Aras Digital Twin Core

12.0R3

User Guide

Document #: 12.0R3.2021030204

Last Modified: *12/23/2021*

Copyright Information

Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810

Phone: 978-806-9400
Fax: 978-794-9826

E-mail: support@aras.com
Website: https://www.aras.com

Table of Contents

Send Us Your Comments

Aras Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for future revisions.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where and what level of detail?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, indicate the document title, and the chapter, section, and page number (if available).

You can send comments to us in the following ways:

Email:
TechDocs@aras.com
Subject: Aras Product Documentation

Or,

Postal service:
Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Attention: Aras Technical Documentation

If you would like a reply, provide your name, email address, address, and telephone number.

If you have usage issues with the software, visit https://www.aras.com/support/

Document Conventions

The following table highlights the document conventions used in the document:

Table 1: Document Conventions

Convention	Description
Bold	This shows the names of menu items, dialog boxes, dialog box elements, and commands. Example: Click OK.
Code	Code examples appear in courier font. It may represent text you type or data you read.
Yellow highlight	Code highlighted in yellow draws attention to the code that is being indicated in the content.
Yellow highlight with red text	Red text highlighted in yellow indicates the code parameter that needs to be changed or replaced.
Italics	Reference to other documents.
Note:	Notes contain additional useful information.
Warning	Warnings contain important information. Pay special attention to information highlighted this way.
Successive menu choices	Successive menu choices may appear with a greater than sign (-->) between the items that you will select consecutively. Example: Navigate to File --> Save --> OK.

1 Application Overview

The Aras Innovator Digital Twin Core (DTC) application is a business-ready solution that enables the recording and management of changes to a product's physical configuration at the completion of manufacturing and any time after as it is operated and maintained. This Digital Twin configuration can be created for any product or system of products in any industry.

The Digital Twin configuration is the digital representation of an existing asset in the context of its as-built and as-maintained configuration. Every individual Digital Twin configuration can be linked to its related content across the lifecycle using the Digital Thread. This includes specifications, engineering models, drawings, problem reports, service bulletins, and work orders that describe the asset materials, components, geometry, and behavior. For example, an aircraft Digital Twin configuration uniquely distinguishes the given aircraft from others and describes its current properties. The unique aircraft identifier is its serial number as well as its related serialized components that make up the configuration of the aircraft. The aircraft-related Digital Thread information is linked to specific serialized components. It can include aircraft aerodynamic and 3D models, engineering changes cut in during the production cycle, and any deviations from the original design specifications approved due to issues and workarounds on the specific product unit.

The DTC application implementation of the Digital Twin concept is a Physical Part Item linked to the appropriate Engineering/Design Part Item and capable of being in Physical Part BOMs—Physical Part Item assembly-component (parent-child) relationships.

The Digital Thread is an information trail of the Digital Twin throughout each life stage: design, manufacturing, operation, and decommissioning. It is a set of digital items and the relationships between them that describes the product and each component's current operational activities and keeps the operational history from early designing to final decommissioning. For example, the aircraft Digital Thread can define who can replace the aircraft components and how they are replaced. The Digital Thread keeps the replacement history.

The DTC application includes a PhysicalPart Lifecycle Map, DTC Identities, DTC Permissions, Life Units, Life Parameters, Life Policies, Part Policies, Operational Event Types, and Operational Events as well as their constraints and historical viewing of the Physical Part BOMs and Physical Part life values.

All the application functionality is available via the Aras Innovator 12 User Interface (UI).

2 DTC Identities

The DTC application has built-in Identities that are referred to as DTC Identities. They are strictly defined to achieve secure control over the DTC application Items: who can create, access, modify, and delete such Items. The DTC Identities consist of two groups:

- Asset Identities
- Operational Event Identities

Your abilities to perform actions on the DTC Items and their Relationships depend on your DTC Identity membership. An Aras Innovator Administrator adds and removes DTC Identity members per business needs.

This section describes the out-of-the-box configuration of the DTC Identities.

Some DTC features are restricted to certain DTC Identities. If a feature is Identity-limited, this document will start the section on that feature with a note telling for whom it is available.

If a feature (command) is not available for the given Identity, it may not be displayed or may be displayed in a lighter color, or an error may be raised when trying to save a DTC Item and having this feature applied to the Item. Unallowed Item modifications cannot be saved in this case.

2.1 Asset Identities

The Asset Identities are intended to access and work with the **Physical Part**, **Physical Part BOM**, **Life Unit**, **Life Parameter**, **Life Policy**, **Part Policy** Items, and their Relationships.

The Asset Identities are as follows:

- **Asset Viewer**: an observer of the allowed Items and their Relationships with view-only Access Rights to them. For example, this person can be an inspector that investigates asset and component histories and is not allowed to manipulate assets and components.
- **Asset User**: a regular user of the allowed Items and their Relationships with basic Access Rights to them. For example, this person can be an in-the-field technician that installs, removes, or replaces real-world parts in assemblies and keeps track of the changes made.
- **Asset Editor**: a creator of some allowed Items and their Relationships with advanced Access Rights to them. For example, this person can be a workshop supervisor or an officer responsible for digital twins of real-world assets.
- **Asset Admin**: an administrator of the allowed Items with unlimited Access Rights to them. For example, this person can be a senior officer that manages an MRO department.

2.2 Operational Event Identities

The Operational Event Identities are intended to access and work with the **Operational Event Type** and **Operational Event** Items. They can also access and view **Physical Part** Items.

The Operational Event Identities are as follows:

- **Operational Event Creator**: a creator of the **Operational Event** Items with basic Access Rights to them. For example, this person can be an asset operator responsible for logging and tracking the in-the-field operational data of the asset.

- **Operational Event Reviewer**: a creator and reviewer of the **Operational Event** and **Operational Event Type** Items with advanced Access Rights to them. For example, this person can be a shift supervisor or a manager responsible for tracking real-world assets.
- **Operational Event Admin**: an administrator of the **Operational Event** and **Operational Event Type** Items with administrative Access Rights to them. For example, this person can be a senior officer that manages an operational department.

3 DTC UI Overview

This section describes only the User Interface (UI) specifics of the DTC Items. It assumes that you are familiar with the standard Aras Innovator 12 UI.

The DTC application reuses the standard Aras Innovator 12 UI. This UI is dynamic: some of its components, such as fields, buttons, and menu commands, are shown or hidden depending on the current Item context. Additionally, when a UI component is displayed in a lighter color, it indicates that this component is not available in the given context.

The current Item State, claimed status, and your Identity membership define the context for this Item.

Note: The screenshots may differ from what you see on your Aras Innovator screen because of your Identity membership and Aras Innovator instance customization.

3.1 Accessing DTC Items

Out of the box, the DTC Items are located under **Contents** --> **Assets**.

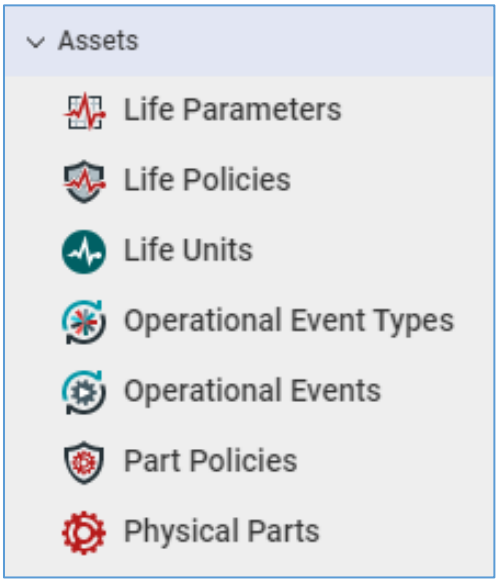


Figure 1.

You can also access them from the Aras Innovator background page.

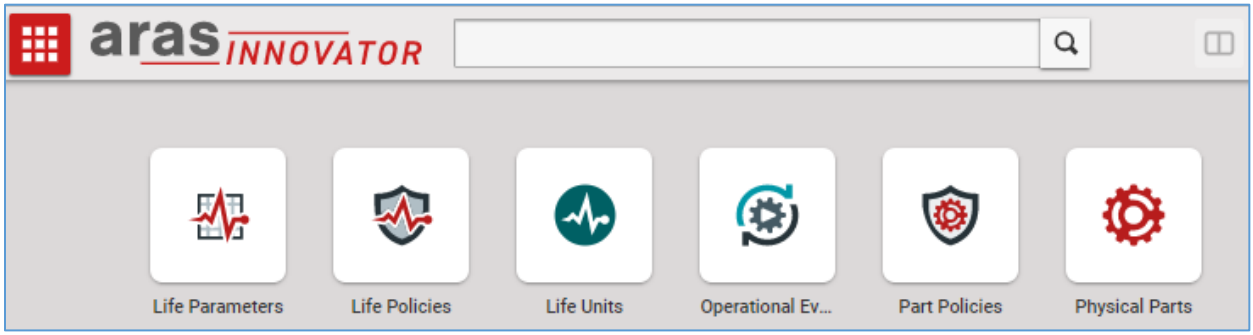


Figure 2.

Their visibility depends on your Asset Identity. They are the only DTC Items that you can manage:

- **Asset Viewer**:
  - **Operational Events**
  - **Physical Parts**
- **Asset User**:
  - **Operational Events**
  - **Physical Parts**
- **Asset Editor**:
  - **Operational Events**
  - **Part Policies**
  - **Physical Parts**
- **Asset Admin**:
  - **Operational Events**
  - **Life Parameters**
  - **Life Policies**
  - **Life Units**
  - **Part Policies**
  - **Physical Parts**
- **Operational Event Creator**:
  - **Operational Events**
  - **Physical Parts**
- **Operational Event Editor**:
  - **Operational Events**
  - **Operational Event Types**
  - **Physical Parts**
- **Operational Event Admin**:
  - **Operational Events**
  - **Operational Event Types**
  - **Physical Parts**

3.2 Quick Search for DTC Items

The DTC Item **Contents** entry features **Quick Search** for Items by an Item **Number** or **Code** (like **Policy Code**) depending on a given ItemType. For example, you can search for a **Physical Part** Item by **Part Number** and inventory-control identifier (**Serial Number** or **Lot / Batch Number**).

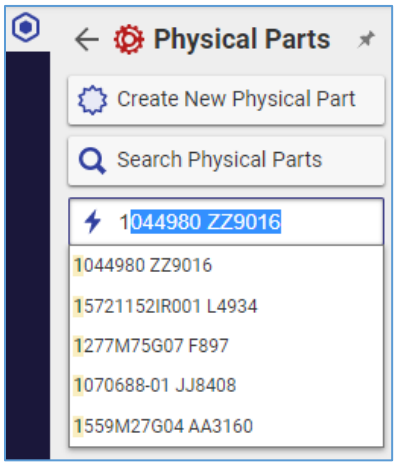


Figure 3.

Start typing the necessary Item **Number** or **Code**, and the system will display a list of matching Items. Click the **required Item** to open its Item view.

3.3 Item Action Menu

An Item action context menu displays the standard Aras Innovator Item commands for the given ItemType in the Search Grid. Right-click the **Item** in the Grid to access its menu.

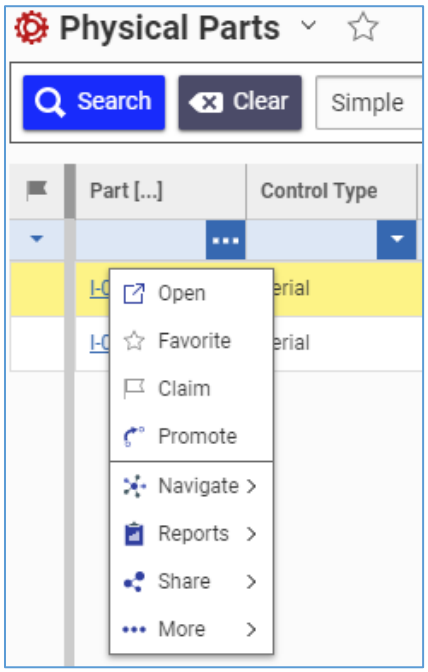


Figure 4.

The Item action menu is an alternative access point to the functionality provided in the Item toolbar when you open this Item. The action menu contents vary according to the Item context.

3.4 Item view

The view of a DTC Item has the following standard Aras Innovator 12 UI components:

1. The Page Title bar shows a given Item **Number** or **Code** (like **Policy Code**). For example, it displays **Part Number** along with **Serial Number** or **Lot / Batch Number** for a **Physical Part** Item.
2. The Item toolbar provides the standard Aras Innovator Item commands available for the Item under the current Item context.
3. The Form accordion stores the Item properties.
4. The Relationship accordion keeps Relationships associated with the Item.

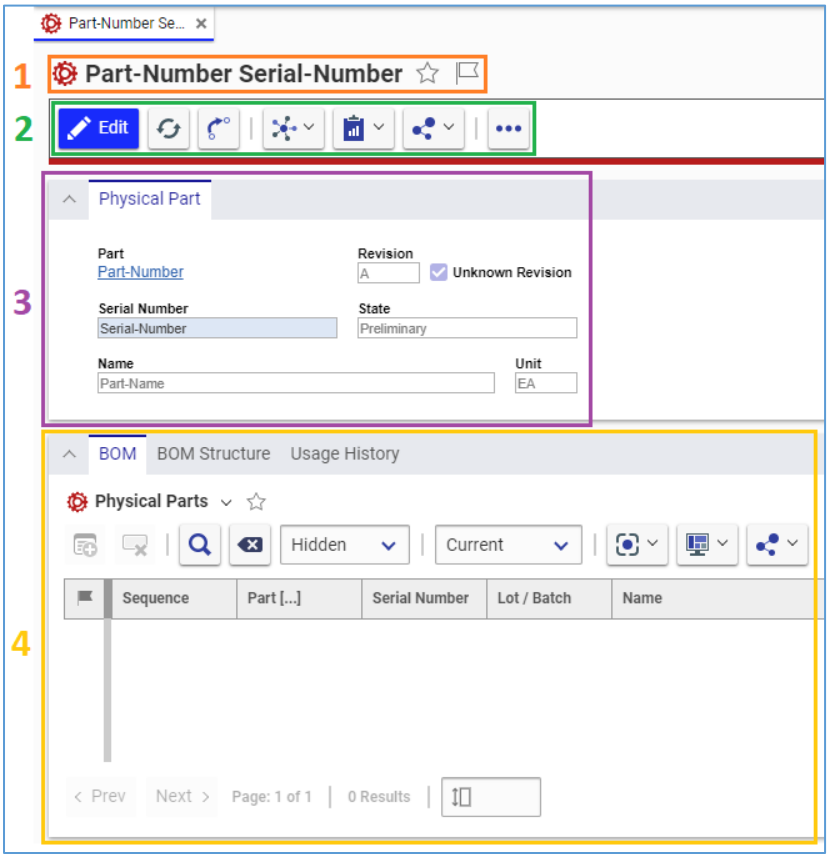


Figure 5.

The contents of all the components vary according to the Item context.

3.5 Specifying Date and Time values

The DTC application works on multiple Aras Innovator 12.0 Service Packs (SPs), across which there are three ways of specifying date and time values:

1. Old **Date dialog**

2. New **Select Date** dialog
3. Typing

Which ones are available to you depends on a field (cell) and your Aras Innovator SP version.

The old **Date dialog** is a UI feature for filling the date and time fields and Grid cells in Aras Innovator. Aras Innovator 12 SP 14 introduces the new **Select Date** dialog that should replace the old one. Some fields or cells may still use the old one in this SP or later.

3.5.1 Old Date dialog

Use the following procedure to specify a date and time value using the old **Date dialog**:

1. Click a **necessary** field (Grid cell) or the **Calendar** button in it. The **Date dialog** pops up.

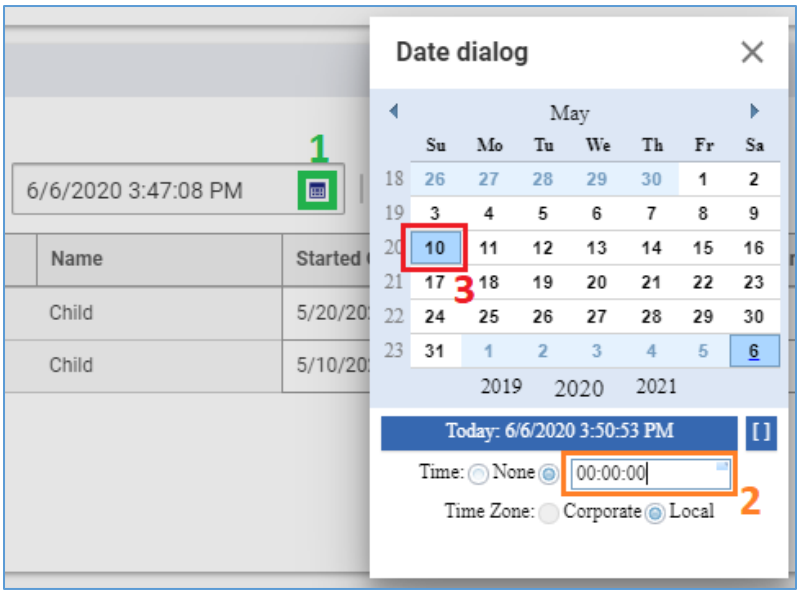


Figure 6.

2. Select a time.
3. Select a date.

   The **Date dialog** disappears, and the selected date and time populate the **necessary** field or Grid cell.

3.5.2 New Select Date dialog

Use the following procedure to specify a date and time value using the new **Date dialog**:

1. Click a **necessary** field (cell) or a **Calendar** button in it.

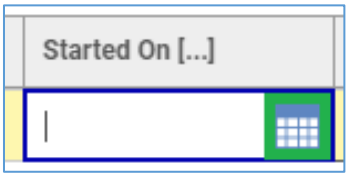


Figure 7.

   The **Select Date** dialog pops up.

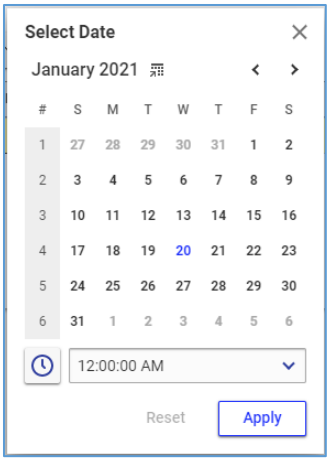


Figure 8.

2. Select the time from the list or type it to specify it.

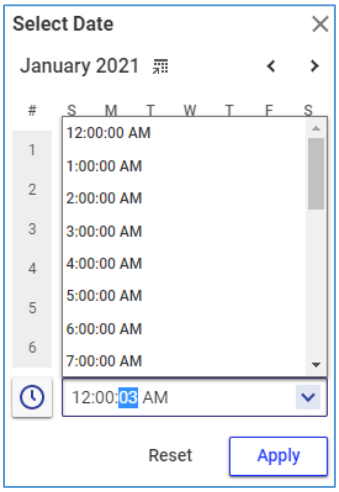


Figure 9.

3. Click a date to specify it.

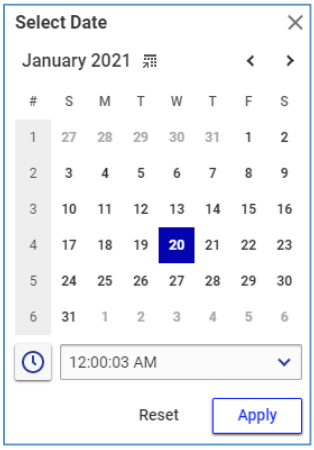


Figure 10.

Note: In this dialog, the specifying order is irrelevant: you can specify a date and then a time.

4. Click the **Clock** button to populate the **time** field with your current time.

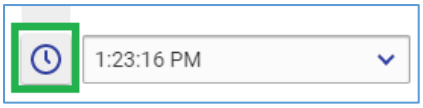


Figure 11.

5. Click the **back** and **forth arrow** buttons to scroll the calendar between the months.

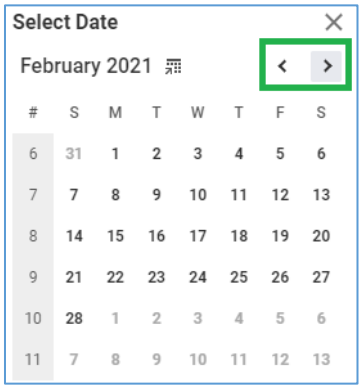


Figure 12.

6. Click the **Apply** button.

   The **Date dialog** disappears, and the selected date and time populate the **necessary** field or Grid cell.

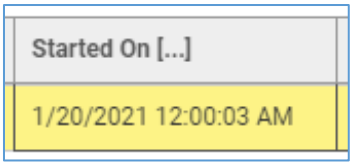


Figure 13.

To specify a month quickly:

1. Click the **Month Year** button.

Figure 14.

2. Click the **necessary month** in the **Months** grid. The dialog shows the days calendar for this month.

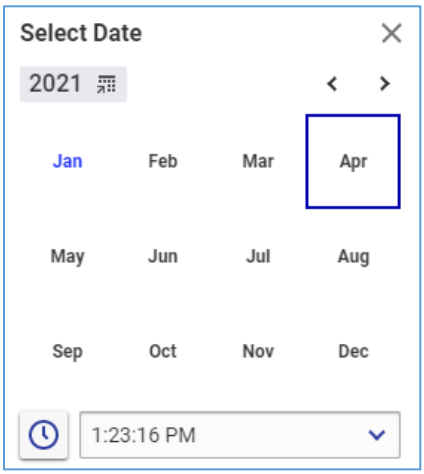


Figure 15.

3. Click the **back** and **forth arrow** buttons to scroll the **Months** grid between years.

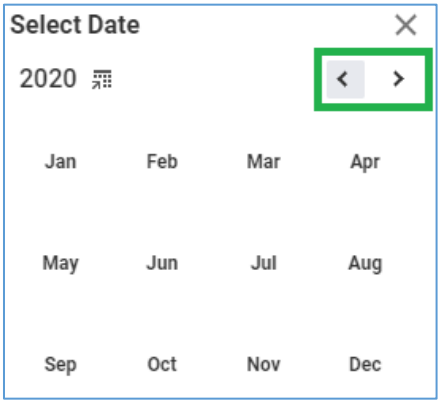


Figure 16.

To specify a year quickly:

1. Click the **Year** button.

Figure 17.

2. Click the necessary **year** in the grid. The dialog shows this year's months as the grid.

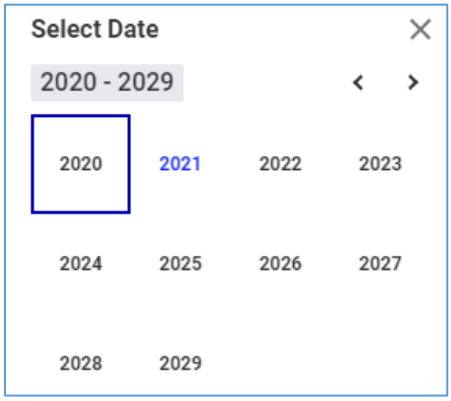


Figure 18.

3. Click the **back** and **forth arrow** buttons to scroll through the **Years** grid.

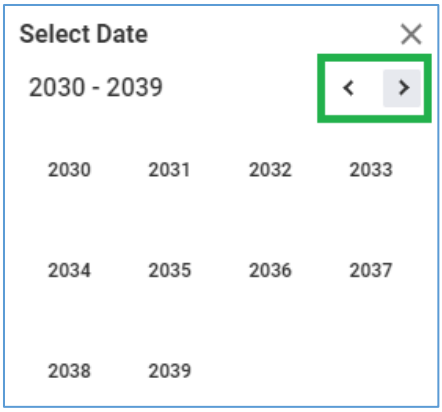


Figure 19.

3.5.3 Entering a Date and Time

You can enter a value in some of the date and time fields and Grid cells. The entered value must meet the date and time template for your Locale setting.

For example, the US Locale date and time template is as follows:

MO/DD/YYYY HH:MI:SS XM

where:

MO—month as a number

DD—day

YYYY—year

HH—hour

MI—minute

SS—second

XM—AM or PM

An error tooltip may be shown, and an Item may not be saved if the typed value is invalid.

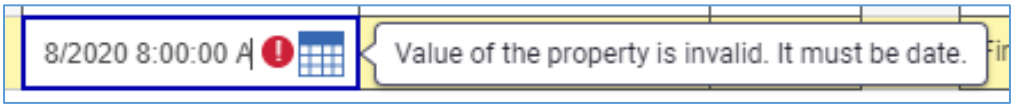


Figure 20.

3.6 Sequence Values in the DTC Relationships Grids

The DTC application customizes the automatic assignment of the **Sequence** values for the DTC Items. When creating a new Relationship Item in some Relationship grids of the DTC Items using the Aras Innovator UI, this Relationship Item has its **Sequence** value automatically set as a ten-divisible number next after the highest existing **Sequence** value. For example, the first Relationship Item has its **Sequence** set to 10, the second to 20, the third to 30, and so on.

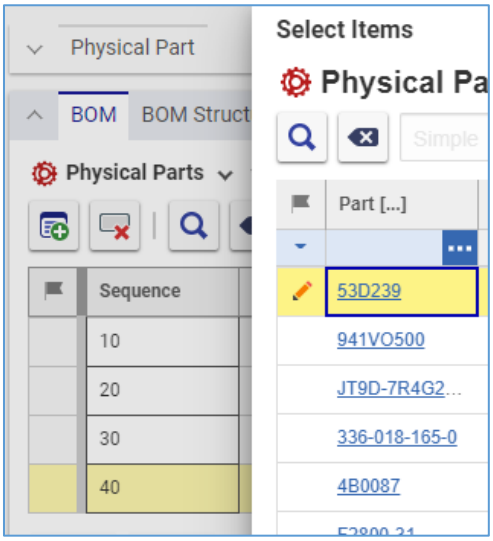


Figure 21.

It is highly recommended that you stick to this logic.

If some existing **Sequence** values are manually changed, the new one will still be assigned as described. For example, if there are Relationship Items with **Sequence** values such as 10, 20, 24, 30, 39, the next one will have 40; if 12, 32, 41, 55, 63, it will be 70; if 10, 28, 33, 27, it will be 40.

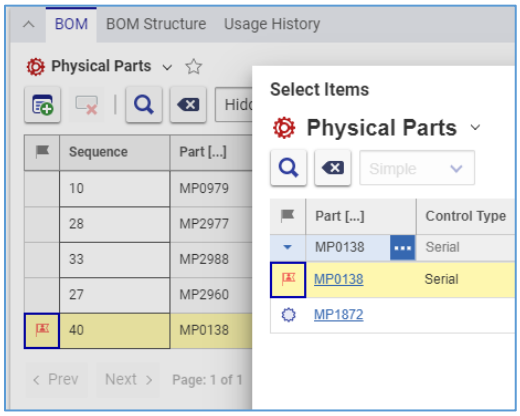


Figure 22.

3.7 Automatically managed DTC Relationships Grids

The DTC application automatically manages some of the Relationships Grids. No one can manage such Grids manually, for example, the **Life Units** Relationships Grid of an **Operational Event** Item. However, when editing a source Item, such a Relationships Grid has the standard **Add *Items***, **New *Item***, **New *Relationship Item***, and **Delete *Relationship Item*** buttons enabled on its toolbar.

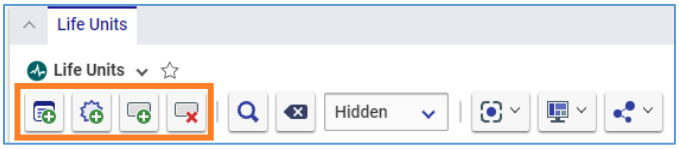


Figure 23.

But, manual management of such Relationship Items is still forbidden—an error is raised upon saving the source Item if either:

- An existing Item is added manually to the Grid.

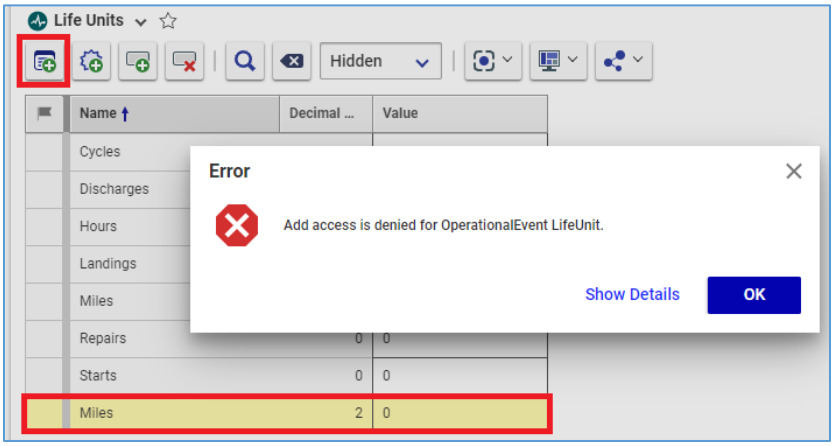


Figure 24.

- A new Item is created and added manually to the Grid.

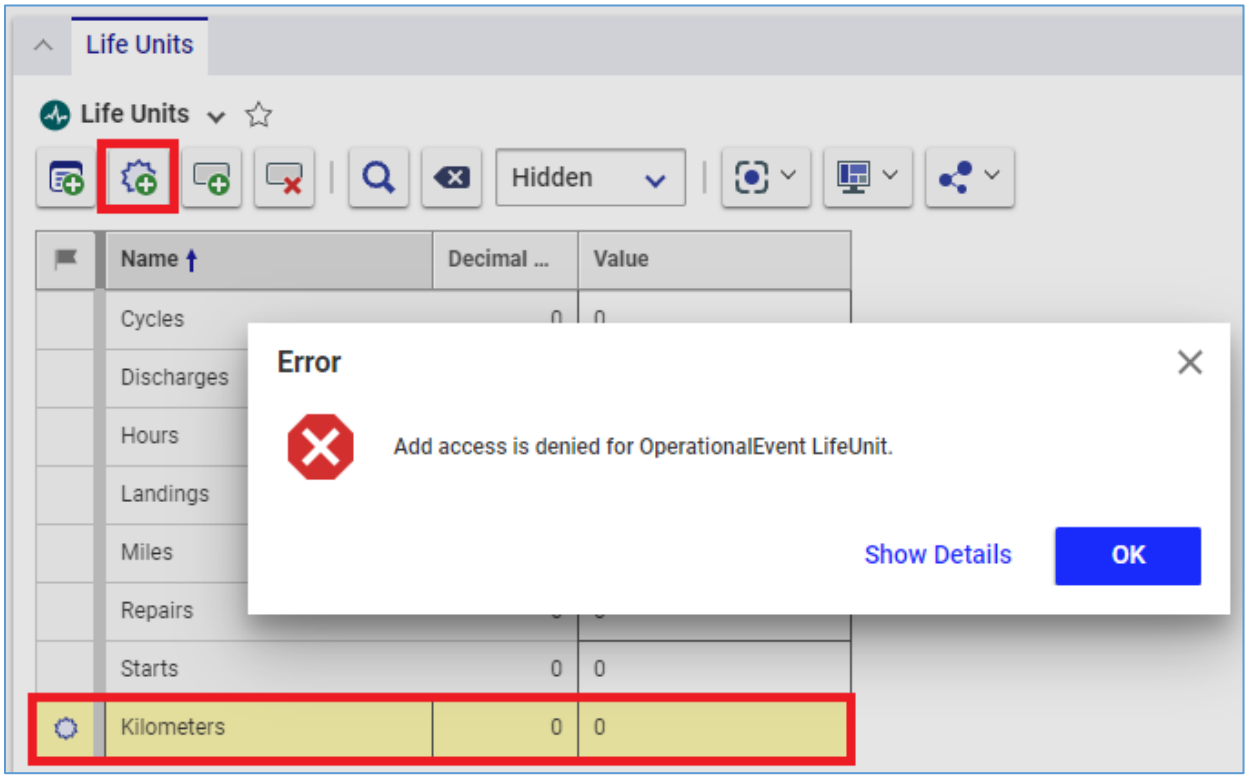


Figure 25.

- A new Relationship Item is created manually.

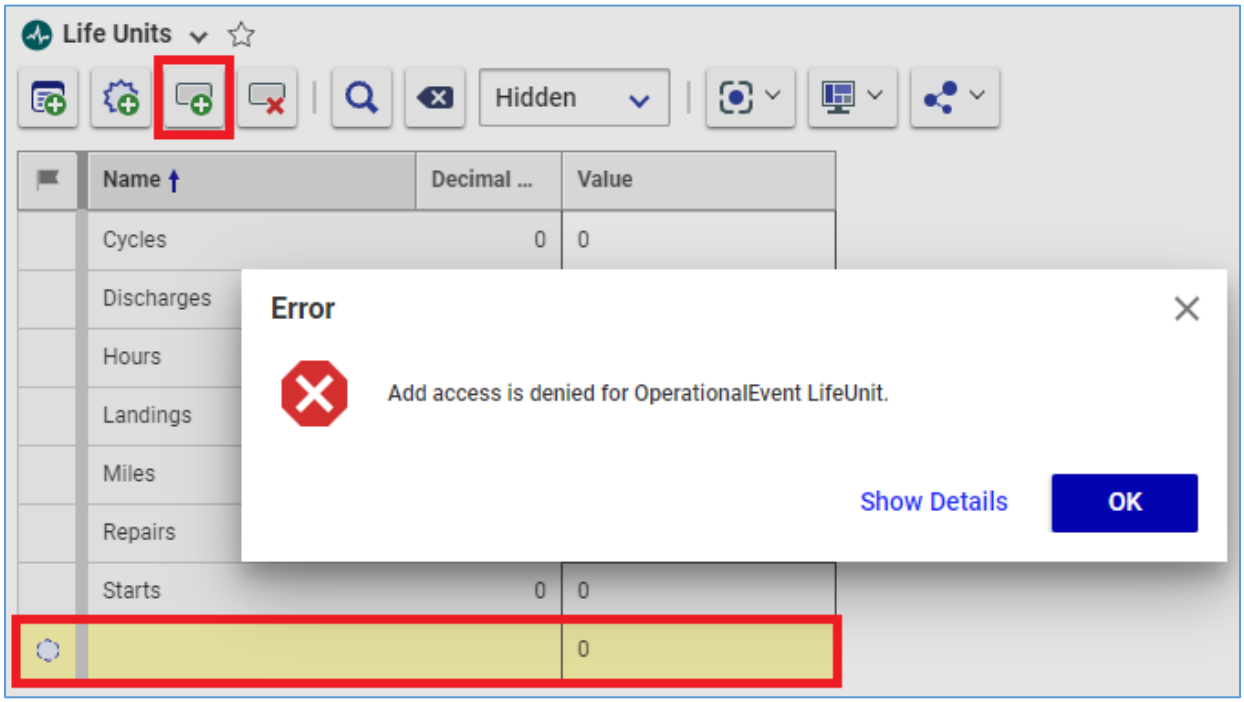


Figure 26.

- An existing Relationship Item is removed manually from the Grid.

Note: A related Item cannot be removed from a source Item in an automatically managed Relationship Item. If it is necessary to edit or delete such a related Item from the database, its source Item should be deleted first.

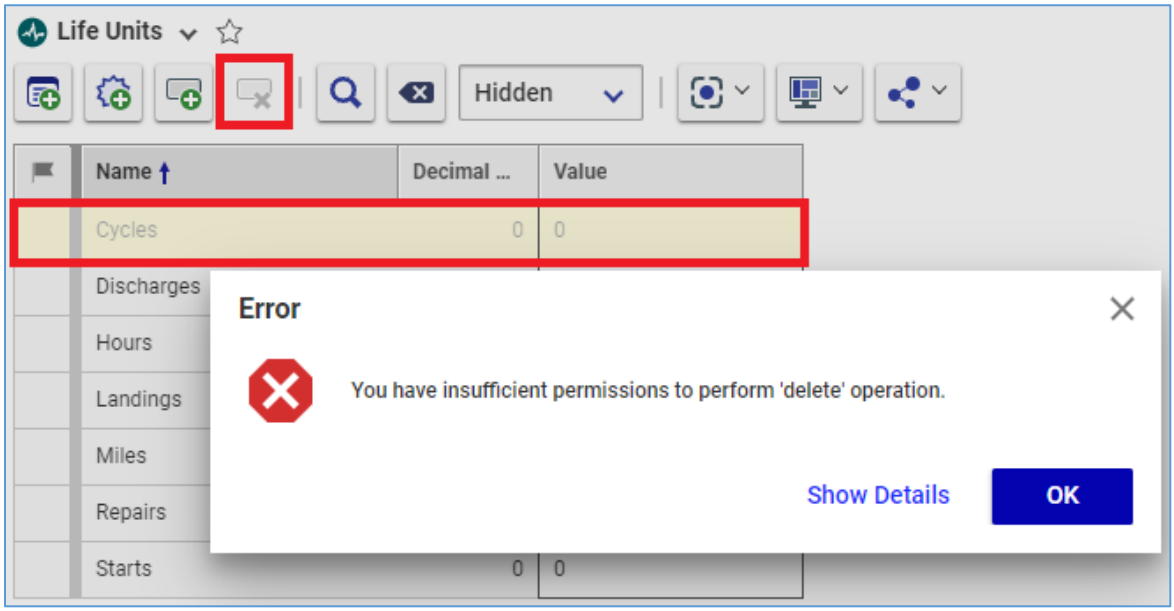


Figure 27.

Note: Avoid using the discussed Relationships toolbar buttons because canceling an unallowed change requires clicking the **Discard** button that will lead to canceling all unsaved updates of a given source Item.

When editing a source Item, the action menu of an automatically managed Relationship Item has the standard **Edit**, **Replace**, **Remove**, and **Claim** commands available.

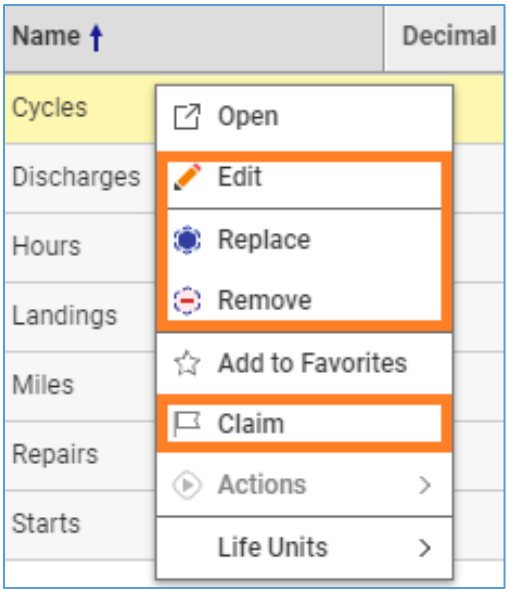


Figure 28.

Again, manual update of such Relationship Items is forbidden—an error is raised if either:

- To click the **Edit** command.

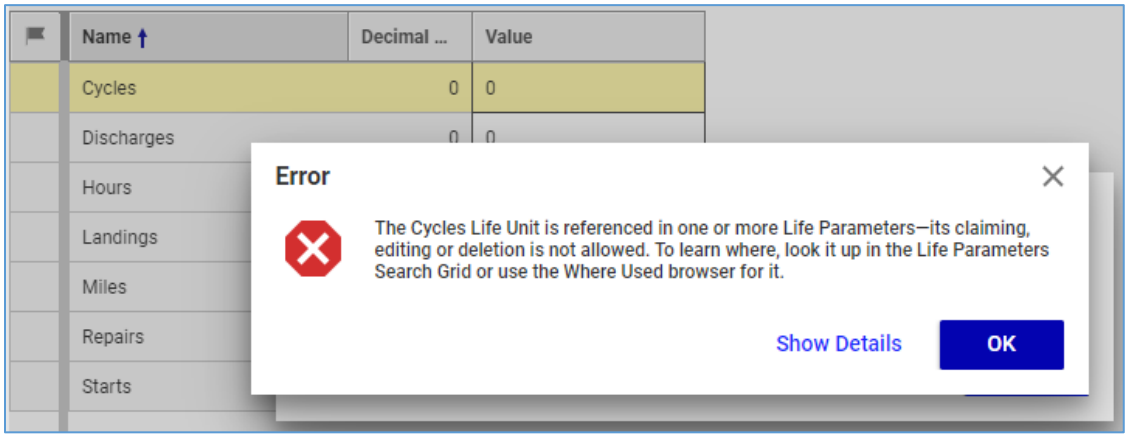


Figure 29.

- To save the source Item with a Relationship Item manually updated with the **Replace** command.

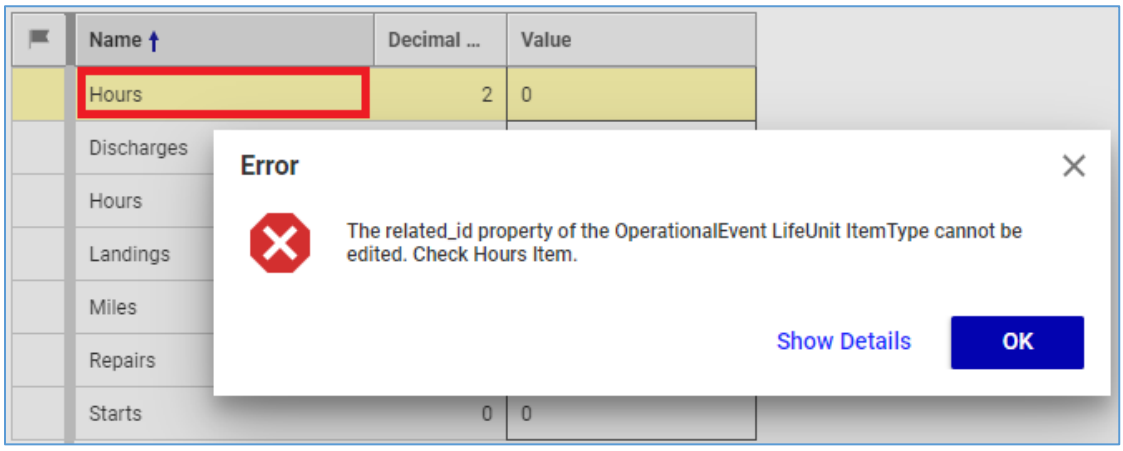


Figure 30.

- To save the source Item with a Relationship Item manually updated with the **Remove** command.

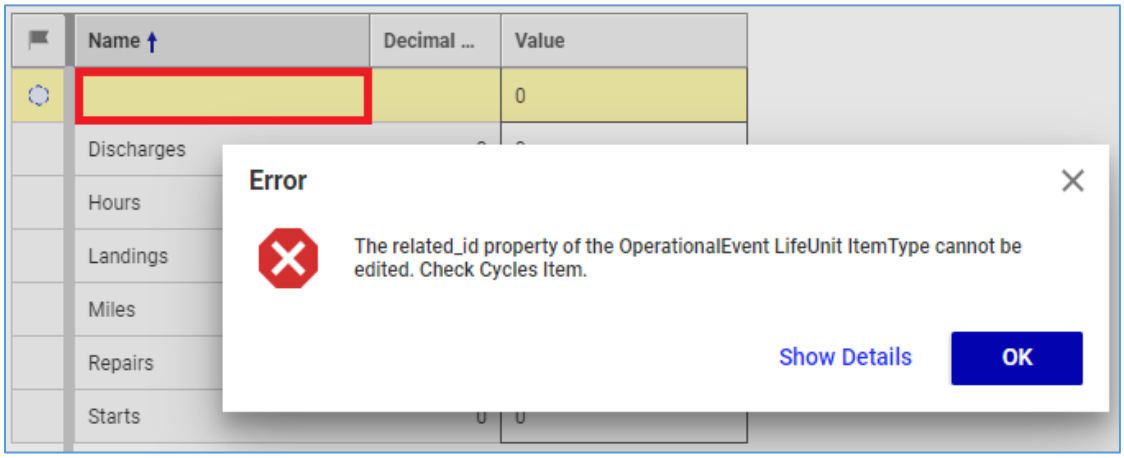


Figure 31.

- To click the **Claim** command.

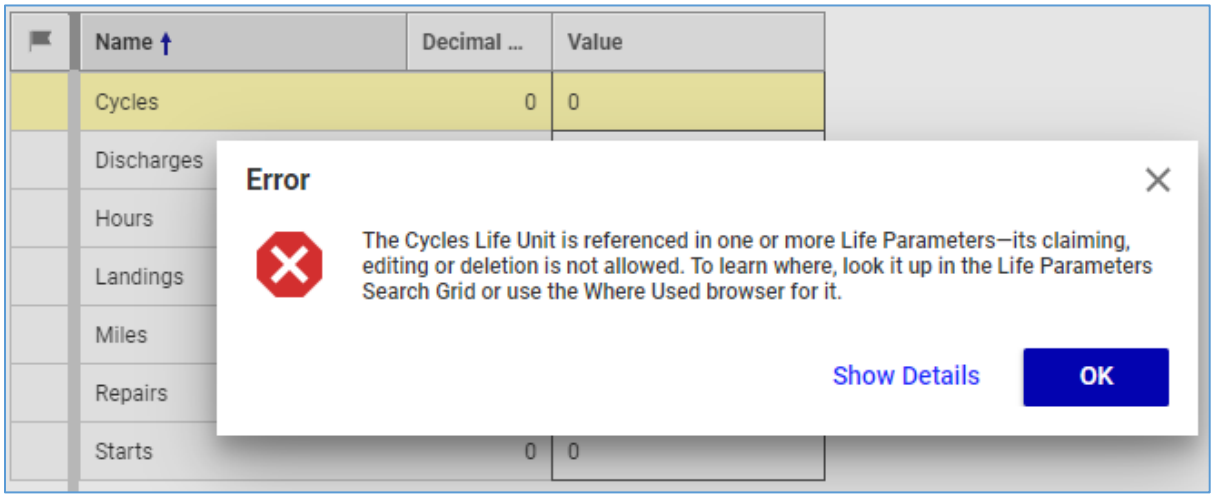


Figure 32.

Note: Avoid using the discussed action menu commands because canceling an unallowed change may require clicking the **Discard** button that will lead to canceling all unsaved updates of a given source Item.

4 Physical Parts

This section discusses in detail all the available features for working with the core Item of the DTC application—the **Physical Part** Item. Some of these features are available for specific Identities.

Since the DTC R2 application, a **Part Policy** Item is a prerequisite for a **Physical Part** Item. The **Part Policy** Item must exist for a **Part** Item from which the **Physical Part** Item will be created. For details on the **Part Policy** Items, see section 9 Part Policies. For details on using **Physical Part** Items created in DTC 12.0R1, see section 14.1 Migrating from DTC 12.0R1 to 12.0R3.

4.1 Understanding Physical Parts

An implementation of the digital twin concept in Aras Innovator is a **Physical Part** Item. It should represent an individual real-world asset (product) describing the asset properties as they currently exist. For example, the **Physical Part** Item can be a laptop or car of its current configuration, including all customization.

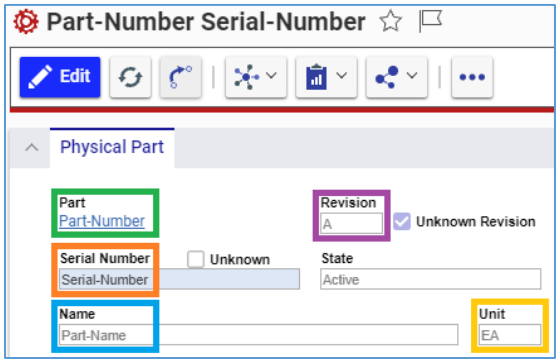


Figure 33.

4.1.1 Physical Parts and their Parts

To link an asset with its engineering design, a **Physical Part** Item is coupled with a corresponding **Part** Item.

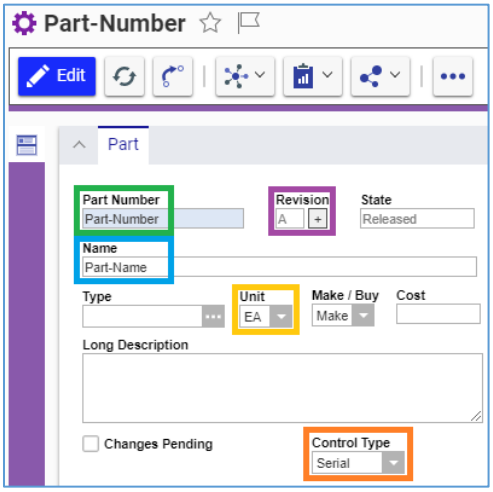


Figure 34.

A **Physical Part** Item can be derived only from a **Part** Item that has been **Released**, meaning that the physical assets have been produced according to the **Part** Item design. These **Part** Items can be in the following States:

- **Released**
- **In Change**
- **Superseded**
- **Obsolete**
- **Manual Change**

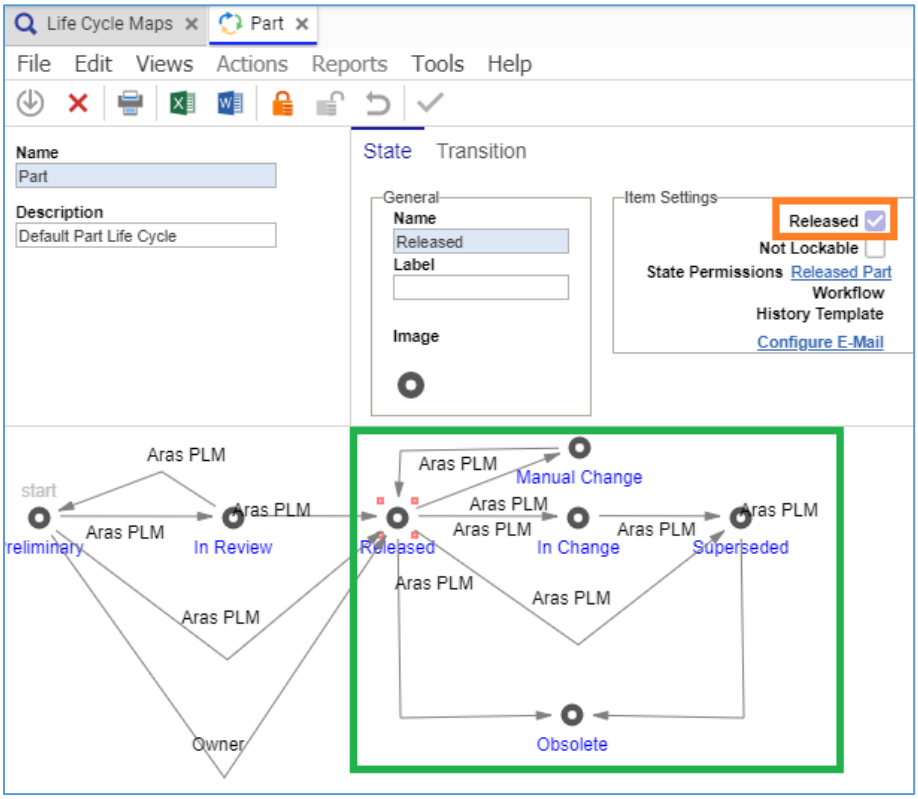


Figure 35.

Once the **Part** Item has been **Released**, its design is frozen as a Revision of this **Part** Item. Post-production design modifications are possible either through manual change or a change management process. If the modifications are approved for production and do not introduce assigning a new **Part Number** value to this Item, it gets another Revision. One **Part** Item can have one or more Revisions.

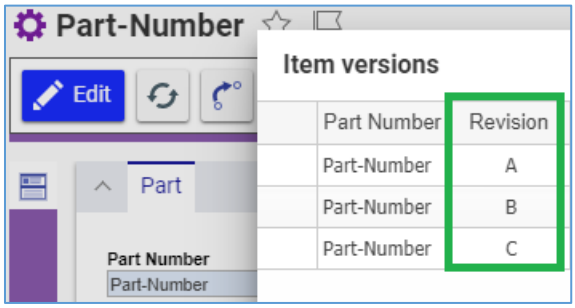


Figure 36.

The **Physical Part** Item can be associated with any **Part** Item Revision. By default, the latest **Part** Item Revision is set. You can edit this and select the necessary Revision.

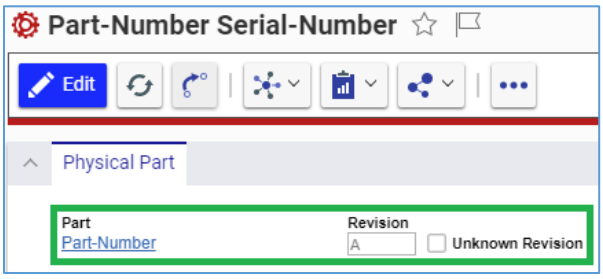


Figure 37.

If it is not possible to identify to which particular **Part** Item Revision the **Physical Part** Item belongs, you should select the latest released **Part** Item Revision along with the **Unknown Revision** check box. This check box is selected by default when creating a new **Physical Part** Item.

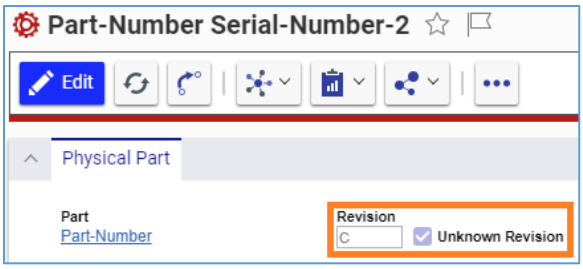


Figure 38.

The system does not allow the deletion of a **Part** Item Revision related to one or more **Physical Part** Items.

If the **Part** Item gets a new Revision, a **Physical Part** Item that originates from this **Part** Item remains related to the Revision given during this **Physical Part** Item creation.

The DTC application introduces the **Control Type** property to the **Part** Items for organizations where the engineering team may be responsible for a given **Part** Item inventory control type definition.

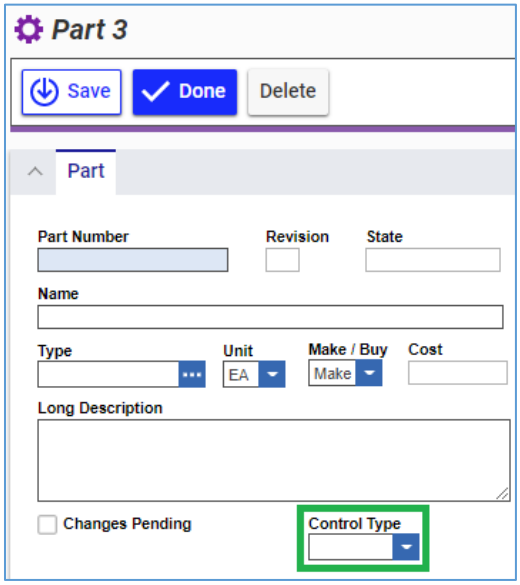


Figure 39.

This property determines how **Physical Part** Items derived from a given **Part** Item must be identified in the real-world inventory:

- Serial-controlled (serialized): **Serial**
- Lot-controlled: **Lot / Batch**
- Non-controlled: **No Control**

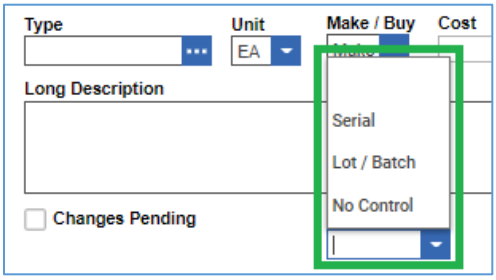


Figure 40.

The **Control Type** property is not required for the **Part** Items because they belong to another application—Product Engineering (PE). There are many existing Aras Innovator databases with legacy **Part** Items without this property.

There are no specific constraints for the **Part** Item **Control Type** property. The **Part** Item may not have its **Control Type** property set in any of its States, including **Released**. Or, the **Part** Item can have its **Control Type** value changed during its life cycle across the same or different revisions. For example, it was not clear to which Control Type the **Part** Item belongs at its early design stage, or the **Part** Item was first produced as a prototype.

Therefore, the **Control Type** property value of a given **Part** Item initially input by engineering becomes a default value for the **Control Type** property of a **Part Policy** Item that governs this **Part** Item. Normally, the **Part Control Type** value will be used by the **Part Policy** Item by default, but the logic does allow the **Part** and **Part Policy Control Type** values to differ: authorized users can change the **Part Policy** Item **Control Type** value during creation or editing.

Since DTC 12.0R3, a **Physical Part** Item is always connected to a **Part** Item given at the **Physical Part** Item creation. Nobody can reconnect the **Physical Part** Item to another **Part** Item in any **Physical Part** Item State. If trying to save the **Physical Part** Item with such a change, the system will not allow you to save this **Physical Part** Item and raise an error message.

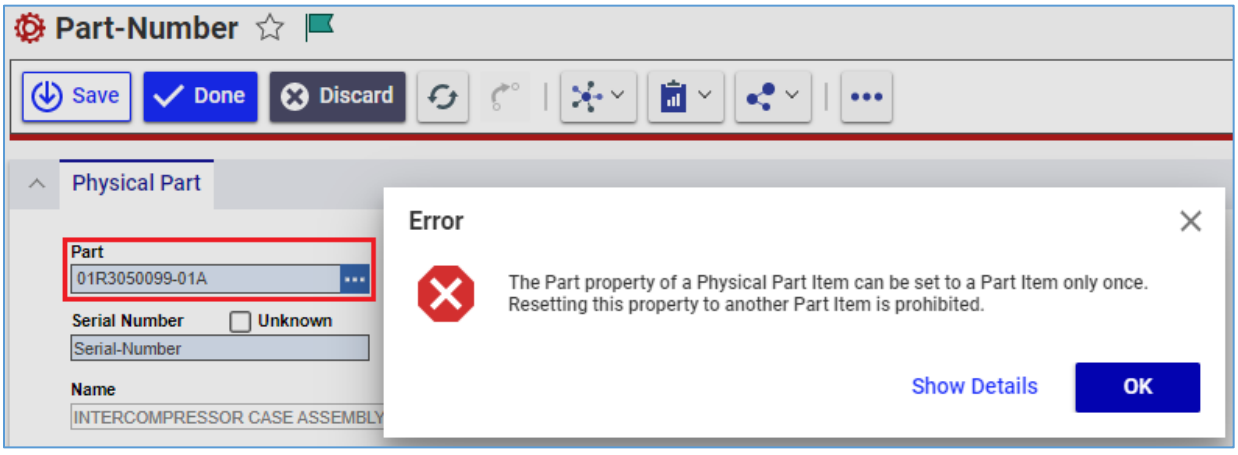


Figure 41.

4.1.2 Physical Parts and their Part Policies

Since the DTC 12.0R2 application version, a **Part Policy** Item is a prerequisite for a **Physical Part** Item. It must exist in the **Active** state for a **Part** Item from which the **Physical Part** Item will be created. Otherwise, an error is raised when saving the **Physical Part** Item after its **Part** property is set.

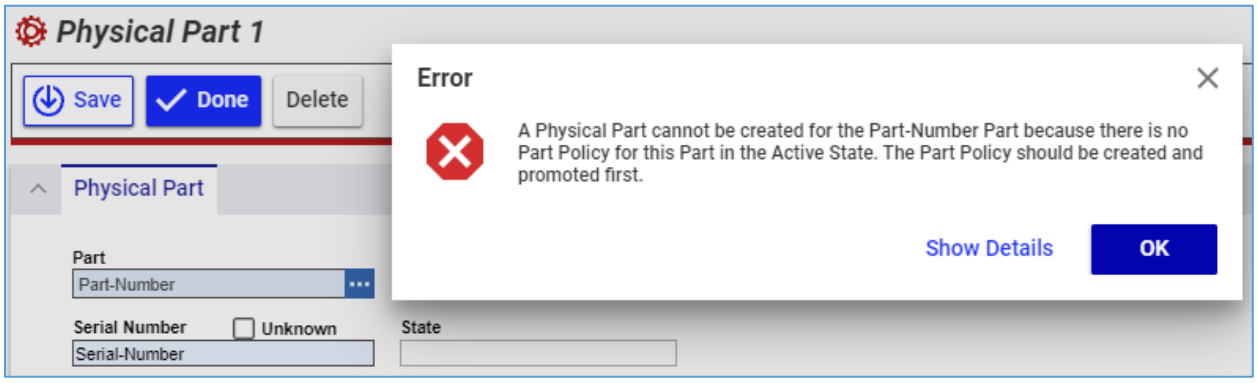


Figure 42.

The **Part Policy** Item is a set of rules and policies that govern the **Part** Item and, consequently, **Physical Part** Items originated from this **Part** Item. These rules and policies are frozen during the lives of these **Physical Part** Items: the **Part Policy** Item cannot be edited or modified when at least one of these **Physical Part** Items exists. For details, see section 9.1 Understanding Part Policies.

Since DTC 12.0R2, the **Part Policy** Item defines the **Control Type** for its governed **Physical Part** Items. See also section 9.4.1 Part Item Related to Part Policy Item.

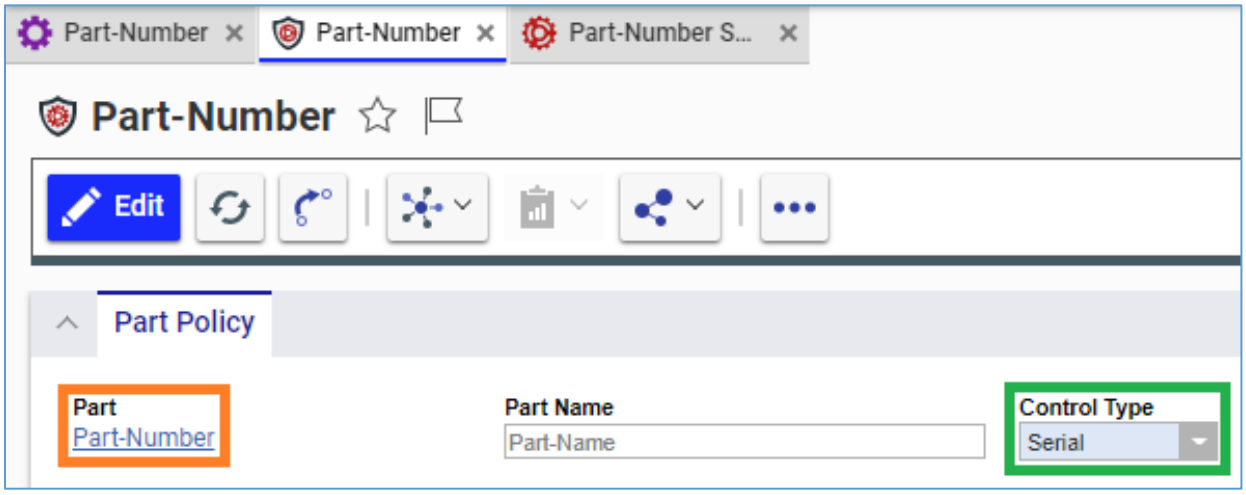


Figure 43.

The **Physical Part** Items get their **Control Type** values from the **Part Policy** Items, not from the **Part** Items. Therefore, all **Physical Part** Items with the same **Part Number** value are always of the same **Control Type** value.

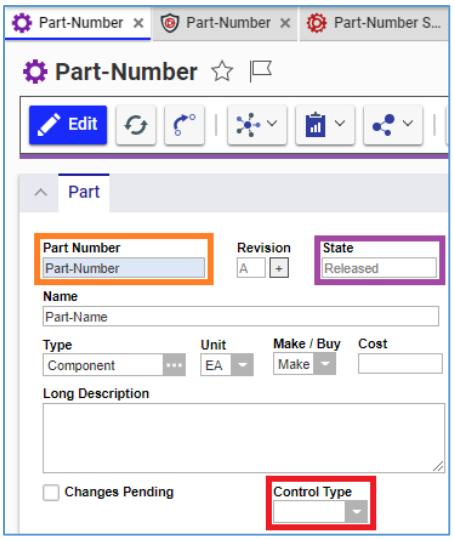


Figure 44.

Note: In DTC 12.0R1, the **Part Policy** Items do not exist, and the control type definition is on the **Part** Item. To start using **Physical Part** Items imported from 12.0R1, see section 14 Migrating from DTC 12.0R1 to 12.0R3.

In addition, the **Part Policy** Item can define tracking **Life Parameter** Items if it is necessary to track the life variables of its governed **Physical Part** Items. A set of such **Life Parameter** Items is a **Life Policy** Item applied to this **Part Policy** Item. For details, see section 9.4.3 Life Parameter Items Related to Part Policy Item.

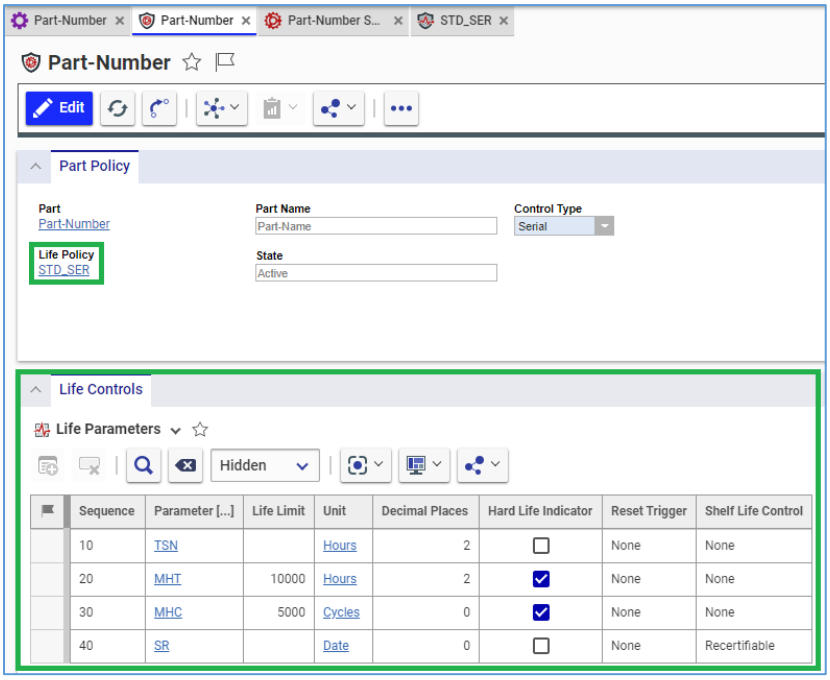


Figure 45.

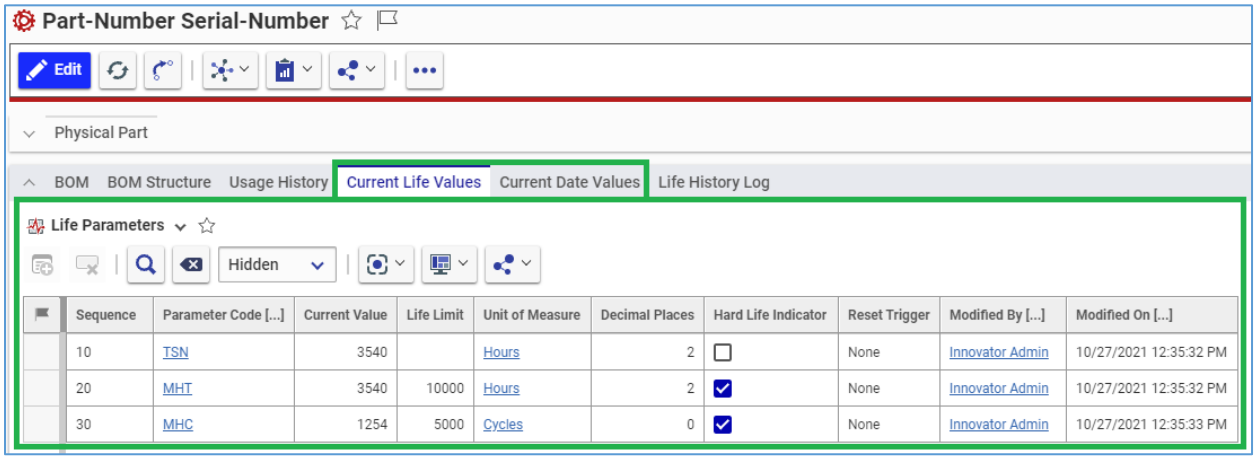
4.1.3 Physical Parts and their Life Parameters

Some characteristics of real-world assets may change during their operation. Some assets require tracking of specific characteristics that affect their operability. Such characteristics are also known as asset life variables. They tell when an asset should be inspected, repaired, overhauled, removed, and so on. For example, car mileage can indicate when a given car should have its brakes inspected, or its oil replaced, and other procedures done.

A **Life Parameter** Item represents a real-world operational (life) variable of an asset. For details, see section 7.1 Understanding Life Parameters.

If a **Physical Part** Item represents an asset with one or more life variables that should be tracked, its **Part Policy** Item should include a **Life Policy** Item with **Life Parameter** Items representing these life variables. For details, see section 9.4.2 Life Policy Item Applied to Part Policy Item.

The **Physical Part** Item keeps these life variables as **Life Parameter** Items given on its **Current Life Values** and **Current Date Values** Relationships tabs. For details, see section 4.7 Tracking Lives of Physical Parts.



Sequence	Parameter Code [...]	Current Value	Life Limit	Unit of Measure	Decimal Places	Hard Life Indicator	Reset Trigger	Modified By [...]	Modified On [...]
10	TSN	3540		Hours	2	☐	None	Innovator Admin	10/27/2021 12:35:32 PM
20	MHT	3540	10000	Hours	2	☑	None	Innovator Admin	10/27/2021 12:35:32 PM
30	MHC	1254	5000	Cycles	0	☑	None	Innovator Admin	10/27/2021 12:35:33 PM

Figure 46.

4.1.4 Physical Parts and their Operational Events

Your organization can track and update life variables of a single asset by operational activities that this asset has completed. An operational activity consumes some piece of the asset life, changing the asset life variables. For example, an asset can be an aircraft with such life variables as the number of landings, engine starts, flying (working) hours, and so on. An aircraft regular operational activity is a flight. An aircraft operator should update the aircraft life variables by the flights made by the aircraft. One regular flight adds one landing, one engine start, some flying (working) hours, and so on.

An **Operational Event** Item digitally represents a discrete in-field operational activity that a single real-world asset has performed over some time. For details, see section 11.1 Understanding Operational Events.

If a **Physical Part** Item represents a single asset that performs operational activities, the **Operational Event** Items representing these activities should be applied to this **Physical Part** Item. For details, see section 4.7.2.3 Operational Event update of Life Values.

The **Physical Part** Item keeps the **Operational Event** Items applied to it on its **Life History Log** Relationships tab. For details, see section 4.7.4 Tracking history of Numeric Life Variables.

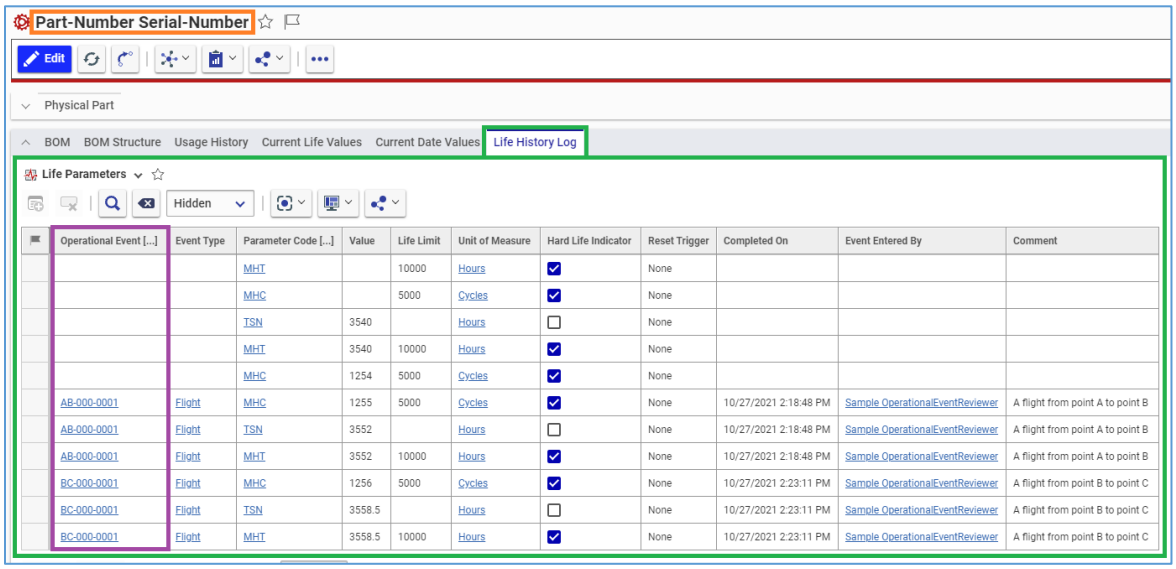


Figure 47.

You can also search **Operational Event** Items applied to a **Physical Part** Item using this **Physical Part** Item as a filter criterion in the **Operational Events** search grid.

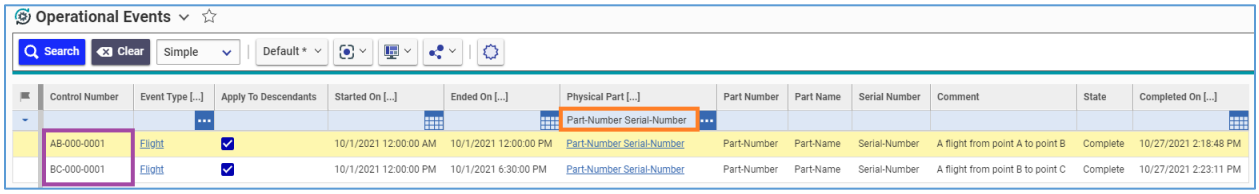


Figure 48.

4.1.5 Serial-controlled Physical Parts

A **Part Policy** Item of the **Serial** Control Type requires its governed **Physical Part** Items to be serial-controlled (serialized). Such **Physical Part** Items represent physical assets that are identified in inventory by serial numbers. Each of such assets must have one unique serial number. One serial-controlled **Physical Part** Item represents one single real-world asset.

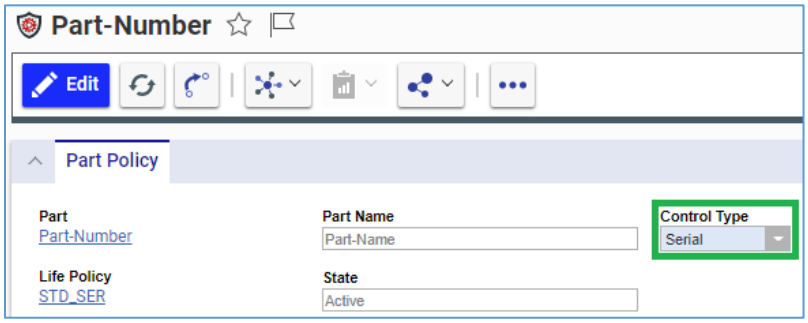


Figure 49.

A non-serialized **Physical Part** Item is a term for a physical asset that is not under serial number inventory control: it is either lot-controlled or non-controlled.

A serial number is a unique identifier of an individual real-world asset against a series of similar assets. One serial number can belong only to one asset among the given asset series. Rules for assigning and naming serial numbers depend on an organization, industry, standards, legal regulations, etc. For example, some ISO standards define the vehicle identification number (VIN) rules for the automotive industry worldwide. But each automotive component manufacturer has their own rules for their car part serial numbers.

A serialized **Physical Part** Item uses the **Serial Number** property to keep a serial number of a represented physical asset. This property can have a sequence of up to 32 characters of any type except for white space characters. The system removes white spaces from a serial number entered or modified in this property upon saving the **Physical Part** Item.

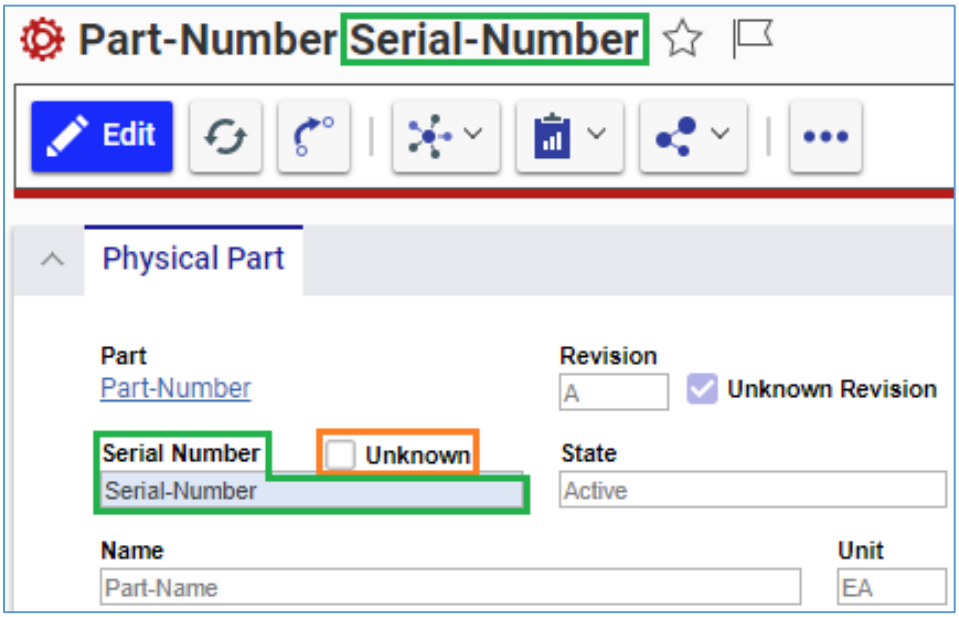


Figure 50.

The **Unknown** check box accompanies the **Serial Number** field on the **Physical Part** Item form.

A serial-controlled **Physical Part** Item uses the **Unknown** Boolean property to indicate whether your organization knows or not a serial number given to a represented real-world asset—whether a value entered in the **Serial Number** property matches or not the serial number that the asset actually bears:

- **Cleared** (**false**): your organization knows it—the entered value exactly matches the serial number on the asset.
- **Selected** (**true**): your organization does not know it—the entered value does not match somewhat or at all the serial number on the asset. An asset may have it lost partially or entirely. Or, your organization cannot possess information about such serial numbers because they belong to third parties.

The **Unknown** property is singularly useful in the **Physical Parts** Search Grid for locating **Physical Part** Items whose serial IDs are not known.

When creating a new serial-controlled **Physical Part** Item, the **Unknown** property is set to **false** by default. If having the editing rights, you can reset it to **true** or **false**. You can also promote a **Physical Part** Item from the **Preliminary** to **Active** State regardless of the **Unknown** property value.

Your organization can configure the DTC application to use Sequences for populating the **Serial Number** property with a dummy system-generated value when not knowing an actual serial number of an asset.

