug the main process. You can double-click the subprocess to check the detail of it. Alternatively, you can also stop at the subprocess step and choose **Debug Subprocess** if you want to debug the subprocess. This will navigate you to the subprocess and stop at the start event of the subprocess. You can then execute steps one by one or set breakpoints in the subprocess. Choosing **Next Step** on the end event will navigate you back to the main process and stop at the next step after this subprocess. If you choose **Next Breakpoint** but there are no more breakpoints in the subprocess, you will be navigated back to the main process and stop at the next breakpoint after this subprocess.

If you want to have an overview of the main process when debugging the subprocess, choose the back button of the navigation bar. Note that you cannot do any other operations on the main process when debugging subprocess. You can double-click the subprocess to continue the subprocess debugging. Choose **Restart Debugging** will terminate the main process and start another one. Choose **Exit Debugging** will terminate the subprocess as well as the main process and go back to the normal design time of Production Process Designer.

Production Process Debug Page

Graphic View

The graphic view provides a whole picture of the process workflow as you see in normal mode. You have also the option to zoom in, zoom out the graphic, and zoom back to its original size.

The breakpoint is the stop point you can set to control which step the debugging process is going to stop at. You can use the **Edit Breakpoint** toggle to add breakpoint on different elements.

Selected Step View

The step view shows the selected step's name, description, step ID, input and output parameter information.

Processing Step - Input and Previous Step - Output View

The parameter view shows the current processing step name, input parameter name, type, expression and value; the previous step name, output parameter name, type and value. You can edit the previous step output parameter value to update next steps' values.

Note

Executed step will not execute again even if its input is edited.

Processing Step - Log View

The log view shows real-time log information with debug log level.

Parameters and Variables View

The parameter and variable lists process variables, output parameters of previous steps, input parameters of the process instance. You can edit the value of each parameter.

Note

You can configure which variable or parameter to display by choosing **Select Variables**.

Related Information

Monitor and Track Production Processes
Configure Debug Mode Settings

Run a Production Process After Deployment

After a production process is deployed and activated, you can run it directly from the cloud and verify if the production process actually works in the runtime environment.

Context

When you create a cloud process, you can choose whether to run it asynchronously or synchronously. By default, cloud processes run synchronously which means that you can see outputs immediately. If you choose to run a cloud process asynchronously, you can only check for the execution status and, if successful, the output in the **Monitor Production Processes** app.

An automation sequence is always run synchronously.

Procedure

1. Open the **Design Production Processes** app.
2. Open a production process design.
3. Select a production process and choose **Run**.
4. If required, enter input values.

   Example
   - Enter `"red"` for a string value.
   - Enter `["red", "pink"]` for a string array value.
   - Enter `100` for an integer value.

5. If required, for a cloud process in the current Production Process Designer, you can choose whether or not to run it asynchronously or synchronously. You can also change the log level of the cloud process in the **Production Process Administration** app and see the log detail in the **Monitor Production Processes** app. For more information, see Log Level and Monitor and Track Production Processes.

   Note

   The **Default** log level is the log level you set in the **Production Process Administration** app. If no log level is set, the default option will use the **Error** log level.

6. Choose **Run**.

   Note

   For synchronously-run production processes, complex data type output parameter value is displayed as a link. You can click the link and choose **Copy** to copy the whole value.

   The production process is executed with booster mode on by default. For more information, see Booster Mode.

Manage Automatic Triggers

In **Manage Automatic Triggers** app you have the flexibility to embed production processes into business scenarios by using the following types of triggers.

- Subscription - see Manage Subscriptions for details.
- Timer - see Manage Timers for details.
- Business Rule - see Manage Business Rules for details.

The automatic trigger has the following statuses:

- **Draft**: The automatic trigger has just been created or edited before being submitted to deployment group.
- **Modified**: A new version of a deployed automatic trigger has been saved.
- **Awaiting Deployment**: The automatic trigger has been submitted to the deployment group but not deployed.
- **Deployed**: The automatic trigger has been sent to the runtime environment but not activated.
- **Current**: The deployed automatic trigger has been activated.
- **Paused**: The current automatic trigger has been stopped for a while.
- **Awaiting Un-deployment**: The deployed automatic trigger has been submitted to a new deployment group for deletion.
- **Archived**: The automatic trigger has been deleted from the runtime environment.
- **Failed**: Deployment of the automatic trigger failed. Either try to deploy it again, or fix the configuration issue.
- **Unknown**: There's inconsistent data between the design time on the cloud and the runtime environment. Copy the automatic trigger and deploy it again.

ℹ Note

The **Manage Automatic Triggers** app uses the time zone of your selected plant as the standard time. If no plant is selected, the time of your current browser will be displayed instead.

Manage Subscriptions

By creating a subscription to one or more indicators or an MQTT message, you can enable an action to be triggered when certain conditions are met. The subscription needs to be deployed and activated to take effect.

→ Tip

Applicability of the Terms Asset and Equipment in this Documentation

Asset or equipment refers to a digital representation of a physical asset, a machine, or a tool used in production.

Please refer to the applicability of the terms "asset" and "equipment" in the context of this documentation:

- The term asset is applicable for those tenants that are onboarded to the Asset Model.
- The term equipment is applicable for those tenants that are onboarded to the Asset Central Model.

There are two types of subscriptions: indicator subscription and message subscription.

In indicator subscriptions, action is triggered when the value of your subscribed indicator changes.

The subscribed indicators are provided by shop floor systems attached to asset/equipment and are connected to asset/equipment tags for values like temperature and pressure.

Before creating indicator subscriptions, you need to:

- Configure the asset/equipment and indicators in Production Connectivity Model corresponding to the tags in the shop floor that you want to monitor. For details, see Managing Assets (Asset Model), Templates and Equipment (Asset Central Model). See Asset Model for an overview of this model.
- Create a shop floor system and map the indicators to tags through this shop floor system. A corresponding source system and an agent instance are created in Production Connector / SAP Plant Connectivity. This Production Connector / SAP Plant Connectivity agent instance monitors the tags for tag value changes. When a subscription is created using the shop floor system, a notification object is created for the corresponding Production Connector / SAP Plant Connectivity agent instance. For more information, see Add Shop Floor System, Manage Asset Connectivity (Asset Model) and Manage Equipment Connectivity (Asset Central Model).

In message subscriptions, action is triggered when an MQTT client publishes a message to a message broker. For more information, see: SAP MQTT Broker)

A Production Connector can subscribe to messages published by an MQTT client to a topic of a message broker.

Before creating message subscriptions, you need to:

- Create the message broker and the topics under it in the **Manage Message Brokers** app. Connect a Production Connector to the message broker you created. If you want the Production Connector to subscribe to messages published by an MQTT client to the topic, add the relevant topic and check **Subscribe**. For details, see Production Connectors.
- Under the **Schemas** tab of the **Manage Service Registry** app, configure a schema that will be used for the message to subscribe on the message brokers. Note that this is needed only if your message payload is a complex data type. In case the message payload is just a single string or numbers, schema definition is not a prerequisite. For details, see Create a Schema.

Runtime Web Server	Web Server to Connect	Object Included	Note
Production Connector on Windows web server / SAP Plant Connectivity on Windows web server	(not required)	shop floor services (OPC UA methods)	[2]
		automation sequences	[3]
	DMC cloud web server	cloud processes	[11], [7], [4]
		DMC cloud services	[1], [5], [10], [4]
		registered cloud processes	[9], [10], [5], [4]
	user-defined web server (cloud)	third-party services (RESTful)	[6], [8], [4]
	user-defined web server (on-premise)	third-party services (RESTful)	

ⓘ Note

1. For details, see Create a Destination for SAP Digital Manufacturing.
2. These services are provided by external service providers that are created on the selected runtime Production Connector / SAP Plant Connectivity system. Besides, a client proxy must be configured for each service. For details,

see Configure Shop Floor Services (OPC UA Methods). You need to select the client proxy when using the service or production process.

3. The automation sequence must be run by the same Production Connector / SAP Plant Connectivity system as the subscription is.
4. A client proxy must be configured for each service or production process. For details, see Configure Client Proxy for Third-Party Service. You need to select the client proxy when using the service or production process.
5. Connect the runtime web server with the DMC cloud web server. Then deploy services you want to use from the connected web server. For details, see Manage Web Servers.
6. Any services that you manually register with SAP Digital Manufacturing in the **Manage Service Registry** app. Not all RESTful services are suppported. For details, see Registering Services in the SaaS Tenant of SAP Digital Manufacturing.
7. Check **Visible to Production Connector / Plant Connectivity Runtime** in Production Process Designer to make the cloud process you created visible to Production Connector / SAP Plant Connectivity runtime.
8. Connect the runtime web server with the web server you created to use the services. For details, see Manage Web Servers.
9. Cloud processes published from the Production Process Designer to the **Manage Service Registry** app.
10. To enable communication between Production Connector / SAP Plant Connectivity and SAP Digital Manufacturing systems, configure Production Connector / SAP Plant Connectivity as a system in SAP Digital Manufacturing and add the trust configurations. For details, see Authentication for Automation Sequence and Subscription.
11. Connect the runtime web server with the DMC cloud web server. For details, see Manage Web Servers.

In the **Subscriptions** tab of **Manage Automatic Triggers** app, you can do the following:

- Create, run, update, copy and delete subscriptions

  Note that if you want to edit a subscription that has already been deployed, you will create a new version of the subscription in the **Draft** status. The new version needs to be deployed and activated to replace the **Current** version.
- Submit subscriptions for deployment by sending them to deployment groups
- Pause and resume indicator subscriptions that have been deployed and activated from deployment groups

Create a Subscription

You can create a subscription to monitor the value change of one or more indicators or the publication of an MQTT message. When the predefined conditions are met, an action—a production process or a service—is triggered.

Prerequisites

- You have the role of `Automation_Engineer`.
- You have the role of `AC_ORG_EXPERT`.
- You can choose production processes either in **Draft**, **Deployed**, **Awaiting Deployment** or **Modified** status. When deploying subscription whose action is production process in **Draft** or **Modified** status, you will be asked to add it to the same deployment group first and then deploy together. For production process in **Awaiting Deployment** status, go to its deployment group and deploy it first, then go back to the deployment group of the subscription and deploy.

Create an Indicator Subscription

Procedure

1. Open the **Manage Automatic Triggers** app and choose the **Subscriptions** tab.
2. Choose **Create**.
3. Enter a unique name and, optionally, a description.

   Note that the name is **case-sensitive**.
4. In the context section, choose **Indicator** as the subscription type.
5. Choose **Add** to select the asset/equipment, shop floor system, and indicators. Select the asset/equipment and indicator modeled in the production connectivity model corresponding to the shop floor source tags that you want to monitor. Shop floor systems associated with the tags are available for selection. Select all the indicators, whose values need to be sent later to the service or trigger condition expression. You can select more than one indicator and define an alias for each indicator.

   You can use the indicator as a variable in expressions by referring to its alias. These expressions can be used to define the trigger condition or the input parameters of the selected action.

   You can also map output parameters of the selected action to the indicator.

   The subscription supports two types of indicators: 1) indicator mapped to a primitive data type tag; 2) indicator mapped to a component (primitive data type) of a structured data type tag. Note that only when the shop floor system is linked to Production Connector (not lower than version 1.2.0) can you select the second type of indicator. See Manage Equipment Connectivity for details.

   Once you have added the indicator and defined its alias, you can see all relevant information, such as the asset, structure name (or equipment, template, indicator group), and shop floor system in a popover by clicking the indicator name. The indicator data type, the data source tag ID, and its data type are also listed in the context section. You can see the path of the tag through the tag ID link if it is of structured data type (currently displayed as **Object**).

   ### ℹ Note

   For each subscription, the indicators must come from the same shop floor system.

   You can create subscription for below shop floor systems:

   - OPC UA Server
   - OPC DA Server
   - OPC HDA Server
   - Asset Framework Server
   - Proficy Historian Server

   ### ℹ Note

   IP21 Server is not supported.

   Depending on your role, the way of selecting indicators is different. If you have only the `Automation_Engineer` role, you can select indicators directly under a shop floor system. If you have the `Production_Engineer` role, you select indicators according to the asset, and structure name (or equipment, template, indicator group).
6. Define the trigger condition under which an action is triggered.

   For all condition types except for **On Change**, you need to define a Boolean expression for the trigger condition. See Data Types and Data Type Conversion and Supported Operators in Expressions, and Supported Functions in Expressions in Automation Sequences for more information. When an indicator mapped to a component (primitive data type) of a

structured data type tag is selected in this Boolean expression, or in the input/output parameter value of the action, the system only takes the value of the component mapping to that indicator. Other component values under the same tag are not affected.

Condition Type	Description
On Change	When the value of any monitored indicator changes, the defined action is triggered. **Note** In the **Check Changes for** field, choose from indicators you have added in the context section to subscribe to the value change of this indicator. For an indicator mapped to a component (primitive data type) of a structured data type tag, any value change of this tag will trigger the action. If the shop floor system is linked to Production Connector (lower than version 1.2.0) or Plant Connectivity, the value expression you used in the input parameter of the selected action will be automatically added to the **Check Changes for** field. Note that the removal of such value expression from the input parameter will not lead to its removal from this field, while you are not allowed to remove it from this field if it is still in use in the input parameter.
On Change to True	When the value of any monitored indicator changes and the expression is evaluated to TRUE, the action is triggered. The action is triggered only once.
On Change to False	When the value of any monitored indicator changes and the expression is evaluated to FALSE, the action is triggered. The action is triggered only once.
While True	When the value of any monitored indicator changes and the expression is evaluated to TRUE, the action is triggered. The action is triggered repeatedly until the expression is evaluated to FALSE.
While False	When the value of any monitored indicator changes and the expression is evaluated to FALSE, the action is triggered. The action is triggered repeatedly until the expression is evaluated to TRUE.

Additionally, you can set the retention time on subscription. Retention time is used to set the time period on how often to detect the tag values from shop floor layer through Production Connector / SAP Plant Connectivity.

Note

Each agent has a retention time. You can check, start or stop the agents and view the retention time in the shop floor system tab of the **Configure Production Connectivity** app. For details, see Agent Instances.

Select one of the existing retention times in the dropdown list of **Minimum Retention Time (Seconds)** or choose **Set Custom Data** to set a new one.

Note

The retention time is only available for Production Connector, SAP Plant Connectivity 15.4.2 or above.

You are recommended to start running the agent when activating the subscription. If the agent is not running, the subscription cannot work properly.

7. Define the action to be triggered. See Manage Subscriptions for action details.

a. Select **Production Process Call** or **Service Call** as the action type.

b. If you selected an automation sequence under the **Production Process Call** action type, the automation sequence should be on the same Production Connector / SAP Plant Connectivity system as the indicator shop floor system is on.

c. If you selected a cloud process under the **Production Process Call** action type, choose whether to run it asynchronously or synchronously.

- To run the cloud process synchronously when triggered by the subscription, uncheck the **Asynchronous Call** checkbox. If the process has any output parameters and is run successfully, the process will output values.
- To run the cloud process asynchronously when triggered by the subscription, keep the **Asynchronous Call** checkbox selected. The process will **not** output any values even though you have defined output parameters for it. If there is no need to write output parameters back to any indicator, this option is highly recommended as the performance is better than synchronous call.

d. If you selected **Service Call** as the action type, you can retry the action when a failure happens during execution. Turn on the toggle to define the time and interval in the **Automatic Retry Settings**.

e. Add a value for each input parameter of the process or service.

Example

To pass the name of the equipment (Robot A) to an input parameter, define this expression: `"Robot A"`.

Note

Use double quotes for `String` data type.

For details on adding parameter values, see Add Value for Parameter or Variable.

f. Optionally, map each output parameter to an indicator.

At runtime, the output value will be written into the indicator. However, this is not possible if it is a cloud process run asynchronously.

8. Save your entries.

Note

You will see an *(error)* next to the service or production process if it has been deleted in the service registry. You can replace it with another one. In other cases, you will see a *(warning)* if the service or production process has a compatible update. Choose *(synchronize)* to refresh it. If the update is incompatible, you will see an *(error)*. Choose *(synchronize)* to refresh it and reassign the values. If the update is about a general information change, such as URL, path, HTTP method, or protocol, you will also see an *(error)*. These updates are refreshed automatically when you modify the subscription.

You can review all the above errors or warnings in the message popover at the footer.

Create a Message Subscription

Procedure

1. Open the **Manage Automatic Triggers** app and choose the **Subscriptions** tab.
2. Choose **Create**.
3. Enter a unique name and, optionally, a description.

   Note that the name is **case-sensitive**.
4. In the context section, choose **Message** as the subscription type.

5. Select from the dropdown list the message broker.
6. Select from the dropdown list the Production Connector system connected to your selected message broker.
7. Select the topic to which your subscribed message will be published. The topics are maintained in your selected message broker on the connected Production Connector in the **Manage Message Brokers** app.
8. Select the message payload data type first and then select the message payload schema coming from the **Schemas** tab of the **Manage Service Registry** app.

   > **Note**
   >
   > You need to check and ensure that there is no mistake in configuration if another subscription of the same message broker and topic already exists.

   You can assign aliases to your intended primitive data type properties in the message payload structure. The alias can be used in the condition expression and input parameter value of action later.

9. Define the trigger condition under which an action is triggered.

   Choose the condition type and define a Boolean expression for the trigger condition. Use the aliases you defined in the previous step to build your expression. See Data Types and Data Type Conversion, Supported Operators in Expressions, and Supported Functions in Expressions in Automation Sequences for more information.

Condition Type	Description
While True	When an MQTT message based on your context selection is detected and the expression is evaluated to TRUE, the action is triggered. The action is triggered repeatedly until the expression is evaluated to FALSE.
While False	When an MQTT message based on your context selection is detected and the expression is evaluated to FALSE, the action is triggered. The action is triggered repeatedly until the expression is evaluated to TRUE.
Always	The action is always triggered when an MQTT message based on your context selection is detected.

10. Define the action to be triggered. See Manage Subscriptions for action details.
    a. Select **Production Process Call** or **Service Call** as the action type.
    b. If you selected an automation sequence under the **Production Process Call** action type, the automation sequence should be on the same Production Connector system as what you selected in Step 6.
    c. If you selected a cloud process under the **Production Process Call** action type, choose whether to run it asynchronously or synchronously.
       - To run the cloud process synchronously when triggered by the subscription, uncheck the **Asynchronous Call** checkbox. If the process has any output parameters and is run successfully, the process will output values.
       - To run the cloud process asynchronously when triggered by the subscription, keep the **Asynchronous Call** checkbox selected. The process will **not** output any values even though you have defined output parameters for it.
    d. If you selected **Service Call** as the action type, you can retry the action when a failure happens during execution. Turn on the toggle to define the time and interval in the **Automatic Retry Settings**.
    e. Add a value for each input parameter of the process or service.

       > **Example**
       >
       > To pass the name of the equipment (Robot A) to an input parameter, define this expression: `"Robot A"`.

       > **Note**

Use double quotes for `String` data type.

For details on adding parameter values, see Add Value for Parameter or Variable.

11. Save your entries.

> **Note**
>
> You will see an *(error)* next to the service, production process, or message payload schema if it has been deleted in the service registry. You can replace it with another one. In other cases, you will see a *(warning)* if the service, production process, or message payload schema has a compatible update. Choose *(synchronize)* to refresh it. If the update is incompatible, you will see an *(error)*. Choose *(synchronize)* to refresh it and reassign the values. If the update is about a general information change, such as URL, path, HTTP method, or protocol, you will also see an *(error)*. These updates are refreshed automatically when you modify the subscription.
>
> You can review all the above errors or warnings in the message popover at the footer.

Deploy and Activate a Subscription

To use a subscription, assign it to a deployment group and then deploy the subscription from the deployment group to the runtime environment.

Prerequisites

- You have the role of `Automation_Engineer`.
- The subscription is in the status of **Draft**.

Procedure

1. In the **Manage Automatic Triggers** app, select a subscription.
2. Choose **Select Deployment Group** in the dropdown menu of **Quick Deploy**.

> **Note**
>
> For a fast deployment, you can choose **Quick Deploy** to complete the following steps all at once.

3. In the **Add to Deployment Group** window, confirm the default deployment group automatically created and choose **Submit for Deployment**. The status of the deployment group becomes **Awaiting Deployment**.

   Alternatively, instead of using the default deployment group, you can select other existing deployment groups by choosing **Select Another Group**. In the **Select Deployment Group** window, you can also choose *(add)* to create a new deployment group. If you create a new deployment group, you become its administrator automatically.

4. In the **Deployment Group** window, adjust the shop floor elements that you want to deploy.
5. Choose **Deploy and Activate**.

> **Note**
>
> You can still go to the **Deploy Shop Floor Elements** app to deploy the deployment group afterwards. See Deploy and Activate Shop Floor Elements for details.
>
> Different from indicator subscription, message subscription will be activated automatically after deployment. You need to delete it if you want to deactivate it.

Results

The status of the subscription becomes **Current**.

Related Information

Deploy Shop Floor Elements
Deploy and Activate Shop Floor Elements

Pause and Resume Subscription

You have the option to pause a subscription and resume a paused subscription.

Prerequisites

You have the role of `Automation_Engineer`.

Context

If notifications due to a subscription need to be stopped for a while, you can pause the subscription. The paused subscription can be resumed when required.

> **Note**
> Message Subscription does not support pausing or resuming. Delete it if you want to pause.

Procedure

1. Choose **Manage Automatic Triggers** and select the subscription that you want to pause.

   Subscriptions in current status can be paused.
2. Choose **Pause** .
3. To resume a paused subscription, choose the subscription and select **Resume**.

   The subscription is resumed and set back to the current status.
4. Subscriptions can also be paused and resumed from the subscriptions list.

Copy a Subscription

You can create a new subscription by copying an existing one.

Prerequisites

You have the role of `Automation_Engineer`.

Context

You can create a new subscription by copying an existing one.

Procedure

1. Open the **Manage Automatic Triggers** app and choose the **Subscriptions** tab.
2. Open a subscription.
3. Choose **Copy** from the dropdown menu next to **Quick Deploy** / **Quick Delete**.
4. Enter a new name that is also unique.
5. Optionally, change the description.
6. Choose **Copy**.

Results

A new subscription is generated. You need to go to the detailed page to change the configuration.

Delete Subscription

When a subscription is in the **Draft**, **Delete**, **Modified**, **Archived** status, you can always delete it. After deploying the subscription from the cloud to the runtime environment, you must first delete the subscription from the runtime environment (un-deploy) and then from the cloud.

Prerequisites

You have the role of `Automation_Engineer`.

Delete a Deployed Subscription from the Runtime Environment

Procedure

1. Open the **Manage Automatic Triggers** app.
2. Open a subscription in the status of **Deployed**.
3. Choose **Select Deployment Group** in the dropdown menu of **Quick Delete**.

   Note

   For a fast deletion, you can choose **Quick Delete** to complete the following steps all at once (while still need to delete it from cloud afterwards).

4. In the **Add to Deployment Group** window, confirm the default deployment group automatically created and choose **Submit for Deployment**. The status of the subscription becomes **Awaiting Un-deployment**.

   Alternatively, instead of using the default deployment group, you can select other existing deployment groups by choosing **Select Another Group**. In the **Select Deployment Group** window, you can also choose *(add)* to create a new deployment group. If you create a new deployment group, you become its administrator automatically.

5. In the Deployment Group window, adjust the shop floor elements that you want to un-deploy.
6. Choose **Deploy and Activate**.

   Note

   You can still go to the **Deploy Shop Floor Elements** app to un-deploy the subscription afterwards. See Deploy and Activate Shop Floor Elements for details.

Results

- All runtime configurations are deleted.
- The status of the subscription becomes **Archived**.

Delete from the Cloud

Procedure

1. Open the **Manage Automatic Triggers** app.
2. Choose *(Filter)* and select **Archived**.
3. For a subscription in the status of **Archived**, choose *(Delete)*.
4. Confirm the deletion.

Delete in Batch

Procedure

1. Open the **Manage Automatic Triggers** app.
2. Select the subscriptions that you want to delete in batch.
3. Choose **Delete**.

> **Note**
> Subscriptions in the **Draft**, **Archived**, **Modified** status will be deleted directly. Subscriptions in the **Awaiting Un-deployment**, **Awaiting Deployment** or **Failed** status will be ignored. **Deployed** designs will be submitted to a deployment group for your further action.

Manage Timers

By creating a timer, you can enable an action to be triggered in a frequency or a specific time you predefined.

The timer needs to be deployed and activated to take effect.

Runtime Web Server	Web Server to Connect	Object Included	Note
DMC cloud web server	(not required)	DMC cloud services	[1]
		registered cloud processes	[2]
	user-defined web server (cloud)	third-party services (RESTful, OData)	[3], [4], [5]
	user-defined web server (on-premise)	third-party services (RESTful)	

> **Note**

1. For details, see Create a Destination for SAP Digital Manufacturing.

2. Cloud processes published from the Production Process Designer to the **Manage Service Registry** app.
3. Any services that you manually register with SAP Digital Manufacturing in the **Manage Service Registry** app. The content type of service request and response must be application/json. Not all RESTful and OData services are supported. For details, see Registering Services in the SaaS Tenant of SAP Digital Manufacturing.
4. Connect the runtime web server with the web server you created to use the services. For details, see Manage Web Servers.
5. For details, see Prerequisites for Using Third-Party Service in Cloud Process.

In the **Timers** tab of **Manage Automatic Triggers** app, you can do the following:

- Create, run, update, copy and delete timers

  Note

  If you want to edit a timer that has already been deployed, you will create a new version of the timer in the **Draft** status. The new version needs to be deployed and activated to replace the **Current** version.

  The operation of copying and deleting timer is similar to that of subscription. For details, see Copy a Subscription and Delete Subscription.

- Submit timers for deployment by sending them to deployment groups

  Note

  The operation is similar to that of subscription. For details, see Deploy and Activate a Subscription.

- Pause and resume timers that have been deployed and activated from deployment groups

  Note

  The operation is similar to that of subscription. For details, see Pause and Resume Subscription.

Create a Timer

You can create a timer to execute actions in a frequency or a specific time you predefined.

Prerequisites

- You have the role of `Production_Engineer`.

  Note

  You may see an error message asking for Asset Central permission. This is for using subscriptions. Ignore it for timers.

Procedure

1. Open the **Manage Automatic Triggers** app and choose the **Timers** tab.
2. Choose **Create**.
3. Enter a unique name and, optionally, a description.

   Note that the name is **case-sensitive**.

4. Define cron expression type timer to manipulate precisely the execution frequency of actions.

   You can input directly the cron expression of a string comprising six fields separated by white space that represents second, minute, hour, day of the month, month of the year and day of the week. Each field is defined separately and used in combination as a whole. Alternatively, you can use cron expression helper to generate one for you. The timers use UTC±00:00 as standard time in the settings.

   The cron expression helper provides options to set frequency for a set of times.

   a. Choose **Cron Expression Helper**.

   b. Designate settings for each field.

   > **Note**
   >
   > You cannot designate "second" in Cron Expression Helper. The field of second is automatically set as "0" by default.

   c. Choose **Generate**.

5. Define the action to be triggered.

   a. In action item, choose a service or process coming from the **Manage Service Registry** app.

   b. Choose whether you would like to call the action synchronously or asynchronously.

   > **Note**
   >
   > Production process will time out if the execution lasts over 20 seconds. Enable asynchronous call to avoid such issue.

   a. Add a value for each input parameter of the process or service.

   > **Note**
   >
   > Primitive, primitive array and complex data types are supported. Use double quotes for `String` data type.
   >
   > For details on adding parameter values, see Add Value for Parameter or Variable.

6. Save your entries.

> **Note**
>
> You will see an *(error)* next to the service or production process if it has been deleted in the service registry. You can replace it with another one. In other cases, you will see a *(warning)* if the service or production process has a compatible update. Choose *(synchronize)* to refresh it. If the update is incompatible, you will see an *(error)*. Choose *(synchronize)* to refresh it and reassign the values. If the update is about a general information change, such as URL, path, HTTP method, or protocol, you will also see an *(error)*. These updates are refreshed automatically when you modify the timer.
>
> You can review all the above errors or warnings in the message popover at the footer.

Manage Business Rules

By creating a business rule, you can enable an action to be triggered before or after a specific business service is executed, when there is an error during the business service execution, or after an event.

There are two types of business rule: business service and event. For business service, you can trigger action before, after its execution, or when there is an error during the execution. For event, you can only trigger the action after it. Besides the difference, business services and events provide various properties respectively. Check Business Service List and Event List for detailed information on their property descriptions.

A business service can be, for example, a goods receipt or a release order to SFC. In some cases, you may want to be informed or approve the workflow before such a business service is executed, while in other cases, perform an action after the business service execution.

The business rule needs to be deployed and activated to take effect. For event type rules, the triggered action can be production processes published from production process designer to the **Manage Service Registry** app. For business service type rules, the triggered action is as follows:

- DMC cloud services: [1]
- registered cloud processes: [2]
- user-defined web services (cloud): [3], [4], [5]

Note

1. For details, see Create a Destination for SAP Digital Manufacturing.
2. Cloud processes published from the Production Process Designer to the **Manage Service Registry** app.
3. Any services that you manually register with SAP Digital Manufacturing in the **Manage Service Registry** app. The content type of service request and response must be application/json. Not all RESTful and OData services are supported. For details, see Registering Services in the SaaS Tenant of SAP Digital Manufacturing.
4. Connect the runtime web server with the web server you created to use the services. For details, see Manage Web Servers.
5. For details, see Prerequisites for Using Third-Party Service in Cloud Process.

In the **Business Rules** tab of **Manage Automatic Triggers** app, you can do the following:

- Create, run, update, copy and delete business rules

  ### Note

  If you want to edit a business rule that has already been deployed, you will create a new version of the business rule in the **Draft** status. The new version needs to be deployed and activated to replace the **Current** version.

  The operation of copying and deleting business rule is similar to that of subscription. For details, see Copy a Subscription and Delete Subscription.

- Submit business rules for deployment by sending them to deployment groups

  ### Note

  The operation is similar to that of subscription. For details, see Deploy and Activate a Subscription.

  You can deploy 200 business rules at most to the same plant.

  Running multiple business rules with overlapping conditions of the same business service and trigger point can cause performance issues. You can activate the business rule you selected while the conflicting one will be deactivated accordingly.

- Pause and resume business rules that have been deployed and activated from deployment groups

  ### Note

  The operation is similar to that of subscription. For details, see Pause and Resume Subscription.

Create a Business Rule

You can create a business rule to execute actions before or after a specific business service is executed, when there is an error during the business service execution, or after an event.

Prerequisites

- You have the role of `Production_Engineer`.
- Create a destination for SAP Digital Manufacturing in the SAP BTP cockpit. For details, see Create a Destination for SAP Digital Manufacturing.

Procedure

1. Open the **Manage Automatic Triggers** app and choose the **Business Rules** tab.
2. Choose **Create**.
3. Enter a unique name and, optionally, a description.

   Note that the name is **case-sensitive**.
4. Choose either **Business Service** or **Event** from the two types of business rule provided in the general settings section.
   a. If you choose **Business Service** as the rule type:
      i. Choose a business service in the dropdown list. You can see a description of the business service next to it.
      ii. Choose the trigger point. For business service type rules, you can trigger the action at three time points:
      - **Before Service Execution**: trigger action before a business service is executed
      - **After Service Execution**: trigger action after a business service is executed
      - **Error During Service Execution**: trigger action when there is error during business service execution

   b. If you choose **Event** as the rule type:
      i. Choose an event type in the dropdown list. You can see a description of the event next to it.

ℹ Note

Refer to Business Service List and Event List for detailed information on business services and events available and their property descriptions. Do not use business services or events that are marked as **deprecated** to create business rules.

5. Define the business rule in the condition section.

ℹ Note

Plant is a mandatory parameter. All other parameters are optional. You can input directly or use the value help to add condition values.

The condition section supports primitive, primitive array and complex data types as condition property data types, for example:

Example

Condition Property Data Type	Name	Operator	Value
String	`plant`	EQUALS / STARTS WITH	“plant01”

Condition Property Data Type	Name	Operator	Value
StringArray	`sfcs`	EXISTS IN	“SPAS-01”, “SPAS-02”
Structure	`order.bom.material`	EQUALS / STARTS WITH	“MAT-01”
StructureArray	`sfcs[*].material`	EXISTS IN / EQUALS / STARTS WITH	“CAR-CHASSIS”

For array-type condition properties (business service rule type only), the operator **EXISTS IN** allows you to enter multiple values so that any value falling into the ones you have defined will trigger the business rule.

6. Define the action to be triggered.

   a. In action item, choose a service (only for business service as the rule type) or production process from the value help.

   b. Choose whether you would like to call the action synchronously or asynchronously.

   > **Note**
   >
   > Production process or service will time out if the execution lasts over 10 seconds. Enable asynchronous call to avoid such issue.
   >
   > For synchronous call, choose whether or not to ignore the action execution error and continue with the business service. If you choose **Ignore Execution Error**, errors thrown from action will not stop the business service execution. You can check the errors of production process later in the **Monitor Production Processes** app.

   c. Conduct the following operation according to different sitations:

		Parameter Type of the Action	
	Synchronous/Asynchronous Call	**Input Parameter**	**Output Parameter**
Before Service Execution	**Synchronous Call**	Input directly a constant value. Alternatively, select a business service request parameter as a variable in the expression for input parameter of the action.	Select an output parameter of the action to write back to a business service request parameter.
	Asynchronous Call		/
Error During Service Execution	**Synchronous Call**	/	/
	Asynchronous Call	Input directly a constant value. Alternatively, select a business service request parameter as a variable in the expression for input parameter of the action.	
After Service Execution	**Synchronous Call**	Input directly a constant value. Alternatively, select a business service response parameter as a variable in the expression for input parameter of the action.	Select an output parameter of the action to write back to a business service response parameter.
	Asynchronous Call		/
Event	**Synchronous Call**	/	/

		Parameter Type of the Action	
	Synchronous/Asynchronous Call	**Input Parameter**	**Output Parameter**
	Asynchronous Call	Input directly a constant value. Alternatively, select an event property as a variable in the expression for input parameters of the action.	

Primitive, primitive array and complex data types are supported. Use double quotes for `String` data type.

For details on adding parameter values, see Add Value for Parameter or Variable.

7. Save your entries.

> **Note**
>
> You will see an *(error)* next to the service or production process if it has been deleted in the service registry. You can replace it with another one. In other cases, you will see a *(warning)* if the service or production process has a compatible update. Choose *(synchronize)* to refresh it. If the update is incompatible, you will see an *(error)*. Choose *(synchronize)* to refresh it and reassign the values. If the update is about a general information change, such as URL, path, HTTP method, or protocol, you will also see an *(error)*. These updates are refreshed automatically when you modify the business rule.
>
> You can review all the above errors or warnings in the message popover at the footer.

Event List

Get an overview of events available in the business rule page of the **Manage Automatic Triggers** app.

BOM Created

A business event is sent when a bill of material is created.

Event Properties

Name	Data Type	Description	Note
`plant`	String	The plant where the BOM is created	Condition property
`billOfMaterial`	String	The name of the BOM	Condition property
`ref`	String	The unique identifier of the BOM	
`version`	String	The version of the BOM	
`currentVersion`	Boolean	The current version of the BOM	Condition property
`description`	String	The description of the BOM	
`bomType`	String	The type of BOM	Condition property

Name	Data Type	Description	Note
status	String	The status of the BOM. It could be any of the following statuses: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOL • HOLD_SPC_WARN	Condition property
released	Boolean	The property indicates if any SFCs have been released for the BOM.	
baseQuantity	Double	The base quantity of the BOM	
baseUom	String	The base unit measure of the BOM	
validFrom	Double	This property specifies the starting date of the validity period of an object.	
createdDateTime	Double	The date and time at which the BOM was crated	
customData	StructureArray		
customData.[i]	Structure		
customData.[i].key	String		
customData.[i].value	String		
components	StructureArray		
components.[i]	Structure		
components.[i].ref	String	The unique identifier of the BOM component	
components.[i].sequence	Double	The order component is displayed relative to other BOM components.	
components.[i].alternativeItemGroup	String	The alternative item group to the BOM component	
components.[i].quantity	Double	The number of pieces in the BOM	
components.[i].toleranceUnder	Double	The upper bound tolerance value	

Name	Data Type	Description	Note
components.[i].toleranceOver	Double	The lower bound tolerance value	
components.[i].componentScrap	Double	Percentage of scrap of the BOM component that occurs during production	
components.[i].totalQuantity	Double	The total quantity of the BOM component	
components.[i].alternatesEnabled	Boolean	If true, the BOM component alternates will be used for this component instead of material alternates.	
components.[i].assemblyQuantityAsRequired	Boolean	If true, the quantity of this BOM component isn't tracked.	
components.[i].storageLocation	String	The name of the storage location	
components.[i].componentType	String	The BOM component type	
components.[i].backflushEnabled	Boolean	If true, the material consumption is automatically posted.	
components.[i].unitOfMeasure	String	The unit of measure of the BOM	
components.[i].alternates	StructureArray		
components.[i].alternates.[i]	Structure		
components[i].alternates.[i].enabled	Boolean	If true, this component is a valid substitute for the BOM component.	
components[i].alternates.[i].required	Boolean	If true, this component must be substituted for the BOM component.	
components.[i].alternates.[i].validFrom	Double	The starting date in which the alternate starts being a valid substitute for the BOM component	
components.[i].alternates.[i].validTo	Double	The ending date in which this alternate stops being a valid substitute for the BOM component	
components.[i].alternates.[i].priority	Double	The priority assigned to the alternate component	
components.[i].alternates.[i].material	Structure		
components.[i].alternates.[i].material.ref	String	The unique identifier of the material	
components.[i].alternates.[i].material.plant	String	The plant of the material	Condition property

Name	Data Type	Description	Note
components.[i].alternates.[i].material.material	String	The name of the material	Condition property
components.[i].alternates.[i].material.version	String	The version of the material	
components.[i].material	Structure		
components.[i].material.ref	String	The unique identifier of the material	
components.[i].material.plant	String	The plant of the material	Condition property
components.[i].material.material	String	The name of the material	Condition property
components.[i].material.version	String	The version of the material	
components.[i].operation	Structure		
components.[i].operation.ref	String	The unique identifier of the operation	
components.[i].operation.operation	String	The name of the operation	Condition property
components.[i].operation.quantity	Double	The number of pieces to assemble	
components.[i].operation.description	String	The description of an operation at which the component is assembled.	
components.[i].BomRefDes	Structure	The reference designator where a BOM component should be assembled.	
components.[i].BomRefDes.refDes	String	The reference designator where a BOM component should be assembled.	Condition property
components.[i].BomRefDes.quantity	Double	The number of pieces to assemble at this reference designator.	
components.[i].BomRefDes.sequence	Double	The reference designator sequence number.	
components.[i].BomRefDes.erpRefDesSequence	Double	The reference designator sequence number in ERP.	
components.[i].dataType	Structure		
components.[i].dataType.ref	String	The unique identifier of the data type	
components.[i].dataType.plant	String	The plant of the data type	Condition property
components.[i].dataType.dataType	String	The name of the data type	

Name	Data Type	Description	Note
components.[i].dataType.category	String	The category of the data type. It could be any of these categories: • ASSEMBLY • NC • PACKING_CONTAINER • PACKING_SFC • RMA_SFC • RMA_SHOP_ORDER • SFC	
components.[i].customData	StructureArray		
components.[i].customData.[i]	Structure		
components.[i].customData.[i].key	String		
components.[i].customData.[i].value			
dataType	Structure		
dataType.ref	String	The unique identifier of the data type	
dataType.plant	String	The plant of the data type	Condition property
dataType.dataType	String	The name of the data type.	Condition property
dataType.category	String	The category of the data type. It could be any of these categories: • ASSEMBLY • NC • PACKING_CONTAINER • PACKING_SFC • RMA_SFC • RMA_SHOP_ORDER • SFC	

Related Information

Creating BOMs

BOM Updated

A business event is sent when a BOM is updated.

Event Properties

Name	Data Type	Description	Note
ref	String	Unique identifier of the BOM.	
plant	String	The plant for which the BOM is defined.	Condition property
billOfMaterial	String	Name of the BOM.	Condition property
version	String	Version of the BOM.	
currentVersion	Boolean	If true, then this BOM is current.	Condition property
description	String	The description of the BOM.	
bomType	String	The type of the BOM. • MASTER • SHOP_ORDER • SFC	Condition property
status	String	The status of the BOM. • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOL • HOLD_SPC_WARN	Condition property
released	Boolean	Indicates if any SFCs have been released on this BOM.	
baseQuantity	number	Base quantity of the BOM.	
baseUom	String	Base unit of measure of the BOM.	
validFrom	number	Specifies the start date for the validity period of an object.	
createdDateTime	number	The time when the BOM was created.	
dataType			
BomComponent	Structure		
BomComponent[i]ref	String	Unique identifier of the BOM Component.	

Name	Data Type	Description	Note
BomComponent[i]sequence	number	The order the component is displayed relative to other BOM components.	
BomComponent[i]erpSequence	number	The order the component as displayed in ERP.	
BomComponent[i]alternativeItemGroup	String	BOM component alternative item group.	
BomComponent[i]quantity	number	The number of pieces in the BOM.	
BomComponent[i]toleranceUnder	number	The upper bound tolerance value.	
BomComponent[i]toleranceOver	number	The lower bound tolerance value.	
BomComponent[i]componentScrap	number	Percentage of scrap of the BOM component that occurs during production.	
BomComponent[i]totalQuantity	number	Total quantity of the BOM component.	
BomComponent[i]alternatesEnabled	Boolean	If true, then the BOM Component alternates will be used for this component instead of Material alternates.	
BomComponent[i]assemblyQuantityAsRequired	Boolean	If true, the quantity of this BOM component is not tracked.	
BomComponent[i]storageLocation	String	Name of the storage location.	
BomComponent[i]componentType	String	BOM component type. • NORMAL • TEST • PHANTOM • CO_PRODUCT • BY_PRODUCT	
BomComponent[i]backflushEnabled	Boolean	If true, the material consumption will be automatically posted.	
BomComponent[i]unitOfMeasure	String	BOM component unit of measure.	
BomComponent[i]refDes			
BomComponent.[i].CustomValue	Structure		
BomComponent.[i].CustomValue[i]key	String		
BomComponent.[i].CustomValue[i]value	String		

Name	Data Type	Description	Note
BomComponent.[i].AlternateComponent	Structure		
BomComponent.[i].AlternateComponent[i]enabled	Boolean	If true, then this component is a valid substitute for the BOM component.	
BomComponent.[i].AlternateComponent[i]required	Boolean	If true, then this component must be substituted for the BOM component.	
BomComponent.[i].AlternateComponent[i]validFrom	number	The starting date that this alternate is a valid substitute for the BOM component.	
BomComponent.[i].AlternateComponent[i]validTo	number	The ending date that this alternate is a valid substitute for the BOM component.	
BomComponent.[i].AlternateComponent[i]priority	number	The priority of the Alternate component.	
Material	Structure		
Material.ref	String	Unique identifier of the Material.	
Material.plant	String	The plant of the Material.	Condition property
Material.material	String	Name of the Material.	Condition property
Material.version	String	Version of the Material.	
Operation	Structure		
Operation.ref	String	Unique identifier of the Operation.	
Operation.operation	String	Name of the Operation.	Condition property
Operation.quantity	number	The number of pieces to assemble.	
Operation.description	String	The description of an operation at which the component is assembled.	
BomRefDes	Structure		
BomRefDes.refDes	String	The reference designator where a BOM component should be assembled.	Condition property
BomRefDes.quantity	number	The number of pieces to assemble at this reference designator.	
BomRefDes.sequence	number	The reference designator sequence number.	
BomRefDes.erpRefDesSequence	number	The reference designator sequence number in ERP.	
DataType	Structure		

Name	Data Type	Description	Note
`DataType.ref`	String	Unique identifier of the Data Type.	
`DataType.plant`	String	The plant of the Data Type.	Condition property
`DataType.dataType`	String	Name of the Data Type.	Condition property
`DataType.category`	String	Category of the Data Type. • ASSEMBLY • NC • PACKING_CONTAINER • PACKING_SFC • RMA_SFC • RMA_SHOP_ORDER • SFC	

Related Information

Creating BOMs

BOM Deleted

A business event is sent when a bill of material is deleted.

Event Properties

Name	Data Type	Description	Note
`ref`	String	The unique identifier of the BOM	
`plant`	String	The plant where the BOM is deleted	Condition property
`billOfMaterial`	String	The name of the BOM	Condition property
`version`	String	The version of the BOM	

Batch Status Changed

A business event is sent when a batch is created or a batch status is changed.

Event Properties

Name	Data Type	Description	Note

Name	Data Type	Description	Note
`trigger`	String	The trigger of the batch status change. It could be any of the following triggers: • BATCH_CREATION_BY_UI Batch is created manually in the Order POD UI. • BATCH_CREATION_BY_API Batch is created via the batch creation public API. • BATCH_CREATION_CALLBACK Batch is created or batch status is updated with EWM stock change. • ORDER_DOWNLOAD Batch is created or batch status is updated with order dowload. • INVENTORY_DOWNLOAD Batch is created or batch status is updated with inventory dowload. • MANAGE_FLOOR_STOCK Batch is created or batch status is updated with creation of the local stock that has a batch number in the **Manage Floor Stocks** app.	
`plant`	String	The plant where the batch status is changed	Condition property
`batch`	String	The relevant batch number to the business event	Condition property
`material`	String	The material to which the batch number is assigned	Condition property
`originalStatus`	String	The original batch status. It could be any of the following statuses: • CREATED • VALIDATED • INVALIDATED	Condition property
`currentStatus`	String	The current batch status. It could be any of the following statuses: • CREATED • VALIDATED • INVALIDATED	Condition property

Name	Data Type	Description	Note
source	String	The source of the batch. It indicates where the batch is originally created. It could be either ERP or LOCAL. LOCAL indicates that the batch is created in SAP Digital Manufacturing.	
order	String	The relevant order for the batch status change. It can be optionally filled.	

Extended Warehouse Management Staging Confirmed

A business event is created when Extended Warehouse Management staging is confirmed.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant where the staging is confirmed	Condition property
warehouseNumber	String	Warehouse number	Condition property
productionSupplyArea	String	Production supply area	Condition property
shopOrder	String	Shop order	
material	String	Material	
handlingUnitNumber	String	Handling unit	
syncedInventories	String	Synchronized inventories	

Related Information

Goods Receipt Completed

A business event is created when a goods receipt is completed.

Event Properties

Name	Data Type	Description	Note
source	String	The source of the goods receipt.	
plant	String	The plant where the goods receipt is completed.	Condition property
order	String	The order relevant to the goods receipt.	Condition property
material	String	Material	Condition property
packingMaterial	String	Packing Material	
orderItem	String	Order item	
orderOperation	String	Order operation	

Name	Data Type	Description	Note
storageLocation	String	Storage location	
batch	String	Batch	
goodsMovementCode	String	Goods movement code	
goodsMovementType	String	Goods movement type	
goodsMovementRefDocType	String	Goods movement reference document type	
quantityInEntryUnit	Double	Quantity in entry unit	
entryUnit	String	Entry unit	
manufacturingDateTime	String	The date and time at which the goods receipt is manufactured.	
ewmProgramId	String	The ERP logical system used by EWM for ERP integration.	
ewmDestination	String	The EWM destination	
warehouse	String	Warehouse	
handlingUnit	String	Handling unit	

Goods Receipt Requested

A business event is sent when a goods receipt is created.

Event Property

Name	Data Type	Description	Note
plant	String	The plant where the goods receipt is performed.	Condition property
order	String	The order for which the goods receipt is performed.	Condition property
sfc	String	The Shop Floor Control number relevant to the goods receipt.	Condition property
material	String	The material name of the product posted in the goods receipt. The product can be finished goods, co-products or by-products.	Condition property
materialVersion	String	The material version of the product posted in the goods receipt. The product can be finished goods, co-products or by-products.	
warehouse	String	The warehouse number of the stock created upon goods receipt posting.	

Name	Data Type	Description	Note
packingMaterial	String	The packaging material of the product posted in the goods receipt. It's applicable only to goods receipts to Extended Warehouse Management.	
handlingUnit	String	The handling unit number of the product. It's applicable only to goods receipts to Extended Warehouse Management.	Condition Property
operation	String	The final operation of the order where the Shop Floor Control is completed.	
storageLocation	String	The storage location of the stock created upon goods receipt posting.	Condition property
quantity	Double	Indicates how many products are posted in the goods receipt.	
internalUnitOfMeasure	String	SAP internal unit of measure	
batchNumber	String	The batch number of the product	
serialNumbers	StringArray	A list of serial numbers for the products	
comments	String	The comments you want to add to the goods receipt posting.	
postingDateTime	DateTime	The posting date time of the goods receipt displayed in the time zone configured in SAP Digital Manufacturing Cloud plant. Format: YYYY-MM-DD. Example: 2021-01-31	
manufacturingDateTime	DateTime	The production date time of the product in UTC. Format : YYYY-MM-DD HH:MM:SS. Example: 2021-05-19T16:59:24.000Z	
additionalParameterNames	StringArray	The additional custom parameter name. You can add customized parameters to the goods receipt posting.	
additionalParameterValues	StringArray	The additional custom parameter value. You can add customized parameters to the goods receipt posting.	

Inspection Results Collection

The business event is triggered when you choose **Auto Collection** on the **Quality Inspection** plugin. It collects the inspection single results if the **Results Origin** of the characteristic is **Auto Collection**.

Event Properties

Name	Data Type	Description	Note
`plant`	String	The plant where the inspection results are collected.	Condition property
`sfc`	String	The shop floor control relevant to the inspection results.	Condition property
`inspectionLot`	String	Inspection lot	Condition property
`inspectionOperation`	String	Inspection operation	Condition property
`inspectionCharacteristic`	String	Inspection characteristic	Condition property
`shopOrder`	String	Shop order	
`inspectionPoint`	String	Inspection point	
`inspectionSpecificationText`	String	Inspection Specification Text	
`characteristicAttributeCodes`	String	Characteristic attribute code set for qualitative characteristic	

Logistics Order Outbound

A business event is created when a logistics order is changed.

Event Properties

Name	Data Type	Description	Note
`plant`	String	The plant where the logistics order is changed.	Condition property
`orderNumber`	String	The number of the logistics order.	Condition property
`transportSystem`	String	The transport system of the logistics order.	Condition property
`orderType`	String	The type of logistics order.	
`priority`	Double	The priority assigned to the logistics order.	
`status`	String	The status of the logistics order.	Condition property
`referenceType`	String	The reference type.	Condition property
`referenceNumber`	String	The unique reference number of the logistics order.	Condition property
`packingMaterial`	String	packing used materials.	
`carrierNumber`	String	The carrier number.	

Name	Data Type	Description	Note
dueDateTime	DateTime	The due time of the logistics order.	
plannedStartDateTime	DateTime	The planned start time of the logistics order.	
plannedCompleteDateTime	DateTime	The planned completion time of the logistics order.	
sourceLocation	String	The source location of the logistics order.	Condition property
destinationLocation	String	The destination location of the logistics order.	Condition property
transactionId	String	The transaction ID of the logistics order.	

Material Created

A business event is sent when material is created.

Event Properties

Name	Data Type	Description	Note
ref	String	The unique identifier of the material	
plant	String	The plant where the material is defined.	Condition property
material	String	The name of the material	Condition property
version	String	The version of the material	
descriptions	StructureArray		
descriptions.[i]	Structure		
descriptions.[i].locale	String		
descriptions.[i].description	String		
currentVersion	Boolean	If this property is true, the material is current.	Condition property

Name	Data Type	Description	Note
`status`	String	The status of the material.It could be any of the following statuses: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOL • HOLD_SPC_WARN	
`materialType`	String	The material type. It could be any of the following types: • CONFIGURABLE • FINISHED • GENERAL • NONSTOCK • NONVALUATED • OPERATING_SUPPLIES • PACKAGING • RETURNABLE_PACKAGING • SEMIFINISHED_PRODUCT • SERVICE • SERVICES • SOFTWARE_NONVALUATED • SPARE_PARTS • TRADING_GOODS • RAW • CUSTOM • PIPELINE • PRT	

Name	Data Type	Description	Note
procurementType	String	The procurement type of the material. It could be any of the following types: • MANUFACTURED • PURCHASED • MANUFACTURED_PURCHASED	
baseUOM	String	The value of the unit of measure	
lostSize	Double	The lot size of the material	
materialGroups	Structure	Assigned material groups	
isAutocompleteAndConfirmed	Boolean	If enabled, an operation activity is set to autocomplete.	
quantityRestriction	String	Controls the processing of fractional quantities.	
customData	StructureArray		
customData.[i]	Structure		
customData.[i].key	String		
customData.[i].value	String		
createdDateTime	Double	The date and time at which the material is created	

Material Updated

A business event is sent when the material is updated.

Event Properties

Name	Data Type	Description	Note
ref	String	The unique identifier of the material	
plant	String	The plant where the material is updated	Condition property
material	String	The name od the material	Condition property
version	String	The version of the material	
descriptions	StructureArray		
descriptions.[i]	Structure		
descriptions.[i].locale	String		
descriptions.[i].description	String		
currentVersion	Boolean	If true, the material is current.	Condition property

Name	Data Type	Description	Note
`status`	String	The status of the mateiral. It could be any of the following statuses: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOL • HOLD_SPC_WARN	
`materialType`	String	The material type. It could be any of the following types: • CONFIGURABLE • FINISHED • GENERAL • NONSTOCK • NONVALUATED • OPERATING_SUPPLIES • PACKAGING • RETURNABLE_PACKAGING • SEMIFINISHED_PRODUCT • SERVICE • SERVICES • SOFTWARE_NONVALUATED • SPARE_PARTS • TRADING_GOODS • RAW • CUSTOM • PIPELINE • PRT	

Name	Data Type	Description	Note
procurementType	String	The procurement type of the material. It could be any of the following types: • MANUFACTURED • PURCHASED • MANUFACTURED_PURCHASED	
baseUOM	String	The value of the unit of measure	
lotSize	Double	The lot size of the material	
isAutocompleteAndConfirmed	Boolean	If enabled, an operation activity is set to autocomplete.	
quantityRestriction	String	Controls the processing of fractional quantities.	
materialGroups	Structure	Assigned material groups	
customData	StructureArray		
customData.[i]	Structure		
customData.[i].key	String		
customData.[i].value	String		
modifiedDateTime	Double	The time at which the material is updated	

Material Deleted

A business event is sent when a material is deleted.

Event Properties

Name	Data Type	Description	Note
ref	String	The unique identifier of the material	
plant	String	The plant for which the material is defined	Condition property
material	String	The name of the material	Condition property
version	String	The version of the material	

Nonconformance Incident Logged

A business event is sent when a nonconformance incident is logged.

Event Properties

Name	Data Type	Description	Note

Name	Data Type	Description	Note
plant	String	The plant where the nonconformance incident is logged	Condition property
code	String	The code ID of the nonconformance	
sfc	String	The SFC associated with a logged nonconformance	
quantity	Double	The quantity of SFC associated with a logged nonconformance	
comments	String	The comments associated with a logged nonconformance	
component	Structure		
component.material	String	The name of the material	Condition property
component.plant	string	The plant where the material is defined	
component.version	String	The version of the material	Condition property
componentSfc	Structure		
componentSfc.material	String	The name of the material	Condition property
componentSfc.plant	string	The plant where the material is defined	
componentSfc.version	String	The version of the material	Condition property
category	String	The category of the logged nonconformance	
state	String	The state of the logged nonconformance. It could be any of the following statuses: • OPEN • CLOSED • CANCELLED • CLOSE_PENDING	
modifiedOn	Date-time	The date and time at which the logged nonconformance is modified in UTC. Format (ISO-8601): yyyy-MM-dd'T'HH:mm:ss'Z', for example, "2022-03-10T16:59:24.000Z".	