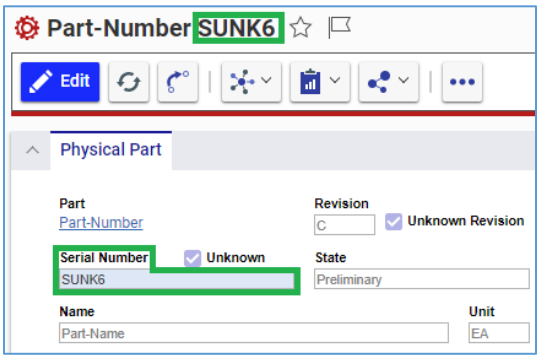


Figure 51.

The out-of-the-box Sequence template for **Serial Number** property values is the following:

SUNK#

where:

SUNK—short for **S**erial **UNK**nown—a prefix constant for all generated values

#—an integer number incremented by **1** for each newly generated value

In this configuration, when you save a serial-controlled **Physical Part** Item in any State with the **Serial Number** property being empty and the **Unknown** property being **true**, the server side automatically populates this empty **Serial Number** property with the next value generated from the Sequence.

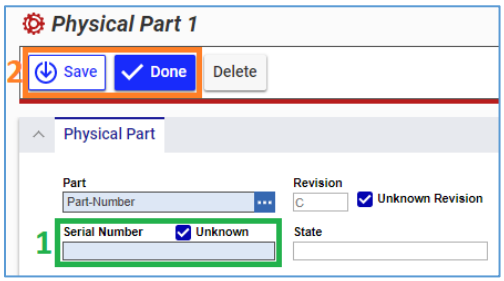


Figure 52.

This Sequence template is global: the system outputs values according to it for **Physical Part** Items regardless of their associated **Part** Item. **Physical Part** Items rooted in different **Part** Items can have dummy **Serial Number** values generated by the discussed Sequence.

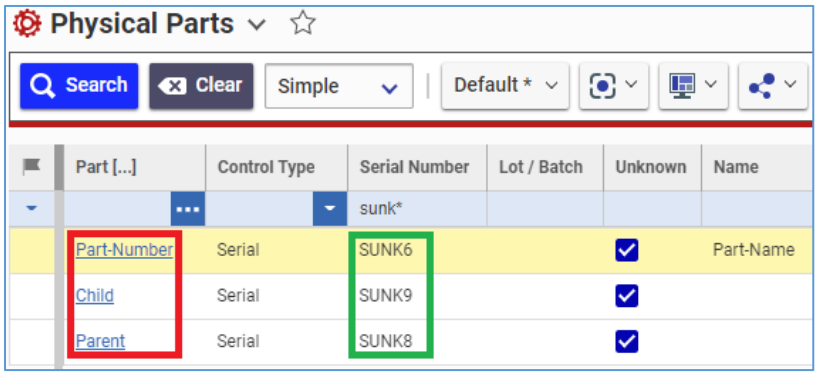


Figure 53.

Your organization can change this default template as they consider necessary.

You can modify or delete a Sequence-generated **Serial Number** property value when Permissions allow you to edit a **Physical Part** Item. If you save a **Physical Part** Item after deleting a Sequence-generated value from the **Serial Number** property while keeping this property empty and **Unknown true**, the system will populate it with the next value from the Sequence.

In the default out-of-the-box DTC 12.0R3 configuration, each **Physical Part** Item, regardless of its State, must have a unique **Serial Number** property value across all **Physical Part** Items originated from one given **Part** Item. An error is raised when trying to save a **Physical Part** Item not meeting this requirement.

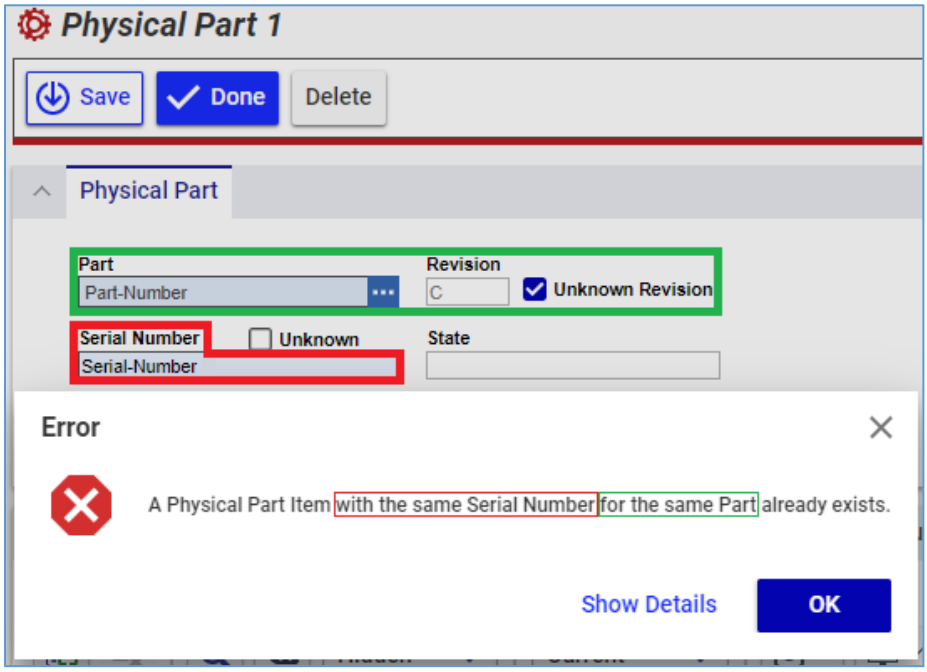


Figure 54.

Your organization can switch off this uniqueness if it is necessary to allow serial number duplicates in the system. In this case, more than one serial-controlled **Physical Part** Items derived from the same **Part** Item can have the same **Serial Number** property value regardless of their State.

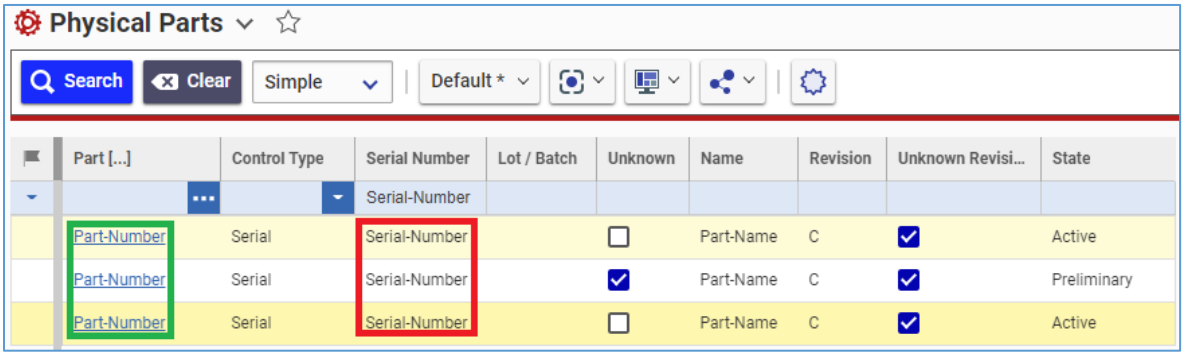


Figure 55.

The on or off uniqueness configuration is global: it affects all **Physical Part** Items in the system regardless of their properties. The **Unknown** or any other property or Relationship Item does not define whether a given **Serial Number** property value should or should not be unique.

The Sequence populating the **Serial Number** property is always used with the uniqueness switched on. Only one **Physical Part** Item can have the same Sequence-generated **Serial Number** value for the same **Part** Item.

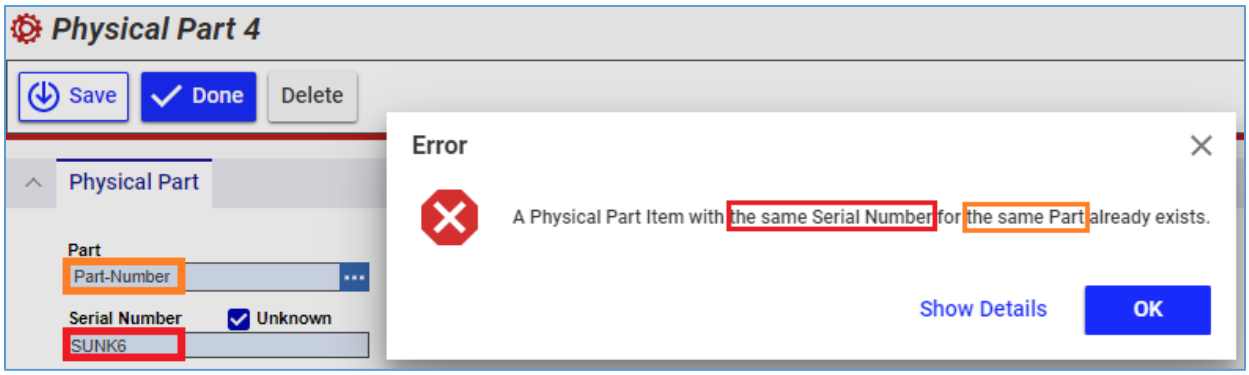


Figure 56.

When a **Physical Part** Item is in the **Preliminary** state, its **Serial Number** property can be empty regardless of the global serial-number uniqueness configuration or its **Unknown** property value setting. This case does not mean that the represented physical asset has no serial number. It means that the exact serial number value has not been provided yet.

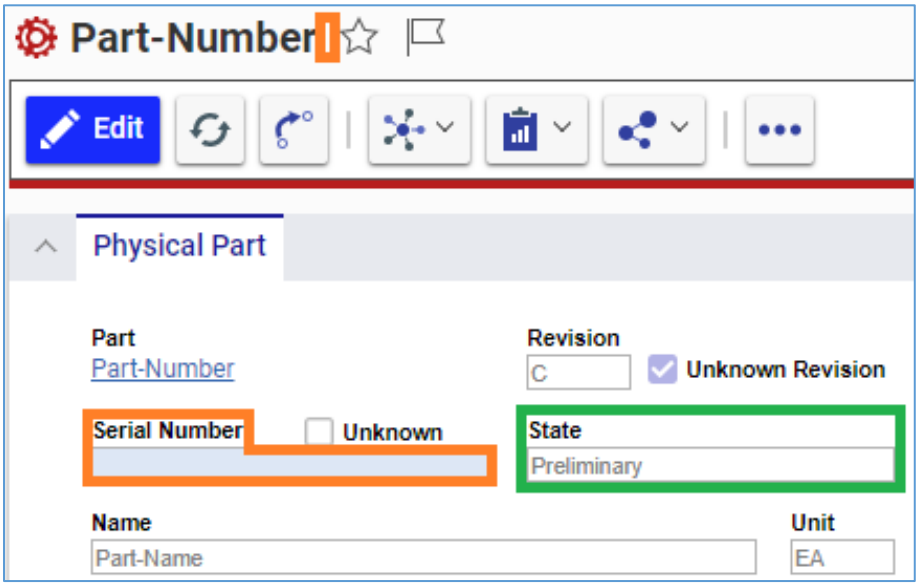


Figure 57.

The **Serial Number** property is required for the **Active** State. Even when you have rights, you cannot promote a **Physical Part** Item to the **Active** State if its **Serial Number** property has no value. Also, the server does not allow you to save a **Physical Part** Item in the **Active** State with its **Serial Number** property being blank. An error is raised in such cases.

4.1.6 Lot-controlled Physical Parts

A **Part Policy** Item of the **Lot / Batch** Control Type requires its governed **Physical Part** Items to be lot-controlled. Such **Physical Part** Items represent physical assets that are identified in inventory by lot (batch) numbers. Each of these assets must have one lot (batch) number. One lot-controlled **Physical Part** Item represents one distinct set of many similar real-world assets as a whole, not as a single asset or some quantity from this set. For example, a lot-controlled **Physical Part** Item can represent all bolts

with the same lot number, not one or some bolts with the same lot number. Or, one lot-controlled **Physical Part** Item is all oil of the same batch but not some oil quantity from that batch.

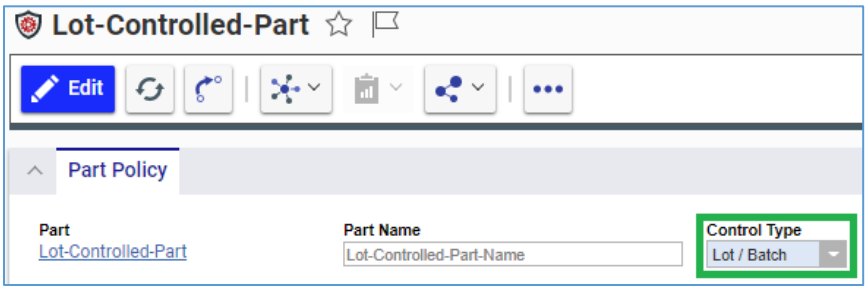


Figure 58.

A non-lot-controlled **Physical Part** Item is a term for a physical asset that is not under lot (batch) number inventory control: it is either serial-controlled or non-controlled.

A lot (batch) number is a unique identifier for a particular set (lot, batch, quantity) of real-world assets against a series of similar assets. One lot number can belong to one or many assets among the given asset series. Rules for assigning and naming lot numbers depend on an organization, industry, standards, legal regulations, etc. For example, drug regulators administer how drug lot numbers should be assigned. But each drug manufacturer can have their own rules for their drug lot number templates.

A lot-controlled **Physical Part** Item uses the **Lot / Batch** property to keep a lot number of a represented physical asset set. This field can have a sequence of up to 32 characters of any type except for the white space characters. The system removes white spaces from a lot or batch number entered or modified in this field upon saving the **Physical Part** Item.

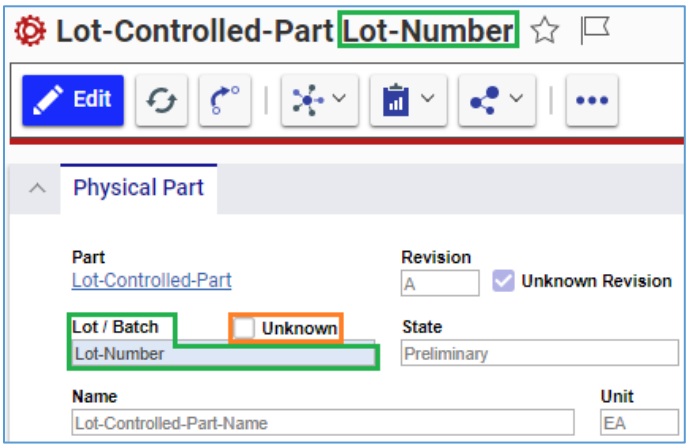


Figure 59.

The **Unknown** check box accompanies the **Lot / Batch** field on the **Physical Part** Item form.

A lot-controlled **Physical Part** Item uses the **Unknown** Boolean property to indicate whether your organization knows or not a lot/batch number given to a represented real-world asset set—whether a value entered in the **Lot / Batch** property matches or not the lot/batch number that the assets actually bear or more generally as the asset certification documentation records display:

- **Cleared** (**false**): your organization knows it—the entered value exactly matches the lot/batch number on the assets, their documentation, or packaging.
- **Selected** (**true**): your organization does not know it—the entered value does not match somewhat or at all the lot/batch number of the asset. An asset may have it lost partially or entirely, or your organization cannot possess information about such lot/batch numbers because they belong to third parties.

The **Unknown** property is singularly useful in the **Physical Parts** Search Grid for locating **Physical Part** Items whose lot IDs are not known.

When creating a new lot-controlled **Physical Part** Item, the **Unknown** property is set to **false** by default. If having the editing rights, you can reset it to **true** or **false**. You can also promote a **Physical Part** Item from the **Preliminary** to **Active** State regardless of the **Unknown** property value.

Your organization can configure the DTC application to use Sequences for populating the **Lot / Batch** property with a dummy system-generated value when not knowing an actual lot/batch number of an asset set.

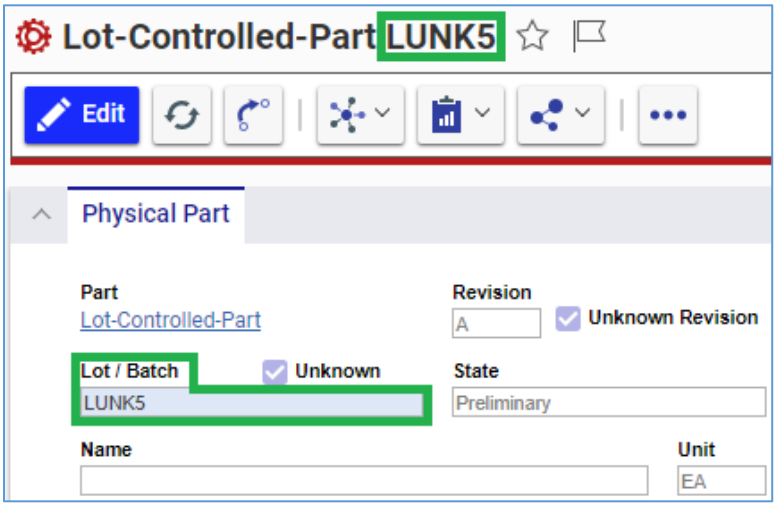


Figure 60.

The out-of-the-box Sequence template for **Lot / Batch** property values is the following:

LUNK#

where:

LUNK—short for **L**ot **UNK**nown—a prefix constant for all generated values

#—an integer number incremented by **1** for each newly generated value

In this configuration, when you save a lot-controlled **Physical Part** Item in any State with the **Lot / Batch** property being empty and the **Unknown** property being **true**, the server side automatically populates this empty **Lot / Batch** property with the next value generated from the Sequence.

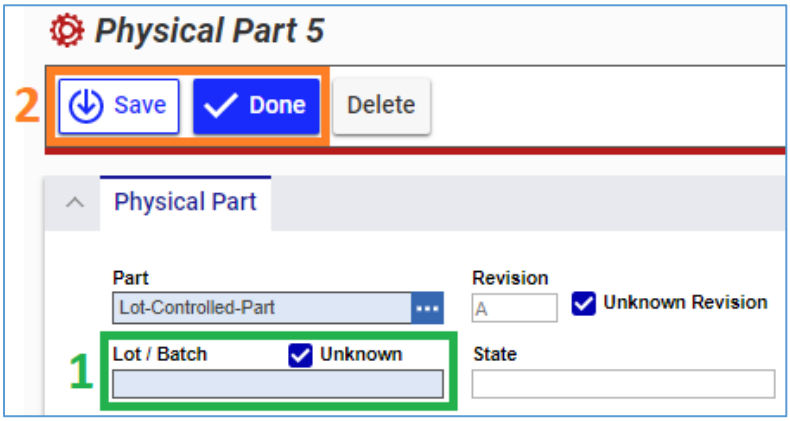


Figure 61.

This Sequence template is global: the system outputs values according to it for **Physical Part** Items regardless of their associated **Part** Item. **Physical Part** Items rooted in different **Part** Items can have dummy **Lot / Batch** values generated by the discussed Sequence.

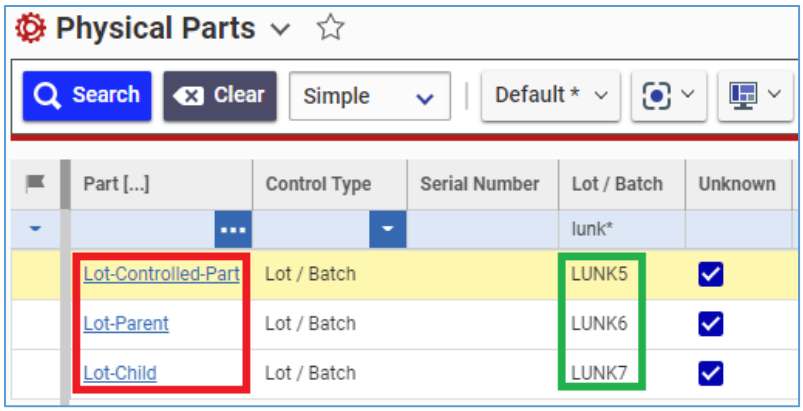


Figure 62.

Your organization can change this default template as they consider necessary.

You can modify or delete a Sequence-generated **Lot / Batch** property value when Permissions allow you to edit a **Physical Part** Item. If you save a **Physical Part** Item after deleting a Sequence-generated value from the **Lot / Batch** property while keeping this property empty and **Unknown true**, the system will populate it with the next value from the Sequence.

In the default out-of-the-box DTC 12.0R3 configuration, each **Physical Part** Item, regardless of its State, must have a unique **Lot / Batch** property value across all **Physical Part** Items originated from one given **Part** Item. An error is raised when trying to save a **Physical Part** Item not meeting this requirement.

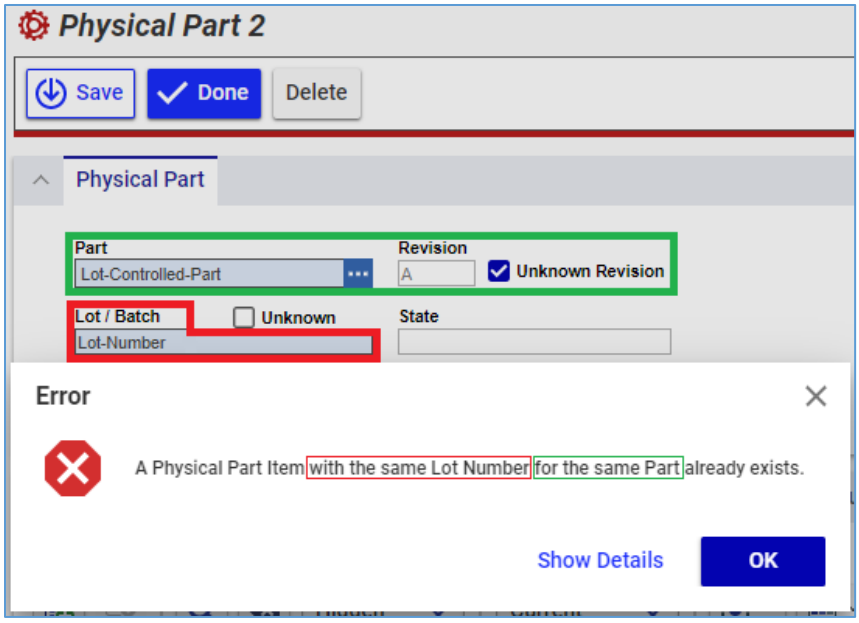


Figure 63.

Warning Before DTC 12.0R3, more than one **Physical Part** Items originating from one **Part** Item were allowed to have the same **Lot / Batch** property value. If migrating from a lower DTC version, ensure that the imported legacy database does not have multiple **Physical Part** Items with the same **Lot / Batch** value and **Part** Item.

Your organization can switch off this uniqueness if it is necessary to allow lot/batch number duplicates in the system. In this case, more than one lot-controlled **Physical Part** Items derived from the same **Part** Item can have the same **Lot / Batch** property value regardless of their State.

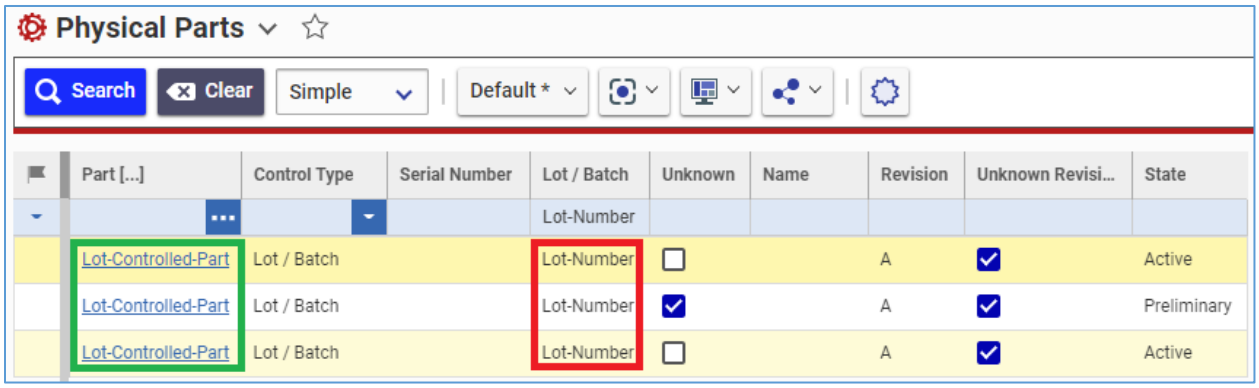


Figure 64.

The on or off uniqueness configuration is global: it affects all **Physical Part** Items in the system regardless of their properties. The **Unknown** or any other property or Relationship Item does not define whether a given **Lot / Batch** property value should or should not be unique.

The Sequence populating the **Lot / Batch** property is always used with the uniqueness switched on. Only one **Physical Part** Item can have the same Sequence-generated **Lot / Batch** value for the same **Part** Item.

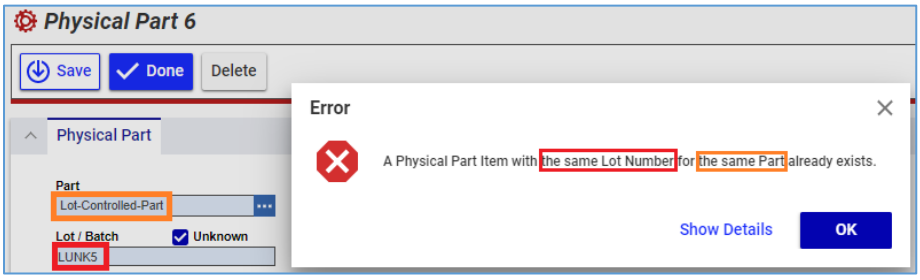


Figure 65.

When a **Physical Part** Item is in the **Preliminary** state, its **Lot / Batch** property can be empty regardless of the global lot-number uniqueness configuration or its **Unknown** property value setting. This case does not mean that the represented physical asset has no lot or batch number. It means that the exact lot or batch number value has not been provided yet.

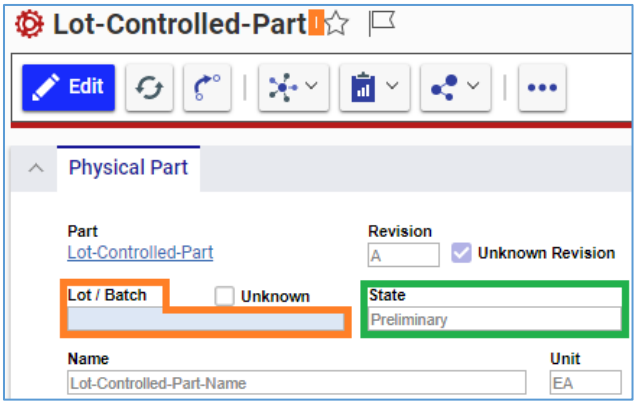


Figure 66.

The **Lot / Batch** property is required for the **Active** State. Even when you have rights, you cannot promote a **Physical Part** Item to the **Active** State if its **Lot / Batch** property has no value. Also, the server does not allow you to save a **Physical Part** Item in the **Active** State with its **Lot / Batch** property being blank. An error is raised in such cases.

4.1.7 Non-controlled Physical Parts

A **Part Policy** Item of the **No Control** Control Type requires its governed **Physical Part** Items to be non-controlled. These **Physical Part** Items represent physical assets that have no inventory identifier. Each asset has neither serial nor lot (batch) numbers. One non-controlled **Physical Part** Item represents all similar real-world assets as a whole, not as a single asset or some quantity from this asset kind. For example, a non-controlled **Physical Part** Item can represent all washes with the same part number, not one or some washers with the same part number.

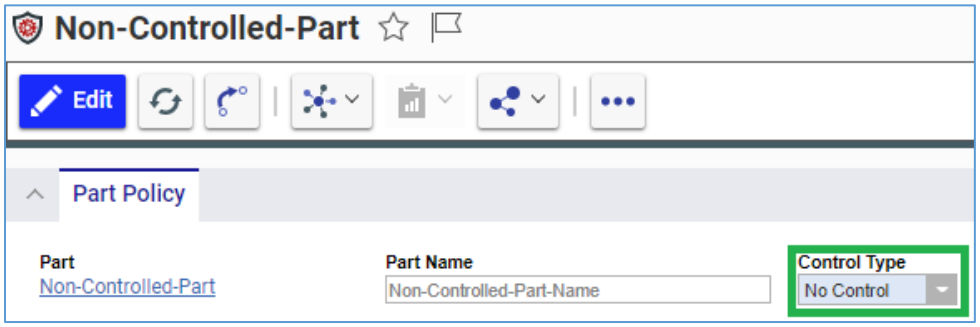


Figure 67.

A controlled **Physical Part** Item is a term for a physical asset that is under inventory control by an identifier: it is either serial-controlled or lot-controlled.

There can be one or many assets without identifiers for the same asset product. For example, many identical nuts and bolts can be used in many assemblies.

Across all Revisions of one **No Control Part** Item, one non-controlled **Physical Part** Item can exist. An error is raised when trying to save a **Physical Part** Item not meeting this requirement.

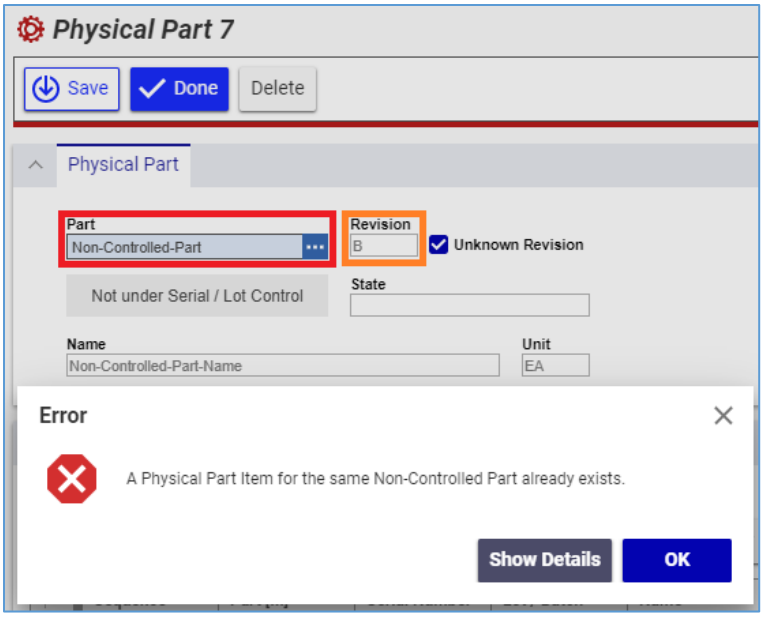


Figure 68.

A non-controlled **Physical Part** Item has the **Not under Serial/Lot Control** label instead of a field for a serial (**Serial Number**) or lot/batch number (**Lot / Batch**). It has no **Unkown** check box. It is not possible to provide any identifier for such an Item.

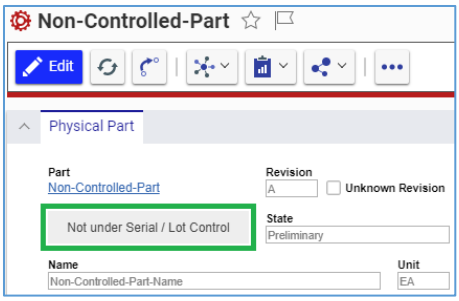


Figure 69.

4.2 Physical Parts Search Grid

The **Physical Parts** Search Grid has the standard features and the following columns:

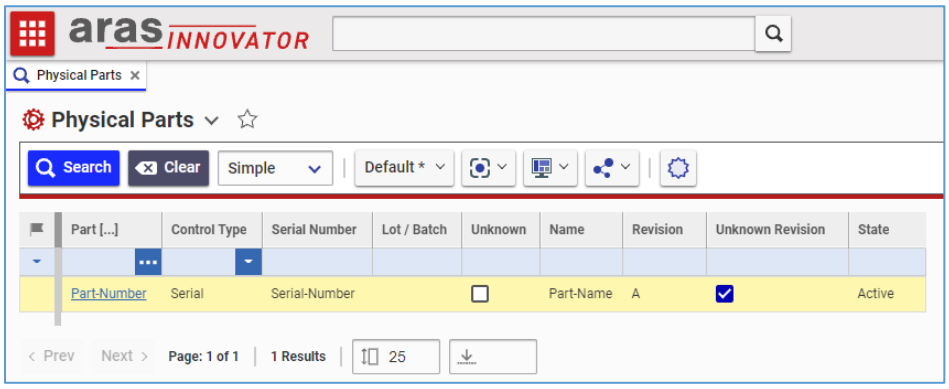


Figure 70.

- **Part**
- **Control Type**
- **Serial Number**
- **Lot / Batch**
- **Unknown**
- **Name**
- **Revision**
- **Unknown Revision**
- **State**

Each column shows the corresponding **Physical Part** Item property. The **Part** column provides a link to a related **Part** Item, not to the **Physical Part** Item. For the property details, see section *4.3 Physical Part properties*.

Note: In the **Part** column, the **Physical Part** Items are sorted by globally unique identifiers (GUID) of the **Part** Items, not by the **Part Numbers**.

4.3 Physical Part properties

The following properties of a **Physical Part** Item are available at the **Form** accordion of this **Physical Part** Item view:

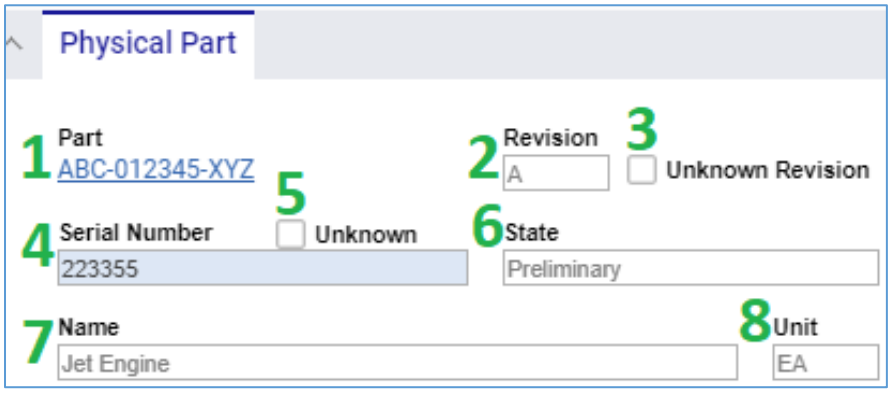


Figure 71.

1. **Part**: a **Part Number** of a related **Part** Item. It is also a link to this **Part** Item. This is the only required property for a **Physical Part** Item.

   The **Part** property is not resettable: it can be set only once during the **Physical Part** Item creation and cannot be reset to other **Part** Item after the **Physical Part** Item has been saved for the first time.

2. **Revision**: a view-only **Revision** of the selected **Part** Item (1).

3. **Unknown Revision**: a flag indicating when **selected** that it is unknown to which specific **Part** Item Revision the given **Physical Part** Item belongs. It is **selected** by default when creating a new **Physical Part** Item.

4. One of the following properties is shown depending on the **Part** Item **Control Type** (1):

   a. **Serial Number**: a serial number of the **Physical Part** Item if it is serialized. See section *4.1.5 Serial-controlled Parts*.

   b. **Lot/Batch**: a lot (batch) number of the **Physical Part** Item if it is lot-controlled. See section *4.1.6 Lot-controlled Parts*.

   c. **Not under Serial/Lot Control**: a label indicating that the **Physical Part** Item is non-controlled. See section *4.1.7 Non-controlled Parts*.

5. **Unknown**: a flag to indicate whether your organization knows or not an inventory ID (serial or lot number) given to a represented real-world asset—whether a value entered in the **Serial Number** (4.a) or **Lot/Batch** (4.b) property matches or not the serial or lot number that the asset actually bears:

   a. **Selected** (**true**): an organization does not know an inventory ID—a given value does not match an actual one.

   b. **Cleared** (**false**): an organization knows an inventory ID—a given value matches an actual one.

   The **Unknown** check box is **cleared** by default when creating a new **Physical Part** Item. It can be **false** or **true** in the **Preliminary** State. The **Physical Part** Item can be promoted from the **Preliminary** to **Active** State with any Boolean value in the **Unknown** property. In the **Active** State, the **Unknown** property can be toggled only from **true** to **false**, and if switching it from **false** to **true**, the server does not allow to save the **Physical Part** Item and raises an error message.

The **Unknown** check box accompanies the **Serial Number** (4.a) or **Lot/Batch** (4.b) field.

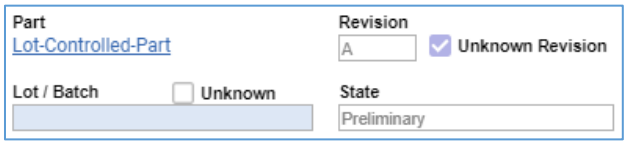


Figure 72.

The **Unknown** check box is not present on the form with the **Not under Serial/Lot Control** (4.c) label as being not applicable to non-controlled **Physical Part** Items.

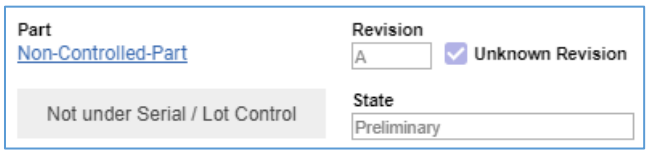


Figure 73.

6. **State**: the current State of the **Physical Part** Item. See section *4.5 Physical Part Life Cycle and Promotion*.
7. **Name**: a view-only **Name** of the selected **Part** Item (1).
8. **Unit**: a view-only unit of measure for the selected **Part** Item (1).

4.4 Physical Part Relationships and Usage

A **Physical Part** Item can have Relationship Items with other DTC Items. For these Relationship Items, the **Relationships** accordion contains the following Relationship tabs:

- **BOM**: a single-level BOM editor to view, edit, and manage immediate components of the given **Physical Part** Item. For more details, see section *5.2 Single-level Physical Part BOMs*.

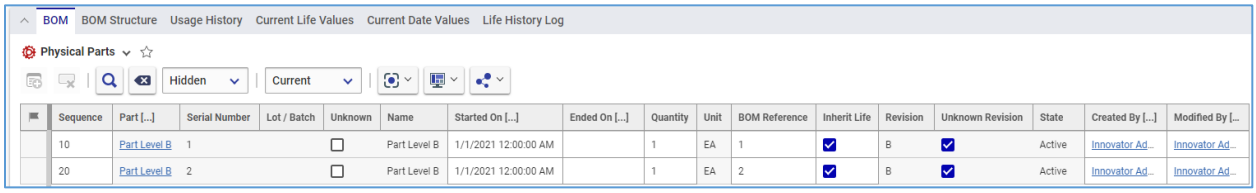


Figure 74.

- **BOM Structure**: a multi-level, read-only tree view of a Physical Part BOM structure below the given **Physical Part** Item. For more details, see section *5.3 Multi-level Physical Part BOM structures*.

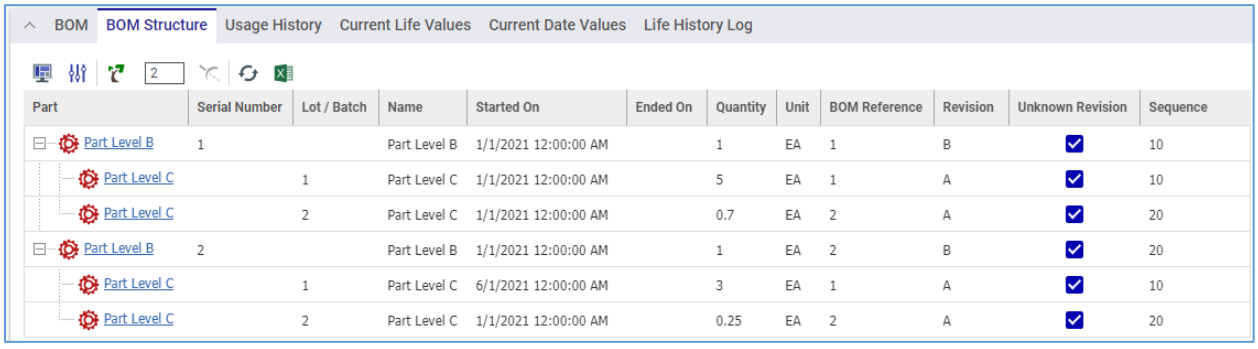


Figure 75.

- **Usage History**: a read-only grid of **Physical Part** Items where the given **Physical Part** Item has been an immediate component (child). For more details, see section *5.4 Physical Part Usage History*.

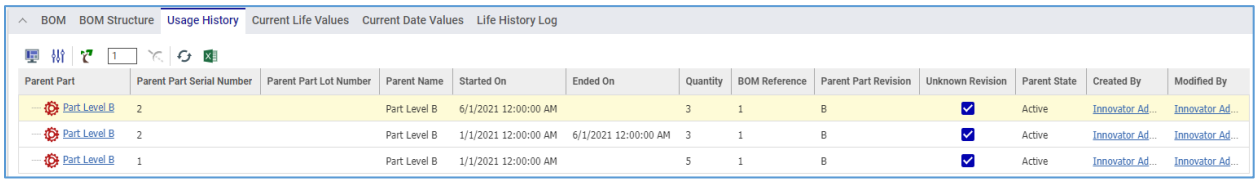


Figure 76.

- **Current Life Values**: an editor to view and edit the current values of the numeric life variables of the given **Physical Part** Item. For more details, see section 4.7.2 Tracking Numeric Life Variables of Physical Parts.

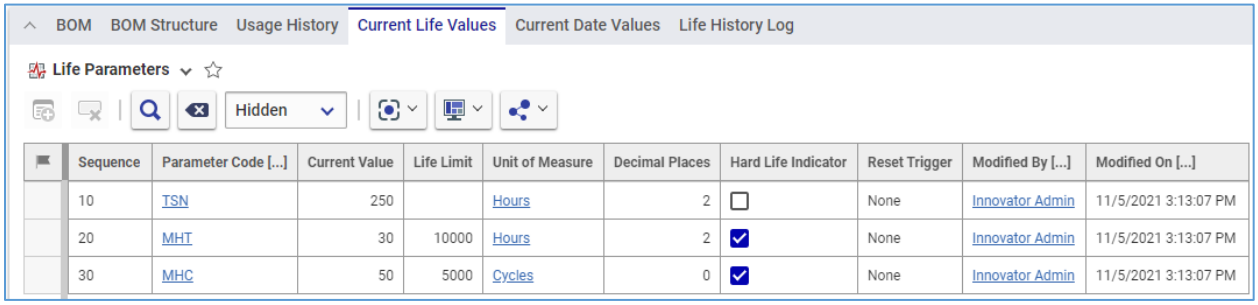


Figure 77.

- **Current Date Values**: an editor to view and edit the current life dates of the given **Physical Part** Item. For more details, see section 4.7.3 Tracking Life Dates of Physical Parts.

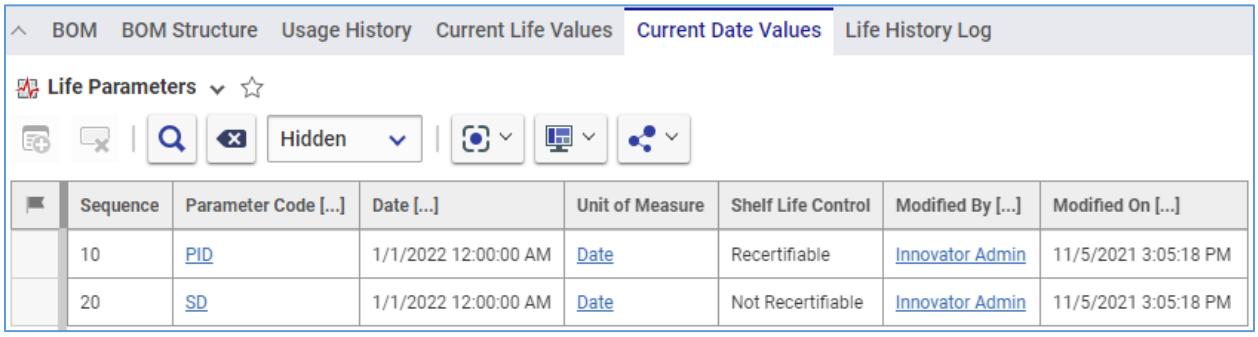
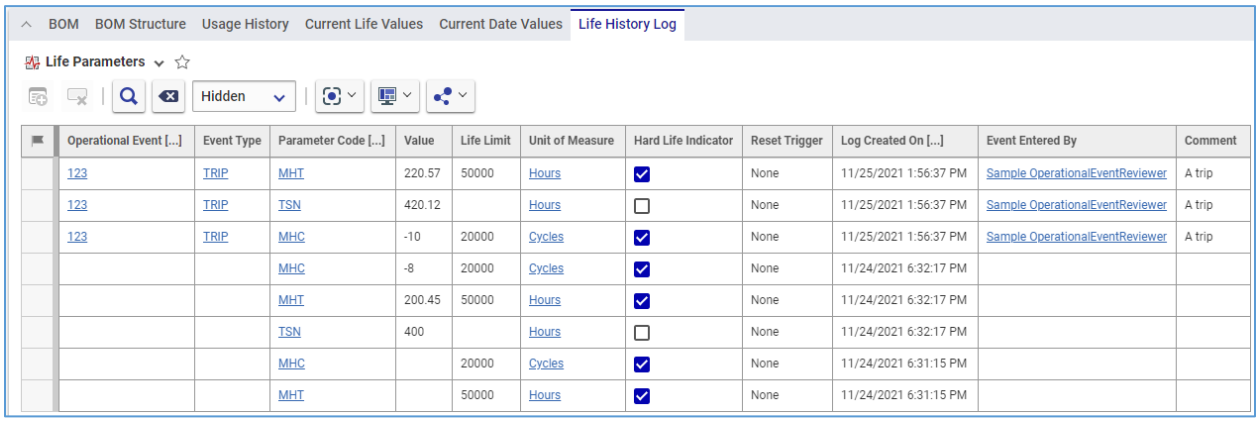


Figure 78.

- **Life History Log**: a read-only grid with the history of the numeric life variable values of the given **Physical Part** Item. For more details, see section 4.7.4 History of numeric Physical Part life variables.



Operational Event [...]	Event Type	Parameter Code [...]	Value	Life Limit	Unit of Measure	Hard Life Indicator	Reset Trigger	Log Created On [...]	Event Entered By	Comment
123	TRIP	MHT	220.57	50000	Hours	☑	None	11/25/2021 1:56:37 PM	Sample OperationalEventReviewer	A trip
123	TRIP	TSN	420.12		Hours	☐	None	11/25/2021 1:56:37 PM	Sample OperationalEventReviewer	A trip
123	TRIP	MHC	-10	20000	Cycles	☑	None	11/25/2021 1:56:37 PM	Sample OperationalEventReviewer	A trip
		MHC	-8	20000	Cycles	☑	None	11/24/2021 6:32:17 PM		
		MHT	200.45	50000	Hours	☑	None	11/24/2021 6:32:17 PM		
		TSN	400		Hours	☐	None	11/24/2021 6:32:17 PM		
		MHC		20000	Cycles	☑	None	11/24/2021 6:31:15 PM		
		MHT		50000	Hours	☑	None	11/24/2021 6:31:15 PM		

Figure 79.

4.5 Physical Part Life Cycle and Promotion

Note: Only an **Asset Admin** or **Asset Editor** Identity member can promote **Physical Part** Items.

The **Physical Part** Life Cycle Map defines the following States for the **Physical Part** Items:

1. **Preliminary**
2. **Active**

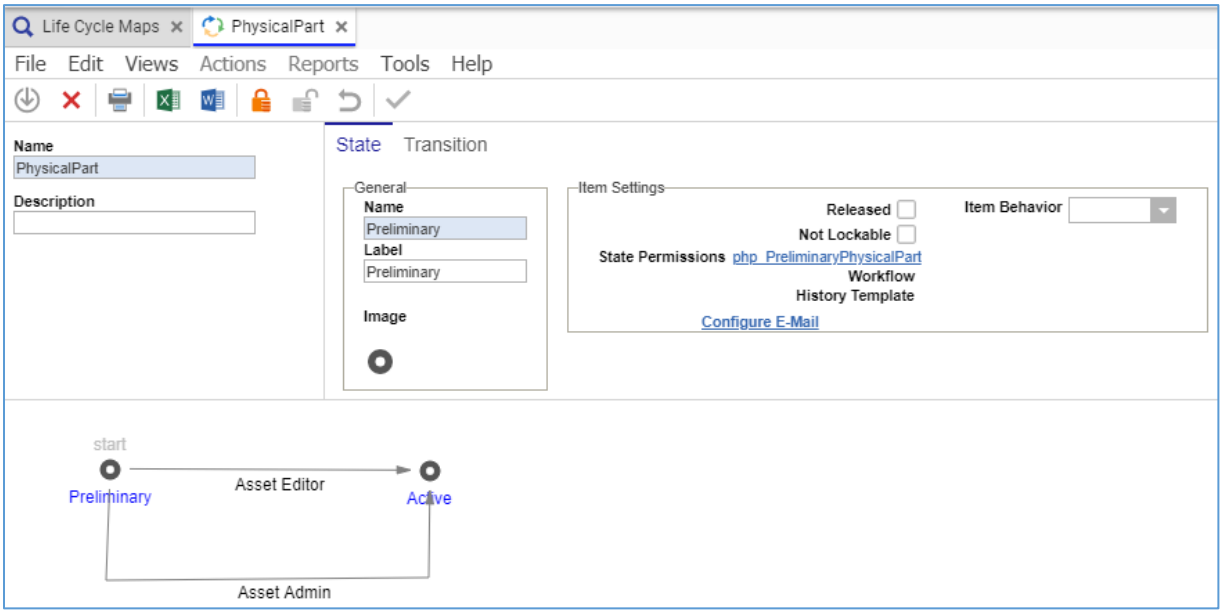


Figure 80.

Use the standard Item promotion procedure to promote the **Physical Part** Items.

Once created, a new **Physical Part** Item is in the **Preliminary** State, where it is a draft: a member of the **Asset Editor** or **Asset Admin** Identity has full editing rights for its properties and single-level BOM.

For example:

- An inventory control identifier (**Serial Number** or **Lot/Batch**) can be missing.
- This **Physical Part** can be deleted if it is not used as a Physical Part BOM component.

Physical Part Item promotion to the **Active** State is commissioning it into operation. A represented physical asset is ready for or in service. For example, an aircraft is airworthy and can operate flights. An engine is or can be installed on the aircraft and can run.

For the **Active** State promotion, a controlled **Physical Part** Item must have an appropriate inventory control identifier (serial or lot number). If the **Physical Part** Item has a Physical Part BOM, all its components must be in the **Active** State before it can be promoted.

An error is raised if promoting a controlled **Physical Part** Item without an appropriate inventory control identifier: a serialized **Physical Part** Item without a **Serial Number** value or a lot-controlled **Physical Part** Item without a **Lot / Batch** value.

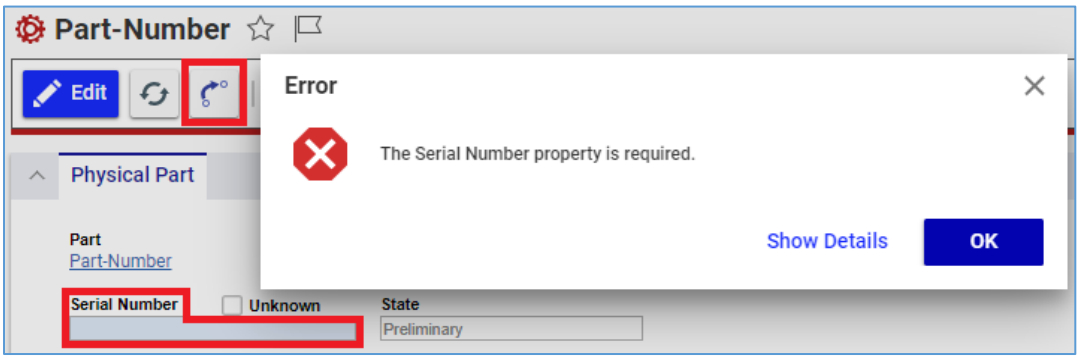


Figure 81.

Another error is raised if promoting a **Physical Part** Item that has a Physical Part BOM with at least one component **Physical Part** Item in the **Preliminary** State. All the component Items must be promoted first.

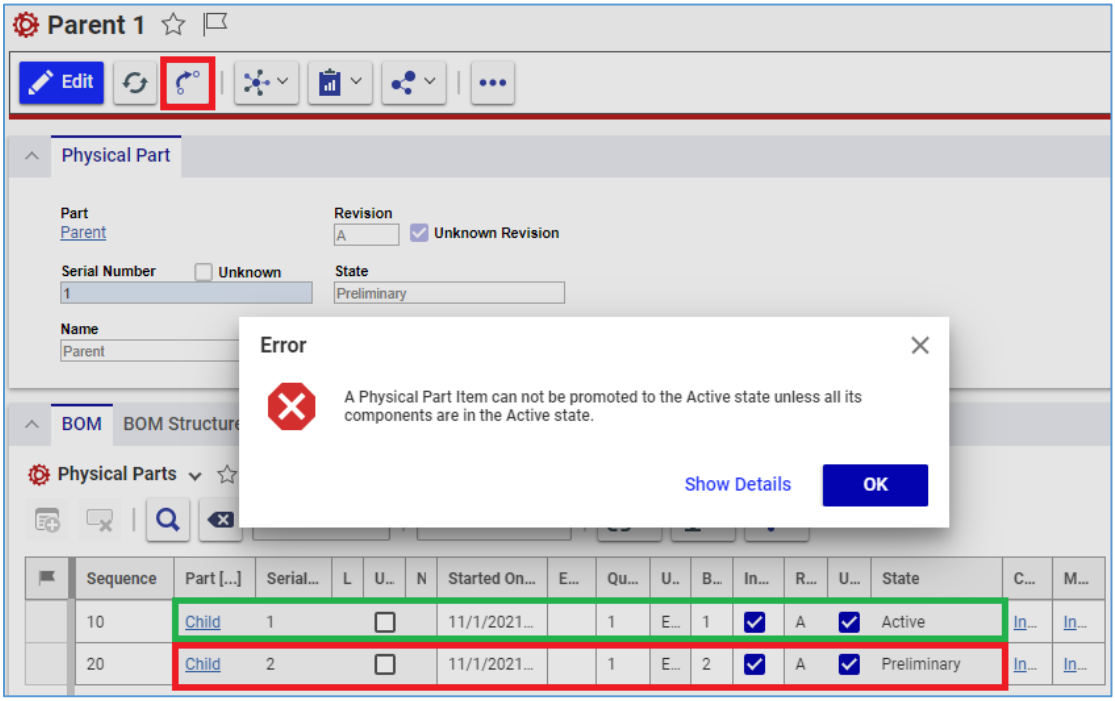


Figure 82.

Editing and deleting the **Physical Part** Item in the **Active** State is limited. A member of the **Asset User** or **Asset Editor** Identity can only use a Remove-and-Replace operation to update its single-level Physical Part BOM; see section 5.2.3 Removing-and-replacing Physical Parts in BOMs. They can also update the **Current Value** and the **Life Limit** cells in the **Current Life Values** tab grid and the **Date** cell in the **Current Date Values** tab grid; see sections 4.7.2.2 Manual update of life values and 4.7.3 Current values of Physical Part life dates.

A member of the **Asset Admin** Identity has full editing rights for the **Physical Part** Item properties and single-level BOM. **Asset Admin** also can delete it if it is not used as a **Physical Part** BOM component.

4.6 Physical Part history

A **Physical Part** Item can have history logs for its Physical Part BOM structure and life variables but not for its properties.

The Physical Part BOM structure of the **Physical Part** Item is trackable as histories of its:

- Immediate components (children) on the **BOM** tab. For more details, see section *5.2.1 Exploring Single-level Physical Part BOMs*.

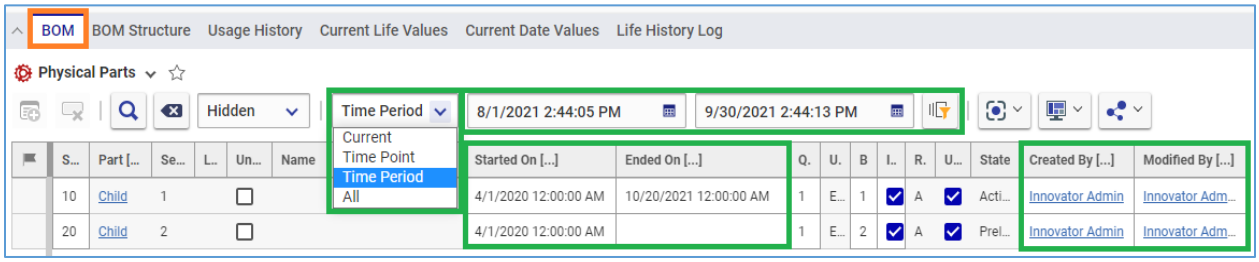


Figure 83.

- Multi-level BOM structure on the **BOM Structure** tab. For more details, see section *5.3 Multi-level Physical Part BOM structures*.

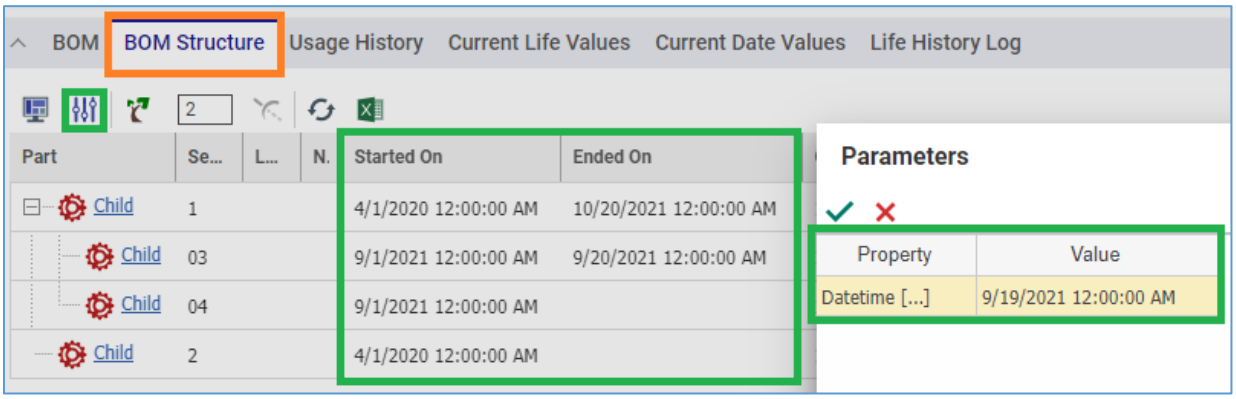


Figure 84.

- Usage as a BOM component (its parents) on the **Component History** tab. For more details, see section *5.4 Physical Part Usage History*.

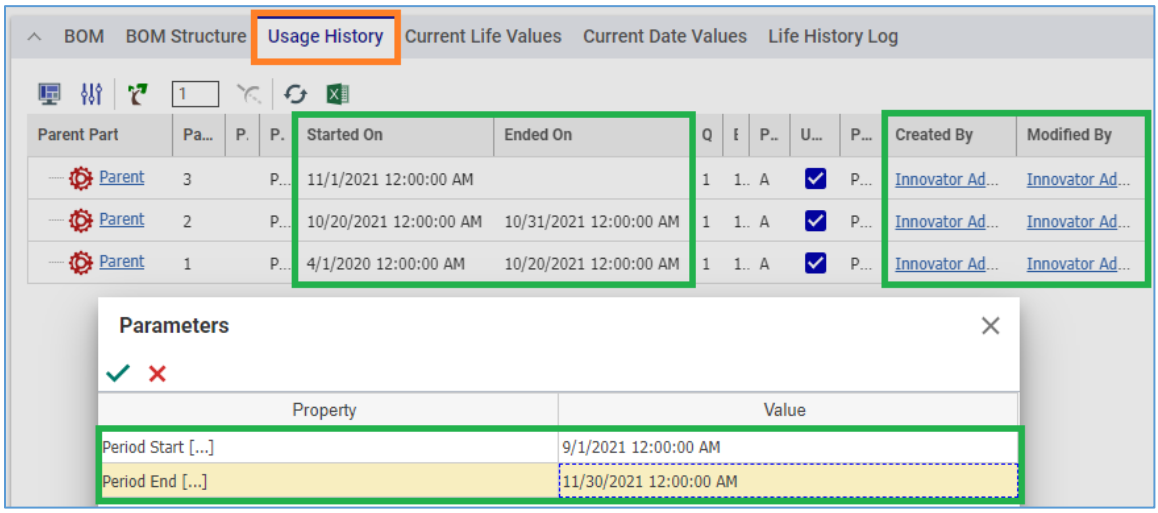


Figure 85.

The history of life variables of a **Physical Part** Item is tracked as records of their updates on the **Life History Log** tab. For more details, see section 4.7.4 History of numeric Physical Part life variables..

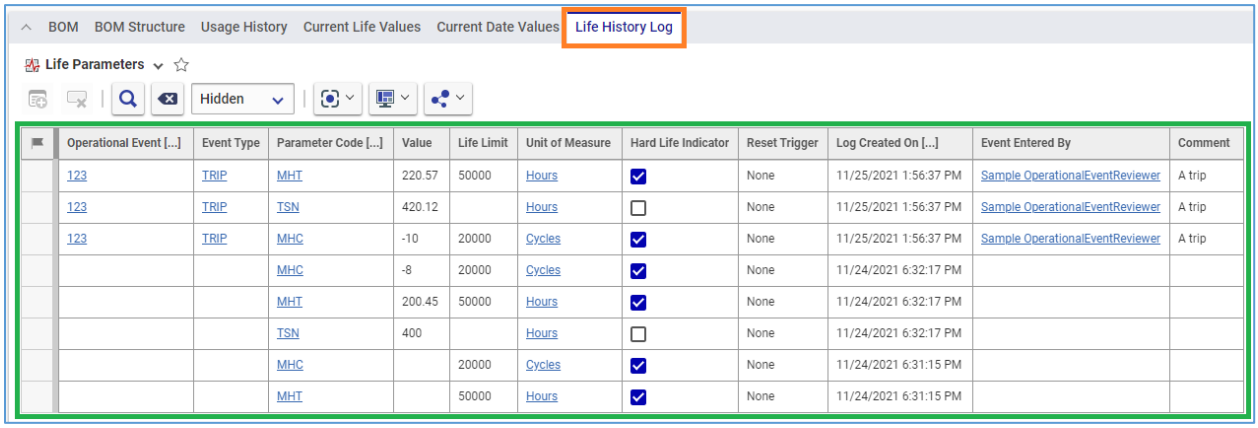


Figure 86.

The view-only foreign properties of the **Physical Part** Item are kept and can be traced in a related **Part** Item. The native properties of the **Physical Part** Item are not trackable.

The **Versions** command is disabled in the **Physical Part** Item toolbar.

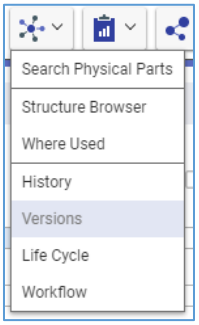


Figure 87.

Clicking the **History** command raises an error.

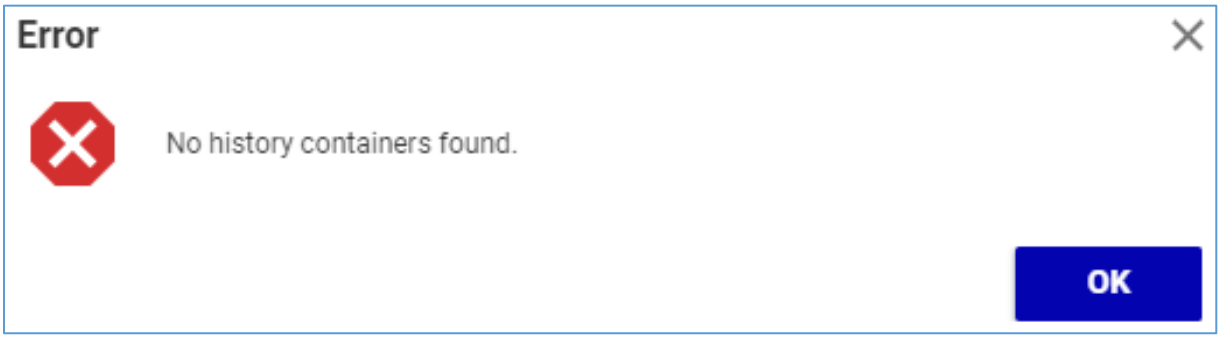


Figure 88.

The **Physical Part** Item always (in any State, after any change) has the same Major Revision **A** and Generation **1** since it is not versionable.

Figure 89.

The **Item Properties** dialog shows an Identity that created the **Physical Part** Item and when it was created. It also shows the last Identity that modified the **Physical Part** Item and when it was modified.

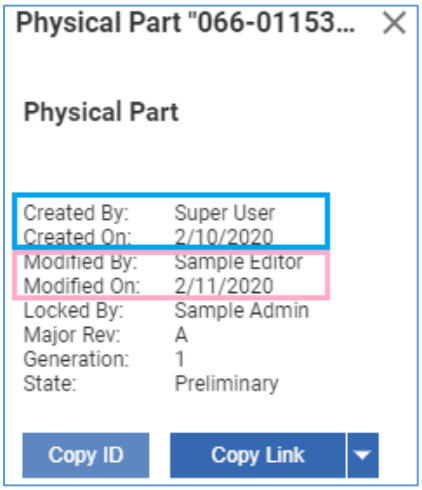


Figure 90.

To access the **Item Properties** dialog, use either:

- The **Physical Part** Item action menu: **More** --> **Properties**

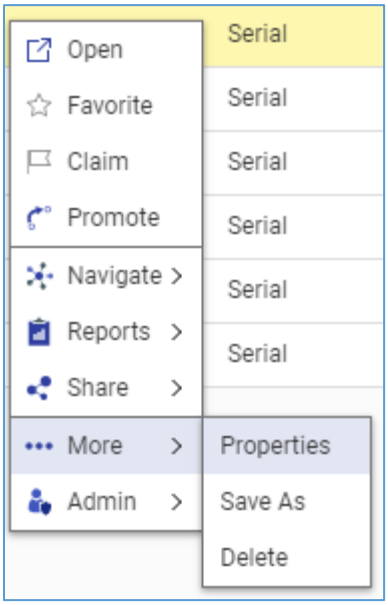


Figure 91.

- The **Physical Part** Item toolbar: **More --> Properties**.

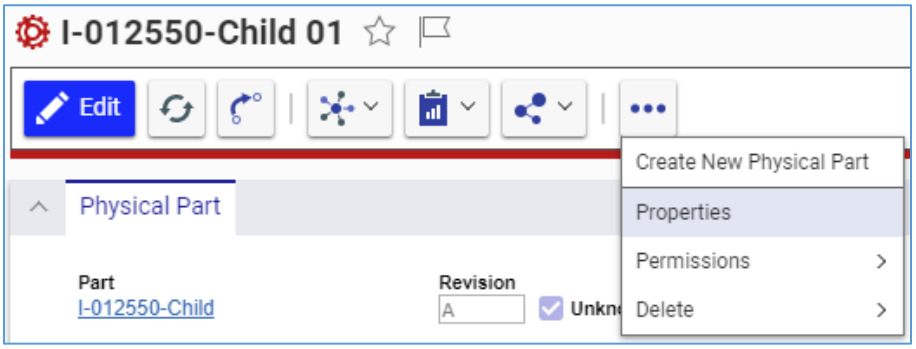


Figure 92.

4.7 Tracking Lives of Physical Parts

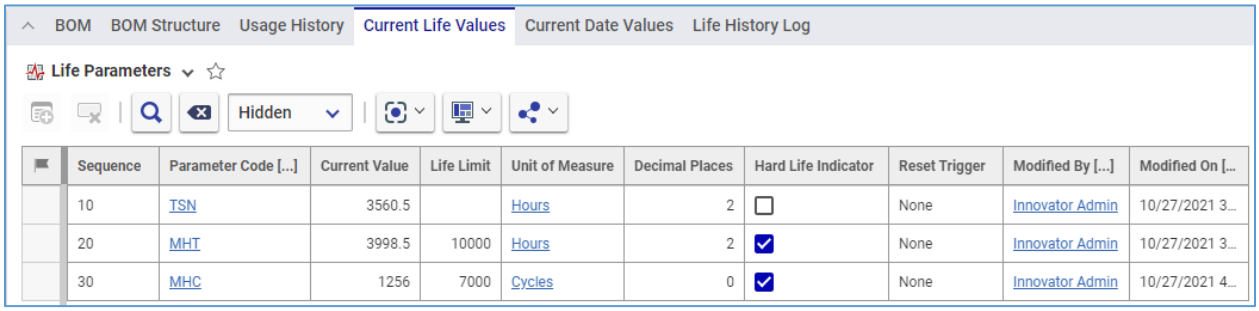
If a **Physical Part** Item represents an asset with one or more life variables or dates that should be tracked, this **Physical Part** Item should be populated with tracking **Life Parameter** Items as described in section 4.7.1 Managing Physical Parts with Life Parameters.

Once a **Physical Part** Item is populated with a **Life Parameter** Item, a **PhysicalPart LifeValue** or **PhysicalPart DateValue** Relationship Item is created to define a connection between the given **Physical Part** and **Life Parameter** Items and to keep properties of this connection. Such a Relationship Item represents a life variable of a given real-world asset.

Once a new value is set in the **Current Value** or **Life Limit** property of a **PhysicalPart LifeValue** Relationship Item, a **PhysicalPart LifeHistoryLog** Relationship Item is created to record this value in the history of life variable values. Such a Relationship Item represents a historical record of a life variable value of a given real-world asset at a specific point in time.

The **Physical Part** Item form has three Relationships tabs for the discussed Relationship Items:

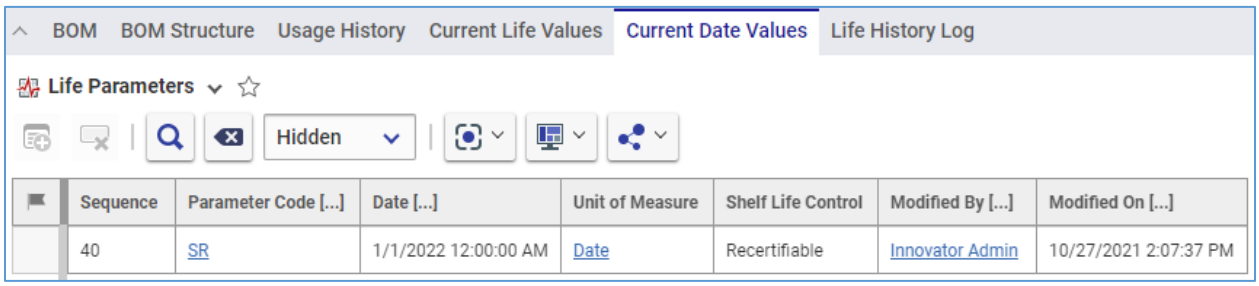
- **Current Life Values** for the **PhysicalPart LifeValue** Relationship Items that represent numeric life variables of an asset with their current values. For details, see section 4.7.2 Current values of numeric Physical Part life variables.



Sequence	Parameter Code [...]	Current Value	Life Limit	Unit of Measure	Decimal Places	Hard Life Indicator	Reset Trigger	Modified By [...]	Modified On [...
10	TSN	3560.5		Hours	2	☐	None	Innovator Admin	10/27/2021 3...
20	MHT	3998.5	10000	Hours	2	☑	None	Innovator Admin	10/27/2021 3...
30	MHC	1256	7000	Cycles	0	☑	None	Innovator Admin	10/27/2021 4...

Figure 93.

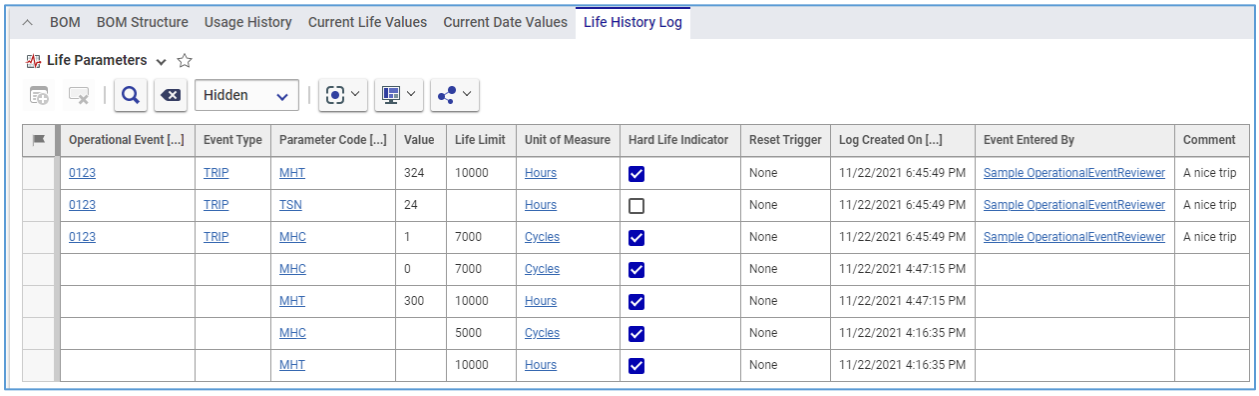
- **Current Date Values** for the **PhysicalPart DateValue** Relationship Items that represent life dates of an asset with their current values. For details, see section 4.7.3 Current values of Physical Part life dates.



Sequence	Parameter Code [...]	Date [...]	Unit of Measure	Shelf Life Control	Modified By [...]	Modified On [...]
40	SR	1/1/2022 12:00:00 AM	Date	Recertifiable	Innovator Admin	10/27/2021 2:07:37 PM

Figure 94.

- **Life History Log** for the **PhysicalPart LifeHistoryLog** Relationship Items that represent historical records of values of numeric life variables. For details, 4.7.4 History of numeric Physical Part life variables.



Operational Event [...]	Event Type	Parameter Code [...]	Value	Life Limit	Unit of Measure	Hard Life Indicator	Reset Trigger	Log Created On [...]	Event Entered By	Comment
0123	TRIP	MHT	324	10000	Hours	☑	None	11/22/2021 6:45:49 PM	Sample OperationalEventReviewer	A nice trip
0123	TRIP	TSN	24		Hours	☐	None	11/22/2021 6:45:49 PM	Sample OperationalEventReviewer	A nice trip
0123	TRIP	MHC	1	7000	Cycles	☑	None	11/22/2021 6:45:49 PM	Sample OperationalEventReviewer	A nice trip
		MHC	0	7000	Cycles	☑	None	11/22/2021 4:47:15 PM		
		MHT	300	10000	Hours	☑	None	11/22/2021 4:47:15 PM		
		MHC		5000	Cycles	☑	None	11/22/2021 4:16:35 PM		
		MHT		10000	Hours	☑	None	11/22/2021 4:16:35 PM		

Figure 95.

Each record in these tabs is either a **PhysicalPart LifeValue**, **PhysicalPart DateValue**, or **PhysicalPart LifeHistoryLog** Relationship Item, not a **Life Parameter** Item. Thus, record properties are properties of a given Relationship Item but not of a related **Life Parameter** Item. Some properties are sourced from the **Life Parameter** Item. The given **Life Parameter** Items are related Items in some of these Relationship Items.

If a **Physical Part** Item represents an asset with no life variables or life dates that should be tracked, its **Part Policy** Item should not include **Life Policy** Items: its **Current Life Values**, **Current Date Values**, or **Life History Log** Relationships Grid is empty.

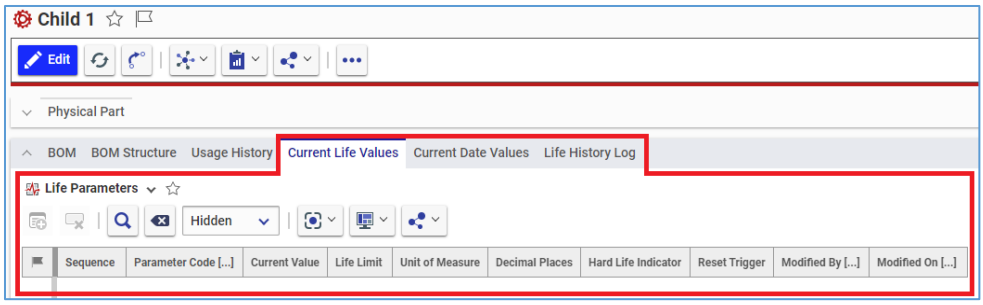


Figure 96.

4.7.1 Managing Physical Parts with Life Parameters

A **Physical Part** Item gets **Life Parameter** Items automatically from a **Life Policy** Item applied to a **Part Policy** Item that governs a **Part** Item from which this **Physical Part** Item derives. For details, see section 9.4 Part Policy Relationships. These **Life Parameter** Items are related Items in the **PhysicalPart LifeValue** and **PhysicalPart DateValue** Relationship Items. The **PhysicalPart LifeHistoryLog** Relationship Items record the history of the **PhysicalPart LifeValue** Relationship Items. All these Relationship Items are fully automatically managed as described later in this section. Their manual management is not allowed. However, when allowed, you can update the **PhysicalPart LifeValue** and **PhysicalPart DateValue** Relationship Items for the up-to-date life variable values.

During the creation of a new **Physical Part** Item, its **Current Life Values** and **Current Date Values** Relationship tabs are initially empty, even after a **Part** Item is specified.

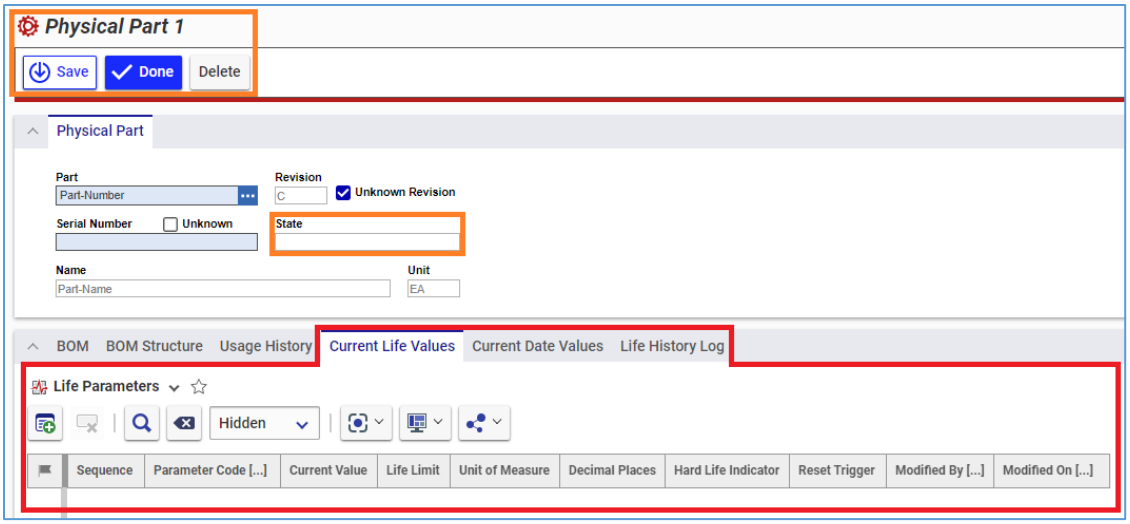


Figure 97.