Name	Data Type	Description	Note
createdOn	Date-time	The date and time at which the logged nonconformance is created in UTC. Format (ISO-8601): yyyy-MM-dd'T'HH:mm:ss'Z', for example, "2022-03-10T16:59:24.000Z".	
routingStepId	String	The routing step ID where a nonconformance is logged	
material	Structure		
material.material	String	The name of the material	Condition property
material.plant	String	The plant where the material is defined	
material.version	String	The version of the material	Condition property
resource	String	The name of the resource	
workCenter	String	The name of the work center	
parentNonConformanceIncident	Structure		
plant	String	The plant	
sfc	String	The SFC associated with a parent logged nonconformance.	
userId	String	The ID of the user who logged a parent nonconformance.	
code	String	The parent nonconformance code ID.	
reportedDateTime	Date-time	The incident date and time for a logged parent nonconformance in UTC. Format (ISO-8601): yyyy-MM-dd'T'HH:mm:ss:SSS'Z'	
sequence	Integer	The sequence of the logged parent nonconformance.	
operationActivity	Structure		
operationActivity.operationActivity	String	The name of the operation activity	Condition property
operationActivity.version	String	The version of the operation activity	Condition property
incidentNumber	Structure		
incidentNumber.incidentNumber	String	The incident number	

Name	Data Type	Description	Note
`incidentNumber.modifiedOn`	Date-time	The time and date on which the logged nonconformance is modified in UTC. Format (ISO-8601): yyyy-MM-dd'T'HH:mm:ss'Z', for example, "2022-03-10T16:59:24.000Z".	
`incidentNumber.createdOn`	Date-time	The time and date on which the logged nonconformance is created in UTC. Format (ISO-8601): yyyy-MM-dd'T'HH:mm:ss'Z', for example, "2022-03-10T16:59:24.000Z".	

Related Information

Logging NC Codes

Operation Activity Created

A business event is sent when an operation activity is created.

Event Properties

Name	Data Type	Description	Note
`ref`	String	The unique identifier of the operation activity	
`plant`	String	The plant where the operation activity is created	Condition property
`operation`	String	The name of the operation activity	Condition property
`version`	String	The version of the operation activity	
`currentVersion`	Boolean	If true, the operation activity is current.	Condition property
`description`	String	The description of the operation activity	
`type`	String	The type of operation activity. It could be any of the following types: • NORMAL_OPERATION • SPECIAL_OPERATION • TEST_OPERATION	Condition property

Name	Data Type	Description	Note
status	String	The status of the operation activity. It could be any of the following statuses: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOL • HOLD_SPC_WARN	Condition property
resourceType	String	The type of resource associated with the operation activity	
resource	String	The default resource associated with the operation activity	
requiredTimeInProcess	String	This property defines the required elapsed time (in minutes) for the operation activity.	
workCenter	String	The work center associated with the operation activity	
customValues	StructureArray		
customValues.[i]	Structure		
customValues.[i].key	String		
customValues.[i].value	String		
specialRouting	Structure		
specialRouting.ref	String	The unique identifier of the routing	
specialRouting.plant	String	The plant for which the routing is defined	
specialRouting.routing	String	The name of the routing	Condition property
specialRouting.version	String	The version of the routing	
createdDateTime	Double	The time at which the operation is created	

Operation Activity Deleted

A business event is sent when an operation activity is deleted.

Event Properties

Name	Data Type	Description	Note
`ref`	String	The unique identifier of the operation activity	
`plant`	String	The plant for which the operation activity is defined	Condition property
`operation`	String	The name of the operation activity	Condition property
`version`	String	The version of the operation activity	

Operation Activity Updated

A business event is sent when an operation activity is updated.

Event Properties

Name	Data Type	Description	Note
`ref`	String	The unique identifier of the operation activity	
`plant`	String	The plant where the operation activity is updated	Condition property
`operation`	String	The name of the operation activity	Condition property
`version`	String	The version of the operation activity	
`currentVersion`	Boolean	If true, the operation activity is current.	Condition property
`description`	String	The description of the operation activity	
`type`	String	The type of operation activity. It could be any of the following types: • NORMAL_OPERATION • SPECIAL_OPERATION • TEST_OPERATION	Condition property

Name	Data Type	Description	Note
status	String	The status of the operation activity. It could be any of the following statuses: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOL • HOLD_SPC_WARN	Condition property
resourceType	String	The type of resource associated with the operation activity	
resource	String	The default resource associated with the operation activity	
requiredTimeInProcess	String	This property defines the required elapsed time (in minutes) for the operation activity.	
workCenter	String	The work center associated with the operation activity	
customValues	StructureArray		
customValues.[i]	Structure		
customValues.[i].key	String		
customValues.[i].value	String		
specialRouting	Structure		
specialRouting.ref	String	The unique identifier of the routing	
specialRouting.plant	String	The plant for which the routing is defined	
specialRouting.routing	String	The name of the routing	Condition property
specialRouting.version	String	The version of the routing	
modifiedDateTime	Double	The time at which the operation is updated	

Order Completed

A business event is sent when an order is completed.

Event Properties

Name	Data Type	Description	Note
`plant`	String	The plant where the order is completed	Condition property
`order`	String	The number of the order	Condition property
`buildQuantity`	Double	The build quantity of the order	
`orderedQuantity`	Double	The quantity ordered for the order	
`releasedQuantity`	Double	The quantity that is released for execution	
`doneQuantity`	Double	The done quantity of the order	
`baseUnitOfMeasure`	String	The base unit of measure of the order	
`releaseStatus`	String	The current release status of the order. It could be any of the following statuses: • PARTIALLY_CREATED • RELEASABLE • PARTIALLY_RELEASED • RELEASED • HOLD • NOT_RELEASABLE	Condition property
`executionStatus`	String	The current execution status of the order. It could be any of the following statuses: • NOT_IN_EXECUTION • ACTIVE • HOLD • CLOSED • COMPLETED • DISCARDED	Condition property
`actualStartDate`	DateTime	The start date and time of the order. For example,"2021-04-08T07:00:05.000Z".	
`actualCompletionDate`	DateTime	The date and time at which the order is completed. For example,"2021-04-08T07:00:05.000Z".	

Order Created

A business event is sent when an order is created.

Event Properties

Title	Data Type	Description	Note
plant	String	The plant where the order is created	Condition property
shopOrder	Structure		
shopOrder.batchNumber	String	The batch number of the order	
shopOrder.bomRef	String	The handle for the BOM assigned to the order	
shopOrder.buildQuantity	Double	The order's build quantity in basic UoM	
shopOrder.createdDateTime	DateTime	The date and time at which the order is created. For example, "2021-07-25T22:00:00.000+00:00".	
shopOrder.customer	String	The name of the order's customer	
shopOrder.customerOrder	String	The customer order	
shopOrder.doneQuantity	Double	The done quantity of the order in basic UoM	
shopOrder.erpOrder	Boolean	This property indicates if the order is ERP	
shopOrder.erpPutawayStorageLocation	String	The putaway storage location of the order	
shopOrder.erpUnitOfMeasure	String	The order's basic unit of measure in SAP uint code format. For example, ST.	
shopOrder.inspectionLot	String	The inspection lot of the order	
shopOrder.materialRef	String	The handle of the material assigned to the order	
shopOrder.modifiedDateTime	DateTime	The date and time at which the order is modified. For example,"2021-07-25T22:00:00.000+00:00".	
shopOrder.orderNumber	String	The number of the order	Condition property
shopOrder.orderedQuantity	Double	The ordered quantity in basic UoM	
shopOrder.overDeliveryIsAllowed	Boolean	This property indicates if unlimited overdelivery is allowed.	
shopOrder.plannedCompletiondate	DateTime	The planned completion date and time of the order. For example, "2021-07-25T22:00:00.000+00:00".	

Title	Data Type	Description	Note
shopOrder.plannedStartdate	DateTime	The planned start date and time of the order. For example, "2021-07-25T22:00:00.000+00:00".	
shopOrder.plant	String	The order's plant	
shopOrder.priority	Double	The priority assigned to the order	
shopOrder.productionQuantity	Double	The order's build quantity in production UoM	
shopOrder.productionUnitOfMeasure	String	The order's production unit of measure in SAP unit code format. For example, "ST".	
shopOrder.productionVersion	String	The production version of the order	
shopOrder.ref	String	The order handle	
shopOrder.releaseStatus	String	The release status of the order. It could be any of the following statuses: • UNDEFINED • PARTIALLY_CREATED • RELEASABLE • PARTIALLY_RELEASED • RELEASED	
shopOrder.releasedQuantity	Double	The released quantity of the order in basic UoM	
shopOrder.routingRef	String	The handle for the routing assigned to the order	
shopOrder.scheduledCompletionDate	DateTime	The scheduled completion date and time of the order. For example, "2021-07-25T22:00:00.000+00:00".	
shopOrder.scheduledStartDate	DateTime	The scheduled starting date and time of the order. For example, "2021-07-25T22:00:00.000+00:00".	
shopOrder.scrappedQuantity	Double	The order's scrapped quantity in basic UoM	
shopOrder.shopOrdertype	Structure		
shopOrder.shopOrdertype.description	String	The description of the order type	
shopOrder.shopOrdertype.oderProcessingType	String	The processing time of the order	

Title	Data Type	Description	Note
`shopOrder.shopOrdertype.orderType`	String	The order type. It could be either "PRODUCTION" or "REPETITIVE".	
`shopOrder.shopOrdertype.plant`	String	The plant of the order type	
`shopOrder.shopOrdertype.ref`	String	The handle of the order type	
`shopOrder.status`	String	The status of the order. It could be any of the following statuses: • RELEASABLE • HOLD • DONE • CLOSED • DISCARDED	
`shopOrder.toleranceOver`	Double	The overdelivery tolerance in %	
`shopOrder.toleranceUnder`	Double	The underdelivery tolerance in %	
`shopOrder.warehouseNumber`	String	The warehouse number of the order	

Order Released

A business event is sent when an order is released.

Event Properties

Name	Data Type	Description	Note
`plant`	String	The plant where the order is released.	Condition property
`order`	String	The order's number	Condition property
`routing`	Structure		
`routing.routing`	String	The Name of the routing / recipe	Condition property

Name	Data Type	Description	Note
routing.routingType (Deprecated)	String	Use routing.type instead. Indicates the internal type of Routing / Recipe. There are the following types: • PRODUCTION • NC • SPECIAL • DISPOSITION_FUNCTION • SFC_SPECIFIC • SHOPORDER_SPECIFIC • CONFIGURABLE • PRODUCTION_RECIPE • SHOPORDER_SPECIFIC_RECIPE	Condition property
routing.type	String	Indicates the type of routing / recipe. There are the following types: • PRODUCTION • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABLE • PRODUCTION_RECIPE • SHOPORDER_SPECIFIC_RECIPE	Condition property
routing.version	String	The version of the routing	Condition property
Bom	Structure		
Bom.bom	String	The ID of the BOM	Condition property
Bom.version	String	The version of the BOM	Condition property
Bom.bomType (Deprecated)	String	Use Bom.type instead. Indicates the internal type of BOM. There are the following types: • USERBOM • SHOPORDERBOM • SFCBOM • CONFIGURABLEBOM	

Name	Data Type	Description	Note
`Bom.type`	String	Indicates the type of BOM. There are the following types: • `MASTER` • `SHOP_ORDER` • `SFC`	
`createdDateTime`	String	The date and time (UTC) at which the order is created. For example, "2021-04-08T07:00:05.000Z".	
`material`	Structure		
`material.material`	String	The name of the material	Condition property
`material.plant`	String	The plant where the material is defined	
`material.ref`	String	The handle of the material	
`material.version`	String	The version of the material	Condition property
`batchesJson`	String	List of the batches	
`sfcsJson`	String	The SFC list JSON string	
`previousReleaseStatus`	String	The previous release status of the order	Condition property
`releaseStatus`	String	The current release status of the order. There are the following release statuses: • `PARTIALLY_CREATED` • `RELEASABLE` • `PARTIALLY_RELEASED` • `RELEASED` • `HOLD` • `NOT_RELEASABLE`	
`modifiedDateTime`	Integer	The date and time (UTC) at which the order was last modified . For example, "2021-04-08T07:00:05.000Z".	
`plannedQuantity`	Number	The planned build quantity of the order	
`releasedQuantity`	Number	The quantity that is released for execution	
`currentlyReleasedQuantity`	Number	The quantity that is released in the current release event **Example** Your order contains 10 pieces, and you've already released 5 pieces. When you release now 1 piece, your releasedQuantity is 6 pieces and the currentlyReleasedQuantity is 1 piece.	
`baseUnitOfMeasure`	String	The base unit of measure for this order	

Name	Data Type	Description	Note
sfcUnitOfMeasure	String	The unit of measure of the released SFCs	

Print Task Updated

A business event is sent when a print task is updated.

Event Properties

Name	Data Type	Description	Note
id	String	The print task ID	
plant	String	The plant	Condition property
document	String	The document print template	Condition property
documentVersion	String	The document print template version	Condition property
printer	String	The printer	Condition property
resource	String	The resource	Condition property
order	String	The order	Condition Property
workcenter	String	The work center	Condition Property
sfc	String	The SFC	Condition property
status	String	The print task status. The following statuses are possible: • SENT_TO_PRINTER • FAILED • PRINTED	Condition property
statusDescription	String	The print task status description	
modifiedOn	String	The modification date and time for a print task in UTC. Format (ISO-8601): yyyy-MM-dd'T'HH:mm:ss'Z'	

Production Changed

A business event is sent when production is changed.

Event Property

Name	Data Type	Description	Note
plant	String	The plant where the production change is performed	Condition property

Name	Data Type	Description	Note
sfc	Structure		
sfc.identifier	String	The shop floor control (SFC) to which the production change is performed	Condition property
originalOrder	Structure		
originalOrder.order	String	The order to execute before the production change is performed	Condition property
originalOrder.isAdjustBuildQuantity	Boolean	Indicates whether the build quantity of the order is changed.	
originalOrder.isAdjustReleaseQuantity	Boolean	Indicates whether the released quantity of the order is changed.	
targetOrder	Structure		
targetOrder.order	String	The order to execute after the production change is performed	Condition property
targetOrder.isAdjustBuildQuantity	Boolean	Indicates whether the build quantity of the order is changed.	
targetOrder.isAdjustReleaseQuantity	Boolean	Indicates whether the released quantity of the order is changed.	
newRouting	Structure		
newRouting.routing	String	The new routing assigned to the SFC	Condition property
newRouting.routingType	String	The type of the new routing. It can be any of the following types: PRODUCTION, SHOPORDER_SPECIFIC.	
newRouting.version	String	The version of the new routing	
newRouting.routingStepId	String	The routing step of the new routing. It determines from which routing step the SFC should be started.	
newRouting.resource	String	The new resource assigned to the SFC	
newRouting.operatorID	String	The operator who could execute the SFC to which production change is performed	
newBom	Structure		

Name	Data Type	Description	Note
newBom.bom	String	The new bill of materials (BOM) assigned to the SFC	Condition property
newBom.bomType	String	The type of the new BOM. It can be any of the following types: USERBOM, SHOPORDERBOM, SFCBOM.	
newBom.version	String	The version of the new BOM	
newMaterial	Structure		
newMaterial.material	String	The new material assigned to the SFC	Condition property
newMaterial.version	String	The version of the new material	

Resource Bin Unloaded

A business event is sent when a resource is unloaded.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant where the resource is unloaded	Condition property
resource	String	The name of the resource	Condition property
bin	String	The bin of the resource	Condition property
material	String	The unloaded material	Condition property
quantity	Double	The unloaded quantity	

Resource Downtimes Created

An event is sent when a resource downtime is created.

Event Properties

Name	Data Type	Description	Note
resource	Structure		
resource.plant	String	The plant of the resource	
resource.ref	String	The resource handle on which the downtime is being reported	
resource.resource	String	The name of the resource	Condition property
Downtime	Structure		

Name	Data Type	Description	Note
Downtime.ref	String	The unique identifier of the downtime	
Downtime.startTime	Integer	The start time of the downtime in milliseconds (UTC). For example, 1617861205000.	
Downtime.endTime	Integer	The end time of the downtime in milliseconds (UTC). For example, 1617861205000.	
Downtime.markLineDown	Boolean	Flag to notify that the work center is also down with this resource	
Downtime.reasonCode	Structure		
Downtime.reasonCode.ref	String	The reason code handle	
Downtime.reasonCode.plant	String	The plant of the reason code	
Downtime.reasonCode.reasonCode	String	The name of the reason code	
Downtime.**timeElementType**	Structure		
Downtime.timeElementType.ref	String	The handle of the time element	
Downtime.timeElementType.plant	String	The plant for which the time element is created	
Downtime.timeElementType.timeElementType	String	The description of the time element type	
Downtime.comments	**String**	The comments added by the operator for the downtime	

Resource Downtimes Deleted

A business event is sent when a resource downtime is deleted.

Event Properties

Name	Data Type	Description	Note
resource	Structure		
resource.plant	String	The plant of the resource	
resource.ref	String	The resource handle on which the downtime is being reported	
resource.resource	String	The name of the resource	Condition property
downtime	Structure		

Name	Data Type	Description	Note
downtime.ref	String	The unique identifier of the downtime event	
downtime.startTime	Integer	The start time of the downtime in milliseconds (UTC). For example, 1617862205000.	
downtime.endTime	Integer	The start time of the downtime in milliseconds (UTC). For example, 1617862205000.	
plant	String	The plant of the resource	

Resource Downtimes Updated

A business event is sent when a resource downtime is updated.

Event Properties

Name	Data Type	Description	Note
resource	Structure		
resource.plant	String	The plant of the resource	
resource.ref	String	The resource handle on which the downtime is being reported	
resource.resource	String	The name of the resource	Condition property
Downtime	Structure		
Downtime.ref	String	The unique identifier of the downtime	
Downtime.startTime	Integer	The start time of the downtime in milliseconds (UTC). For example, 1617861205000.	
Downtime.endTime	Integer	The end time of the downtime in milliseconds (UTC). For example, 1617861205000.	
Downtime.markLineDown	Boolean	Flag to notify that the work center is also down with this resource.	
Downtime.reasonCode	Structure		
Downtime.reasonCode.ref	String	The reason code handle	
Downtime.reasonCode.plant	String	The plant of the reason code	
Downtime.reasonCode.reasonCode	String	The name of the reason code	

Name	Data Type	Description	Note
Downtime.timeElementType	Structure		
Downtime.timeElementType.ref	String	The handle of the time element	
Downtime.timeElementType.plant	String	The plant for which the time element is created.	
Downtime.timeElementType.timeElementType	String	The description of the time element type	
Downtime.comments	String	The comments added by the operator for the downtime	

Resource Loaded

A business event is sent when a resource is loaded.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant of the resource	Condition property
resource	String	The name of the resource	Condition property
bin	String	The resource bin	Condition property
material	String	The loaded material	Condition property
quantity	Double	The loaded quantity	
dataFields	StructureArray		
dataFields.[i]	structure		
dataFields.[i].name	String	The name of the data field	
dataFields.[i].value	String	The value of the data field	
dataFields.[i].required	Boolean	This property indicates whether the data field is required or not.	

Resource Status Changed

A business event is sent when the status of a resource is changed.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant of the resource	Condition property
resource	String	The name of the resource	Condition property

Name	Data Type	Description	Note
`status`	String	The status of the resource. It could be any of the following statuses: • UNKNOWN • PRODUCTIVE • STANDBY • ENGINEERING • SCHEDULED_DOWN • UNSCHEDULED_DOWN • NON_SCHEDULED • ENABLED • DISABLED • HOLD • HOLD_SPC_VIOL • HOLD_SPC_WARN • HOLD_CONSEC_NC • HOLD_YIELD_RATE	Condition property
`user`	String	The resource user	
`machineCode`	String	The machine code	

Routing Created

A business event is sent when a routing is created.

Event Properties

Name	Data Type	Description	Note
`ref`	String	The unique identifier of the routing	
`plant`	String	The plant where the routing is defined.	Condition property
`routing`	String	The name of the routing	Condition property
`version`	String	The version of the routing	
`objectType`	String	Indicates the type of the object is a routing or a recipe.	
`description`	String	The description of the routing	
`currentVersion`	Boolean	If true, then the routing is current.	Condition Property
`quantityValidation`	Boolean	Indicates if quantity validation is required during production.	

Name	Data Type	Description	Note
type	String	Indicates the type of routing or recipe. There are the following different types: • PRODUCTION • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABLE	Condition property
status	String	The status of the routing. It could be any of the following statuses: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOLATION • HOLD_SPC_WARNING	Condition property
steps	StructureArray		
automaticGoodsReceipt	Boolean	Indicates if the routing is configured for an automatic goods receipt.	
relaxedFlow	Boolean	Indicates if the routing is an enforced or a relaxed flow routing.	
customData	StructureArray		
createdDateTime	Double	The date and time when the routing was created.	
Routing	Structure		
Routing.ref	String	The unique identifier of the routing	
Routing.plant	String	The plant for which the routing is defined.	Condition property
Routing.routing	String	The name of the routing	Condition property
Routing.version	String	The version of the routing	
CustomValue	Structure		

Name	Data Type	Description	Note
CustomValue.key	String		
CustomValue.value	String		
RoutingStep	Structure		
RoutingStep.ref	String	The unique identifier of a routing step	
RoutingStep.stepId	String	The step ID of a routing step	
RoutingStep.description	String	The description of a routing step	
RoutingStep.workcenter	String	The work center of a routing step	
RoutingStep.controlKey		The control key of a routing step	
RoutingStep.nextSteps	StructureArray		
RoutingStep.queueDecision	String	This determines who decides where an SFC should be placed in queue when completing this step. It can have the following values: • COMPLETING_OPERATOR • NEXT_OPERATOR	
RoutingStep.entryActivity	Boolean	The flag indicates that the routing step is an entry activty.	
RoutingStep.reworkActivity	Boolean	If true, then the time spent at this step is considered as a rework activity.	
RoutingStep.lastReportingActivity	Boolean	The flag indicates it is the last reporting step from ERP.	
RoutingStep.productionProcessId	String	The ID of the production process	
RoutingStep.productionProcessDefKey	String	The production process definition key	
RoutingStep.reportingStep	String	Indicates how this step is known in the external system.	
RoutingStep.sequence	Double	The sequence of the routing step.	
RoutingLink	Structure		
RoutingLink.ref	String	The unique identifier of the routing link	
RoutingOperation	Structure		
RoutingOperation.stepType	String	The type of routing operation step.	
RoutingOperation.erpAutoGr	Boolean	The flag marks a phase that is configured for automatic GR.	
RoutingOperation.longText	String	The long description of this operation.	

Name	Data Type	Description	Note
RoutingOperation.weighRelevant	Boolean	The flag which tells that weight is relevant.	
RoutingOperation.maxLoop	Double	Limits how many times an SFC may be restarted. If 0, then there is no limit.	
RoutingOperation.customData	StructureArray		
Operation	Structure		
Operation.ref	String	The unique identifier of the operation.	
Operation.plant	String	The plant for which the operation is defined.	Property condition
Operation.operation	String	The name of the operation	Property condition
Operation.version	String	The version of the operation	
WorkCenter	Structure		
WorkCenter.ref	String	The unique identifier of the work center	
WorkCenter.plant	String	The plant for which the work center is defined.	
WorkCenter.workcenter	String	The name of the work center	
ControlKey	Structure		
ControlKey.ref	String	The unique identifier of the control key	
ControlKey.plant	String	The plant for which the control key is defined	
ControlKey.controlKey	String	The name of the control key	
RoutingNextStep	Structure		
RoutingNextStep.ref	String	The unique identifier of the next routing step	
RoutingNextStep.routingStepActivity	Double	The sequence number of the next element	

Routing Updated

A business event is sent when a routing is updated.

Event Properties

Name	Data Type	Description	Note
ref	String	The unique identifier of the routing	
plant	String	The plant where the routing is defined.	Condition property

Name	Data Type	Description	Note
routing	String	The name of the routing	Condition property
version	String	The version of the routing	
objectType	String	Indicates the type of the object is a routing or a recipe.	
description	String	The description of the routing	
currentVersion	Boolean	If true, then the routing is current.	Condition Property
quantityValidation	Boolean	Indicates if quantity validation is required during production.	
type	String	Indicates the type of routing or recipe. There are the following different types: • PRODUCTION • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABLE	Condition property
status	String	The status of the routing. It could be any of the following statuses: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOLATION • HOLD_SPC_WARNING	Condition property
steps	StructureArray		
automaticGoodsReceipt	Boolean	Indicates if the routing is configured for an automatic goods receipt.	
relaxedFlow	Boolean	Indicates if the routing is an enforced or a relaxed flow routing.	
customData	StructureArray		
createdDateTime	Double	The date and time when the routing was updated.	

Name	Data Type	Description	Note
Routing	Structure		
Routing.ref	String	The unique identifier of the routing	
Routing.plant	String	The plant for which the routing is defined.	Condition property
Routing.routing	String	The name of the routing	Condition property
Routing.version	String	The version of the routing	
CustomValue	Structure		
CustomValue.key	String		
CustomValue.value	String		
RoutingStep	Structure		
RoutingStep.ref	String	The unique identifier of a routing step	
RoutingStep.stepId	String	The step ID of a routing step	
RoutingStep.description	String	The description of a routing step	
RoutingStep.workcenter	String	The work center of a routing step	
RoutingStep.controlKey		The control key of a routing step	
RoutingStep.nextSteps	StructureArray		
RoutingStep.queueDecision	String	This determines who decides where an SFC should be placed in queue when completing this step. It can have the following values: • COMPLETING_OPERATOR • NEXT_OPERATOR	
RoutingStep.entryActivity	Boolean	The flag indicates that the routing step is an entry activty.	
RoutingStep.reworkActivity	Boolean	If true, then the time spent at this step is considered as a rework activity.	
RoutingStep.lastReportingActivity	Boolean	The flag indicates it is the last reporting step from ERP.	
RoutingStep.productionProcessId	String	The ID of the production process	
RoutingStep.productionProcessDefKey	String	The production process definition key	
RoutingStep.reportingStep	String	Indicates how this step is known in the external system.	
RoutingStep.sequence	Double	The sequence of the routing step.	
RoutingLink	Structure		

Name	Data Type	Description	Note
RoutingLink.ref	String	The unique identifier of the routing link	
RoutingOperation	Structure		
RoutingOperation.erpAutoGr	Boolean	The flag marks a phase that is configured for automatic GR.	
RoutingOperation.longText	String	The long description of this operation.	
RoutingOperation.stepType	String	The type of routing operation step	
RoutingOperation.weighRelevant	Boolean	The flag which tells that weight is relevant.	
RoutingOperation.maxLoop	Double	Limits how many times an SFC may be restarted. If 0, then there is no limit.	
RoutingOperation.customData	StructureArray		
Operation	Structure		
Operation.ref	String	The unique identifier of the operation.	
Operation.plant	String	The plant for which the operation is defined.	Property condition
Operation.operation	String	The name of the operation	Property condition
Operation.version	String	The version of the operation	
WorkCenter	Structure		
WorkCenter.ref	String	The unique identifier of the work center	
WorkCenter.plant	String	The plant for which the work center is defined.	
WorkCenter.workcenter	String	The name of the work center	
ControlKey	Structure		
ControlKey.ref	String	The unique identifier of the control key	
ControlKey.plant	String	The plant for which the control key is defined	
ControlKey.controlKey	String	The name of the control key	
RoutingNextStep	Structure		
RoutingNextStep.ref	String	The unique identifier of the next routing step	
RoutingNextStep.routingStepActivity	Double	The sequence number of the next element	

Routing Deleted

A business event is sent when a routing is deleted.

Event Properties

Name	Data Type	Description	Note
`ref`	String	The unique identifier of the routing	
`plant`	String	The plant for which the routing is defined	Condition property
`routing`	String	The name of the routing	Condition property
`version`	String	The version of the routing	

SFC Created

A business event is sent when a shop floor control is created.

Event Property

Name	Data Type	Description	Note
`plant`	String	The plant of the shop floor where the SFC is created	Condition property
`order`	Structure		
`order.order`	String	The name of the order	Condition property
`router`	Structure		
`router.routee`	String	The description of the SFC router	Condition property
`router.version`	String	The version of the SFC router	Condition property
`sfc`	Structure		
`sfc.identifier`	String	The unique numeric identifier of the SFC	
`sfc.buildQuantity`	Double	The total build quantity. The total quantity might differ from the original release quantity, since creating an SFC increases the build quantity.	
`sfc.quantity`	Double	The quantity created. It could be the full quantity or a partial quantity.	
`sfc.unitOfMeasure`	String	ISO Code of the measure of the quantity	
`priority`	Long	The priority assigned to the SFC	
`status`	String	The status of the SFC	Condition property
`createdTime`	DateTime	The UTC time at which the SFC is created	

Related Information

Creating SFCs in the Operation Activity POD

SFC Data Collected

A business event is sent when data is collected while processing shop floor controls (SFCs).