Once you save the new **Physical Part** Item, the discussed tabs are automatically sourced with **Life Parameter** Items from a **Life Policy** Item given in a governing **Part Policy** Item. The **PhysicalPart LifeValue** and **PhysicalPart DateValue** Relationship Items are created and shown accordingly. The properties of these Relationship Items are set automatically with the following specifics:

- If a given **Life Parameter** Item is a numeric life variable, its **PhysicalPart LifeValue** Relationship Item has the following properties set as follows:

- **Current Value** gets **0** (zero), which is the default value for this property.

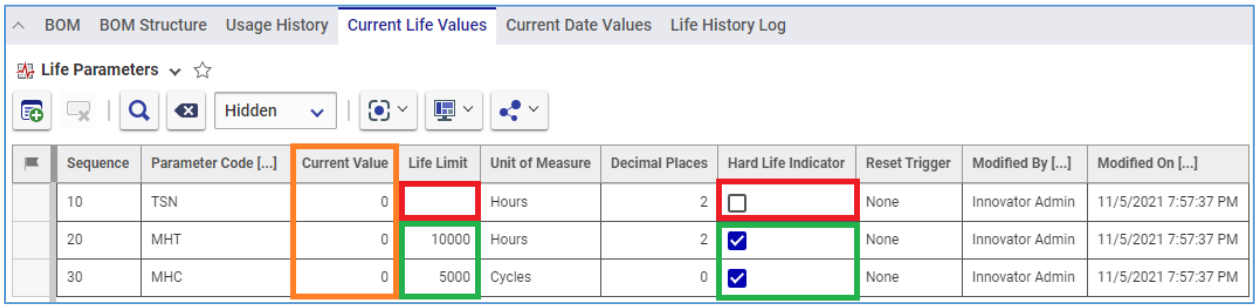


Figure 98.

- **Life Limit** gets a value set in the governing **Part Policy** Item if the given related **Life Parameter** Item has its **Hard Life Indicator** property set to **true**. This property remains empty if **Hard Life Indicator** is **false**.

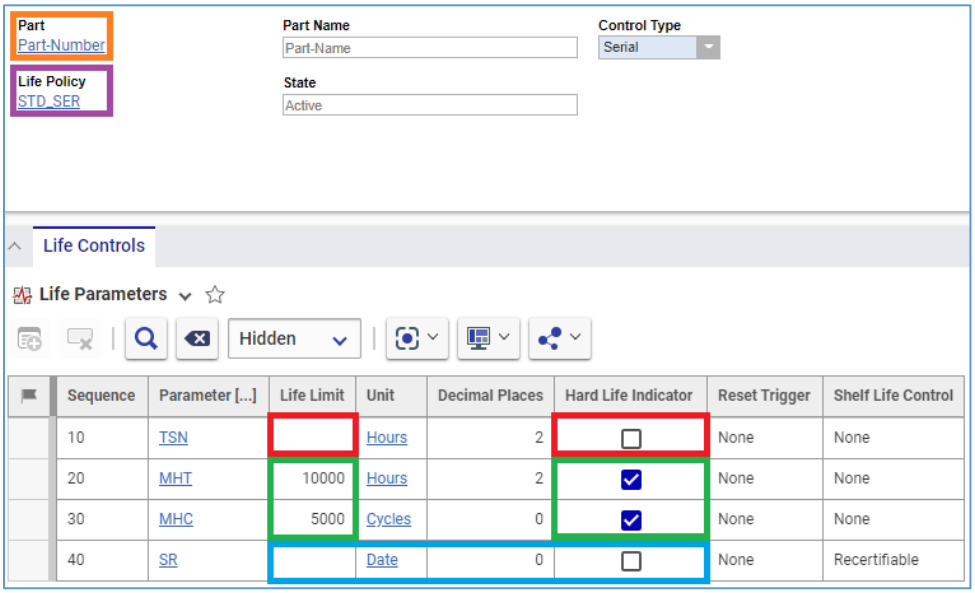


Figure 99.

- If a given **Life Parameter** Item is a life date, its **PhysicalPart DateValue** Relationship Item has the **Date** property set to your current date and time + 24 hours.

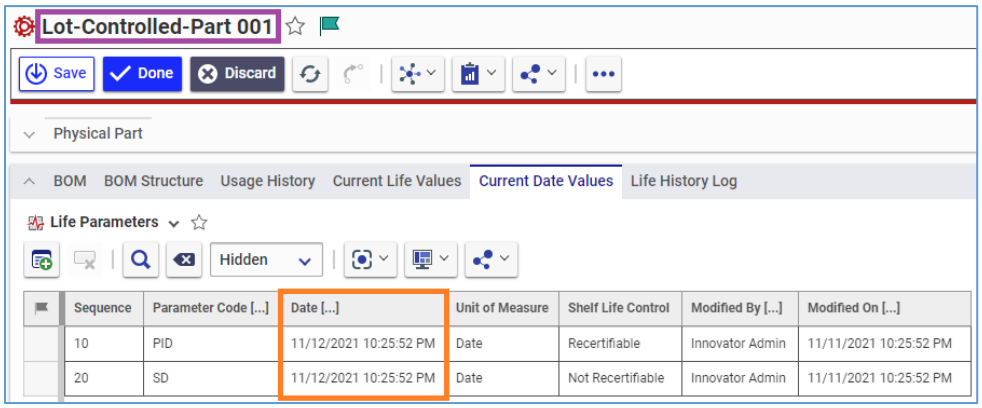


Figure 100.

If the applied **Life Policy** Item does not include **Life Parameter** Items of the numeric, date, or both kinds, the dedicated Relationships tabs will remain empty. For example, if a **Life Policy** Item has only **Life Parameter** Items that are date, the **Current Date Values** tab will be populated, but the **Current Life Values** and **Life History Log** tabs will be empty.

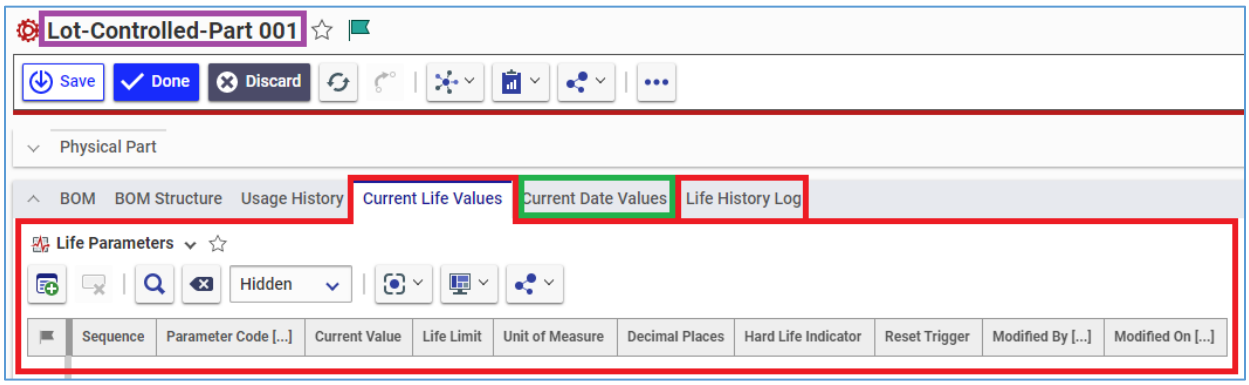


Figure 101.

The **Life History Log** Relationships tab is also initially empty during the creation of a new **Physical Part** Item.

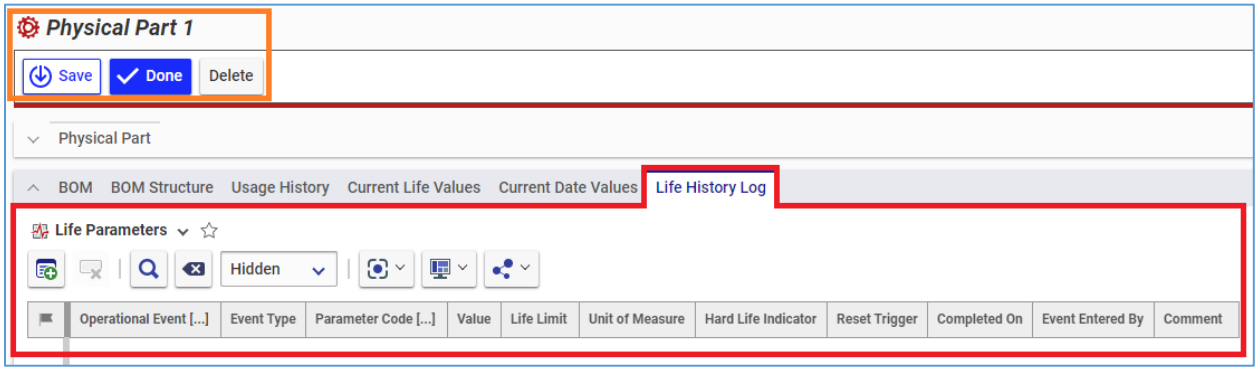


Figure 102.

However, this tab is not sourced with **Life Parameter** Items from the given **Life Policy** Item upon the **Physical Part** Item creation. The system creates a **PhysicalPart LifeHistoryLog** Relationship Item for each **Life Limit** value sourced from the governing **Part Policy** Item.

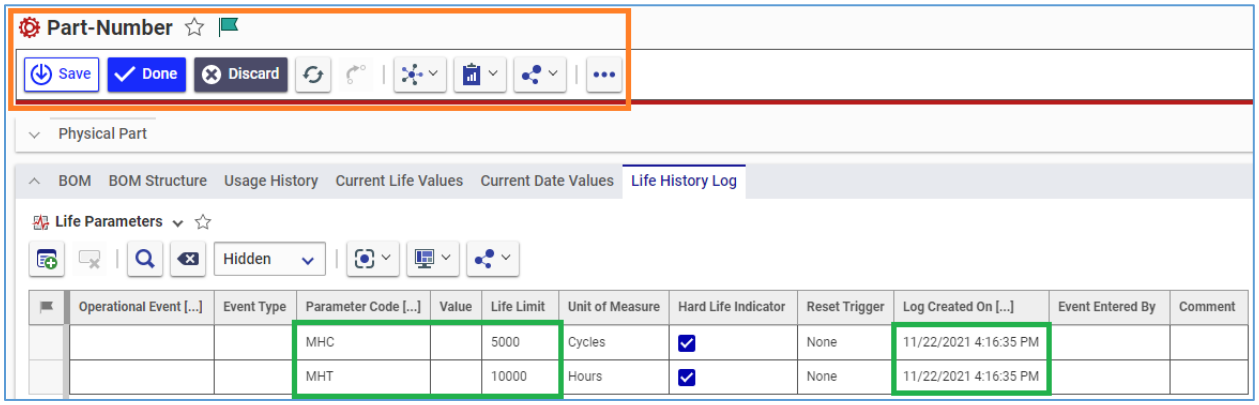
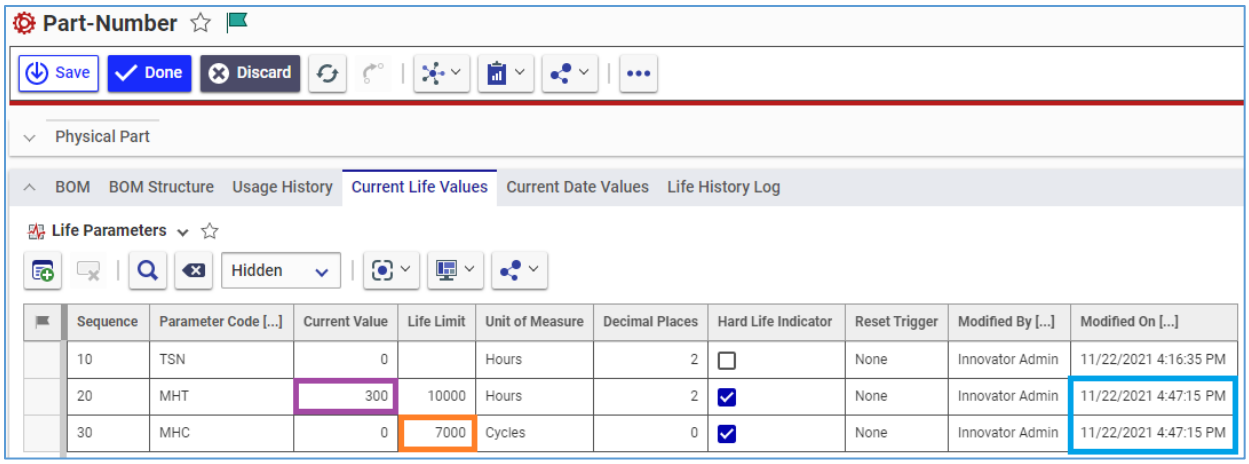


Figure 103.

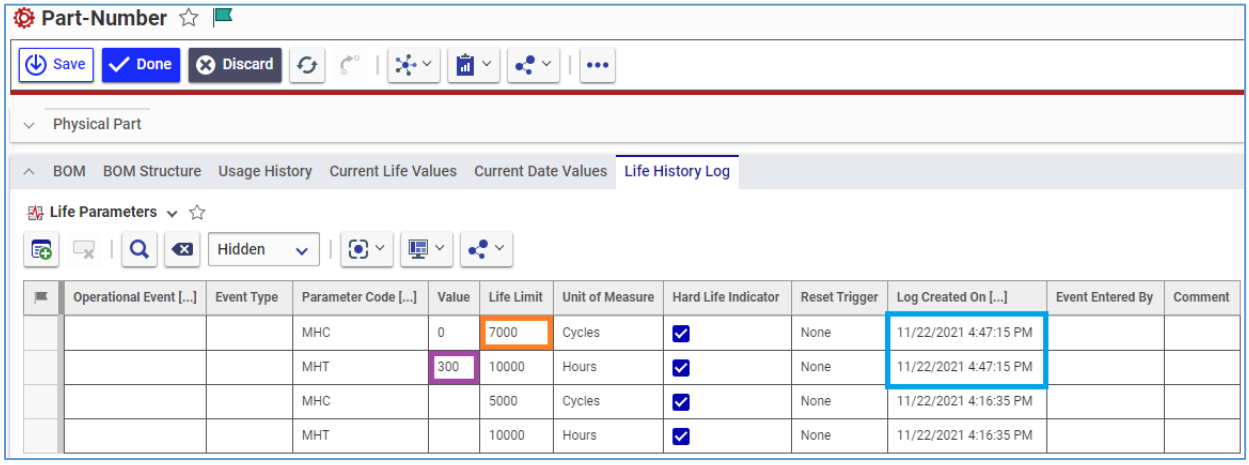
You can save a **Physical Part** Item with one or more **PhysicalPart LifeValue** Relationship Items whose **Current Value** or **Life Limit** properties had been updated manually. For details, see section 4.7.2.2 Manual update of Life Values.



Sequence	Parameter Code [...]	Current Value	Life Limit	Unit of Measure	Decimal Places	Hard Life Indicator	Reset Trigger	Modified By [...]	Modified On [...]
10	TSN	0		Hours	2	☐	None	Innovator Admin	11/22/2021 4:16:35 PM
20	MHT	300	10000	Hours	2	☑	None	Innovator Admin	11/22/2021 4:47:15 PM
30	MHC	0	7000	Cycles	0	☑	None	Innovator Admin	11/22/2021 4:47:15 PM

Figure 104.

In this case, the system automatically creates a **PhysicalPart LifeHistoryLog** Relationship Item for each valid value manually entered in these **Current Value** or **Life Limit** properties to keep all these values in the history. For details, see section 4.7.4 History of numeric Physical Part life variables.



Operational Event [...]	Event Type	Parameter Code [...]	Value	Life Limit	Unit of Measure	Hard Life Indicator	Reset Trigger	Log Created On [...]	Event Entered By	Comment
		MHC	0	7000	Cycles	☑	None	11/22/2021 4:47:15 PM		
		MHT	300	10000	Hours	☑	None	11/22/2021 4:47:15 PM		
		MHC		5000	Cycles	☑	None	11/22/2021 4:16:35 PM		
		MHT		10000	Hours	☑	None	11/22/2021 4:16:35 PM		

Figure 105.

The system also automatically creates a **PhysicalPart LifeHistoryLog** Relationship Item for each value entered in a **Current Value** property by an **Operational Event** promotion. For details, see section 4.7.2.3 Operational Event update of Life Values.

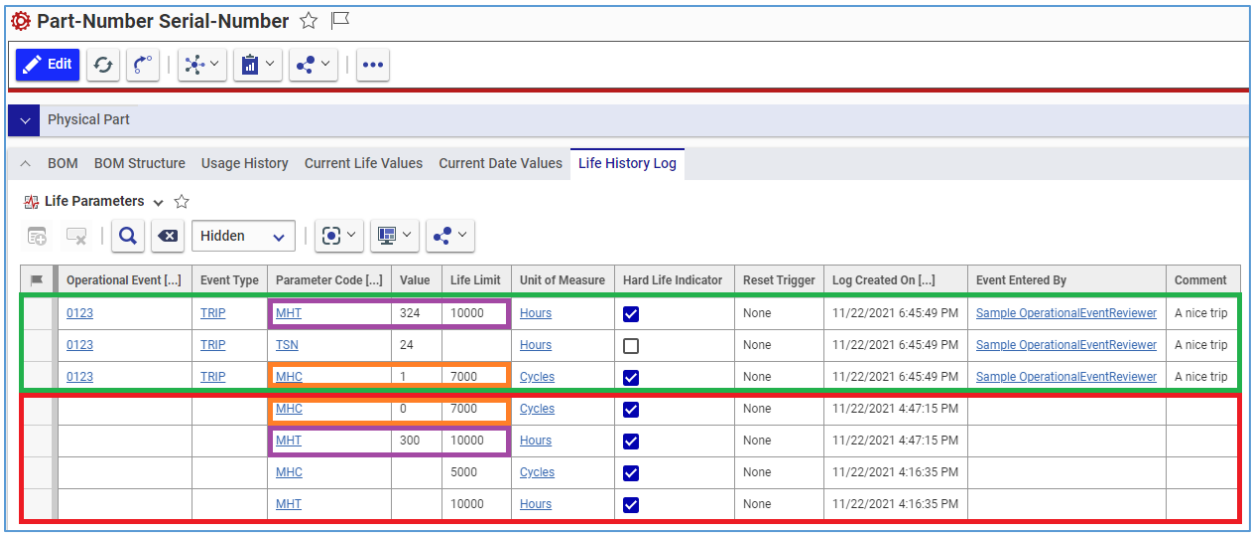


Figure 106.

The system does not create a **PhysicalPart LifeHistoryLog** Relationship Item for a value invalid for the **Current Value** or **Life Limit** property; for the property validness details, see section 4.7.2.2 Manual update of Life Values. Updated Items cannot be saved, and an error message is raised then; for the error message details, see section 4.8.4 Validation of Physical Parts.

When editing a **Physical Part** Item, its **Current Life Values**, **Current Date Values**, and **Life History** Relationships Grids have their toolbar and action menus enabled. But still, the manual management of these Grids is forbidden—an error is raised upon saving a **Physical Part** Item after adding, editing, or removing a **PhysicalPart LifeValue**, **PhysicalPart DateValue**, or **PhysicalPart LifeHistoryLog** Relationship Item with any toolbar or action menu button. For details, see section 3.7 Automatically managed DTC Relationships Grids.

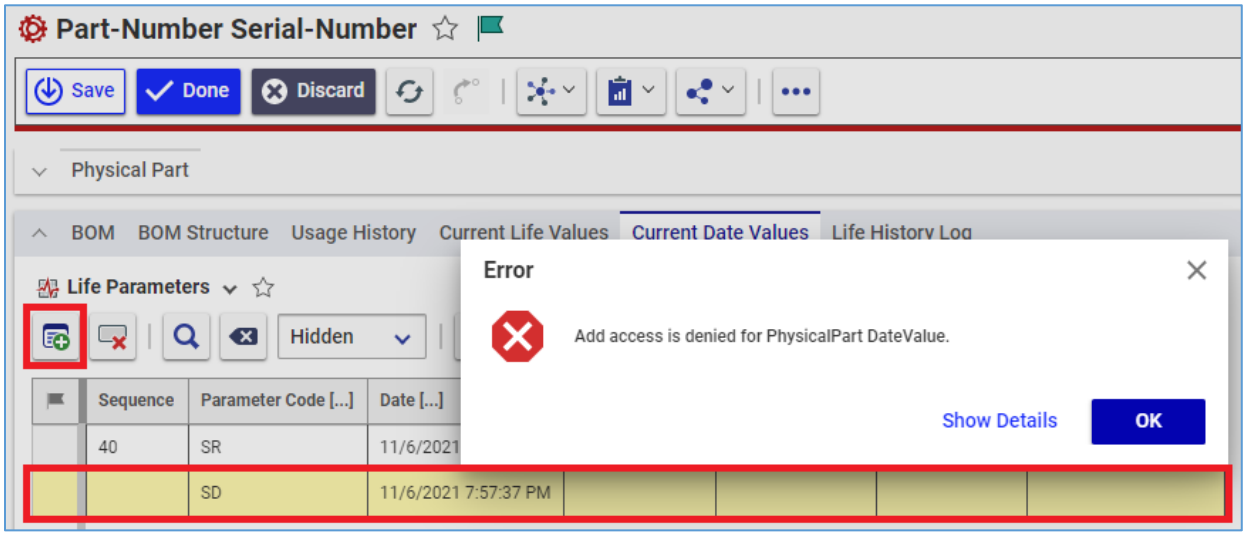


Figure 107.

4.7.2 Current values of numeric Physical Part life variables

If a **Physical Part** Item represents an asset with one or more numeric life variables whose values should be tracked, this **Physical Part** Item should have its **Current Life Values** Relationships Grid populated with the **PhysicalPart LifeValue** Relationship Items to keep the current values of these numeric life variables and their life limits. The Grid is automatically managed as described in section 4.7.1 Managing Physical Parts with Life Parameters.

There are two ways to update the current values of the numeric life variables:

- Manually—a member of an allowed Identity enters them in the Grid; see section 4.7.2.2 Manual update of Life Values.
- Automatically—the system sources them from an **Operational Event** Item performed by a given **Physical Part** Item or its Physical Part BOM structure; see section 4.7.2.3 Operational Event update of Life Values.

Whenever a numeric life variable or its life limit gets a new value manually or automatically, the system automatically records such an update in the **Life History Log** Relationships Grid that represents the history of all numeric life variable and life limit values.

Only serial-controlled **Physical Part** Items can have related numeric **Life Parameter** Items because we can track individually only serial-controlled real-word assets. The lot-controlled and non-controlled **Physical Part** Items cannot have numeric related **Life Parameter** Items because we cannot track lot-controlled and non-controlled real-word assets individually. The lot-controlled assets are tracked by batches. The non-controlled assets are not trackable by their definition. One lot-controlled or non-controlled **Physical Part** Item can be present in one or more Physical Part BOM structures in quantity not equal to one as multiple items from one batch can be present in different asset assemblies or at various places in the same assembly.

4.7.2.1 Current Life Values Relationships Grid

The **Current Life Values** Relationships tab of a **Physical Part** Item hosts a Grid with the **PhysicalPart LifeValue** Relationship Items describing Relationships between the given **Physical Part** Item and tracking **Life Parameter** Items. Such Relationship Items express numeric life variables of an asset along with their current values and life limits. Each row in the Grid is a separate Relationship Item. The row properties are the Relationship Item properties.

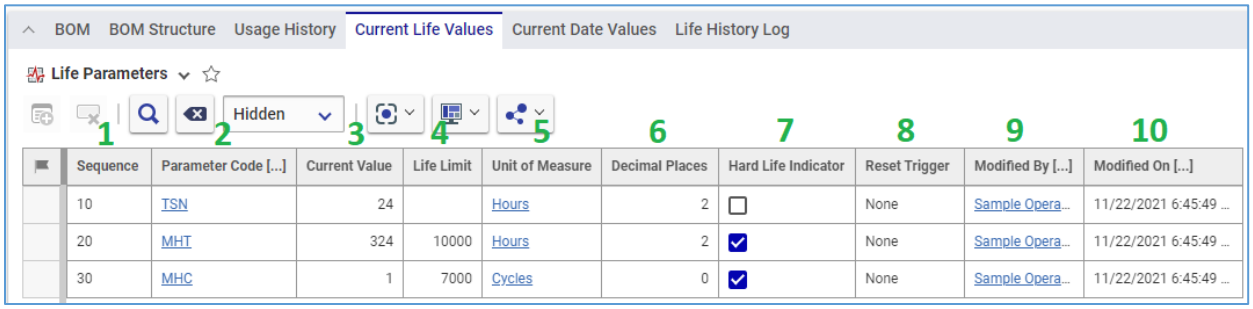


Figure 108.

The **Current Life Values** Relationship Grid shows properties of a **PhysicalPart LifeValue** Relationship Item as follows:

1. **Sequence**: a view-only sort order number of a given related **Life Parameter** Item in the Grid. It is the same as this **Life Parameter** Item has in the sourcing **Life Controls** Relationships Grid of an applied **Life Policy** Item.
2. **Parameter Code**: a view-only **Parameter Code** property value of the given **Life Parameter** Item. It is also a link to this **Life Parameter** Item.

3. **Current Value**: a current numeric value of a corresponding asset life variable. Its default value is **0** (zero). This property is editable and required: it cannot be empty in any State of the source **Physical Part** Item.

   If trying to save the **Physical Part** Item with a blank **Current Value** cell, an error pop-up dialog appears, and the default **0** value is set in these cells. For the error message, see section 4.8.4 Validation of Physical Parts.

   You should populate a **Current Value** cell only with real, true, up-to-date data.

   A valid value for this property is zero or a positive or negative number with the number of digits after the decimal point equal to or less than the **Decimal Places** property (6) defines.

4. **Life Limit**: a numeric value of the maximum failure (hard) limit for a corresponding life-limiting variable of the asset. This property should be set only for a **Life Parameter** Item that limits the **Physical Part** Item lives: its **Hard Life Indicator** (7) is **true**.

   This property is editable and not required: it can be empty in any State of the source **Physical Part** Item.

   It is automatically sourced from a governing **Part Policy** Item when the **Current Life Values** Relationship Grid is populated. For details, see section 4.7.1 Managing Physical Parts with Life Parameters.

   A valid value for this property is zero or a positive or negative number with the number of digits after the decimal point equal to or less than the **Decimal Places** property (6) defines.

5. **Unit of Measure**: a view-only **Life Unit** Item given in the **Unit of Measure** property of the **Life Parameter** Item. It is also a link to this **Life Unit** Item.

6. **Decimal Places**: a view-only **Decimal Places** property value of the given **Life Unit** Item. This property defines the precision for the **Current Value** (3) and **Life Limit** (4) property values of the given **Physical Part Life Value** Relationship Item.

7. **Hard Life Indicator**: a view-only **Hard Life Indicator** property value of the **Life Parameter** Item that allows if set to **true** the given the **Life Limit** property (4) to have a valid value.

8. **Reset Trigger**: a view-only **Reset Trigger** property value of the **Life Parameter** Item.

9. **Modified By**: the last Identity that modified the **PhysicalPart LifeValue** Relationship Item. This property is view-only.

10. **Modified On**: a view-only timestamp of the last modification of the **PhysicalPart LifeValue** Relationship Item.

In the Grid, the Items are sorted by their **Sequence** values in the ascending order (from 0 to 9) by default.

4.7.2.2 Manual update of life values

Note: Only an **Asset Admin**, **Asset Editor**, or **Asset User** Identity member can update editable properties of the **PhysicalPart LifeValue** Relationship Items.

If allowed, you can manually update current life variable values in the **Current Value** cells and life limits in the **Life Limit** cells in the **Life Parameters** Relationships Grid following the standard **Physical Part** Item modifying procedure.

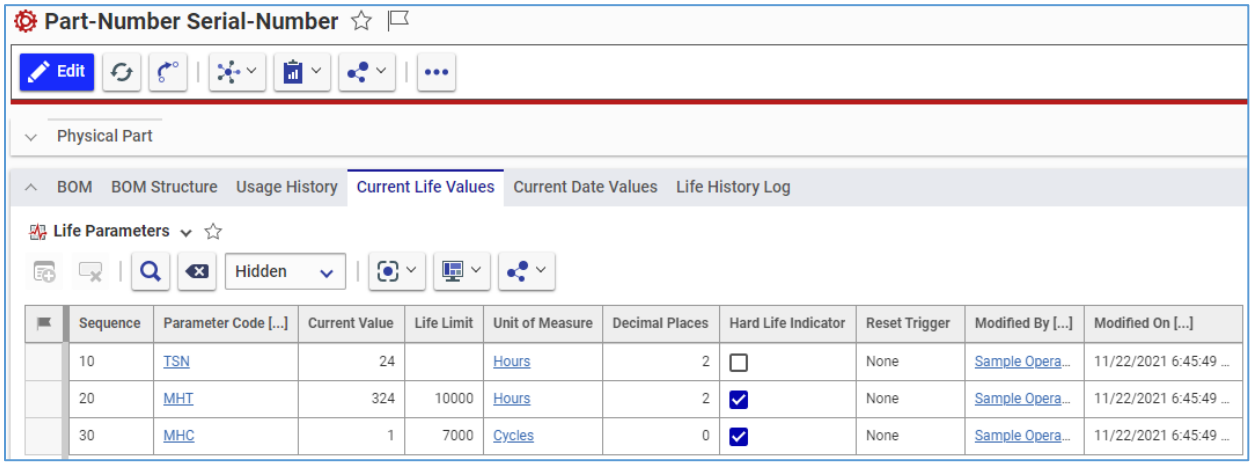


Figure 109.

You can use the manual update procedure to correct **Current Value** property values wrongly set by an applied **Operational Event** Item; see section 4.7.2.3 Operational Event update of life values. Another option is to create and promote a new **Operational Event** Item with values that will produce correct **Current Value** property values; see section see section 11.5 Operational Event Life Cycle and Promotion.

To provide up-to-date life variable values accumulated by a represented real-world asset during its activities, enter such values in the appropriate **Current Value** cells in the **Life Parameters** Relationships Grid of a representing **Physical Part** Item.

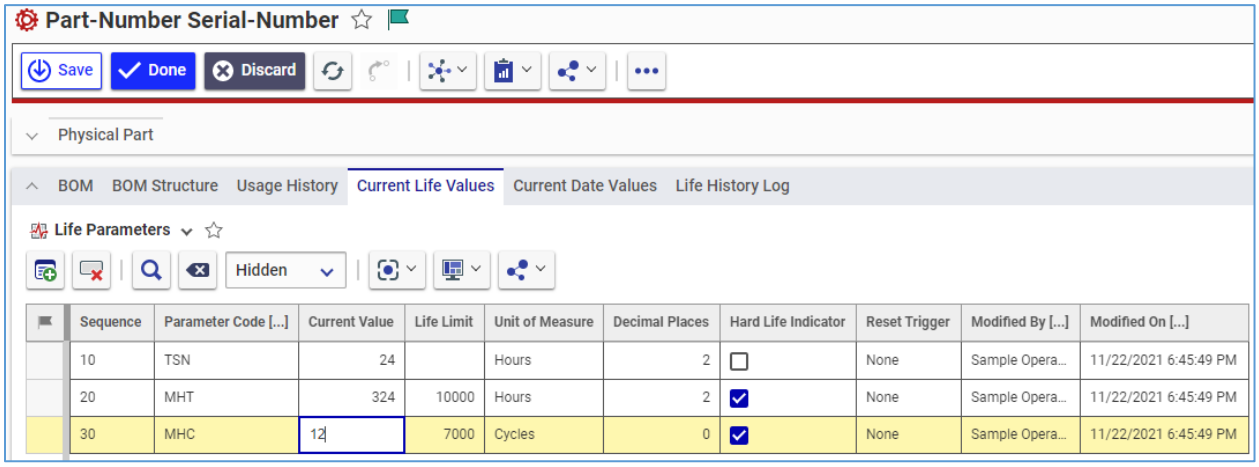


Figure 110.

When you save a **Physical Part** Item with one or more new values set in the **Current Value** properties of its **Physical Part Life Value** Relationship Items, the system automatically records them in its **Life History Log** Relationships Grid as well. This Grid shows all values for a given related **Life Parameter** Item, including the previous one. For details, see section 4.7.4 History of numeric Physical Part life variables.

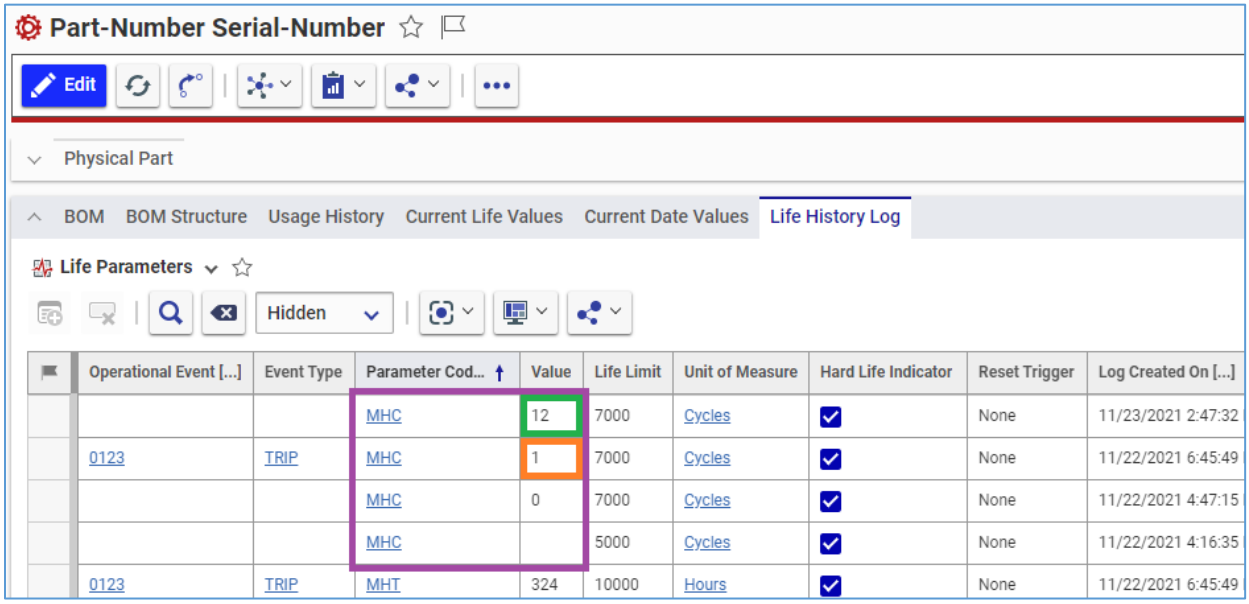


Figure 111.

To change life variable limits for a represented real-world asset, enter such values in the appropriate **Life Limit** cells in the **Life Parameters** Relationships Grid of a representing **Physical Part** Item.

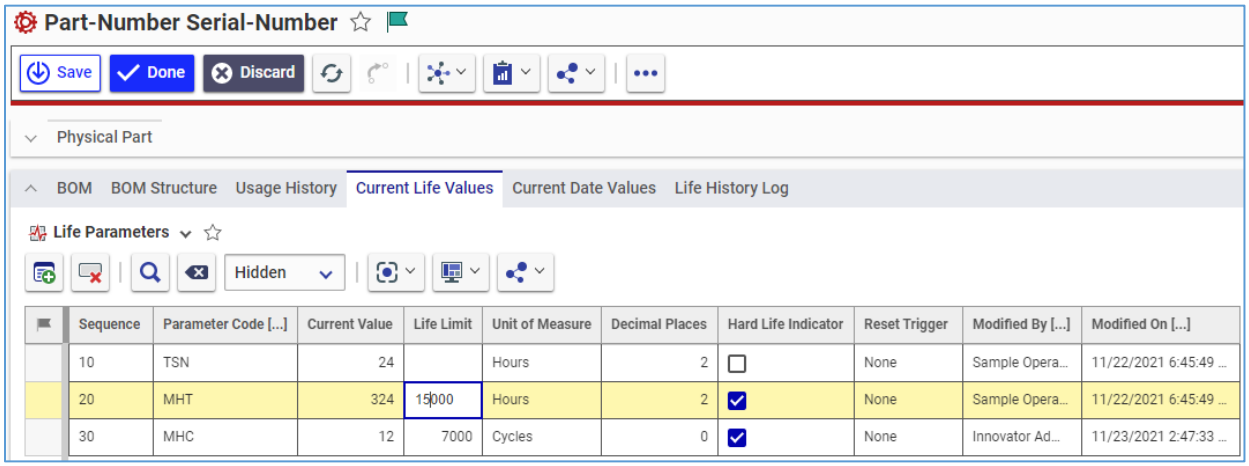


Figure 112.

When you save a **Physical Part** Item with one or more new values set in the **Life Limit** properties of its **Physical Part Life Value** Relationship Items, the system automatically records them in its **Life History Log** Relationships Grid as well. This Grid shows the current and all previously entered life limits for a given related **Life Parameter** Item. For details, see section 4.7.4 History of numeric Physical Part life variables.

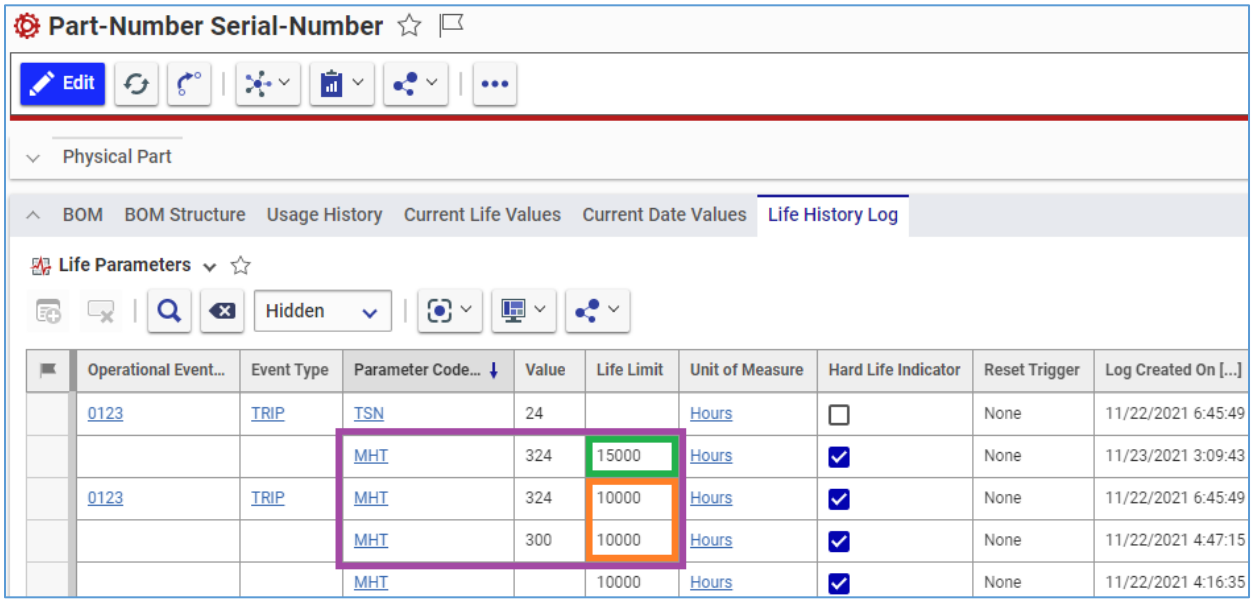


Figure 113.

If it is necessary, you can enter new values in the **Current Value** and **Life Limit** cells of one **Physical Part Life Value** Relationship Item simultaneously.

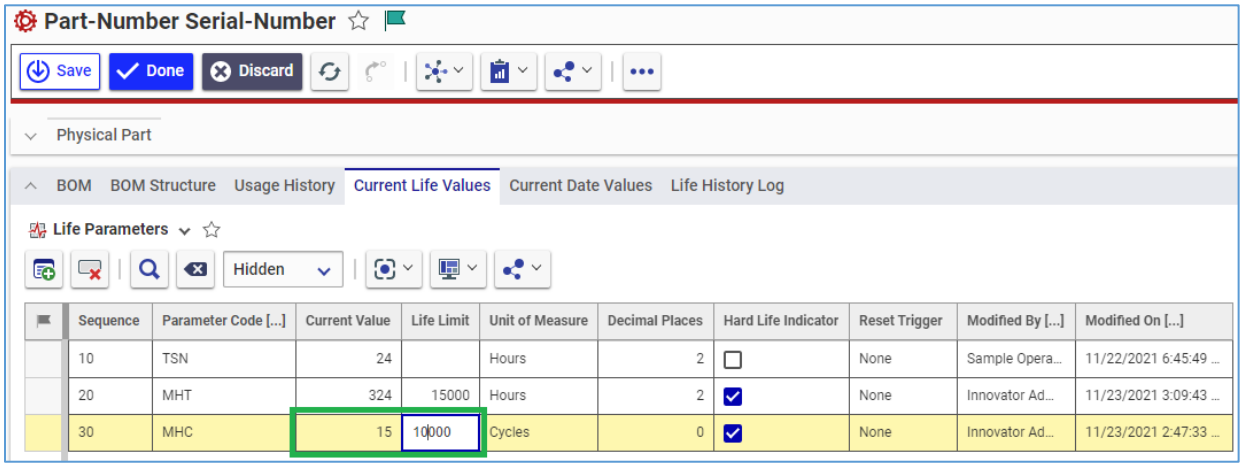


Figure 114.

When you save a **Physical Part** Item with this compound change, the system automatically records both new values in the **Life History Log** Relationships Grid as one entry. For details, see section 4.7.4 History of numeric Physical Part life variables.

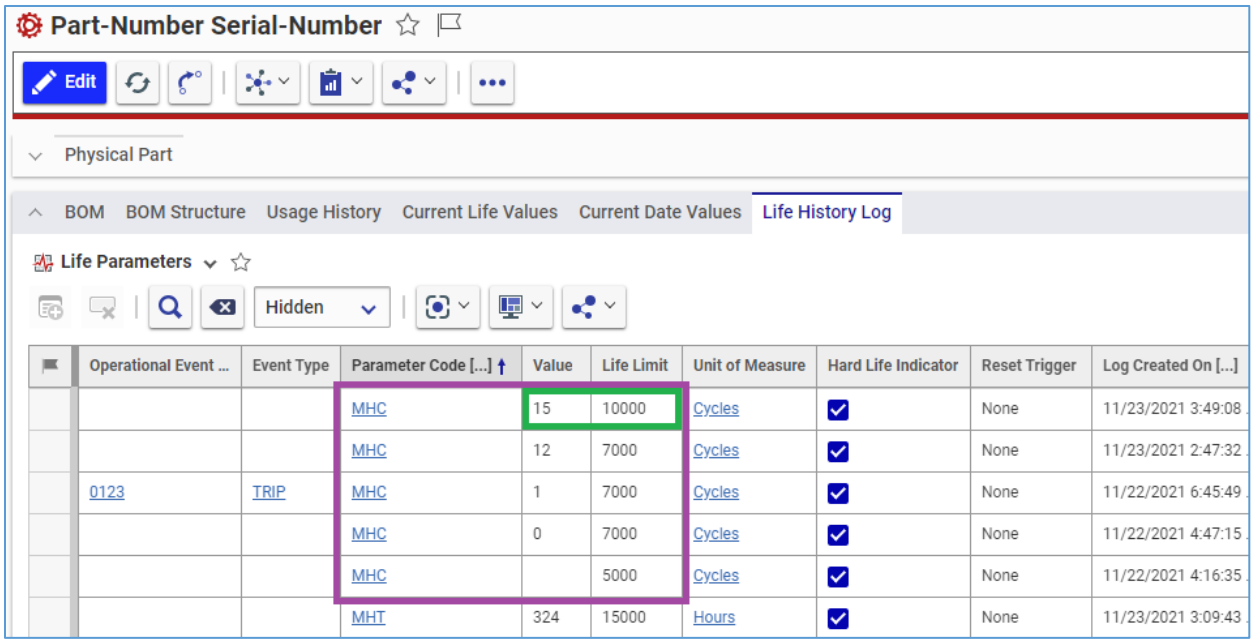


Figure 115.

Entered values should be valid for the **Current Value** and **Life Limit** cells; for details, see section 4.7.2.1 Current Life Values Relationships Grid. Otherwise, the changes cannot be saved, and an error message pops up.

The **Life Limit** cells should have values only for the **Life Parameter** Items with the **Hard Life Indicator** set to **true**. Upon saving a **Physical Part** Item, an error is raised, and this **Physical Part** Item cannot be saved if there is a **Life Limit** cell populated for a **Life Parameter** Item with the **Hard Life Indicator** set to **false**.

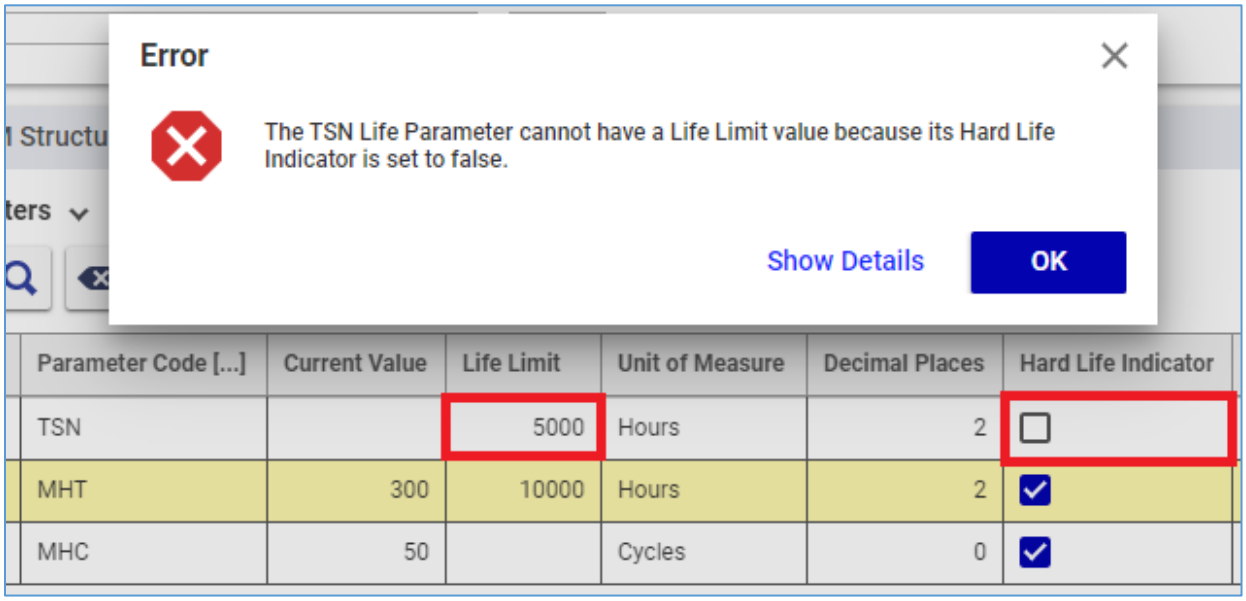


Figure 116.

The **Decimal Places** property defines the maximum number of digits after a decimal point for the **Current Value** and **Life Limit** property values of their common **Physical Part Life Value** Relationship Item. For example, if the **Decimal Places** value is 2, the corresponding **Current Value** and **Life Limit** properties can have up to 2 digits in their fractional part. Upon saving a **Physical Part** Item, an error is raised, and this **Physical Part** Item cannot be saved if the **Current Value** or **Life Limit** value has more decimal digits than their **Decimal Places** property allows.

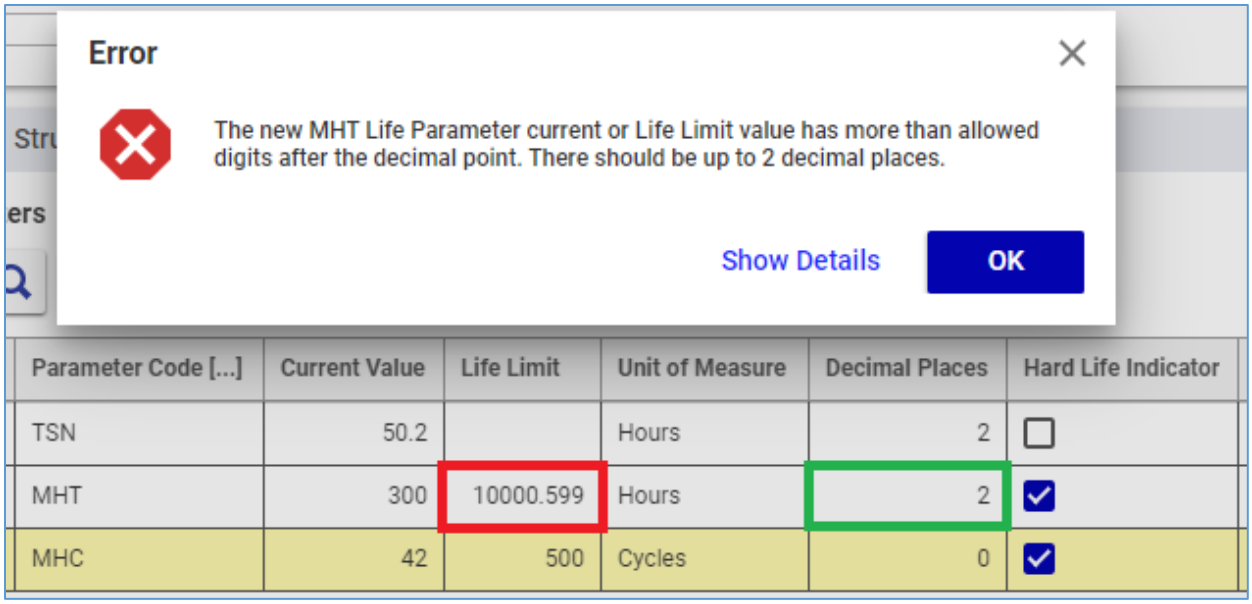


Figure 117.

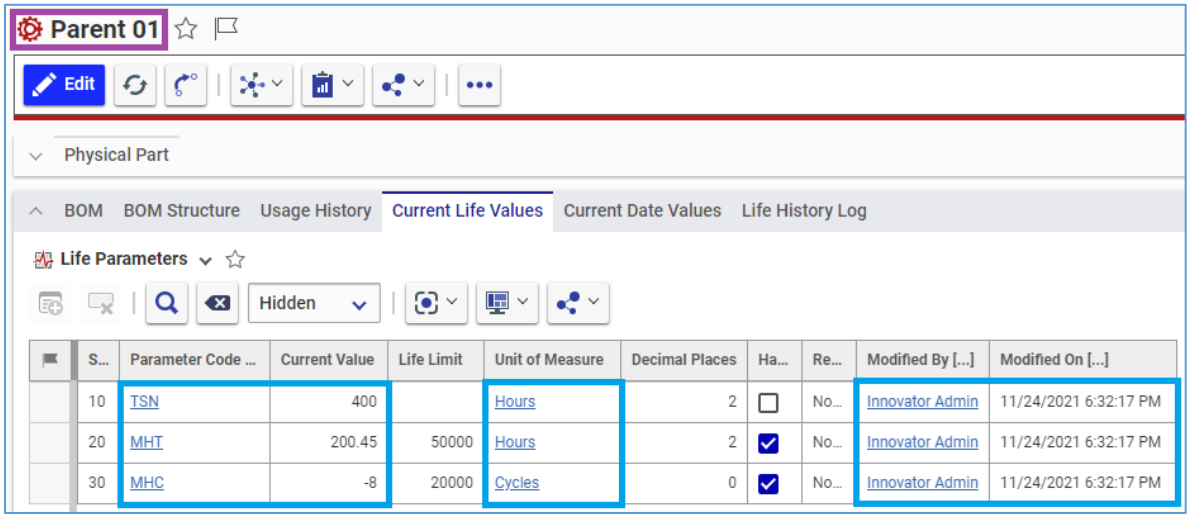
4.7.2.3 Operational Event update of life values

A real-world asset can be up and running—regularly performing some operational activities in the field, for example, an aircraft that makes flights.

Each operational activity updates asset life characteristics because it consumes a piece of asset life, like one flight costs a plane one engine start, one landing, some flying hours, and so on.

Since 12.0R3, the DTC application can automatically update the current life values of an asset by operational activities that this asset has performed.

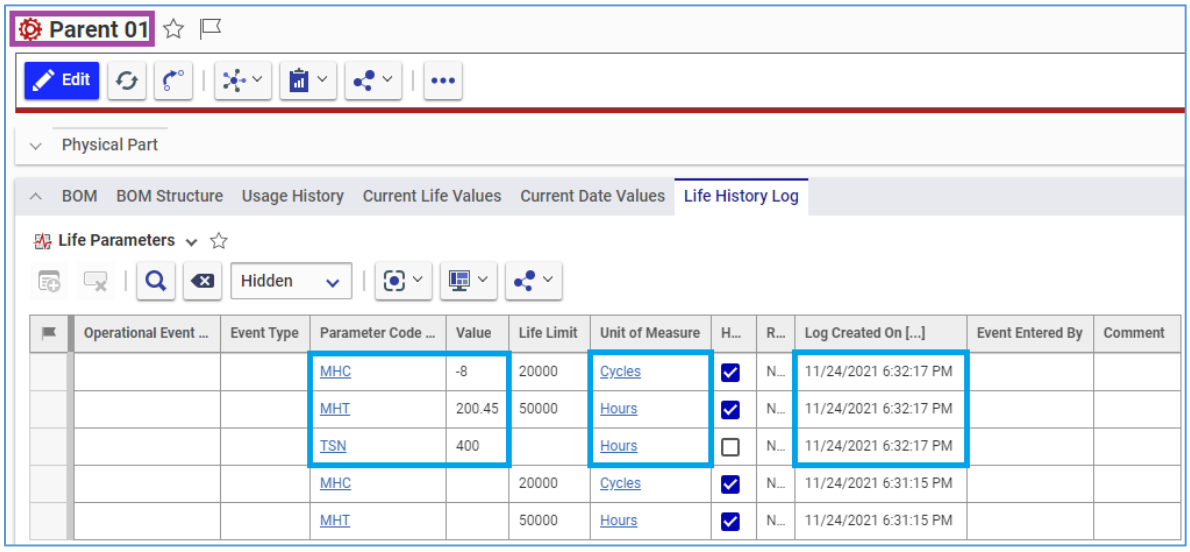
If a **Physical Part** Item is a real-world operational asset, like an aircraft, this **Physical Part** Item should have related **Life Parameter** Items in the **Current Life Values** Relationships tab that represents the asset life variables, like the total number of working hours since the asset's birth, a hard limit for working hours accumulated during the asset's life, and so on.



S...	Parameter Code ...	Current Value	Life Limit	Unit of Measure	Decimal Places	Ha...	Re...	Modified By [...]	Modified On [...]
10	TSN	400		Hours	2	☐	No...	Innovator Admin	11/24/2021 6:32:17 PM
20	MHT	200.45	50000	Hours	2	☑	No...	Innovator Admin	11/24/2021 6:32:17 PM
30	MHC	-8	20000	Cycles	0	☑	No...	Innovator Admin	11/24/2021 6:32:17 PM

Figure 118.

The asset may have already or have not yet accumulated some values for its life variables. The **Life History Log** Relationships tab of the **Physical Part** Item shows the history of the asset life characteristics.



Operational Event ...	Event Type	Parameter Code ...	Value	Life Limit	Unit of Measure	H...	R...	Log Created On [...]	Event Entered By	Comment
		MHC	-8	20000	Cycles	☑	N...	11/24/2021 6:32:17 PM		
		MHT	200.45	50000	Hours	☑	N...	11/24/2021 6:32:17 PM		
		TSN	400		Hours	☐	N...	11/24/2021 6:32:17 PM		
		MHC		20000	Cycles	☑	N...	11/24/2021 6:31:15 PM		
		MHT		50000	Hours	☑	N...	11/24/2021 6:31:15 PM		

Figure 119.

An **Operational Event** Item is a representation of a single real-world operational activity completed by an asset, like a flight made by an aircraft. For more details on the **Operational Event** Items, refer to section 11.1 Understanding Operational Events.

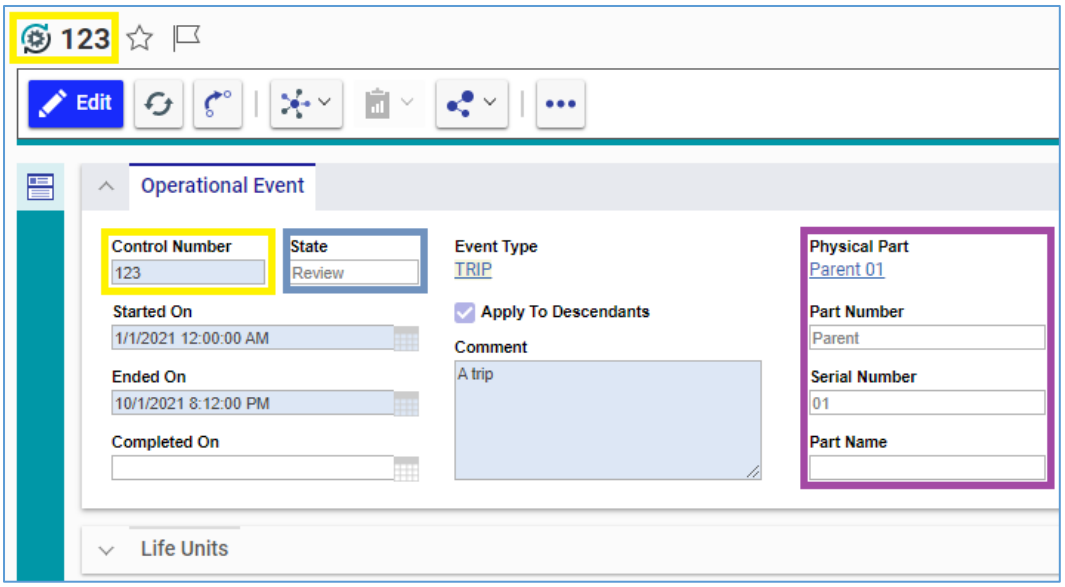


Figure 120.

The **Life Units** Relationships tab of an **Operational Event** Item holds increments of life variables that an asset has gained with the completion of some operational activity, for example, the number of engine starts, landings, cycles, flying hours, etc. that an aircraft did during a flight. These life variable increments are numeric values entered in the **Value** properties of the given **OperationalEvent LifeUnit** Relationship Items. The **Operational Event** Item records and keeps the increments by units of measure (the **Life Unit** Items) but not by life variables (the **Life Parameter** Items). For more details on **OperationalEvent LifeUnit** Relationship Items, refer to section 11.4.3 OperationalEvent LifeUnit Relationships.

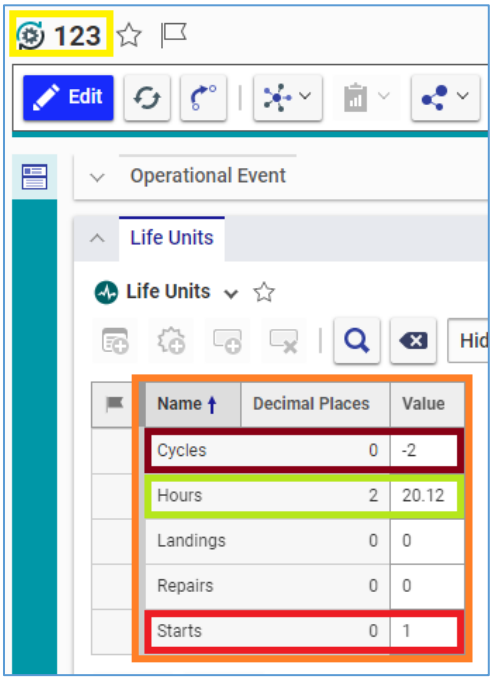


Figure 121.

During the promotion of the **Operational Event** Item to the **Complete** State, the system automatically adds its operational increments to life variables of a performing **Physical Part** Item.

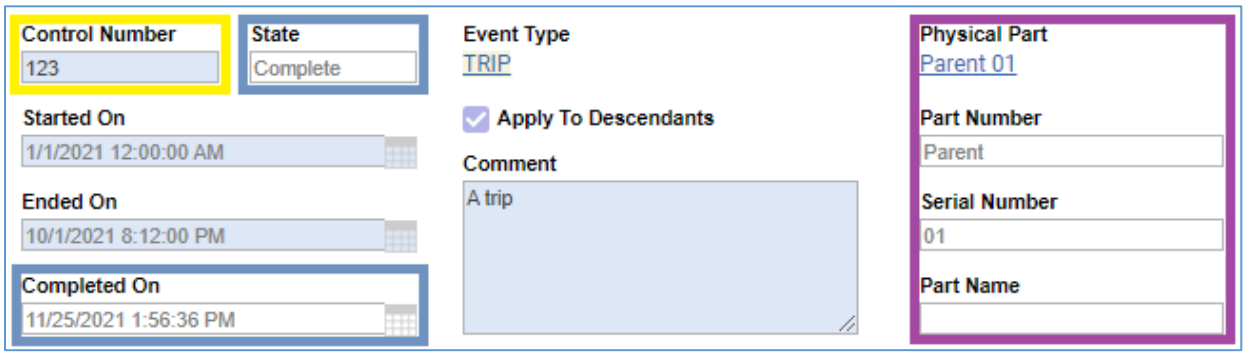


Figure 122.

Namely, the system mathematically adds a numeric value set in the **Value** property of a given **OperationalEvent LifeUnit** Relationship Item to a numeric value in the **Current Value** property of a **PhysicalPart LifeValue** Relationship Item that has the same unit of measure (**Life Unit** Item). For example, the system adds a value given for the **Hours** Life Unit Item in the **Operational Event** Item to every **Life Parameter** Item with the **Hours** Life Unit Item in the **Physical Part** Item.

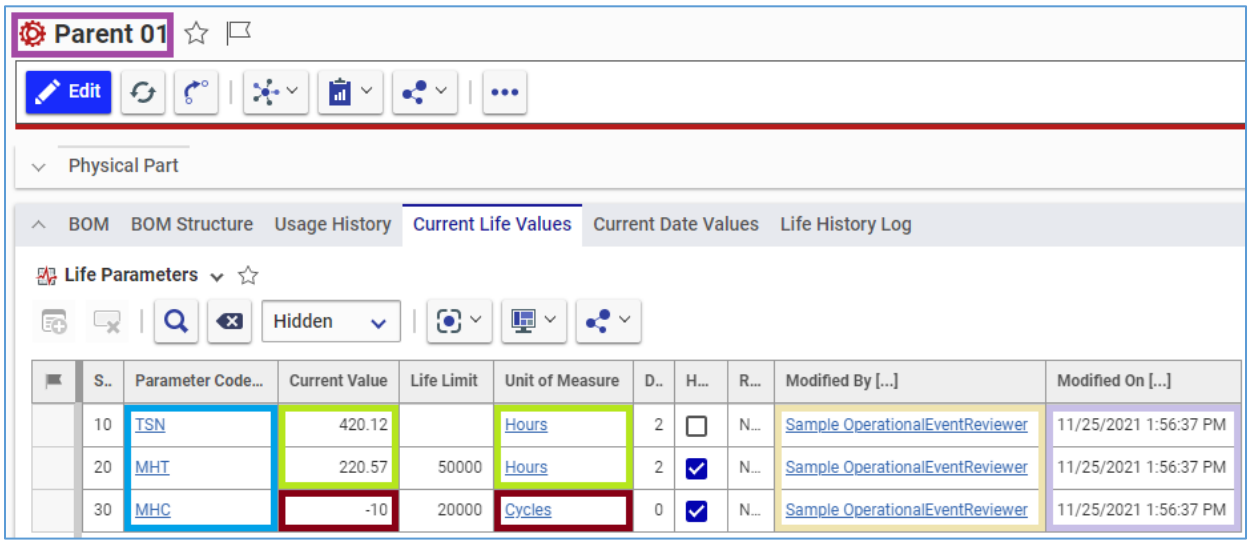
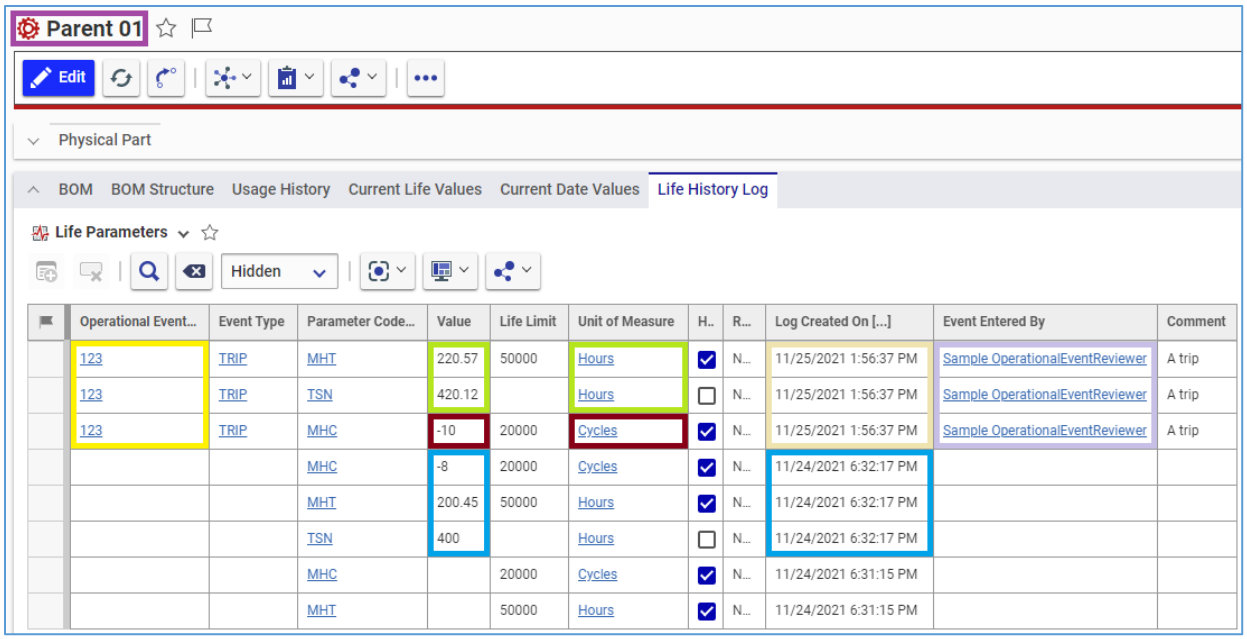


Figure 123.

During updating life variables, the system silently ignores a value entered for a **Life Unit** Item in the **Operational Event** Item if the performing **Physical Part** Item does not include a related **Life Parameter** Item with this **Life Unit** Item. For example, an **Operational Event** Item has the **Starts** Life Unit Item with a value other than zero. And, a **Physical Part** Item has no **Life Parameter** Items that track **Starts**. Thus, the **Starts** Life Unit value is ignored during the **Operational Event** promotion without raising any error or warning dialog or message.

The value of zero (**0**) is always ignored: it does not update a life variable value.

Once the **Operational Event** Item **Complete** State promotion has successfully finished, the **Current Life Values** tab of the **Physical Part** Item shows the life variable values updated by the **Operational Event** Item. The **Life History Log** tab of the **Physical Part** Item keeps the new and old values.



Operational Event...	Event Type	Parameter Code...	Value	Life Limit	Unit of Measure	H..	R...	Log Created On [...]	Event Entered By	Comment
123	TRIP	MHT	220.57	50000	Hours	☑	N...	11/25/2021 1:56:37 PM	Sample OperationalEventReviewer	A trip
123	TRIP	TSN	420.12		Hours	☐	N...	11/25/2021 1:56:37 PM	Sample OperationalEventReviewer	A trip
123	TRIP	MHC	-10	20000	Cycles	☑	N...	11/25/2021 1:56:37 PM	Sample OperationalEventReviewer	A trip
		MHC	-8	20000	Cycles	☑	N...	11/24/2021 6:32:17 PM		
		MHT	200.45	50000	Hours	☑	N...	11/24/2021 6:32:17 PM		
		TSN	400		Hours	☐	N...	11/24/2021 6:32:17 PM		
		MHC		20000	Cycles	☑	N...	11/24/2021 6:31:15 PM		
		MHT		50000	Hours	☑	N...	11/24/2021 6:31:15 PM		

Figure 124.

If the discussed promotion fails, the system rolls back the update of life variables—no changes are applied to the **Current Value** properties of the involved **PhysicalPart LifeValue** Relationship Items. The **Current Life Values** and **Life History Log** tabs of the **Physical Part** Item show the life variable values as before the unsuccessful promotion.

If an **Operational Event** Item has been promoted to the **Complete** state with one or more wrong values in the **Value** properties of its **OperationalEvent LifeUnit** Relationship Items, there are two ways to correct the appropriate, affected **Current Value** property values of the involved **PhysicalPart LifeValue** Relationship Items:

1. To create and promote a new **Operational Event** Item with values that will produce correct **Current Value** property values; see section 11.5 Operational Event Life Cycle and Promotion.

Note: A correctional **Operational Event** Item should be of a specifically dedicated **Operational Event Type**, like **Admin** or **Correction**.

2. To correct wrong **Current Value** property values manually; see section 4.7.2.2 Manual update of life values.

An **Operational Event** Item can have its **Apply To Descendants** property set to **true**.

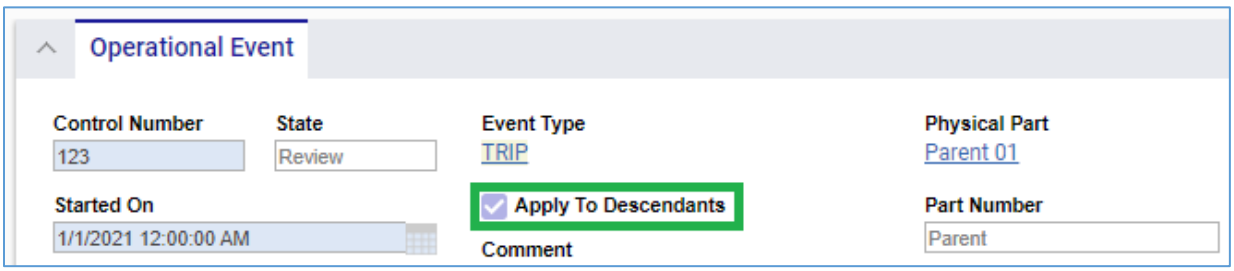


Figure 125.

In this case, during the promotion, the system can update not only life variables of an operational **Physical Part** Item set in the **Operational Event** Item but also the life variables of one or more dedicated **Physical Part** Items that are children in the multi-level Physical Part BOM structure where this operational **Physical Part** Item is a root.

The system traverses the Physical Part BOM structure down from the **Physical Part** Item given in the **Operational Event** Item and acts depending on the **Inherit Life** setting for a given child:

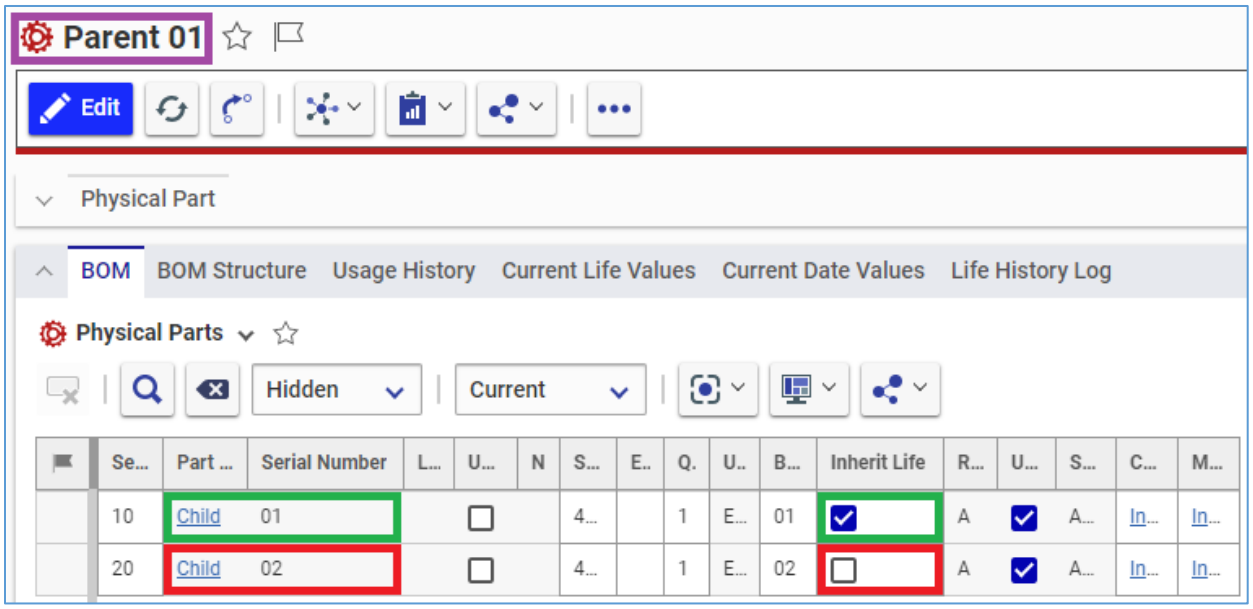


Figure 126.

- **True—**updates life variables of this child, recursively propagates down this child's BOM branch, and updates children there according to their **Inherit Life** settings. The life inheritance continues in this child.

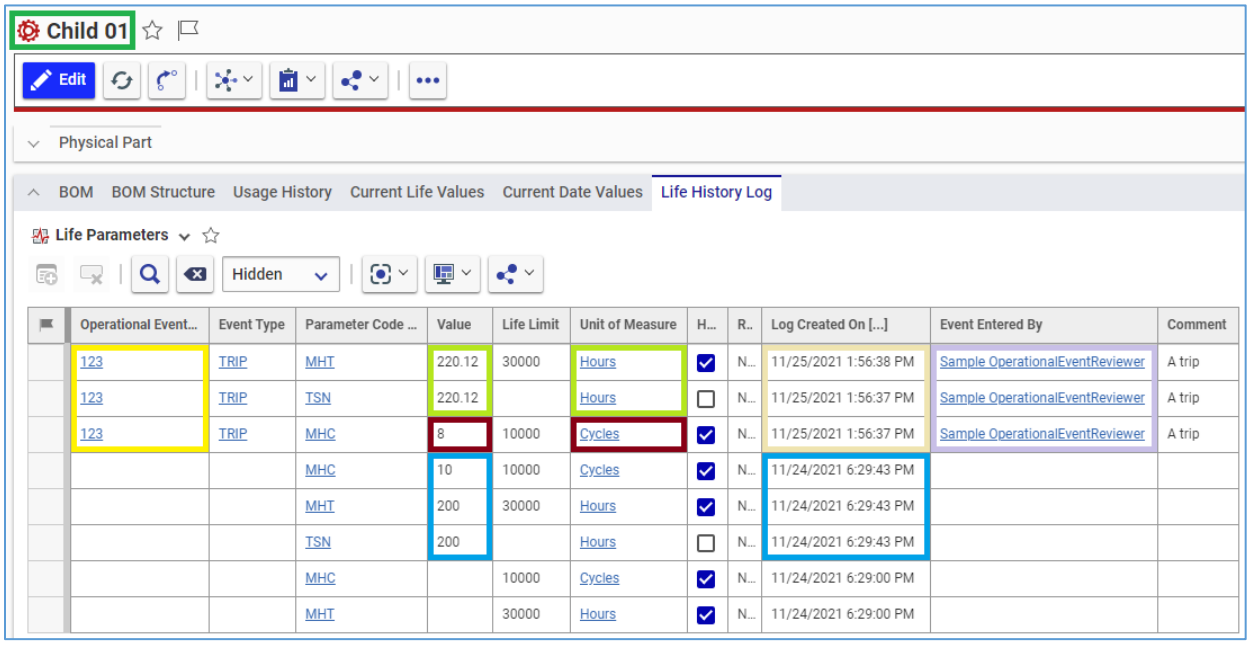


Figure 127.

- **False**—the system does not update this child and ignores all children in this child's BOM branch regardless of their **Inherit Life** settings. The life inheritance breaks in this child.

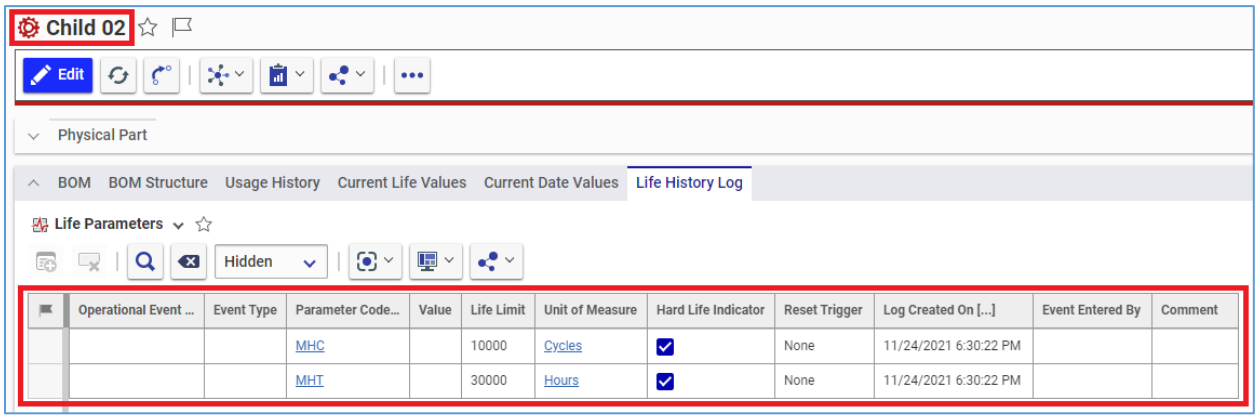


Figure 128.

An **Operational Event** Item can have its **Apply To Descendants** property set to **false**.