Event Properties

Name	Data Type	Description	Note
plant	String	The plant where the data collection originated from	Condition property
sfcs	StringArray	The SFCs for which data has been collected	
workCenter	String	The work center where the data collection originated from.	Condition property
userId	String	The ID of the user initiating the data collection	Condition property
dcGroup	Structure		
dcGroup.dcGroup	String	The name of the data collection group	Condition property
dcGroup.version	String	The version of the data collection group	Condition property
operation	Structure		
operation.operation	String	The name of the operation	Condition property
operation.version	String	The version of the operation	Condition property
resource	String	The resource where the SFC data is collected	Condition property
parameters	StructureArray		
parameters.[i]	Structure		
parameters.[i].name	String	The name of a parameter within the data collection group	
parameters.[i].value	String	The value assigned to the parameter	

SFC Dispositioned

A business event is sent when a shop floor control is dispositioned.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant where the SFC is dispositioned	Condition property
sfcDispositions	StructureArray		
sfcDispositions.[i]	Structure		

Name	Data Type	Description	Note
sfcDispositions.[i].sfcRef	String	The reference string of the SFC	
sfcDispositions.[i].sfc	String	The name of the SFC	
sfcDispositions.[i].type	String	The disposition type of the SFC. It could be any of the following types: • COMPLETE_PENDING • DONE • DYNAMIC_ROUTING • NO_DISPOSITION • PARTIAL_COMPLETE • QUEUE • RETURN • SCRAP • TRANSFER	
resource	String	The name of the resource where the SFC is dispositioned	
operation	String	The name of the operation activity where the SFC is dispositioned	
order	String	The order name of the dispositioned SFC	

SFC Operation Activity Completed

A business event is sent when all shop floor quantity is completed.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant where the operation activity is completed	Condition property
routingStep	Structure		
routingStep.operationActivityNumber	String	The step specific activity identifier	Condition property
routingStep.operationActivity	Structure		
routingStep.operationActivity.operationActivity	String	The name of the operation	Condition property
routingStep.operationActivity.version	String	The version of the operation	Condition property

Name	Data Type	Description	Note
routingStep.routing	Structure		
routingStep.routing.routing	String	The name of the routing number	Condition property
routingStep.routing.type	String	The routing type	Condition property
routingStep.routing.version	String	The version of the routing	Condition property
resource	Structure		
resource.resource	String	The name of the resource	Condition property
workCenter	Structure		
workCenter.workCenter	String	The name of the resource. This is only set if the resource is a member of the work center.	Condition property
order	Structure		
order.order	String	The order ID	Condition property
sfcs	StructureArray		
sfc.[i]	Structure		
sfc.[i].identifier	String	The unique identifier of the SFC	
previousCompletedTime	Integer	The UTC time in milliseconds at which the SFC was previously completed	
completedTime	Integer	The UTC time in milliseconds at which the SFC is completed	

SFC Operation Activity Started

A business event is sent the first time a shop floor control is started at a given operation and resource, or when the start time is updated via a post production process. All the shop floor controls are on the same router and order.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant where the SFC is started	Condition property
sfcs	StructureArray	The list of the started SFCs	

Name	Data Type	Description	Note
sfcs.[i]	Structure		
sfcs.[i].identifier	String	The identification number of the SFC	
resource	Structure	The resource where the SFC is started	
resource.resource	String	The name of the resource	Condition property
order	Structure	The shop order of the SFC	
order.order	String	The name of the order. The order applies to all SFCs	Condition property
routingStep	Structure	The routing step where the SFC is started	
routingStep.operationActivityNumber	String	The step specific operation activity identifier	Condition property
routingStep.operationActivity	Structure		
routingStep.operationActivity.operationActivity	String	The name of the operation	Condition property
routingStep.operationActivity.version	String	The version of the operation	Condition property
routingStep.routing	Structure		
routingStep.routing.routing	String	The name of the routing.	Condition property
routingStep.routing.type	String	The type of routing	Condition property
routingStep.routing.version	String	The version of the routing	Condition property
startedTime	Integer	The UTC time in milliseconds at which the SFCs are started	
previousStartedTime	Integer	The UTC time in milliseconds at which the SFCs were previously started	

SFC Relabel

An event is sent when a shop floor control is relabeled.

Event Properties

Name	Data Type	Description	Note

Name	Data Type	Description	Note
plant	String	The plant where the SFCs are released.	Condition property
sfc	String	The original SFC for relabeling.	Condition property
newSfc	String	The SFC that is relabeled.	Condition property
order	String	The order of the destination SFC.	Condition property
routingStepId	String	The routing step where an SFC is relabeled.	Condition property
routing	String	The routing where an SFC is relabeled.	Condition property
routingType	String	The type of the routing where an SFC is relabeled.	Condition property

SFC Scrapped

Business event sent when shop floor control is scrapped.

Event Properties

Name	Data Type	Description	Note
plant	String	Plant of the shop floor control that was scrapped.	Condition property
scrappedTime	Long	The UTC time in milliseconds when the shop floor controls were scrapped.	
Sfc	Structure		
Sfc[i]identifier	String	The shop floor control identifier as generated by numbering.	
Sfc[i]buildQuantity	Double	The total build quantity. This quantity may be different than the original release quantity. For example, scrap reduces the build quantity.	
Sfc[i]quantity	Double	The quantity was scrapped. This should be a full quantity of SFC. Scrapping of partial SFC quantity is currently not supported.	
Sfc[i]unitOfMeasure	String	ISO Code of the measure of the quantity.	
Resource	Structure		

Name	Data Type	Description	Note
Resource.resource	String	The resource name.	Condition property
ReasonCode	Structure		
ReasonCode.reasonCode	String	The reason code related to resource scrapping.	Condition property
operationActivity	String	The operation name.	Condition property
version	String	The operation version.	Condition property
Order	Structure		
Order.order	String	The order name.	Condition property
WorkCenter	Structure		
WorkCenter.workCenter	String	The workcenter name.	Condition property
UserId	Structure		
UserId.user	String	The user id of user performing scrap	Condition property
RoutingStep	Structure		
RoutingStep.operationActivityNumber	String	The step specific operation activity identifier.	Condition property
RoutingStep.Routing	Structure		
RoutingStep.Routing.routing	String	The routing name.	Condition property
RoutingStep.Routing.type	String	The routing type.	Condition property
RoutingStep.Routing.version	String	The routing version.	Condition property

SFC Serialized

An event is sent when a shop floor control is serialized.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant where the SFCs are released.	Condition property
sfc	String	The original SFC for serializing.	Condition property
routingStepId	String	The routing step where an SFC is serialized.	Condition property
routing	String	The routing where an SFC is serialized.	Condition property
routingType	String	The type of the routing where an SFC is serialized.	Condition property
order	String	The order of the destination SFC.	Condition property

Name	Data Type	Description	Note
newSfcs	StringArray	SFCs that are serialized.	

SFC Split

An event is sent when a shop floor control is split.

Event properties

Name	Data Type	Description	Note
plant	String	The plant where the SFC is split	Condition property
order	String	The number of the order	Condition property
sfc	String	The original SFC that is split	Condition property
stepId	String	The routing step in which the SFC is split	Condition property
newSfcs	StringArray	The list of new SFCs and quantities that are created during the SFC split	
buildQuantity	Double	The build quantity of the order	
routingType	String	The type of routing used to determine where to split an SFC.	Condition property
routing	String	The name of the routing indicating the location for splitting an SFC.	Condition property

Related Information

SFC Split

SFC Started

A business event is sent when a shop floor control is started.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant of the SFC that is started	Condition property
sfcs	StructureArray		
sfcs.[i]	Structure		
sfcs.[i].identifier	String	The unique identifier of the SFC	

Name	Data Type	Description	Note
sfcs.[i].buildQuantity	Double	The total build quantity. This quantity differs from the original release quantity. For example, scrapping reduces the build quantity.	
sfcs.[i].quantity	Double	The quantity started. This quantity can be full or partial	
sfcs.[i].unitOfMeasure	String	The ISO Code of the measure of the quantity	
resource	Structure		
resource.resource	String	The name of the resource	Condition property
workCenter	Structure		
workCenter.workCenter	String	The name of the work center. This is only set if the resource is a member of a work center	Condition property
routingStep	Structure		
routingStep.operationActivityNumber	String	The step specific operation activity number	Condition property
routingStep.operationActivity	Structure		
routingStep.operationActivity.operationActivity	String	The name of the operation	Condition property
routingStep.operationActivity.version	String	The version of the operation	Condition property
startedTime	Integer	The UTC time in milliseconds at which the SFC is started	

SFC Yield

A business event is sent when the SFC quantity is yielded.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant of the shop floor where the SFC quantity is yielded	Condition property

Name	Data Type	Description	Note
routingStep	Structure		
routingStep.operationActivityNumber	String	The step specific operation activity identifier	Condition property
routingStep.operationActivity	Structure		
routingStep.operationActivity.operationActivity	String	The name of the operation	Condition property
routingStep.operationActivity.version	String	The version of the operation	Condition property
routingStep.routing	Structure		
routingStep.routing.routing	String	The name of the routing	Condition property
routingStep.routing.type	String	The routing type	Condition property
routingStep.routing.version	String	The version of the routing	Condition property
order	Structure		
order.order	String	The name of the order	Condition property
resource	Structure		
resource.resource	String	The name of the resource	Condition property
workCenter	Structure		
workCenter.workCenter	String	The name of the work center. This is set only if the resource is a member of a work center.	Condition property
sfcs	StructureArray		
sfcs.[i]	Structure		
sfcs.[i].identifier	String	The unique identifier of the SFC	Condition property
sfcs.[i].buildQuantity	Double	This quantity differs from the original release quantity. For example, scrapping reduces the build quantity	Condition property
sfcs.[i].quantity	Double	The yielded quantity. This quantity can be full or partial.	Condition property
sfcs.[i].unitOfMeasure	String	The ISO code of the measure of the quantity	Condition property

Name	Data Type	Description	Note
yieldTime	Integer	The UTC time in milliseconds of the quantity yield	

SFCs Merged

A business event is sent when shop floor controls are merged.

Event Properties

Name	Data Type	Description	Note
plant	String	The plant where the SFCs are released.	Condition property
destinationSfc	String	The destination SFC for merging.	Condition property
stepId	String	The routing operation activity where to merge an SFC.	Condition property
routing	String	The routing where to merge an SFC.	Condition property
routingType	String	The type of the routing where to merge an SFC.	Condition property
order	String	The order of the destination SFC.	Condition property
sourceSfcs	StringArray	SFCs that are merged.	

Related Information

SFC Merge

Stock Changed

A business event is sent when an EWM stock is changed.

ℹ Note

To use the **Stock Changed** event, you must enable **Send Event Upon Stock Change** in the **Inventory** category of the **Manage Business Settings** app.

Event Property

Name	Data Type	Description	Note
plant	String	The plant where the stock change is performed	Condition property
material	String	The material name of the stock	Condition property
materialVersion	String	The material version of the stock	Condition property

Name	Data Type	Description	Note
`storageLocation`	String	The storage location of the stock	Condition property
`productionSupplyArea`	String	The production supply area of the stock. This property only applies to EWM stocks.	Condition property
`handlingUnit`	String	The handling unit of the stock's outermost package. This property only applies to EWM stocks.	Condition property
`batchNumber`	String	The batch number of the stock	Condition property
`inventoryId`	String	The inventory ID of the stock in SAP Digital Manufacturing	Condition property
`quantityOnHand`	Number	The available quantity of the stock	Condition property
`quantityChanged`	Number	The quantity that's changed for the stock	Condition property
`storageBin`	String	The storage bin where the stock is stored at the production supply area	Condition property
`currentStockStatus`	String	The current status of the stock. It can be any of the following statuses: • AVAILABLE • HOLD • QUARANTINE	Condition property
`previousStockStatus`	String	The previous status of the stock before the stock change. It can be any of the following statuses: • AVAILABLE • HOLD • QUARANTINE	Condition property
`reservedOperation`	String	The operation for which the stock is reserved	Condition property
`reservedOrder`	String	The order for which the stock is reserved	Condition property
`serialNumber`	String	The serial number of the stock	Condition property

Name	Data Type	Description	Note
`stockChangeCategory`	String	The trigger points of the stock change. It can be any of the following triggers: • GR The stock change is triggered by a goods receipt. • GR_CANCELLATION The stock change is triggered by a goods receipt cancellation. • GI The stock change is triggered by a goods issue. • GI_CANCELLATION The stock change is triggered by a goods issue cancellation. • COMPONENT_REMOVAL The stock change is triggered by the component removal. • EWM_STOCK_SYNC The stock change is triggered by the stock change notification from the integrated EWM system.	Condition property
`assemblyData`	StructureArray		
`assemblyData.[i]`	Structure		
`assemblyData.[i].dataAttribute`	String	The field value of the stock data type	
`assemblyData.[i].dataField`	String	The field name of the stock data type	
`assemblyData.[i].sequence`	Number	The sequence of the data fields. The number defines in which sequence the data fields are displayed.	
`unitOfMeasure`	Structure		
`unitOfMeasure.internalUnitOfMeasure`	String	The stock's quantity in SAP internal unit of measure	

Tool Calibrated

A business event is sent when a tool is calibrated.

Event Property

Name	Data Type	Description	Note
`plant`	String	The plant where the tool calibration is performed.	Condition property
`toolNumber`	String	The tool number of the tool that's calibrated.	Condition property
`maintenanceType`	String	The type of the tool calibration. It can be one of the following types: • MANUAL • ERP	
`createdOn`	DateTime	The date and time when the tool calibration was performed (UTC). For example, "2023-01-11T16:59:24.000Z".	

Tool Created

A business event is sent when a tool is created.

Event Property

Name	Data Type	Description	Note
`plant`	String	The plant where the tool creation is performed.	Condition property
`toolNumber`	String	The tool number of the tool that's created.	Condition property
`status`	String	The status of the tool. It can be one of the following statuses: • DISABLED • PRODUCTIVE • ENGINEERING • ENABLED	
`equipmentId`	String	The equipment ID of the equipment relevant to the tool. It's applicable to the Asset Central Model.	
`assetId`	String	The asset ID of the equipment relevant to the tool. It's applicable to the Digital Manufacturing Asset Model.	

Name	Data Type	Description	Note
material.[i]	Structure		
material.[i].material	String	The name of the material.	
material.[i].version	String	The version of the material.	
serialNumber	String	The serial number of the tool.	
origin	String	The origin of the tool. It can be one of the following origins: • LOCAL Indicates that the tool is created in the system. • ERP Indicates that the tool is transferred from an integrated ERP system.	Condition property
description	String	The description of the tool.	
location	String	The current location of the tool.	
loggingSetup.[i]	Structure		
loggingSetup.[i].loggingMethod	String	The logging method of the tool. It can be one of the following methods: • MANUAL • AUTOMATIC • NON_TRACKED	
loggingSetup.[i].manual.[i]	Structure		
loggingSetup.[i].manual.[i].isByCounter	Boolean	Indicates that the tool usage is calculated by counts.	
loggingSetup.[i].manual.[i].isByTime	Boolean	Indicates that the tool usage is calculated by duration of time.	
loggingSetup.[i].automatic.[i]	Structure		
loggingSetup.[i].automatic.[i].counter.[i]	Structure		
loggingSetup.[i].automatic.[i].counter.[i].isByCounter	Boolean	Indicates that the tool usage is calculated by counts.	

Name	Data Type	Description	Note
`loggingSetup.[i].automatic.[i].counter.[i].autoMethodBaseCount`	Number	The base count by which the used quantity of the tool is counted. Base count allows a maximum of 10 integer digits and 2 decimal digits.	
`loggingSetup.[i].automatic.[i].counter.[i].autoMethodCountBy`	String	Indicates how the used quantity of the tool is counted. It can be one of the following: • SFC • SFC_QUANTITY	
`loggingSetup.[i].automatic.[i].timer.[i]`	Structure		
`loggingSetup.[i].automatic.[i].timer.[i].isByTime`	Boolean	Indicates that the tool usage is calculated by duration of time.	
`loggingSetup.[i].automatic.[i].timer.[i].autoMethodBaseTime`	Number	The base time by which the used time of the tool is calculated. It's measured in seconds and it only accepts integers.	
`loggingSetup.[i].automatic.[i].timer.[i].autoMethodCalculateBy`	String	Indicates how the used time of the tool is calculated. It can be one of the following: • SFC • SFC_QUANTITY	
`CustomData.[i]`	Structure		
`CustomData.[i].attribute`	String	The custom data field name.	
`CustomData.[i].value`	String	The custom data field value.	

Tool Loading Saved

A business event is sent when a tool is loaded or unloaded.

Event Property

Name	Data Type	Description	Note
`plant`	String	The plant where the tool loading is performed	Condition property
`toolNumber`	String	The tool number of the tool that's loaded or unloaded	Condition property

Name	Data Type	Description	Note
`status`	String	Tool status after loading or unloading	Condition property
`currentLocation`	String	Indicates the location where the tool is after loading or unloading	Condition property
`currentLocationType`	String	Indicates the type of the location where the tool is after loading or unloading	
`toolType`	String	The type of tool that's loaded or unloaded. It can be one of the following types: • EQUIPMENT A tool with reference to an equipment • MATERIAL A tool with reference to a material	
`modifiedOn`	DateTime	The UTC date time when the tool is loaded or unloaded. Example: 2023-01-11T16:59:24.000Z	
`flag`	String	Indicates whether the tool is loaded or unloaded. It can be one of the following types: • LOAD • UNLOAD	Condition property

Tool Updated

A business event is sent when a tool is updated.

Event Property

Name	Data Type	Description	Note
`plant`	String	The plant where the tool update is performed.	Condition property
`toolNumber`	String	The tool number of the tool that's updated.	Condition property
`status`	String	The status of the tool. It can be one of the following statuses: • DISABLED • PRODUCTIVE • ENGINEERING • ENABLED	Condition property

Name	Data Type	Description	Note
`equipmentId`	String	The equipment ID of the equipment relevant to the tool. It's applicable to the Asset Central Model.	
`assetId`	String	The asset ID of the equipment relevant to the tool. It's applicable to the Digital Manufacturing Asset Model.	
`material.[i]`	Structure		
`material.[i].material`	String	The name of the material.	
`material.[i].version`	String	The version of the material.	
`serialNumber`	String	The serial number of the tool.	
`origin`	String	The origin of the tool. It can be one of the following origins: • LOCAL The tool is created in the system. • ERP The tool is transferred from an integrated ERP system.	Condition property
`description`	String	The description of the tool.	
`location`	String	The current location of the tool.	Condition property
`loggingSetup.[i]`	Structure		
`loggingSetup.[i].loggingMethod`	String	The logging method of the tool. It can be one of the following methods: • MANUAL • AUTOMATIC • NON_TRACKED	
`loggingSetup.[i].manual.[i]`	Structure		
`loggingSetup.[i].manual.[i].isByCounter`	Boolean	Indicates that the tool usage is calculated by counts.	
`loggingSetup.[i].manual.[i].isByTime`	Boolean	Indicates that the tool usage is calculated by duration of time.	
`loggingSetup.[i].automatic.[i]`	Structure		
`loggingSetup.[i].automatic.[i].counter.[i]`	Structure		

Name	Data Type	Description	Note
`loggingSetup.[i].automatic.[i].counter.[i].isByCounter`	Boolean	Indicates that the tool usage is calculated by counts.	
`loggingSetup.[i].automatic.[i].counter.[i].autoMethodBaseCount`	Number	The base count by which the used quantity of the tool is counted. Base count allows a maximum of 10 integer digits and 2 decimal digits.	
`loggingSetup.[i].automatic.[i].counter.[i].autoMethodCountBy`	String	Indicates how the used quantity of the tool is counted. It can be one of the following: • SFC • SFC_QUANTITY	
`loggingSetup.[i].automatic.[i].timer.[i]`	Structure		
`loggingSetup.[i].automatic.[i].timer.[i].isByTime`	Boolean	Indicates that the tool usage is calculated by duration of time.	
`loggingSetup.[i].automatic.[i].timer.[i].autoMethodBaseTime`	Number	The base time by which the used time of the tool is calculated. It's measured in seconds and it only accepts integers.	
`loggingSetup.[i].automatic.[i].timer.[i].autoMethodCalculateBy`	String	Indicates how the used time of the tool is calculated. It can be one of the following: • SFC • SFC_QUANTITY	
`toolType`	String	The type of the tool. It can be one of the following types: • EQUIPMENT_PRT The tool is a equipment type of tool. • MATERIAL_PRT The tool is a material type of tool.	Condition property
`CustomData.[i]`	Structure		
`CustomData.[i].attribute`	String	The custom data field name.	
`CustomData.[i].value`	String	The custom data field value.	
`prtNumber`	String	The production resources/tools (PRT) number of the tool.	Condition property

Name	Data Type	Description	Note
createdOn	DateTime	The date and time when the tool was created (UTC). For example, "2023-01-11T16:59:24.000Z".	
modifiedOn	DateTime	The date and time when the tool was last modified (UTC). For example, "2023-01-11T16:59:24.000Z".	

Tool Usage Logged

A business event is sent when usage is logged for a tool.

Event Property

Name	Data Type	Description	Note
plant	String	The plant where the tool usage is logged.	Condition property
sfc.[i]	Structure		
sfc.[i].identifier	String	The shop floor control (SFC) against which the tool usage is logged.	Condition property
resource	String	The resource to which the tool is assigned.	Condition property
operation.[i]	Structure		
operation.[i].operation	String	The operation against which the tool usage is logged.	Condition property
operation.[i].operationVersion	String	The version of the operation.	
toolUsageLogs	StructureArray		
toolUsageLogs.[i]	Structure		
toolUsageLogs.[i].id	String	The ID of the tool usage log table.	
toolUsageLogs.[i].toolNumber	String	The tool number of the tool for which the the usage is logged.	Condition property
toolUsageLogs.[i].loggingMethod	String	The logging method of the tool. It can be any of the following methods: • MANUAL • AUTOMATIC • NON_TRACKED	
toolUsageLogs.[i].usageCount	Number	The number of times the tool is used.	

Name	Data Type	Description	Note
`toolUsageLogs.[i].startTime`	DateTime	The date and time when the tool started being used (UTC). For example, "2023-01-11T16:59:24.000Z".	
`toolUsageLogs.[i].endTime`	DateTime	The date and time when the tool stopped being used (UTC). For example, "2023-01-11T16:59:24.000Z".	
`toolUsageLogs.[i].usageTime`	Integer	The duration of time the tool is used. It's measured in seconds.	
`toolUsageLogs.[i].comments`	String	Comments of the tool usage logging record.	
`toolUsageLogs.[i].createdOn`	DateTime	The date and time when the tool usage logging record was created (UTC). For example, "2023-01-11T16:59:24.000Z".	
`toolUsageLogs.[i].modifiedOn`	DateTime	The date and time when the tool usage logging record was last modified (UTC). For example, "2023-01-11T16:59:24.000Z".	

Work Instruction Created

A business event is sent when a work instruction is created.

Event Properties

Name	Data Type	Description	Note
`ref`	String	Unique identifier of the work instruction.	
`plant`	String	The plant for which the work instruction is defined.	Condition property
`workInstruction`	String	The name of the work instruction.	Condition property
`currentVersion`	Boolean	If true, then this work instruction is current.	Condition property
`version`	String	The version of the work instruction.	Condition property
`descriptions`	String	The description of the work instruction.	
`status`	String	The status of the work instruction. • `NEW` • `RELEASABLE` • `OBSOLETE` • `HOLD`	

Name	Data Type	Description	Note
erpWorkInstruction	Boolean	Indicates whether this work instruction was sent from ERP.	
trackViewing	Boolean	If true, then the viewing of this work instruction is tracked for each user.	
createdDateTime	number	The time when the work instruction was created.	
CustomValue	Structure		
CustomValue[i]key	String	Custom field key name.	
CustomValue[i]value	String	Custom field value.	
WorkInstructionAttachedPoint	Structure		
WorkInstructionAttachedPoint[i]ref	String	The unique identifier of the work instruction attached point.	
WorkInstructionAttachedPoint[i]resource	String	A resource to which the work instruction is attached.	
WorkInstructionAttachedPoint[i]routingStepId	String	A routing step to which the work instruction is attached.	
WorkInstructionAttachedPoint[i]sequence	number	The sequence of the attachment point.	
WorkInstructionAttachedPoint[i]shopOrder	String	A shop order to which the work instruction is attached.	
WorkInstructionAttachedPoint[i]workCenter	String	A work center to which the work instruction is attached.	
WorkInstructionAttachedPoint[i]headerMaterial	Boolean	Defines if a material is a header material.	
WorkInstructionElement	Structure		
WorkInstructionElement[i]ref	String	The unique identifier of the work instruction element.	
WorkInstructionElement[i]fileName	String	The file name of the element.	
WorkInstructionElement[i]fileExternalUrl	String	The URL of the file of the element.	
WorkInstructionElement[i]mimeType	String	The MIME type of the file of the element.	
WorkInstructionElement[i]sequence	Long	The sequence of the element.	
WorkInstructionElement[i]text	String	The text content of the element.	
WorkInstructionElement[i]description	String	The description of the element.	
WorkInstructionElement[i]type	String	The type of the element. • TEXT • FILE • URL • HEADER_TEXT • LOCAL_FILE	
WorkInstructionElement[i]url	String	The URL of the element.	

Name	Data Type	Description	Note
WorkInstructionElement[i]newWindow	Boolean	Defines if a work instruction can be opened in a new window.	
WorkInstructionElement[i]erpFilename	String	The ERP file name.	
WorkInstructionElement[i]metadataCategory	String	The metadata category of a 3D file.	
WorkInstructionElement[i]metadataComponent	String	The 3D file metadata component name.	
WorkInstructionElement[i]metadataRefDes	String	The 3D file metadata installation point.	
WorkInstructionElement[i]storageType	String	The storage type of the element. • AMAZON • GOOGLE • AZURE	
Material	Structure		
Material.ref	String	The unique identifier of the material.	
Material.plant	String	The plant the material belongs to.	
Material.material	String	The name of the material.	
Material.version	String	The version of the material.	
Operation	Structure		
Operation.ref	String	The unique identifier of the operation.	
Operation.plant	String	The plant the operation activity belongs to.	
Operation.operation	String	The name of the operation.	
Operation.version	String	The version of the operation.	
Routing	Structure		
Routing.ref	String	The unique identifier of the routing.	
Routing.plant	String	The plant for which the routing is defined.	
Routing.routing	String	The name of the routing.	
Routing.version	String	The version of the routing.	
Routing.type	String	The type of the routing. • PRODUCTION • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABLE • PRODUCTION_RECIPE • SHOPORDER_SPECIFIC_RECIPE	

Work Instruction Updated

A business event is sent when a work instruction is updated.

Event Properties

Name	Data Type	Description	Note
`ref`	String	Unique identifier of the work instruction.	
`plant`	String	The plant for which the work instruction is defined.	Condition property
`workInstruction`	String	The name of the work instruction.	Condition property
`currentVersion`	Boolean	If true, then this work instruction is current.	Condition property
`version`	String	The version of the work instruction.	Condition property
`descriptions`	String	The description of the work instruction.	
`status`	String	The status of the work instruction. • `NEW` • `RELEASABLE` • `OBSOLETE` • `HOLD`	
`erpWorkInstruction`	Boolean	Indicates whether this work instruction was sent from ERP.	
`trackViewing`	Boolean	If true, then the viewing of this work instruction is tracked for each user.	
`createdDateTime`	number	The time when the work instruction was created.	
`CustomValue`	Structure		
`CustomValue[i]key`	String	Custom field key name.	
`CustomValue[i]value`	String	Custom field value.	
`WorkInstructionAttachedPoint`	Structure		
`WorkInstructionAttachedPoint[i]ref`	String	The unique identifier of the work instruction attached point.	
`WorkInstructionAttachedPoint[i]resource`	String	A resource to which the work instruction is attached.	
`WorkInstructionAttachedPoint[i]routingStepId`	String	A routing step to which the work instruction is attached.	
`WorkInstructionAttachedPoint[i]sequence`	number	The sequence of the attachment point.	
`WorkInstructionAttachedPoint[i]shopOrder`	String	A shop order to which the work instruction is attached.	
`WorkInstructionAttachedPoint[i]workCenter`	String	A work center to which the work instruction is attached.	

Name	Data Type	Description	Note
WorkInstructionAttachedPoint[i]headerMaterial	Boolean	Defines if a material is a header material.	
WorkInstructionElement	Structure		
WorkInstructionElement[i]ref	String	The unique identifier of the work instruction element.	
WorkInstructionElement[i]fileName	String	The file name of the element.	
WorkInstructionElement[i]fileExternalUrl	String	The URL of the file of the element.	
WorkInstructionElement[i]mimeType	String	The MIME type of the file of the element.	
WorkInstructionElement[i]sequence	Long	The sequence of the element.	
WorkInstructionElement[i]text	String	The text content of the element.	
WorkInstructionElement[i]description	String	The description of the element.	
WorkInstructionElement[i]type	String	The type of the element. • TEXT • FILE • URL • HEADER_TEXT • LOCAL_FILE	
WorkInstructionElement[i]url	String	The URL of the element.	
WorkInstructionElement[i]newWindow	Boolean	Defines if a work instruction can be opened in a new window.	
WorkInstructionElement[i]erpFilename	String	The ERP file name.	
WorkInstructionElement[i]metadataCategory	String	The metadata category of a 3D file.	
WorkInstructionElement[i]metadataComponent	String	The 3D file metadata component name.	
WorkInstructionElement[i]metadataRefDes	String	The 3D file metadata installation point.	
WorkInstructionElement[i]storageType	String	The storage type of the element. • AMAZON • GOOGLE • AZURE	
Material	Structure		
Material.ref	String	The unique identifier of the material.	
Material.plant	String	The plant the material belongs to.	
Material.material	String	The name of the material.	
Material.version	String	The version of the material.	
Operation	Structure		
Operation.ref	String	The unique identifier of the operation.	
Operation.plant	String	The plant the operation activity belongs to.	

Name	Data Type	Description	Note
Operation.operation	String	The name of the operation.	
Operation.version	String	The version of the operation.	
Routing	Structure		
Routing.ref	String	The unique identifier of the routing.	
Routing.plant	String	The plant for which the routing is defined.	
Routing.routing	String	The name of the routing.	
Routing.version	String	The version of the routing.	
Routing.type	String	The type of the routing. • PRODUCTION • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABLE • PRODUCTION_RECIPE • SHOPORDER_SPECIFIC_RECIPE	
ref	String	Unique identifier of the work instruction.	
plant	String	The plant for which the work instruction is defined.	Condition property
workInstruction	String	The name of the work instruction.	Condition property
version	String	The version of the work instruction.	Condition property

Work Instruction Deleted

A business event is sent when a work instruction is deleted.

Event Properties

Name	Data Type	Properties	Note
ref	String	Unique identifier of the work instruction.	
plant	String	The plant for which the work instruction is defined.	Condition property
workInstruction	String	The name of the work instruction.	Condition property
currentVersion	Boolean	If true, then this work instruction is current.	Condition property

Name	Data Type	Properties	Note
version	String	The version of the work instruction.	Condition property
descriptions	String	The description of the work instruction.	
status	String	The status of the work instruction. • NEW • RELEASABLE • OBSOLETE • HOLD	
erpWorkInstruction	Boolean	Indicates whether this work instruction was sent from ERP.	
trackViewing	Boolean	If true, then the viewing of this work instruction is tracked for each user.	
modifiedDateTime	number	The time when the work instruction was modified.	
CustomValue	Structure		
CustomValue[i]key	String	Custom field key name.	
CustomValue[i]value	String	Custom field value.	
WorkInstructionAttachedPoint	Structure		
WorkInstructionAttachedPoint[i]ref	String	The unique identifier of the work instruction attached point.	
WorkInstructionAttachedPoint[i]resource	String	A resource to which the work instruction is attached.	
WorkInstructionAttachedPoint[i]routingStepId	String	A routing step to which the work instruction is attached.	
WorkInstructionAttachedPoint[i]sequence	number	The sequence of the attachment point.	
WorkInstructionAttachedPoint[i]shopOrder	String	A shop order to which the work instruction is attached.	
WorkInstructionAttachedPoint[i]workCenter	String	A work center to which the work instruction is attached.	
WorkInstructionAttachedPoint[i]headerMaterial	Boolean	Defines if a material is a header material.	
WorkInstructionElement	Structure		
WorkInstructionElement[i]ref	String	The unique identifier of the work instruction element.	
WorkInstructionElement[i]fileName	String	The file name of the element.	
WorkInstructionElement[i]fileExternalUrl	String	The URL of the file of the element.	
WorkInstructionElement[i]mimeType	String	The MIME type of the file of the element.	
WorkInstructionElement[i]sequence	Long	The sequence of the element.	

Name	Data Type	Properties	Note
`WorkInstructionElement[i]text`	String	The text content of the element.	
`WorkInstructionElement[i]description`	String	The description of the element.	
`WorkInstructionElement[i]type`	String	The type of the element. • `TEXT` • `FILE` • `URL` • `HEADER_TEXT` • `LOCAL_FILE`	
`WorkInstructionElement[i]url`	String	The URL of the element.	
`WorkInstructionElement[i]newWindow`	Boolean	Defines if a work instruction can be opened in a new window.	
`WorkInstructionElement[i]erpFilename`	String	The ERP file name.	
`WorkInstructionElement[i]storageType`	String	The storage type of the element. • `AMAZON` • `GOOGLE` • `AZURE`	
`Material`	Structure		
`Material.ref`	String	The unique identifier of the material.	
`Material.plant`	String	The plant the material belongs to.	
`Material.material`	String	The name of the material.	
`Material.version`	String	The version of the material.	
`Operation`	Structure		
`Operation.ref`	String	The unique identifier of the operation.	
`Operation.plant`	String	The plant the operation activity belongs to.	
`Operation.operation`	String	The name of the operation.	
`Operation.version`	String	The version of the operation.	
`Routing`	Structure		
`Routing.ref`	String	The unique identifier of the routing.	
`Routing.plant`	String	The plant for which the routing is defined.	
`Routing.routing`	String	The name of the routing.	
`Routing.version`	String	The version of the routing.	

Name	Data Type	Properties	Note
`Routing.type`	String	The type of the routing. • `PRODUCTION` • `NC` • `SPECIAL` • `DISPOSITION` • `SFC` • `SHOP_ORDER` • CONFIGURABLE • PRODUCTION_RECIPE • SHOPORDER_SPECIFIC_RECIPE	

Business Service List

Get an overview of business services available in the business rule page of the **Manage Automatic Triggers** app.

Batch Creation

Batch creation triggered via public API.

Trigger point: Before Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where the batch is created	Condition property
`batchNumber`	String	The batch number	
`material`	String	The material to which the batch is assigned	Condition property
`source`	String	The source of the batch number. It indicates whether the batch number is generated by SAP Digital Manufacturing or by an integrated ERP system.	Condition property

Complete Operation Activity

You can complete an operation activity of an SFC via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where the complete operation activity is performed.	Condition property
`order`	String	The order of the SFC	Condition property
`sfc`	String	The shop floor control number for which the start operation is performed at an operation activity.	Condition property
`operationActivity`	String	The unique identifier of the operation activity	Condition property
`workCenter`	String	The work center associated with the operation activity	Condition property
`actualCompleteDateTime`	String	The UTC time at which the SFC is completed	Condition property
`finalConfirmation`	Boolean	If set to true, it indicates that the current operation activity is confirmed, and no further confirmation can be posted against it.	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`plant`	String	The plant name	Condition property
`order`	String	The order of the SFC	Condition property
`sfc`	String	The shop floor control number	Condition property
`operationActivity`	String	The phase of an operation activity	Condition property
`workCenter`	String	The work center name	Condition property
`actualCompleteDateTime`	String	The date and time at which an operation activity has started.	Condition property
`finalConfirmation`	String		Condition property

Complete Phase

You can complete the**Charge** via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

i Note

The use of `Complete Phase` API is no longer recommended as it has been deprecated. Instead, please use this alternative `Complete Operation Activity` API to perform the desired operation. More information on APIs that perform production activities for a production order, see Process Manufacturing Execution.

Request Parameter

Name	Data Type	Description	Note
plant	String	The plant where the complete operation is performed.	Condition property
order	String	The order of the charge	Condition property
charge	String	The unique identifier number of the charge for which the complete operation is performed at an operation activity.	Condition property
phase	String	The unique identifier of the operation activity	Condition property
workCenter	String	The work center associated with the charge	Condition property
actualStartDateTime	String	The UTC time at which the charge phase is started	Condition property
finalConfirmation	Boolean	If set to true, it indicates that the current operation activity is confirmed, and no further confirmation can be posted against it.	Condition property

Response Parameter

200

Name	Data Type	Description	Note
plant	String	The plant name	Condition property
order	String	The order of the charge	Condition property
charge	String	The shop floor control number	Condition property
phase	String	The phase of an operation activity	Condition property
workCenter	String	The work center name	Condition property
actualCompleteDateTime	String	The date and time at which the phase has completed.	Condition property
finalConfirmation	Boolean		Condition property

Final Quantity Confirmation on Completed Operation Activity

Use the API to post final quantity confirmations without updating the quantity for a given SFC.

Request Parameter

Trigger point: Before Service Execution, Error During Service Execution, After Service Execution

Name	Data Type	Description	Note
plant	String	The plant where the quantity confirmation is posted	Condition property

Name	Data Type	Description	Note
`shopOrder`	String	The order that the quantity confirmation is posted against. It can be a production order or a process order.	Condition property
`sfc`	String	The shop floor control that the quantity confirmation is posted against	Condition property
`operationActivity`	String	The operation activity of the quantity confirmation	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`transactionId`	String	The transaction ID of the quantity confirmation	

Goods Issue

Performs goods issue for a BOM or non-BOM component.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where goods issue is performed	Condition property
`order`	String	The order at which goods issue is performed	Condition property
`charge`	String	Charge of the order at which goods issue is performed	Condition property
`phase`	String	The phase at which goods issue is performed	Condition property
`workCenter`	String	Work center at which goods issue is performed	Condition property
`component`	Object / String	Request object for component • **`material`**: Request object for material • `sequence`: Sequence of the component	
`isBomComponent`	Boolean	BOM component or non-BOM component	

Name	Data Type	Description	Note
`bom`	String	Request object for BOM • `bom`: The name of the BOM • `version`: The version of the BOM	
`inventoryId`	String	Inventory ID for goods issue	
`quantity`	Number	Quantity for goods issue	
`unitOfMeasure`	String	Unit of measure of the quantity	
`postedBy`	String	ID of the user who posted the goods issue	
`postingDateTime`	String	Goods issue posting date and time	
`comments`	String	Comments when posting goods issue	
`handlingUnitNumber`	String	Handling unit number for goods issue	

Response Parameter

200

Name	Data Type	Description	Note
`order`	String	The order at which goods issue is performed	
`material`	String	Material for which goods issue is performed	
`phase`	String	Phase at which goods issue is performed	
`totalConsumedQuantity`	Number	Total consumed quantity	
`unitOfMeasure`	String	Unit of measure of the consumed quantity	

Goods Receipt

Goods receipts to EWM triggered in Goods Receipt plugin, or via public API.

Trigger point: Before Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where the goods receipt is performed	Condition property
`shopOrder`	String	The order for which the goods receipt is performed	Condition property
`sfc`	String	The SFC for which the goods receipt is performed	Condition property

Name	Data Type	Description	Note
material	String	The material name of the product posted in the goods receipt. The product can be finished goods, co-products or by-products	Condition property
postedBy	String	The user who posted the goods receipt	Condition property
handlingUnitNumber	String	The handling unit number of the product. It's applicable only to goods receipts to Extended Warehouse Management	Condition property
batchNumber	String	The batch number of the product	Condition property
quantity	Structure	Indicates how many products are posted in the goods receipt	
quantity.value	Double	Indicates how many products are posted in the goods receipt	
quantity.unitOfMeasure	Structure	The unit of measure of the product	
quantity.unitOfMeasure.uom	String	The commercial unit of measure of the product	
quantity.unitOfMeasure.internalUom	String	SAP internal unit of measure of the product	
quantity.unitOfMeasure.shortText	String	The short text of the commercial unit of measure	
quantity.unitOfMeasure.longText	String	The long text of the commercial unit of measure	
comments	String	The comments you want to add to the goods receipt posting	

Load or Unload Tool

Load or unload a tool using the Tool Loading plugin or the public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
plant	String	The plant where the tool loading is performed	Condition property
toolNumber	String	The tool number of the tool that's to be loaded or unloaded	

Name	Data Type	Description	Note
`location`	String	If the tool is to be loaded, it indicates the target location of the tool after it's loaded. If the tool is to be unloaded, it indicates the location from which the tool is unloaded.	
`locationType`	String	Indicates the type of the location. The location can only be of the type Resource.	
`flag`	String	Indicates whether the tool is to be loaded or unloaded. It can be one of the following types: LOAD, UNLOAD	

Response Parameter

Name	Data Type	Description	Note
`toolNumber`	String	The tool number of the tool that has been loaded or unloaded	

Log Nonconformance

You can log nonconformances against SFCs by using plugins in the Production Operator Dashboard (POD) or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
plant	String	The plant where a nonconformance is logged against an SFC.	Condition property
sfcs	String	SFCs against which nonconformances are logged.	
resource	String	The resrouce where a nonconformance is logged against an SFC.	Condition property
processLotNumber	String	The process lot to log nonconformance codes for the SFCs belonging to it.	Condition property
workCenter	String	The work center where the nonconformance is logged against SFCs.	Condition property
code	String	The nonconformance code to log a nonconformance.	
routingStepId	String	The routing step where the nonconformance was logged.	
quantity	BigDecimal	The nonconformant SFC quantity.	

Response Parameter

200

Name	Data Type	Description	Note
ids:sfc	String	The SFC for which nonconfomances were logged.	The `ids` is a list of objects (SFC and the list of unique nonconformance IDs).
ids:ids	String	The list of the unique IDs for the logged nonconformances.	

Order Release

Release orders using the **Order Release** business services in the **Manage Automatic Triggers** app.

In the **Manage Automatic Triggers** app, you can choose between two business services to release orders:

- **Release Order (Deprecated)**

  Use this business service if you want to trigger the service interception (Before Service Execution, Error During Service Execution, After Service Execution) from the user interface of the **Manage Orders** app.

- **Release Order**

  Use this business service if you want to trigger the service interception (Before Service Execution, Error During Service Execution, After Service Execution) using the Order API (`order/v2/orders/release`). This business service is documented here.

Trigger point: Before Service Execution, Error During Service Execution, After Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	Plant	Condition property
`quantityToRelease`	Number	The quantity to be released. If you provide this field, then `targetSfcs` is not accepted. If you want to use `targetSfcs` to control the SFCs to be released, this field must be removed or must be empty.	Condition property
`order`	String	The order to be released	Condition property
`ignoreReleasableQuantity`	Boolean	Set it to `true` if you want to release more than the available order quantity and ignore the order release status as well. If it's `false`, you can't release more than the order quantity, and you can't release the order if it's already in status `Released`.	Condition property

Name	Data Type	Description	Note
`ignoreMaterialLotSize`	Boolean	Set it to `true` if you want to ignore the lot size configuration on material. If it's `false`, the lot size configuration takes effect during the request validation and release process.	Condition property
`targetSFCs`	Array	This object represents a target SFC in release order request.	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`sfcs`	StructureArray		
`sfcs.[i].[identifier]`	String	The SFC identifier	Condition property
`sfcs.[I].[quantity]`	Number	The SFC quantity	Condition property

Order Release in Background

Order Release in background triggered in **Manage Orders** app or via public API.

Trigger point: Before Service Execution, Error During Service Execution, After Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	Plant	Condition property
`order`	String	The order to be released	Condition property
`quantityToRelease`	Number	The quantity to be released. If you provide this field, then `targetSfcs` is not accepted. If you want to use `targetSfcs` to control the SFCs to be released, this field must be removed or must be empty.	Condition property
`ignoreReleasableQuantity`	Boolean	Set it to `true` if you want to release more than the available order quantity and ignore the order release status as well. If it's `false`, you can't release more than the order quantity, and you can't release the order if it's already in status `Released`.	Condition property

Name	Data Type	Description	Note
`ignoreMaterialLotSize`	Boolean	Set it to `true` if you want to ignore the lot size configuration on material. If it's `false`, the lot size configuration takes effect during the request validation and release process.	Condition property
`targetSFCs`	StructureArray		
`targetSFCs.[i]`	Structure		
`targetSFCs.[i].identifier`	String	The identifier that the new SFC is generated with. If it's not provided, the system generates an identifier according to the numbering configuration.	
`targetSFCs.[i].quantity`	number	The quantity that the new SFC is generated with. If it's not provided, the system calculates the quantity by material lot size (if lot size is not ignored explicitly), or raises an exception accordingly.	
`targetSFCs.[i].unitOfMeasure`	String	Internal code of the unit of measure of the SFC to be released. It's converted to the material base unit of measure if an alternative unit is provided.	

Response Parameter

200

Name	Data Type	Description	Note
`transactionId`	String	The transaction ID	
`sfcs`	StructureArray		
`sfcs.[i]`	Structure		
`sfcs.[i].sfc`	String	The SFC identifier	
`sfcs.[i].quantity`	Number	The SFC quantity	

Packing Unit

Get an overview of business services related to packing units.

Add Content to a Packing Unit

You can add content to a packing unit in the POD by using the **Packing List** plugin, or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where content is added to a packing unit	Condition property
`packingUnit`	String	The packing unit number	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`plant`	String	The plant where content is added to the packing unit	Condition property
`number`	String	The packing unit number	Condition property
`status`	String	The status of the packing unit. A packing unit can have any of the following statuses: • `OPEN` • `CLOSED` • `UNLOADED`	Condition property

Change the Packing Unit Carrier Number

You can change the packing unit carrier number in the POD by using the **Packing List** plugin, or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The name of the plant	Condition property
`oldCarrierNumber`	String	The old number of the carrier	Condition property
`newCarrierNumber`	String	The new number of the carrier	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`plant`	String	The name of the plant	Condition property
`number`	String	The packing unit number	Condition property

Name	Data Type	Description	Note
`status`	String	The status of the packing unit. A packing unit can have any of the following statuses: • `OPEN` • `CLOSED` • `UNLOADED`	Condition property

Close or Open a Packing Unit

You can close or open a packing unit in the POD by using the **Packing List** plugin, or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The name of the plant	Condition property
`packingUnitNumber`	String	The packing unit number	Condition property
`status`	String	The packing unit status. A packing unit can have any of the following statuses: • `OPEN` • `CLOSED` • `UNLOADED`	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`plant`	String	The name of the plant	Condition property
`number`	String	The packing unit number	Condition property
`status`	String	The packing unit status. A packing unit can have any of the following statuses: • `OPEN` • `CLOSED` • `UNLOADED`	Condition property

Create a Packing Unit

You can create a packing unit in the POD by using the **Packing List** plugin, or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The name of the plant	Condition property
`number`	String	The packing unit number	Condition property
`status`	String	The packing unit status. A packing unit can have any of the following statuses: • `OPEN` • `CLOSED` • `UNLOADED`	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`plant`	String	The name of the plant	Condition property
`number`	String	The packing unit number	Condition property
`status`	String	The packing unit status. A packing unit can have any of the following statuses: • `OPEN` • `CLOSED` • `UNLOADED`	Condition property

Find a Packing Unit

You can find a packing unit in the POD by using the **Packing List** plugin, or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The name of the plant	Condition property
`number`	String	The packing unit number	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`plant`	String	The name of the plant	Condition property
`number`	String	The packing unit number	Condition property
`status`	String	The packing unit status. A packing unit can have any of the following statuses: • `OPEN` • `CLOSED` • `UNLOADED`	Condition property

Print Document

You can print document print templates via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`document:document`	String	The name of the document to be printed.	Condition property
`document:version`	String	The version of the document to be printed.	Optional property
`printer:printer`	String	The printer name.	Optional property

Response Parameter

200

Name	Data Type	Description	Note
`printTaskId`	String	The name of the print task.	

Production Change

Production changes to one or more SFCs

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where the production change is performed	Condition property

Name	Data Type	Description	Note
sfcs	StringArray	The list of shop floor controls (SFCs) to which the production change is performed	
routing	Structure		
routing.routing	String	The new routing assigned to the SFC	
routing.routingType	String	The type of the new routing. It can be any of the following types: PRODUCTION, SHOPORDER_SPECIFIC.	
routing.version	String	The version of the new routing	
routingStepId	String	The routing step ID of the new routing that's assigned to the specified SFCs. It determines from which routing step the SFC should be started.	
bom	Structure		
bom.bom	String	The new bill of materials (BOM) assigned to the SFC	
bom.bomType	String	The type of the new BOM. It can be any of the following types: USERBOM, SHOPORDERBOM, SFCBOM.	
bom.version	String	The version of the new BOM	
resource	String	The resource assigned to the specified SFCs	
operatorID	String	The operator who is allowed to execute the SFCs after the production change is performed	

Response Parameter

Name	Data Type	Description	Note
success	StringArray	The list of SFCs to which the production change is performed	
failed	StringArray	The list of SFCs to which the production change failed to be performed	

Quantity Confirmation

Quantity Confirmation triggered via public API to post yield and scrap quantities for an SFC.

Request Parameter

Trigger point: Before Service Execution, Error During Service Execution, After Service Execution

Name	Data Type	Description	Note
`plant`	String	The plant where the quantity confirmation is posted	Condition property
`shopOrder`	String	The order that the quantity confirmation is posted against. It can be a production order or a process order.	Condition property
`sfc`	String	The shop floor control that the quantity confirmation is posted against	Condition property
`operationActivity`	String	The operation activity of the quantity confirmation	Condition property
`resource`	String	The resource for which the quantity confirmation is posted	
`workCenter`	String	The work center for which the quantity confirmation is posted	
`yieldQuantity`	number	The yield quantity for the SFC	
`yieldQuantityUnit`	String	The base unit of measure for the yield quantity	
`yieldQuantityIsoUnit`	String	The ISO unit of measure for the yield quantity	
`scrapQuantity`	number	The scrap quantity for the SFC	
`reasonCodeKey`	String	The resource reason code key. A resource reason code may optionally be provided with the SFC scrap. The reason code will be used in reports and synchronized to the integrated ERP system.	
`scrapQuantityUnit`	String	The base unit of measure for the scrap quantity	
`scrapQuantityIsoUnit`	String	The ISO unit of measure for the scrap quantity	
`postedBy`	String	Specify the person who posted the quantity confirmation. It can be the user ID, user name or user email address, etc.	
`batchNumber`	String	The batch number that is used in the quantity confirmation	

Name	Data Type	Description	Note
storageLocation	String	The storage location for the quantity confirmation	
postingDateTime	String	The posting date time (in UTC) when the quantity confirmation is performed.	
finalConfirmation	Boolean	If set to true, it indicates that the current operation activity is confirmed, and no further quantity confirmation can be posted against it.	
checkSchedulingAndOeeRelevant	Boolean	This flag is set to true by default, which means that a check is performed whether the current resource is relevant for OEE and scheduling. If it's not OEE and scheduling relevant, the quantity confirmation will be rejected.	

Response Parameter

200

Name	Data Type	Description	Note
transactionId	String	The transaction ID of the quantity confirmation	

Recipe

Get an overview of business services related to recipes.

Create a Recipe

You can create a recipe by using the **Manage Routings/Recipes** app, or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
plant	String	The plant this recipe is assigned to	Condition property
recipe	String	The recipe name	Condition property
version	String	The recipe version	Condition property

Name	Data Type	Description	Note
type	String	The recipe type. It may be any of the following types: • PRODUCTION • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABLE • PRODUCTION_RECIPE • SHOPORDER_SPECIFIC_RECIPE	Condition property
status	String	The status of the recipe. A recipe may have one of the following statuses: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOLATION • HOLD_SPC_WARNING	Condition property
currentVersion	Boolean	If true, the version of the recipe is current.	
description	String	The description of the recipe	
relaxedFlow	Boolean	Indicates if the recipe has a relaxed flow.	
entryPhaseId	String	The entry recipe phase for the recipe	
phases	StructureArray		
phases.[i]	Structure	The recipe phase	
phases.[i].phaseId	String	The step ID of the recipe phase	
phases.[i].description	String	The description of the recipe phase	
phases.[i].entry	Boolean	If true, this is an entry recipe phase.	
phases.[i].recipeOperation	StructureArray		
phases.[i].recipeOperation.[i]	Structure		
phases.[i].recipeOperation.[i].phaseType	String	The type of recipe operation phase	

Name	Data Type	Description	Note
phases.[i].recipeOperation.[i].maxLoop	Double	Limits how many times an SFC may be restarted. If 0, then there is no limit.	
phases.[i].recipeOperation.[i].operationActivity	StructureArray		
phases.[i].recipeOperation.[i].operationActivity.[i]	Structure	The operation to be performed at a routing step	
phases.[i].recipeOperation.[i].operationActivity.[i].operationActivity	String	The name of the operation	
phases.[i].recipeOperation.[i].operationActivity.[i].version	String	The version of the operation	
phases.[i].recipeOperation.[i].baseQuantity	StructureArray		
phases.[i].recipeOperation.[i].baseQuantity.[i]	Structure	The base quantity of the recipe operation step	
phases.[i].recipeOperation.[i].baseQuantity.[i].value	Double	The value of the quantity	
phases.[i].recipeOperation.[i].weighRelevant	Boolean	Indicates if the phase is weight relevant.	
phases.[i].recipeOperation.[i].automaticGoodsReceipt	Boolean	Indicates if the phase is configured for an automatic goods receipt.	
phases.[i].recipeOperation.[i].customValues	StructureArray		
phases.[i].recipeOperation.[i].customValues.[i]	Structure	Custom data for the recipe operation phase	
phases.[i].recipeOperation.[i].customValues.[i]	String	The attribute name	
phases.[i].recipeOperation.[i].customValues.[i].attribute	String	The value of the attribute	
phases.[i].scrapPhaseId	String	The recipe scrap phase	
phases.[i].returnPhaseId	String	The recipe return phase	
phases.[i].holdPhaseId	String	The recipe hold phase	
phases.[i].donePhaseId	String	The recipe done phase	
phases.[i].workCenter	String	The work center for the recipe phase	
phases.[i].reportingStep	String	Indicates how this step is known in the external system.	
phases.[i].productionProcessId	String	The ID of the production process	
phases.[i].productionProcessDefKey	String	The production process definition key	
phases.[i].nextPhaseList	StructureArray		

Name	Data Type	Description	Note
phases.[i].nextPhaseList.[i]	Structure	The list of next possible recipe phases	
phases.[i].nextPhaseList.[i].items	String	The list of next possible recipe phases	
phases.[i].recipeLink	StructureArray		
phases.[i].recipeLink.[i]	Structure	The routing link associated with the routing step	
phases.[i].recipeLink.[i].phaseId	String	The recipe phase the link refers to	
phases.[i].recipeLink.[i].recipe	StructureArray		
phases.[i].recipeLink.[i].recipe.[i]	Structure	The recipe the link refers to	
phases.[i].recipeLink.[i].recipe.[i].plant	String	The plant this recipe is assigned to	
phases.[i].recipeLink.[i].recipe.[i].recipe	String	The name of the recipe	
phases.[i].recipeLink.[i].version	String	The recipe version	
quantityValidation	Boolean	Indicates if quantity validation is required during production.	
automaticGoodsReceipt	Boolean	Indicates if the recipe is configured for an automatic goods receipt.	
customValues	StructureArray	Recipe custom data	
customValues.[i]	Structure		
customValues.[i].attribute	String	The attribute name	
customValues.[i].value	String	The value of the attribute	
operationGroups	StructureArray	The operation groups of the recipe	
operationGroups.[i]	Structure	The operation groups of the recipe	
operationGroups.[i].recipeOperationGroup	String	The name of the recipe operation group	
operationGroups.[i].description	String	The description of the recipe operation group	
operationGroups.[i].recipeOperationGroupSteps	StructureArray	The steps of the recipe operation group	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i]	Structure	The recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].phaseId	String	The step ID of a recipe phase	

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].description	String	The description of a recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].entry	Boolean	If true, this is an entry recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i]	Structure	The operation to be performed at the recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].phaseType	String	The type of recipe operation phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].maxLoop	Double	Limits how many times an SFC may be restarted. If 0, then there is no limit.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity.[i]	Structure	The operation to be performed at the routing step	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity.[i].operationActivity	String	The name of the operation	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity.[i].version	String	The version of the operation	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].baseQuantity	Structure	The base quantity of the recipe operation step	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].baseQuantity.[i].value	Double	The value of the quantity	