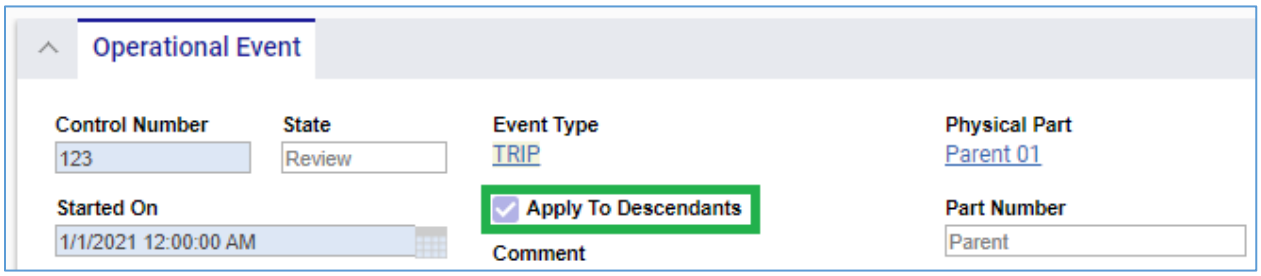


Figure 129.

And, a **Physical Part** Item given in the **Operational Event** Item can have a child with **Inherit Life** set to **true**.

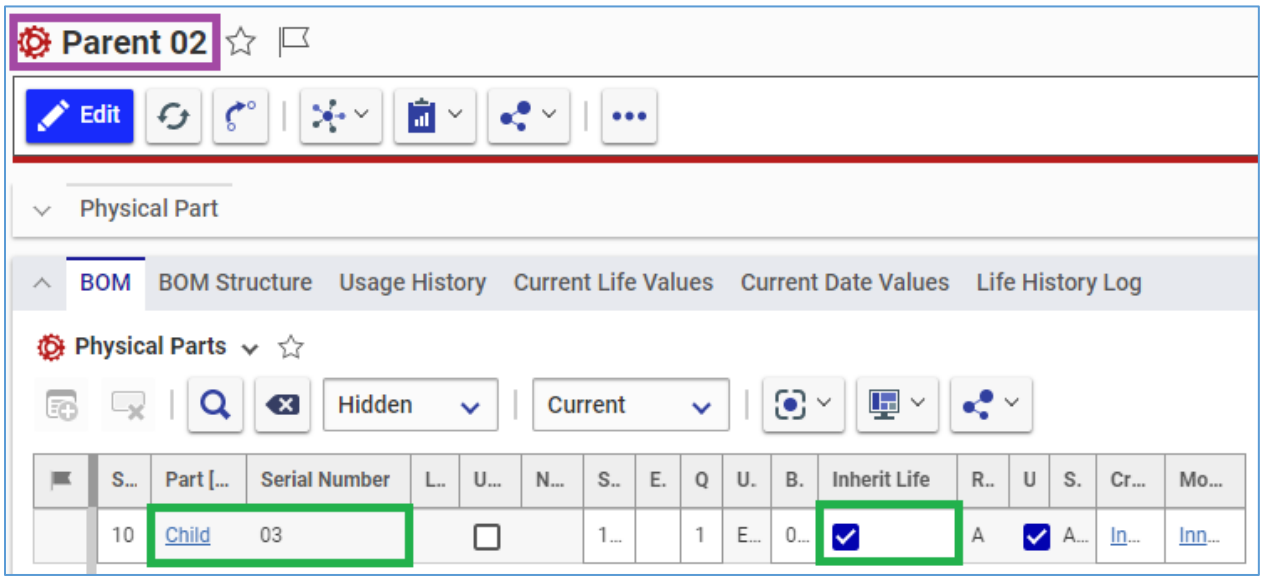


Figure 130.

In this case, the **Operational Event** Item **Complete** State promotion updates only the life variables of the **Physical Part** Item set in the **Operational Event** Item.

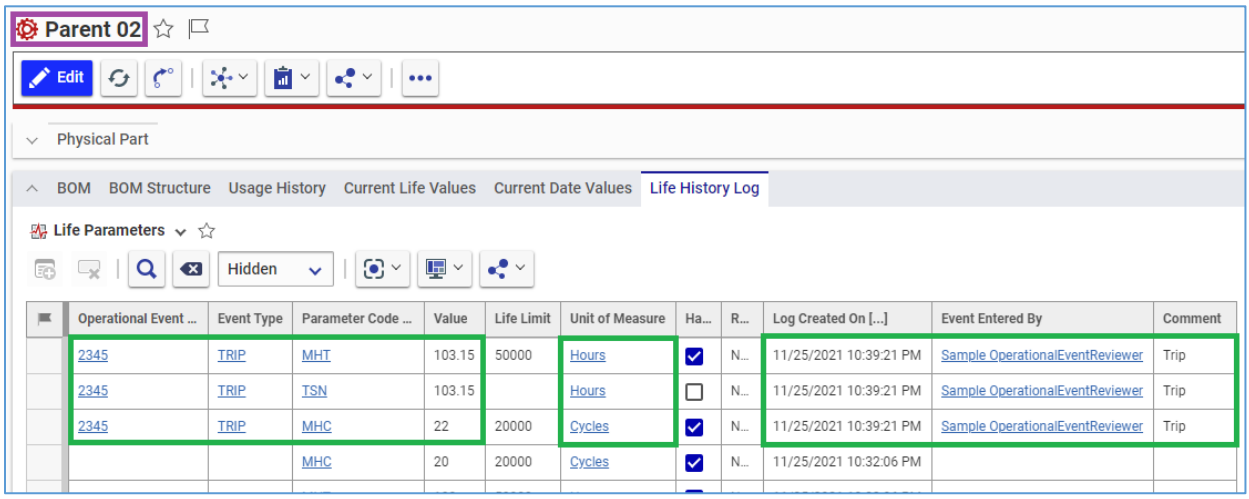


Figure 131.

The life variables of the child in question are not updated even if its **Inherit Life** property defines that it should inherit life from its parent.

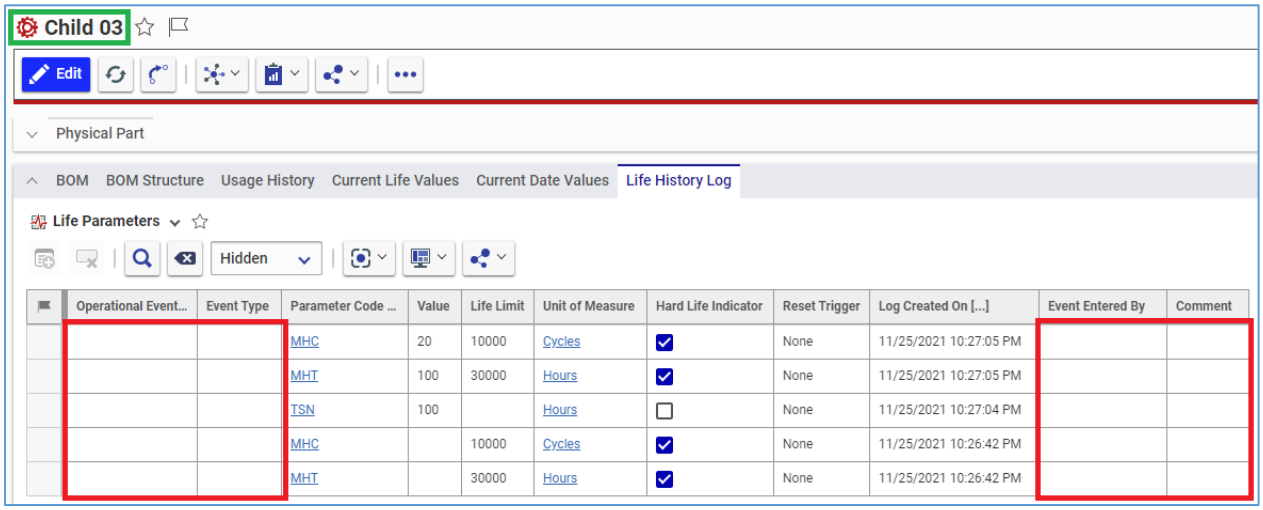


Figure 132.

4.7.3 Current values of Physical Part life dates

Note: Only the **Asset Admin**, **Asset Editor**, or **Asset User** Identity can update the editable properties of the **PhysicalPart DateValue** Relationship Items.

The **Current Date Values** Relationships tab of a **Physical Part** Item hosts a Grid with the **PhysicalPart DateValue** Relationship Items describing relationships between the given **Physical Part** Item and tracking **Life Parameter** Items that express currently upcoming calendar life events of an asset. Each row in the Grid is a separate Relationship Item. The row properties are the Relationship Item properties. The Grid is automatically managed as described in section 4.7.1 Managing Physical Parts with Life Parameters.

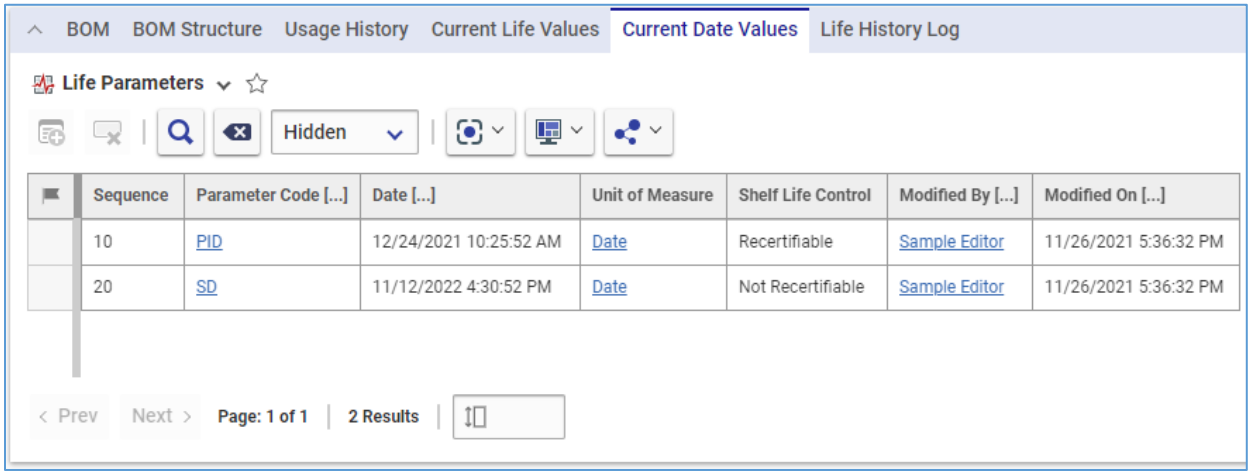


Figure 133.

The **Current Date Values** Relationship Grid shows properties of a **PhysicalPart DateValue** Relationship Item and a related **Life Parameter** Item as follows:

- **Sequence**: a sort order number of the given **Life Parameter** Item in the Grid. It is the same as in the sourcing **Life Controls** Relationships Grid of an applied **Life Policy** Item.
- **Parameter Code**: a view-only **Parameter Code** property value of the **Life Parameter** Item. It is also a link to this **Life Parameter** Item.
- **Date**: an event date and time value. The value is managed automatically when the **Current Life Values** Relationship Grid is populated or repopulated. For details, see section 4.7.1 Managing Physical Parts with Life Parameters.
- **Unit of Measure**: a view-only **Life Unit** Item given in the **Unit of Measure** property of the **Life Parameter** Item. It is also a link to this **Life Unit** Item.
- **Shelf Life Control**: a view-only **Shelf Life Control** property value of the **Life Parameter** Item.
- **Modified By**: the last Identity that modified the **PhysicalPart DateValue** Relationship Item. This property is view-only.
- **Modified On**: a view-only timestamp of the last modification of the **PhysicalPart DateValue** Relationship Item.

In the Grid, the Items are sorted by their **Sequence** values in ascending order (from 0 to 9).

Except for the **Date** cell, the **Life Parameters** Relationships Grid is not editable. The **Date** property is required in any State of the **Physical Part** Item. Upon saving a **Physical Part** Item, an error is raised, and this **Physical Part** Item cannot be saved if it has a **Date** property that is not set.

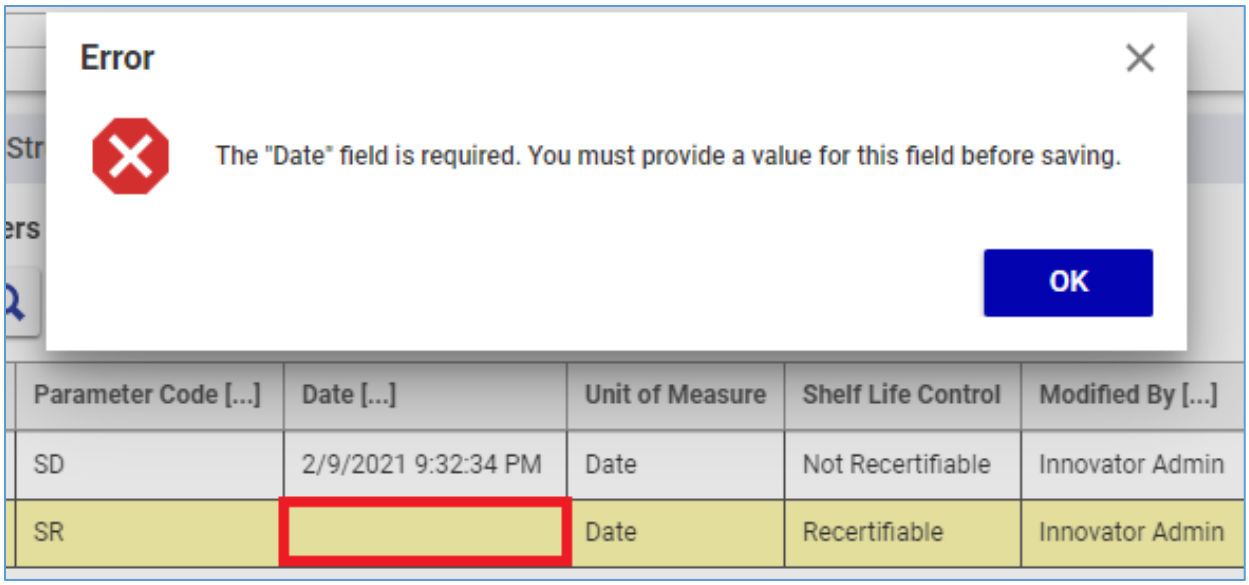


Figure 134.

A **Date** property value should be in the future regardless of a **Physical Part** Item State. Upon saving a **Physical Part** Item, an error is raised, and this **Physical Part** Item cannot be saved if it has a **Date** value that is not later than your current system date and time.

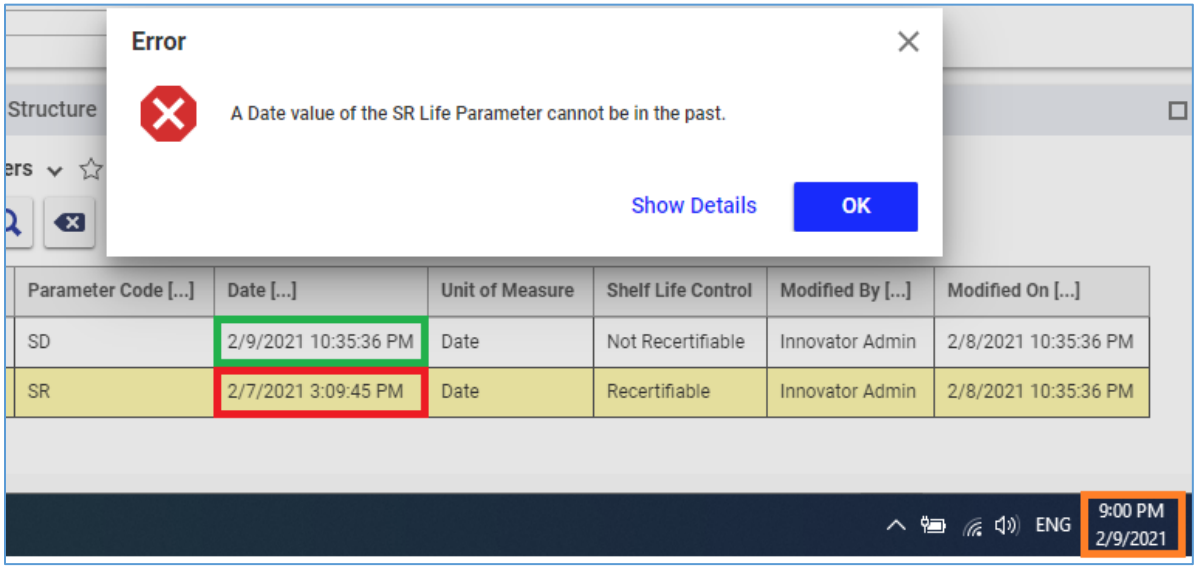


Figure 135.

The ability to modify the **Date** property depends on a given **Shelf Life Control** value:

- **Not Recertifiable**: the **Date** property can be modified only in the **Preliminary** State of a **Physical Part** Item. A represented asset life event can occur only once. For example, it can be a hard expiry date that cannot be prolonged.
- **Recertifiable**: the **Date** property can be modified in any State of a **Physical Part** Item. A represented asset life event can re-occur many times. For example, it can be an annual test inspection.

Upon saving a **Physical Part** Item in the **Active** State, an error is raised, and this **Physical Part** Item cannot be saved if there had been a modification of a **Date** value that is **Not Recertifiable**.

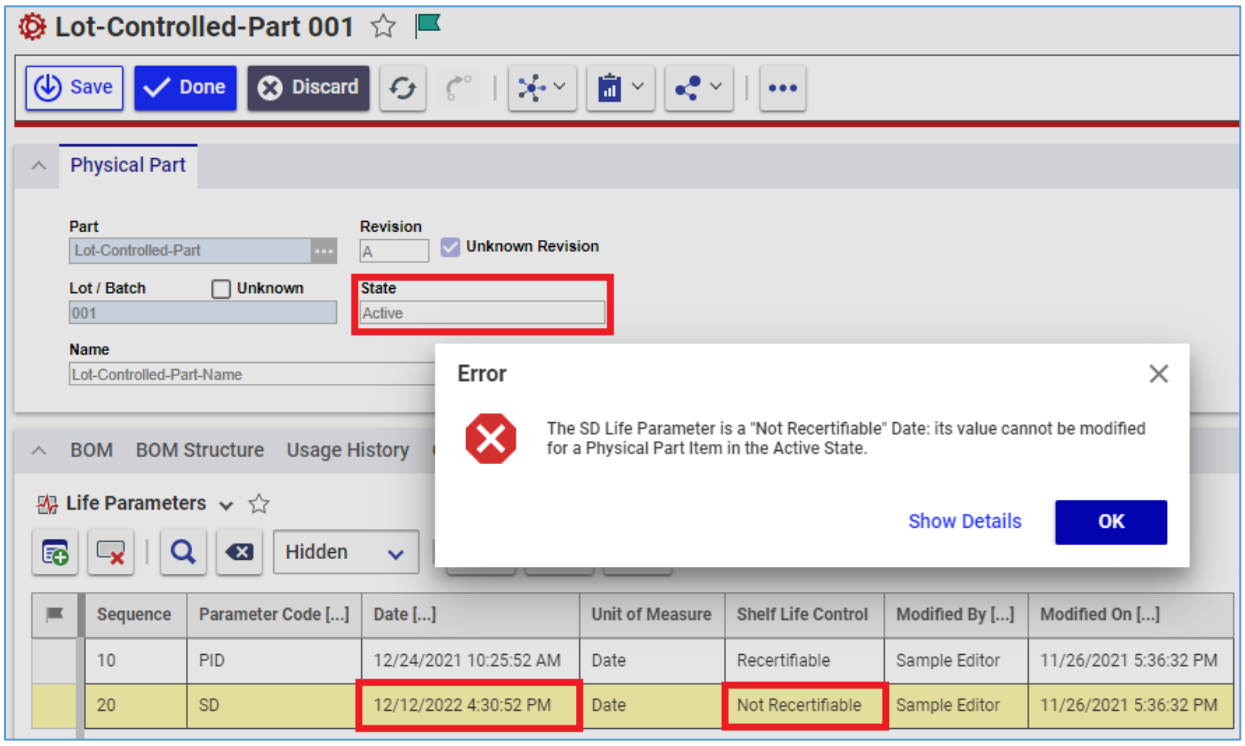


Figure 136.

To correct mistakes, a member of the **Asset Admin** Identity can set the **Date** property to any value, including in the past, in any State of a **Physical Part** Item regardless of a **Shelf Life Control** value.

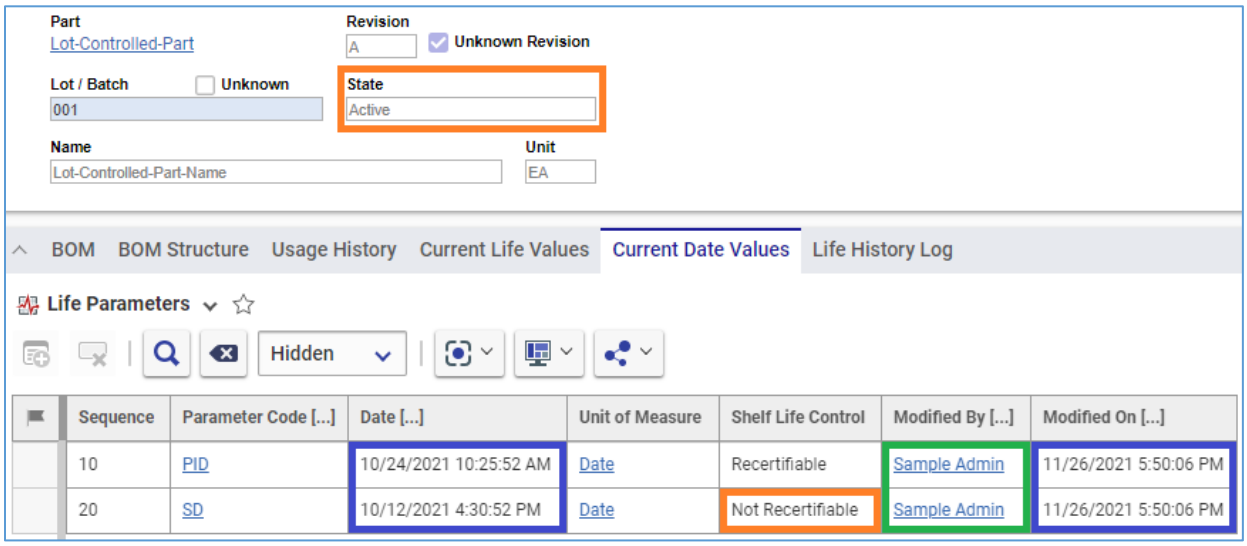


Figure 137.

4.7.4 History of numeric Physical Part life variables

The **Life History Log** Relationships tab of a **Physical Part** Item hosts a Grid with the **PhysicalPart LifeHistoryLog** Relationship Items describing the history of asset life variables and life limits. Each row in

the Grid is a separate Relationship Item. The row properties are the Relationship Item properties. The Grid is fully automatically managed as described in section 4.7.1 Managing Physical Parts with Life Parameters.

A **PhysicalPart LifeHistoryLog** Relationship Item is a snapshot of the following property values of a given **PhysicalPart LifeValue** Relationship Item at a specific point in time:

- **Current Value**
- **Life Limit**

There are two ways to update the abovementioned properties:

- Manually—you enter a new value when allowed; see section 4.7.2.2 Manual update of Life Values.
- Automatically—the system provides a new value as a sum of the old one and an increment from a given **Operational Event** Item when you promote this **Operational Event** Item to the **Complete** State; see section 4.7.2.3 Operational Event update of Life Values.

When any of the properties in question gets a new value, the system automatically creates a new **PhysicalPart LifeHistoryLog** Relationship Item to record this value in the history.

The **Life History Log** Relationship Grid shows properties of a **PhysicalPart LifeHistoryLog** Relationship Item as follows:

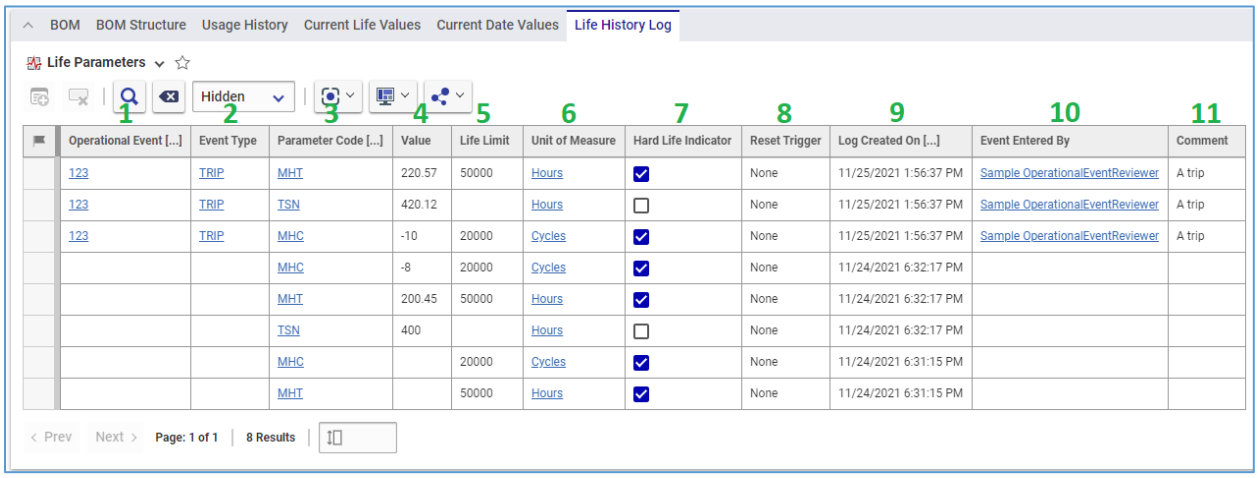


Figure 138.

1. **Operational Event**: a **Control Number** property value of an **Operational Event** Item which promotion to the **Complete** State outputted a value recorded in the **Value** (4) property.

   The system populates this property only when the **Operational Event** Item promotion updates the **Current Value** property. It is always blank for the manual updates of the **Current Value** and **Life Limit** properties.

2. **Event Type**: an **Event Type** property value of the given **Operational Event** Item (1). It is also a link to this **Operational Event Type** Item.

   The system populates this property only when the **Operational Event** Item promotion updates the **Current Value** property. It is always blank for the manual updates of the **Current Value** and **Life Limit** properties.

3. **Parameter Code**: a **Parameter Code** property value of a given **Life Parameter** Item that defines an asset life variable as a related Item in the given **PhysicalPart LifeValue** Relationship Item. It is also a link to this **Life Parameter** Item.

4. **Value**: a numeric value that the **Current Value** property of the given **PhysicalPart LifeValue** Relationship Item had at a given time point (9).
5. **Life Limit**: a numeric value that the **Life Limit** property of the given **PhysicalPart LifeValue** Relationship Item had at a given time point (9).
6. **Unit of Measure**: a **Unit of Measure** property value of the given **Life Parameter** Item (3). It is a link to a **Life Unit** Item that defines how the given **Life Parameter** Item (life variable) is tracked.
7. **Hard Life Indicator**: a **Hard Life Indicator** property value of the given **Life Parameter** Item (3). It is a flag indicating when selected (**true**) that the given **Life Parameter** Item limits the life of the source **Physical Part** Item.
8. **Reset Trigger**: a **Reset Trigger** property value of the given **Life Parameter** Item. It is a setting that defines whether the given **Life Parameter** Item actual values can be reset.
9. **Log Created On**: a timestamp of the **PhysicalPart LifeHistoryLog** Relationship Item creation.

   A date and time value in this property is either equal to or slightly different from the one in the **Modified On** property of the corresponding given **PhysicalPart LifeValue** Relationship Item because this property should record the date and time of saving this **PhysicalPart LifeValue** Relationship Item updated with a new **Current Value** (4) or **Life Limit** (5) property value.

   In the case of an update using an **Operational Event** (1), the discussed value should also be either equal to or slightly different from the one in the **Completed On** property of the given **Operational Event** Item.
10. **Event Entered By**: an Identity who promoted the given **Operational Event** Item (1) to the **Complete** state.

    The system populates this property only when the **Operational Event** Item promotion updates the **Current Value** property. It is always blank for the manual updates of the **Current Value** and **Life Limit** properties.
11. **Comment**: a **Comment** property value of the given **Operational Event** Item.

    The system populates this property only when the **Operational Event** Item promotion updates the **Current Value** property. It is always blank for the manual updates of the **Current Value** and **Life Limit** properties.

The **Life History Log** Relationships Grid is view-only. You are not allowed to edit any properties of a **PhysicalPart LifeHistoryLog** Relationship Item—an error is raised upon saving a **Physical Part** Item that includes such a property with a manual change. For details, see section 3.7 Automatically managed DTC Relationships Grids.

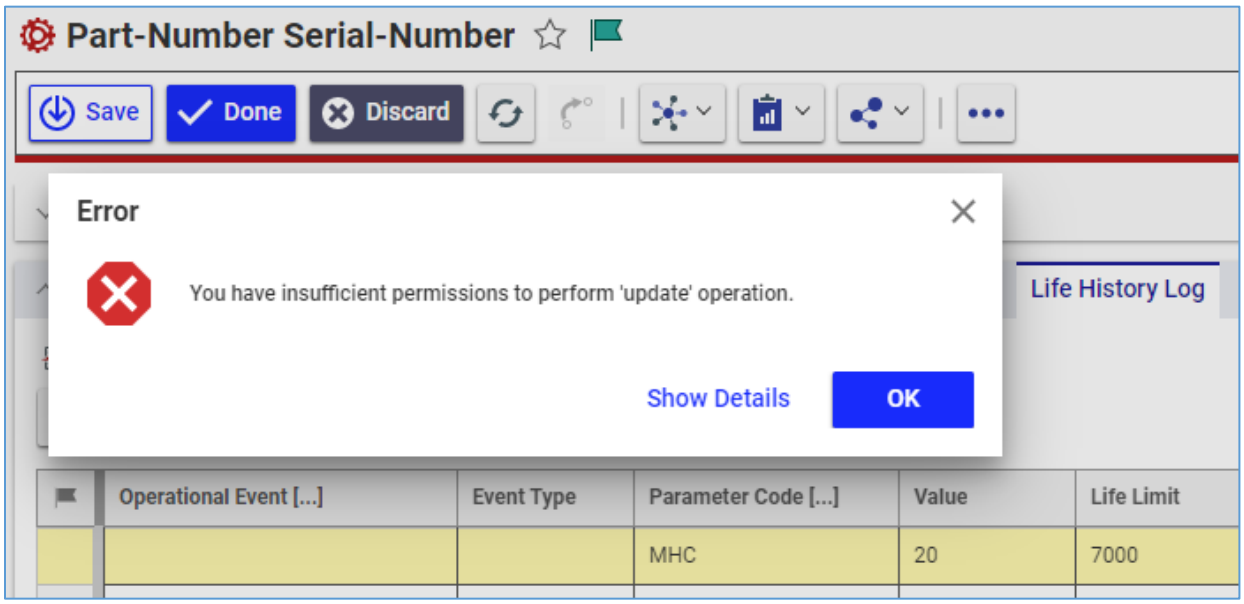


Figure 139.

4.8 Managing Physical Parts

Your ability to manage a **Physical Part** Item depends on your Asset Identity membership, current Item State, and Item Relationships.

4.8.1 Creating Physical Parts

Note: Only an **Asset Admin** or **Asset Editor** Identity member can create **Physical Part** Items.

To create a new **Physical Part** Item:

1. Ensure that a governing **Part Policy** Item in the **Active** State exists for a **Part** Item from which the new **Physical Part** Item should derive. If there is none, create the required **Part Policy** Item and promote it to the **Active** State see sections 9.6.1 Creating Part Policies and 9.5 Part Policy Life Cycle and Promotion.
2. Go to **Contents** --> **Assets** --> **Physical Parts**.
3. Click **Create New Physical Part**. The **Physical Part** # Item view appears.

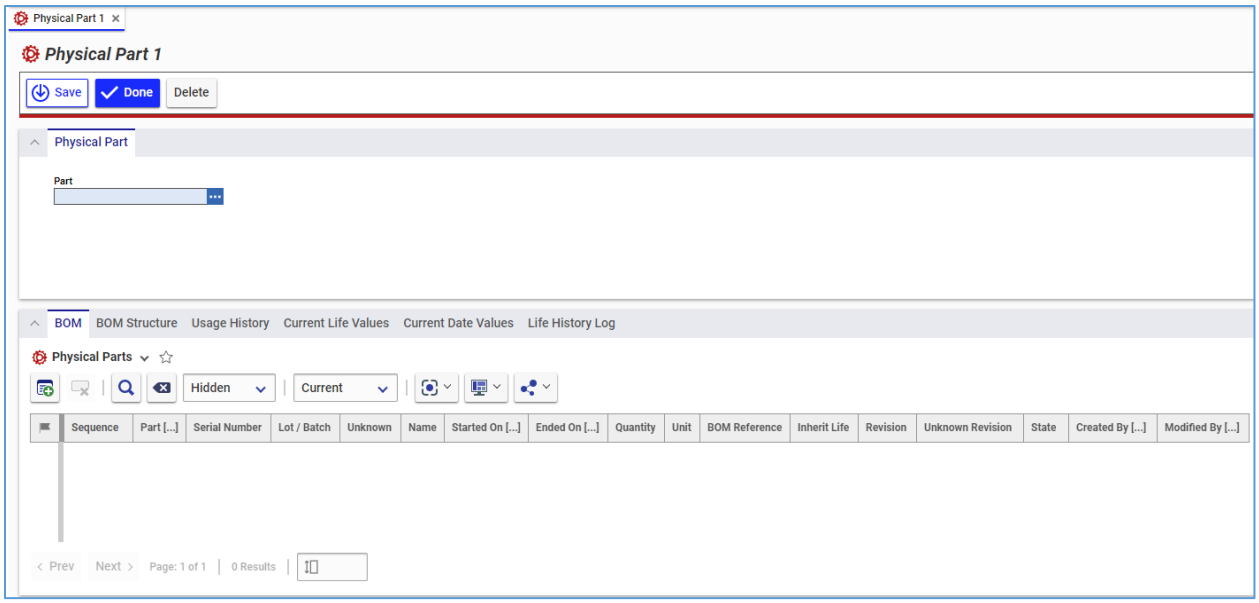


Figure 140.

4. Click the **ellipsis** button in the **Part** field. The **Select Items - Parts** dialog appears.

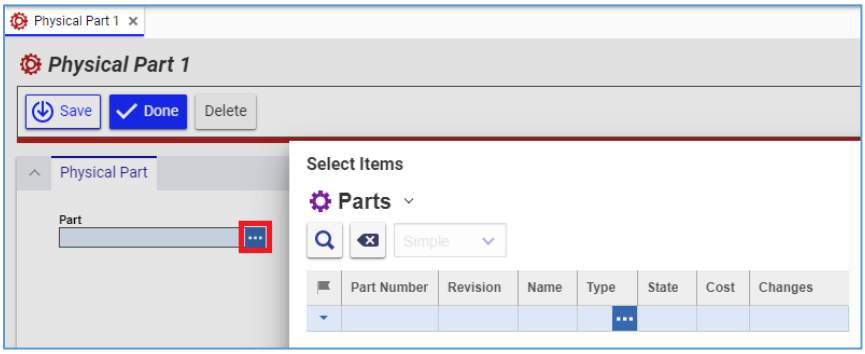


Figure 141.

5. Using the standard search procedure, search for the **Part** Item from which the **Physical Part** Item should derive.

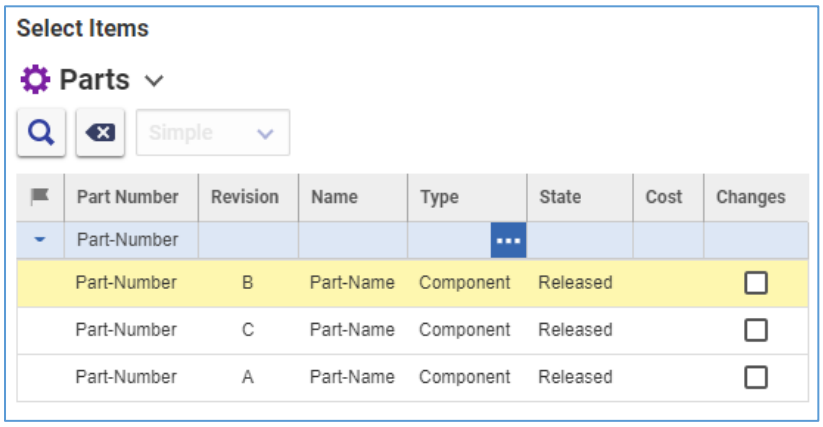


Figure 142.

6. Select the **Part** Item **Revision** to which the **Physical Part** Item belongs and click **OK**. If you do not know to which **Revision** the **Physical Part** Item belongs, select the latest **Revision**. Properties of the selected **Part** automatically populate the **Physical Part** form. The form gets fields according to the **Part** Item **Control Type**.

Note: When creating a non-controlled **Physical Part** Item, always select the latest **Revision** because this Item will represent all Revisions of a given **Part** Item.

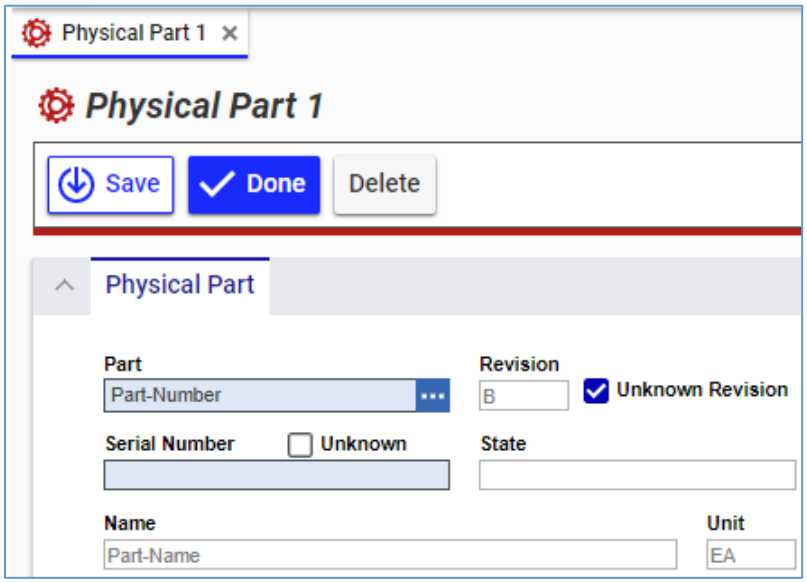


Figure 143.

Warning You can specify a **Part** Item for a **Physical Part** Item only once. After you have saved a **Physical Part** Item with the **Part** property set to a **Part** Item, you will not be able to reset this property to another **Part** Item. If trying to save a **Physical Part** Item with such a change, the system will not allow you to save this **Physical Part** Item and raise an appropriate error message; see section 4.8.4 Validation of Physical Parts.

7. If you are sure that the **Physical Part** Item belongs to the selected **Revision**, clear the **Unknown Revision** check box that is selected automatically by default after the **Part** field is populated.

Note: When creating a non-controlled **Physical Part** Item, always keep the **Unknown Revision** check box selected because such an Item will represent all Revisions of a given **Part** Item.

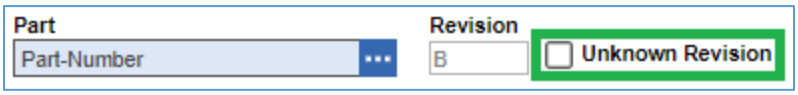


Figure 144.

8. If the **Physical Part** Item is controlled, in the appropriate field (**Serial Number** or **Lot/Batch**), enter ***an inventory control identifier*** (***serial*** or ***lot number***) of the represented physical asset exactly as this asset bears it.

   If you do not know the asset inventory control identifier when creating the representing **Physical Part** Item, you can skip this step and provide the identifier later.

Note: The white space characters are not allowed in the inventory control identifiers: the system clears an entered ***inventory control identifier*** from them upon saving a **Physical Part** Item.

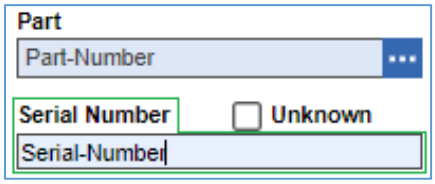


Figure 145.

9. If you are sure that the ***inventory control identifier*** (***serial*** or ***lot number***) you have entered in the **Serial Number** or **Lot/Batch** field in the previous step exactly matches the one that the represented asset bears, keep the **Unknown** check box clear. If not, select this check box.

Figure 146.

Note: If your organization has configured the DTC application to provide dummy serial and lot numbers using Sequences, and if you are not sure about the inventory control identifier correctness, keep the **Serial Number** or **Lot/Batch** field empty in step 8 and select the **Unknown** check box in step 9. The system will populate this field with a ***serial*** or ***lot number*** generated from a Sequence upon saving this newly created **Physical Part** Item (step 11 or 17).

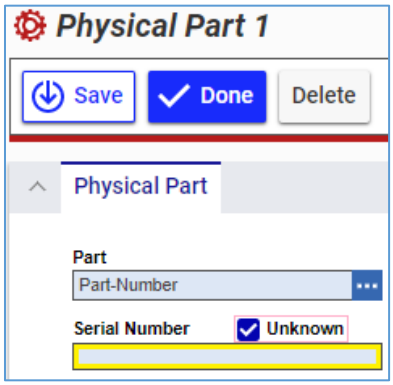


Figure 147.

10. If the **Physical Part** Item tracks the life variables of the represented asset, go to the next step. If otherwise, proceed to the last step (17). For details on asset life variables, see section 4.7 Tracking Lives of Physical Parts.
11. Click **Save** on the **Physical Part** Item toolbar. The **Current Life Values** and **Current Date Values** Relationships tabs are populated with **Life Parameter** Items sourced from a **Life Policy** Item applied to the governing **Part Policy** Item.
12. If the **Physical Part** Item tracks the numeric asset life variables, go to the **Current Life Values** Relationships tab. If not, go to step 15.

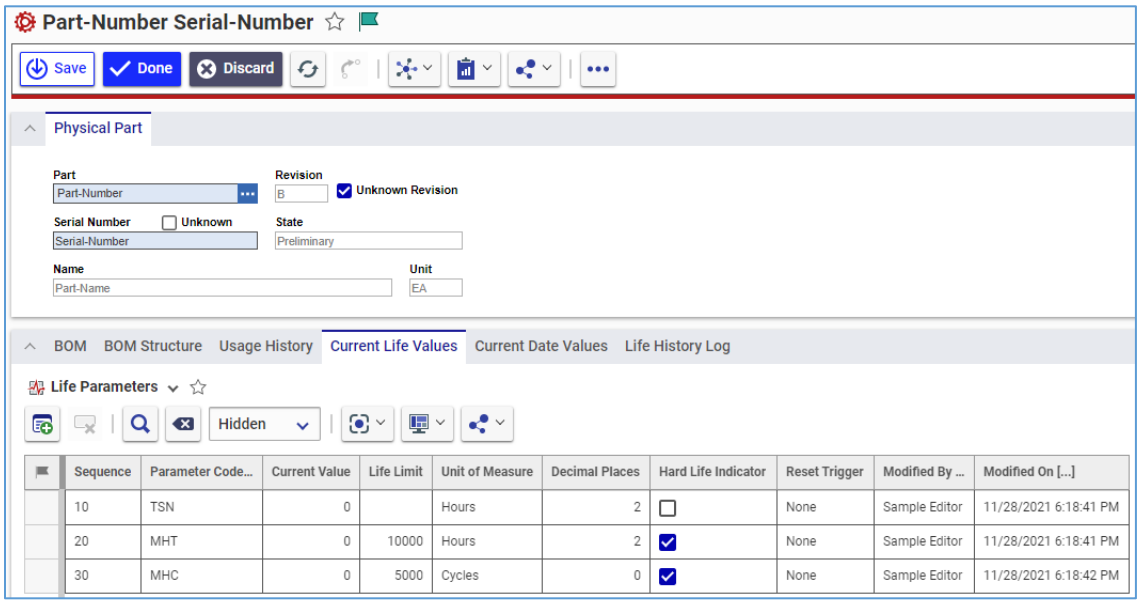


Figure 148.

13. In the **Current Value** cell for each related **Life Parameter** Item, enter a matching numeric value that the asset has accumulated currently. If you do not know the asset current value when creating the **Physical Part** Item, you can skip this step and provide the value later.

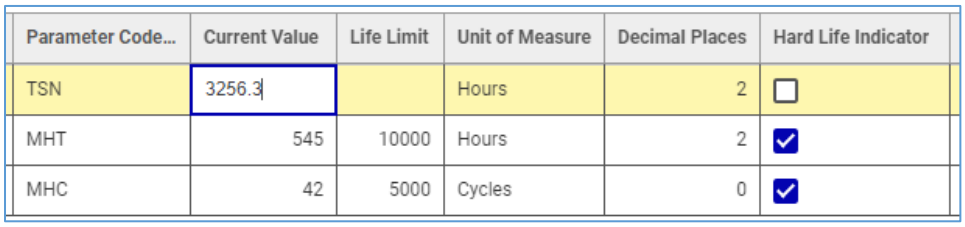


Figure 149.

14. In the **Life Limit** cell for a related **Life Parameter** Item that limits the asset life, correct the life-limiting numeric value if necessary. If a value automatically sourced from the governing **Part Policy** Item is OK in some cells, skip such cells.

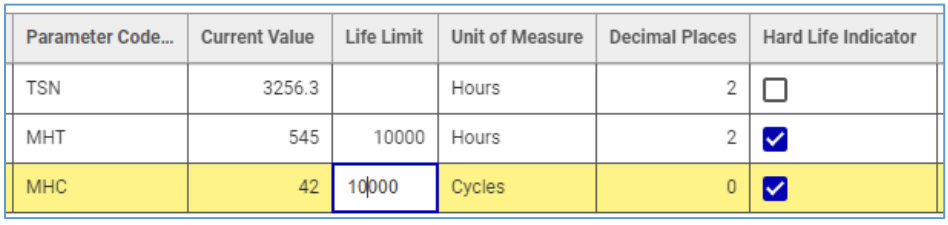


Figure 150.

15. If the **Physical Part** Item tracks the asset calendar life events, go to the **Current Date Values** Relationships tab. If not, go to the last step (17).

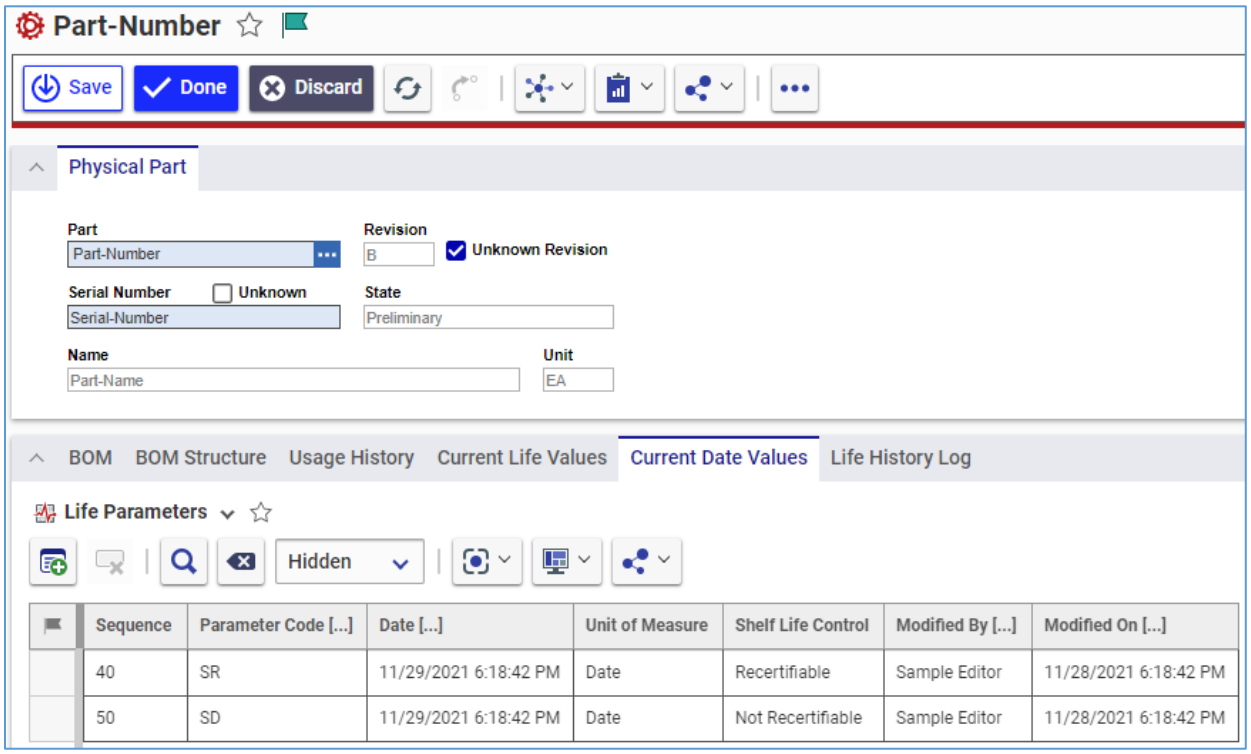


Figure 151.

16. In the **Date** cell of each **Life Parameter** Item, enter a date and time value for a matching asset calendar life event that will happen in the future. If an automatically assigned value is OK in some cells, skip such cells.

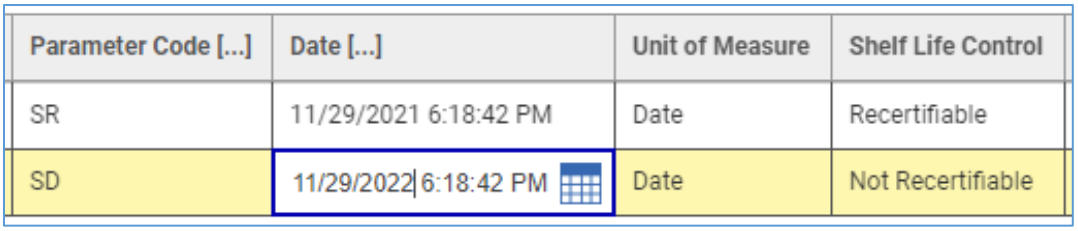


Figure 152.

17. Click either **Save** or **Done** on the **Physical Part** Item toolbar.
    The new **Physical Part** Item is created and is in the **Preliminary** State.

The **Physical Part** Item **Innovator Tab** and **Page Title** bar show the Item properties as follows:

Part Number space ***inventory control identifier*** (***serial*** or ***lot number***)

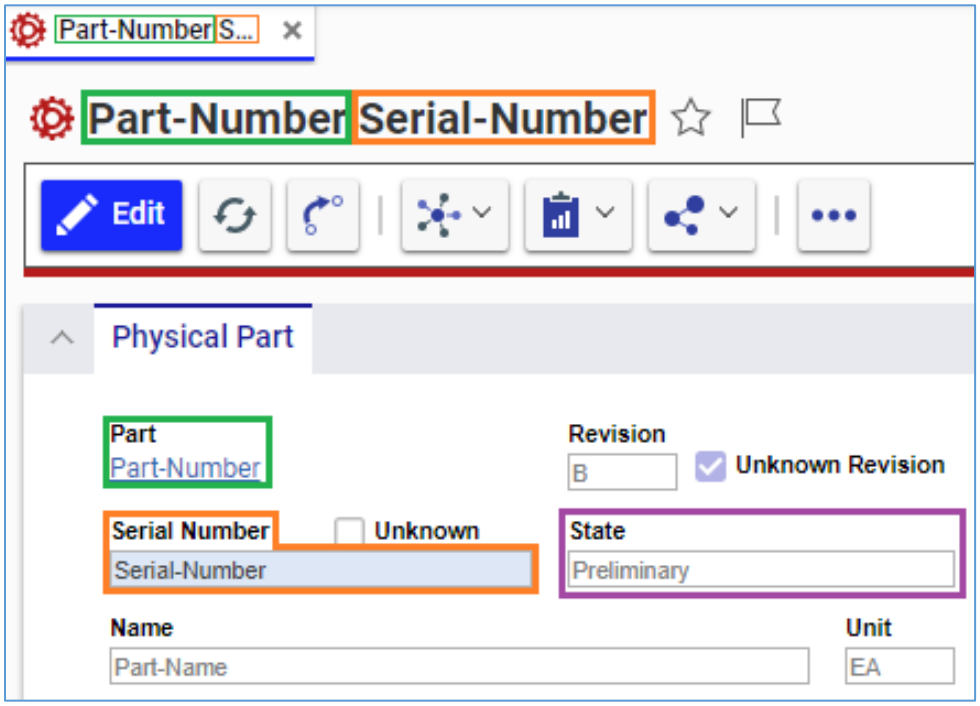


Figure 153.

If the **Physical Part** Item tracks the numeric asset life variables, the **Life History Log** Relationships tab shows the life variable and life limit values entered during **Physical Part** Item creation.

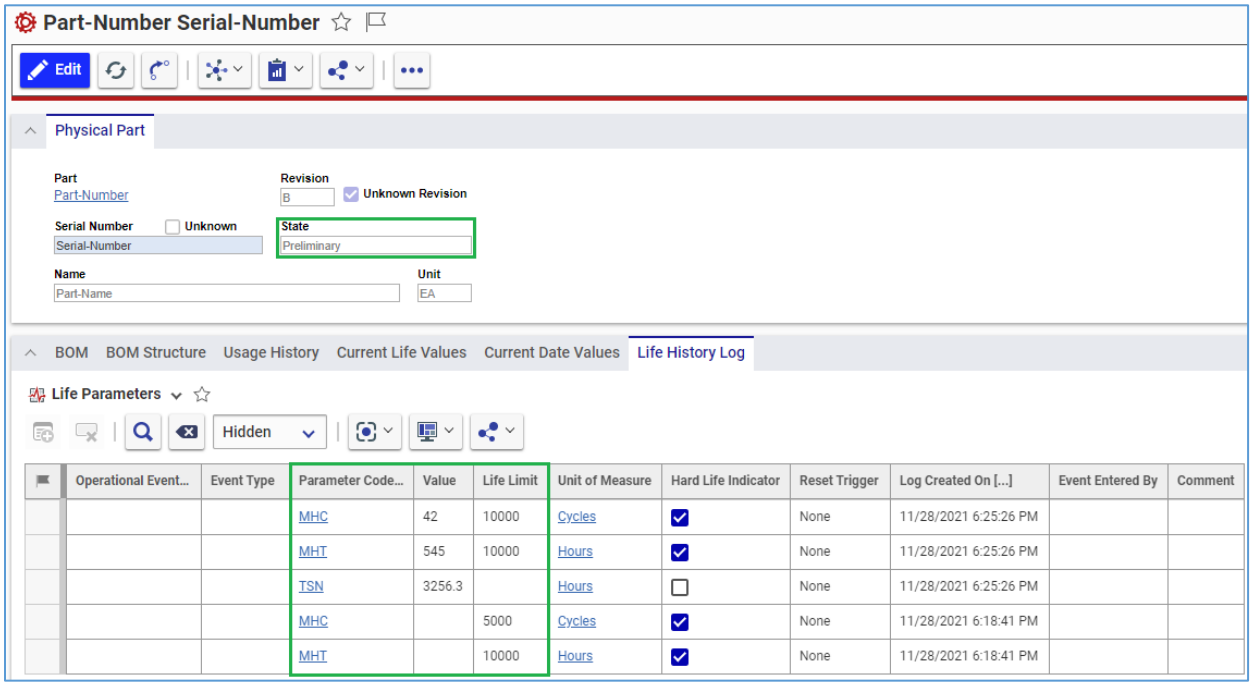


Figure 154.

The **Physical Part** Item is now available using search, subject to the **Asset** Permissions.

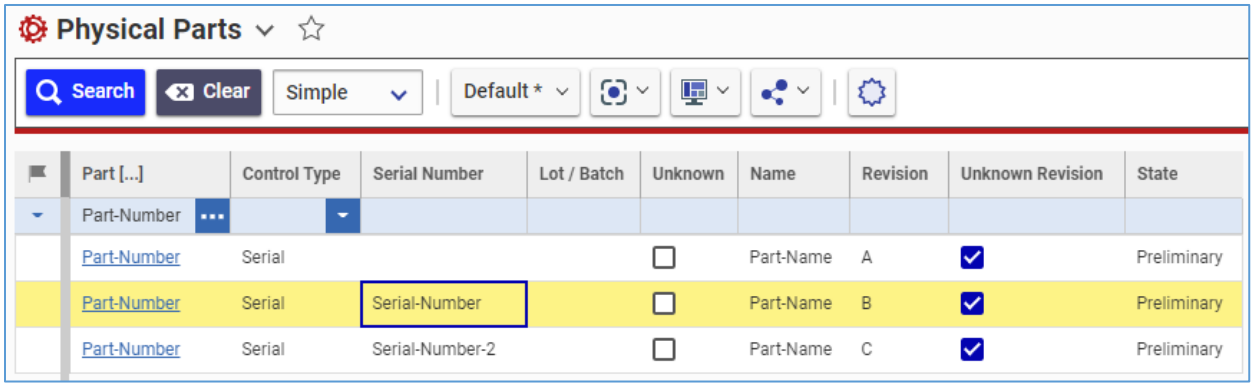


Figure 155.

Upon saving a **Physical Part** Item, the system validates this Item and raises an intelligent error if the validation fails, for example, if the Item has a blank field for a required property or a duplicating **Serial Number** value, discussed in section 4.8.4 Validation of Physical Parts.

4.8.2 Modifying Physical Parts

Note: Only an **Asset Admin** Identity member can modify the editable properties of a **Physical Part** Item regardless of its State. An **Asset Editor** Identity member can do this if the Item is in the **Preliminary** State.

To modify a **Physical Part** Item, use the standard modifying procedure.

Since DTC 12.0R3, nobody can reconnect the **Physical Part** Item to another **Part** Item in any **Physical Part** Item State.

Upon saving a **Physical Part** Item, the system validates this Item and raises an intelligent error if the validation fails, for example, if the Item has a blank field for a required property or a duplicating **Serial Number** value, discussed in section 4.8.4 Validation of Physical Parts.

4.8.3 Deleting Physical Parts

Note: Only an **Asset Admin** Identity member can delete **Physical Part** Items regardless of their State. Also, an **Asset Editor** Identity member can do this if the Item is in the **Preliminary** State.

To delete a **Physical Part** Item, use the standard deletion procedure.

An existing **Physical Part** Item in any State can be entirely deleted from the system only if it is not used either:

- In an **Operational Event** Item, for details, see section 11.4.2 Operational Event Item and Physical Part Item connection.
- As a child in a Physical Part BOM structure, for details, see section 5.2 Single-level Physical Part BOMs.

To enable the deletion of a used **Physical Part** Item, you should remove it from where it is used. A **Physical Part** Item cannot be removed from an **Operational Event** Item in the **Complete** State, and such an **Operational Event** Item cannot be deleted.

An error is raised when trying to delete a used **Physical Part** Item.

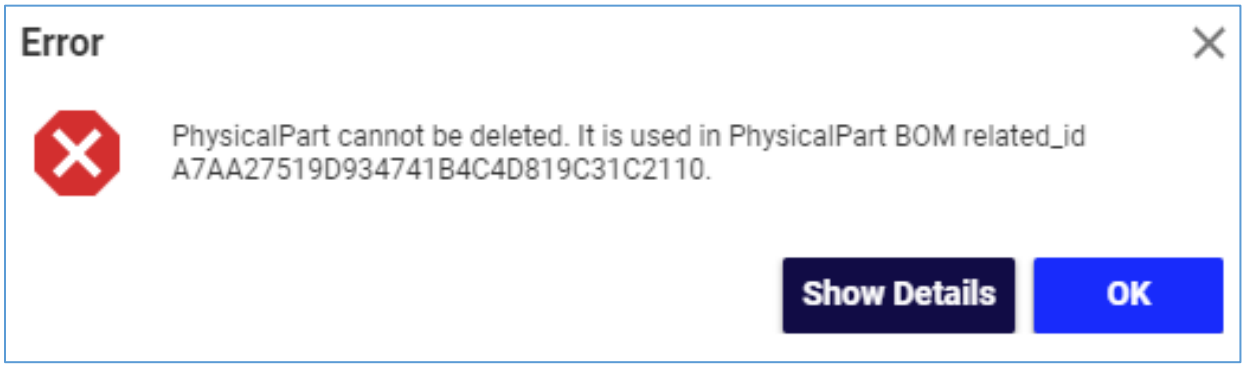


Figure 156.

4.8.4 Validation of Physical Parts

Once you click **Save** or **Done** on a **Physical Part** Item toolbar, the system validates this Item. The system does not save the Item and raises an appropriate error message if this Item either:

- Is missing a **Part** value.

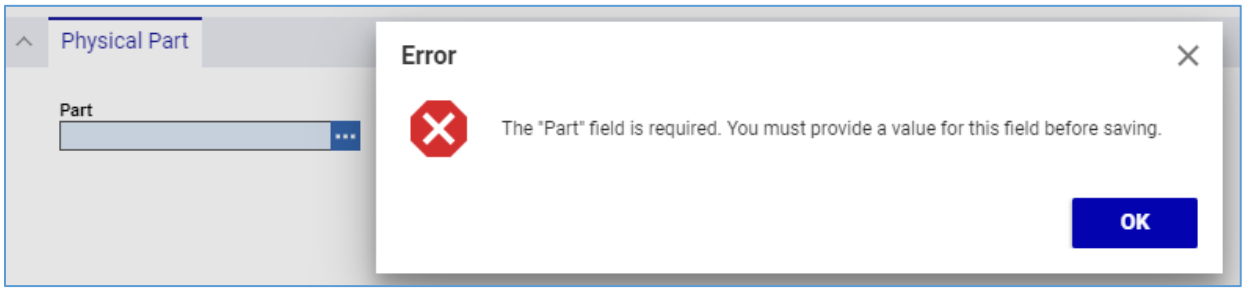


Figure 157.

- Does not have a governing **Part Policy** Item in the **Active** State.

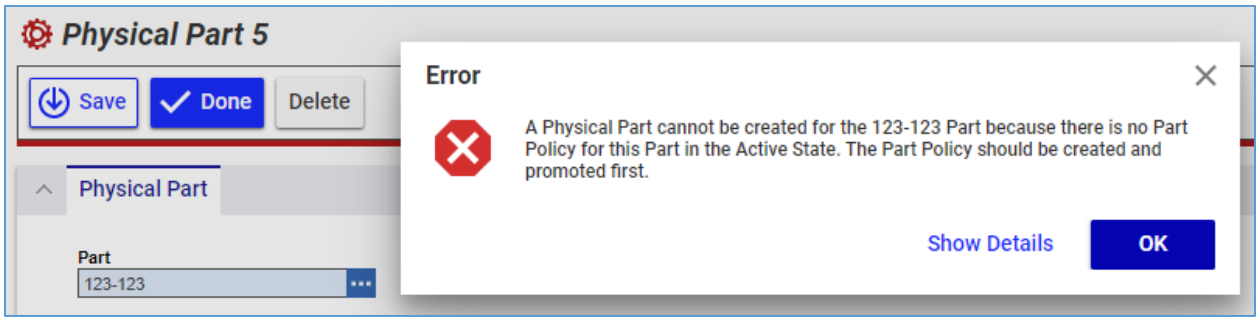


Figure 158.

- Has the **Part** property reset to another **Part** Item.

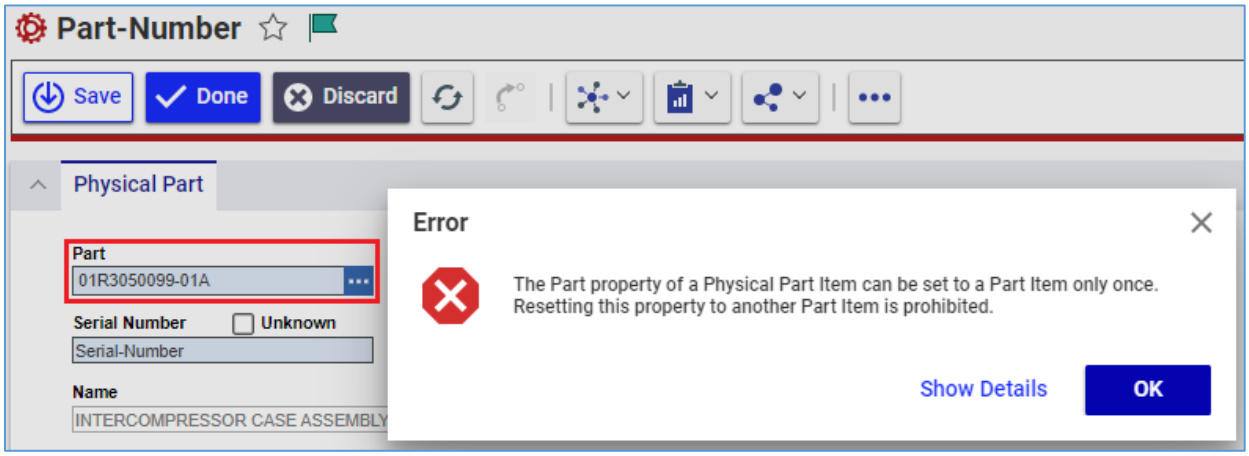


Figure 159.

- Has a **Serial Number** or **Lot / Batch** property value that is already given to another **Physical Part** Item rooted in the same **Part** Item.

Note: More than one **Physical Part** Items deriving from the same **Part** Item can have the same **Serial Number** or **Lot / Batch** property value if uniqueness is switched off for these properties. For details, see section 4.1.8 Physical Part serial and lot number uniqueness.

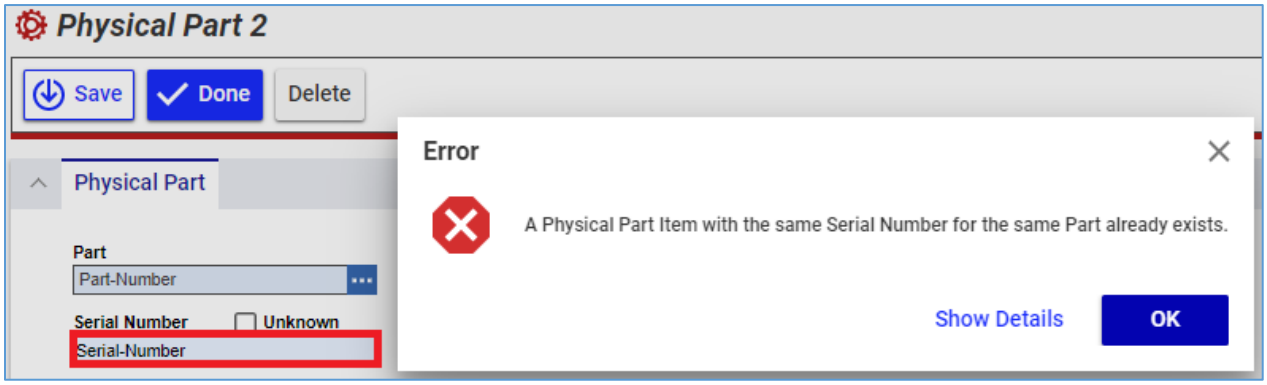


Figure 160.

- Has a **PhysicalPart LifeValue** Relationship Item with the **Current Value** property being empty.

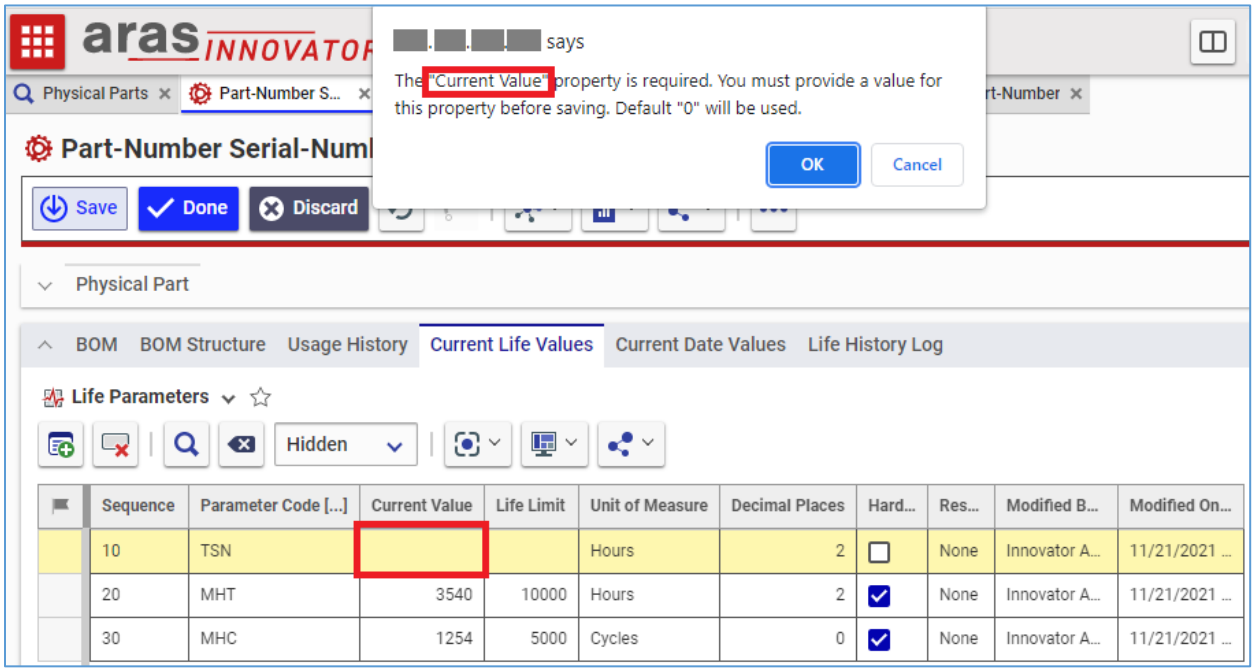


Figure 161.

- Has a **Current Value** or **Life Limit** property value in the **Current Life Values** Relationships tab with more decimal digits than the matching **Decimal Places** property allows.

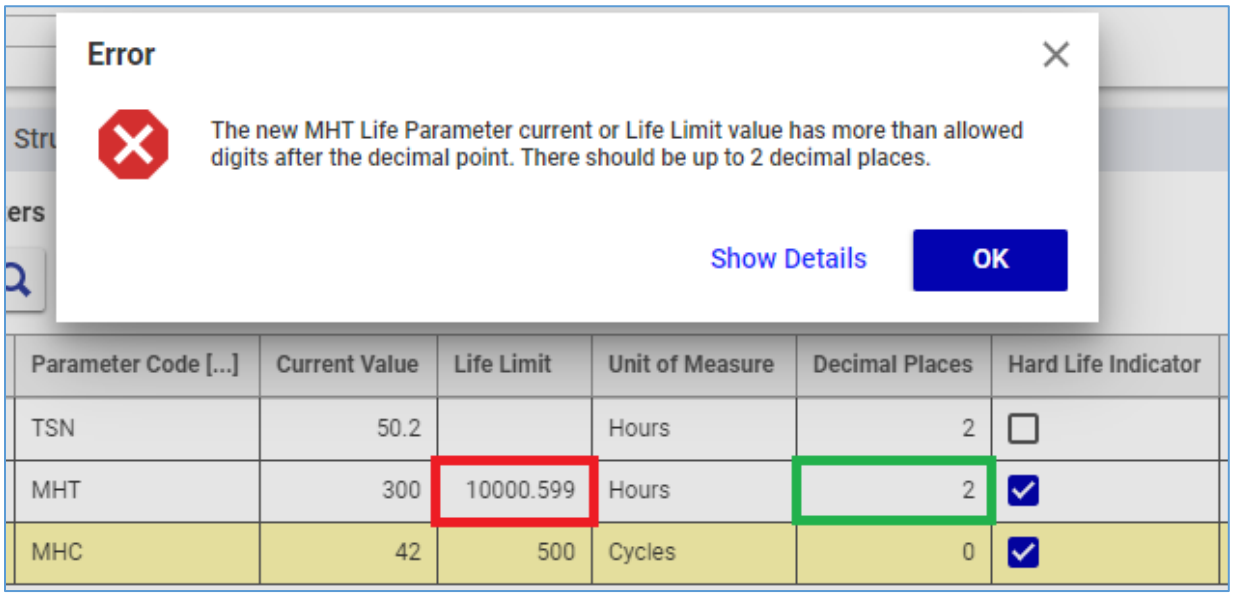


Figure 162.

- Has a **Life Limit** cell in the **Current Life Values** Relationships tab that is populated for a related **Life Parameter** Item with **Hard Life Indicator** set to **false**.

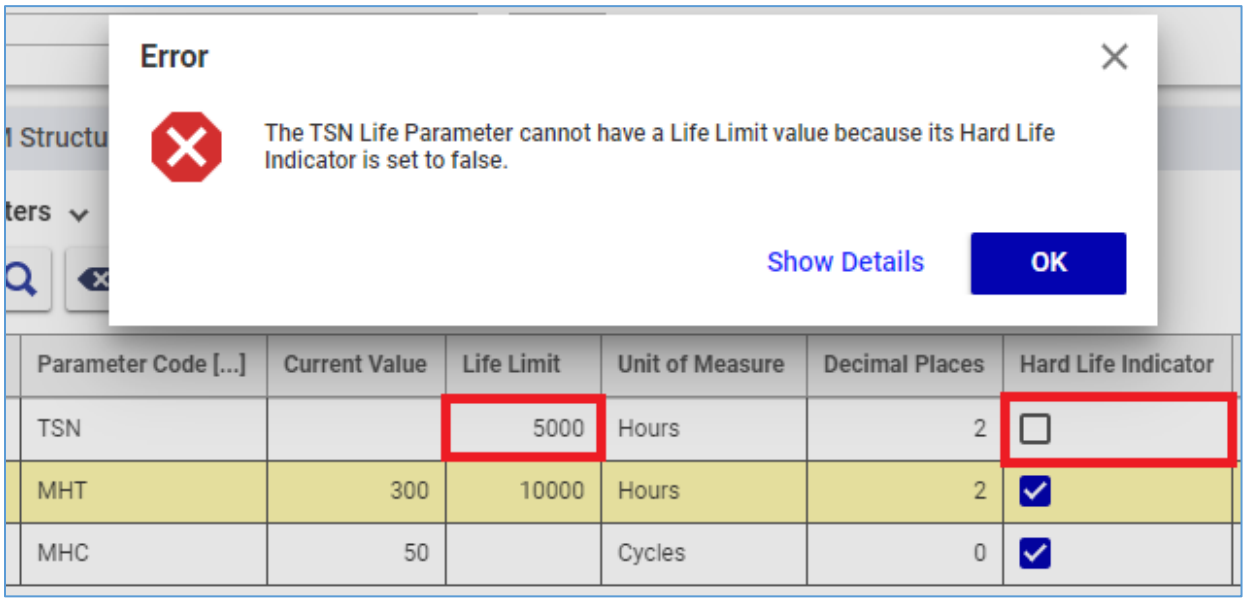


Figure 163.

- Has a **Date** cell in the **Current Date Values** Relationships tab that is empty.

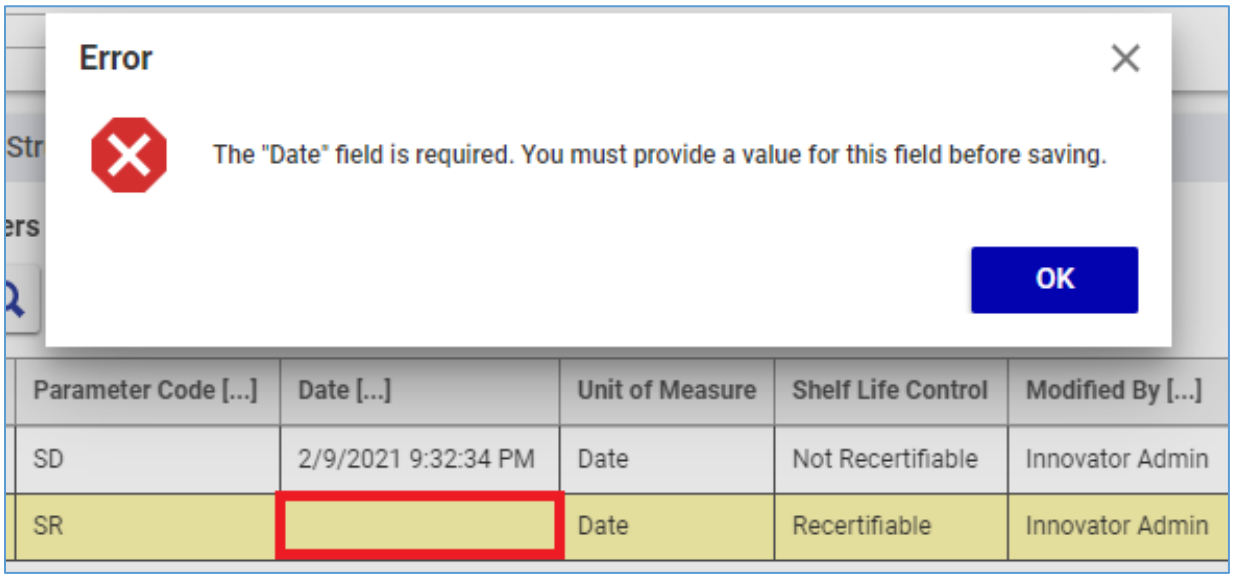


Figure 164.

- Has a **Date** cell in the **Current Date Values** Relationships tab with a given value that is not later than your current system date and time.

Note: Only an **Asset Admin** can set a **Date** cell in the **Current Date Values** Relationships tab to a value in the past.

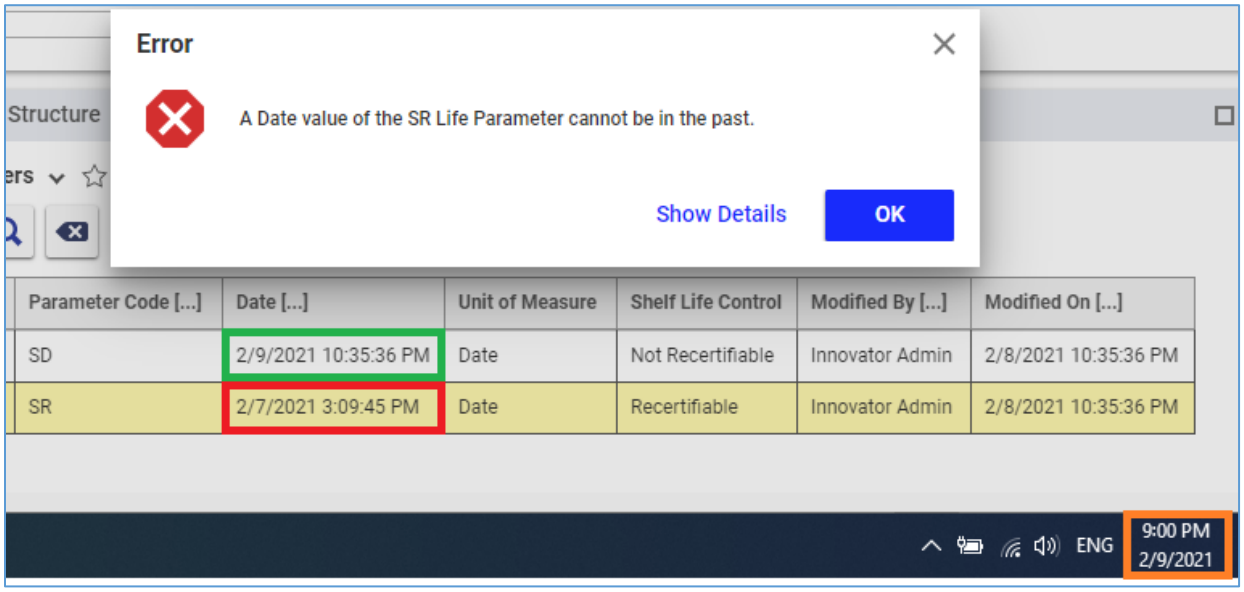


Figure 165.

- Has value modification in a **Date** cell in the **Current Date Values** Relationships tab that is **Not Recertifiable** when this **Physical Part** Item is in the **Active** State.

Note: Only an **Asset Admin** can update a **Date** cell in the **Current Date Values** Relationships tab that is **Not Recertifiable** when a given **Physical Part** Item is in the **Active** State.

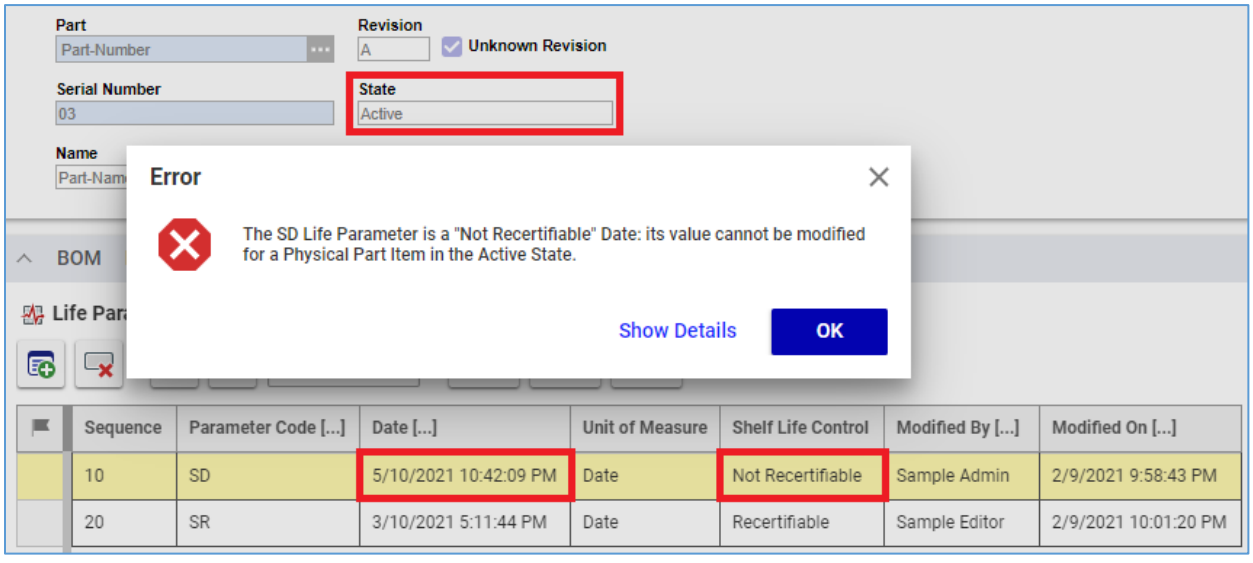


Figure 166.

5 Physical Part BOMs

A Bill of Materials (BOM) is a list of components and the quantities of each that make up an assembly. A BOM is sometimes called a part list, product structure, or associated list. In process industries, it is also known as a formula, recipe, or ingredient list.

A **Physical Part** Item can be a Digital Twin configuration of a complex product that consists of other real-world objects (components): parts, materials, software, other assemblies. If some other **Physical Part** Items are Digital Twin configurations of these components, the **Physical Part** Item can keep them in a Physical Part BOM.

We can compare the Physical Part BOM against the Part BOM as follows:

- The Physical Part BOM represents the real-world life of the product digitally in the context of its component configuration history: what is currently installed, and what had been in the past.
- The Part BOM is a blueprint of the product component configuration: what was designed to be installed.

The Physical Part BOM is a BOM of a **Physical Part** Item. This **Physical Part** Item is an assembly that has **Physical Part** Items as its components. Another industry-common term is a parent that has **Physical Part** Items as children. In technical terms, it is a **Physical Part** Item that has **Physical Part BOM** Relationship Items with other **Physical Part** Items.

A child can be a single **Physical Part** Item or another parent (assembly). The latter case is a multi-level Physical Part BOM structure.

To set up Physical Part BOM Relationships:

1. Create all necessary children, discussed in section *4.8.1 Creating Physical Parts*.
2. Create a parent.
3. Populate the parent **BOM** tab with the children, discussed in section *5.2.2.1 Adding Physical Part Items to BOMs*.

5.1 Understanding Physical Part BOM Relationship Items

A **Physical Part BOM** Relationship Item is a Relationship between a parent **Physical Part** Item and a child **Physical Part** Item defined in the **BOM** Relationships accordion tab of the parent. It represents digitally one discrete installation of a real-world component in an assembly over time.

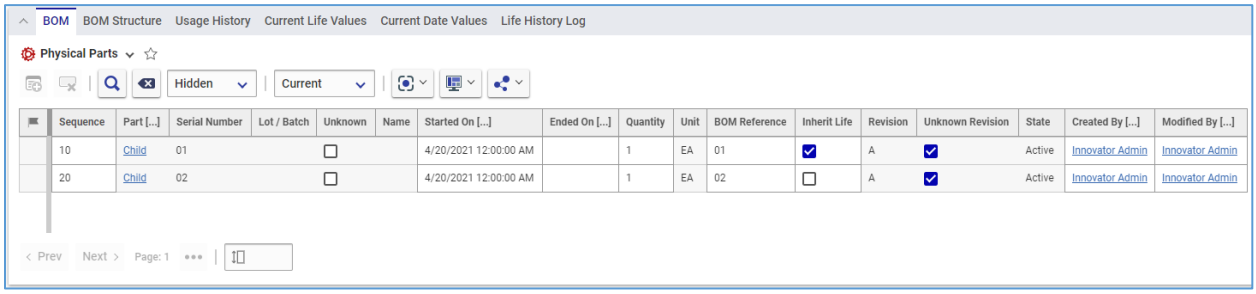


Figure 167.

The **Physical Part BOM** Relationships grid shows properties of a given **Physical Part BOM** Relationship Item and a child (related) **Physical Part** Item as follows:

- **Sequence**: a sort order number of the given **Physical Part BOM** Item in the grid.

- **Part**: a **Part Number** of a **Part** Item from which the child derives. It is also a link to this **Part** Item. This property is view-only and required.
- **Serial Number**: a serial number of the child if it is serialized. This property is view-only.
- **Lot / Batch**: a lot (batch) number of the child if it is lot-controlled. This property is view-only.
- **Unknown**: a flag indicating when selected that a value entered in the **Serial Number** or **Lot/Batch** property of the child does not match a serial or lot number that a represented asset actually bears. This property is view-only.
- **Name**: a **Name** of the **Part** Item from which the child derives. This property is view-only.
- **Started On**: a date and time value when the child has been installed at the BOM. This property is required. An entered date and time value should not be in the future.
- **Ended On**: a date and time value when the child has been removed from the BOM. An entered date and time value should not be in the future.
- **Quantity**: a quantity of the child in the BOM. This property is required.

  The default value for this property is **1**. The value should always be **1** for a serial-controlled child. It can be any numeric value greater than 0 (zero) for a lot-controlled or non-controlled child.
- **Unit**: a unit of measure for a **Part** Item from which the child derives. This property is view-only.
- **BOM Reference**: a reference designator of the **Physical Part BOM** Item. This property is required.
- **Inherit Life**: a flag that defines whether the child and its branch should inherit life variable increments from the parent when updated by an **Operational Event** Item:
  - **True**—yes, it should.
  - **False**—no, it should not.

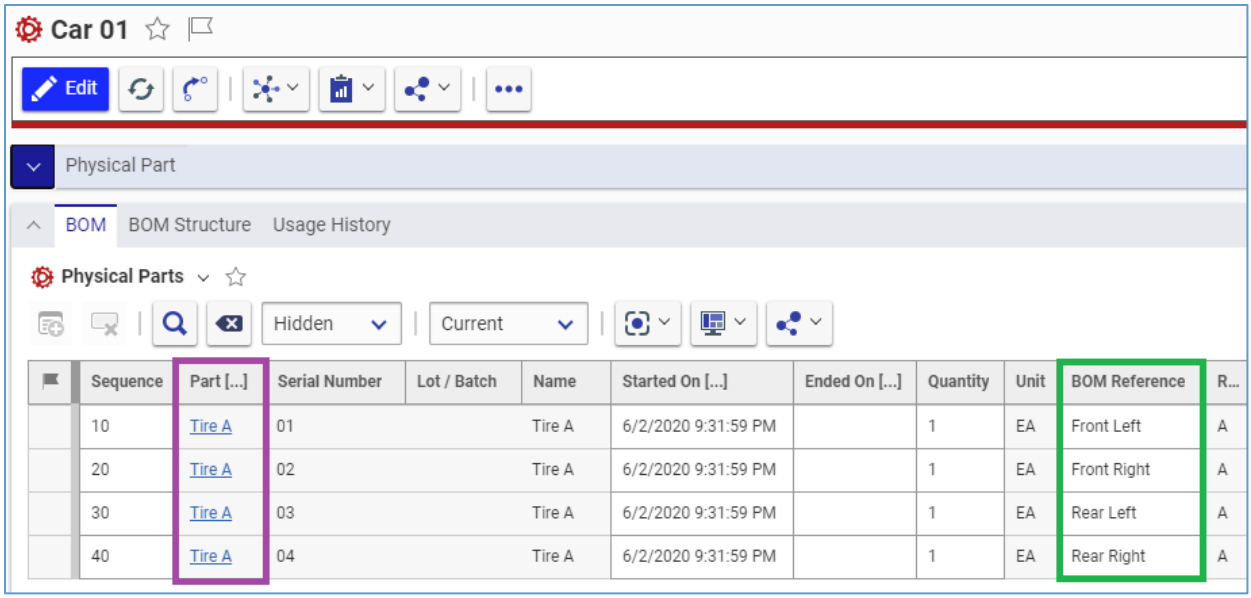
  This property is required. The default value for this property is **true**. The value should always be **false** for a lot-controlled or non-controlled child. It can be **true** or **false** for a serial-controlled child.

  This property is a tool to configure life inheritance as necessary when tracking asset life variables. For example, an aircraft engine can inherit aircraft flying hours as its own working hours because it runs during a regular aircraft flight cycle. An auxiliary power unit (APU) should not inherit aircraft flying hours as its own working hours because it should power the aircraft on the ground. Thus, a child representing an engine should have **true** in **Inherit Life**, while the one representing an APU—**false**.

  This property is used only when the life variables of the parent are automatically updated from an **Operational Event** Item; for more details, refer to section 4.7.2.3 Operational Event update of life values. It does not provide life inheritance when the life variables of the parent are updated manually, as discussed in section 4.7.2.2 Manual update of life values.
- **Revision**: a **Revision** of the **Part** Item from which the child derives. This property is view-only.
- **Unknown Revision**: a flag indicating when selected that it is unknown to which specific **Part** Item **Revision** the child belongs to. This property is view-only.
- **State**: the current **State** of the child. This property is view-only.
- **Created By**: an Identity that created the **Physical Part BOM** Item. This property is view-only.
- **Modified By**: the last Identity that modified the **Physical Part BOM** Item. This property is view-only.

A **Sequence** value is automatically set once a **Physical Part BOM** Item is added to a single-level Physical Part BOM as a number divisible by ten. It is highly recommended to stick to this default logic and not to modify the **Sequence** values. For details, see section 3.6 Sequence Values in the DTC Relationships Grids. One Physical Part BOM can have several **Physical Part BOM** Items with the same **Sequence** value.

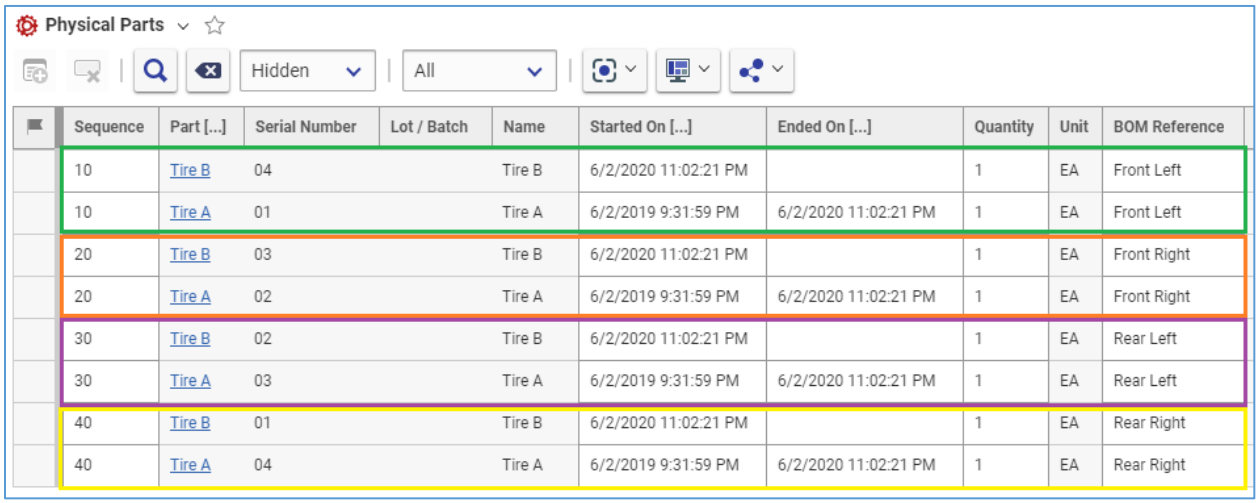
A **BOM Reference** value is manually set as a text up to 50 characters long. It should distinguish a given **Physical Part BOM** Item against other **Physical Part BOM** Items in a given Physical Part BOM. A good practice is to provide the **BOM Reference** property with data on the represented component location in the assembly, like a reference number or description from an illustrated part catalog. For example, a car has four identical tires, and **BOM Reference** can identify each location separately. One **Physical Part** BOM can have several **Physical Part BOM** Items with the same **BOM Reference** value.



Sequence	Part [...]	Serial Number	Lot / Batch	Name	Started On [...]	Ended On [...]	Quantity	Unit	BOM Reference	R...
10	Tire A	01		Tire A	6/2/2020 9:31:59 PM		1	EA	Front Left	A
20	Tire A	02		Tire A	6/2/2020 9:31:59 PM		1	EA	Front Right	A
30	Tire A	03		Tire A	6/2/2020 9:31:59 PM		1	EA	Rear Left	A
40	Tire A	04		Tire A	6/2/2020 9:31:59 PM		1	EA	Rear Right	A

Figure 168.

A **Sequence**, **BOM Reference**, and **Inherit Life** value combination defines a single logical or physical BOM location for a specific compatible child installation. If several **Physical Part BOM** Items have the same **Sequence** value, each of them should have the same **BOM Reference** and **Inherit Life** values. A group of **Physical Part BOM** Items with the same **Sequence**–**BOM Reference**– **Inherit Life** combination denotes a series of similar installations at the given BOM location over parent life. For example, if the car has its four tires changed twice, the **Car** BOM should represent these installations with eight **Physical Part BOM** Items as four groups of two Items with the same **Sequence**–**BOM Reference**– **Inherit Life** combination.



Sequence	Part [...]	Serial Number	Lot / Batch	Name	Started On [...]	Ended On [...]	Quantity	Unit	BOM Reference
10	Tire B	04		Tire B	6/2/2020 11:02:21 PM		1	EA	Front Left
10	Tire A	01		Tire A	6/2/2019 9:31:59 PM	6/2/2020 11:02:21 PM	1	EA	Front Left
20	Tire B	03		Tire B	6/2/2020 11:02:21 PM		1	EA	Front Right
20	Tire A	02		Tire A	6/2/2019 9:31:59 PM	6/2/2020 11:02:21 PM	1	EA	Front Right
30	Tire B	02		Tire B	6/2/2020 11:02:21 PM		1	EA	Rear Left
30	Tire A	03		Tire A	6/2/2019 9:31:59 PM	6/2/2020 11:02:21 PM	1	EA	Rear Left
40	Tire B	01		Tire B	6/2/2020 11:02:21 PM		1	EA	Rear Right
40	Tire A	04		Tire A	6/2/2019 9:31:59 PM	6/2/2020 11:02:21 PM	1	EA	Rear Right

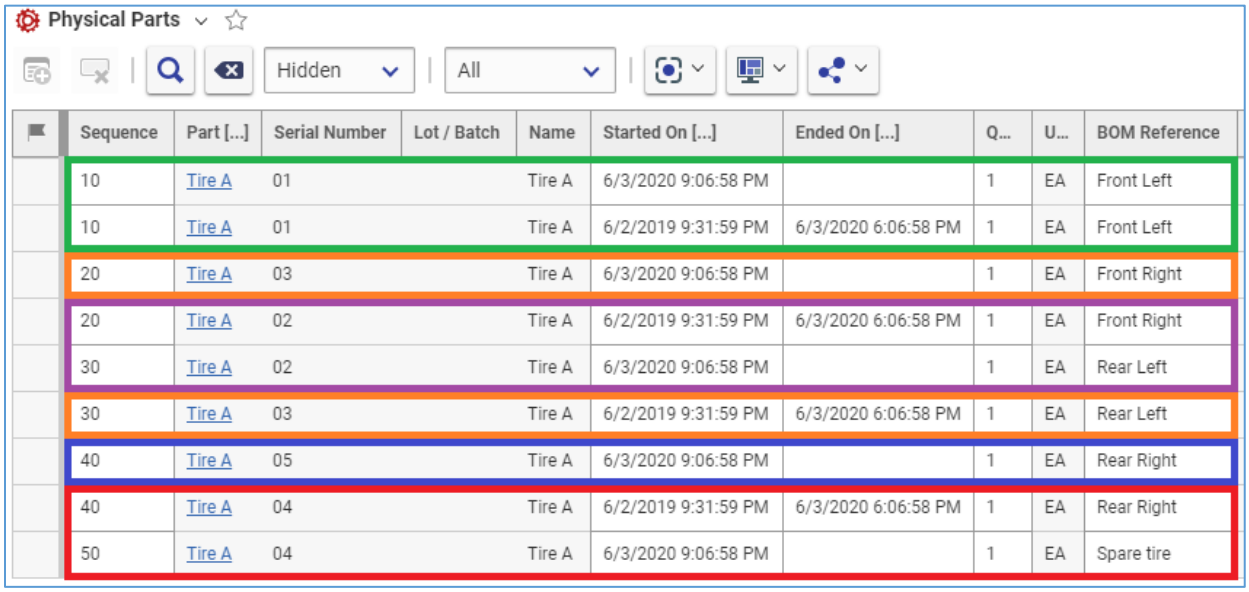
Figure 169.

If the **Ended On** property is blank, a given **Physical Part BOM** Item is *current*: a represented component is currently installed in the assembly. Once the **Ended On** property gets a value, the **Physical Part BOM** Item becomes *historical*: the component has been removed from the assembly at the given time and is not currently installed there.

The **Ended On** property should not be in the future because this property is retrospective: it states that a component was in an assembly until that time point. If an **Ended On** value is later than the current system date and time, a **Physical Part** Item cannot be saved, and an error is raised when saving this **Physical Part** Item; see section 5.2.2.4 Physical Part BOM validation.

A **Physical Part** Item can be a child in the same BOM more than once over a parent life. When the represented component is removed for inspection, maintenance, or repair, the representing **Physical Part BOM** Item should become historical: its **Ended On** property is a removal timestamp. When the component is reinstalled, once the work on it is done, a new current representing **Physical Part BOM** Item should be created. The new Item **Started On** property is a reinstallation timestamp. The new Item **Started On** date and time should be the same or later as the old Item **Ended On** date and time. Depending on the assembly location where the component is reinstalled, the new **Physical Part BOM** Item should have its **Sequence** and **BOM Reference** properties set as follows:

- If the same location—the same values as for the previous representing **Physical Part BOM** Item. For example, a flat car tire was removed, repaired, and put back.
- If other existing location—values different from the previous representing **Physical Part BOM** Item but the same as for a **Physical Part BOM** Item that had represented this other location. For example, two tires were swapped because of their wear.
- If other new location—new values different from the existing **Physical Part BOM** Items. For example, an old tire became a spare tire.



Sequence	Part [...]	Serial Number	Lot / Batch	Name	Started On [...]	Ended On [...]	Q...	U...	BOM Reference
10	Tire A	01		Tire A	6/3/2020 9:06:58 PM		1	EA	Front Left
10	Tire A	01		Tire A	6/2/2019 9:31:59 PM	6/3/2020 6:06:58 PM	1	EA	Front Left
20	Tire A	03		Tire A	6/3/2020 9:06:58 PM		1	EA	Front Right
20	Tire A	02		Tire A	6/2/2019 9:31:59 PM	6/3/2020 6:06:58 PM	1	EA	Front Right
30	Tire A	02		Tire A	6/3/2020 9:06:58 PM		1	EA	Rear Left
30	Tire A	03		Tire A	6/2/2019 9:31:59 PM	6/3/2020 6:06:58 PM	1	EA	Rear Left
40	Tire A	05		Tire A	6/3/2020 9:06:58 PM		1	EA	Rear Right
40	Tire A	04		Tire A	6/2/2019 9:31:59 PM	6/3/2020 6:06:58 PM	1	EA	Rear Right
50	Tire A	04		Tire A	6/3/2020 9:06:58 PM		1	EA	Spare tire

Figure 170.

A **Physical Part** Item can be a child in more than one Physical Part BOMs over its life. The represented component is removed from one assembly and installed into another.

The Remove-and-Replace operation digitally represents a situation when a component was removed from an assembly for inspection, repair, maintenance, or replacement and was either reinstalled back or replaced by another component at its place in the assembly. For details, see section 5.2.3 Removing-and-replacing Physical Parts in BOMs.

The Control Type of a **Physical Part** Item defines in how many Physical Part BOM slots it can be a child (related Item) at the same time:

- A *serial-controlled* **Physical Part** Item can be a child *in only one* Physical Part BOM slot at the same time: the time periods of its BOM installations must be different and not overlap. If trying to save a parent **Physical Part** Item with a *serial-controlled* **Physical Part** Item that is already a child somewhere at a given time point or period, the server will not allow to save this parent and raise an error message; see section 5.2.2.4 Physical Part BOM validation.
- A *lot-controlled* or *non-controlled* **Physical Part** Item can be a child *in more than one* Physical Part BOM slots at the same time: the time periods of its BOM installations can be the same, different, and overlap.

The Control Type of a parent **Physical Part** Item also defines **Physical Part** Items of which Control Type can be its children:

- Serial-controlled:
  - Serial-controlled
  - Lot-controlled
  - Non-controlled
- Lot-controlled:
  - Lot-controlled
  - Non-controlled
- Non-controlled:
  - Non-controlled

If trying to save a parent **Physical Part** Item with a child **Physical Part** Item of an unallowed Control Type, the server will not allow you to save this parent and raise an error message; see section 5.2.2.4 Physical Part BOM validation.

5.2 Single-level Physical Part BOMs

A single-level Physical Part BOM is all the immediate children of a parent **Physical Part** Item. A single-level Physical Part BOM is displayed as the **Physical Parts** Grid on the **BOM** accordion tab of a parent **Physical Part** Item view.

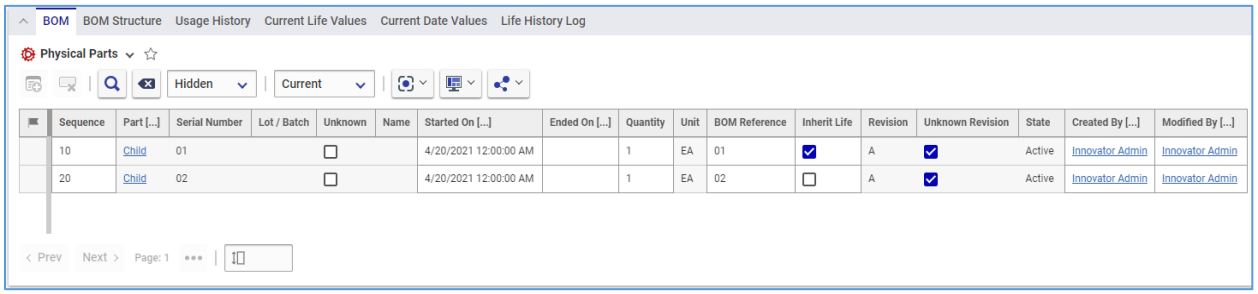


Figure 171.

The **BOM** accordion tab has the **Physical Parts** Grid toolbar to manage and filter the **Physical Parts** Grid. The default toolbar content is the following:

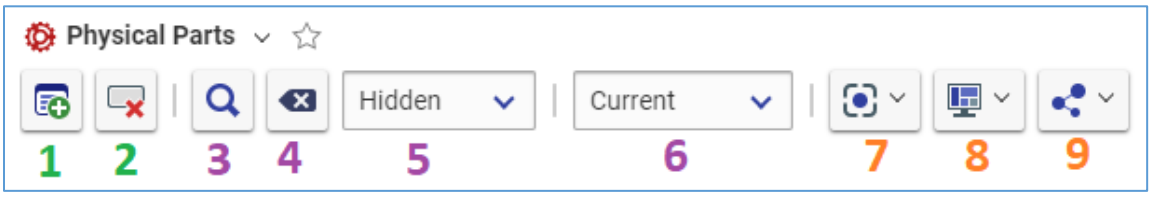


Figure 172.

1. **Add Physical Parts**: a button to create **Physical Part BOM** Items; discussed in section *5.2.2.1 Adding Physical Part Items to BOMs*.
2. **Delete Physical BOM Line**: a button to delete **Physical Part BOM** Items; discussed in section *5.2.2.4 Removing Physical Parts from BOMs*.
3. **Run Search**: a button to apply a search filter with given criteria to the **Physical Part** BOM.
4. **Clear Search Criteria**: a button to clear filter from the last given search criteria. As a rule of thumb, use it when switching between different Search Modes.
5. **Search Modes**: a drop-down list of Search Modes.

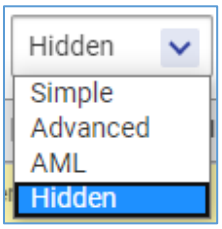


Figure 173.

6. **When Active**: a drop-down list of filter options for the **Simple** and **Hidden** Search Modes; discussed in section *5.2.1 Exploring Single-level Physical Part BOMs*.

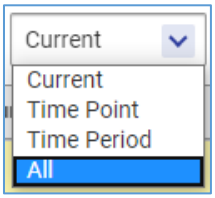


Figure 174.

7. **Refine**: a pop-up view to set up the Grid columns.
8. **Display**: a menu of options that determine the level of details to display when you select an Item.
9. **Share**: a menu of **Physical Part** BOM export options.

The **Physical Parts** Grid toolbar is dynamic: its content depends on a given **Search Mode** and **When Active** filter option.

This document discusses only features for managing (1-2) and exploring (3-6) **Physical Part BOM** Items.

5.2.1 Exploring single-level Physical Part BOMs

When a **Physical Part** Item view is first opened, the **BOM** accordion tab is shown in the **Hidden** Search Mode with only current **Physical Part BOM** Items.

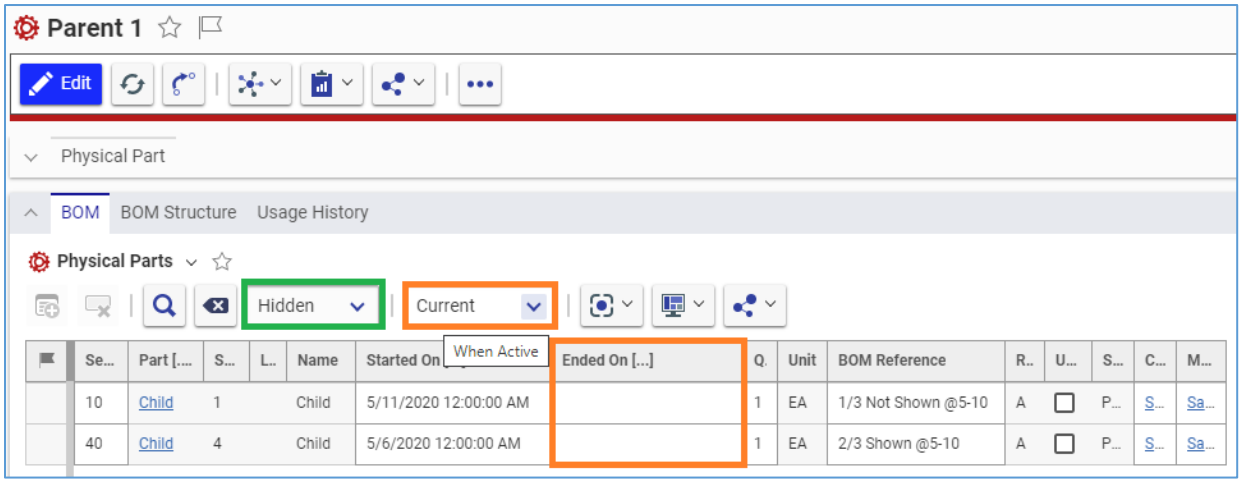


Figure 175.

The default **BOM** Grid sorting is first by the **Sequence** values in ascending order (from the lowest to the highest number) within all the Items and then by the **Started On** values in descending order (from the last to the first date) within the same **Sequence** value group.

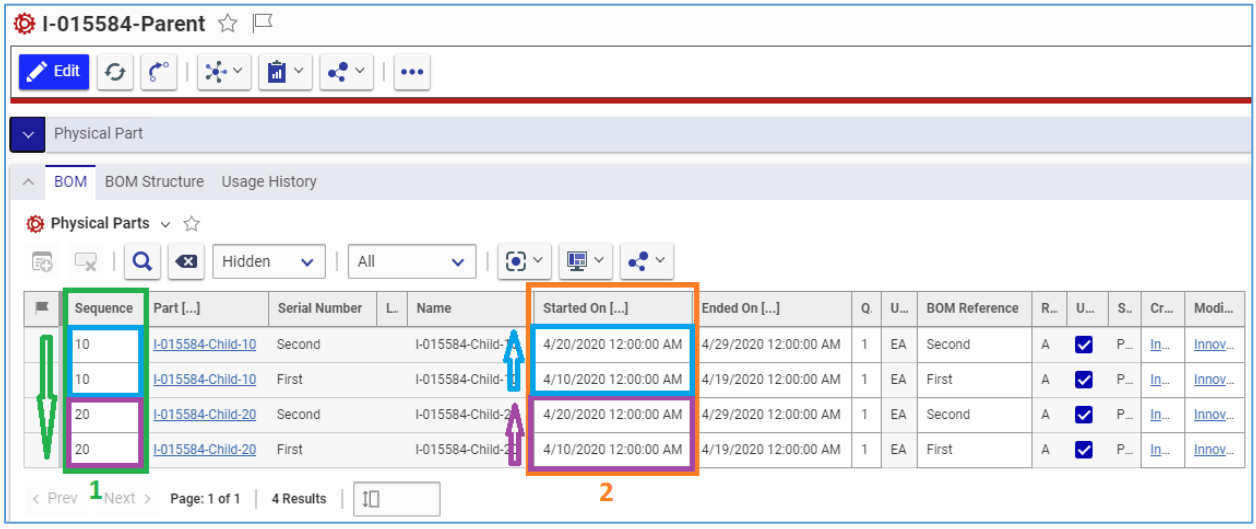


Figure 176.

5.2.1.1 Physical Part BOM When Active filter

The **Physical Parts BOM** Relationships grid toolbar features the **When Active** drop-down list to filter the **Physical Part BOM** Items against time constraints:

- **Current**: shows only current **Physical Part BOM** Items.
- **Time Point**: shows **Physical Part BOM** Items filtered against a given time point as discussed in section *5.2.1.2 Exploring Single-level Physical Part BOMs at a time point*.
- **Time Period**: shows **Physical Part BOM** Items filtered against a given time period as discussed in section *5.2.1.3 Exploring Single-level Physical Part BOMs over a time period*.

- **All**: shows all **Physical Part BOM** Items.

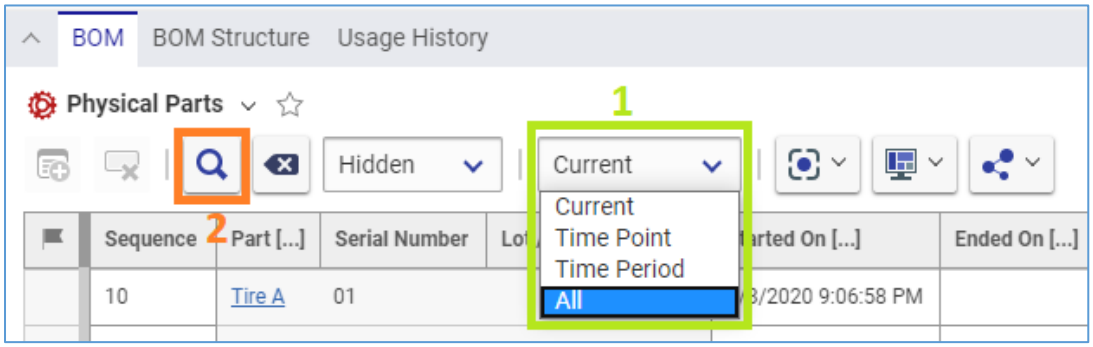


Figure 177.

To apply a **When Active** filter:

1. Select the filter in the **When Active** drop-down list.
2. Click the **Run Search** button.

The **When Active** drop-down filter list is available only for the **Simple** and **Hidden** Search Modes.

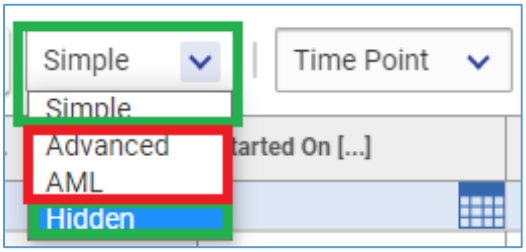


Figure 178.

In the **Advanced** and **AML** Search Modes, use their advanced filtering capabilities.

Note: As a rule of thumb, clear search criteria when switching between different Search Modes.

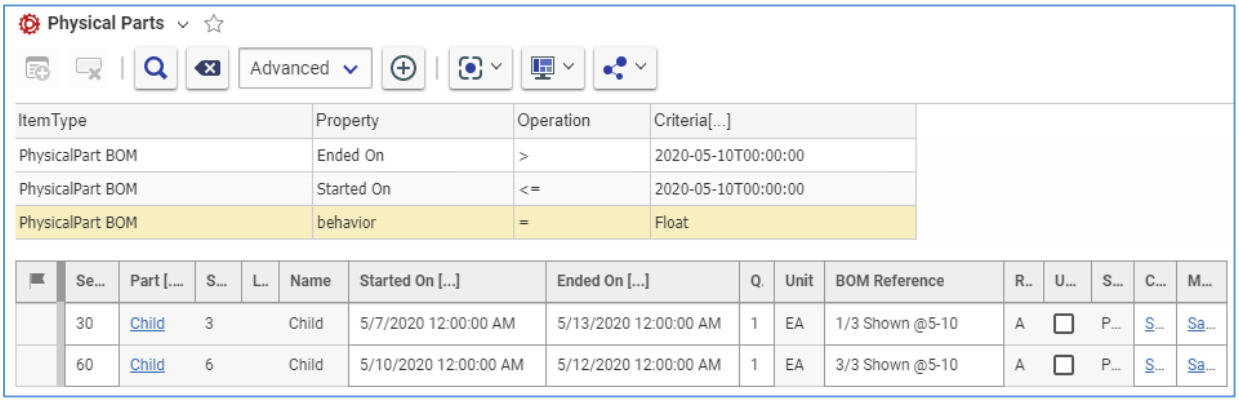
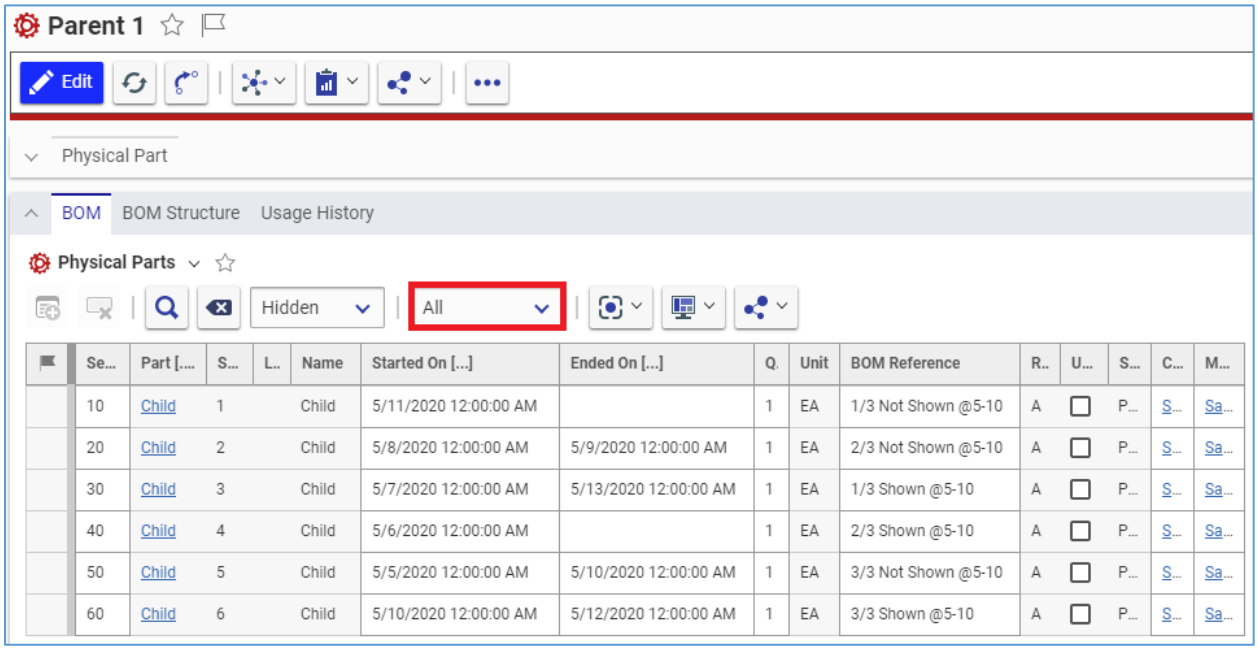


Figure 179.

5.2.1.2 Exploring single-level Physical Part BOMs at a time point

Apply the **Time Point** filter of the **When Active** drop-down list to filter **Physical Part BOM** Items at a given point of time.



Se...	Part [....	S...	L..	Name	Started On [...]	Ended On [...]	Q.	Unit	BOM Reference	R..	U...	S...	C...	M...
10	Child	1		Child	5/11/2020 12:00:00 AM		1	EA	1/3 Not Shown @5-10	A		P...	S...	Sa...
20	Child	2		Child	5/8/2020 12:00:00 AM	5/9/2020 12:00:00 AM	1	EA	2/3 Not Shown @5-10	A		P...	S...	Sa...
30	Child	3		Child	5/7/2020 12:00:00 AM	5/13/2020 12:00:00 AM	1	EA	1/3 Shown @5-10	A		P...	S...	Sa...
40	Child	4		Child	5/6/2020 12:00:00 AM		1	EA	2/3 Shown @5-10	A		P...	S...	Sa...
50	Child	5		Child	5/5/2020 12:00:00 AM	5/10/2020 12:00:00 AM	1	EA	3/3 Not Shown @5-10	A		P...	S...	Sa...
60	Child	6		Child	5/10/2020 12:00:00 AM	5/12/2020 12:00:00 AM	1	EA	3/3 Shown @5-10	A		P...	S...	Sa...

Figure 180.

The time point is a date and time value given in the **Active On** field. This field is displayed next to the **When Active** drop-down list when the **Time Point** filter is specified in the list.

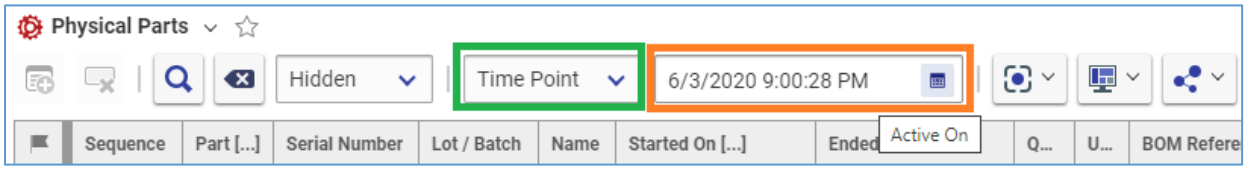


Figure 181.

To specify the time point, provide date and time value in the **Active On** fields using the **Date dialog**.

Once the **Active On** time point is given, and the **Time Point** filter is applied, the **Physical Part BOM** Grid displays only **Physical Part BOM** Items that are current at the given time point.

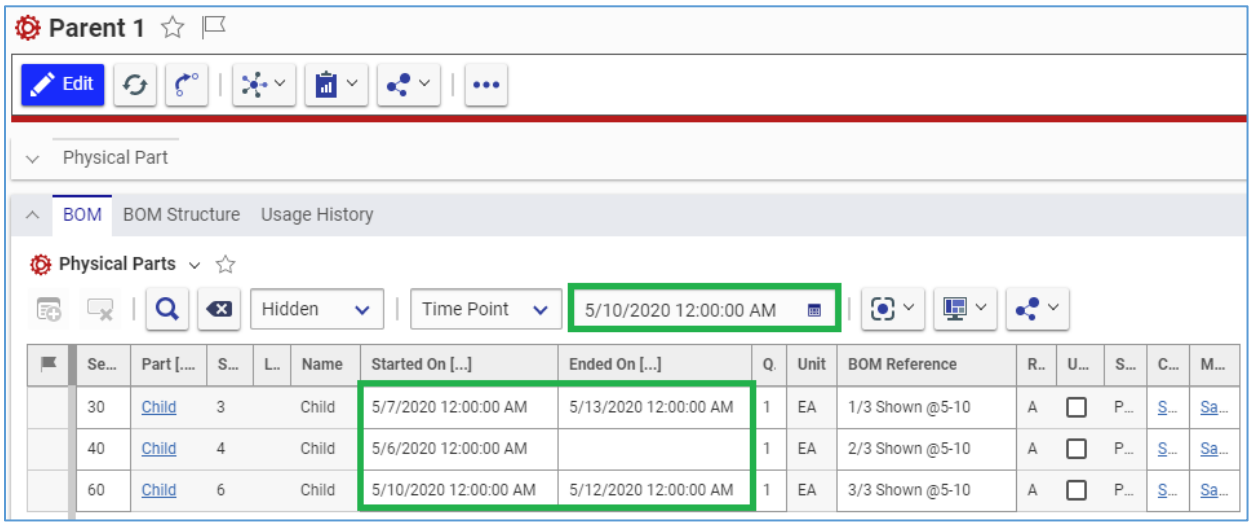


Figure 182.

To be evaluated as current at a particular time point, a **Physical Part BOM** Item must set its property values as follows:

- **Started On** less than or equal to the time point.

  AND

- **Ended On** greater than the time point or blank.

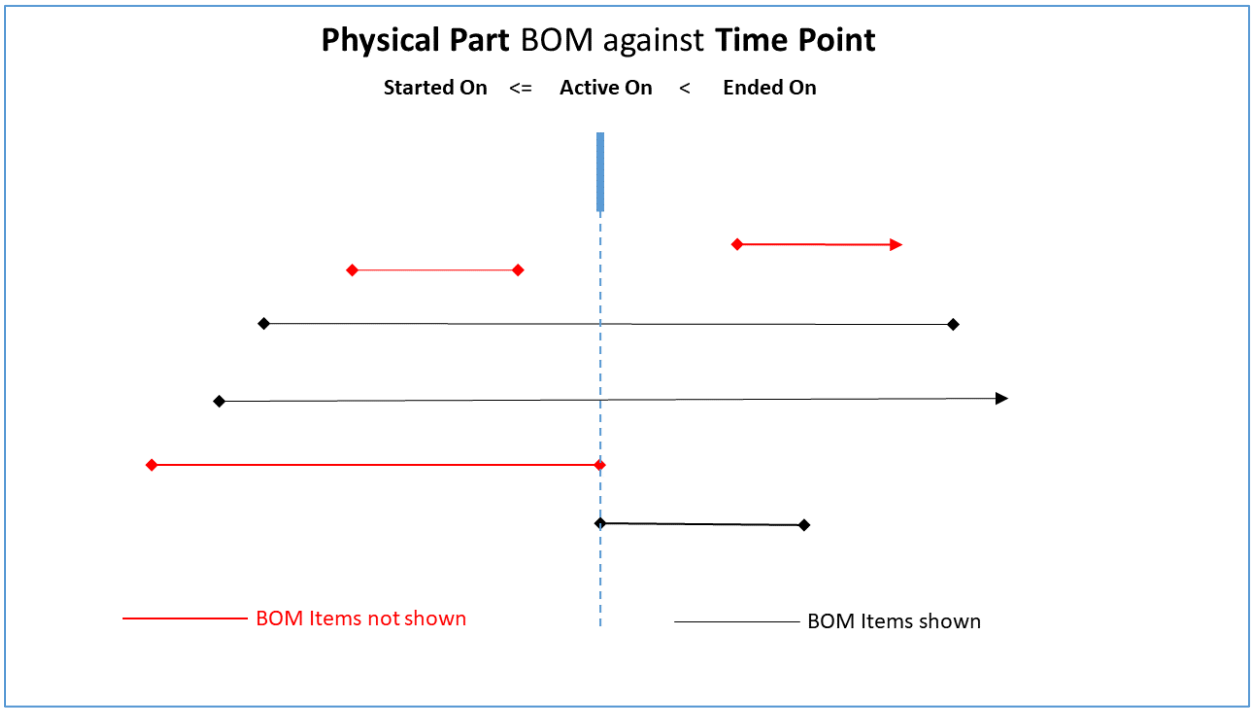
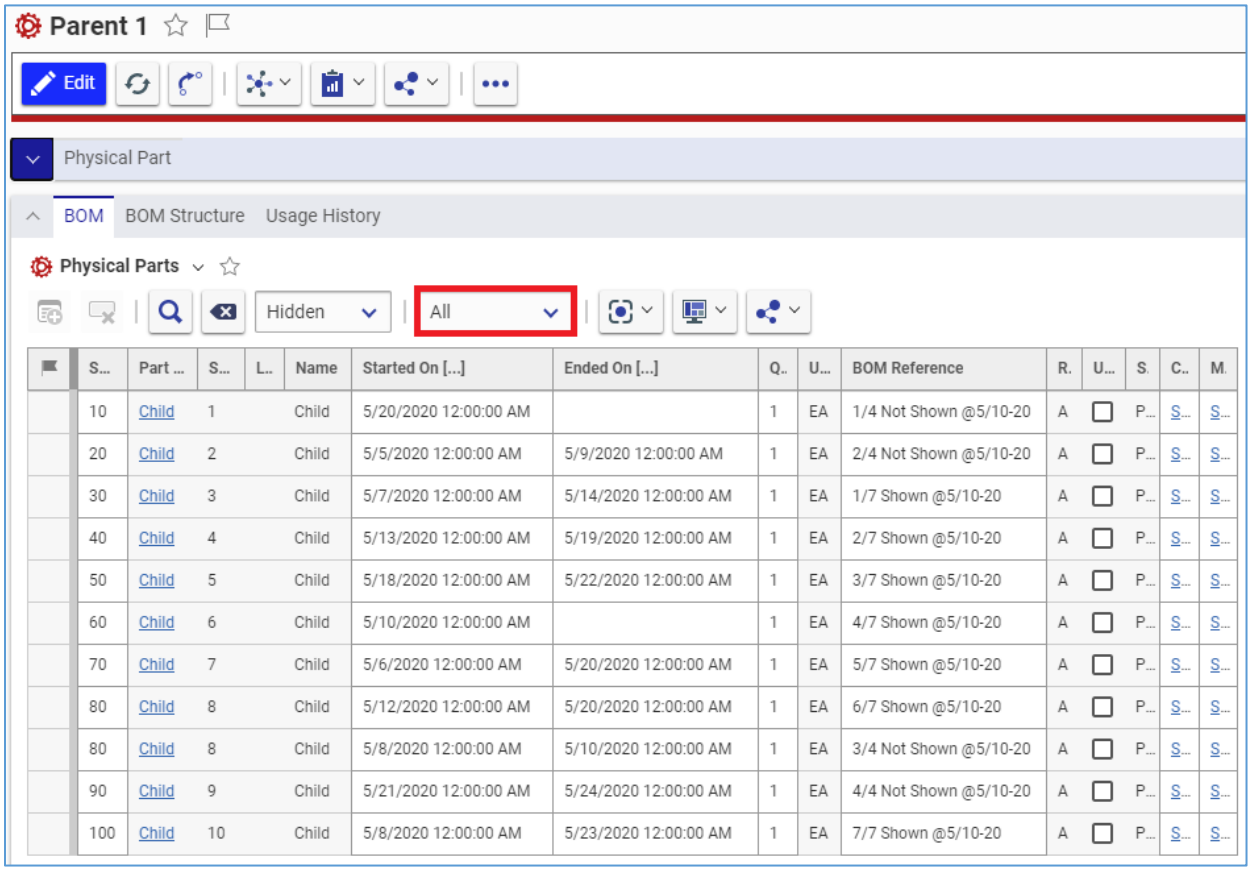


Figure 183.

5.2.1.3 Exploring single-level Physical Part BOMs over a time period

Apply the **Time Period** filter of the **When Active** drop-down list to filter **Physical Part BOM** Items over a given period of time.



S...	Part ...	S...	L...	Name	Started On [...]	Ended On [...]	Q..	U...	BOM Reference	R.	U...	S.	C..	M.
10	Child	1		Child	5/20/2020 12:00:00 AM		1	EA	1/4 Not Shown @5/10-20	A		P...	S...	S...
20	Child	2		Child	5/5/2020 12:00:00 AM	5/9/2020 12:00:00 AM	1	EA	2/4 Not Shown @5/10-20	A		P...	S...	S...
30	Child	3		Child	5/7/2020 12:00:00 AM	5/14/2020 12:00:00 AM	1	EA	1/7 Shown @5/10-20	A		P...	S...	S...
40	Child	4		Child	5/13/2020 12:00:00 AM	5/19/2020 12:00:00 AM	1	EA	2/7 Shown @5/10-20	A		P...	S...	S...
50	Child	5		Child	5/18/2020 12:00:00 AM	5/22/2020 12:00:00 AM	1	EA	3/7 Shown @5/10-20	A		P...	S...	S...
60	Child	6		Child	5/10/2020 12:00:00 AM		1	EA	4/7 Shown @5/10-20	A		P...	S...	S...
70	Child	7		Child	5/6/2020 12:00:00 AM	5/20/2020 12:00:00 AM	1	EA	5/7 Shown @5/10-20	A		P...	S...	S...
80	Child	8		Child	5/12/2020 12:00:00 AM	5/20/2020 12:00:00 AM	1	EA	6/7 Shown @5/10-20	A		P...	S...	S...
80	Child	8		Child	5/8/2020 12:00:00 AM	5/10/2020 12:00:00 AM	1	EA	3/4 Not Shown @5/10-20	A		P...	S...	S...
90	Child	9		Child	5/21/2020 12:00:00 AM	5/24/2020 12:00:00 AM	1	EA	4/4 Not Shown @5/10-20	A		P...	S...	S...
100	Child	10		Child	5/8/2020 12:00:00 AM	5/23/2020 12:00:00 AM	1	EA	7/7 Shown @5/10-20	A		P...	S...	S...

Figure 184.

The time period is defined by date and time values given in the **Period Start** and **Period End** fields. These fields are displayed next to the **When Active** drop-down list when the **Time Period** filter is specified in the list.

Figure 185.

To specify the time period, provide valid date and time values in the **Period Start** and **Period End** fields using the **Date dialog**.

To be valid date and time values, **Period Start** must be before **Period End**. Otherwise, an error is raised.

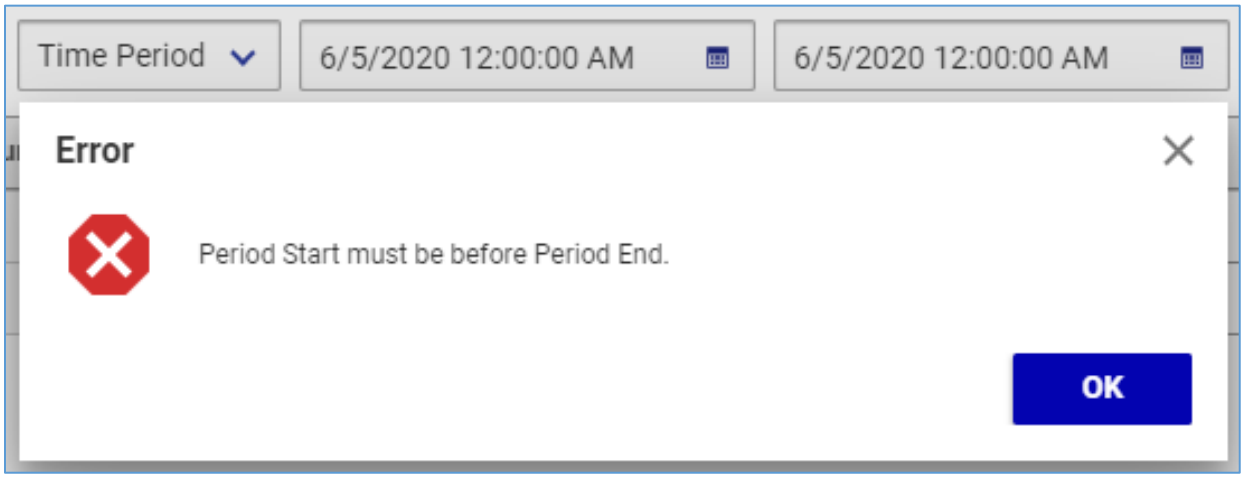
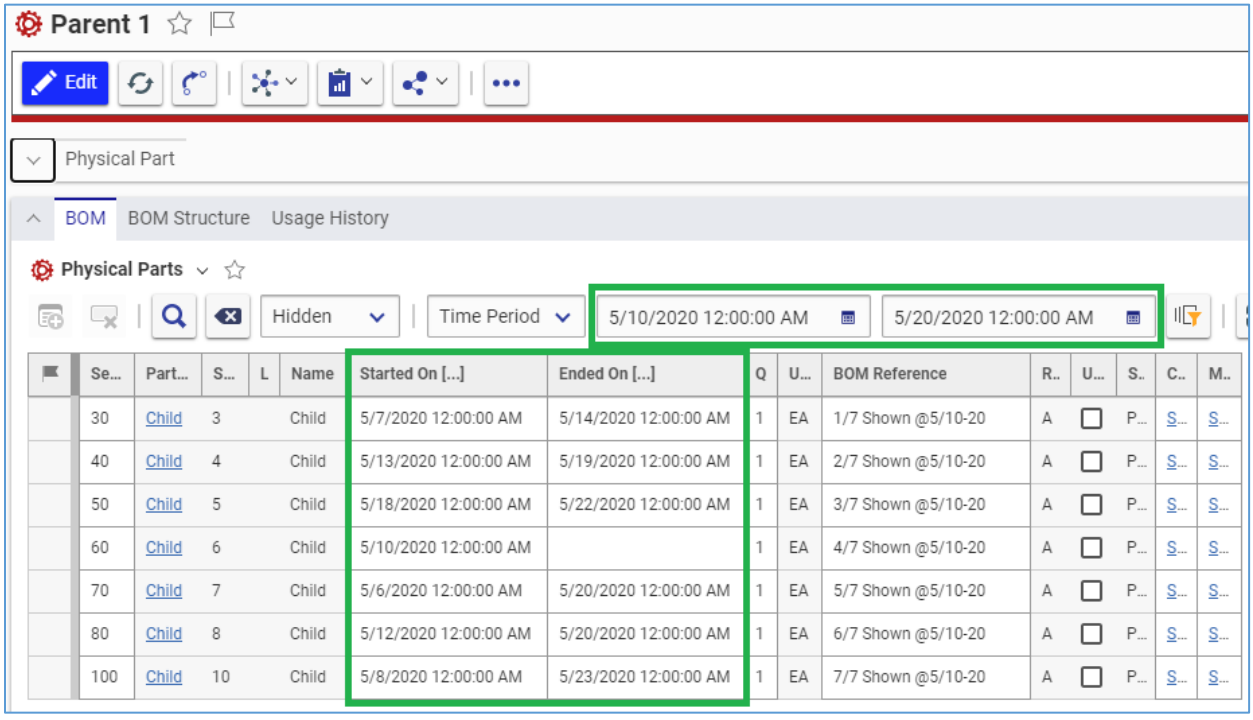


Figure 186.

Once the time period is given, and the **Time Period** filter is applied, the **Physical Part BOM** Grid displays only **Physical Part BOM** Items that are current during the given time period.



Se...	Part...	S...	L	Name	Started On [...]	Ended On [...]	Q	U...	BOM Reference	R..	U...	S..	C..	M..
30	Child	3		Child	5/7/2020 12:00:00 AM	5/14/2020 12:00:00 AM	1	EA	1/7 Shown @5/10-20	A		P...	S...	S...
40	Child	4		Child	5/13/2020 12:00:00 AM	5/19/2020 12:00:00 AM	1	EA	2/7 Shown @5/10-20	A		P...	S...	S...
50	Child	5		Child	5/18/2020 12:00:00 AM	5/22/2020 12:00:00 AM	1	EA	3/7 Shown @5/10-20	A		P...	S...	S...
60	Child	6		Child	5/10/2020 12:00:00 AM		1	EA	4/7 Shown @5/10-20	A		P...	S...	S...
70	Child	7		Child	5/6/2020 12:00:00 AM	5/20/2020 12:00:00 AM	1	EA	5/7 Shown @5/10-20	A		P...	S...	S...
80	Child	8		Child	5/12/2020 12:00:00 AM	5/20/2020 12:00:00 AM	1	EA	6/7 Shown @5/10-20	A		P...	S...	S...
100	Child	10		Child	5/8/2020 12:00:00 AM	5/23/2020 12:00:00 AM	1	EA	7/7 Shown @5/10-20	A		P...	S...	S...

Figure 187.

To be evaluated as current over a time period, a **Physical Part BOM** Item must have its property values as follows:

- **Started On** less than **Period End**.

AND

- **Ended On** greater than **Period Start** or blank.

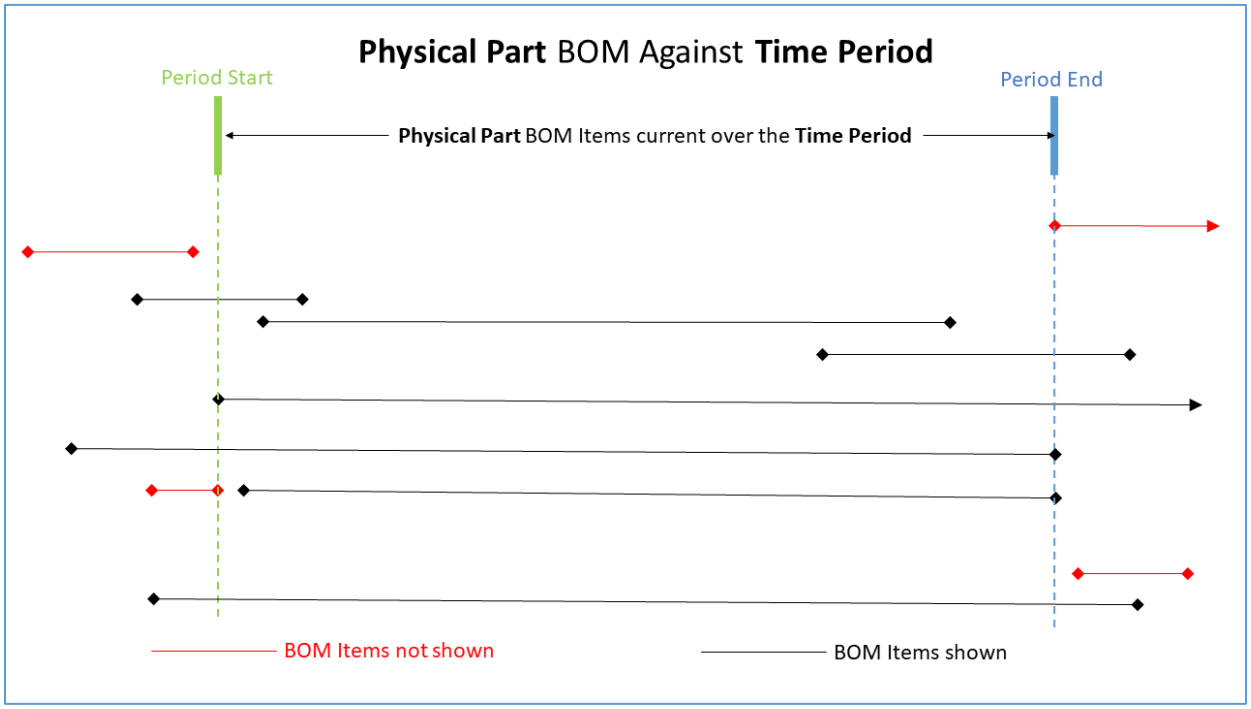
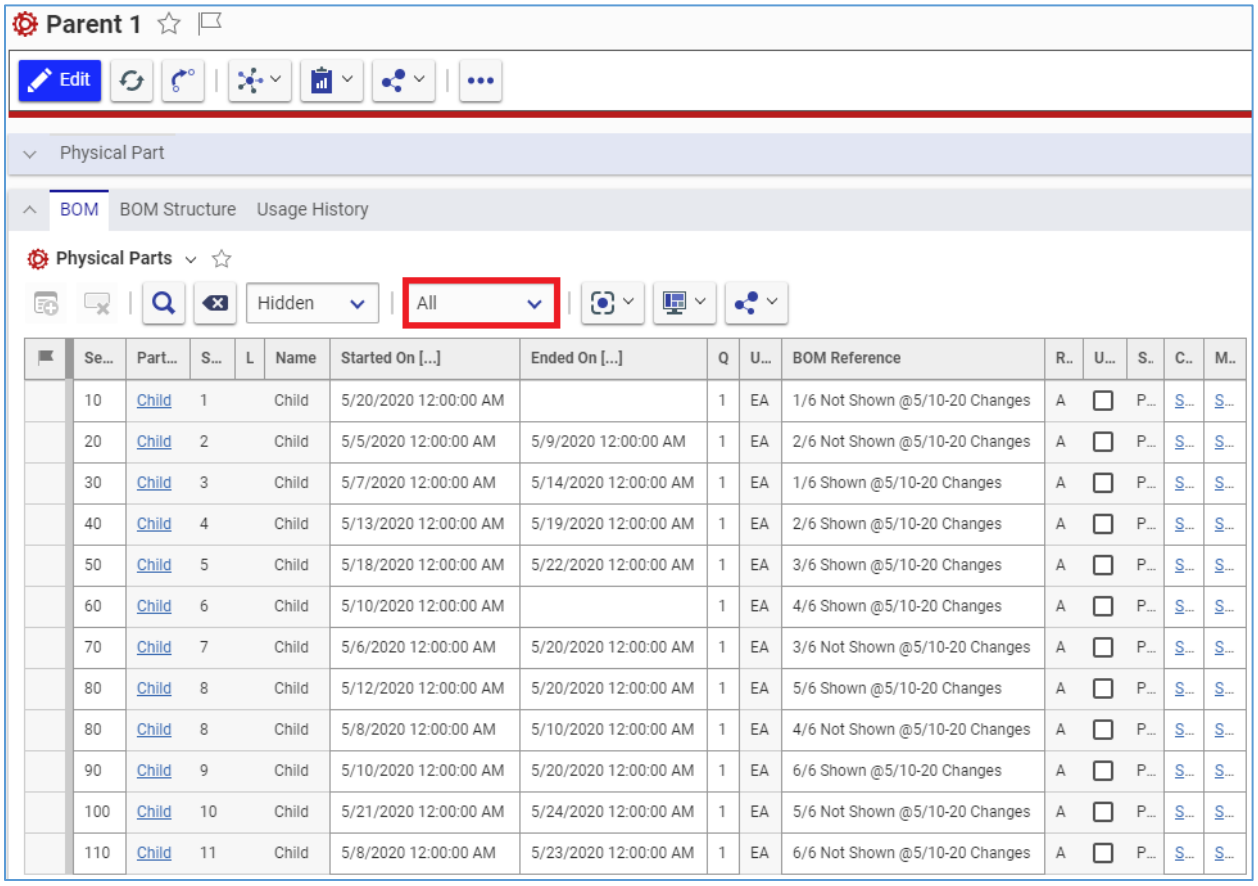


Figure 188.

5.2.1.4 Exploring single-level Physical Part BOM changes over a time period

A single-level Physical Part BOM structure can be changed over time: some of its **Physical Part BOM** Items are modified to be historical, or new **Physical Part BOM** Items are added.



Se...	Part...	S...	L	Name	Started On [...]	Ended On [...]	Q	U...	BOM Reference	R..	U...	S..	C..	M..
10	Child	1		Child	5/20/2020 12:00:00 AM		1	EA	1/6 Not Shown @5/10-20 Changes	A		P...	S...	S...
20	Child	2		Child	5/5/2020 12:00:00 AM	5/9/2020 12:00:00 AM	1	EA	2/6 Not Shown @5/10-20 Changes	A		P...	S...	S...
30	Child	3		Child	5/7/2020 12:00:00 AM	5/14/2020 12:00:00 AM	1	EA	1/6 Shown @5/10-20 Changes	A		P...	S...	S...
40	Child	4		Child	5/13/2020 12:00:00 AM	5/19/2020 12:00:00 AM	1	EA	2/6 Shown @5/10-20 Changes	A		P...	S...	S...
50	Child	5		Child	5/18/2020 12:00:00 AM	5/22/2020 12:00:00 AM	1	EA	3/6 Shown @5/10-20 Changes	A		P...	S...	S...
60	Child	6		Child	5/10/2020 12:00:00 AM		1	EA	4/6 Shown @5/10-20 Changes	A		P...	S...	S...
70	Child	7		Child	5/6/2020 12:00:00 AM	5/20/2020 12:00:00 AM	1	EA	3/6 Not Shown @5/10-20 Changes	A		P...	S...	S...
80	Child	8		Child	5/12/2020 12:00:00 AM	5/20/2020 12:00:00 AM	1	EA	5/6 Shown @5/10-20 Changes	A		P...	S...	S...
80	Child	8		Child	5/8/2020 12:00:00 AM	5/10/2020 12:00:00 AM	1	EA	4/6 Not Shown @5/10-20 Changes	A		P...	S...	S...
90	Child	9		Child	5/10/2020 12:00:00 AM	5/20/2020 12:00:00 AM	1	EA	6/6 Shown @5/10-20 Changes	A		P...	S...	S...
100	Child	10		Child	5/21/2020 12:00:00 AM	5/24/2020 12:00:00 AM	1	EA	5/6 Not Shown @5/10-20 Changes	A		P...	S...	S...
110	Child	11		Child	5/8/2020 12:00:00 AM	5/23/2020 12:00:00 AM	1	EA	6/6 Not Shown @5/10-20 Changes	A		P...	S...	S...

Figure 189.

By default, viewing a single-level Physical Part BOM structure over a given period of time with the **Time Period** filter shows **Physical Part BOM** Items that are current at the given period.

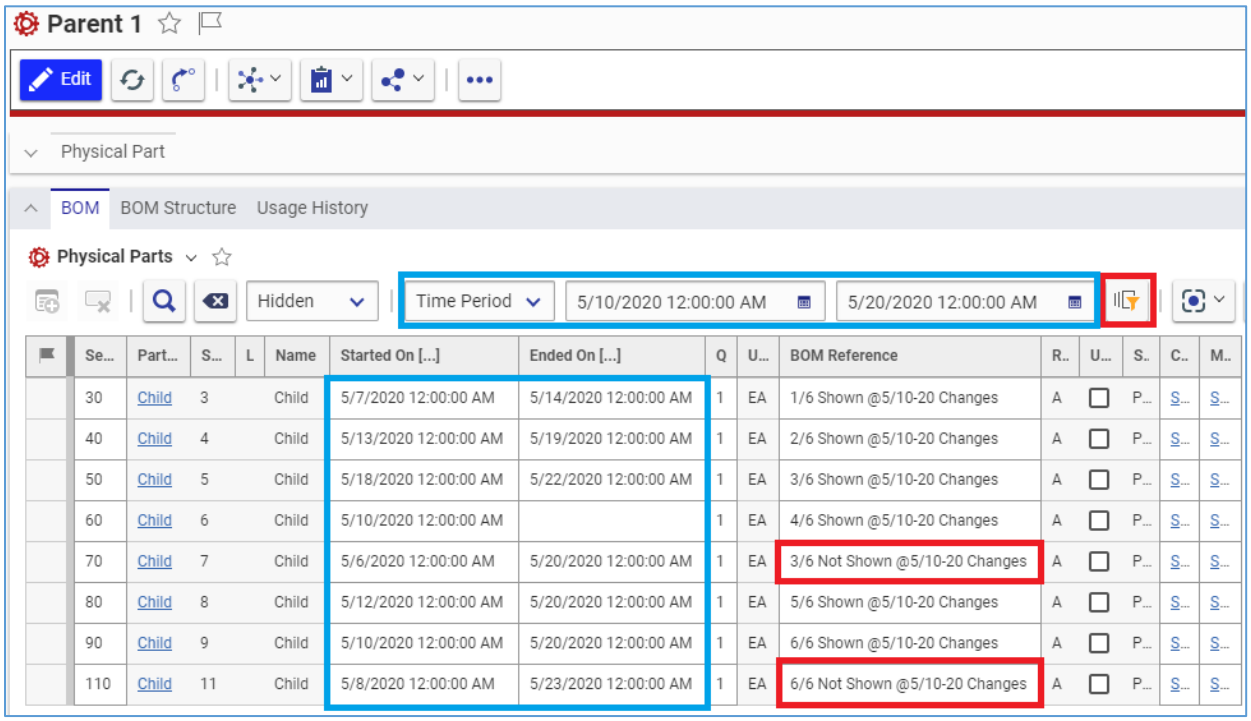


Figure 190.

The **Time Period** filter option of the **When Active** drop-down list features the **Show Change Only** button. This button is displayed next to the **Period End** field when the **Time Period** filter is specified in the list. It is off by default: its background is light grey.

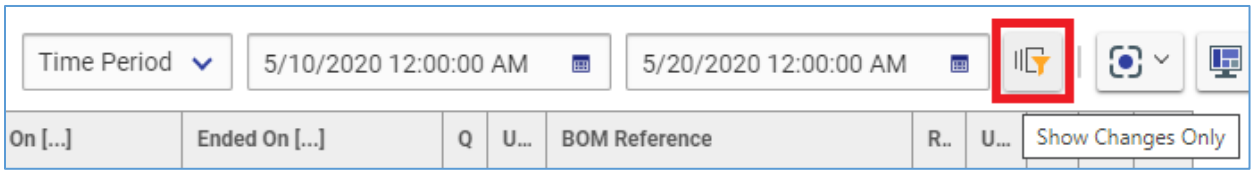


Figure 191.

When the **Show Change Only** button is on, its background is light blue.

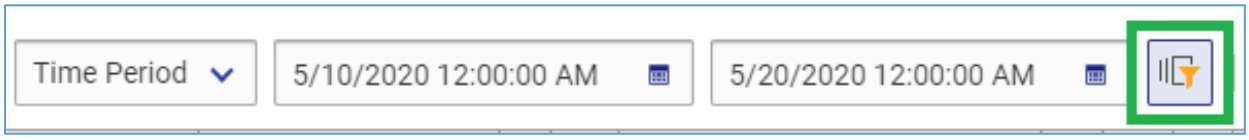
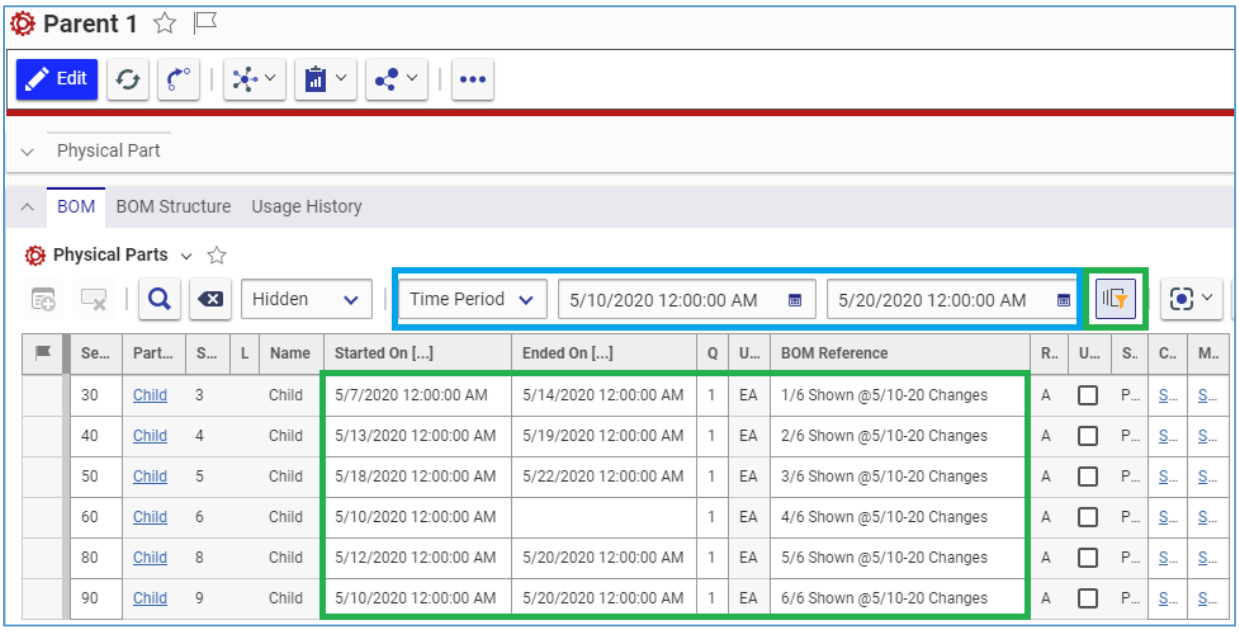


Figure 192.

Click the button to switch it off or on.

The **Physical Part BOM** Grid shows only **Physical Part** Items that are created or end during a given time period if the **Time Period** filter is applied with the **Show Change Only** button on.



Se...	Part...	S...	L	Name	Started On [...]	Ended On [...]	Q	U...	BOM Reference	R..	U...	S..	C..	M..
30	Child	3		Child	5/7/2020 12:00:00 AM	5/14/2020 12:00:00 AM	1	EA	1/6 Shown @5/10-20 Changes	A		P...	S...	S...
40	Child	4		Child	5/13/2020 12:00:00 AM	5/19/2020 12:00:00 AM	1	EA	2/6 Shown @5/10-20 Changes	A		P...	S...	S...
50	Child	5		Child	5/18/2020 12:00:00 AM	5/22/2020 12:00:00 AM	1	EA	3/6 Shown @5/10-20 Changes	A		P...	S...	S...
60	Child	6		Child	5/10/2020 12:00:00 AM		1	EA	4/6 Shown @5/10-20 Changes	A		P...	S...	S...
80	Child	8		Child	5/12/2020 12:00:00 AM	5/20/2020 12:00:00 AM	1	EA	5/6 Shown @5/10-20 Changes	A		P...	S...	S...
90	Child	9		Child	5/10/2020 12:00:00 AM	5/20/2020 12:00:00 AM	1	EA	6/6 Shown @5/10-20 Changes	A		P...	S...	S...

Figure 193.

To be evaluated as a **Physical Part** BOM change during a time period, a **Physical Part BOM** Item must have its property values as follows:

- **Started On** greater than or equal to **Period Start** and less than **Period End**.

OR

- **Ended On** less than or equal to **Period End** and more than **Period Start**.

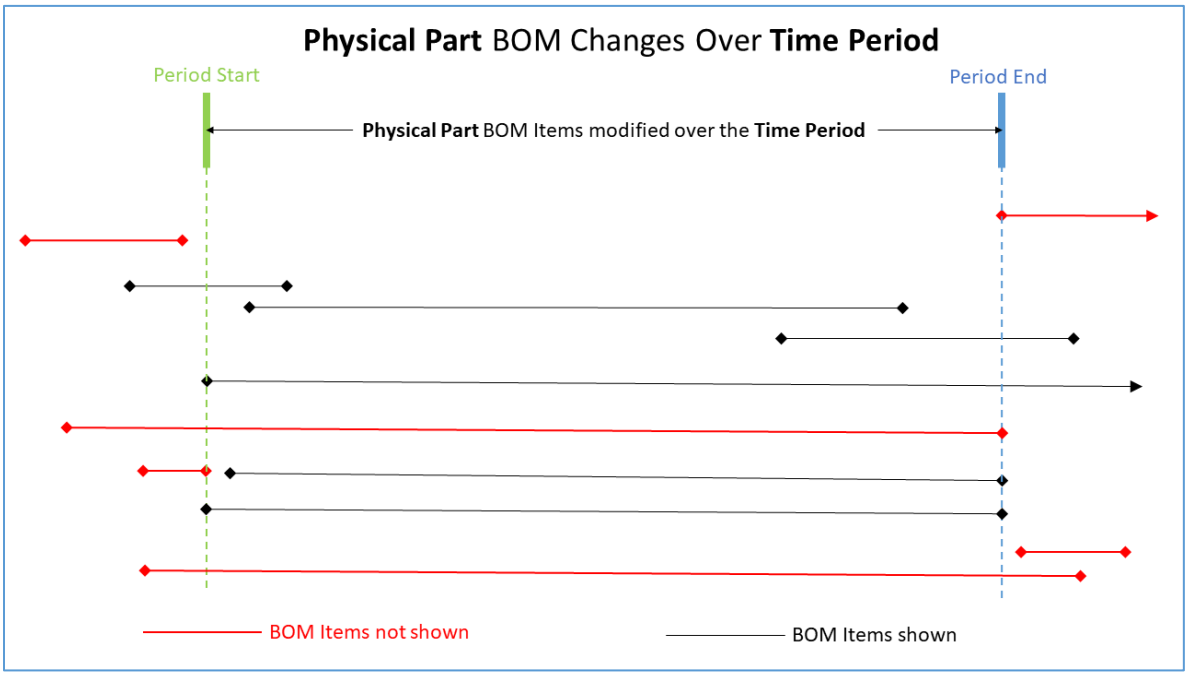


Figure 194.