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].weighRelevant	Boolean	Indicates if the phase is weight relevant	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].automaticGoodsReceipt	Boolean	Indicates if the phase is configured for an automatic goods receipt.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues.[i]	Structure	Custom data for the recipe operation phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues.[i].attribute	String	The attribute name	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues.[i].value	String	The value of the attribute	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].scrapPhaseId	String	The recipe scrap phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].returnPhaseId	String	The recipe return phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].holdPhaseId	String	The recipe hold phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].donePhaseId	String	The routing done step	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].workCenter	String	The work center for the recipe phase	

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].reportingStep	String	Indicates how this step is known in the external system.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].productionProcessId	String	The ID of the production process	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].productionProcessDefKey	String	The production process definition key	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].nextPhaseList	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].nextPhaseList.[i]	Structure	The list of the next possible recipe phases	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].nextPhaseList.[i].items	String	The list of the next possible recipe phases	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i]	Structure		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].phaseId	String	The recipe phase the link refers to	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i]	Structure		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i].plant	String	The plant this recipe is assigned to	

Name	Data Type	Description	Note
`operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i].recipe`	String	The name of the recipe	
`operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i].version`	String	The recipe version	

Response Parameter

200

Name	Data Type	Description	Note
`plant`	String	The plant this recipe is assigned to	Condition property
`recipe`	String	The name of the recipe	Condition property
`version`	String	The recipe version	Condition property
`type`	String	Indicates the type of the recipe. There can be the following types of recipes: • `PRODUCTION` • `NC` • `SPECIAL` • `DISPOSITION` • `SFC` • `SHOP_ORDER` • `CONFIGURABLE` • `PRODUCTION_RECIPE` • SHOPORDER_SPECIFIC_RECIPE	Condition property
`status`	String	The status of the recipe. There can be any of the following types of statuses: • `NEW` • `RELEASABLE` • `FROZEN` • `OBSOLETE` • `HOLD` • `HOLD_YIELD_RATE` • `HOLD_CONSEC_NC` • `HOLD_SPC_VIOLATION` • `HOLD_SPC_WARNING`	Condition property

Name	Data Type	Description	Note
currentVersion	Boolean	If true, the version of the recipe is current	
description	String	The description of the recipe	
entryPhaseId	String	The entry recipe phase for the recipe	
phases	StructureArray		
phases.[i]	Structure	Recipe phases in the recipe	
phases.[i].phaseId	String	The step ID of the recipe phase	
phases.[i].description	String	The description of the recipe phase	
phases.[i].entry	Boolean	If true, this is an entry recipe phase	
phases.[i].recipeOperation	StructureArray		
phases.[i].recipeOperation.[i]	Structure		
phases.[i].recipeOperation.[i].phaseType	String	The type of recipe operation phase	
phases.[i].recipeOperation.[i].maxLoop	Double	Limits how many times an SFC may be restarted. If 0, then there is no limit.	
phases.[i].recipeOperation.[i].operationActivity	StructureArray		
phases.[i].recipeOperation.[i].operationActivity.[i]	Structure	The operation to be performed at the routing step	
phases.[i].recipeOperation.[i].operationActivity.[i].operationActivity	String	The name of the operation	
phases.[i].recipeOperation.[i].operationActivity.[i].version	String	The version of the operation	
phases.[i].recipeOperation.[i].baseQuantity	StructureArray		
phases.[i].recipeOperation.[i].baseQuantity.[i]	Structure	The base quantity of the recipe operation step	
phases.[i].recipeOperation.[i].baseQuantity.[i].value	Double	The value of the quantity	
phases.[i].recipeOperation.[i].weighRelevant	Boolean	Indicates if the phase is weight relevant.	
phases.[i].recipeOperation.[i].automaticGoodsReceipt	Boolean	Indicates if the phase is configured for an automatic goods receipt	
phases.[i].recipeOperation.[i].customValues	StructureArray	Recipe custom data	
phases.[i].recipeOperation.[i].customValues.[i]	Structure	Custom data for the recipe operation phase	

Name	Data Type	Description	Note
phases.[i].recipeOperation.[i].customValues.[i].attribute	String	The attribute name	
phases.[i].recipeOperation.[i].customValues.[i].value	String	The value of the attribute	
phases.[i].scrapPhaseId	String	The recipe scrap phase	
phases.[i].returnPhaseId	String	The recipe return phase	
phases.[i].holdPhaseId	String	The recipe hold phase	
phases.[i].donePhaseId	String	The recipe done phase	
phases.[i].workCenter	String	The work center for the recipe phase	
phases.[i].reportingStep	String	Indicates how the step is known in the external system	
phases.[i].productionProcessId	String	The production process ID	
phases.[i].productionProcessDefKey	String	The production process definition key	
phases.[i].nextPhaseList	StructureArray		
phases.[i].nextPhaseList.[i]	Structure	The list of next possible recipe phases	
phases.[i].nextPhaseList.[i].items	String	The list of next possible recipe phases	
phases.[i].recipeLink	StructureArray		
phases.[i].recipeLink.[i]	Structure	The routing link associated with the routing step	
phases.[i].recipeLink.[i].phaseId	String	The recipe phase the link refers to	
phases.[i].recipeLink.[i].recipe	StructureArray		
phases.[i].recipeLink.[i].recipe.[i]	Structure	The recipe the link refers to	
phases.[i].recipeLink.[i].recipe.[i].plant	String	The plant this recipe is assigned to	
phases.[i].recipeLink.[i].recipe.[i].recipe	String	Name of the recipe	
phases.[i].recipeLink.[i].recipe.[i].version	String	The recipe version	

Delete a Recipe

You can delete a recipe via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant this recipe is assigned to	Condition property
`recipe`	String	The name of the recipe	Condition property
`version`	String	The recipe version	Condition property
`type`	String	Indicates the type of the recipe. The recipe can be of the following type: • `PRODUCTION` • `SHOP_ORDER`	Condition property

Update a Recipe

You can update a recipe by using the **Manage Routings/Recipes** app, or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant this recipe is assigned to	Condition property
`recipe`	String	The name of the recipe	Condition property
`version`	String	The recipe version	Condition property
`type`	String	Indicates the type of the recipe. There are the following types of recipes: • `PRODUCTION` • `SHOP_ORDER`	Condition property
`phases`	StructureArray		
`phases.[i]`	Structure	Recipe phases in the recipe	
`phases.[i].phaseId`	String	The step ID of the recipe phase	
`phases.[i].productionProcessId`	String	The ID of the production process	
`phases.[i].productionProcessDefKey`	String	The Production Process Definition Key	
`phases.[i].recipeOperation`	StructureArray		

Name	Data Type	Description	Note
phases.[i].recipeOperation.[i]	Structure	The operation to be performed at the recipe phase	
phases.[i].recipeOperation.[i].weighRelevant	Boolean	Indicates if the phase is weight relevant	
quantityValidation	Boolean		

Response Parameter

200

Name	Data Type	Description	Note
plant	String	The plant this recipe is assigned to	Condition property
recipe	String	The name of the recipe	Condition property
version	String	The recipe version	Condition property
type	String	Indicates the type of the recipe. There are the following types of recipes: • PRODUCTION • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABLE • PRODUCTION_RECIPE • SHOPORDER_SPECIFIC_RECIPE	Condition property
status	String	The status of the recipe. Recipes can have the following statuses: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOLATION • HOLD_SPC_WARNING	Condition property
currentVersion	Boolean	If true, the version of the recipe is current	

Name	Data Type	Description	Note
description	String	The entry recipe phase for the recipe	
entryPhaseId	String	The entry recipe phase for the recipe	
phases	StructureArray		
phases.[i]	Structure	The recipe phase	
phases.[i].phaseId	String	The step ID of the recipe phase	
phases.[i].description	String	The description of the recipe phase	
phases.[i].entry	Boolean	If true, this is an entry recipe phase	
phases.[i].recipeOperation	StructureArray		
phases.[i].recipeOperation.[i]	Structure	The operation to be performed at the recipe phase	
phases.[i].recipeOperation.[i].phaseType	String	The type of recipe operation phase	
phases.[i].recipeOperation.[i].maxLoop	Double	Limits how many times an SFC may be restarted. If 0, then there is no limit.	
phases.[i].recipeOperation.[i].operationActivity	StructureArray		
phases.[i].recipeOperation.[i].operationActivity.[i]	Structure	The operation to be performed at the routing step	
phases.[i].recipeOperation.[i].operationActivity.[i].operationActivity	String	Name of the operation	
phases.[i].recipeOperation.[i].operationActivity.[i].version	String	Version of the operation	
phases.[i].recipeOperation.[i].baseQuantity	StructureArray		
phases.[i].recipeOperation.[i].baseQuantity.[i]	Structure	Base quantity of the recipe operation step	
phases.[i].recipeOperation.[i].baseQuantity.[i].value	Double	The value of the quantity	
phases.[i].recipeOperation.[i].weighRelevant	Boolean	Indicates if the phase is weight relevant	
phases.[i].recipeOperation.[i].automaticGoodsReceipt	Boolean	Indicates if the phase is configured for an automatic goods receipt.	
phases.[i].customValues	StructureArray		
phases.[i].customValues.[i]	Structure	Custom data for the recipe operation phase	
phases.[i].customValues.[i].attribute	String	The attribute name	
phases.[i].customValues.[i].value	String	The value of the attribute	

Name	Data Type	Description	Note
phases.[i].scrapPhaseId	String	The recipe scrap phase	
phases.[i].returnPhaseId	String	The recipe return phase	
phases.[i].holdPhaseId	String	The recipe hold phase	
phases.[i].donePhaseId	String	The routing done step	
phases.[i].workCenter	String	The work center for the recipe phase	
phases.[i].reportingStep	String	Indicates how this step is known in the external system	
phases.[i].productionProcessId	String	The ID of the production process	
phases.[i].productionProcessDefKey	String	The production process definition key	
phases.[i].nextPhaseList	StructureArray		
phases.[i].nextPhaseList.[i]	Structure	The list of next possible recipe phases	
phases.[i].nextPhaseList.[i].items	String	The list of next possible recipe phases	
phases.[i].recipeLink	StructureArray		
phases.[i].recipeLink.[i].	Structure	The recipe phase	
phases.[i].recipeLink.[i].phaseId	String	The recipe phase the link refers to	
phases.[i].recipeLink.[i].recipe	StructureArray		
phases.[i].recipeLink.[i].recipe.[i]	Structure	The recipe the link refers to	
phases.[i].recipeLink.[i].recipe.[i].plant	String	The plant this recipe is assigned to	
phases.[i].recipeLink.[i].recipe.[i].recipe	String	Name of the recipe	
phases.[i].recipeLink.[i].recipe.[i].version	String	The recipe version	
quantityValidation	Boolean	Indicates if quantity validation is required during production.	
automaticGoodsReceipt	Boolean	Indicates if the recipe is configured for an automatic goods receipt.	
customValues	StructureArray		
customValues.[i]	Structure	Recipe custom data	
customValues.[i].attribute	String	The attribute name	
customValues.[i].value	String	The value of the attribute	
operationGroups	StructureArray		
operationGroups.[i]	Structure	The operation groups of the recipe	

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroup	String	The name of the recipe operation group	
operationGroups.[i].description	String	The description of the recipe operation group	
operationGroups.[i].recipeOperationGroupSteps	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i]	Structure	The steps of the recipe operation group	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i]	Structure	The recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].phaseId	String	The step ID of the recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].description	String	The description of the recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].entry	Boolean	If true, this is an entry recipe phase.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i]	Structure	The operation to be performed at the recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].phaseType	String	The type of recipe operation phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].maxLoop	Double	Limits how many times an SFC may be restarted. If 0, then there is no limit.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity	StructureArray		

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity.[i]	Structure	The operation to be performed at the routing step.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity.[i].operationActivity	String	Name of the operation	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity.[i].version	String	Version of the operation	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].baseQuantity	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].baseQuantity.[i]	Structure	Base quantity of the recipe operation step	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].baseQuantity.[i].value	Double	The value of the quantity	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].weighRelevant	Boolean	Indicates if the phase is weight relevant.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].automaticGoodsReceipt	Boolean	Indicates if the phase is configured for an automatic goods receipt.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues.[i]	Structure	Custom data for the recipe operation phase	

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues.[i].attribute	String	The attribute name	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues.[i].value	String	The value of the attribute	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].scrapPhaseId	String	The recipe scrap phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].returnPhaseId	String	The recipe return phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].holdPhaseId	String	The recipe hold phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].donePhaseId	String	The routing done step	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].workCenter	String	The work center for the recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].reportingStep	String	Indicates how this step is known in the external system.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].productionProcessId	String	The ID of the production process	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].productionProcessDefKey	String	The Production Process Definition Key	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].nextPhaseList	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].nextPhaseList.[i]	Structure	The list of next possible recipe phases	

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].nextPhaseList.[i].items	String	The list of next possible recipe phases	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i]	Structure	The routing link associated with the routing step.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].phaseId	String	The recipe phase the link refers to.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i]	Structure	The recipe the link refers to.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i].plant	String	The plant this recipe is assigned to.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i].recipe	String	The name of the recipe	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i].version	String	The recipe version	

Update Recipe Details

You can update recipe details by using the **Manage Routings/Recipes** app, or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note

Name	Data Type	Description	Note
plant	String	The plant this recipe is assigned to	Condition property
recipe	String	The name of the recipe	Condition property
version	String	The recipe version	Condition property
type	String	Indicates the type of the recipe. There are the following types of recipes: • PRODUCTION • SHOP_ORDER	Condition property
phases	StructureArray		
phases.[i]	Structure	Recipe phases in the recipe	
phases.[i].phaseId	String	The step ID of the recipe phase	
phases.[i].productionProcessId	String	The ID of the production process	
phases.[i].productionProcessDefKey	String	The Production Process Definition Key	
phases.[i].recipeOperation	StructureArray		
phases.[i].recipeOperation.[i]	Structure	The operation to be performed at the recipe phase	
phases.[i].recipeOperation.[i].weighRelevant	Boolean	Indicates if the phase is weight relevant	
quantityValidation	Boolean	Indicates if Quantity Validation is required during production.	
relaxedFlow	Boolean	Indicates if the recipe has a relaxed flow.	

Response Parameter

200

Name	Data Type	Description	Note
plant	String	The plant this recipe is assigned to	Condition property
recipe	String	The name of the recipe	Condition property
version	String	The recipe version	Condition property

Name	Data Type	Description	Note
type	String	Indicates the type of the recipe. There are the following types of recipes: • PRODUCTION • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABLE • PRODUCTION_RECIPE • SHOPORDER_SPECIFIC_RECIPE	Condition property
status	String	The status of the recipe. Recipes can have the following statuses: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_RATE • HOLD_CONSEC_NC • HOLD_SPC_VIOLATION • HOLD_SPC_WARNING	Condition property
currentVersion	Boolean	If true, the version of the recipe is current	
relaxedFlow	Boolean	Indicates if the recipe has a relaxed flow.	
entryPhaseId	String	The entry recipe phase for the recipe	
phases	StructureArray		
phases.[i]	Structure	The recipe phase	
phases.[i].phaseId	String	The step ID of the recipe phase	
phases.[i].description	String	The description of the recipe phase	
phases.[i].entry	Boolean	If true, this is an entry recipe phase	
phases.[i].recipeOperation	StructureArray		
phases.[i].recipeOperation.[i]	Structure	The operation to be performed at the recipe phase	
phases.[i].recipeOperation.[i].phaseType	String	The type of recipe operation phase	

Name	Data Type	Description	Note
phases.[i].recipeOperation.[i].maxLoop	Double	Limits how many times an SFC may be restarted. If 0, then there is no limit.	
phases.[i].recipeOperation.[i].operationActivity	StructureArray		
phases.[i].recipeOperation.[i].operationActivity.[i]	Structure	The operation to be performed at the routing step	
phases.[i].recipeOperation.[i].operationActivity.[i].operationActivity	String	Name of the operation	
phases.[i].recipeOperation.[i].operationActivity.[i].version	String	Version of the operation	
phases.[i].recipeOperation.[i].baseQuantity	StructureArray		
phases.[i].recipeOperation.[i].baseQuantity.[i]	Structure	Base quantity of the recipe operation step	
phases.[i].recipeOperation.[i].baseQuantity.[i].value	Double	The value of the quantity	
phases.[i].recipeOperation.[i].weighRelevant	Boolean	Indicates if the phase is weight relevant	
phases.[i].recipeOperation.[i].automaticGoodsReceipt	Boolean	Indicates if the phase is configured for an automatic goods receipt.	
phases.[i].customValues	StructureArray		
phases.[i].customValues.[i]	Structure	Custom data for the recipe operation phase	
phases.[i].customValues.[i].attribute	String	The attribute name	
phases.[i].customValues.[i].value	String	The value of the attribute	
phases.[i].scrapPhaseId	String	The recipe scrap phase	
phases.[i].returnPhaseId	String	The recipe return phase	
phases.[i].holdPhaseId	String	The recipe hold phase	
phases.[i].donePhaseId	String	The routing done step	
phases.[i].workCenter	String	The work center for the recipe phase	
phases.[i].reportingStep	String	Indicates how this step is known in the external system	
phases.[i].controlKey	String	The name of the control key	
phases.[i].lastReportingPhase	Boolean	Indicates if it is the last reporting step	
phases.[i].rework	Boolean	Indicates if this phase is considered as rework	

Name	Data Type	Description	Note
phases.[i].queueDecisionType	String	This determines who decides where an SFC should be placed in queue when completing this phase.	
phases.[i].productionProcessId	String	The ID of the production process	
phases.[i].productionProcessDefKey	String	The production process definition key	
phases.[i].nextPhaseList	StructureArray		
phases.[i].nextPhaseList.[i]	Structure	The list of next possible recipe phases	
phases.[i].nextPhaseList.[i].items	String	The list of next possible recipe phases	
phases.[i].recipePhaseComponentList	StructureArray	The list of components to be assembled at the phase	
phases.[i].recipePhaseComponentList.[i]	Structure		
phases.[i].recipePhaseComponentList.[i].bomComponent	Object	The BOM component to assemble.	
phases.[i].recipePhaseComponentList.[i].sequence	Double	Controls the order of display and execution of the component.	
phases.[i].recipePhaseComponentList.[i].quantity	Double	The number of pieces of the component to be assembled at this step.	
phases.[i].recipePhaseComponentList.[i].toleranceUnder	Double	The upper bound tolerance value.	
phases.[i].recipePhaseComponentList.[i].toleranceOver	Double	The lower bound tolerance value.	
phases.[i].recipeLink	StructureArray		
phases.[i].recipeLink.[i].	Structure	The recipe phase	
phases.[i].recipeLink.[i].phaseId	String	The recipe phase the link refers to	
phases.[i].recipeLink.[i].recipe	StructureArray		
phases.[i].recipeLink.[i].recipe.[i]	Structure	The recipe the link refers to	
phases.[i].recipeLink.[i].recipe.[i].plant	String	The plant this recipe is assigned to	
phases.[i].recipeLink.[i].recipe.[i].recipe	String	Name of the recipe	

Name	Data Type	Description	Note
phases.[i].recipeLink.[i].recipe.[i].version	String	The recipe version	
quantityValidation	Boolean	Indicates if quantity validation is required during production.	
automaticGoodsReceipt	Boolean	Indicates if the recipe is configured for an automatic goods receipt.	
customValues	StructureArray		
customValues.[i]	Structure	Recipe custom data	
customValues.[i].attribute	String	The attribute name	
customValues.[i].value	String	The value of the attribute	
operationGroups	StructureArray		
operationGroups.[i]	Structure	The operation groups of the recipe	
operationGroups.[i].recipeOperationGroup	String	The name of the recipe operation group	
operationGroups.[i].description	String	The description of the recipe operation group	
operationGroups.[i].recipeOperationGroupSteps	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i]	Structure	The steps of the recipe operation group	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i]	Structure	The recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].phaseId	String	The step ID of the recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].description	String	The description of the recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].entry	Boolean	If true, this is an entry recipe phase.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation	StructureArray		

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i]	Structure	The operation to be performed at the recipe phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].phaseType	String	The type of recipe operation phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].maxLoop	Double	Limits how many times an SFC may be restarted. If 0, then there is no limit.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity.[i]	Structure	The operation to be performed at the routing step.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity.[i].operationActivity	String	Name of the operation	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].operationActivity.[i].version	String	Version of the operation	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].baseQuantity	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].baseQuantity.[i]	Structure	Base quantity of the recipe operation step	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].baseQuantity.[i].value	Double	The value of the quantity	

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].weighRelevant	Boolean	Indicates if the phase is weight relevant.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].automaticGoodsReceipt	Boolean	Indicates if the phase is configured for an automatic goods receipt.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues.[i]	Structure	Custom data for the recipe operation phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues.[i].attribute	String	The attribute name	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeOperation.[i].customValues.[i].value	String	The value of the attribute	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].scrapPhaseId	String	The recipe scrap phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].returnPhaseId	String	The recipe return phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].holdPhaseId	String	The recipe hold phase	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].donePhaseId	String	The routing done step	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].workCenter	String	The work center for the recipe phase	

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].reportingStep	String	Indicates how this step is known in the external system.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].controlKey	Array		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].controlKey.[i]	String	The name of the Control Key.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].lastReportingPhase	Boolean	Indicates if it is the last reporting step.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].rework	Boolean	Indicates if this phase is considered as rework.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].queueDecisionType	String	This determines who decides where an SFC should be placed in queue when completing this phase.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].productionProcessId	String	The ID of the production process	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].productionProcessDefKey	String	The Production Process Definition Key	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].nextPhaseList	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].nextPhaseList.[i]	Structure	The list of next possible recipe phases	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].nextPhaseList.[i].items	String	The list of next possible recipe phases	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i]recipePhaseComponentList	Array	The list of components to be assembled at the phase.	

Name	Data Type	Description	Note
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i]recipePhaseComponentList.[i].items	Object	The list of components to be assembled at the phase.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i]recipePhaseComponentList.[i].items.[i].bomComponent	Object	The BOM component to assemble.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i]recipePhaseComponentList.[i].items.[i].sequence	Double	Controls the order of display and execution of the component.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i]recipePhaseComponentList.[i].items.[i].quantity	Double	The number of pieces of the component to be assembled at this step	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i]recipePhaseComponentList.[i].items.[i].toleranceUnder	Double	The upper bound tolerance value.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i]recipePhaseComponentList.[i].items.[i].toleranceOver	Double	The lower bound tolerance value.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i]	Structure	The routing link associated with the routing step.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].phaseId	String	The recipe phase the link refers to.	
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe	StructureArray		
operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i]	Structure	The recipe the link refers to.	

Name	Data Type	Description	Note
`operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i].plant`	String	The plant this recipe is assigned to.	
`operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i].recipe`	String	The name of the recipe	
`operationGroups.[i].recipeOperationGroupSteps.[i].recipePhase.[i].recipeLink.[i].recipe.[i].version`	String	The recipe version	

Requests a Component Assembly

Requests assembly for a planned BOM component.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	Plant for assembly of a component	Condition property
`operationActivity`	String	Operation activity for assembly of a component	Condition property
`sfc`	String	Shop floor control for assembly of a component	Condition property
`component`	String	Component name for assembly of a component	Condition property
`componentVersion`	String	Component version for assembly of a component	
`quantity`	Number	Quantity for assembly of a component	
`resource`	String	Resource name for assembly of a component	Condition property
`sequence`	Integer	Sequence for assembly of a component	

`dataFields`	StructureArray / dataFields	List of assembly data fields for assembly of a component
		• `items`:
		  ◦ **`fieldName`**: Assembly field name
		  ◦ **`fieldValue`**: Assembly field value	

Response Parameter

200

Name	Data Type	Description	Note
`addComponentResponse`	String	Component assembled successfully	

Routing

Get an overview of business services related to routings.

Create a Routing

You can create a routing by using the **Manage Routings/Recipes** app, or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description
`plant`	String	The plant this recipe is a
`routing`	String	The name of the routing
`routingType`	String	The routing or recipe typ the following types: • `PRODUCTION` • `NC` • `SPECIAL` • `DISPOSITION` • `SFC` • `SHOP_ORDER` • `CONFIGURABLE` • `PRODUCTION_P` • `SHOPORDER_SP`
`version`	String	The routing version
`currentVersion`	Boolean	If true, the version of the

Name	Data Type	Description
status	String	The status of the routing have one of the following • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_F • HOLD_CONSEC_ • HOLD_SPC_VIC • HOLD_SPC_WAR
description	String	The description of the rc
relaxedFlow	Boolean	Indicates if the routing h
quantityValidation	Boolean	Indicates if quantity vali during production.
automaticGoodsReceipt	Boolean	Indicates if the routing is automatic goods receipt
routingSteps	StructureArray	
routingSteps.[i]	Structure	The routing steps in the
routingSteps.[i].stepId	String	The step ID of the routin
routingSteps.[i].sequence	Double	The sequence of the rou
routingSteps.[i].description	String	The description of the rc
routingSteps.[i].entry	Boolean	If true, this is an entry rc
routingSteps.[i].productionProcessId	String	The ID of the production
routingSteps.[i].productionProcessDefKey	String	The production process
routingSteps.[i].routingOperation	StructureArray	
routingSteps.[i].routingOperation.[i]	Structure	The operation to be perf routing step.
routingSteps.[i].routingOperation.[i].stepType	String	The type of routing oper
routingSteps.[i].routingOperation.[i].maxLoop	Double	Limits how many times a restarted. If 0, then there
routingSteps.[i].routingOperation.[i].operationActivity	StructureArray	
routingSteps.[i].routingOperation.[i].operationActivity.[i]	Structure	The operation to be perf step
routingSteps.[i].routingOperation.[i].operationActivity.[i].operationActivity	String	The name of the operati

Name	Data Type	Description
routingSteps.[i].routingOperation.[i].operationActivity.[i].version	String	The version of the opera
routingSteps.[i].routingOperation.[i].baseQuantity	StructureArray	
routingSteps.[i].routingOperation.[i].baseQuantity.[i]	Structure	The base quantity of the step
routingSteps.[i].routingOperation.[i].baseQuantity.[i].value	Double	The value of the quantity
routingSteps.[i].routingOperation.[i].weighRelevant	Boolean	Indicates if the operatio relevant.
routingSteps.[i].routingOperation.[i].automaticGoodsReceipt	Boolean	Indicates if the operatio an automatic goods rece
routingSteps.[i].routingOperation.[i].customValues	StructureArray	
routingSteps.[i].routingDoneStep.routingStep.stepId	String	The step ID of the routin
routingSteps.[i].routingOperation.[i].customValues.[i]	Structure	Custom data for the rou
routingSteps.[i].routingOperation.[i].customValues.[i]	String	The attribute name
routingSteps.[i].routingOperation.[i].customValues.[i].attribute	String	The value of the attribut
routingSteps.[i].routingStepGroup	StructureArray	
routingSteps.[i].routingStepGroup.[i]	Structure	The routing step group
routingSteps.[i].routingStepGroup.[i].routingStep	Structure Array	
routingSteps.[i].routingStepGroup.[i].routingStep.[i]	Structure	The routing step that is i
routingSteps.[i].routingStepGroup.[i].routingStep.[i].stepId	String	The step ID of the routin
routingSteps.[i].routingStepGroup.[i].routingStep.[i].routingStepGroupType	String	The type of routing step the following types of ro • ANY_ORDER_GR • SIMULTANEOUS
routingSteps.[i].routingStepGroup.[i].routingStep.[i].routingStepGroupStepList	StructureArray	
routingSteps.[i].routingStepGroup.[i].routingStep.[i].routingStepGroupStepList.[i]	Structure	List of routing step grou
routingSteps.[i].routingStepGroup.[i].routingStep.[i].routingStepGroupStepList.[i].routingStep	StructureArray	
routingSteps.[i].routingStepGroup.[i].routingStep.[i].routingStepGroupStepList.[i].routingStep.[i]	Structure	List of routing step grou
routingSteps.[i].routingStepGroup.[i].routingStep.[i].routingStepGroupStepList.[i].routingStep.[i].stepId	String	The step ID of the routin
routingSteps.[i].routingStepGroup.[i].routingStep.[i].routingStepGroupStepList.[i].routingStep.[i].sequence	Double	Controls the order the st within the group.
routingSteps.[i].routingStepGroup.[i].routingStep.[i].routingStepGroupStepList.[i].routingStep.[i].successors	String	A comma-delimited list successors for this parti member.