5.2.2 Managing single-level Physical Part BOMs with standard procedures

Note: Only an **Asset Admin** Identity member can manage one or more **Physical Part BOM** Items regardless of the current State of their parent (source) **Physical Part** Item. An **Asset Editor** Identity member can do this if the parent is in the **Preliminary** State.

Depending on your Asset Identity membership and the current State of a parent **Physical Part** Item, you can create, modify, or delete **Physical Part BOM** Items given in this **Physical Part** Item following the standard Aras Innovator procedures for Relationship Items as discussed in this section.

Additionally, the DTC application features the Remove-and-Replace operation on UI to simplify editing of a single-level Physical Part BOM structure when a component is replaced in an assembly. For details, refer to section 5.2.3 Removing-and-replacing Physical Parts in BOMs.

5.2.2.1 Adding Physical Part Items to BOMs

Adding a **Physical Part** Item to another **Physical Part** Item as a child creates a new **Physical Part BOM** Relationship Item that defines the given parent-child installation in the Physical Part BOM structure.

To add one or more child (related) **Physical Part** Items to a parent (source) **Physical Part** Item:

1. Open the Item view of the parent **Physical Part** Item.
2. Click **Edit** on the parent **Physical Part** Item toolbar.
3. Go to the **BOM** accordion tab.
4. Click **Add Physical Parts** on the **Physical Parts** Grid toolbar.

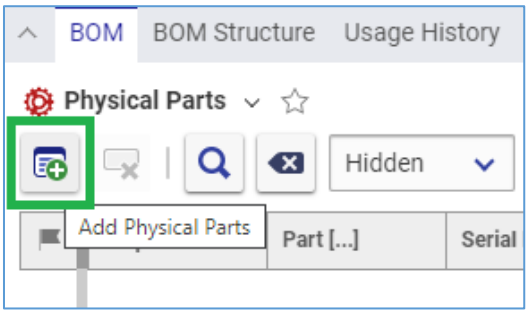


Figure 195.

The **Select Items – Physical Parts** dialog pops up.

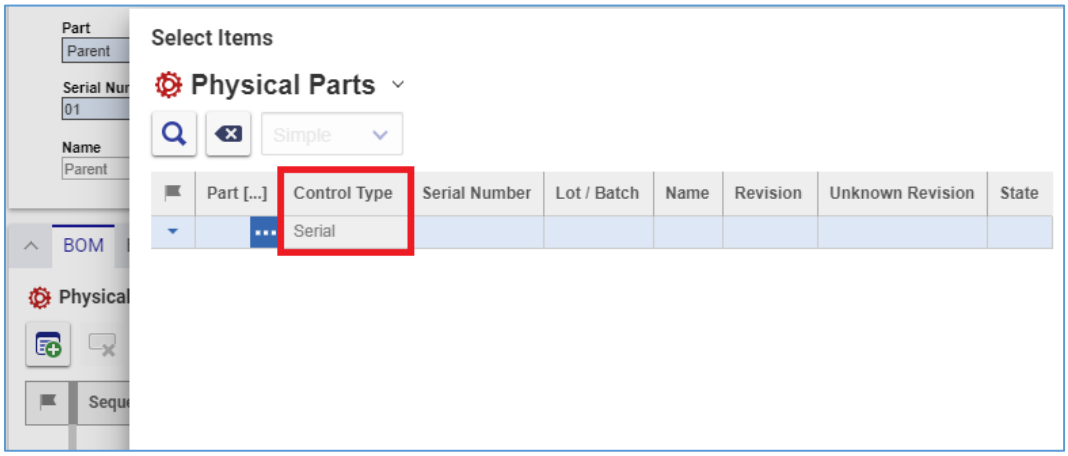


Figure 196.

5. Using the standard search procedure, search for and select all necessary, valid child **Physical Part** Items. The selected Items populate the **Physical Parts** BOM Grid.

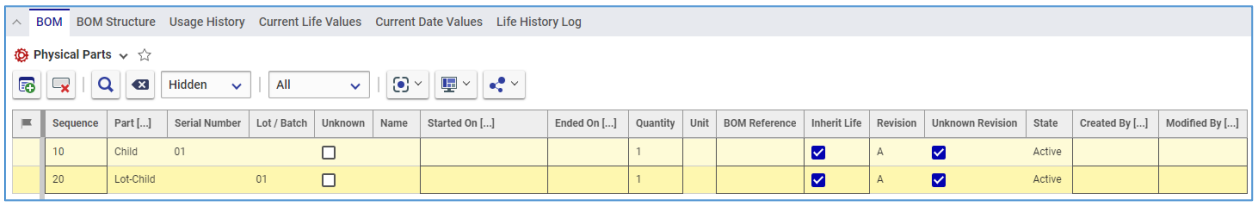


Figure 197.

Note: An error is raised if you try to add the parent **Physical Part** Item to itself.

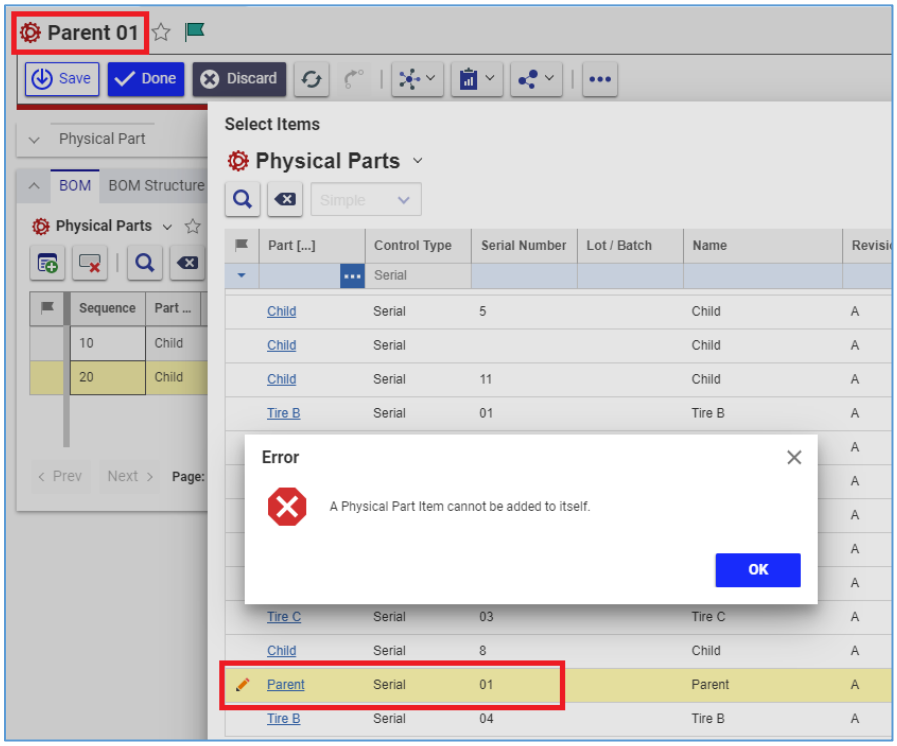


Figure 198.

6. If it is necessary, correct a **Sequence** cell value.

Note: You should assign the **Sequence** values as numbers divisible by ten; see section 3.6 Sequence Values in the DTC Relationships Grids. If several **Physical Part BOM** Items have the same **Sequence** value, each of them should have the same **BOM Reference** value; see section 5.1 Understanding Physical Part BOM Relationship Items.

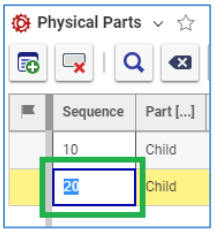


Figure 199.

7. Fill in the **Started On** cell of each new **Physical Part BOM** Item with the **date** and **time** when the represented component has been installed to the assembly.

Note: A **Started On** or **Ended On** date and time value should not be in the future.

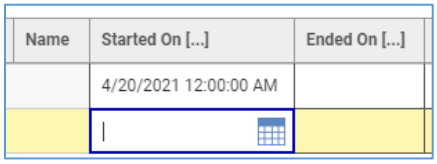


Figure 200.

8. If a given new **Physical Part BOM** Item is historical, provide its **Ended On** cell with the **date** and **time** when the given represented component has been removed from the assembly.

Note: The component cannot be removed before being installed: the given Item **Ended On** must be later than its **Started On**.

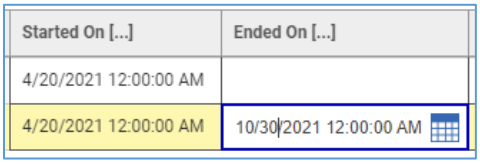


Figure 201.

9. If it is necessary, correct the **Quantity** cell value.

Note: A serial-controlled child **Physical Part** Item can be in a BOM only in the quantity of **1**.

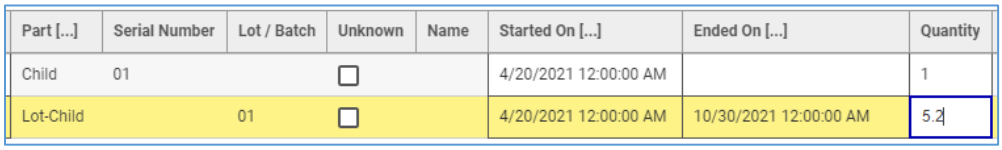


Figure 202.

10. Fill in the **BOM Reference** cell of each new **Physical Part BOM** Item with a reference designator of the given represented component against other components in the assembly.

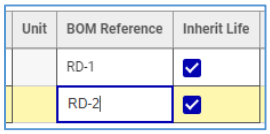


Figure 203.

11. If it is necessary, correct the **true** value set by default in the **Inherit Life** check box.

Note: The **Inherit Life** check box can be only **false** for a lot-controlled or non-controlled child **Physical Part** Item.

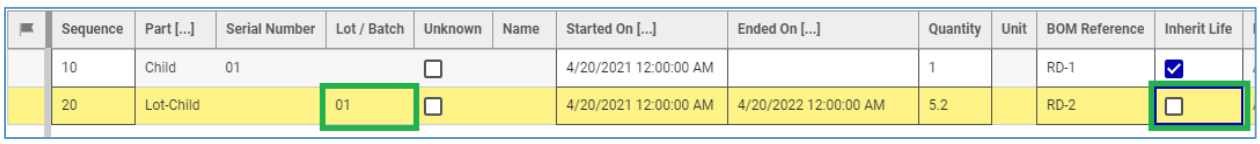


Figure 204.

12. Click either **Save** or **Done** on the parent **Physical Part** Item toolbar.

    The given child **Physical Part** Items are added to the parent **Physical Part** Item. Corresponding new **Physical Part BOM** Relationship Items are created and keep BOM installation data on these children. The **Created By** and **Modified By** cells are automatically populated with **your** Identity. The **Physical Parts BOM** Grid shows only the current **Physical Part BOM** Items if the **When Active** drop-down is set to **Current**.

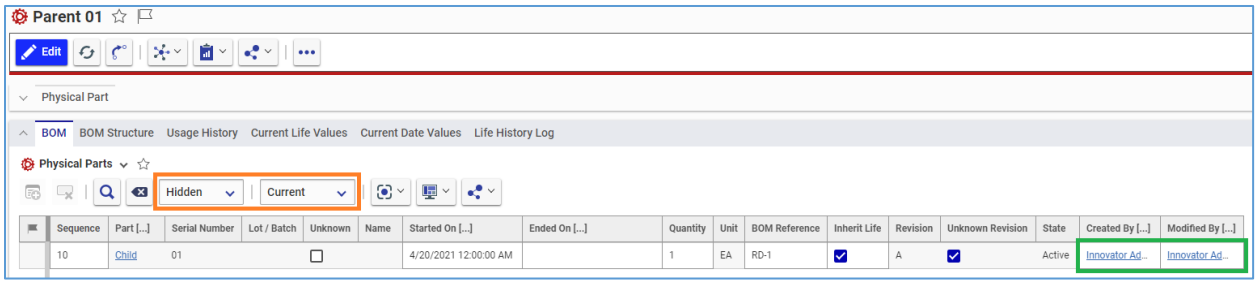


Figure 205.

To see all or some of the created **Physical Part BOM** Items, use the **When Active** drop-down filter as discussed in section 5.2.1 Exploring single-level Physical Part BOMs.

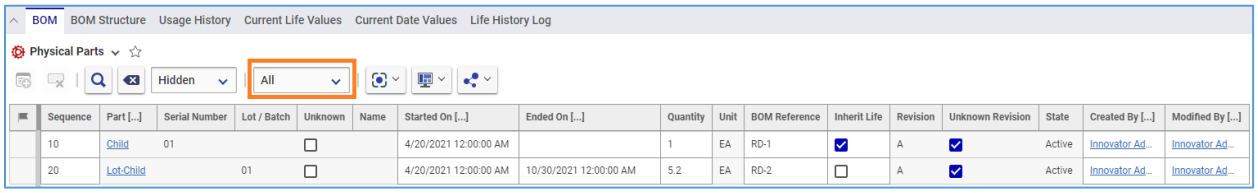


Figure 206.

Upon saving the parent **Physical Part** Item, the system validates the **Physical Part BOM** Items and raises an intelligent error if the validation fails; see section *5.2.2.5 Physical Part BOM validation*.

5.2.2.2 Modifying Physical Part BOM Items

Modifying a **Physical Part BOM** Item is changing the editable properties of a Relationship Item between parent and child **Physical Part** Items:

- **Sequence**
- **Started On**
- **Ended On**
- **Quantity**
- **BOM Reference**
- **Inherit Life**

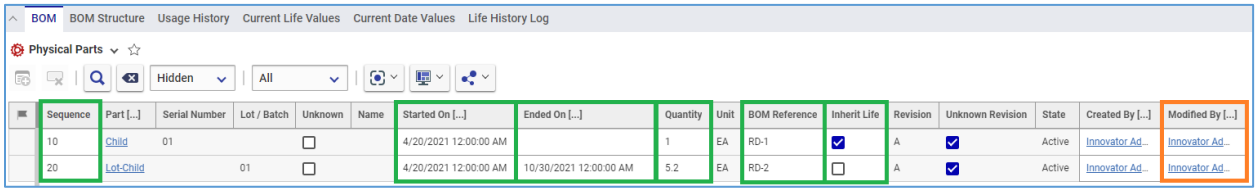


Figure 207.

To modify a **Physical Part BOM** Item:

1. Open the Item view of its source (parent) **Physical Part** Item.
2. Click **Edit** on the parent **Physical Part** Item toolbar.
3. Go to the **BOM** accordion tab.
4. If it is necessary, correct the **Sequence** cell value.

Note: You should assign the **Sequence** values as numbers divisible by ten; see section 3.6 Sequence Values in the DTC Relationships Grids. If several **Physical Part BOM** Items have the same **Sequence** value, each of them should have the same **BOM Reference** value; see section 5.1 Understanding Physical Part BOM Relationship Items.

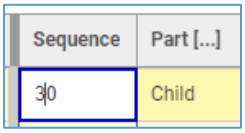


Figure 208.

5. If it is necessary, correct the **Started On** cell value by typing or using the **Date dialog**.

Note: A **Started On** or **Ended On** date and time value should not be in the future.

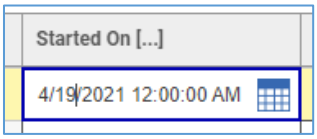


Figure 209.

6. If it is necessary, populate the **Ended On** cell or correct its existing value by typing or using the **Date dialog**.

   If this cell had no value before and gets some at this step, the given **Physical Part BOM** Item becomes historical: its represented component has been removed from the assembly.

Note: The component cannot be removed before being installed: the given Item **Ended On** must be later than its **Started On**.

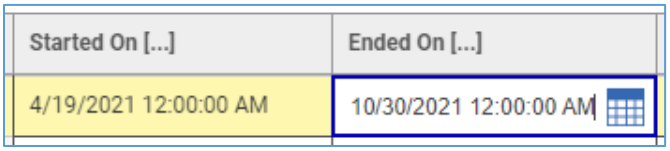


Figure 210.

7. If it is necessary, correct the **Quantity** cell value.

Note: A serial-controlled child **Physical Part** Item can be in a BOM only in the quantity of **1**.

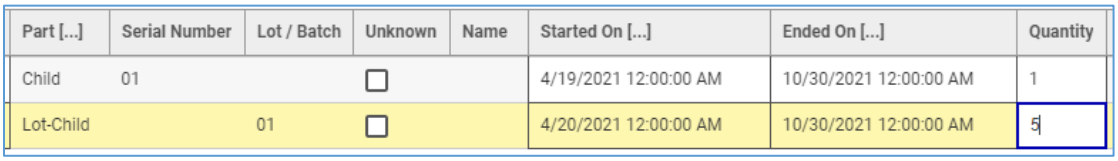


Figure 211.

8. If it is necessary, correct the **BOM Reference** cell value.

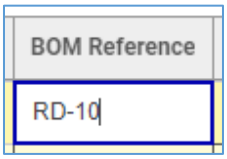
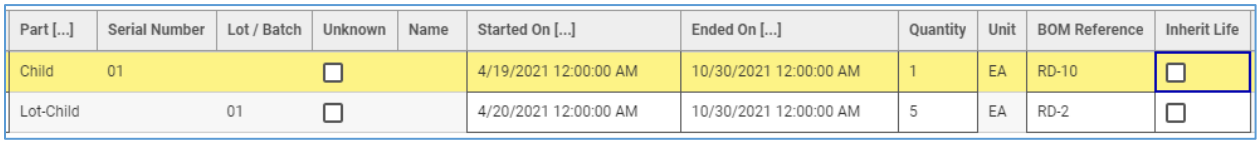


Figure 212.

9. If it is necessary, correct the **Inherit Life** cell value.

Note: The **Inherit Life** check box can be only **false** for a lot-controlled or non-controlled child **Physical Part** Item.



Part [...]	Serial Number	Lot / Batch	Unknown	Name	Started On [...]	Ended On [...]	Quantity	Unit	BOM Reference	Inherit Life
Child	01		☐		4/19/2021 12:00:00 AM	10/30/2021 12:00:00 AM	1	EA	RD-10	☐
Lot-Child		01	☐		4/20/2021 12:00:00 AM	10/30/2021 12:00:00 AM	5	EA	RD-2	☐

Figure 213.

10. Click either **Save** or **Done** on the parent **Physical Part** Item toolbar.

    The given **Physical Part BOM** Items are edited and keep new data on the existing children. The **Modified By** cell is automatically set with **your** Identity. The **Physical Parts BOM** Grid shows only the **Physical Part BOM** Items that remain current if the **When Active** drop-down is set to **Current**.

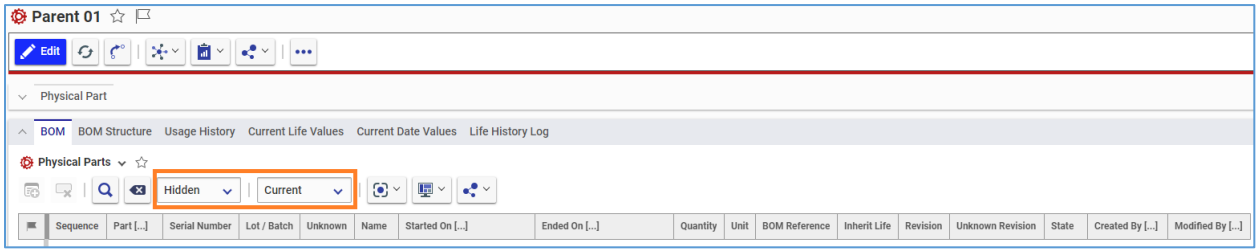


Figure 214.

To see all or some of the created **Physical Part BOM** Items, use the **When Active** drop-down filter as discussed in section 5.2.1 Exploring single-level Physical Part BOMs.

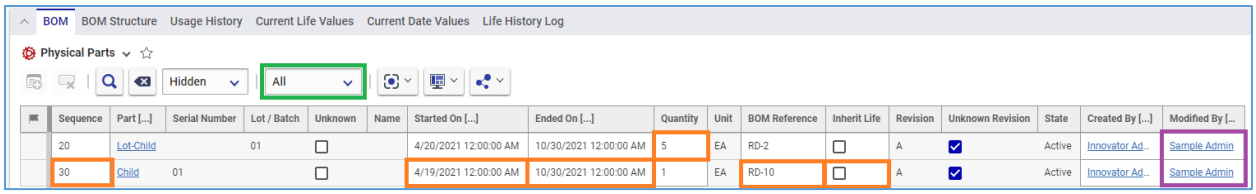


Figure 215.

Upon saving the parent **Physical Part** Item, the system validates the **Physical Part BOM** Items and raises an intelligent error if the validation fails; see section *5.2.2.5 Physical Part BOM validation*.

5.2.2.3 Removing Physical Parts from BOMs

To remove one or more child **Physical Part** Items from a parent **Physical Part** Item:

1. Open the Item view of the parent **Physical Part** Item.
2. Click **Edit** on the parent **Physical Part** Item toolbar.

3. Go to the **BOM** accordion tab.
4. Click **each** of the **Physical Part** Items to be removed. The given **Physical Part BOM** Items are highlighted.

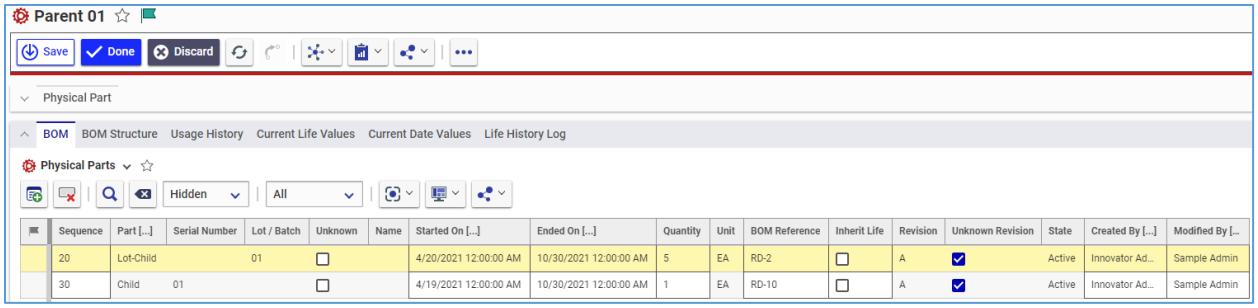


Figure 216.

5. Click the **Delete Physical BOM Line** button on the **Physical Parts** Grid toolbar.

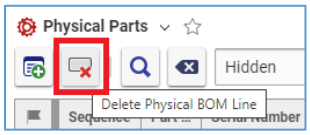


Figure 217.

The given **Physical Part BOM** Items become unavailable: the text is light grey.

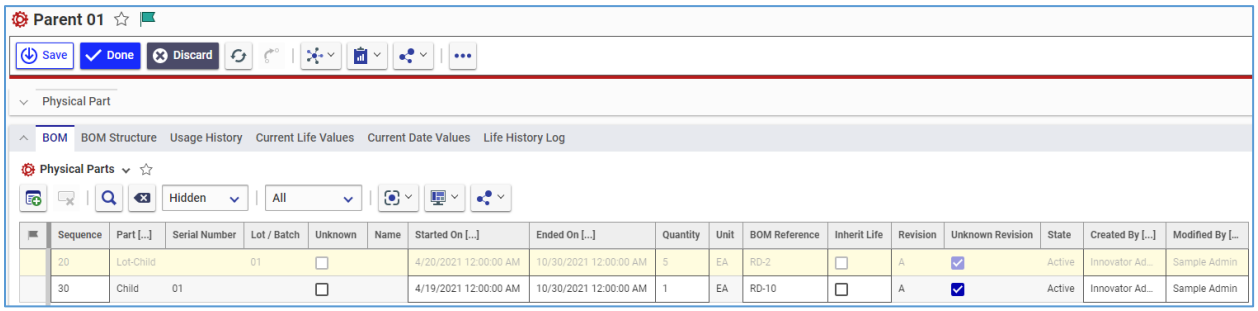


Figure 218.

6. Click either **Save** or **Done** on the parent **Physical Part** Item toolbar.

   The given unavailable child **Physical Part** Items are removed from the parent **Physical Part** Item. Their **Physical Part BOM** Relationship Items are completely deleted from the system. The given child **Physical Part** Items are not deleted from the system.

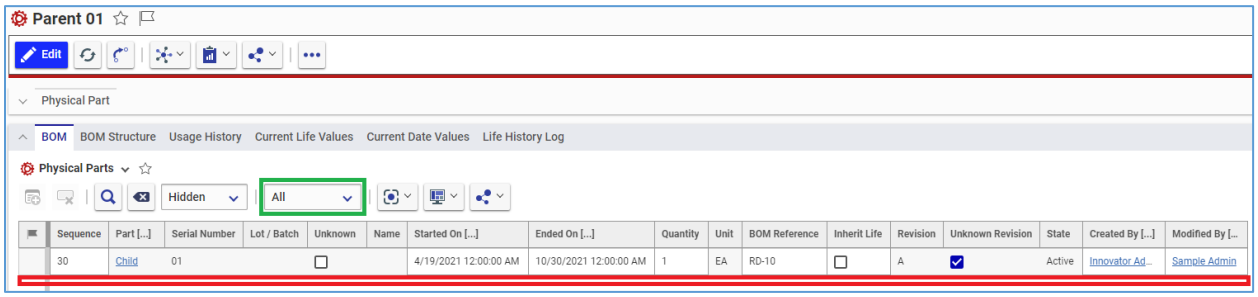


Figure 219.

Warning: The deleted **Physical Part BOM** Items cannot be restored.

5.2.2.4 Physical Part BOM validation

Once you click **Save** or **Done** on a parent **Physical Part** Item toolbar, the system validates its single-level Physical Part BOM structure and raises an appropriate error if at least one **Physical Part BOM** Item either:

- Is missing a **BOM Reference** value.

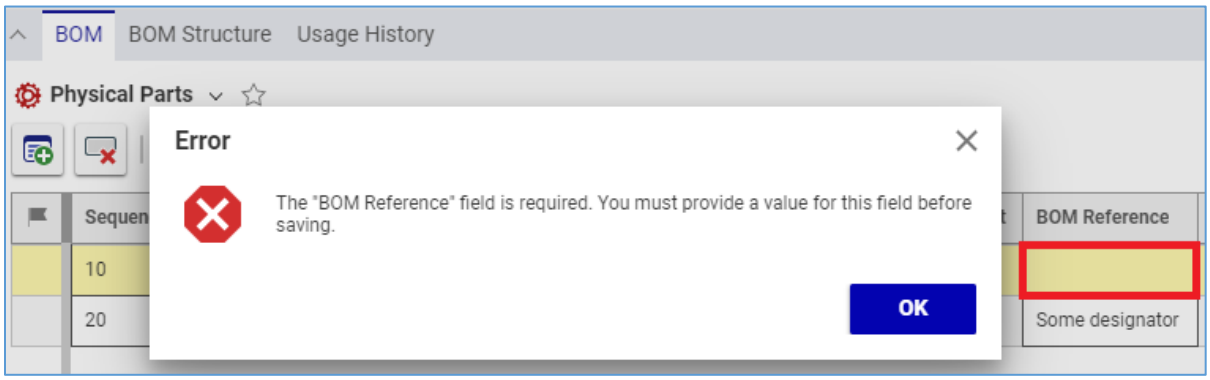


Figure 220.

- Is missing a **Started On** value.

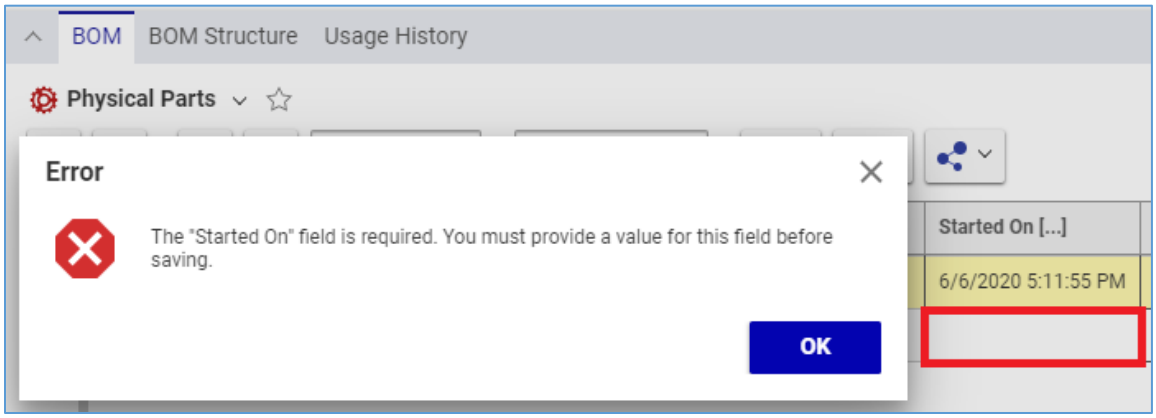


Figure 221.

- Has some time point or period when a given related **Physical Part** Item (child) is already used in the given or another Physical Part BOM structure.

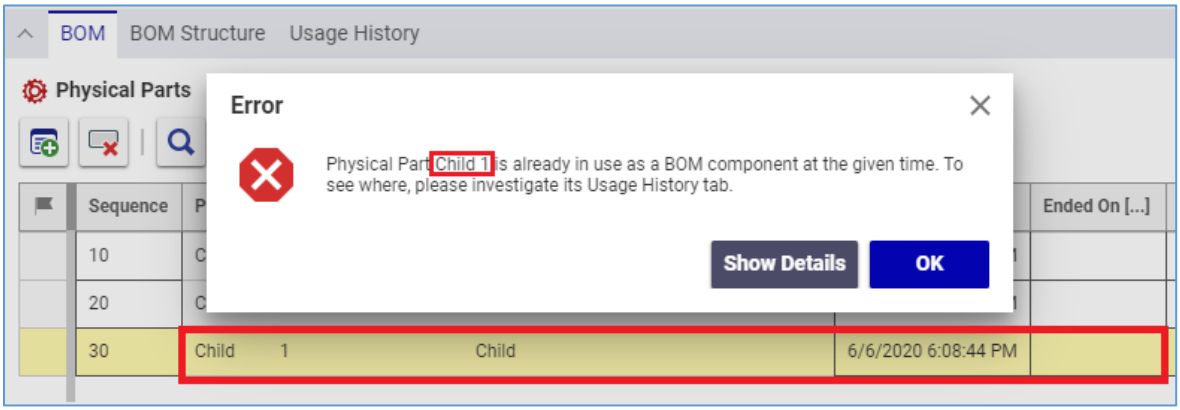


Figure 222.

- Has **Ended On** before **Started On**.

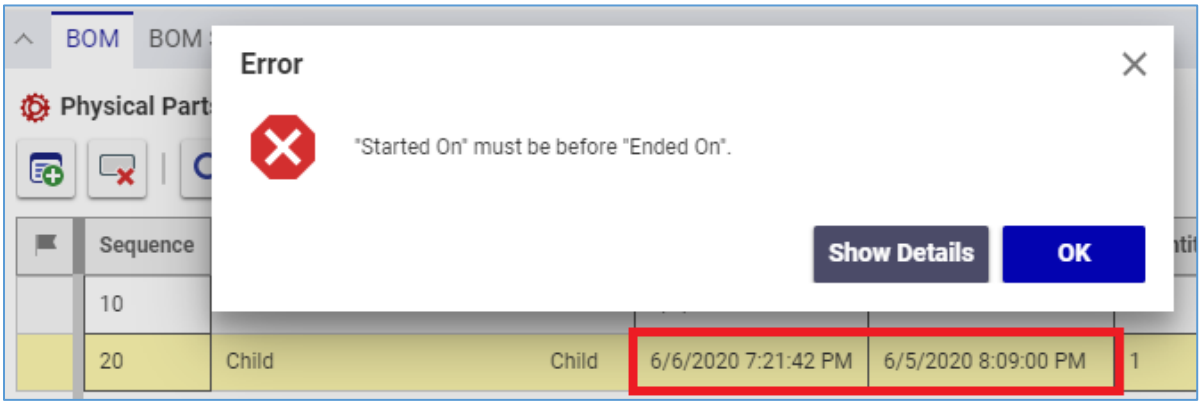


Figure 223.

- Has **Started On** later than the current system date and time.

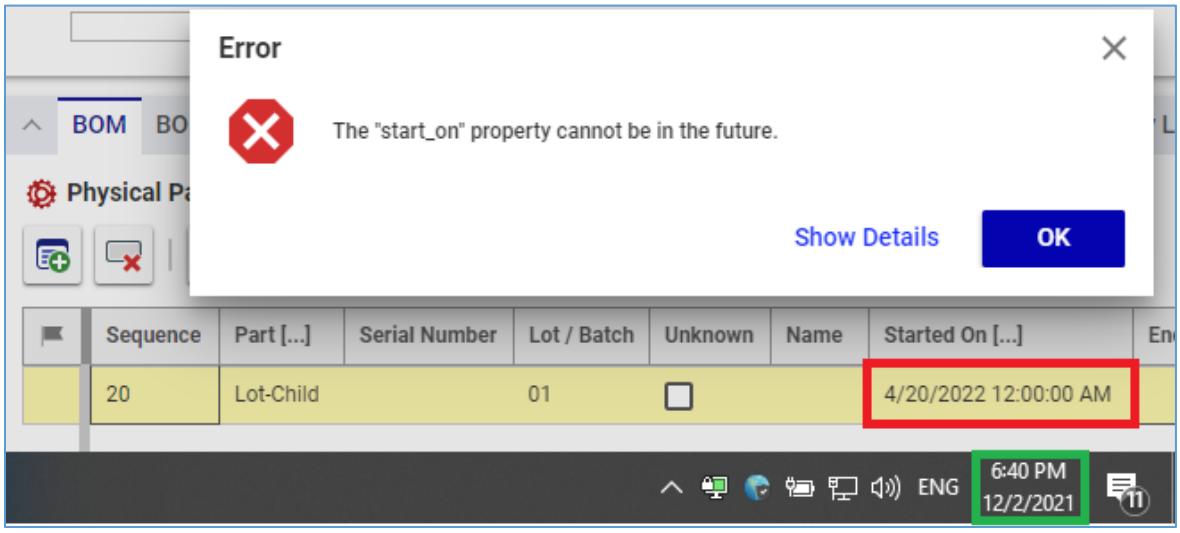


Figure 224.

- Has **Ended On** later than the current system date and time.

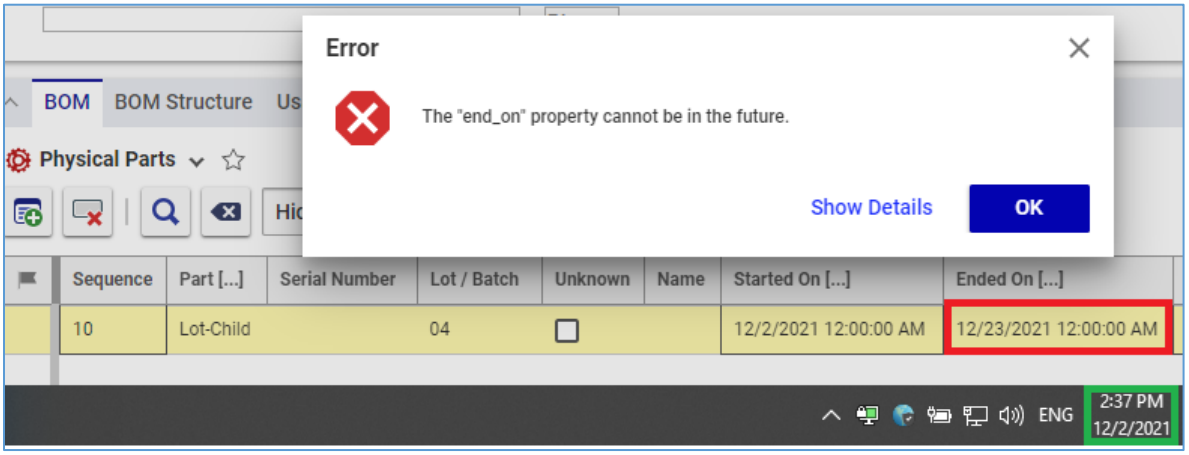


Figure 225.

- Has **Quantity** other than **1** for a serial-controlled related **Physical Part** Item (child).

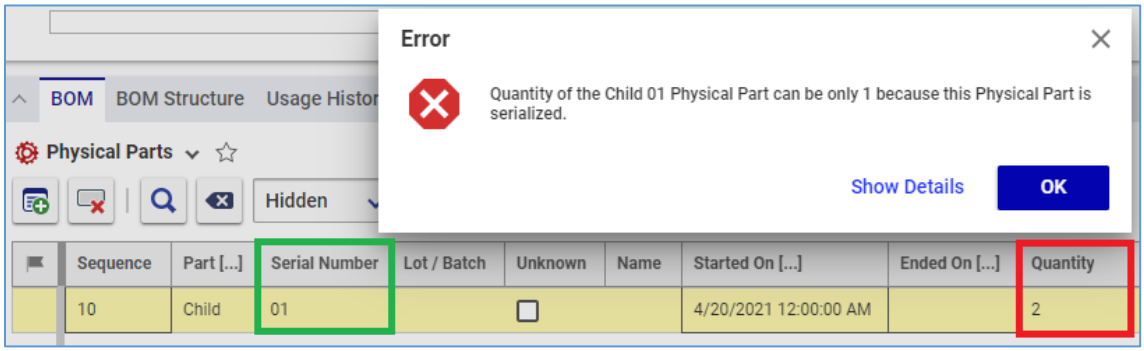


Figure 226.

- Has **Quantity** equal to or less than **0** (zero) for a lot-controlled or non-controlled related **Physical Part** Item (child).

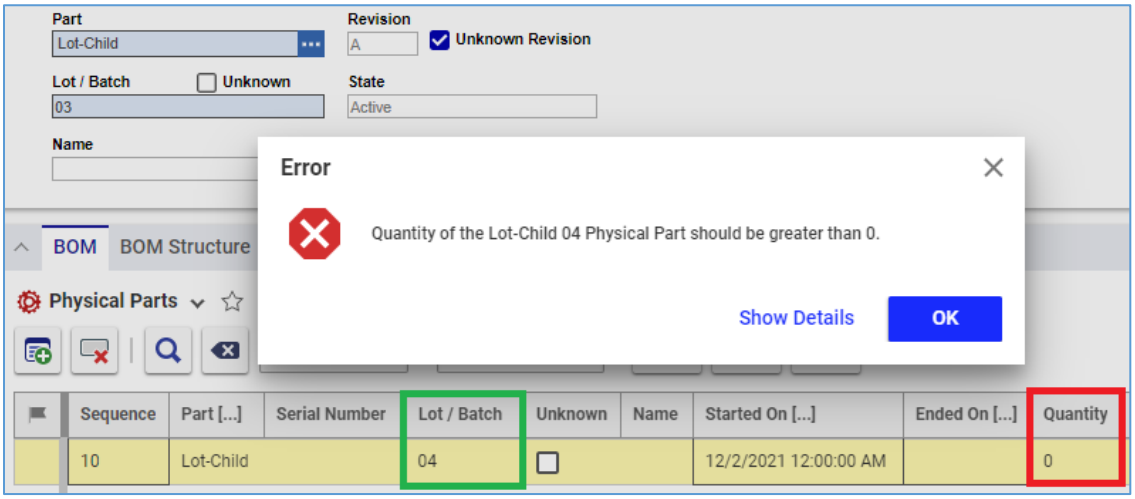


Figure 227.

- Has a related **Physical Part** Item (child) of an unallowed Control Type.

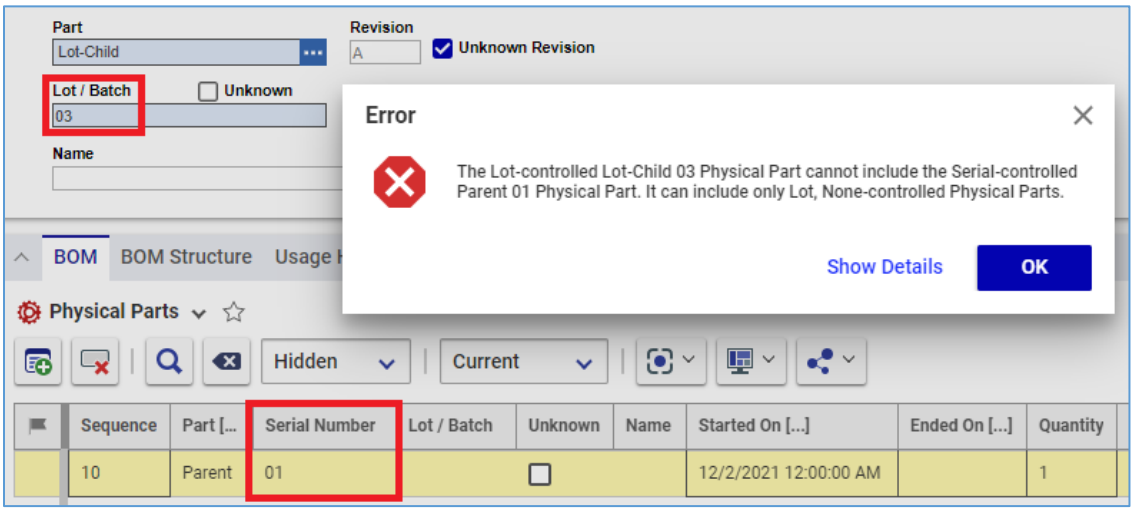


Figure 228.

- Has a lot-controlled or non-controlled related **Physical Part** Item (child) with **Inherit Life** set to **true**.

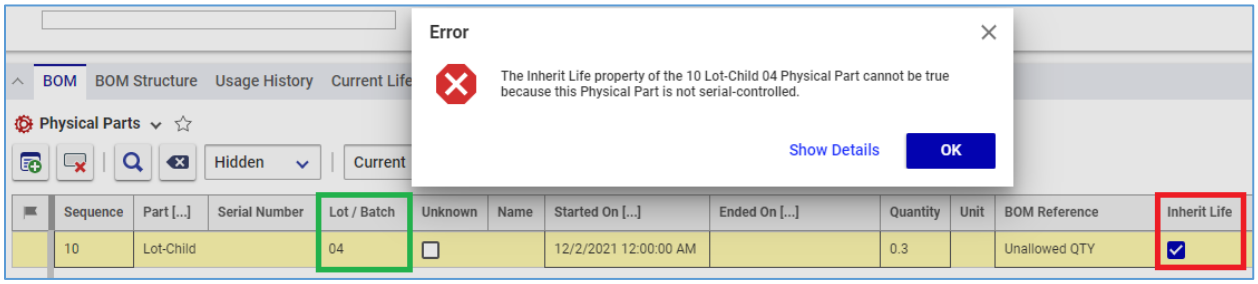


Figure 229.

5.2.3 Removing-and-replacing Physical Parts in BOMs

Note: Only an **Asset User**, **Asset Editor**, or **Asset Admin** Identity member can apply a Remove-and-Replace operation to one or more **Physical Part BOM** Relationship Items and only when their parent (source) **Physical Part** Item is in the **Active** State.

A **Physical Part** Item in the **Active** State should represent a manufactured asset commissioned into service. This asset should be fully assembled, and its engineering BOM is not supposed to be changed. During regular operational running, the asset may require service and maintenance, including parts replacement, overhaul, or repair.

The DTC application imposes constraints and requires full Access Rights to manage a **Physical Part** Item in the **Active** State, including the Rights to create, edit, or delete a **Physical Part BOM** Relationship Item following the standard Aras Innovator procedures for Relationship Items; see section 5.2.2 Managing single-level Physical Part BOMs with standard procedures.

To simplify manipulations over a single-level Physical Part BOM structure of a parent (source) **Physical Part** Item in the **Active** State, the DTC application features a Remove-and-Replace (R&R) UI operation that digitally represents a situation when a component was removed from an assembly for inspection, repair, maintenance, or replacement and was either reinstalled back or replaced by another component at its place in the assembly.

A single R&R operation manipulates over a single BOM slot defined by the same **Sequence**, **BOM Reference**, and **Inherit Life** value combination of one or more **Physical Part BOM** Relationship Items. For more details on BOM slots, refer to section *5.1 Understanding Physical Part BOM Relationship Items*. The system raises an error and does not allow to save an R&R operation if it includes **Physical Part BOM** Relationship Items from different BOM slots; for error message see section 5.2.3.4 Validation of a Remove-and-Replace operation.

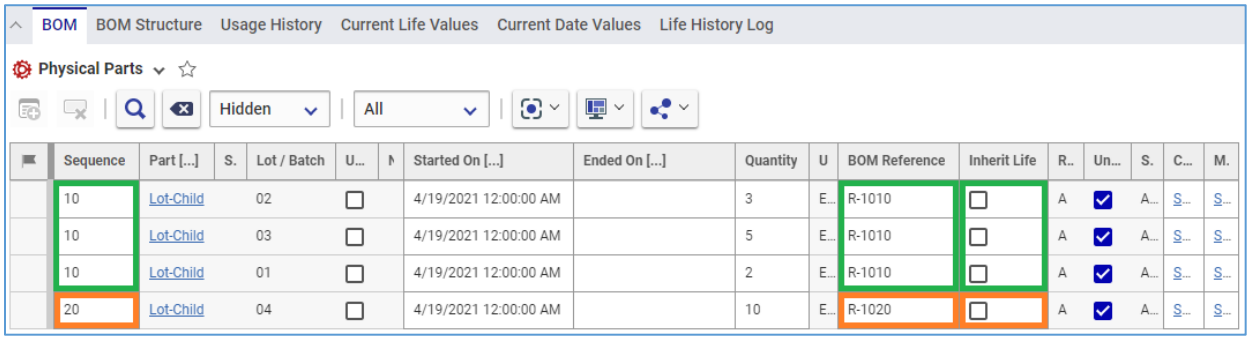


Figure 230.

The R&R operation can have three cases for related **Physical Part** Items (children):

- One-to-one replacement: *one serial-controlled* child replaces *one serial-controlled* child; see section 5.2.3.1 One-to-one Remove-and-Replace case.
- Split: *one* or *more non-serialized* children replace *one non-serialized* child; see section 5.2.3.2 Split Remove-and-Replace case.
- Merge: *one non-serialized* child replaces *multiple non-serialized* children; see section 5.2.3.3 Merge Remove-and-Replace case.

The DTC application does not support *many-to-many* replacement: *multiple* children replace *multiple* children regardless of their Control Type. You should break multiple replacements into a series of R&R operations. The system will allow you to perform an R&R operation with a *many-to-many* replacement but not save it: you will be required to discard all changes. See section 5.2.3.4 Validation of a Remove-and-Replace operation.

This section discusses the R&R operation details general for all cases. Dedicated subsections describe specifics of each case, including the procedure.

Each case has:

- A *replaced* **Physical Part BOM** Relationship Item with a related **Physical Part** Item (child) and its **Quantity** value that should be removed from a BOM at a given date and time.
- A *replacing* **Physical Part BOM** Relationship Item with a related **Physical Part** Item (child) and its **Quantity** value that should be installed into a BOM instead of the removed (*replaced*) child at a given date and time.

To initiate a necessary R&R operation case, you should select *one* or *more replaced* **Physical Part BOM** Relationship Items within *one* BOM slot that are valid for an R&R operation, right-click them, and click the **Remove-and-Replace** action menu button.

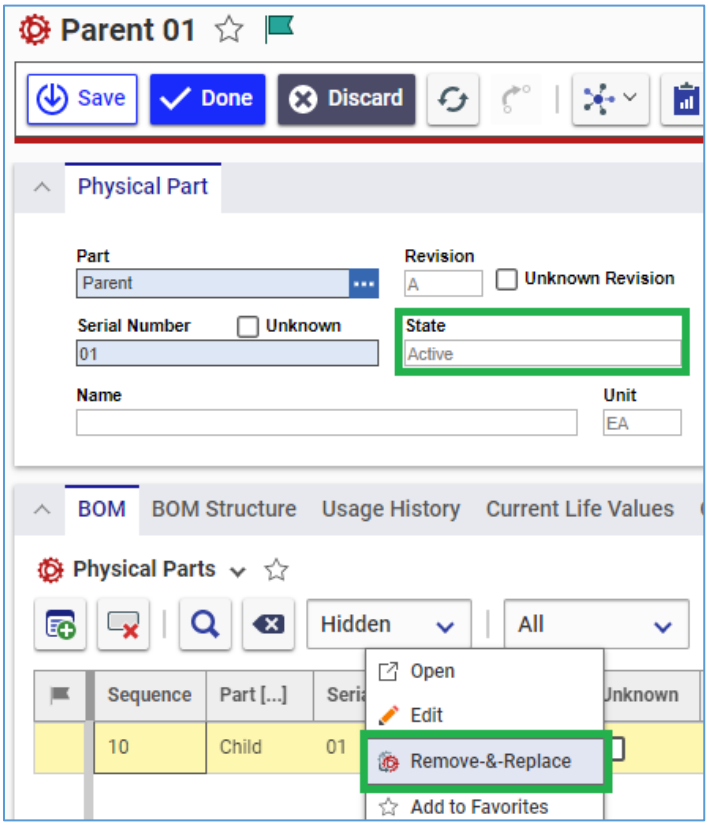


Figure 231.

The **Remove-and-Replace** button is not available when a parent (source) **Physical Part** Item is in the **Preliminary** State.

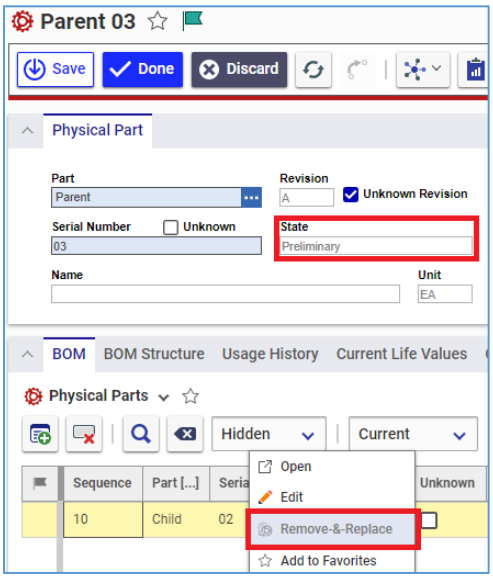


Figure 232.

The selected *replaced* **Physical Part BOM** Relationship Items are valid for an R&R operation when all their child **Physical Part** Items have the same Control Type within a single BOM slot. If you had selected **Physical Part** Items with different Control Types, the system will allow you to perform an R&R operation but not save it: you will be required to discard all changes. See section 5.2.3.4 Validation of a Remove-and-Replace operation.

Depending on a combination of the quantity of the selected *replaced* **Physical Part BOM** Relationship Items and their Control Types, the system automatically deduces the R&R case:

- *One serial-controlled* **Physical Part BOM** Relationship Item—one-to-one replacement.
- *One non-serialized* **Physical Part BOM** Relationship Item—split.
- *Multiple non-serialized* **Physical Part BOM** Relationship Items—merge.

If the system detects none of the abovementioned combinations, it will allow you to perform an R&R operation but not save it: you will be required to discard all changes. See section 5.2.3.4 Validation of a Remove-and-Replace operation. The selected *replaced* **Physical Part BOM** Relationship Items are illegal for an R&R operation.

Then, you provide a removal and installation date and time value that populates:

- The **Ended On** cells of the *replaced* **Physical Part BOM** Relationship Items.
- The **Started On** cells of the *replacing* **Physical Part BOM** Relationship Items.

These properties are required for an R&R operation. You can correct their values as you need before saving the R&R operation. However, the *replacing* child cannot be installed before the *replaced* one: a **Started On** value of a *replacing* **Physical Part BOM** Relationship Item should be equal to or later than an **Ended On** value of a *replaced* **Physical Part BOM** Relationship Item.

An R&R operation is retrospective: it represents a part replacement that was in the past. All **Ended On** and **Started On** values should be earlier than your system date and time.

The system raises an error and does not allow to save an R&R operation if it includes an unallowed value in an abovementioned **Ended On** or **Started On** property or if such a property is empty; see section 5.2.3.4 Validation of a Remove-and-Replace operation.

The **Ended On** properties of *replacing* **Physical Part BOM** Relationship Items must be empty. The system raises an error and does not allow to save an R&R operation if it includes a populated **Ended On** property of a *replacing* **Physical Part BOM** Relationship Item; see section 5.2.3.4 Validation of a Remove-and-Replace operation.

An R&R operation is available only for currently installed children: such *replaced* **Physical Part BOM** Relationship Items should have their **Ended On** properties empty before the operation. The system will allow you to perform an R&R operation over historical *replaced* **Physical Part BOM** Relationship Items but not save it: you will be required to discard all changes. See section 5.2.3.4 Validation of a Remove-and-Replace operation.

Finally, you should also select one or more *replacing* **Physical Part** Items (children) valid for an R&R operation. To be valid, all *replacing* **Physical Part** Items should be in the **Active** State and have the same **Control Type** value that all *replaced* **Physical Part** Items have.

A BOM slot should have the same Control Type before and after an R&R operation. The system raises an error and does not allow to save an R&R operation if it includes *replacing* and *replaced* **Physical Part** Items of different Control Types; see section 5.2.3.4 Validation of a Remove-and-Replace operation.

You can replace a related **Physical Part** Item (child) by itself. This may be a case when a part was removed for testing, maintenance, or other reason and put back in its original place.

The system creates a new *replacing* **Physical Part BOM** Relationship Item for each *replacing* child **Physical Part** Item you had selected. It copies the **Sequence**, **BOM Reference**, and **Inherit Life** property values from the old *replaced* **Physical Part BOM** Relationship Item to the new *replacing* **Physical Part BOM** Relationship Item.

Filling in the **Quantity** property of the new *replacing* **Physical Part BOM** Relationship Item is specific for each R&R operation case: see the dedicated subsections.

An R&R operation should not change a BOM slot quantity. The sum of the **Quantity** property values of all *replaced* **Physical Part BOM** Relationship Items should equal the sum of the **Quantity** property values of all *replacing* **Physical Part BOM** Relationship Items. The system raises an error and does not allow to save an R&R operation if this operation changes the BOM slot quantity; see section 5.2.3.4 Validation of a Remove-and-Replace operation.

Currently, the DTC application supports only one R&R operation over a single BOM slot in a single edit session. You should have done only one R&R operation over one BOM slot before you click either **Save** or **Done** on a parent (source) **Physical Part** Item toolbar. If you need to perform multiple replacements in one BOM slot, do each replacement in a separate edit session. Also, you should not combine an R&R operation with any other Aras Innovator standard operation, like creation, modification, or removal, over the parent **Physical Part** Item or its single-level BOM.

If you try to save a parent with multiple R&R operations over a single BOM slot or one R&R operation combined with any other standard operation, the system will not allow you to save your changes and raise an error message; see section 5.2.3.4 Validation of a Remove-and-Replace operation. In such a case, you need to discard all your changes and do each R&R operation separately.

When you click either **Save** or **Done** on a parent (source) **Physical Part** Item toolbar after you had done an R&R operation, the server side validates your updates against the following constraints:

- General for **Physical Part BOM** Relationship Items; discussed in section 5.1 Understanding Physical Part BOM Relationship Items. For their error messages, see section *5.2.2.5 Physical Part BOM validation*.
- Specific for a given R&R operation case; discussed in this section and its subsections dedicated for each case. For their error messages, see section 5.2.3.4 Validation of a Remove-and-Replace operation.

5.2.3.1 One-to-one Remove-and-Replace

The *one-to-one replacement* R&R case represents an operation when *one serialized* part replaces *another serialized* part. Thus, this case is available only for *one serial-controlled* **Physical Part BOM** Relationship Item, and the system always populates the **Quantity** property with **1**.

A *non-serialized replaced* or *replacing* **Physical Part** Item is not allowed for a one-to-one replacement operation:

- If you want to replace the whole quantity of a *non-serialized* **Physical Part** Item in the assembly with the whole quantity of another *non-serialized* part, use the split case where select *one replacing* **Physical Part** Item and do not change the **Quantity** property value.
- If you had selected a *replacing non-serialized* **Physical Part** Item, the system will allow you to perform a one-to-one replacement operation but not save it: you will be required to discard all changes. See section 5.2.3.4 Validation of a Remove-and-Replace operation.

To remove and replace a given current *serial-controlled* child **Physical Part** Item in a Physical Part BOM structure:

1. Open the Item view of a **Physical Part** Item which is currently a direct parent of the given child.
2. Click **Edit** on the parent **Physical Part** Item toolbar.
3. Go to the **BOM** accordion tab.
4. Right-click *one* row with the given *serial-controlled* child **Physical Part** Item to be *replaced* and click **Remove-and-Replace**. This row is a *replaced* (old) **Physical Part BOM** Relationship Item.

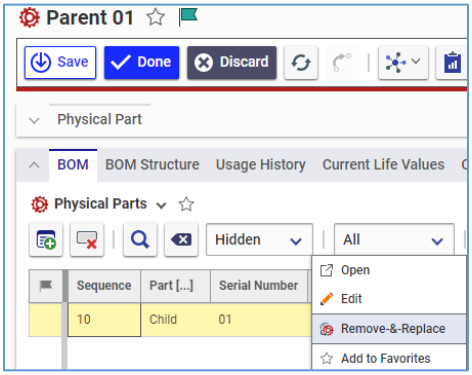


Figure 233.

The **Select Date** dialog pops up.

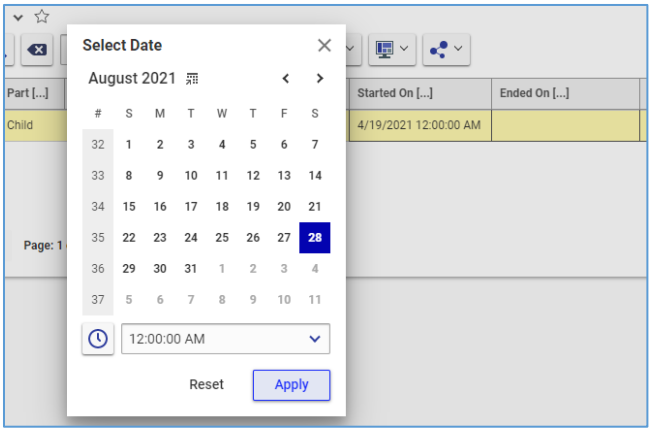


Figure 234.

5. In the **Select Date** dialog, specify remove-and-replace date and time and click **Apply**.

   The **Select Items – Physical Parts** dialog box pops up prepopulated with search criteria as follows:

   - **Part**: a **Part** Item of the *replaced* child **Physical Part** Item. You can change the **Part** criterion as you need.
   - **State**: **Active** and frozen. Only child **Physical Part** Items in the **Active** State can be added to a parent **Physical Part** Item in the **Active** State.

Figure 235.

6. In the **Select Items – Physical Parts** dialog, search for and select a *replacing serial-controlled* child **Physical Part** Item using the standard search procedure.

Note: You can select only one **Physical Part** Item in the **Select Items – Physical Parts** dialog.

If you are replacing the child **Physical Part** Item with itself, the **Confirm** dialog box pops up. Otherwise, no dialog box pops up.

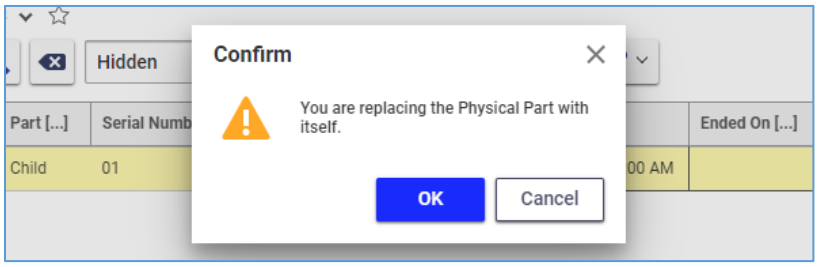


Figure 236.

Note: An error is raised if you try to add a parent (source) **Physical Part** Item to its BOM.

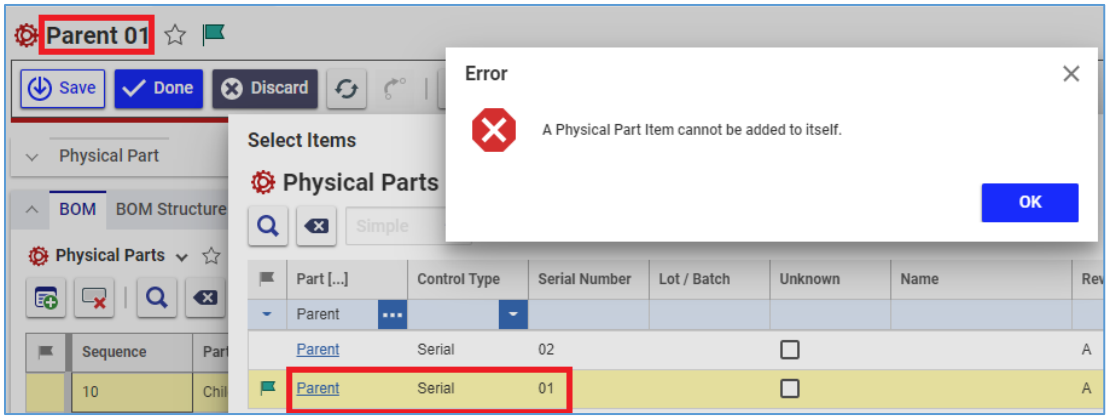


Figure 237.

7. Click **OK** in the **Confirm** dialog box.

Note: You should take step 7 only when you are replacing the child **Physical Part** Item with itself. If replacing with another **Physical Part** Item, the system behaves as below-described right after step 6.

The date and time value you had specified at step 5 populates the **Ended On** property of the *replaced* (old) **Physical Part BOM** Item.

The selected *replacing* child **Physical Part** Item populates the grid as a new (*replacing*) **Physical Part BOM** Item with its properties set as follows:

- **Sequence**, **BOM Reference**, and **Inherit Life**: the same values as the *replaced* (old) **Physical Part BOM** Item has.
- **Started On**: equal to the **Ended On** property value of the *replaced* (old) **Physical Part BOM** Item.
- **Quantity**: **1**.

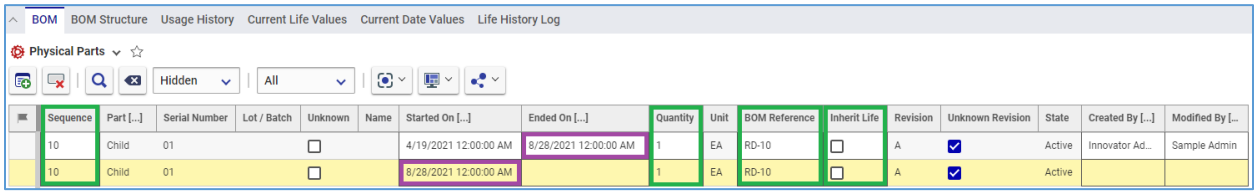


Figure 238.

8. If the *replaced* component was removed from the assembly at a date or time different than specified in the **Select Date** dialog at step 5, correct the **Ended On** property value of the *replaced* (old) **Physical Part BOM** Item by typing or using the **Date dialog**. This value must be the same or earlier than the **Started On** property value of the *replacing* (new) **Physical Part BOM** Item.

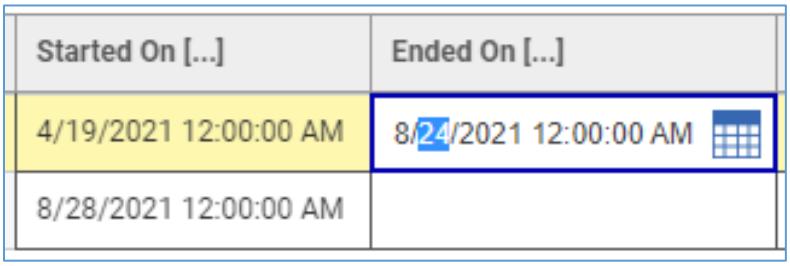


Figure 239.

9. If the *replacing* component was installed into the assembly at a date or time different than specified in the **Select Date** dialog at step 5, correct the **Started On** property value of the *replacing* (new) **Physical Part BOM** Item by typing or using the **Date dialog**. This value must be the same or later than the **Ended On** property value of the *replaced* (old) **Physical Part BOM** Item.

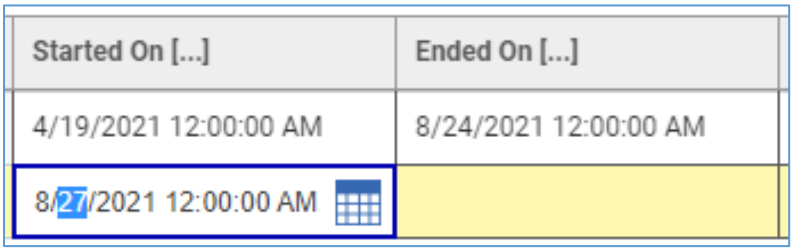


Figure 240.

10. Click either **Save** or **Done** on the parent **Physical Part** Item toolbar.

    The *replaced* (old) **Physical Part** Item becomes a historical child. The *replaced* (old) **Physical Part BOM** Item still exists and keeps installation data on the *replaced* (old) child.

    The *replacing* (new) child **Physical Part** Item is a current child in the BOM. The *replacing* (new) **Physical Part BOM** Item is created to keep BOM installation data on the *replacing* (new) child.

    Both *replaced* (old) and *replacing* (new) **Physical Part BOM** Items have **1** in the **Quantity** property and the same **Sequence**, **BOM Reference**, and **Inherit Life** property values.

    Your Identity automatically populates the **Modified By** property of the *replaced* (old) **Physical Part BOM** Item and the **Created By** and **Modified By** properties of the *replacing* (new) **Physical Part BOM** Item.

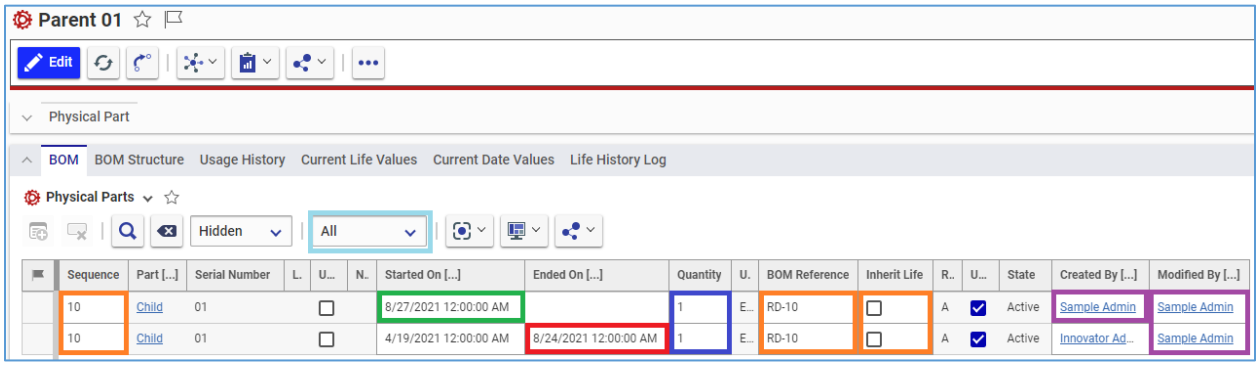


Figure 241.

Upon saving the parent **Physical Part** Item, the server side validates your updates against the following constraints:

- General for **Physical Part BOM** Relationship Items; see section *5.2.2.5 Physical Part BOM validation*.
- Specific for the *one-to-one replacement* R&R case; see section 5.2.3.4 Validation of a Remove-and-Replace operation.

5.2.3.2 Split Remove-and-Replace

The *split* R&R case represents an operation when *one* or *more non-serialized* parts replace *one non-serialized* part. Thus, this case is available only for *one non-serialized replaced* **Physical Part BOM** Relationship Item.

The *replaced* and all *replacing* **Physical Part** Items should be of the same *non-serialized* Control Type: either **Lot / Batch** or **No Control**. If you had selected **Physical Part** Items with different Control Types, the system will allow you to perform a split operation but not save it: you will be required to discard all changes. See section 5.2.3.4 Validation of a Remove-and-Replace operation.

A *serial-controlled* **Physical Part** Item is not allowed for a split operation. If you had selected a *replacing serial-controlled* **Physical Part** Item, the system will allow you to perform a split operation but not save it: you will be required to discard all changes. See section 5.2.3.4 Validation of a Remove-and-Replace operation. The *one-to-one replacement* is the only R&R operation allowed over the *serial-controlled* **Physical Part** Items.

The *split* R&R case includes the following subcases:

- Full one-to-one replacement: *one replacing* **Physical Part BOM** Relationship Item replaces the entire **Quantity** property value of the *replaced* **Physical Part BOM** Relationship Item.

  You should select *one replacing* **Physical Part** Item and do not change the **Quantity** property value for it.

- Full split or one-to-many replacement of whole quantity: *multiple replacing* **Physical Part BOM** Relationship Items replace the entire **Quantity** property value of the *replaced* **Physical Part BOM** Relationship Item.

  You should select *multiple replacing* **Physical Part** Items and provide a **Quantity** property value for each of them. The sum of these **Quantity** property values should be equal to the original one.

- Partial one-to-one replacement: *one replacing* **Physical Part BOM** Relationship Item replaces some of the **Quantity** property value of the *replaced* **Physical Part BOM** Relationship Item.

  You should select *one replacing* **Physical Part** Item and provide a **Quantity** property value for it less than the original one.

  The system will create an additional, new **Physical Part BOM** Relationship Item with the *remaining* original **Quantity** property value and correct the **Quantity** property value of the *replaced* **Physical Part BOM** Relationship Item for the removed quantity.

- Partial split or one-to-many replacement of some quantity: *multiple replacing* **Physical Part BOM** Relationship Items replace some of the **Quantity** property value of the *replaced* **Physical Part BOM** Relationship Item.

  You should select *multiple replacing* **Physical Part** Items and provide a **Quantity** property value for each of them. The sum of these **Quantity** property values should be less than the original one.

  The system will create an additional, new **Physical Part BOM** Relationship Item with the *remaining* original **Quantity** property value and correct the **Quantity** property value of the *replaced* **Physical Part BOM** Relationship Item for the removed quantity.

A split operation should not change a BOM slot quantity:

- In any split case, the sum of the **Quantity** property values of all *replacing* **Physical Part BOM** Relationship Items should be equal to the **Quantity** property value of the *replaced* **Physical Part BOM** Relationship Item.
- In the case of partial quantity replacement, additionally, the sum of the resulted **Quantity** property values of the *replaced* and *remaining* **Physical Part BOM** Relationship Items after a given operation should be equal to the original **Quantity** property value of the *replaced* **Physical Part BOM** Relationship Item before the given operation.

The system raises an error and does not allow to save a split operation if this operation changes the BOM slot quantity; see section 5.2.3.4 Validation of a Remove-and-Replace operation.

As the partial split is the most complicated among other subcases and includes them to some point, this section will describe its procedure. The partial split procedure and the above-mentioned discussion of all split cases will give you an understanding of how to perform any of them. Besides, the given procedure features notes prompting what to do at a specific step if you need a split subcase other than a partial split. If you want to continue the partial split, skip the notes.

To split partially some quantity of a current *non-serialized* child **Physical Part** Item in a Physical Part BOM structure:

1. Open the Item view of a **Physical Part** Item which is currently a direct parent of the given child.
2. Click **Edit** on the parent **Physical Part** Item toolbar.
3. Go to the **BOM** accordion tab.

4. Right-click *one* row with the given current *non-serialized* child **Physical Part** Item to be *replaced* and click **Remove-and-Replace**. This row is a *replaced* (old) **Physical Part BOM** Relationship Item. Its **Quantity** property has an original value to be partially replaced by the same or other parts.

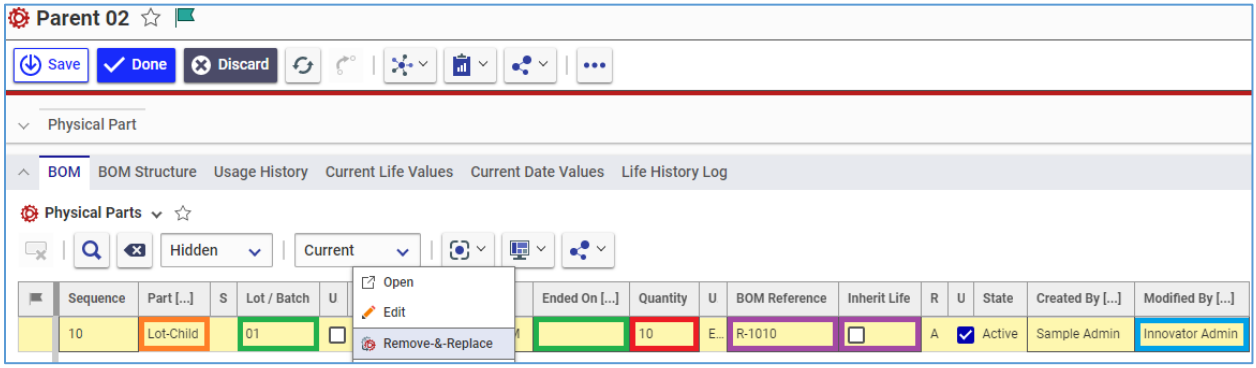


Figure 242.

The **Select Date** dialog pops up.

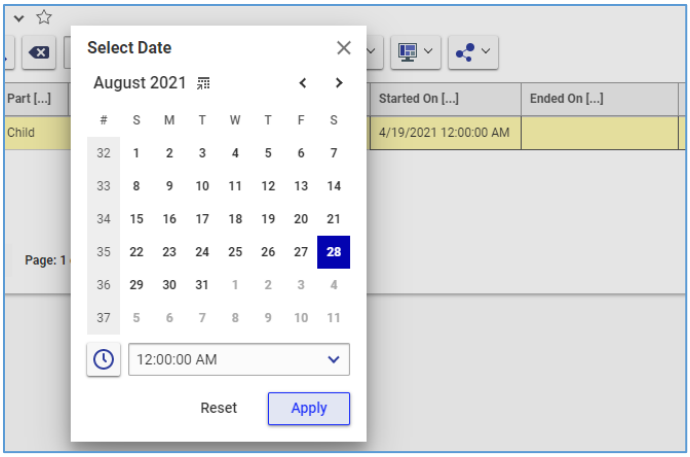


Figure 243.

5. In the **Select Date** dialog, specify remove-and-replace date and time and click **Apply**.

   The **Select Items – Physical Parts** dialog box pops up prepopulated with search criteria as follows:

   - **Part**: a **Part** Item of the *replaced* child **Physical Part** Item. You can change the **Part** criterion as you need.
   - **State**: **Active** and frozen. Only child **Physical Part** Items in the **Active** State can be added to a parent **Physical Part** Item in the **Active** State.

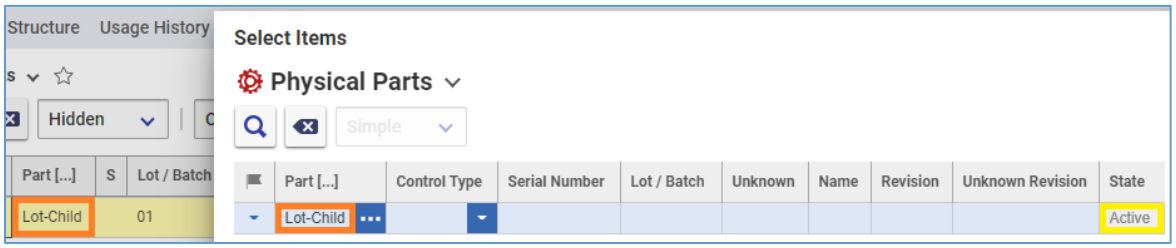


Figure 244.

6. In the **Select Items – Physical Parts** dialog, search for and select a *replacing* child **Physical Part** Item of the same Control Type as the *replaced* one using the standard search procedure.

   If you are replacing the child **Physical Part** Item with itself, the **Confirm** dialog box pops up. Otherwise, no dialog box pops up.

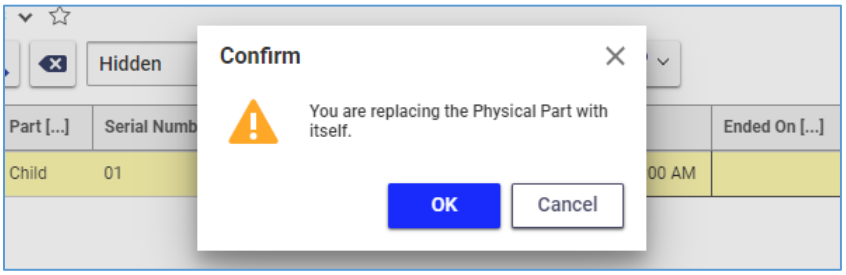


Figure 245.

Note: An error is raised if you try to add a parent (source) **Physical Part** Item to its BOM.

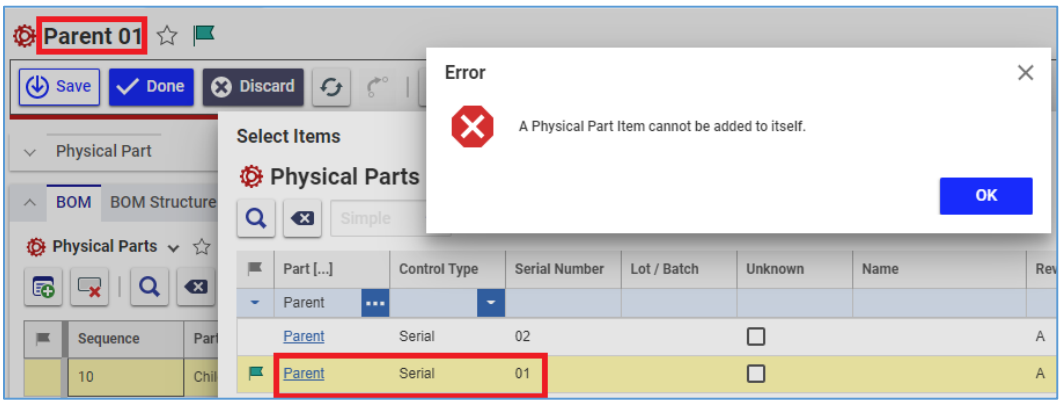


Figure 246.

7. Click **OK** in the **Confirm** dialog box.

Note: You should take step 7 only when you are replacing the child **Physical Part** Item with itself. If replacing with another **Physical Part** Item, the system behaves as below-described right after step 6.

The selected *replacing* child **Physical Part** Item populates the Grid as a new (*replacing*) **Physical Part BOM** Item with its properties set as follows:

- **Sequence**, **Quantity**, and **Inherit Life**: the appropriate default values for a newly created **Physical Part BOM** Item.
- **Started On**, **Ended On**, and **BOM Reference**: empty.

The **Replacing Item Quantity** dialog pops up prepopulated with the original **Quantity** property value of the *replaced* (old) **Physical Part BOM** Relationship Item.

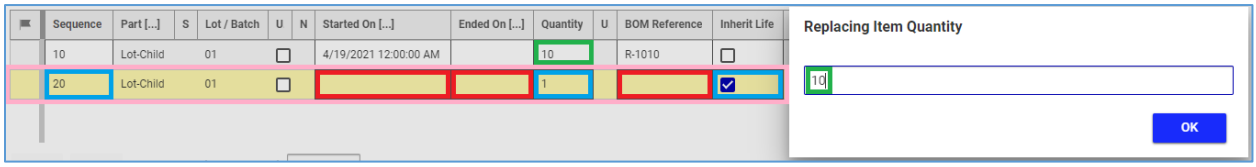


Figure 247.

8. In the **Replacing Item Quantity** dialog, correct the ***prepopulated number*** as you need and click **OK**.

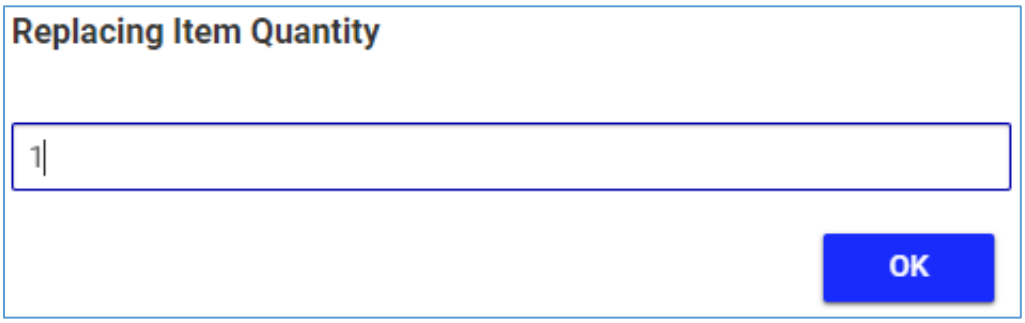


Figure 248.

Note: In the case of full quantity one-to-one replacement, do not change the ***prepopulated number*** and click **OK** in the **Replacing Item Quantity** dialog. This dialog disappears, and no other dialog pops up. The system sets the **Sequence**, **BOM Reference**, and **Inherit Life** properties of the new (*replacing*) **Physical Part BOM** Item to the values copied from the old (*replaced*) Item. The **Started On** property of the new Item and the **Ended On** property of the old Item get the value specified in the **Select Date** dialog at step 5. The **Quantity** property of the new Item gets the value given in the **Replacing Item Quantity** dialog at step 8. The **Quantity** values of the old and new Items are equal. To continue the full one-to-one replacement, jump to step13.

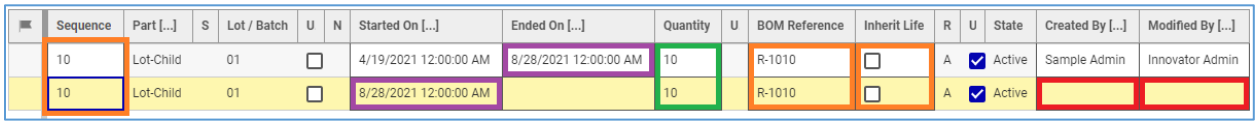


Figure 249.

The **Replacing Item Quantity** dialog disappears.

The **Quantity** property of the new (*replacing*) **Physical Part BOM** Item gets value given in the **Replacing Item Quantity** dialog at step 8.

The **Select Items – Physical Parts** dialog box pops.

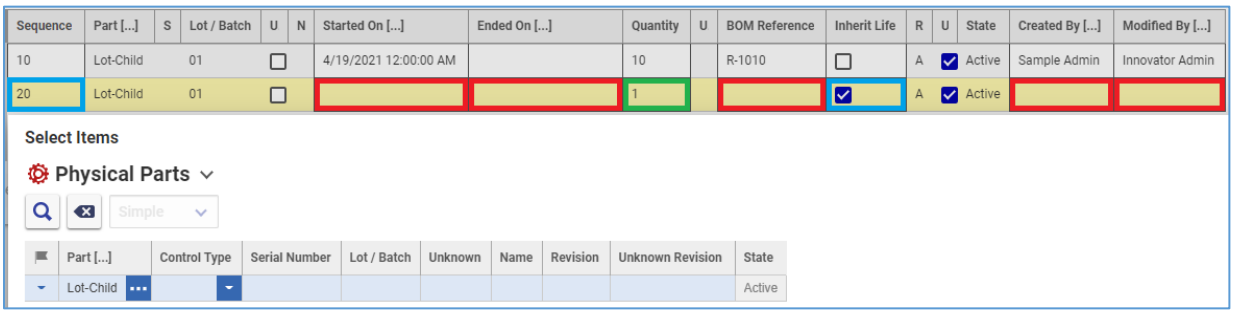


Figure 250.

Note: In the case of partial quantity one-to-one replacement, close the **Select Items – Physical Parts** dialog. No other dialog pops up. The system creates additional new **Physical Part BOM** Item for the *remaining* quantity and sets the **Sequence**, **BOM Reference**, and **Inherit Life** properties of both new (*replacing* and *remaining*) Items to the values copied from the old (*replaced*) Item. The **Started On** property of the *replacing* Item and the **Ended On** property of the *replaced* Item get the value specified in the **Select Date** dialog at step 5. The **Started On** property of the *remaining*

Item gets the value copied from the *replaced* Item. The **Quantity** properties of the *replacing* and *replaced* Item get the value given in the **Replacing Item Quantity** dialog at step 8. The **Quantity** property of the *remaining* Item gets value as the *replacing* **Quantity** value deducted from the *original* one. The sum of the *replacing* and *remaining* **Quantity** values equal the *original* one. To continue the partial quantity one-to-one replacement, jump to step 13.

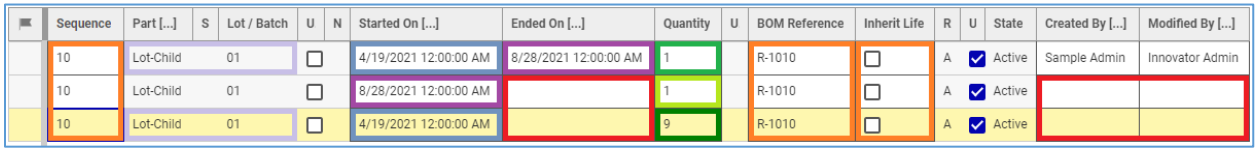


Figure 251.

9. In the **Select Items – Physical Parts** dialog, search for and select another *replacing* child **Physical Part** Item of the same *non-serialized* Control Type as the *replaced* one using the standard search procedure.

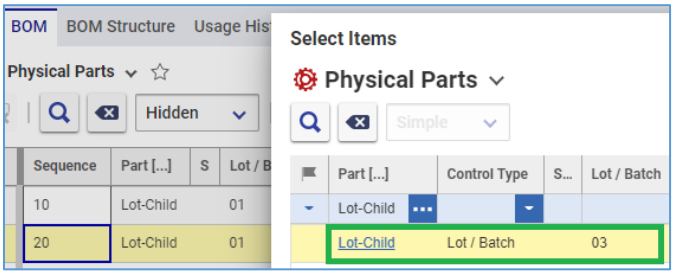


Figure 252.

Another selected *replacing* child **Physical Part** Item populates the grid as another new (*replacing*) **Physical Part BOM** Item with its properties set as described at step 7.

The **Replacing Item Quantity** dialog pops up prepopulated with the **Quantity** property value of the *replaced* (old) **Physical Part BOM** Relationship Item that is not replaced yet. For example, if the *original* **Quantity** property value is **10**, and the already added to the grid *replacing* child has **Quantity** property value as **1**, the **Replacing Item Quantity** dialog will be prepopulated with **9** (**10** – **1** = **9**). If the second *replacing* child has **3**, the dialog will be prepopulated for the third *replacing* child with **6** (**10** – **1** – **3** = **6**).

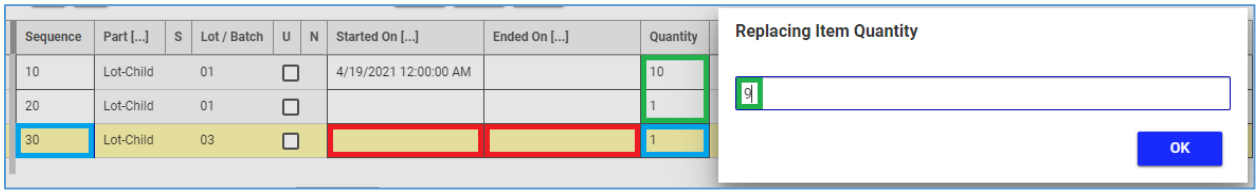


Figure 253.

10. In the **Replacing Item Quantity** dialog, correct the ***prepopulated number*** as you need and click **OK**.

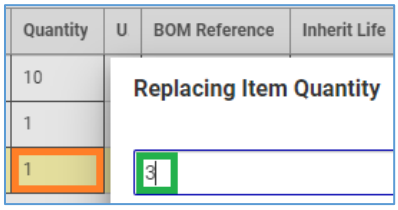
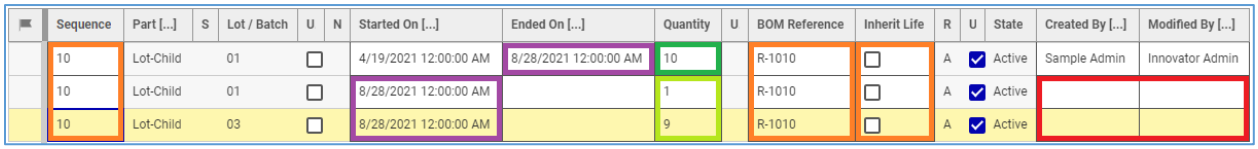


Figure 254.

Note: In the case of full quantity one-to-many replacement (full split), do not change the ***prepopulated number*** and click **OK** in the **Replacing Item Quantity** dialog. This dialog disappears, and no other dialog pops up. The system sets the **Sequence**, **BOM Reference**, and **Inherit Life** properties of all new (*replacing*) **Physical Part BOM** Items to the values copied from the old (*replaced*) Item. The **Started On** properties of all new Items and the **Ended On** property of the old Item get the value specified in the **Select Date** dialog at step 5. The **Quantity** properties of all new Items get values given in the **Replacing Item Quantity** dialog at appropriate steps. The sum of the **Quantity** values of the new Items equals the original one. To continue the full split, jump to step 13.



Sequence	Part [...]	S	Lot / Batch	U	N	Started On [...]	Ended On [...]	Quantity	U	BOM Reference	Inherit Life	R	U	State	Created By [...]	Modified By [...]
10	Lot-Child		01			4/19/2021 12:00:00 AM	8/28/2021 12:00:00 AM	10		R-1010		A		Active	Sample Admin	Innovator Admin
10	Lot-Child		01			8/28/2021 12:00:00 AM		1		R-1010		A		Active		
10	Lot-Child		03			8/28/2021 12:00:00 AM		9		R-1010		A		Active		

Figure 255.

The **Replacing Item Quantity** dialog disappears.

The **Quantity** property of another new (*replacing*) **Physical Part BOM** Item gets value given in the **Replacing Item Quantity** dialog at step 10.

The **Select Items – Physical Parts** dialog box pops.

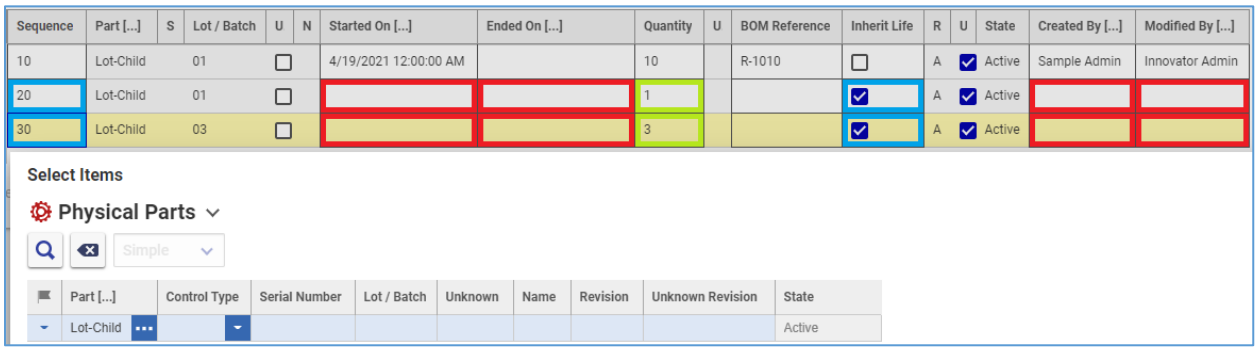
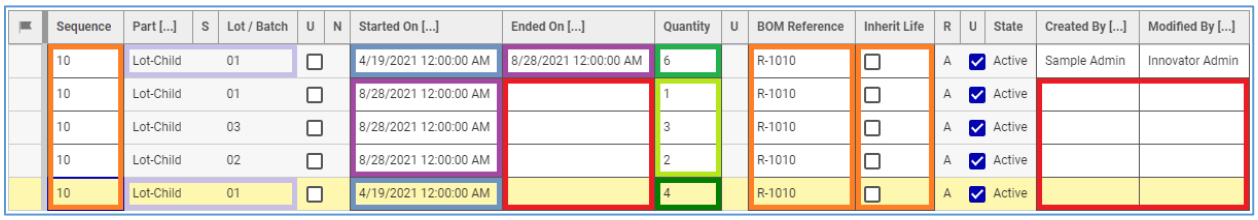


Figure 256.

11. Repeat steps 9 and 10 as many times as necessary to specify all *replacing* **Physical Part** Items.

Note: If you close the **Replacing Item Quantity** dialog, the system understands it as the full quantity one-to-many replacement full split.

12. Close the **Select Items – Physical Parts** dialog.



Sequence	Part [...]	S	Lot / Batch	U	N	Started On [...]	Ended On [...]	Quantity	U	BOM Reference	Inherit Life	R	U	State	Created By [...]	Modified By [...]
10	Lot-Child		01			4/19/2021 12:00:00 AM	8/28/2021 12:00:00 AM	6		R-1010		A		Active	Sample Admin	Innovator Admin
10	Lot-Child		01			8/28/2021 12:00:00 AM		1		R-1010		A		Active		
10	Lot-Child		03			8/28/2021 12:00:00 AM		3		R-1010		A		Active		
10	Lot-Child		02			8/28/2021 12:00:00 AM		2		R-1010		A		Active		
10	Lot-Child		01			4/19/2021 12:00:00 AM		4		R-1010		A		Active		

Figure 257.

No other dialog pops up.

The system creates additional new **Physical Part BOM** Relationship Item for the *remaining* quantity and sets the properties of the old and new Items as follows:

- The old *replaced* **Physical Part BOM** Relationship Item:
  - **Ended On**: the value specified in the **Select Date** dialog at step 5.
  - **Quantity**: the sum of the **Quantity** property values of all *replacing* **Physical Part BOM** Relationship Items.
- The new *replacing* **Physical Part BOM** Relationship Items:
  - **Sequence**, **BOM Reference**, and **Inherit Life**: the values copied from the matching properties of the old (*replaced*) **Physical Part BOM** Relationship Item.
  - **Started On**: the value specified in the **Select Date** dialog at step 5.
  - **Ended On**: empty.
  - **Quantity**: a value entered in the **Replacing Item Quantity** dialog at step appropriate for a given *replacing* **Physical Part** Item.
  - **Created By** and **Modify By**: empty.
- The new *remaining* **Physical Part BOM** Relationship Item:
  - **Sequence**, **BOM Reference**, and **Inherit Life**: the values copied from the matching properties of the old (*replaced*) **Physical Part BOM** Relationship Item.
  - **Started On**: the value copied from the **Started On** property of the old (*replaced*) **Physical Part BOM** Relationship Item.
  - **Ended On**: empty.
  - **Quantity**: a value that equals the *original* **Quantity** property of the old (*replaced*) **Physical Part BOM** Relationship Item before the partial split minus the sum of the **Quantity** property values of all *replacing* **Physical Part BOM** Relationship Items.
  - **Created By** and **Modify By**: empty.

13. If the *replaced* component was removed from the assembly at a date or time different than specified in the **Select Date** dialog at step 5, correct the **Ended On** property value of the *replaced* (old) **Physical Part BOM** Item by typing or using the **Date dialog**. This value must be the same or earlier than the **Started On** property value of any *replacing* (new) **Physical Part BOM** Item.

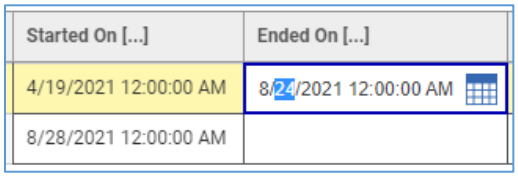


Figure 258.

14. If a *replacing* component was installed into the assembly at a date or time different than specified in the **Select Date** dialog at step 5, correct the **Started On** property value of a corresponding *replacing* (new) **Physical Part BOM** Item by typing or using the **Date dialog**. This value must be the same or later than the **Ended On** property value of the *replaced* (old) **Physical Part BOM** Item.

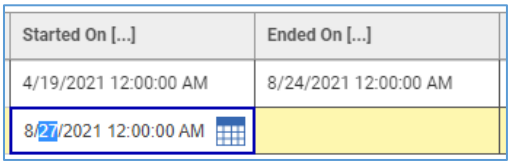


Figure 259.

15. Click either **Save** or **Done** on the parent **Physical Part** Item toolbar.

    The *replaced* (old) **Physical Part** Item becomes a historical child. The *replaced* (old) **Physical Part BOM** Item still exists and keeps installation data on the *replaced* (old) child.

    All *replacing* (new) child **Physical Part** Items are current children in the BOM. The *replacing* (new) **Physical Part BOM** Items are created to keep BOM installation data on the *replacing* (new) children.

    The *remaining* (new) child **Physical Part** Item is a current child in the BOM. The *remaining* (new) **Physical Part BOM** Item is created to keep BOM installation data on the *remaining* quantity of the old child.

    Your Identity automatically populates the **Modified By** property of the *replaced* (old) **Physical Part BOM** Item and the **Created By** and **Modified By** properties of the *replacing* and *remaining* (new) **Physical Part BOM** Items.

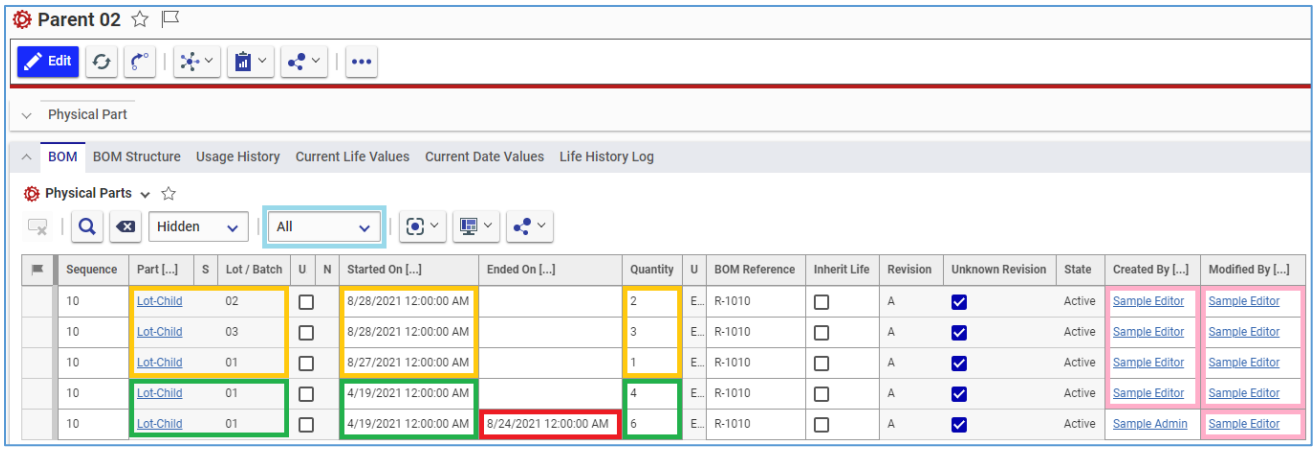


Figure 260.

Upon saving the parent **Physical Part** Item, the server side validates your updates against the following constraints:

- General for **Physical Part BOM** Relationship Items; see section *5.2.2.5 Physical Part BOM validation*.
- Specific for the *one-to-one replacement* R&R case; see section 5.2.3.4 Validation of a Remove-and-Replace operation.

5.2.3.3 Merge Remove-and-Replace

The *merge* R&R case represents an operation when *one non-serialized* part replaces *multiple* similar *non-serialized* parts in one BOM slot. Thus, this case is available only for *multiple non-serialized replaced* **Physical Part BOM** Relationship Items with the same combination of the **Sequence**, **BOM Reference**, and **Inherit Life** property values. If you had selected *replaced* **Physical Part BOM** Relationship Items located in different BOM slots, the system will allow you to perform a merge operation but not save it: you will be required to discard all changes. See section 5.2.3.4 Validation of a Remove-and-Replace operation.

The *replacing* and all *replaced* **Physical Part** Items should be of the same *non-serialized* Control Type: either **Lot / Batch** or **No Control**. If you had selected **Physical Part** Items with different Control Types, the system will allow you to perform a merge operation but not save it: you will be required to discard all changes. See section 5.2.3.4 Validation of a Remove-and-Replace operation.

A *serial-controlled* **Physical Part** Item is not allowed for a merge operation. If you had selected a *replaced* or *replacing serial-controlled* **Physical Part** Item, the system will allow you to perform a merge operation but not save it: you will be required to discard all changes. See section 5.2.3.4 Validation of a

Remove-and-Replace operation. The *one-to-one replacement* is the only R&R operation allowed over the *serial-controlled* **Physical Part** Items.

During merging, the system automatically populates the **Quantity** property of a *replacing* **Physical Part BOM** Relationship Item with the sum of the **Quantity** property values of all *replaced* **Physical Part BOM** Relationship Items. A merge operation should not change a BOM slot quantity. The system raises an error and does not allow to save a merge operation if this operation changes the BOM slot quantity: the **Quantity** property value of a *replacing* **Physical Part BOM** Relationship Item is not equal to the sum of the **Quantity** property values of all *replaced* **Physical Part BOM** Relationship Items; see section 5.2.3.4 Validation of a Remove-and-Replace operation.

To merge quantities of some *non-serialized* child **Physical Part** Items within the same BOM slot in a Physical Part BOM structure:

1. Open the Item view of a **Physical Part** Item which is currently a direct parent of the given children.
2. Click **Edit** on the parent **Physical Part** Item toolbar.
3. Go to the **BOM** accordion tab.
4. Select all rows with the given current *non-serialized* child **Physical Part** Items to be *replaced*. You should select *multiple* rows. Each such row is a *replaced* (old) **Physical Part BOM** Relationship Item.
5. Right-click any of the selected *replaced* rows and click **Remove-and-Replace**.

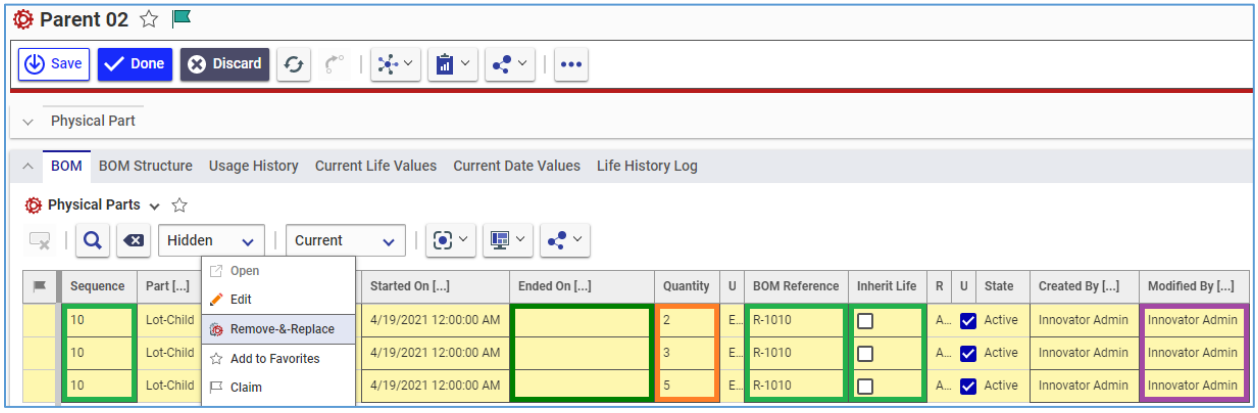


Figure 261.

The **Select Date** dialog pops up.

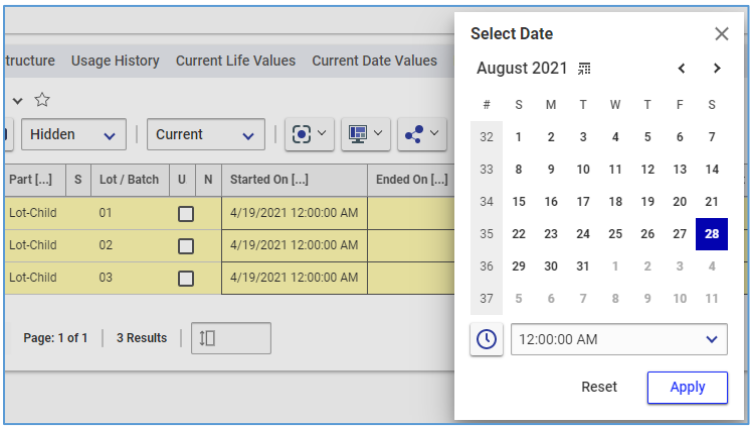


Figure 262.

6. In the **Select Date** dialog, specify remove-and-replace date and time and click **Apply**.

   The **Select Items – Physical Parts** dialog box pops up prepopulated with search criteria as follows:

   - **State**: **Active** and frozen. Only child **Physical Part** Items in the **Active** State can be added to a parent **Physical Part** Item in the **Active** State.

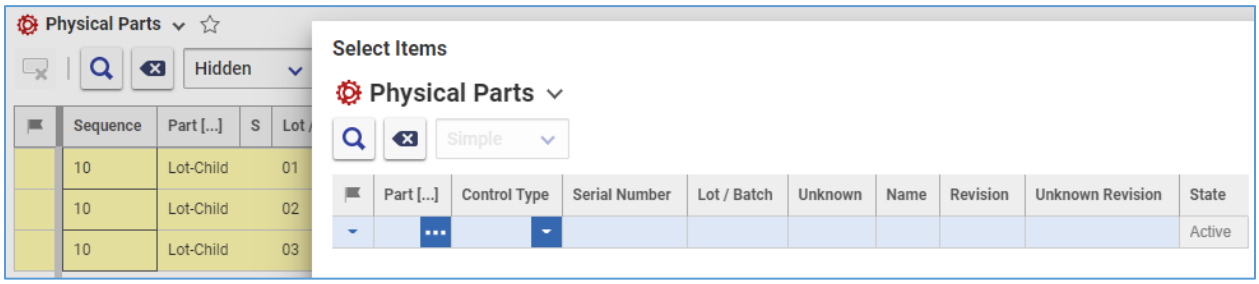


Figure 263.

7. In the **Select Items – Physical Parts** dialog, search for and select a *replacing non-serialized* child **Physical Part** Item using the standard search procedure.

Note: You can select only one **Physical Part** Item in the **Select Items – Physical Parts** dialog when merging.

If you are replacing any of the *replaced* child **Physical Part** Items with itself, the **Confirm** dialog box pops up. Otherwise, no dialog box pops up.

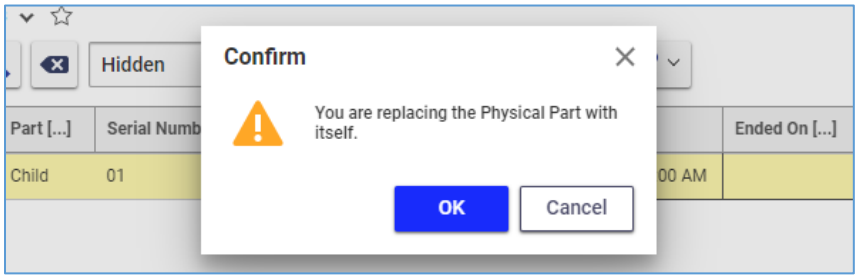


Figure 264.

Note: An error is raised if you try to add a parent (source) **Physical Part** Item to its BOM.

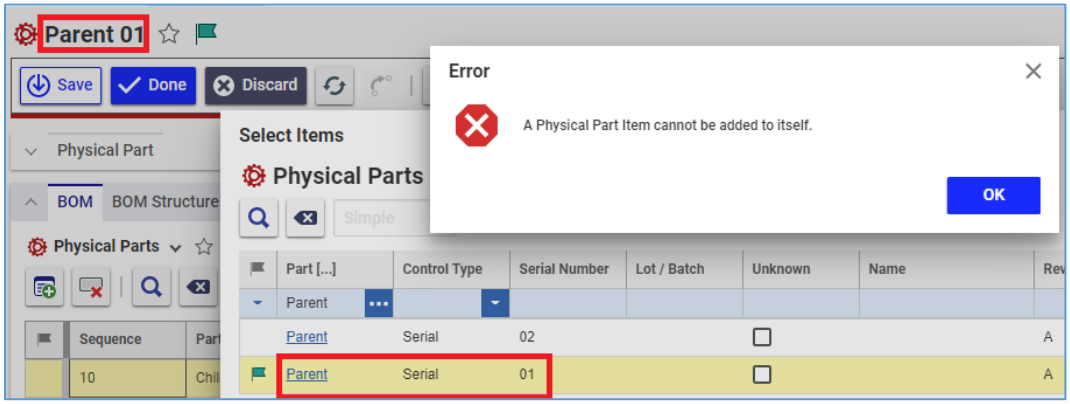


Figure 265.

8. Click **OK** in the **Confirm** dialog box.

Note: You should take step 7 only when you are replacing any child **Physical Part** Item with itself. If replacing with another **Physical Part** Item, the system behaves as below-described right after step 6.

The date and time value you had specified at step 6 populates the **Ended On** property of the *replaced* (old) **Physical Part BOM** Item.

The selected *replacing* child **Physical Part** Items populate the grid as new (*replacing*) **Physical Part BOM** Items with their properties set as follows:

- **Sequence**, **BOM Reference**, and **Inherit Life**: the same values as the *replaced* (old) **Physical Part BOM** Items have.
- **Started On**: equal to the **Ended On** property value of the *replaced* (old) **Physical Part BOM** Items.
- **Quantity**: the sum of the **Quantity** property values of all *replaced* **Physical Part BOM** Relationship Items.

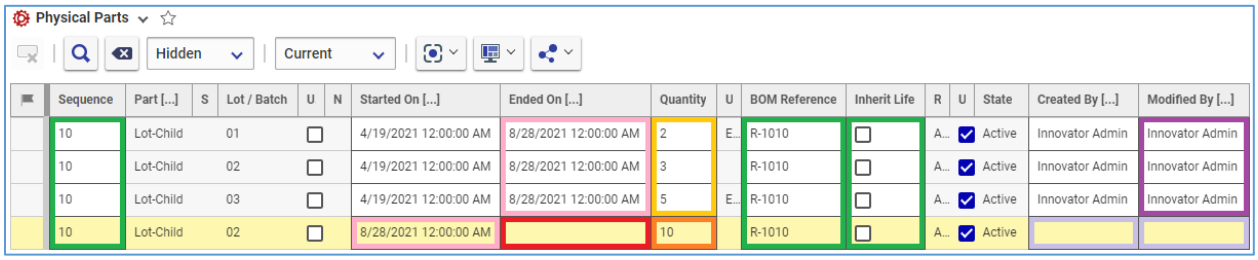


Figure 266.

9. If a *replaced* component was removed from the assembly at a date or time different than specified in the **Select Date** dialog at step 6, correct the **Ended On** property value of this *replaced* (old) **Physical Part BOM** Item by typing or using the **Date dialog**. This value must be the same or earlier than the **Started On** property value of the *replacing* (new) **Physical Part BOM** Item.

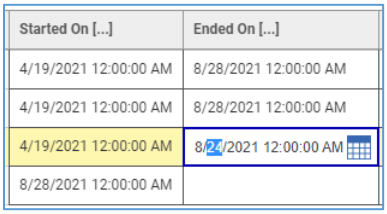


Figure 267.

10. If the *replacing* component was installed into the assembly at a date or time different than specified in the **Select Date** dialog at step 6, correct the **Started On** property value of the *replacing* (new) **Physical Part BOM** Item by typing or using the **Date dialog**. This value must be the same or later than the **Ended On** property value of any *replaced* (old) **Physical Part BOM** Item.

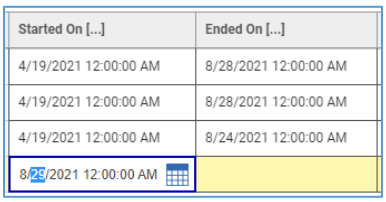


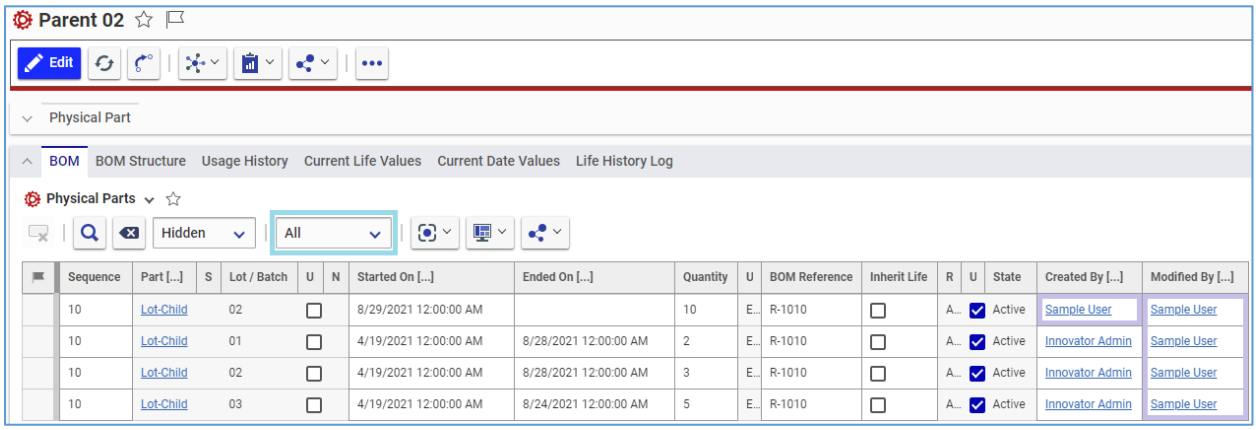
Figure 268.

11. Click either **Save** or **Done** on the parent **Physical Part** Item toolbar.

    The *replaced* (old) **Physical Part** Items become historical children. The *replaced* (old) **Physical Part BOM** Items still exist and keep installation data on the *replaced* (old) children.

    The *replacing* (new) child **Physical Part** Item is a current child in the BOM. The *replacing* (new) **Physical Part BOM** Item is created to keep BOM installation data on the *replacing* (new) child.

    Your Identity automatically populates the **Modified By** property of the *replaced* (old) **Physical Part BOM** Items and the **Created By** and **Modified By** properties of the *replacing* (new) **Physical Part BOM** Item.



Sequence	Part [...]	S	Lot / Batch	U	N	Started On [...]	Ended On [...]	Quantity	U	BOM Reference	Inherit Life	R	U	State	Created By [...]	Modified By [...]
10	Lot-Child		02			8/29/2021 12:00:00 AM		10	E...	R-1010		A...		Active	Sample User	Sample User
10	Lot-Child		01			4/19/2021 12:00:00 AM	8/28/2021 12:00:00 AM	2	E...	R-1010		A...		Active	Innovator Admin	Sample User
10	Lot-Child		02			4/19/2021 12:00:00 AM	8/28/2021 12:00:00 AM	3	E...	R-1010		A...		Active	Innovator Admin	Sample User
10	Lot-Child		03			4/19/2021 12:00:00 AM	8/24/2021 12:00:00 AM	5	E...	R-1010		A...		Active	Innovator Admin	Sample User

Figure 269.

Upon saving the parent **Physical Part** Item, the server side validates your updates against the following constraints:

- General for **Physical Part BOM** Relationship Items; see section *5.2.2.5 Physical Part BOM validation*.
- Specific for the *one-to-one replacement* R&R case; see section 5.2.3.4 Validation of a Remove-and-Replace operation.

5.2.3.4 Validation of a Remove-and-Replace operation

Once you click **Save** or **Done** on a parent (source) **Physical Part** Item toolbar after performing an R&R operation, the system validates its single-level Physical Part BOM structure against the following constraints:

- General for **Physical Part BOM** Relationship Items; see section *5.2.2.5 Physical Part BOM validation*.
- Specific for R&R operations; discussed below in this section.

The system raises an appropriate error if this R&R operation does not meet any of the constraints.

An R&R operation does not meet the R&R-specific constraints if this R&R operation either:

- Has a **Physical Part BOM** Item from a different BOM slot.

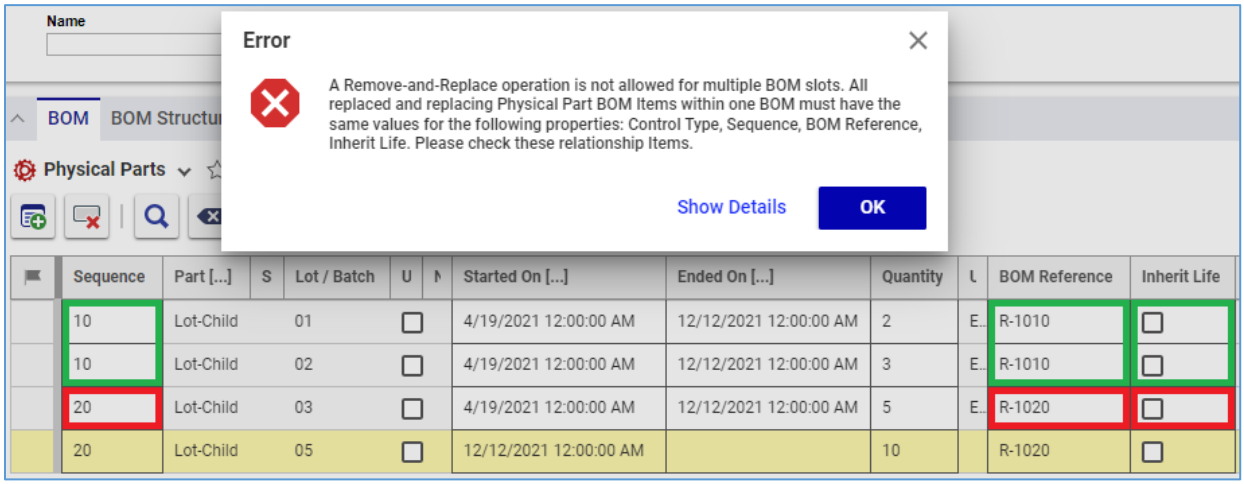


Figure 270.

- Has a historical *replaced* (old) **Physical Part BOM** Item with the **Ended On** property populated before the operation.

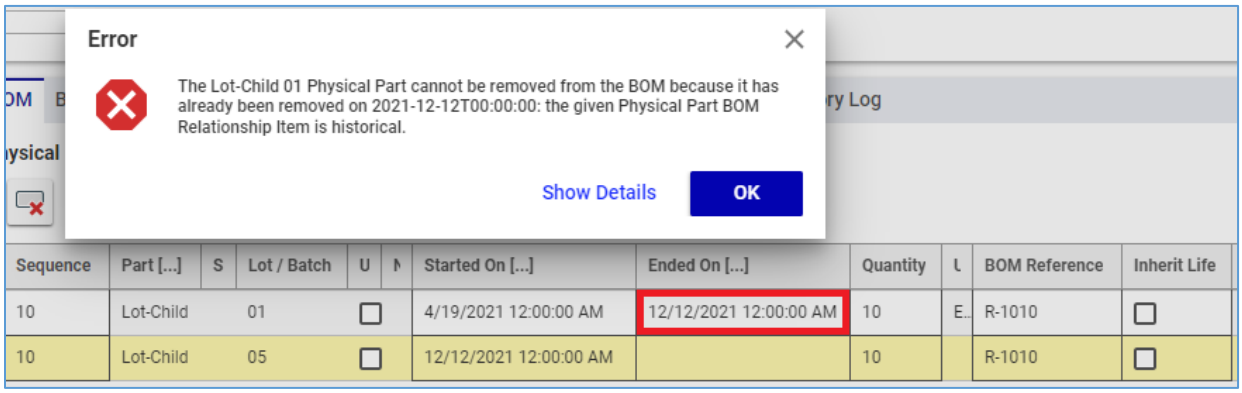


Figure 271.

- Includes **Physical Part** Items of different Control Types.

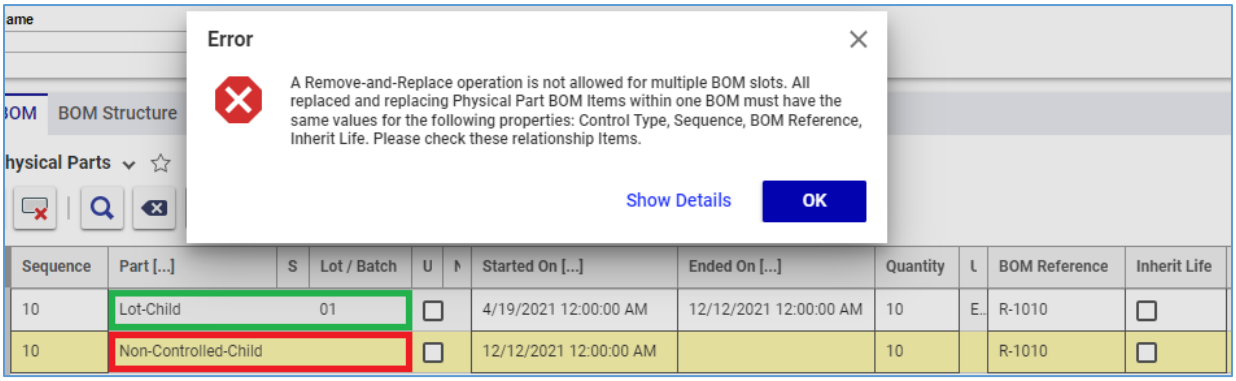


Figure 272.

- Is a merge operation for serial-controlled **Physical Part** Items.

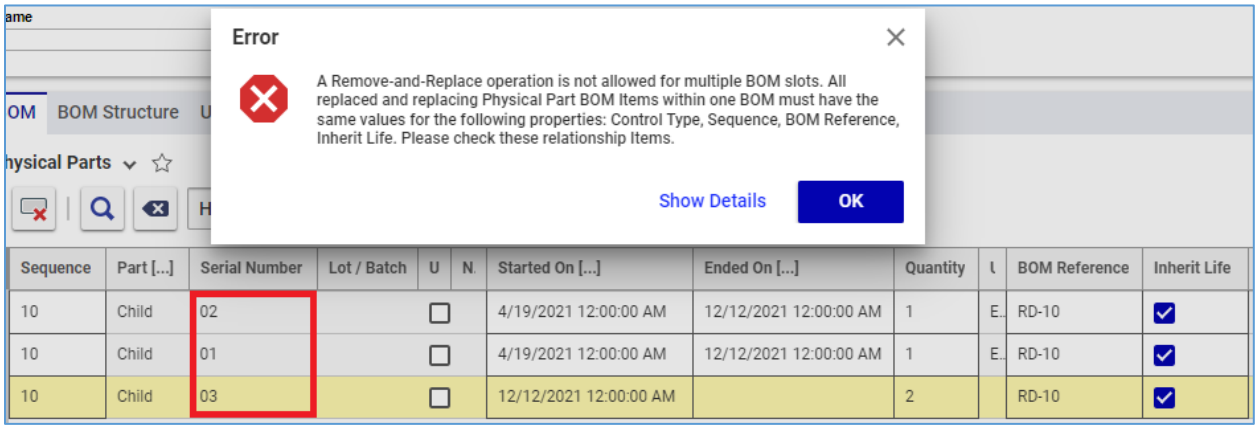


Figure 273.

- Includes a *replaced* (old) **Physical Part BOM** Item with the empty **Ended On** property after the operation.

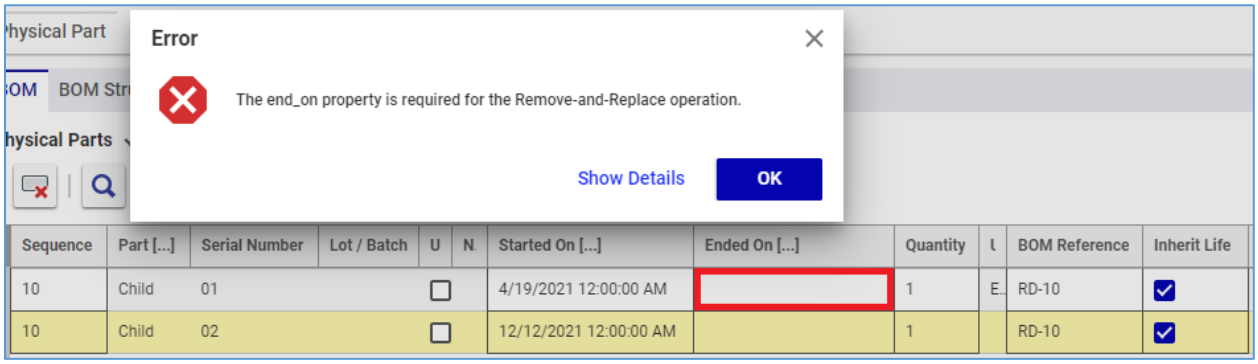


Figure 274.

- Includes a **Started On** property value of a *replacing* (new) **Physical Part BOM** Item that is less than an **Ended On** property value of a *replaced* (old) **Physical Part BOM** Item.

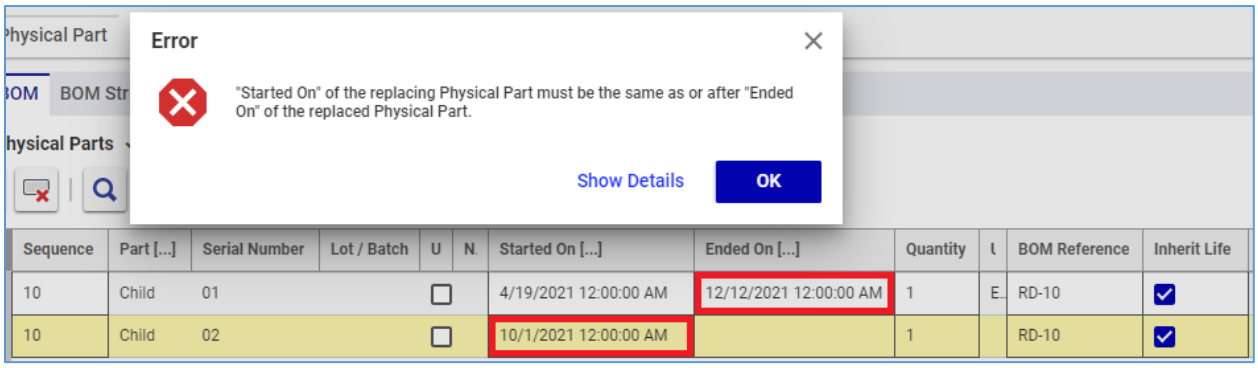


Figure 275.

- Changes the BOM slot quantity.

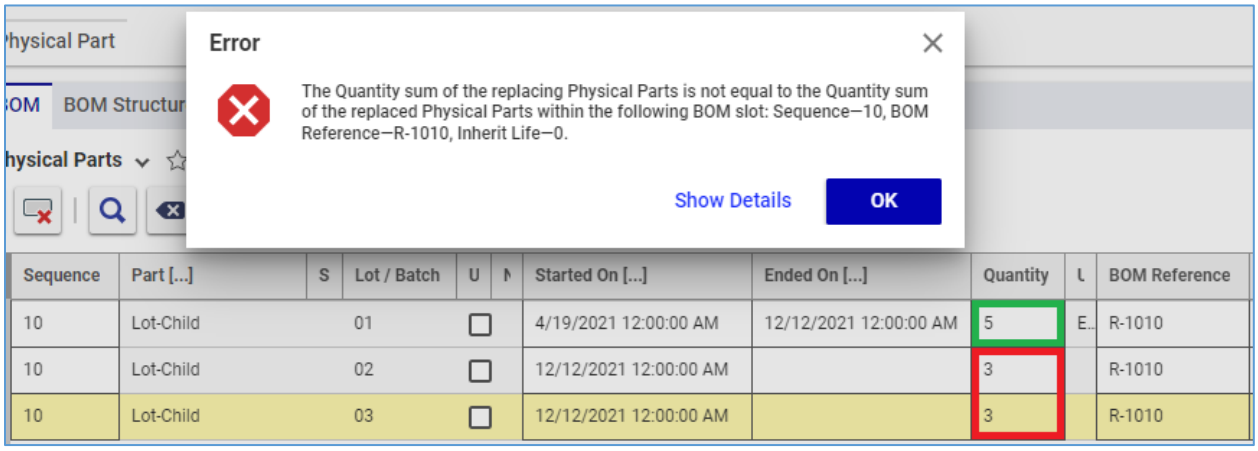


Figure 276.

- Is combined with another R&R operation for a single BOM slot.

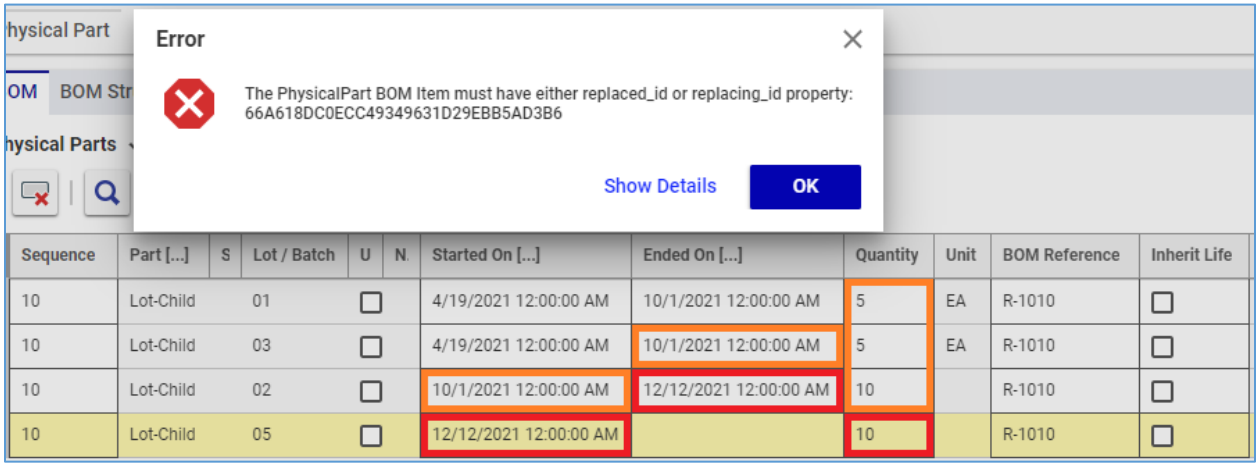


Figure 277.

- Is combined with a standard Aras Innovator operation.

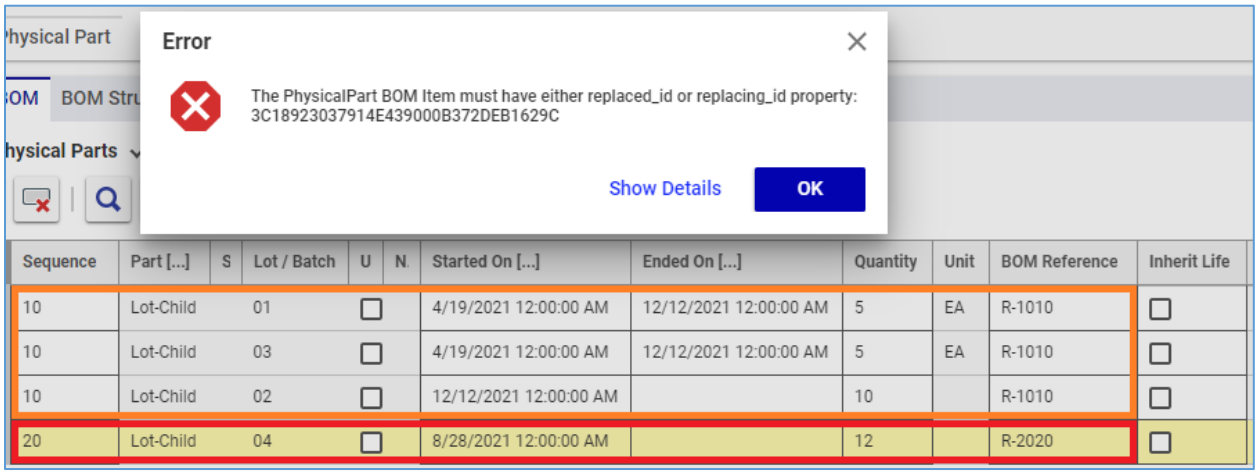


Figure 278.

5.3 Multi-Level Physical Part BOM structures

5.3.1 Understanding Multi-level Physical Part BOMs

A multi-level **Physical Part** BOM is a hierarchy of **Physical Part** Items that digitally represents a component hierarchy of an assembly as a tree of **Physical Part** Items where the assembly is the top-most parent **Physical Part** Item.

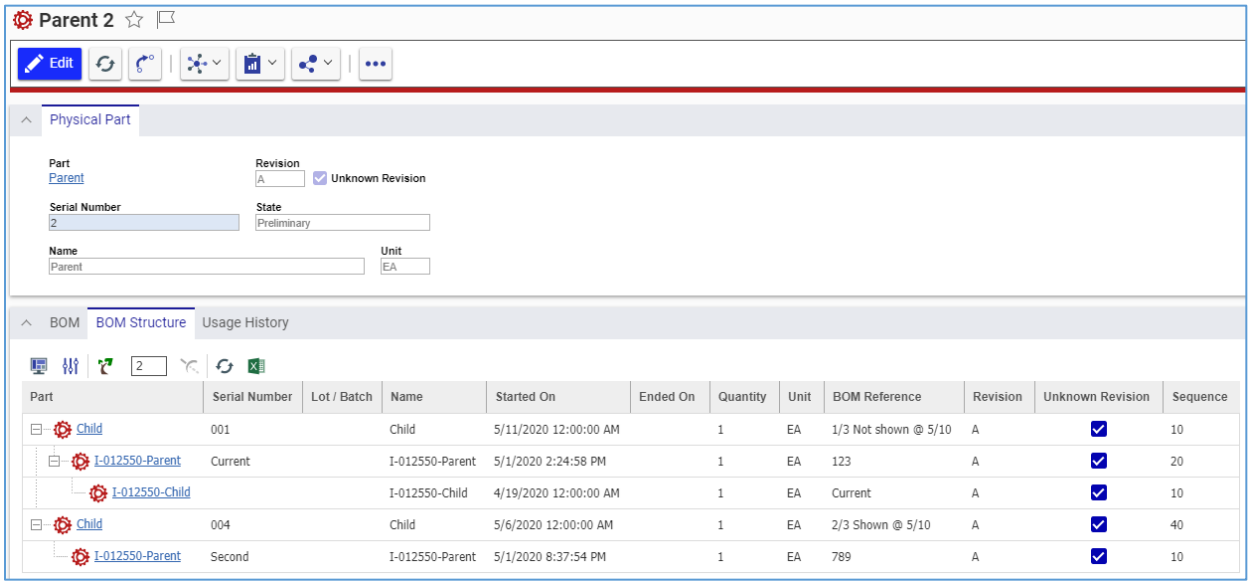


Figure 279.

The multi-level **Physical Part** BOM is displayed as a read-only Tree Grid View (TGV) of the **Physical Part BOM** Items on the **BOM Structure** accordion tab of the parent **Physical Part** Item view. The tree is shown recursively: the first level is the immediate children of the top-most parent, the second level is the children of the immediate children, and so on. A child with its own children is a separate tree branch.

A **Physical Part BOM** Item shows BOM installation data on a child **Physical Part** Item at a given level of a given branch. For example, the child **Started On** property displays when the child has been installed to its immediate parent, this parent **Started On**—when this parent has been installed to its immediate parent, and so on.

When first opened, the **BOM Structure** accordion tab shows a folded structure of the current **Physical Part BOM** Items.

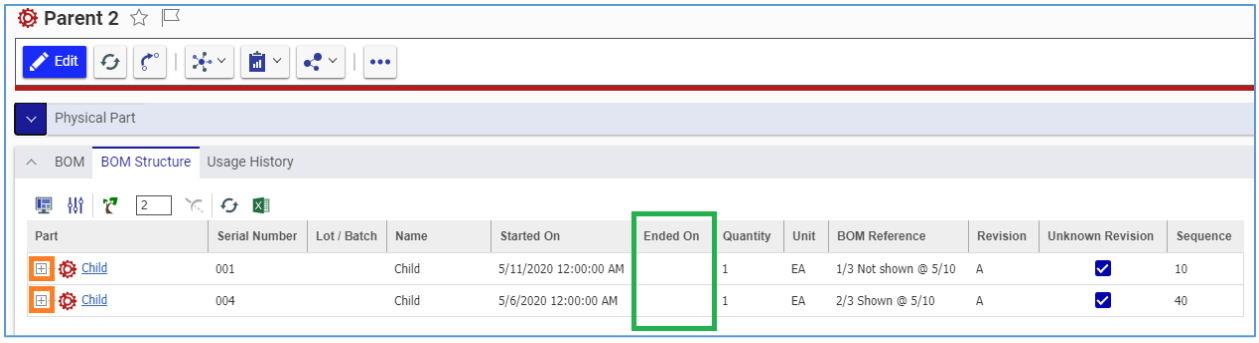


Figure 280.

The default **BOM Structure** Grid sorting is first by the **Sequence** values in ascending order (from the lowest to the highest number) within the **Physical Part BOM** Items of the same branch level and then by the **Started On** values in descending order (from the last to the first date) within the same **Sequence** value group.

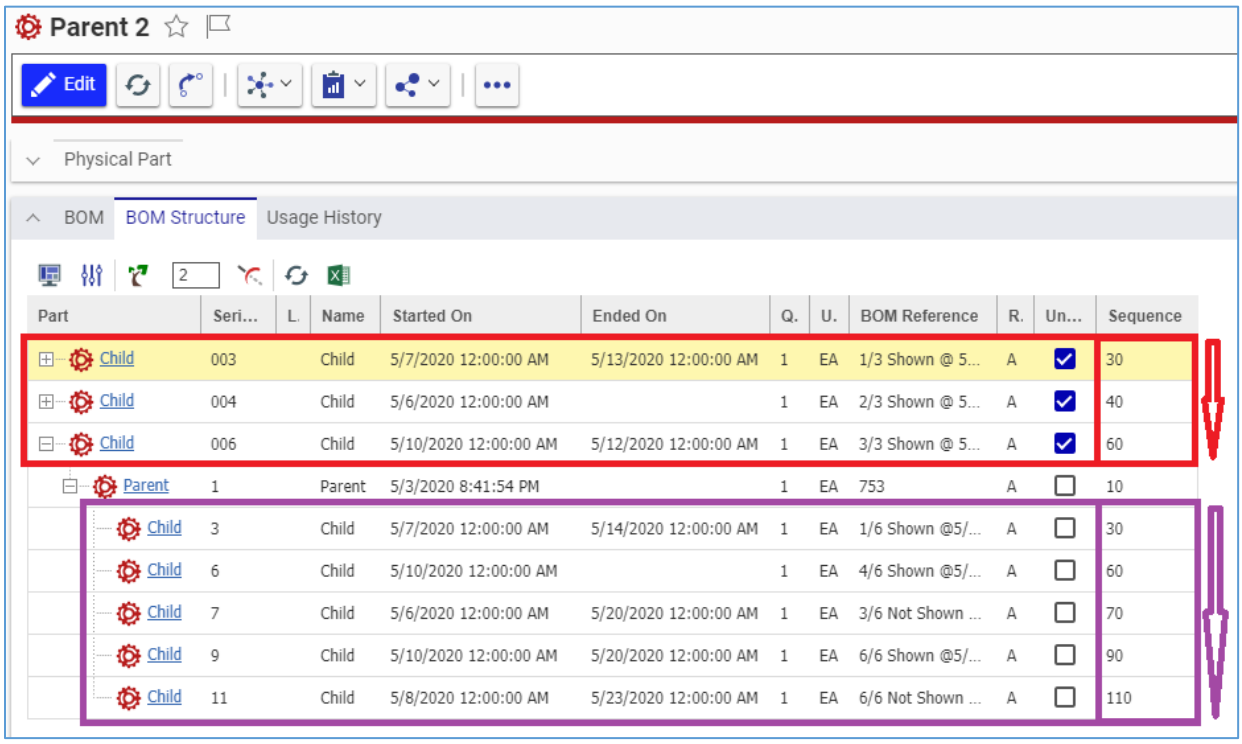


Figure 281.

The **BOM Structure** accordion tab has a TGV toolbar to manage and explore its TGV:

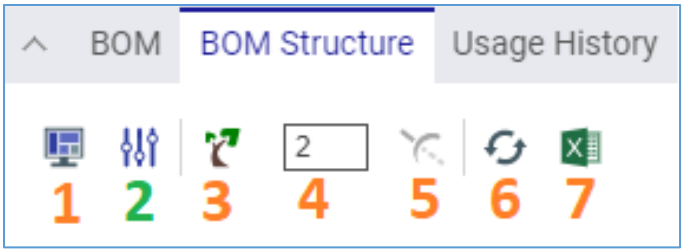


Figure 282.

1. **Display Settings**: a button to display the **Display Settings** dialog box for setting up Item TGV visibility.
2. **Modify Parameters**: a button to display the **Parameters** dialog box for filtering the multi-level **Physical Part** BOM structure against a point of time.
3. **Grow**: a button to expand the multi-level **Physical Part** BOM structure.
4. **Grow Depth**: a field to set up the level of the **Physical Part** BOM structure expansion.
5. **Trim**: a button to collapse the **Physical Part** BOM structure.
6. **Refresh**: a button to update the multi-level **Physical Part** BOM structure with the latest data.
7. **Export to Excel**: a button to save the multi-level **Physical Part** BOM structure as an Excel file.

This document discusses only the filtering feature (2).

To open a **Physical Part** Item view:

1. Right-click the **Physical Part** Item **row**.
2. Click **View**.

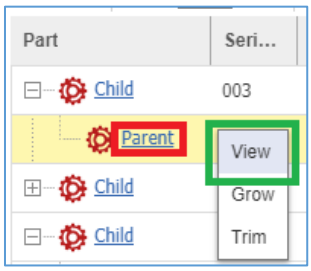


Figure 283.

A link in the **Part** cell opens an Item view of a related **Part** Item, not of the **Physical Part** Item.

5.3.2 Exploring a Multi-level Physical Part BOM Over a Time Point

Use the following procedure to filter a multi-level **Physical Part** BOM against a given point of time:

1. Click the **Modify Parameters** button. The **Parameters** dialog box pops up

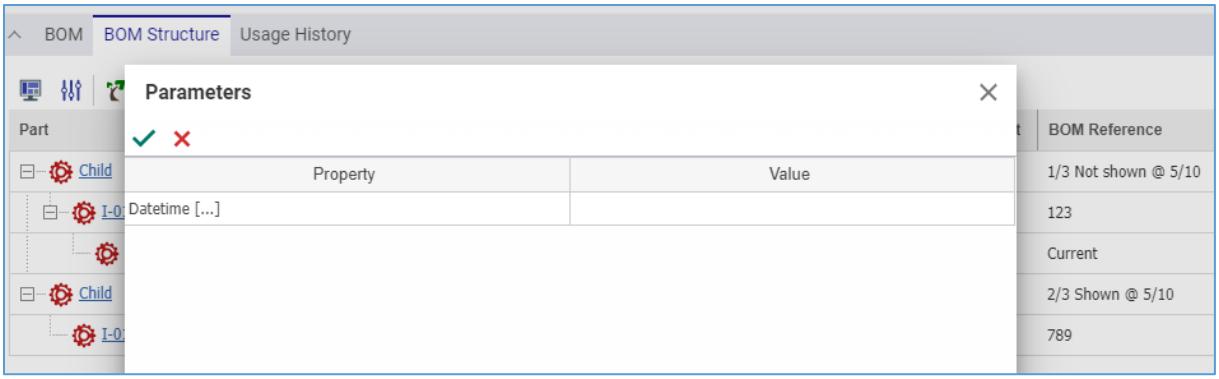


Figure 284.

2. Double-click the **Value** cell. The **Date dialog** pops up.

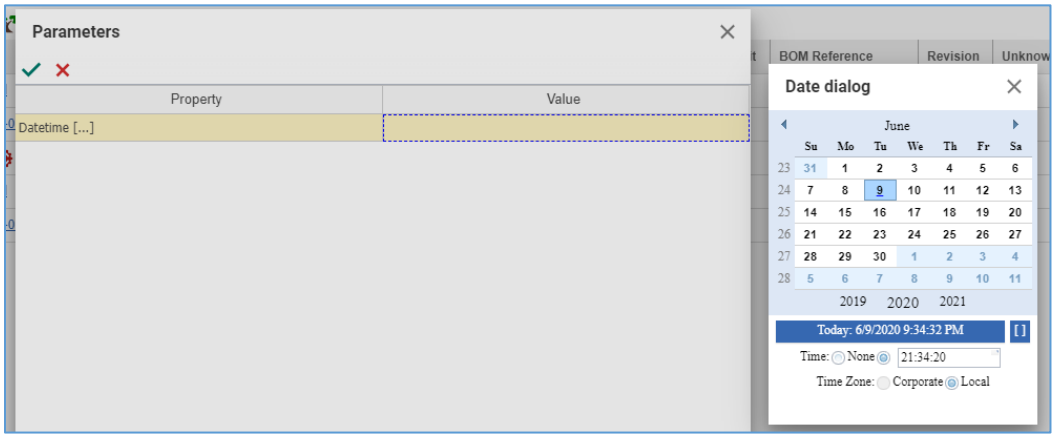


Figure 285.

3. Specify the time point as a date and time value. The **time point** populates the **Value** cell.

Note: If the **Value** cell has no given **time point**, repeat steps 2 and 3.

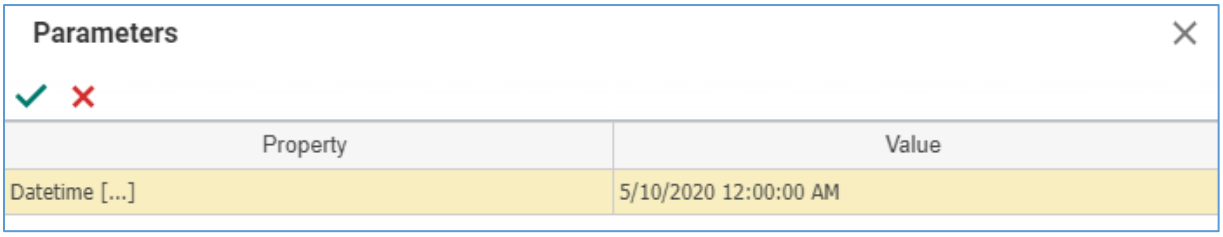


Figure 286.

4. Click **Apply**.

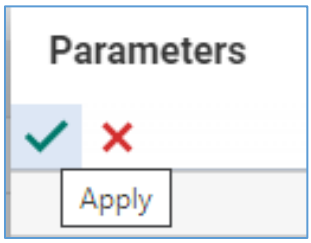


Figure 287.

The **BOM Structure** TGV displays only **Physical Part BOM** Items that are current at the given time point.

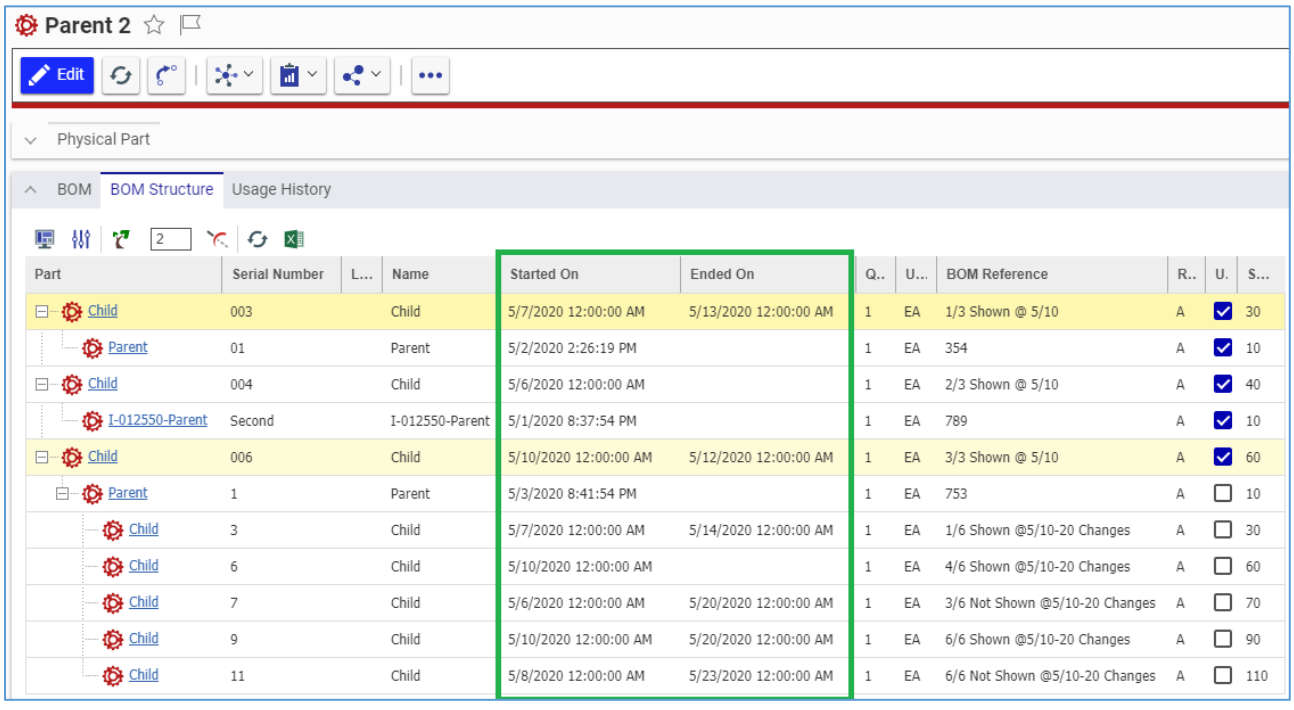


Figure 288.

To be evaluated as current at a time point, a **Physical Part BOM** Item of the child must have its property values as follows:

- **Started On** less than or equal to the time point.

AND

- **Ended On** greater than the time point or blank.

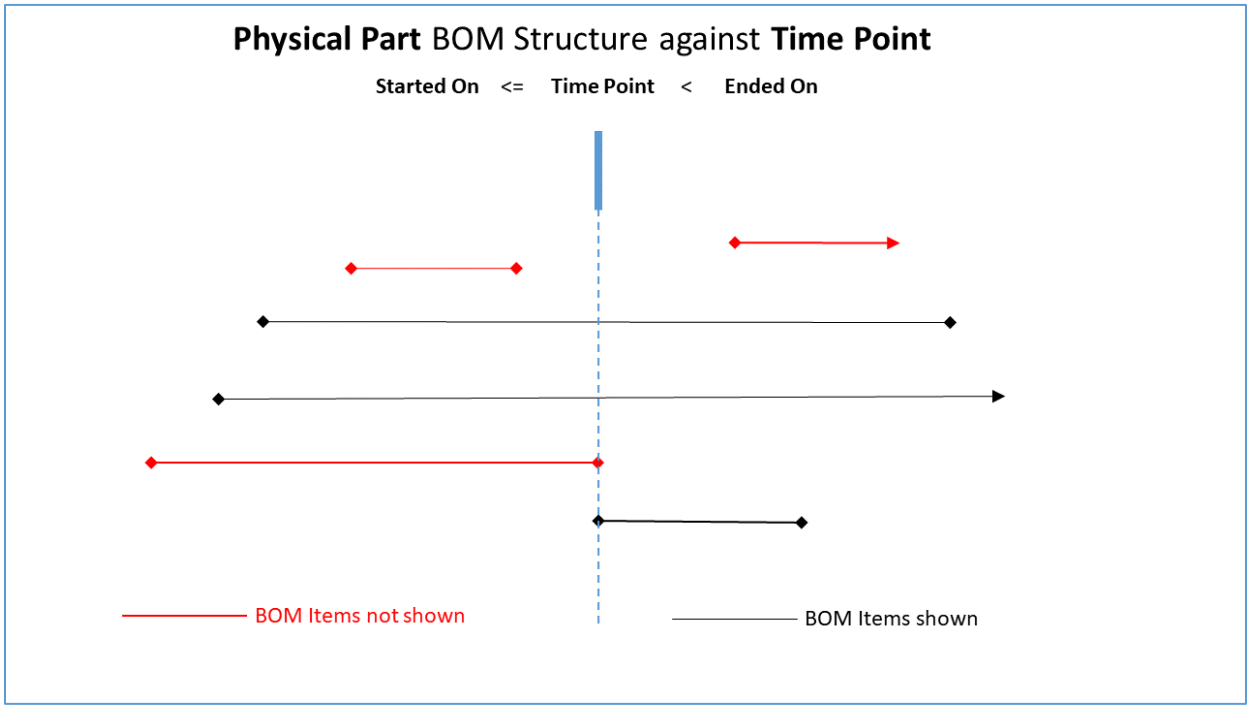


Figure 289.

5.4 Physical Part Usage History

5.4.1 Understanding Physical Part Usage History

A real-world component can be reused in the same or different assemblies over its life. Consequently, a representing **Physical Part** Item can be used as a child in the same or different **Physical Part** BOMs.

The **Usage History** accordion tab of a **Physical Part** Item view displays the immediate parents of this **Physical Part** Item as a read-only grid. The grid items are **Physical Part BOM** Items that show properties of the given child-parent Relationships at the corresponding **Physical Part** BOMs. For example, the **Started On** property displays when the child has been installed to the given parent.

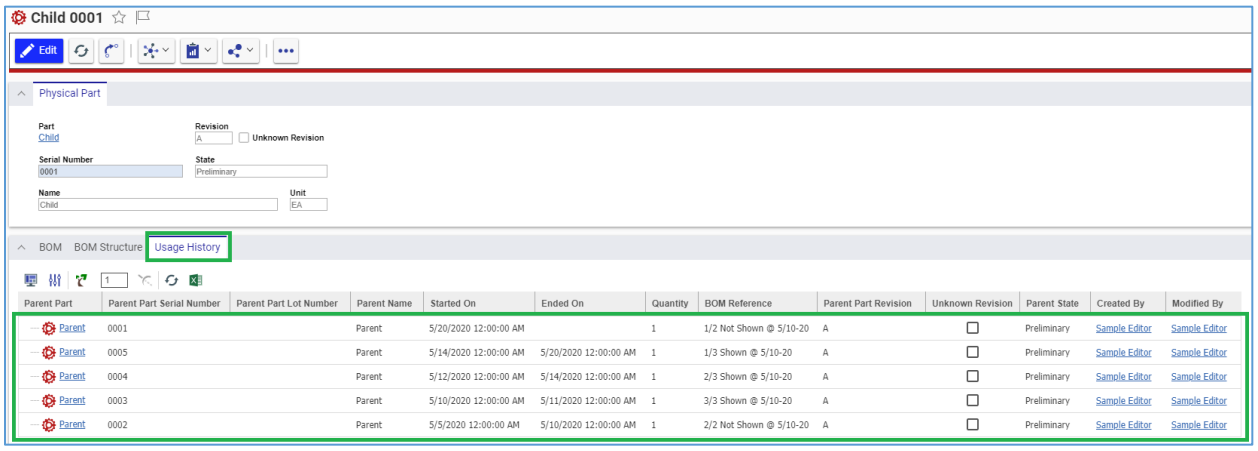


Figure 290.

The default **Usage History** grid sorting is by the **Started On** values in the descending order: from the last to the first date.

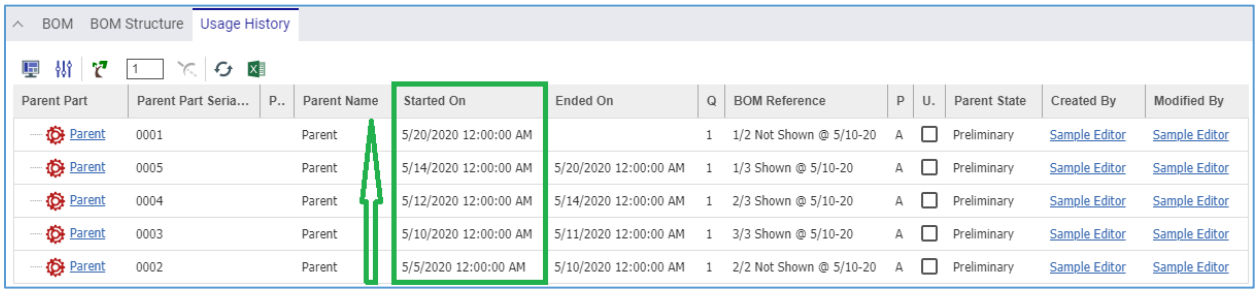


Figure 291.

The **Usage History** accordion tab has a TGV toolbar to manage and explore its TGV:

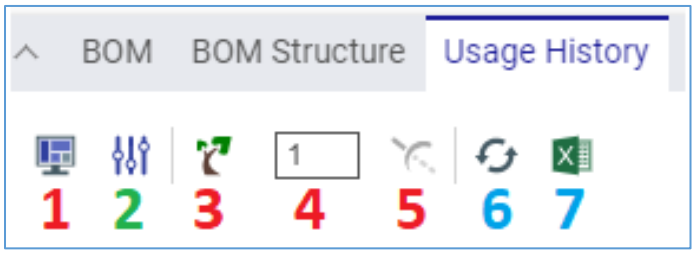


Figure 292.