Name	Data Type	Description
routingSteps.[i].routingScrapStep	Structure	The routing scrap step
routingSteps.[i].routingScrapStep.routingStep	Structure	The router step that is ir
routingSteps.[i].routingScrapStep.routingStep.stepId	String	The step ID of the routin
routingSteps.[i].routingReturnStep	Structure	The routing return step
routingSteps.[i].routingReturnStep.routingStep	Structure	The router step that is ir
routingSteps.[i].routingReturnStep.routingStep.stepId	String	The step ID of the routin
routingSteps.[i].routingHoldStep	Structure	The routing hold step
routingSteps.[i].routingHoldStep.routingStep	Structure	The router step that is ir
routingSteps.[i].routingHoldStep.routingStep.stepId	String	The step ID of the routin
routingSteps.[i].routingDoneStep	Structure	The routing done step
routingSteps.[i].routingDoneStep.routingStep	Structure	The router step that is ir
routingSteps.[i].workCenter	Structure	The work center for the r
routingSteps.[i].workCenter.workCenter	String	The name of the work ce
routingSteps.[i].reportingStep	String	Indicates how this step i external system.
routingSteps.[i].controlKey	Structure	The control key for the r
routingSteps.[i].controlKey.controlKey	String	The name of the Control
routingSteps.[i].lastReportingStep	Boolean	Indicates if it is the last
routingSteps.[i].rework	Boolean	Indicates if this step is rework.
routingSteps.[i].queueDecisionType	String	This determines who de SFC should be placed ir completing this step. Th are possible: • COMPLETING_C • NEXT_OPERATC
routingSteps.[i].nextStepList	StructureArray	
routingSteps.[i].nextStepList.[i]	Structure	The list of next possible
routingSteps.[i].nextStepList.[i].items	String	The list of next possible
customValues	StructureArray	
customValues.[i]	Structure	Routing custom data
customValues.[i].attribute	String	The attribute name
customValues.[i].value	String	The value of the attribut
routingOperationGroups	StructureArray	
routingOperationGroups.[i]	Structure	The operation groups of

Name	Data Type	Description
routingOperationGroups.[i]		
routingOperationGroups.[i].routingOperationGroup	String	The name of the routing
routingOperationGroups.[i].description	String	The description of the rc group
routingOperationGroups.[i].routingOperationGroupSteps	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i]	Structure	The steps of the routing
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep	Structure	The routing step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.sequence	Double	The sequence of the rou
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.description	String	The description of the rc
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.entry	Boolean	If true, this is an entry rc
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.productionProcessId	String	The ID of the production
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.productionProcessDefKey	String	Production Process Defi
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation	Structure	The operation to be perf routing step.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.stepType	String	The type of routing oper
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.maxLoop	Double	Limits how many times a restarted. If 0, then there
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity	Structure	The operation to be perf routing step.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity.operationActivity	String	Name of the operation
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity.version	String	Version of the operation
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.baseQuantity	Structure	Base quantity of the rou
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.baseQuantity.value	Double	The value of the quantity
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.weighRelevant	Boolean	Indicates if operation is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.automaticGoodsReceipt	Boolean	Indicates if the operatio an automatic goods rece
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues	StructureArray	

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues.[i]	Structure	Custom data for the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues.[i].attribute	String	The attribute name
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues.[i].value	String	The value of the attribut
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup	Structure	The routing step group
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStep	Structure	The router step that is ir
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupType	String	The type of router step g the following types of ro • ANY_ORDER_GR • SIMULTANEOUS
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupTypeList	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupTypeList.[i]	Structure	List of routing step grou
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupTypeList.[i].routingStep	Structure	The router step that is ir
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupTypeList.[i].routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupTypeList.[i].routingStep.sequence	Double	Controls the order the st within the group.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupTypeList.[i].routingStep.successors	String	A comma-delimited list successors for this parti member.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingScrapStep	Structure	The routing scrap step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingScrapStep.routingStep	Structure	The router step that is ir
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingScrapStep.routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingReturnStep	Structure	The routing return step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingReturnStep.routingStep	Structure	The router step that is ir

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingReturnStep.routingStep	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingHoldStep	Structure	The routing hold step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingHoldStep.routingStep	Structure	The router step that is ir
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingHoldStep.routingStep	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingDoneStep	Structure	The routing done step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingDoneStep.routingStep	Structure	The router step that is ir
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingDoneStep.routingStep	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.workCenter	Structure	The work center for the r
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.workCenter.workCenter	String	The name of the work ce
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.reportingStep	String	Indicates how this step i external system.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.controlKey	Structure	The control key for the r
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.controlKey.controlKey	String	The name of the Control
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.lastReportingStep	Boolean	Indicates if it is the last
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.rework	Boolean	Indicates if this step is rework.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.queueDecisionType	String	This determines who de SFC should be placed in completing this step. Th are possible: • COMPLETING_C • NEXT_OPERATC
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.nextStepList	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.nextStepList.[i]	Structure	The list of next possible
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.nextStepList.[i].items	String	The list of next possible

Response Parameter

Name	Data Type	Description
plant	String	The plant this routing is
routing	String	The name of the routing
routingType	String	Indicates the type of the be the following types of • PRODUCTION • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABLE • PRODUCTION_F • SHOPORDER_SF
version	String	The routing version
currentVersion	Boolean	If true, the version of the
status	String	The status of the routing of the following types of • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_F • HOLD_CONSEC_ • HOLD_SPC_VIC • HOLD_SPC_WAF
description	String	The description of the ro
relaxedFlow	Boolean	Indicates if the routing is relaxed flow routing.
quantityValidation	Boolean	Indicates if Quantity Vali during production.
automaticGoodsReceipt	Boolean	Indicates if the Routing i automatic goods receipt

Name	Data Type	Description
routingSteps	StructureArray	
routingSteps.[i]	Structure	Routing steps in the rout
routingSteps.[i].stepId	String	The step ID of the routin
routingSteps.[i].sequence	Double	The sequence of the rou
routingSteps.[i].description	String	The description of the rc
routingSteps.[i].entry	Boolean	If true, this is an entry rc
routingSteps.[i].productionProcessId	String	The ID of the production
routingSteps.[i].productionProcessDefKey	String	Production Process Defi
routingSteps.[i].routingOperation	Structure	The operation to be perf routing step.
routingSteps.[i].routingOperation.stepId	String	The type of routing oper
routingSteps.[i].routingOperation.maxLoop	Double	Limits how many times restarted. If 0, then there
routingSteps.[i].routingStepGroup	Structure	The routing step group
routingSteps.[i].routingScrapStep	Structure	The routing scrap step
routingSteps.[i].routingScrapStep.routingStep	Structure	The router step that is ir
routingSteps.[i].routingScrapStep.routingStep.stepId	String	The step ID of the routin
routingSteps.[i].routingReturnStep	Structure	The routing return step
routingSteps.[i].routingReturnStep.routingStep	Structure	The router step that is ir
routingSteps.[i].routingReturnStep.routingStep.stepId	String	The step ID of the routin
routingSteps.[i].routingHoldStep	Structure	The routing hold step
routingSteps.[i].routingHoldStep.routingStep	Structure	The router step that is ir
routingSteps.[i].routingHoldStep.routingStep.stepId	String	The step ID of the routin
routingSteps.[i].routingDoneStep	Structure	The routing done step
routingSteps.[i].routingDoneStep.routingStep	Structure	The router step that is ir
routingSteps.[i].routingDoneStep.routingStep.stepId	String	The step ID of the routin
routingSteps.[i].workCenter	Structure	The work center for the r
routingSteps.[i].workCenter.workCenter	String	The name of the work ce
routingSteps.[i].reportingStep	String	Indicates how this step i external system.
routingSteps.[i].controlKey	Structure	The control key for the r
routingSteps.[i].controlKey.controlKey	String	The name of the Control
routingSteps.[i].lastReportingStep	Boolean	Indicates if it is the last
routingSteps.[i].rework	Boolean	Indicates if this step is rework.

Name	Data Type	Description
routingSteps.[i].queueDecisionType	String	This determines who de SFC should be placed ir completing this step. Th are possible: • COMPLETING_C • NEXT_OPERATC
routingSteps.[i].nextStepList	StructureArray	
routingSteps.[i].nextStepList.[i]	Structure	The list of next possible
routingSteps.[i].nextStepList.[i].items	String	The list of next possible
customValues	StructureArray	
customValues.[i]	Structure	Routing custom data
customValues.[i].attribute	String	The attribute name
customValues.[i].value	String	The value of the attribut
routingOperationGroups	StructureArray	
routingOperationGroups.[i]	Structure	The operation groups of
routingOperationGroups.[i].routingOperationGroup	String	The name of the routing
routingOperationGroups.[i].description	String	The description of the rc group
routingOperationGroups.[i].routingOperationGroupSteps	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i]	Structure	The steps of the routing
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep	Structure	The routing step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.sequence	Double	The sequence of the rou
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.description	String	The description of the rc
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.entry	Boolean	If true, this is an entry rc
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.productionProcessId	String	The ID of the production
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.productionProcessDefKey	String	Production Process Defi
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation	Structure	The operation to be perf routing step.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.stepType	String	The type of routing oper

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.maxLoop	Double	Limits how many times a restarted. If 0, then there
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity	Structure	The operation to be perf routing step.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity.operationActivity	String	Name of the operation
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity.version	String	The version of the opera
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.baseQuantity	Structure	Base quantity of the rou
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.baseQuantity.value	Double	The value of the quantity
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.weighRelevant	Boolean	Indicates if operation is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.automaticGoodsReceipt	Boolean	Indicates if the operatio an automatic goods rece
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues.[i]	Structure	Custom data for the rou
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues.[i].attribute	String	The attribute
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues.[i].value	String	The value of the attribut
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup	Structure	The routing step group
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStep	Structure	The router step that is ir
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupType	String	The type of router step g be the following types of groups: • ANY_ORDER_GR • SIMULTANEOUS
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i]	Structure	List of routing step grou
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].routingStep	Structure	The router step that is ir

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].sequence	Double	Controls the order the st within the group.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].successors	String	A comma-delimited list successors for this parti member.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingScrapStep	Structure	The routing scrap step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingScrapStep.routingStep	Structure	The router step that is ir
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingScrapStep.routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingReturnStep	Structure	The routing return step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingReturnStep.routingStep	Structure	The router step that is ir
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingReturnStep.routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingHoldStep	Structure	The routing hold step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingHoldStep.routingStep	Structure	The router step that is ir
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingHoldStep.routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingDoneStep	Structure	The routing done step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingDoneStep.routingStep	Structure	The router step that is ir
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingDoneStep.routingStep.stepId	String	The step ID of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.workCenter	Structure	The work center for the r
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.workCenter.workCenter	String	The name of the work ce
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.reportingStep	String	Indicates how this step i external system.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.controlKey	Structure	The control key for the r
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.controlKey.controlKey	String	The name of the Control

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.lastReportingStep	Boolean	Indicates if it is the last
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.rework	Boolean	Indicates if this step is c rework.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.queueDecisionType	String	This determines who de SFC should be placed in completing this step. Th are possible: • COMPLETING_C • NEXT_OPERATC
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.nextStepList	Structure	The list of next possible
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.nextStepList.items	String	The list of next possible

Delete a Routing

You can delete a routing via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
plant	String	The plant this routing is assigned to	Condition property
routing	String	The name of the routing	Condition property
routingType	String	Indicates the type of the routing. The routing can be of the following type: • PRODUCTION • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABLE • PRODUCTION_RECIPE • SHOPORDER_SPECIFC_RECIPE	Condition property
version	String	The routing version	Condition property

Update a Routing

You can update a routing by using the **Manage Routings/Recipes** app, or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description
`plant`	String	The plant this routing i
`routing`	String	The name of the routin
`routingType`	String	Indicates the type of th the following types of r • `NC` • `SPECIAL` • `DISPOSITION` • `SFC` • `SHOP_ORDER` • `CONFIGURABL` • `PRODUCTION_` • `SHOPORDER_S`
`version`	String	The routing version
`status`	String	The status of the routir statuses are possible: • `NEW` • `RELEASABLE` • `FROZEN` • `OBSOLETE` • `HOLD` • `HOLD_YIELD_` • `HOLD_CONSEC` • `HOLD_SPC_VI` • `HOLD_SPC_WA`
`description`	String	The description of the
`relaxedFlow`	Boolean	Indicates if the routing relaxed flow routing.

Name	Data Type	Description
quantityValidation	Boolean	Indicates if Quantity Va during production.
automaticGoodsReceipt	Boolean	Indicates if the Routing automatic goods receip
routingSteps	StructureArray	
routingSteps.[i]	Structure	Routing steps in the ro
routingSteps.[i].stepId	String	The step ID of the rout
routingSteps.[i].sequence	Double	The sequence of the ro
routingSteps.[i].description	String	The description of the
routingSteps.[i].entry	Boolean	If true, this is an entry
routingSteps.[i].productionProcessId	String	The ID of the productio
routingSteps.[i].productionProcessDefKey	String	Production Process De
routingSteps.[i].routingOperation	Structure	The operation to be pe routing step.
routingSteps.[i].routingOperation.stepType	String	The type of routing ope
routingSteps.[i].routingOperation.maxLoop	Double	Limits how many times restarted. If 0, then the
routingSteps.[i].routingOperation.operationActivity	Structure	The operation to be pe routing step.
routingSteps.[i].routingOperation.operationActivity.operationActivity	String	Name of the operation
routingSteps.[i].routingOperation.operationActivity.version	String	Version of the operatio
routingSteps.[i].routingOperation.baseQuantity	Structure	Base quantity of the ro
routingSteps.[i].routingOperation.baseQuantity.value	Double	The value of the quanti
routingSteps.[i].routingOperation.weighRelevant	Boolean	Indicates if operation i
routingSteps.[i].routingOperation.automaticGoodsReceipt	Boolean	Indicates if the operati an automatic goods rec
routingSteps.[i].routingOperation.customValues	StructureArray	
routingSteps.[i].routingOperation.customValues.[i]	Structure	Custom data for the ro
routingSteps.[i].routingOperation.customValues.[i].attribute	String	The attribute name
routingSteps.[i].routingOperation.customValues.[i].value	String	The value of the attribu
routingSteps.[i].routingStepGroup	Structure	The routing step group
routingSteps.[i].routingStepGroup.routingStep	Structure	The router step that is
routingSteps.[i].routingStepGroup.routingStep.stepId	String	The step ID of the rout

Name	Data Type	Description
routingSteps.[i].routingStepGroup.routingStepGroupType	String	The type of router step following types are pos • ANY_ORDER_C • SIMULTANEOU
routingSteps.[i].routingStepGroup.routingStepGroupStepList	StructureArray	
routingSteps.[i].routingStepGroup.routingStepGroupStepList.[i]	Structure	List of routing step gro
routingSteps.[i].routingStepGroup.routingStepGroupStepList.[i].routingStep	Structure	The router step that is
routingSteps.[i].routingStepGroup.routingStepGroupStepList.[i].routingStep.stepId	String	The step ID of the rout
routingSteps.[i].routingStepGroup.routingStepGroupStepList.[i].sequence	Double	Controls the order the within the group.
routingSteps.[i].routingStepGroup.routingStepGroupStepList.[i].successors	String	A comma-delimited lis successors for this par member.
routingSteps.[i].routingScrapStep	Structure	The routing scrap step
routingSteps.[i].routingScrapStep.routingStep	Structure	The router step that is
routingSteps.[i].routingScrapStep.routingStep.stepId	String	The step ID of the rout
routingSteps.[i].routingHoldStep	Structure	The routing hold step
routingSteps.[i].routingHoldStep.routingStep	Structure	The router step that is
routingSteps.[i].routingHoldStep.routingStep.stepId	String	The step ID of the rout
routingSteps.[i].routingDoneStep	Structure	The routing done step
routingSteps.[i].routingDoneStep.routingStep	Structure	The router step that is
routingSteps.[i].routingDoneStep.routingStep.stepId	String	The step ID of the rout
routingSteps.[i].workCenter	Structure	The work center for the
routingSteps.[i].workCenter.workCenter	String	The name of the work c
routingSteps.[i].reportingStep	String	Indicates how this step external system.
routingSteps.[i].controlKey	Structure	The control key for the
routingSteps.[i].controlKey.controlKey	String	The name of the Contro
routingSteps.[i].lastReportingStep	Boolean	Indicates if it is the las
routingSteps.[i].rework	Boolean	Indicates if this step is rework.

Name	Data Type	Description
routingSteps.[i].queueDecisionType	String	This determines who d SFC should be placed completing this step. T are possible: • COMPLETING_ • NEXT_OPERAT
routingSteps.[i].nextStepList	Structure	The list of next possib
routingSteps.[i].nextStepList.items	String	The list of next possib
customValues	StructureArray	
customValues.[i]	Structure	Routing custom data
customValues.[i].attribute	String	The attribute name
customValues.[i].value	String	The value of the attribu
routingOperationGroups	StructureArray	
routingOperationGroups.[i]	Structure	The operation groups c
routingOperationGroups.[i].routingOperationGroup	String	The name of the routin
routingOperationGroups.[i].description	String	The description of the group
routingOperationGroups.[i].routingOperationGroupSteps	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i]	Structure	The steps of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep	Structure	The routing step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.sequence	Double	The sequence of the ro
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.description	String	The description of the
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.entry	Boolean	If true, this is an entry
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.productionProcessId	String	The ID of the productio
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.productionProcessDefKey	String	Production Process De
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation	Structure	The operation to be pe routing step.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.stepType	String	The type of routing ope
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.maxLoop	Double	Limits how many times restarted. If 0, then the

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity	Structure	The operation to be pe routing step.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity.operationActivity	String	Name of the operation
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity.version	String	Version of the operatio
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.baseQuantity	Structure	Base quantity of the ro
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.baseQuantity.value	Double	The value of the quanti
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.weighRelevant	Boolean	Indicates if operation i
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.automaticGoodsReceipt	Boolean	Indicates if the operati an automatic goods re
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues	Structure	Custom data for the ro
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues.attribute	String	The attribute name
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues.value	String	The value of the attribu
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup	Structure	The routing step group
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStep	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupType	String	The type of router step following router step g possible: • ANY_ORDER_(• SIMULTANEOU
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i]	Structure	List of routing step gro
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].routingStep.[i]	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].routingStep.[i].stepId	String	The step ID of the rout

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].routingStep.[i].sequence	Double	Controls the order the within the group.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].routingStep.[i].successors	String	A comma-delimited lis successors for this pa member.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingScrapStep	Structure	The routing scrap step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingScrapStep.routingStep	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingScrapStep.routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingReturnStep	Structure	The routing return step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingReturnStep.routingStep	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingReturnStep.routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingHoldStep	Structure	The routing hold step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingHoldStep.routingStep	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingHoldStep.routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingDoneStep	Structure	The routing done step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingDoneStep.routingStep	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingDoneStep.routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.workCenter	Structure	The work center for the
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.workCenter.workCenter	String	The name of the work c
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.reportingStep	String	Indicates how this step external system.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.controlKey	Structure	The control key for the
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.controlKey.controlKey	String	The name of the Contr
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.lastReportingStep	Boolean	Indicates if it is the las

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.rework	Boolean	Indicates if this step is rework.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.queueDecisionType	String	This determines who d SFC should be placed completing this step. T are possible: • COMPLETING_ • NEXT_OPERAT
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.nextStepList	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.nextStepList.[i]	Structure	The list of next possib
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.nextStepList.[i].items	String	The list of next possib

Response Parameter

200

Name	Data Type	Description
plant	String	The plant this routing i
routing	String	The name of the routin
routingType	String	Indicates the type of th the following types of r • NC • SPECIAL • DISPOSITION • SFC • SHOP_ORDER • CONFIGURABL • PRODUCTION_ • SHOPORDER_S
version	String	The routing version
currentVersion	Boolean	If true, the version of th

Name	Data Type	Description
status	String	The status of the routir statuses are possible: • NEW • RELEASABLE • FROZEN • OBSOLETE • HOLD • HOLD_YIELD_ • HOLD_CONSEC • HOLD_SPC_VI • HOLD_SPC_WA
description	String	The description of the
relaxedFlow	Boolean	Indicates if the routing relaxed flow routing.
quantityValidation	Boolean	Indicates if Quantity Va during production.
automaticGoodsReceipt	Boolean	Indicates if the Routing automatic goods receip
routingSteps	StructureArray	
routingSteps.[i]	Structure	Routing steps in the ro
routingSteps.[i].stepId	String	The step ID of the rout
routingSteps.[i].sequence	Double	The sequence of the ro
routingSteps.[i].description	String	The description of the
routingSteps.[i].entry	Boolean	If true, this is an entry
routingSteps.[i].productionProcessId	String	The ID of the productic
routingSteps.[i].productionProcessDefKey	String	Production Process De
routingSteps.[i].routingOperation	Structure	The operation to be pe routing step.
routingSteps.[i].routingOperation.stepType	String	The type of routing ope
routingSteps.[i].routingOperation.maxLoop	Double	Limits how many times restarted. If 0, then the
routingSteps.[i].routingOperation.operationActivity	Structure	The operation to be pe routing step.
routingSteps.[i].routingOperation.operationActivity.operationActivity	String	Name of the operation
routingSteps.[i].routingOperation.operationActivity.version	String	Version of the operatio
routingSteps.[i].routingOperation.baseQuantity	Structure	Base quantity of the ro

Name	Data Type	Description
routingSteps.[i].routingOperation.baseQuantity.value	Double	The value of the quanti
routingSteps.[i].routingOperation.weighRelevant	Boolean	Indicates if operation i
routingSteps.[i].routingOperation.automaticGoodsReceipt	Boolean	Indicates if the operati an automatic goods re
routingSteps.[i].routingOperation.customValues	StructureArray	
routingSteps.[i].routingOperation.customValues.[i]	Structure	Custom data for the ro
routingSteps.[i].routingOperation.customValues.[i].attribute	String	The attribute name
routingSteps.[i].routingOperation.customValues.[i].value	String	The value of the attribu
routingSteps.[i].routingStepGroup	Structure	The routing step group
routingSteps.[i].routingStepGroup.routingStep	Structure	The router step that is
routingSteps.[i].routingStepGroup.routingStep.stepId	String	The step ID of the rout
routingSteps.[i].routingStepGroup.routingStepGroupType	String	The type of router step following types are pos • ANY_ORDER_(• SIMULTANEOU
routingSteps.[i].routingStepGroup.routingStepGroupStepList	StructureArray	
routingSteps.[i].routingStepGroup.routingStepGroupStepList.[i]	Structure	List of routing step gro
routingSteps.[i].routingStepGroup.routingStepGroupStepList.[i].routingStep	Structure	The router step that is
routingSteps.[i].routingStepGroup.routingStepGroupStepList.[i].routingStep.stepId	String	The step ID of the rout
routingSteps.[i].routingStepGroup.routingStepGroupStepList.[i].sequence	Double	Controls the order the within the group.
routingSteps.[i].routingStepGroup.routingStepGroupStepList.[i].successors	String	A comma-delimited lis successors for this pa member.
routingSteps.[i].routingScrapStep	Structure	The routing scrap step
routingSteps.[i].routingScrapStep.routingStep	Structure	The router step that is
routingSteps.[i].routingScrapStep.routingStep.stepId	String	The step ID of the rout
routingSteps.[i].routingHoldStep	Structure	The routing hold step
routingSteps.[i].routingHoldStep.routingStep	Structure	The router step that is
routingSteps.[i].routingHoldStep.routingStep.stepId	String	The step ID of the rout
routingSteps.[i].routingDoneStep	Structure	The routing done step
routingSteps.[i].routingDoneStep.routingStep	Structure	The router step that is
routingSteps.[i].routingDoneStep.routingStep.stepId	String	The step ID of the rout
routingSteps.[i].workCenter	Structure	The work center for the

Name	Data Type	Description
routingSteps.[i].workCenter.workCenter	String	The name of the work c
routingSteps.[i].reportingStep	String	Indicates how this step external system.
routingSteps.[i].controlKey	Structure	The control key for the
routingSteps.[i].controlKey.controlKey	String	The name of the Contr
routingSteps.[i].lastReportingStep	Boolean	Indicates if it is the las
routingSteps.[i].rework	Boolean	Indicates if this step is rework.
routingSteps.[i].queueDecisionType	String	This determines who d SFC should be placed completing this step. T are possible: • COMPLETING_ • NEXT_OPERAT
routingSteps.[i].nextStepList	Structure	The list of next possib
routingSteps.[i].nextStepList.items	String	The list of next possib
customValues	StructureArray	
customValues.[i]	Structure	Routing custom data
customValues.[i].attribute	String	The attribute name
customValues.[i].value	String	The value of the attribu
routingOperationGroups	StructureArray	
routingOperationGroups.[i]	Structure	The operation groups c
routingOperationGroups.[i].routingOperationGroup	String	The name of the routin
routingOperationGroups.[i].description	String	The description of the group
routingOperationGroups.[i].routingOperationGroupSteps	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i]	Structure	The steps of the routin
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep	Structure	The routing step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.sequence	Double	The sequence of the ro
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.description	String	The description of the
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.entry	Boolean	If true, this is an entry
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.productionProcessId	String	The ID of the productio

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.productionProcessDefKey	String	Production Process De
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation	Structure	The operation to be pe routing step.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.stepType	String	The type of routing ope
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.maxLoop	Double	Limits how many times restarted. If 0, then the
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity	Structure	The operation to be pe routing step.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity.operationActivity	String	Name of the operation
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.operationActivity.version	String	Version of the operatio
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.baseQuantity	Structure	Base quantity of the ro
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.baseQuantity.value	Double	The value of the quanti
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.weighRelevant	Boolean	Indicates if operation i
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.automaticGoodsReceipt	Boolean	Indicates if the operati an automatic goods rec
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues	Structure	Custom data for the ro
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues.attribute	String	The attribute name
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingOperation.customValues.value	String	The value of the attribu
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup	Structure	The routing step group
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStep	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupType	String	The type of router step following router step gr possible: • ANY_ORDER_(• SIMULTANEOU

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i]	Structure	List of routing step gro
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].routingStep.[i]	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].routingStep.[i].stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].routingStep.[i].sequence	Double	Controls the order the within the group.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingStepGroupStepList.[i].routingStep.[i].successors	String	A comma-delimited lis successors for this pa member.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingScrapStep	Structure	The routing scrap step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingScrapStep.routingStep	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingScrapStep.routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingReturnStep	Structure	The routing return step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingReturnStep.routingStep	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingReturnStep.routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingHoldStep	Structure	The routing hold step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingHoldStep.routingStep	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingHoldStep.routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingDoneStep	Structure	The routing done step
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingDoneStep.routingStep	Structure	The router step that is
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.routingDoneStep.routingStep.stepId	String	The step ID of the rout
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.workCenter	Structure	The work center for the
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.workCenter.workCenter	String	The name of the work c

Name	Data Type	Description
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.reportingStep	String	Indicates how this step external system.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.controlKey	Structure	The control key for the
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.controlKey.controlKey	String	The name of the Contr
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.lastReportingStep	Boolean	Indicates if it is the las
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.rework	Boolean	Indicates if this step is rework.
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.queueDecisionType	String	This determines who d SFC should be placed completing this step. T are possible: • COMPLETING_ • NEXT_OPERAT
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.nextStepList	StructureArray	
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.nextStepList.[i]	Structure	The list of next possib
routingOperationGroups.[i].routingOperationGroupSteps.[i].routingStep.routingStepGroup.nextStepList.[i].items	String	The list of next possib

SFC

Get an overview of business services related to SFCs.

Start SFC

You can start an SFC via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
plant	String	The plant where the SFC start operation is performed.	Condition property
operation	String	The phase at which the SFC start operation is performed.	Condition property
resource	String	The resource for which the SFC start operation is performed.	Condition property

Name	Data Type	Description	Note
quantity	Number	The quantity of the product to be started in an operation. Partial quantity is not allowed.	
sfcs	String	The shop floor control number for which start operation is performed at an operation activity and a resource.	
processLot	String	The unique identification number of a process lot to which the SFC belongs.	Condition property

Response Parameter

200

Name	Data Type	Description	Note
sfcs	String	The shop floor control numbers	

Complete SFC

You can complete an SFC via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
plant	String	The plant where the SFC complete operation is performed.	Condition property
operation	String	The phase at which the SFC complete operation is performed.	Condition property
resource	String	The resource for which the SFC complete operation is performed.	Condition property
quantity	Number	The quantity of the product to be completed in an operation. Partial quantity is not allowed.	
sfcs	String	The shop floor control number for which complete operation is performed at an operation activity and a resource.	
processLot	String	The unique identification number of a process lot to which the SFC belongs.	Condition property

Response Parameter

200

Name	Data Type	Description	Note

Name	Data Type	Description	Note
sfcs	String	The shop floor control numbers	Condition property
success	Boolean	SFC complete request has succeeded	
queuedOperations	String		
scrapped	Boolean	SFC successfully scrapped	Condition property
buyoffs	String		
doneStatuses	Object		

Serialize SFC

You can serialize an SFC by using a plugin in the Production Operator Dashboard (POD) or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
plant	String	The plant where the SFC is serialized.	Condition property
copyWorkInstructionData	Boolean	Allows you to copy work instruction data from the original SFC.	Condition property
copyComponentTraceabilityData	Boolean	Allows you to copy component traceability data from the original SFC.	Condition property
copyNonConformanceData	Boolean	Allows you to copy nonconformance data from the original SFC.	Condition property
copyBuyoffData	Boolean	Allows you to copy buyoff data from the original SFC.	Condition property
copyDataCollectionData	Boolean	Allows you to copy data collection information from the original SFC.	Condition property
copyActivityLogData	Boolean	Allows you to copy activity log data from the original SFC.	Condition property
sfc	String	The SFC for serialization.	Optional property
quantity	Short	The number of SFCs to serialize.	Optional property
routingStepId	String	The routing step where the SFC is serialized.	Optional property

Response Parameter

200

Name	Data Type	Description	Note
newSfcs	String	The list of SFCs created during serialization.	

Relabel SFC

You can relabel an SFC by using a plugin in the Production Operator Dashboard (POD) or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where the SFC is relabeled.	Condition property
`copyWorkInstructionData`	Boolean	Allows you to copy work instruction data from the original SFC.	Condition property
`copyComponentTraceabilityData`	Boolean	Allows you to copy component traceability data from the original SFC.	Condition property
`copyNonConformanceData`	Boolean	Allows you to copy nonconformance data from the original SFC.	Condition property
`copyBuyoffData`	Boolean	Allows you to copy buyoff data from the original SFC.	Condition property
`copyDataCollectionData`	Boolean	Allows you to copy data collection information from the original SFC.	Condition property
`copyActivityLogData`	Boolean	Allows you to copy activity log data from the original SFC.	Condition property
`sfc`	String	The SFC that is relabeled.	Condition property
`newSfc`	String	The SFC that is created during the relabeling process.	Condition property
`routingStepId`	String	The routing step where the SFC is relabeled.	Condition property

Split SFC

You can split an SFC by using a plugin in the Production Operator Dashboard (POD) or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where the SFC is split.	Condition property
`copyWorkInstructionData`	Boolean	Allows you to copy work instruction data from the original SFC.	Condition property
`copyComponentTraceabilityData`	Boolean	Allows you to copy component traceability data from the original SFC.	Condition property
`copyNonConformanceData`	Boolean	Allows you to copy nonconformance data from the original SFC.	Condition property

Name	Data Type	Description	Note
`copyBuyoffData`	Boolean	Allows you to copy buyoff data from the original SFC.	Condition property
`copyDataCollectionData`	Boolean	Allows you to copy data collection information from the original SFC.	Condition property
`copyActivityLogData`	Boolean	Allows you to copy activity log data from the original SFC.	Condition property
`sfc`	String	The original SFC to split.	Optional property
`splitQuantitySource`	String	The split quantity source from the parent SFC: • `IN_WORK` Quantity in work • `IN_QUEUE` Quantity in queue	Condition property
`routingStepId`	String	The routing step where the SFC is split.	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`newSfcs`	String	The list of SFC names and quantities created during the SFC split.	

Merge SFC

You can merge SFCs by using a plugin in the Production Operator Dashboard (POD) or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

ℹ Note

Standard routers do not allow merging across operations.

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where the SFCs are merged.	Condition property
`copyWorkInstructionData`	Boolean	Allows you to copy work instruction data from the original SFC.	Condition property
`copyComponentTraceabilityData`	Boolean	Allows you to copy component traceability data from the original SFC.	Condition property

Name	Data Type	Description	Note
`copyNonConformanceData`	Boolean	Allows you to copy nonconformance data from the original SFC.	Condition property
`copyBuyoffData`	Boolean	Allows you to copy buyoff data from the original SFC.	Condition property
`copyDataCollectionData`	Boolean	Allows you to copy data collection information from the original SFC.	Condition property
`copyActivityLogData`	Boolean	Allows you to copy activity log data from the original SFC.	Condition property
`parentSfc`	String	The parent SFC for merging.	Optional property
`sourceSfcs`	String	The list of source SFCs for merging.	Optional property
`mergeAcrossOperations`	Boolean	Allows you to merge SFCs at different operation activities.	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`parentSfcQuantity`	BigDecimal	The quantity of the new SFC.	

Set SFC Quantity

You can set quantity for an SFC via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where quantity is set for the SFC.	Condition property
`sfcQuantityRequests:sfc`	String	The SFC for setting a quantity.	Optional property

Note

The set SFC quantity API does not provide support when relaxed flow is set to true, and it is not compatible with any order or simultaneous routing types.

Scrap SFC

You can scrap an SFC via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where the SFC scrap operation is performed.	Condition property
`operation`	String	The phase where the SFC scrap operation is performed.	Condition property
`resource`	String	The resource for which the SFC scrap operation is performed.	Condition property
`sfcs`	String	The shop floor control number for which scrap operation is performed at an operation activity and a resource.	

ℹ Note

- Partial scrapping of SFC is not supported for orders in discrete manufacturing.
- The API call only scraps the parent SFC, and scrapping of sub-assemblies is not supported.

Disposition a Nonconformant SFC

You can disposition nonconformant SFCs using a plugin in the Production Operator Dashboard (POD) or via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where the nonconformant SFCs are dispositioned.	Condition property
`sfcs`	String	The list of SFCs that are nonconformant and need dispositioning.	Optional property
`dateTime`	ZonedDateTime	The date and time for the SFC disposition.	Optional property
`resource`	Boolean	The resource where the nonconformant SFCs are dispositioned.	
`workCenter`	String	The work center where the nonconformant SFCs are dispositioned.	

Response Parameter

200

Name	Data Type	Description	Note
`sfcDispositions: sfc`	String	The dispositioned SFC.	

Name	Data Type	Description	Note
`sfcDispositions:type`	String	The type of the SFC disposition: • `COMPLETE_PENDING` • `DONE` • `DYNAMIC_ROUTING` • `NO_DISPOSITION` • `PARTIAL_COMPLETE` • `QUEUE` • `RETURN` • `SCRAP` • `TRANSFER`	
`sfcDispositions:qty`	BigDecimal	The quantity of the dispositioned SFC	
`sfcDispositions:operation`	String	The next operation activity at which the SFC was queued.	
`sfcDispositions:partialSfc`	Boolean	Indicates whether the remaining quantity was only partially scrapped.	
`dispositionOptions:description`	String	The description of the type.	
`dispositionOptions:type`	String	The disposition options: • `GROUP_MEMBER` • `ROUTING` • `ROUTING_STEP` • `NEXT_STEP` • `RETURN_STEP` • `FUTURE_STEP` • `SCRAP`	
`sfcSplitPerformed`	Boolean	The indicator that shows if the SFC split is performed during the SFC disposition.	

Start Operation Activity

You can start an operation activity of an SFC via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note

Name	Data Type	Description	Note
`plant`	String	The plant where the start operation activity is performed.	Condition property
`order`	String	The order of the SFC	Condition property
`sfc`	String	The shop floor control number for which the start operation is performed at an operation activity.	Condition property
`operationActivity`	String	The unique identifier of the operation activity	Condition property
`workCenter`	String	The work center associated with the operation activity	Condition property
`actualStartDateTime`	String	The UTC time at which the SFC is started	Condition property

Response Parameter

200

Name	Data Type	Description	Note
`plant`	String	The plant name	Condition property
`order`	String	The order of the SFC	Condition property
`sfc`	String	The shop floor control number	Condition property
`operationActivity`	String	The phase of an operation activity	Condition property
`workCenter`	String	The work center name	Condition property
`actualStartDateTime`	String	The date and time at which an operation activity has started.	Condition property

Start Phase

You can start the**Charge** via a public API.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

> **Note**
>
> The use of `Start Phase` API is no longer recommended as it has been deprecated. Instead, please use this alternative `Start Operation Activity` API to perform the desired operation. More information on APIs that perform production activities for a production order, see Process Manufacturing Execution.

Request Parameter

Name	Data Type	Description	Note
`plant`	String	The plant where the start operation is performed.	Condition property
`order`	String	The order of the charge	Condition property

Name	Data Type	Description	Note
charge	String	The unique identifier number of the charge for which the start operation is performed at an operation activity.	Condition property
phase	String	The unique identifier of the operation activity	Condition property
workCenter	String	The work center associated with the charge	Condition property
actualStartDateTime	String	The UTC time at which the charge phase is started	Condition property

Response Parameter

200

Name	Data Type	Description	Note
plant	String	The plant name	Condition property
order	String	The order of the charge	Condition property
charge	String	The shop floor control number	Condition property
phase	String	The phase of an operation activity	Condition property
workCenter	String	The work center name	Condition property
actualStartDateTime	String	The date and time at which the phase has started.	Condition property

Write Set Points

You can use **Write Set Points** via a public API to write target values that you want to achieve for a particular process.

Trigger point: Before Service Execution, After Service Execution, Error During Service Execution

Request Parameter

Name	Data Type	Description	Note
plant	String	The plant where the target value of the set point parameter is consumed	Condition property
operation	Object	The operation to which the upadate is performed	
resource	String	The resource of the SFC activity where the write set points is assigned	Condition property
material	Object	The material name of the product associated with the SFC	
setPointGroup	String	The range of values for a parameter that is set to achieve a desired outcome during production.	Condition property

Name	Data Type	Description	Note
parameter	String	The value that allows to identify a machine setting.	Condition property

Response Parameter

200

Name	Data Type	Description	Note
totalNumberOfIndicators	Integer	Total number of indicators available for a particular process	Condition property
errorResponses	Object	Indicates that an error has occurred while processing a request.	Condition property

Deploy Shop Floor Elements

After completing a design, you need to submit it for deployment by assigning it to a deployment group as a shop floor element. The shop floor elements are then deployed and, if applicable, activated from the deployment group. This process ensures that the deployment to the runtime environment follows a four-eye principle and enables centralized deployment and activation.

Before the deployment, the system validates if all the conditions are met for the shop floor elements to be deployed. You can also run the validation process separately by yourself using the **Validate** option. If any checks fail, you need to fix the errors before you can proceed with the deployment.

Shop Floor Element Structure in Deployment Group

The shop floor elements are displayed in the following structure:

Level 1	Level 2	Level 3	Description
Production Process Design			A production process design created in the Design Production Processes app
	Production Process		A cloud process or an automation sequence
		Trigger Service	A trigger service created for an automation sequence
Service Provider			A service provider created in the Manage Service Providers app
Service			A service created in the Manage Service Providers or Manage Web Servers app
Client Proxy			A client proxy created in the Manage Service Providers or Manage Web Servers app
Message Broker			A message broker created in the Manage Message Brokers app
Shop Floor System			A shop floor system created in the Configure Production Connectivity app
Agent Instance			An agent instance created along with the retention time configured in the subscription

Level 1	Level 2	Level 3	Description
Subscription			A subscription created in the Manage Automatic Triggers app
Timer			A timer created in the Manage Automatic Triggers app
Business Rule			A business rule created in the Manage Automatic Triggers app

Dependency

You can choose Show Dependencies in the deployment group to display the dependency tree for each shop floor element.

Shop Floor Elements	Possible Dependencies
Production Process Design	service, client proxy, service provider, shop floor system
Service	service provider
Client Proxy	service, service provider
Agent Instance	shop floor system
Subscription	agent instance, service, client proxy, service provider, shop floor system, production process design

Deployment Status

A shop floor element or a child element has the following deployment status:

- New: The shop floor element has been submitted to the deployment group but not deployed yet.
- Deployed: The shop floor element has been sent to the runtime environment.
- Failed: Deployment failed and you need to try again to deploy it.
- Awaiting Un-deployment: The deployed shop floor element has been submitted to a new deployment group for deletion.
- Un-deployed: The shop floor element has been deleted from the runtime environment.

A deployment group has the following deployment status:

- New: The shop floor element has been submitted to the deployment group but not deployed or the deployment group is empty as the shop floor element (all statuses except Deployed) has been removed manually from it.
- Validating: The deployment group is being validated to check if all the conditions are met for the shop floor elements to be deployed.
- Deploying: The deployment group is being deployed.

  Note

  Deployment group in Deploying status over one hour will automatically change to Failed status. You need to try again to deploy it.

- Deployed: The shop floor element has been sent to the runtime environment.
- Partially Deployed: Deployment of some shop floor elements has failed but the rest has been successfully sent to the runtime environment.

> **Note**
>
> If a production process design is partially deployed during updating, the successfully deployed part will not get updated in the **Monitor Production Processes** app. For example, if in a production process design, the automation sequence is successfully deployed to the runtime while the cloud process is failed, the automation sequence will not get updated in the **Monitor Production Processes** app.

- **Failed**: Deployment failed and you need to try again to deploy it.

> **Note**
>
> You can delete your deployment group individually or in batch if:
>
> - Deployment group status is **New** and the shop floor element has been removed from the deployment group.
> - Deployment group status is **Deployed** and there is un-deployed shop floor element inside or the shop floor element has been removed from the deployment group.

Activation Status

A shop floor element, a child element or a deployment group has the following activation status:

- **Active**: Activation is done.
- **Activating**: Activation is in progress.
- **Failed**: Activation failed and you need to try again to activate it.
- **Pending**: There are shop floor elements requiring your further operation.

> **Note**
>
> Some shop floor elements do not need to be activated and their activation status will be **Not Applicable**. For example, if a client proxy is configured for a service on an external service provider, the client proxy does not have an activation status. Likewise, shop floor elements awaiting un-deployment also do not have an activation status. If there's nothing to be activated, the **Activate** button won't be available.
>
> The **Deploy Shop Floor Elements** app uses the time zone of your selected plant as the standard time. If no plant is selected, the time of your current browser will be displayed instead.

Deploy and Activate Shop Floor Elements

Deploy shop floor elements to translate design-time models into runtime configurations. The shop floor elements are organized in different groups and the deployment is done at the group level.

Prerequisites

- You either have the `Production_Engineer` role or the `Automation_Engineer` role.
- You are an administrator of the deployment group.

Procedure

1. Open the **Deploy Shop Floor Elements** app.

2. On the **Deployment Groups** tab, select a deployment group.
3. Choose **Deploy** or **Deploy and Activate**.
4. The system starts to validate if all the conditions are met for the shop floor elements to be deployed.
5. When all the validation points are passed, the system deploys the shop floor elements to the runtime environment.
6. If you choose **Deploy and Activate**, the system activates these deployed shop floor elements to make them up and running in the runtime environment.

   If you chose **Deploy**, you can choose **Activate** by yourself to run the activation process.

Results

- Relevant configurations are created or updated in the runtime environment.

  In the corresponding Production Connector / SAP Plant Connectivity systems, for shop floor elements grouped under **Production Process Design**, the configuration elements are prefixed with the production process design name, the version, and the timestamp for deployment.

  For more information, see Mapping Between Process Models and Production Connector / SAP Plant Connectivity Configurations and Mapping Between Machine Model Entities and PCo Configurations.

- The relevant internal service providers are all started (activated), including those deployed from the deployment group and the service providers that the shop floor elements depend on.
- You can publish deployed cloud processes to service registry by choosing **Edit Header** and switching on the **Publish to Service Registry** toggle in the **Design Production Processes** app. The process you publish will be displayed in service library for easy reuse in other apps.

  Note

  The published process will be removed from service registry once you delete the original process in production process design. Any updates of the original process will be synced to that in the service registry.

Manage Users in a Deployment Group

A user must be assigned to a deployment group to work with it. You can assign users to a deployment group either directly or by importing them from work groups.

Prerequisites

- You have the role of `Manufacturing_Admin`, `Automation_Engineer` or `Production_Engineer`.
- You are an administrator of the deployment group.

Context

A user can be either an administrator or a contributor in a deployment group. An administrator can do everything with the deployment group, such as deploying shop floor elements and managing users. A contributor can only add and remove shop floor elements, and perform validation on the deployment group.

By default, the creator of a deployment group is its first administrator and user.

There must be at least one administrator in each deployment group.

Procedure

1. Open the **Deploy Shop Floor Elements** app.
2. On the **Deployment Groups** tab, select a deployment group.
3. Go to the **Users** tab.
4. To add users, choose **Add**.

   Note that if you import users from work groups, all the users in the selected work groups are added to the deployment group.
5. To edit the role of each user, choose **Edit**.
6. To remove users from the group, choose (Delete).

Related Information

Manage Work Groups

Monitor Production Processes

You use the **Monitor Production Processes** app to monitor and track the execution status of all production processes and subprocesses in a specified period of time. **Monitor Production Processes** helps you quickly react in case of error and identify the root cause of execution failures or high defect rates.

> **Note**
> You can use the **Monitor Production Processes** app on cloud. On cloud, the app supports cloud process and Production Connector on Windows / SAP Plant Connectivity on Windows automation sequence instances (only for Production Connector, SAP Plant Connectivity version 15.5.4 or above). Production Connector on Windows / SAP Plant Connectivity on Windows automation sequence instances include those triggered by cloud process as subprocesses, test-run from Production Process Designer, and those triggered by subscription.

You can use the search bar to search for process instances you want to monitor, and specify the status and a period of time to further filter process instances.

The **Monitor Production Processes** app uses the time zone of your selected plant as the standard time for all process instances. If no plant is selected, the time of your current browser will be displayed instead.

Search and Filter Process Instances

You can search for process instances, including archived processes by process name or ID. To search for process instances, do the following:

Prerequisites

You have one of the following roles:

- `Production Supervisor`
- `Production Engineer`
- `Automation Engineer`
- `Plant Manager`
- `Production Operator`

Procedure

1. Open the **Monitor Production Processes** app.
2. Choose a runtime environment from the dropdown list. Choose either **DMC_Cloud** or one of the Production Connector / SAP Plant Connectivity systems.

   > **Note**
   >
   > To search for process instance by instance ID, input the ID in the **Instance ID** field and choose **Go**.

3. Choose *(select production process)* .

   > **Note**
   >
   > For automation sequences in Production Connector / SAP Plant Connectivity runtime environment, enter keywords for the system to filter the automation sequence list for you in the table view.

4. In the **Select Production Processes** window, enter keywords for your production process design and production process, name or the description text.
5. Choose **Go**.
6. Select one or more processes you want to monitor.
7. Choose **OK**.
8. Specify the status, date range, and label to further filter process instances.

   > **Note**
   >
   > You can also filter process instances by parameter value or variable value. The system will return the following process instances if:
   >
   > - the searched value matches the latest or previous input or output parameter value of the process instances
   > - the searched value matches the latest or previous input or output parameter value of any step (except **Error End**, **POD Connection**, **POD Message**, **Condition**) of the process instances
   > - the searched value matches the latest or previous process variable value
   >
   > The input size of the parameter/variable value filter should be no larger than 4000 bytes.

Monitor and Track Production Processes

You will see relative information of process instances in the KPI carousel and list table below the search bar.

The filtered process instances are displayed in the list table, in descending order of start time, including both the main process and subprocesses. To customize columns to be displayed and column order, choose *(Define Column Properties)*. You can add start time, end time, duration, production process design, label, the original trigger type, trigger name and ID of the process instance into the list table. The trigger types are:

- Production Process Designer
- Production Process
- Production Connector / Plant Connectivity
- POD (Production Operator Dashboard)
- Subscription
- Timer

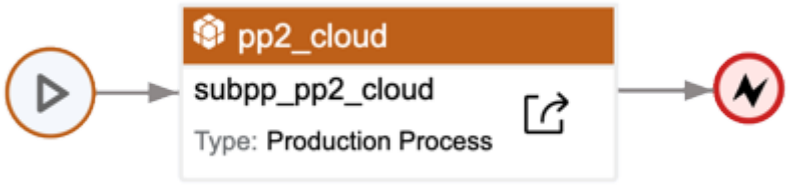
- Rule (Business Rule)
- Monitor Production Processes
- REO (Resource Orchestration)

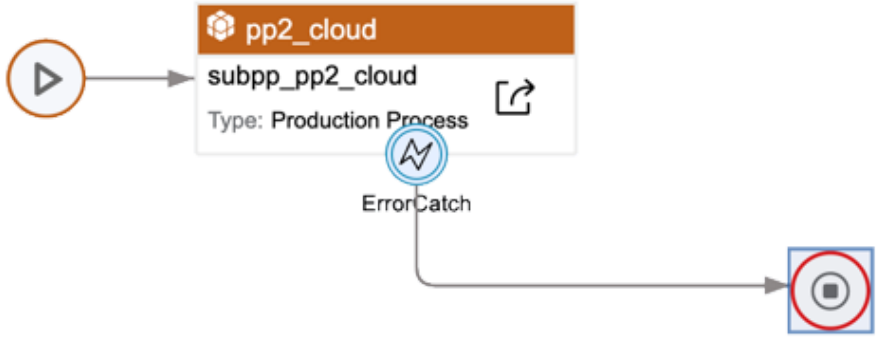
Note

Process re-launched from the **Recover Production Processes** app will be given "**(recovered)**" next to the trigger type. Process triggered by debug mode or debugged in the Production Process Designer will be given "**(debugged)**" next to the trigger type.

Status

The total number of statuses of filtered process instances is displayed in the KPI carousel above the table area. Status for each production process instance is displayed in the table area. The process instances have five different status types:

Name	Note
Running	The process is being executed. **Note** The status of main process is **Running (waiting)** if main process is being executed and waiting to continue. The status of main process is **Running** and subprocess is **Running (waiting)** if subprocess is being executed and waiting to continue. The status of main process is **Running (debugging)** if main process is debugging.
Failed	The process is terminated unexpectedly or ends in error. **Note** If you use an error end element at the end, the process execution result status will be **Completed with Error** instead of **Failed**.
Completed with Error	The process is executed to the expected error end. **Note**  This status is for processes with an error end element used in expectation at the end.

Name	Note
Completed	The process is successfully executed. **Note**  The execution result status for a process catching an error and successfully running to the end is **Completed**.
Terminated	The process is terminated by the user.

Note

Automation sequences only have the following statuses: **Running**, **Failed** and **Completed**.

Process Instance Detail Page

Choose the process instance entry to open the process instance detail page. The process instance detail page provides general information of production process, its step list, a whole picture of the workflow in graphics, input and output parameters and execution log detail information based on the log level you choose.

You can also debug and terminate the process instance in this section. See Configure Debug Mode Settings, Debug Deployed Production Processes, and Terminate Process Instances for details.

General Information section provides process instance name and ID, production process name, production process design name, runtime environment, process instance ID, booster mode, log level, status, duration, start time and end time.

You can quickly navigate to the corresponding production process page and production process administration list page by choosing **Design Production Processes** or **Production Process Administration** from the popover of the production process name link.

Process Instance Hierarchy tab

You can have a hierarchical overview of all the parent and child process instances of the current one, along with their runtime environment information. Navigate through links to their corresponding detail pages.

Process Instance Steps tab

The **Step List** covers detailed information of steps, including services, subprocesses, script tasks, POD plugins, POD messages and so on. You can choose *(Columns)* to customize the columns to be displayed.

Name	Note
Step	A service or a subprocess name. Script task included. You can quickly jump to subprocess when there is a link available.

Name	Note
Type	See Available Services and Subprocesses.
Status	See previous “Status” section.
Message	HTTP status code and error message.
Start Time	Start time (milliseconds) of step execution.
End Time	End time (milliseconds) of step execution.
Duration	Time length of step execution.

You can select segment tabs in the right panel to switch into graphic view, parameter view and log detail view.

ℹ Note

The system uses the date and time of your selected plant as the standard time for all process instances. If no plant is selected, the time of your current browser will be displayed instead.

Graphic View

The graphic view provides a whole picture of the process workflow as you see in production process designer. Steps are displayed with status icon attached to show the status information. Step is highlighted in the graphic view when you choose that step in the step list and vice versa. You have also the option to zoom in, zoom out the graphic, and zoom back to its original size.

Parameter View

The parameter view shows the input and output parameters detail of the current process instance.

Log Detail View

Log information of the process instance is displayed in your preferred languages in the log detail view, with each log level in different color. Translation is provided as you change the user interface language. Choose **Technical Details** to view the log with more technical information. See Log Level for details.

Terminate Process Instances

You can terminate running cloud production process and subprocess instances in the list table section and process instance detail page.

Prerequisites

You have one of the following roles:

- `Production Supervisor`
- `Production Operator`
- `Production Engineer`

Context

When a process is terminated, its parent process and running subprocess instances are terminated automatically. Subprocesses with "Execute Asynchronously" and automation sequences will not be affected.

> **Note**
>
> The termination operation is not reversible.

Procedure

1. Choose **Terminate** in the **Action** column of the list table to terminate process.
2. Alternatively, choose the process instance entry to navigate to the process instance detailed page.
3. Choose **Terminate** on the upper right corner of the page to terminate the current process instance.

Results

The current process instance is terminated. You can check the process is terminated in the status column of the list table and process instance detail page. You can try to terminate the process again if termination failed.

Analyze Process Instances

The statistics report gives you a visualized view of process instance performances in different dimensions.

Prerequisites

You have the role of `Production_Supervisor` or `Production_Engineer`.

Context

You gain insight into the trend of process instances executed in a period of time.

Procedure

1. Open the **Monitor Production Processes** app.
2. Choose **Statistics Report** in the upper right corner.
3. Choose *(select production process)* .

   > **Note**
   >
   > By default, the process trending view displays all process instances executed in the last 10 days.
   >
   > The system uses the date and time of your selected plant as the standard time for all process instances. If no plant is selected, the time of your current browser will be displayed instead.

4. In the **Select Production Processes** window, enter keywords for your production process design and production process name or the description text.
5. Choose **Go**.
6. Choose the dimension and measures.

   > **Note**
   >
   > There are two dimensions.

- **Count by Status**: Compare the number of process instances executed by different statuses.
- **Duration**: Compare the execution duration of process instances.

7. Enter a number between 1 and 28 for the day range.
8. Choose **Go**.

Recover Production Processes

You can use the **Recover Production Processes** app to re-launch asynchronously-run cloud processes terminated with error. The app stores all necessary data of the process for a re-launch. You can change process input parameter values and re-launch the process.

You can search for the process instance by keywords, combined with termination mode or timestamp.

> **Note**
> The **Recover Production Processes** app uses the time zone of your selected plant as the standard time for all process instances. If no plant is selected, the time of your current browser will be displayed instead.

Choose the *(Process Instance Detailed Page)* to view the detailed page of the process instance. The process instance detailed page provides general information of production process and the input parameters with values that you can change later on.

Select and Retry Production Processes

Prerequisites

- You have the role of `Production Supervisor`.
- You have switched on the **Show Process in Recovery List** toggle in the right panel of an error end element of a cloud process in the Production Process Designer. See Controls for details.
- You have run the cloud process in asynchronous way.

Procedure

1. Select the process instances you want to re-launch.
2. Choose **Change Input and Retry**.

   You can input your own input parameter value in the variable list window and re-launch the production process. You can also do this in the process instance detailed page.
3. Optionally, choose **Retry**.

   > **Note**
   > The production process instance will remain in the **Recover Production Processes** app if it is deleted from the **Monitor Production Processes** app or its production process design is deleted in the Production Process Designer. However, you will see an error message in this case.

4. Optionally, choose **Delete**.

   This will remove the process instance in the recovery list only.

Manage Service Registry

You can register your own API services or extensions in SAP Digital Manufacturing using the **Manage Service Registry** app.

For details, see Registering Services in the SaaS Tenant of SAP Digital Manufacturing.