1. **Display Settings**: a button to display the **Display Settings** dialog box for setting up Item TGV visibility.
2. **Modify Parameters**: a button to display the **Parameters** dialog box for filtering the parents against a period of time.
3. **Grow**: a button to expand a structure.
4. **Grow Depth**: a field to set up the level of the structure expansion.
5. **Trim**: a button to collapse the structure.
6. **Refresh**: a button to update the **Usage History** grid with the latest data.
7. **Export to Excel**: a button to save the grid of the parents as an Excel file.

This document discusses only the filtering feature (2). As the grid of the parents is single-level, it does not require features for a multi-level structure (1, 3, 4, and 5).

To open a **Physical Part** Item view:

1. Right-click the **Physical Part** Item **row**.
2. Click **View**.

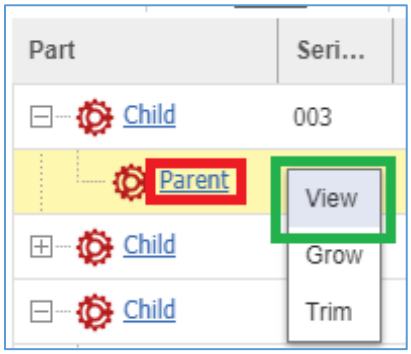


Figure 293.

A link in the **Part** cell opens an Item view of a related **Part** Item, not of the **Physical Part** Item.

5.4.2 Exploring Physical Part usage history over a time period

To filter parents in the **Usage History** grid against a given period of time:

1. Click the **Modify Parameters** button on the **Usage History** toolbar. The **Parameters** dialog box pops up.

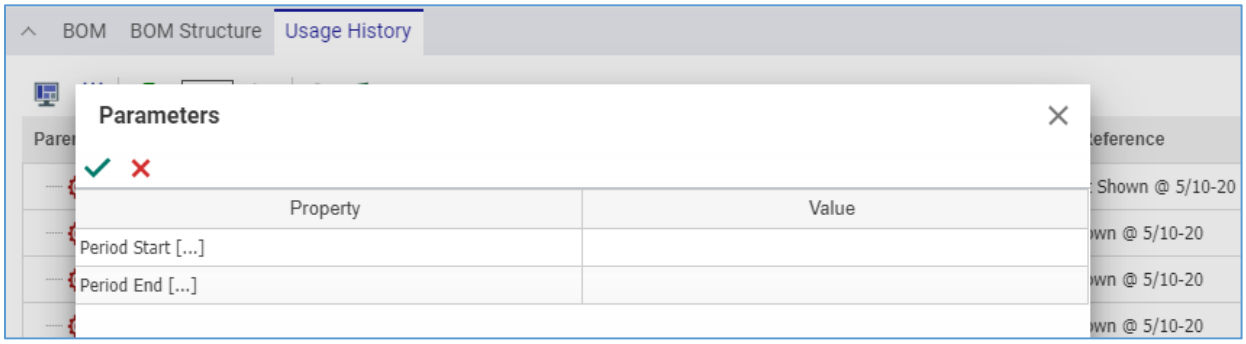


Figure 294.

2. Double-click the **Value** cell of the **Period Start** row. The **Date dialog** pops up.

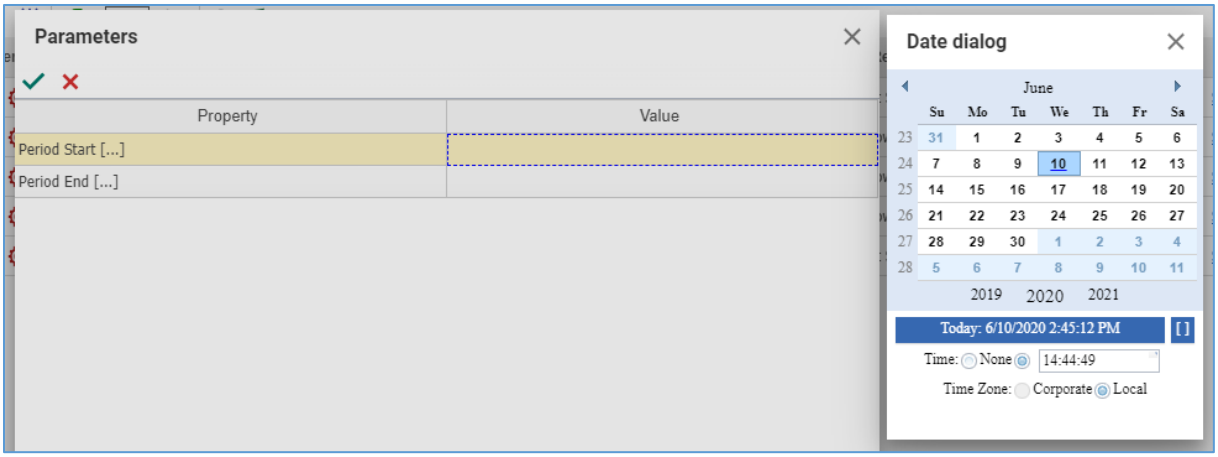


Figure 295.

3. Specify the time point as a date and time value. The **time point** populates the **Value** cell of the **Period Start** row.

Note: If the **Value** cell of the **Period Start** row has no given **time point**, repeat steps 2 and 3.

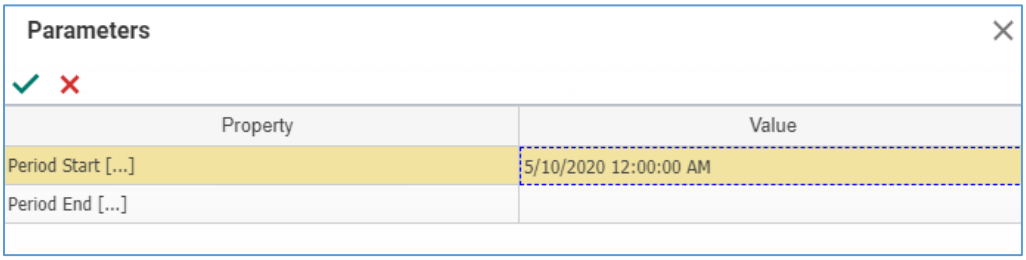


Figure 296.

4. Double-click the **Value** cell of the **Period End** row. The **Date dialog** pops up.
5. Specify the time point as a date and time value. The **time point** populates the **Value** cell of the **Period End** row.

Note: If the **Value** cell of the **Period End** row has no given **time point**, repeat steps 4 and 5.

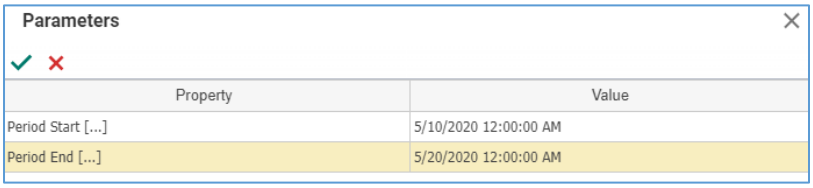


Figure 297.

6. Click **Apply**.

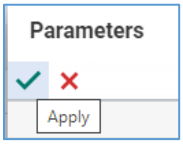


Figure 298.

The **Usage History** grid displays only the **Physical Part BOM** Items where the given **Physical Part** Item is a current child at the given time period.

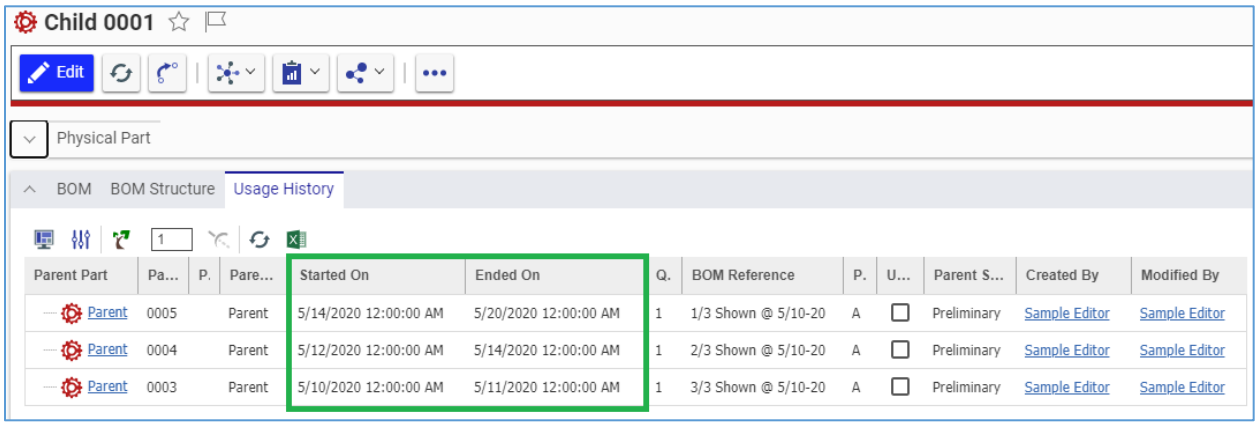


Figure 299.

To be evaluated as current over a time period, a **Physical Part BOM** Item of the child must have its property values as follows:

- **Started On** less than **Period End**.

AND

- **Ended On** greater than **Period Start** or blank.

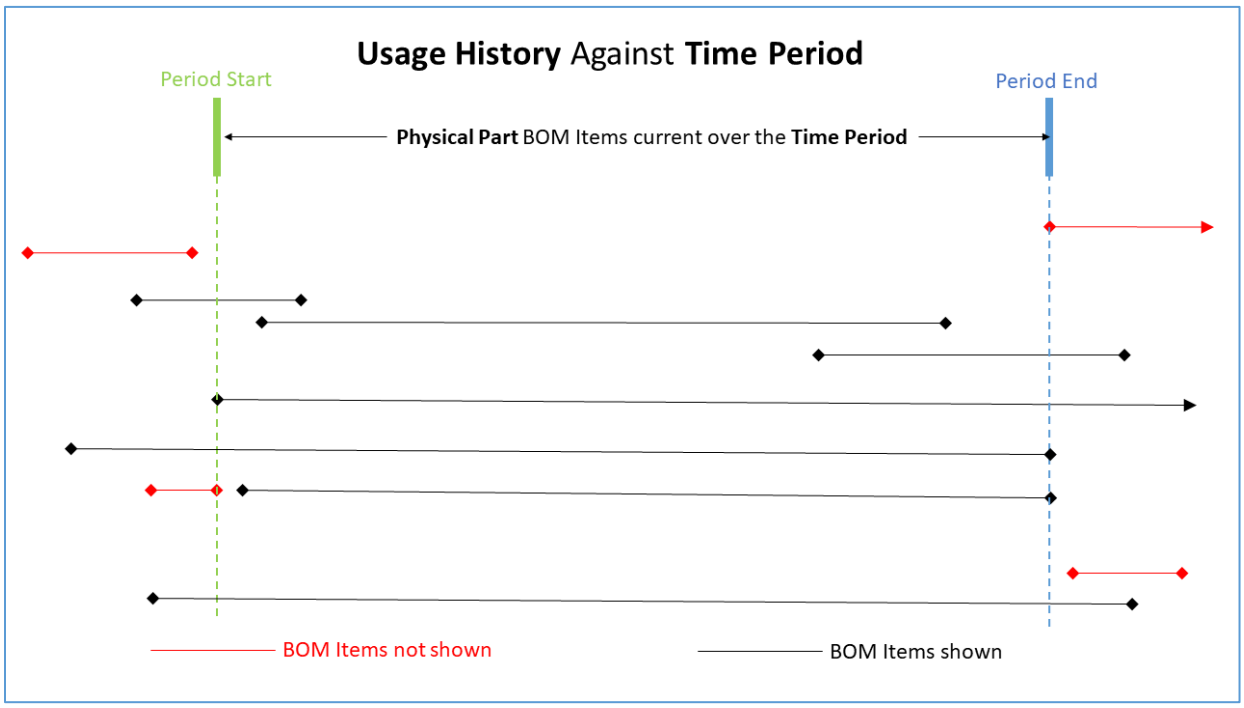


Figure 300.

5.5 Physical BOM Circular Dependency Report

5.5.1 Understanding Circular Dependency

A *Circular Dependency* is when a **Physical Part** BOM becomes “infinite” because of a parent **Physical Part** Item that has, at some level, a child **Physical Part** Item with this parent as its child. A *Circular Reference* is an industry-common synonym of the Circular Dependency.

Circular Dependency is introduced to a **Physical Part** BOM at some time point when a **Physical Part** Item creating a loop is added to this BOM.

The simplest case of a **Physical Part** BOM Circular Dependency is when a given parent **Physical Part** Item has a child **Physical Part** Item.

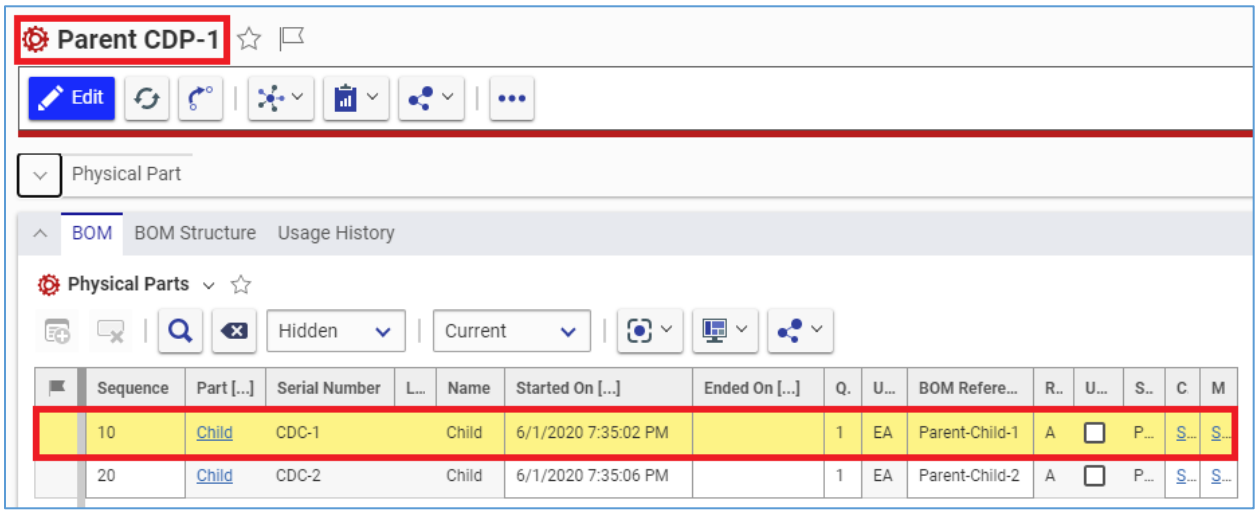


Figure 301.

This child **Physical Part** Item also has this parent **Physical Part** Item as a child.

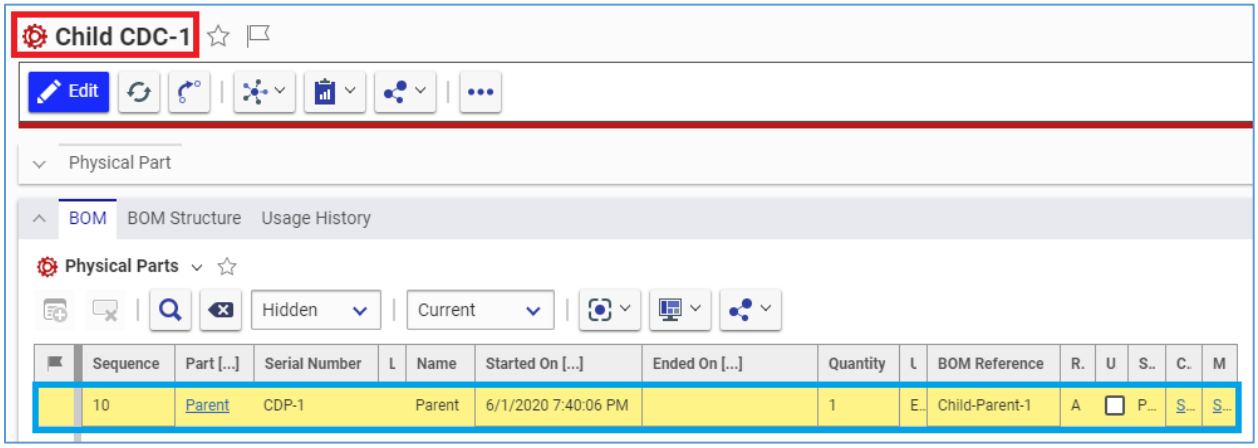


Figure 302.

The parent **Physical Part** BOM is infinite.

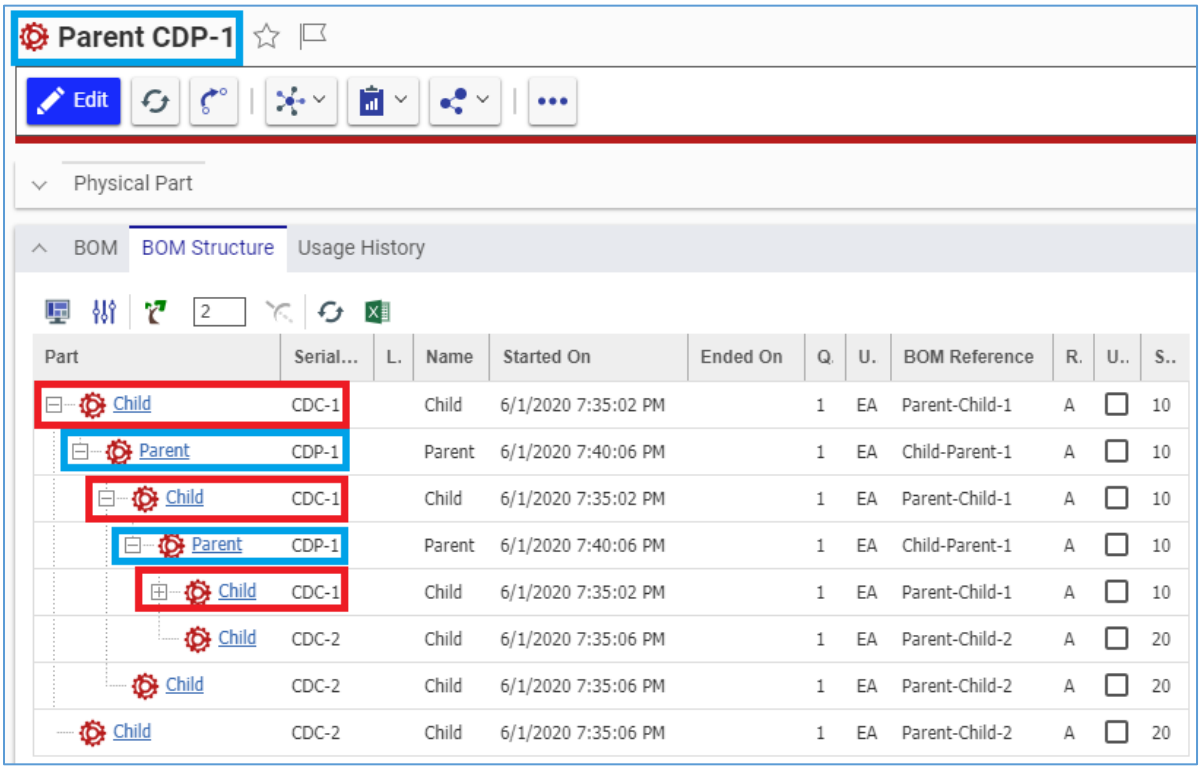


Figure 303.

The child **Physical Part** BOM is infinite too.

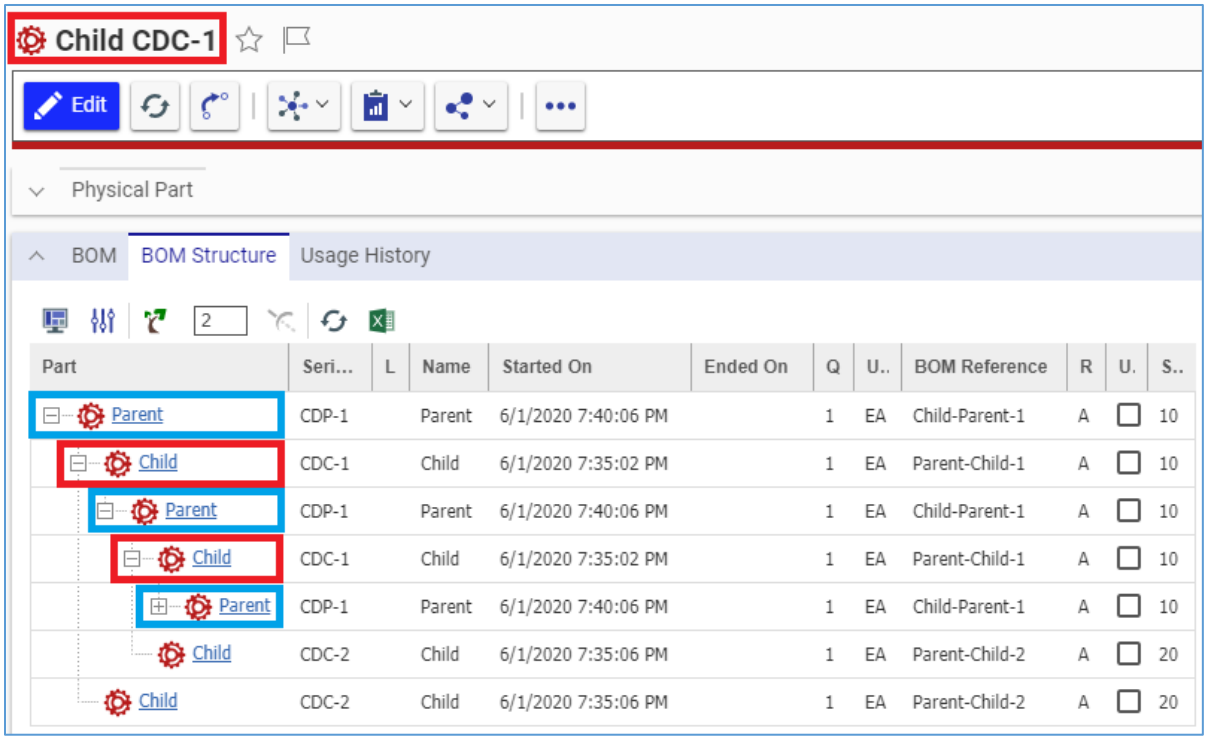


Figure 304.

A more complicated case is when the parent is a child at some **Physical Part** BOM level.

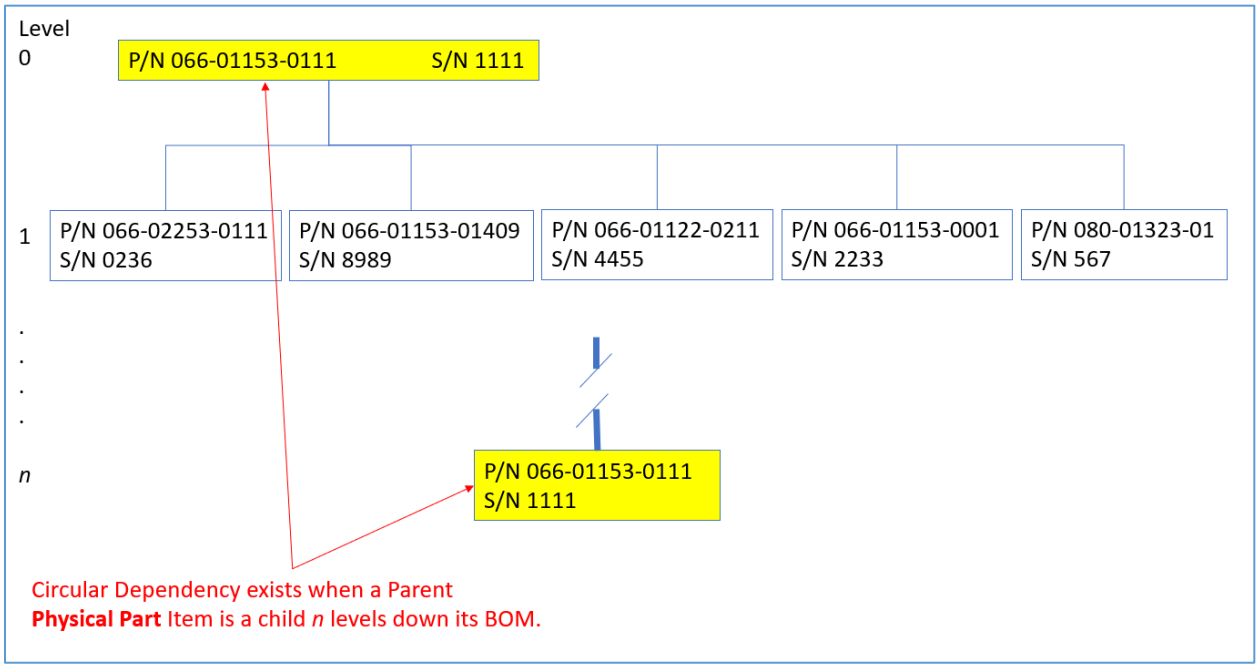


Figure 305.

There is no Circular Dependency when the child is a child in a branch of a different unconnected child in a multi-level **Physical Part** BOM.

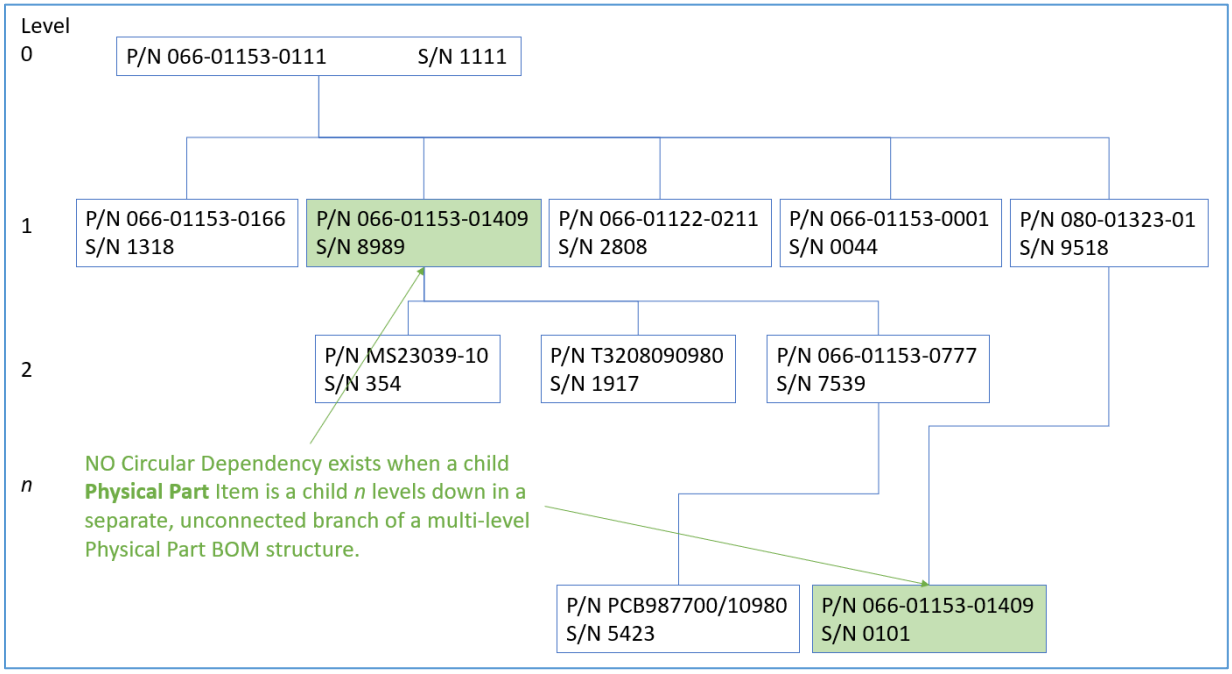


Figure 306.

An applied multi-level **Physical Part** BOM can have varieties of the aforementioned cases over the **Physical Part** BOM life. If there are hundreds or thousands of Items, finding a Circular Dependency manually can be laborious.

5.5.2 Working with the Circular Dependency Report

The DTC application offers the **Physical BOM Circular Dependency** report for detecting Circular Dependencies in a multi-level **Physical Part** BOM.

There are two ways to access this Report:

- Go to the **Physical Parts** Search Grid, right-click a **Physical Part** Item containing a BOM that needs to be checked, and then click **Reports** --> **Physical BOM Circular Dependency**.

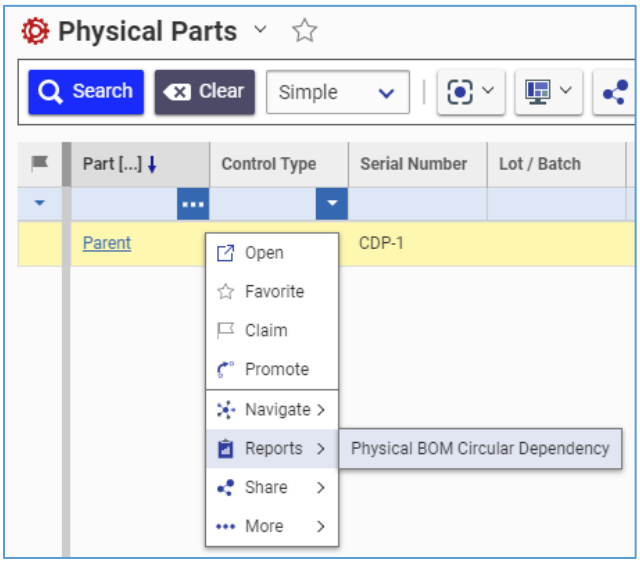


Figure 307.

- Open the view of a **Physical Part** Item containing a BOM that needs to be explored and then click **Reports** --> **Physical BOM Circular Dependency** on the Item toolbar.

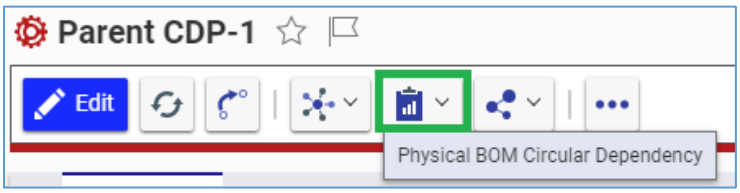


Figure 308.

Use the following procedure to check a multi-level **Physical Part** BOM for Circular Dependencies at a time point:

1. Access the **Physical BOM Circular Dependency** report. The **Circular report settings** dialog box pops up.

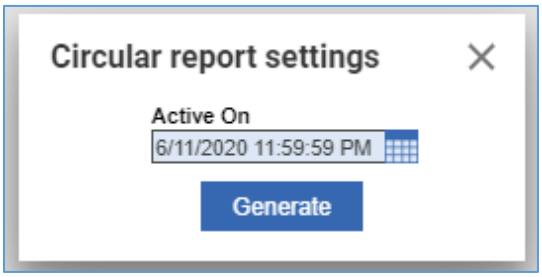


Figure 309.

2. Click the **Calendar** button. The **Date dialog** box pops up.

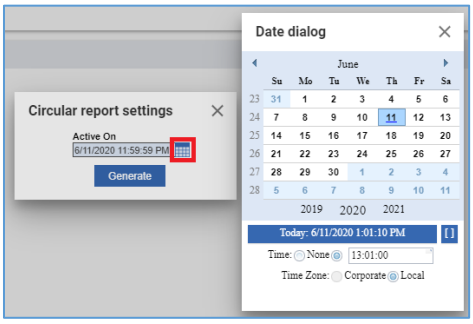


Figure 310.

3. Specify a time point against which the report will be generated. The **Date dialog** box disappears, and the selected time point populates the **Active On** field.
4. Click the **Generate button**.

   The **Physical BOM Circular Dependency** report appears as a new Innovator tab.

Depending on whether a given **Physical Part** BOM has Circular Dependencies or not, the **Physical BOM Circular Dependency** report has different contents:

- If there are Relationship loops indicating which **Physical Part** Items are mutually related to each other.

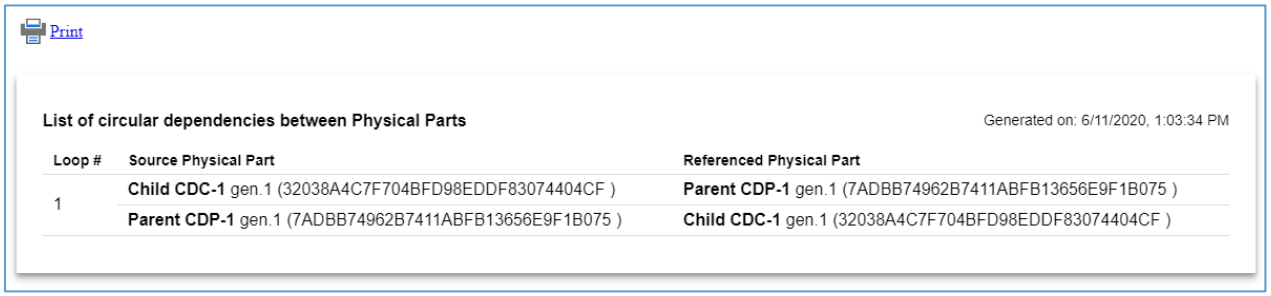


Figure 311.

- If there are none an appropriate message appears.

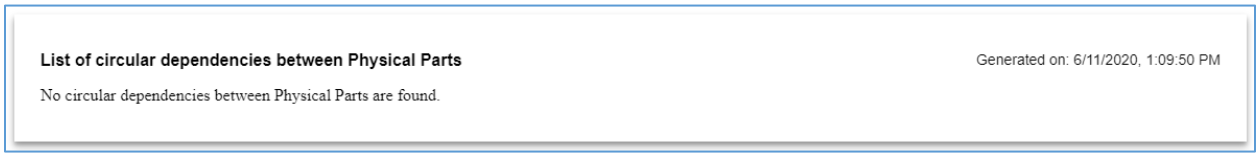


Figure 312.

To quit the **Physical BOM Circular Dependency** report, close its view in Aras Innovator.

At the top left corner of the **Physical BOM Circular Dependency** report Innovator tab, there is a **Print** button. This button uses standard browser print function. Depending on the browser and your specific configuration, various print choices, such as printing to a printer or saving to PDF, are available.

Figure 313.

6 Life Units

The DTC application introduces Life Units, Life Parameters, Life Policies, and Part Policies for tracking, controlling, and limiting the lives of physical assets, whether they are in service, inventory, or under maintenance. If your organization has decided to track asset lives, the Life Units first should be created and set up firstly. The Life Parameters are the next.

6.1 Understanding Life Units

A **Life Unit** Item is a unit of measurement for tracking asset life variables of the same kind. For example, the **Hours** Life Unit Item represents the hour unit of measure and is used for tracking any hours: total running hours since new, running hours since a repair, idle hours, and so on.

Note: The DTC application is shipped with no **Life Unit** Items out of the box. This section uses some industry-agnostic **Life Unit** Items as illustrations.

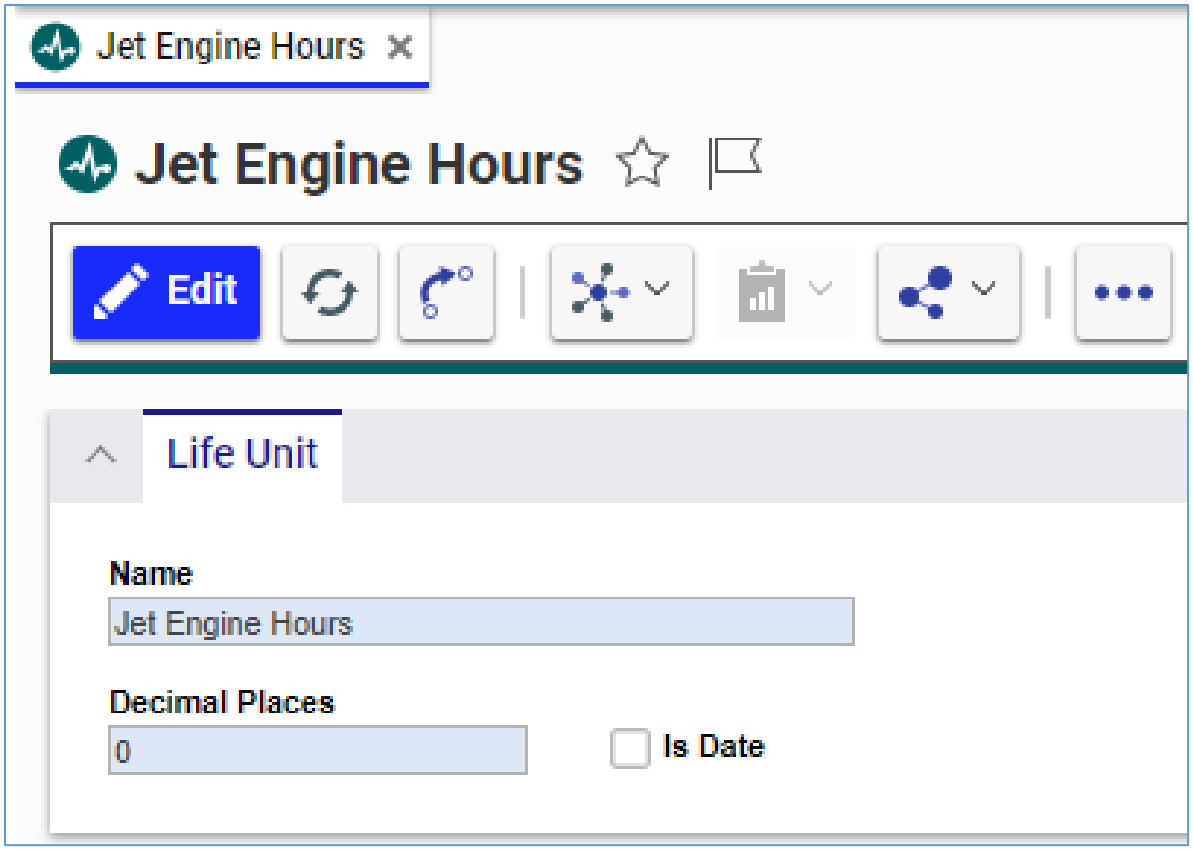


Figure 314.

A **Life Unit** Item can be a numeric or date unit.

The **Life Unit** Items have no Life Cycle map, thus States.

6.2 Life Units Search Grid

The **Life Units** Search Grid has the standard features and the following columns:

- **Name**
- **Decimal Places**
- **Is Date**

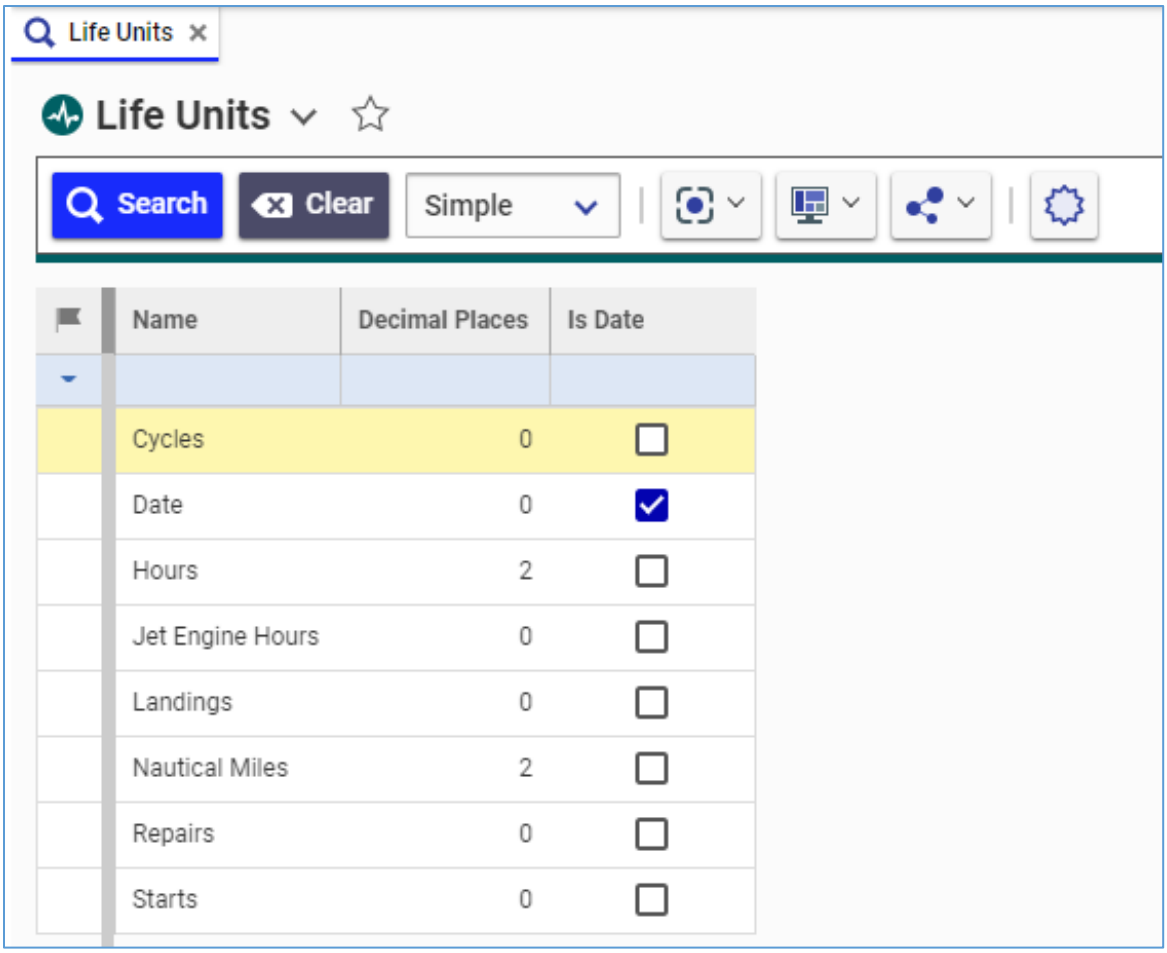


Figure 315.

Each column shows a corresponding **Life Unit** Item property. For the property details, see section 6.3 Life Unit properties.

The **Life Unit** Items are sorted in the Search Grid and other grids by their **Name** values in the ascending order (from A to Z).

6.3 Life Unit properties

The following properties of a **Life Unit** Item are available at the **Form** accordion of this **Life Unit** Item view:

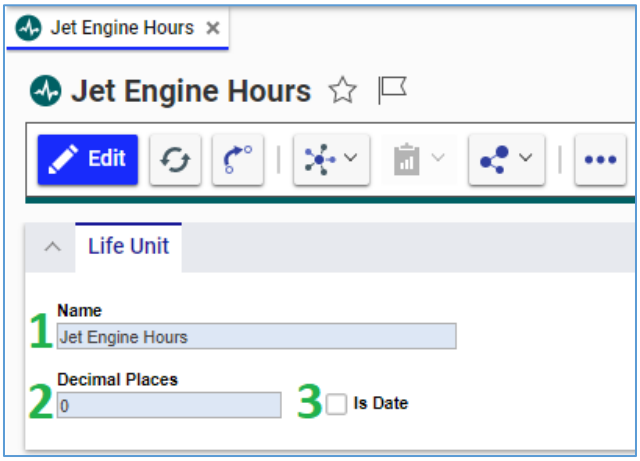


Figure 316.

1. **Name**: a unique name for a given **Life Unit** Item among other Life Units. This property is required and must have a unique text value of up to 32 characters long. The characters can be of any type. The **Name** should briefly explain the **Life Unit** Item as a generic or specific unit of measurement. For example, the **Hours** Life Unit Item is generic because it represents the hour unit of measure for tracking any hours of any assets. The **Generator Hours** Life Unit Item can be created for tracking any hours of generators if their hours should be tracked differently than the generic hours because of different precision or other reasons.
2. **Decimal Places**: the number of decimal places (precision of the fractional part) of the actual values for the given **Life Unit**. This property is required. It is set to **0** (zero) by default when creating a new **Life Unit** Item. It must be **0** for dates and whole integer counters, like **Landings**, **Repairs**, and so on. When it is not **0**, the **Life Unit** Item is a fractional number.
3. **Is Date**: a flag indicating when selected (**true**) that the given **Life Unit** Item is a unit of measure of the date and type; thus, it cannot have numeric values. This property is required. It is not selected (**false**) by default when creating a new **Life Unit** Item.

6.4 Life Unit Relationships and usage

A **Life Unit** Item has no **Relationships** accordion because it is used in:

- A **Life Parameter** Item as a foreign property value.
- An **OperationalEvent LifeUnit** Relationship Item as a related Item.

To become a unit of measure for tracking an asset life variable, a **Life Unit** Item should be applied to a **Physical Part** Item representing this asset as follows:

1. This **Life Unit** Item is specified in a **Life Parameter** Item. It is shown in the **Unit of Measure** field there. One **Life Unit** Item can be referenced in many **Life Parameter** Items. See section 7.3 Life Parameter properties.
2. This **Life Parameter** Item is attached to a **Physical Part** Item representing the asset. One **Life Parameter** Item can be used in many **Physical Part** Items. See section 7.5 Life Parameter Relationships and usage.

3. The current asset life variable values are entered and shown on one of the following **Physical Part** Item Relationship tabs:
   a. **Current Life Values** if the **Life Unit** Item is a number. See section 4.7.2 Tracking Numeric Life Variables of Physical Parts.
   b. **Current Date Values** if the **Life Unit** Item is a date. See section 4.7.3 Tracking Life Dates of Physical Parts.

To track an asset life variable increment gained during the operational asset activity, the **Life Unit** Item should be present in the **Life Units** Relationships grid of an **Operational Event** Item representing this activity. See section 11.4.3 OperationalEvent LifeUnit Relationships.

Warning If using **Operational Event** Items, avoid the creation of test, dummy, or extra **Life Unit** Items and ensure that new **Life Unit** Items are free of errors. The system automatically populates a created or updated **Operational Event** Item with all **Life Unit** Items present in the database. It is not possible to modify or delete a **Life Unit** Item related to an **Operational Event** Item. If it is necessary to edit or delete the related **Life Unit** Item from the database, all of its source **Operational Event** Items should be deleted from the database first. An **Operational Event** Item cannot be deleted in the **Complete** State.

Use the **Where Used** browser to explore the usage of a **Life Unit** Item. See section 13 The Where Used Browser and DTC Items. Also, you can filter the **Life Parameter** Item Search Grid by the given **Life Unit** Item.

6.5 Managing Life Units

Your ability to manage a **Life Unit** Item depends on your Asset Identity membership and Item usage. This section assumes that you are familiar with the **Life Unit** Item properties; discussed in sections 6.3 Life Unit properties.

6.5.1 Creating Life Units

Note: Only an **Asset Admin** can create the **Life Unit** Items.

To create a new **Life Unit** Item:

1. Go to **Contents** --> **Assets** --> **Life Units**.
2. Click **Create Life Unit**. The **Life Unit #** Item view appears.

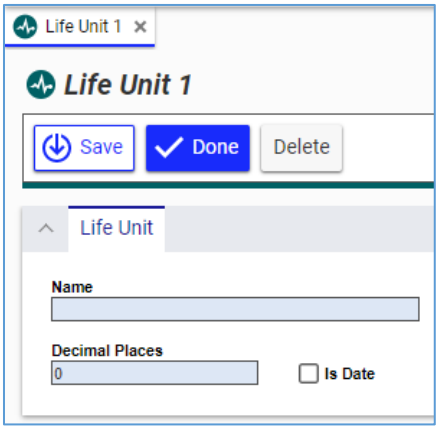


Figure 317.

3. In the **Name** field, enter a unique name for the **Life Unit** Item.

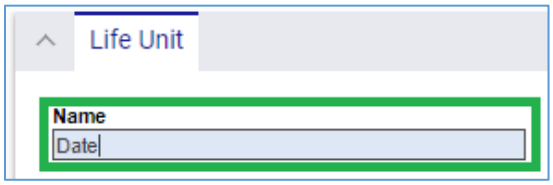


Figure 318.

4. If the represented unit of measure is a decimal fraction number, in the **Decimal Places** field, enter the unit of measure precision: number of digits after the decimal point.

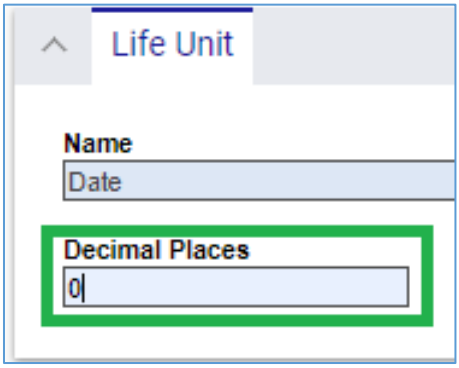


Figure 319.

5. If the represented unit of measure is a date, select the **Is Date** check box.

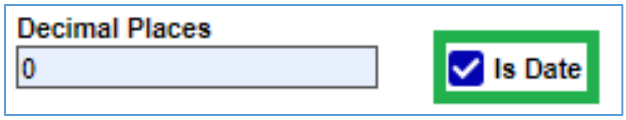


Figure 320.

6. Click either **Save** or **Done** on the **Life Unit** Item toolbar. The new **Life Unit** Item is created. The **Life Unit** Item **Innovator tab** and **Page Title** bar show the Item **Name** value.

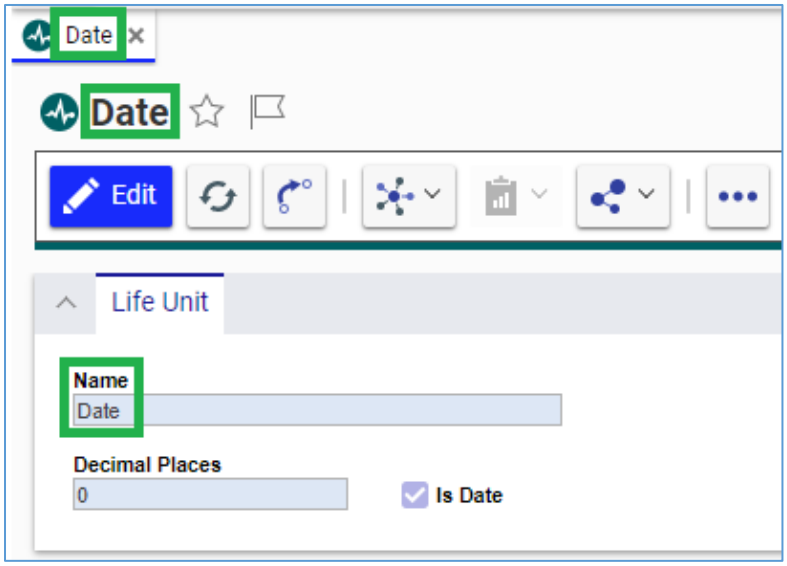


Figure 321.

The **Life Unit** Item is now available via search, subject to the **Asset** Permissions.

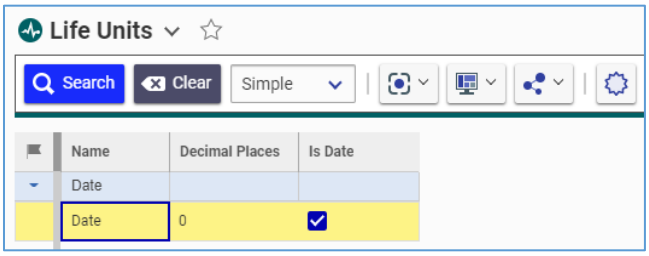


Figure 322.

An error is raised when trying to save a **Life Unit** Item with a blank field for a required property or a duplicating **Name** value. For details, see section 6.5.4 Validation of Life Units.

6.5.2 Modifying Life Units

Note: Only an **Asset Admin** can modify the properties of a **Life Unity** Item and only when this Item is not used. If otherwise, clicking the **Edit** button on the **Life Parameter** Item toolbar raises an error.

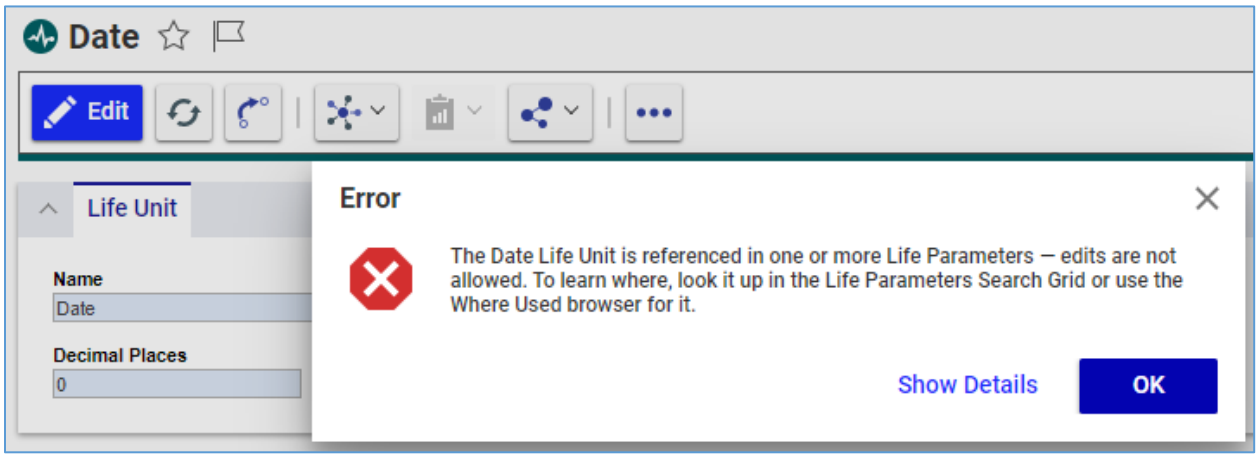


Figure 323.

To modify a **Life Unit** Item, use the standard modifying procedure.

If the **Name** value has been changed, the **Life Parameter** Item **Innovator tab** and **Page Title** bar show its new value.

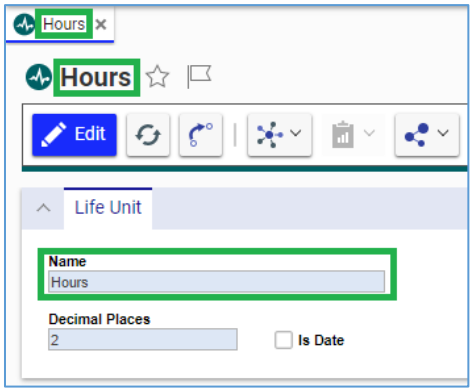


Figure 324.

An error is raised when trying to save a **Life Unit** Item with a blank field for a required property or a duplicating **Name** value. For details, see section 6.5.4 Validation of the Life Units.

6.5.3 Deleting Life Units

Note: Only an **Asset Admin** can delete a **Life Unit** Item and only when this Item is not used. If otherwise, an error is raised.

Figure 325.

To delete a **Life Unit** Item, use the standard deletion procedure.

6.5.4 Validation of Life Units

Once you click **Save** or **Done** on a **Life Unit** Item toolbar, the system validates this Item and raises an appropriate error if the Item:

- Is missing a **Name** value.

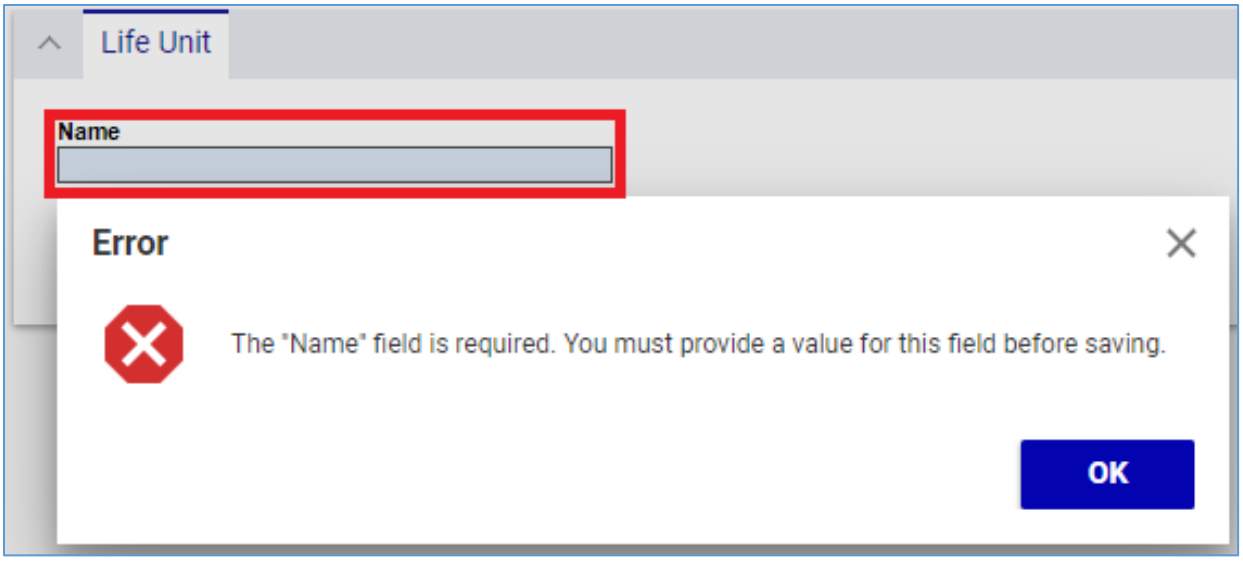


Figure 326.

- Is missing a **Decimal Places** value.

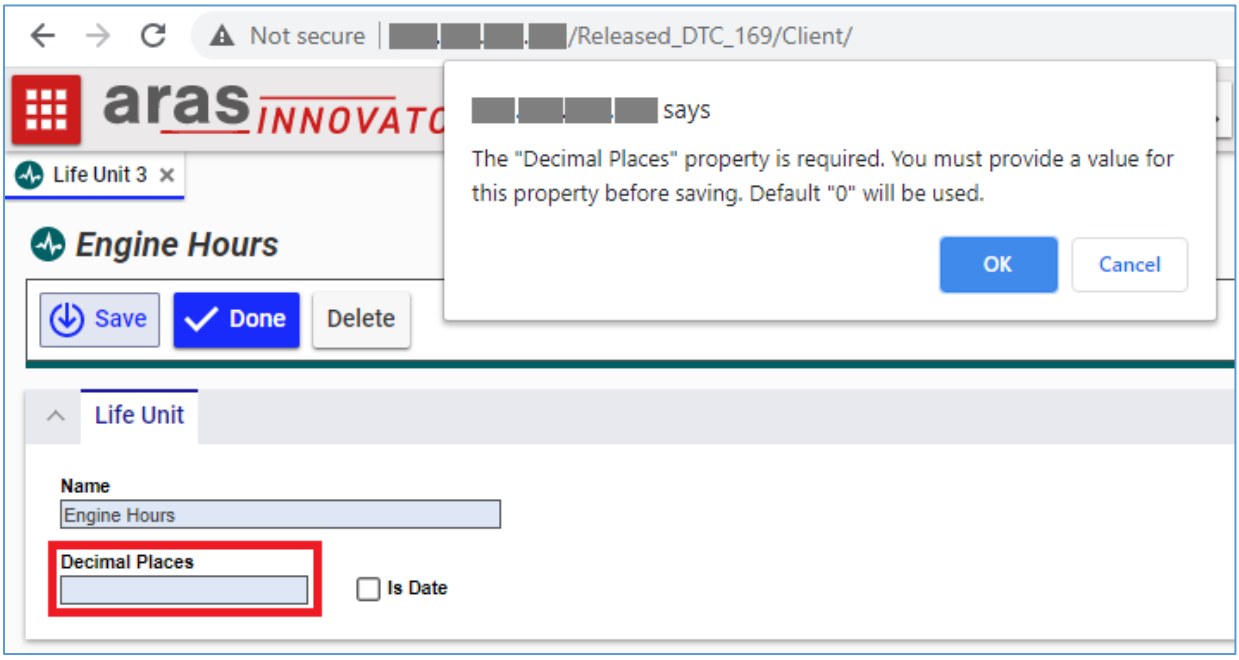


Figure 327.

- Has a **Name** value that is not unique: already given to an existing **Life Unit**.

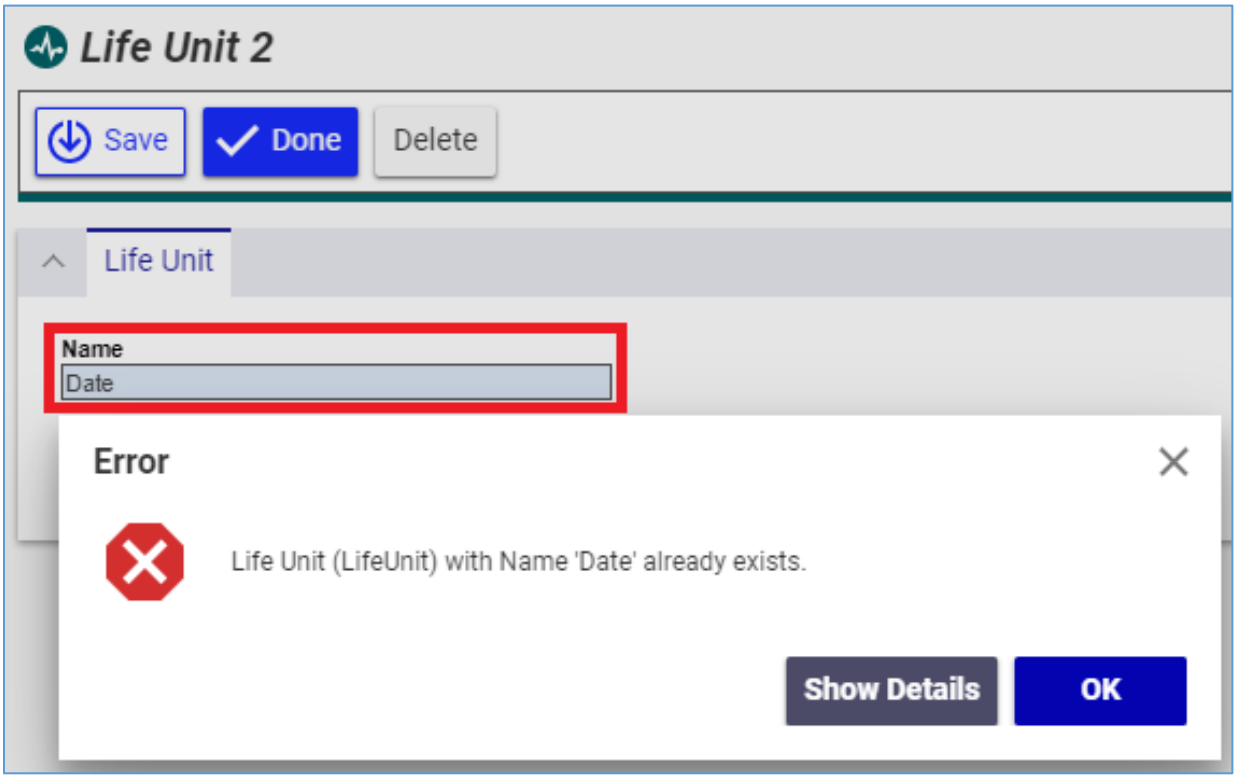


Figure 328.

7 Life Parameters

When all necessary Life Units exist in the system, it is the time to create and set up Life Parameters. Once that work is completed, the turn of Life Policies will come.

7.1 Understanding Life Parameters

Different organizations across many industries require life tracking and controlling of their products in inventory and service. After the products have been manufactured, their individual characteristics change with every minute, usage, or consumption of life.

A **Life Parameter** Item is a template of a measurable metric for a given **Physical Part** Item life condition. It should represent an operational variable or life characteristic of a real-world asset—operational (life) variable. While a represented asset is operating in its field, its life variables cumulatively increase, showing how much of the asset service life has been depleted and remains. For example, the **TSN** (**Time Since New**) Life Parameter is the number of hours a serial-controlled **Physical Part** Item has been operating in service since its manufacturing.

Note: The DTC application is shipped with no **Life Parameter** Items out of the box. This section uses some industry-agnostic **Life Parameter** Items as illustrations.

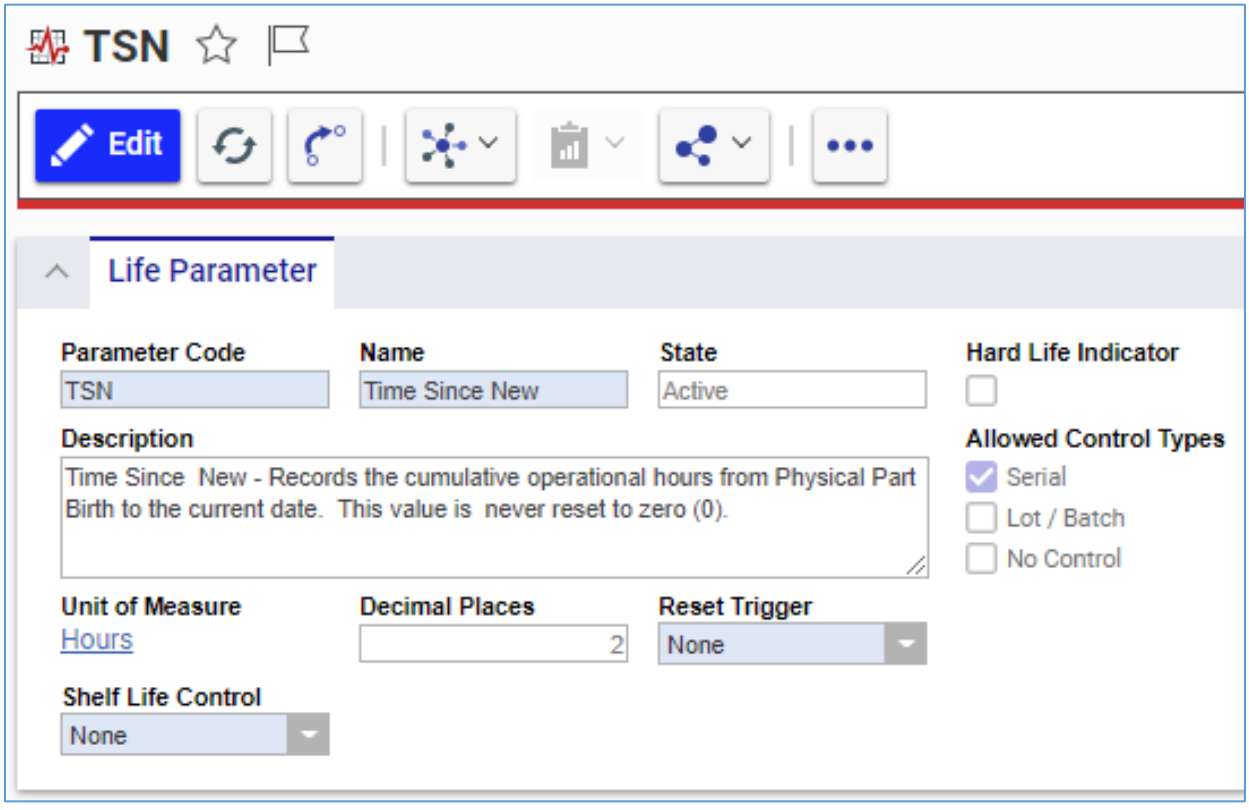


Figure 329.

The **Life Parameter** Item can also define an expiration condition driven by either an operational variable or a date. In this case, it sets a limit when the asset should be terminated. The asset can reach its life limit by either accumulating life variable values or reaching a specific date. For example, **TSN** is accumulated, and **SLD** (**Shelf Life Defined**) is an expiration date.

The **Unit of Measure** property defines how the given **Life Parameter** Item tracks the asset life or defines the expiry condition.

Life variables of a numeric unit of measure can be tracked only for the serial-controlled assets because we can identify individually only such assets. We cannot identify each lot-controlled or non-controlled real-word asset individually. The lot-controlled assets are tracked by batches. The non-controlled assets are not trackable by their definition.

One serial-controlled **Physical Part** Item can be present simultaneously only in one Physical Part BOM structure and only in the quantity of one. One lot-controlled or non-controlled **Physical Part** Item can be present in one or more Physical Part BOM structures in quantity not equal to one as multiple items from one batch can be present in different asset assemblies or at various places in the same assembly.

Thus, a **Life Parameter** Item applicable to the **Serial** Control Type only can have a numeric **Life Unit** in its **Unit of Measure** property. A **Life Parameter** Item that includes **Lot / Batch** or **No Control** in its **Allowed Control Types** property cannot have a numeric **Life Unit** in its **Unit of Measure** property; an error is raised if trying to save such a **Life Parameter** Item.

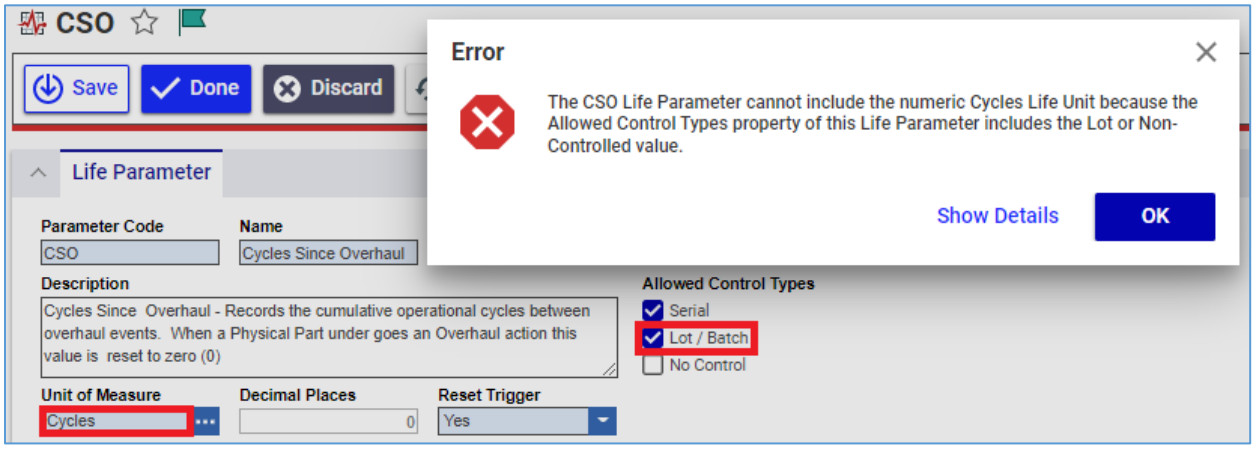


Figure 330.

Warning Before DTC 12.0R3, a **Life Parameter** Item applicable to the **Lot / Batch** or **No Control** Control Type was allowed to include a **Life Unit** Item representing a numeric unit of measure. If migrating from a lower DTC version, ensure that the imported legacy database does not have **Life Parameter** Items with **Lot / Batch** or **No Control** selected in **Allowed Control Types** and a numeric **Life Unit** specified in **Unit of Measure**.

For details on the **Life Parameter** Item properties and their behavior, see sections 7.3 Life Parameter Properties and 7.4 Behavior of Life Parameter properties.

One **Life Parameter** Item can be applied to many **Physical Part** Items within its **Allowed Control Types**.

It is not possible to modify or delete a used **Life Parameter** Item. For details on using Life Parameters, refer to section 7.5 Life Parameter Relationships and usage.

7.2 Life Parameters Search Grid

The **Life Parameters** Search Grid has the standard features and the following columns:

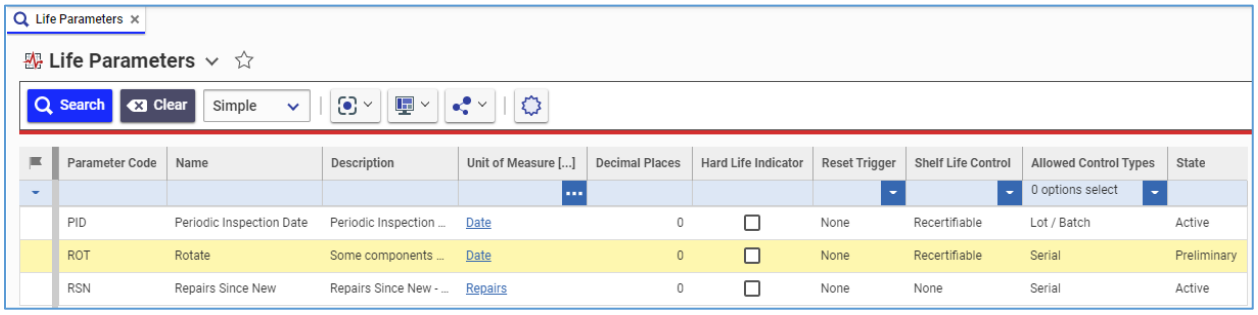


Figure 331.

- **Parameter Code**
- **Name**
- **Description**
- **Unit of Measure**
- **Decimal Places**
- **Hard Life Indicator**
- **Reset Trigger**
- **Shelf Life Control**
- **Allowed Control Types**
- **State**

Each column shows a corresponding **Life Parameter** Item property. For the property details, see section 7.3 Life Parameter Properties.

The **Life Parameter** Items are sorted in the Search Grid and other grids by their **Parameter Code** values in the ascending order (from A to Z).

7.3 Life Parameter Properties

The following properties of a **Life Parameter** Item are available at the **Form** accordion of this **Life Parameter** Item view:

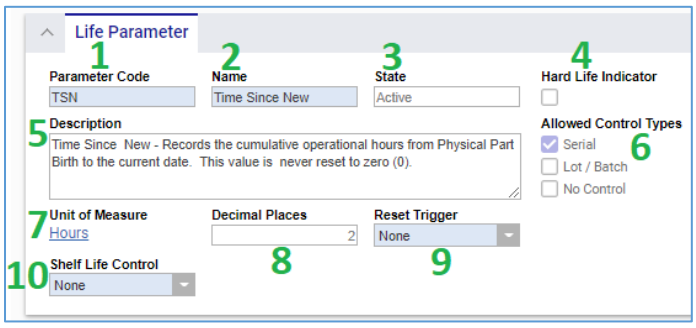


Figure 332.

1. **Parameter Code**: a unique identifier of a given **Life Parameter** Item among other Life Parameters. This property is required and must have a unique value of up to 10 characters long. The characters can be of any type. It is recommended to use an abbreviation of the Item **Name** property as its **Parameter Code**.
2. **Name**: a full name of the **Life Parameter** Item. This property is required and must have a value of up to 30 characters long. The characters can be of any type. The **Name** should briefly explain the **Life Parameter** Item as an asset life variable.
3. **State**: the current State of the **Life Parameter** Item. See section 7.6 Life Parameter Life Cycle and promotion.
4. **Hard Life Indicator**: a flag indicating when selected (**true**) that the given **Life Parameter** Item limits the lives of **Physical Part** Items where it is applied. It is not selected (**false**) by default when creating a new **Life Parameter** Item. The property behavior, including its availability, depends on other properties, see section 7.4 Behavior of Life Parameter properties.
   A critical asset that fails in service due to losing its operational qualities within accumulated wear or fatigue requires tracking its corresponding life variables to their maximum values, where this asset must be withdrawn from the service. A **Life Parameter** Item with **true** for its **Hard Life Indicator** sets the maximum failure (hard) limit for given **Physical Part** Item life variable value and accumulates its current values to this limit. For example, the **Maximum Hard Time** Life Parameter defines the limit for the **Physical Part** Item working hours and accumulates these hours up to this limit. Once the limit is reached, the **Physical Part** Item is prohibited from serving.
5. **Description**: detailed or additional information about the **Life Parameter** Item. This property can have a value of up to 256 characters long. The characters can be of any type.
6. **Allowed Control Types**: a group of check boxes where a selected (**true**) check box defines a control type to which the given **Life Parameter** Item is applicable. For example, if **Serial** and **Lot/Batch** are selected, the **Life Parameter** Item can be applied only to a serial-controlled or lot-controlled **Physical Part** and **Part Policy** Items and used only in a **Life Policy** Item of the **Serial**, **Lot/Batch**, or **Serial** and **Lot/Batch** control types. This property is required.
   A **Life Parameter** Item can be applicable only to the **Physical Part** Items of a specific control type. For example, the **TSN** (**Time Since New**) Life Parameter can accumulate hours only for one asset; thus, it cannot be applied to non-serialized **Physical Part** Items because they represent sets of many assets; thus, it cannot be applied to non-serialized **Physical Part** Items because they represent sets of many assets.
7. **Unit of Measure**: a **Life Unit** Item that indicates how the given **Life Parameter** Item life variable is tracked. For example, the **TSN** Life Parameter uses the **Hours** Life Unit as its **Unit of Measure** for accumulating the asset running time as hours. This property is a **Life Unit** Item and required. Its behavior depends on other properties; see section 7.4 Behavior of Life Parameter properties.
8. **Decimal Places**: a view-only **Decimal Places** property value of the given **Life Unit** Item (7).
9. **Reset Trigger**: a setting that defines whether the given **Life Parameter** Item actual values on the applied **Physical Part** Items can be reset. This property is a list and required. It is set to **None** by default when creating a new **Life Parameter** Item. It is always **None** for the **Life Unit** Items (7) that are dates: their **Is Date** property value is **true** (**selected**). The setting options are the following:
   a. **None**: the actual values cannot be reset. For example, the **TSN** Life Parameter cannot be reset because it accumulates the given **Physical Part** Item running hours since the Item birth.
   b. **Yes**: the actual values can be reset. Some asset life variables are tracked between repair, overhaul, test, or other events. For example, the **TSLO** (**Time Since Last Overhaul**) Life Parameter accumulates the given **Physical Part** Item running hours since the last overhaul. When this **Physical Part** Item is refurbished and its current overhaul event is completed, its

TSLO should be reset to **0** (zero). When put back in operation, the **Physical Part** Item begins to increase its **TSLO** value once again while running till the next overhaul event.

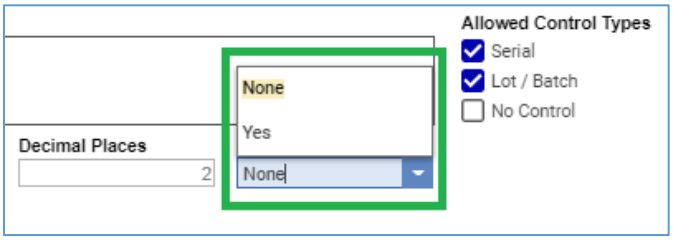


Figure 333.

10. **Shelf Life Control**: a setting that defines whether the **Life Parameter** Item controls the shelf life of the **Physical Part** Items (assets) where it applies. This property is a list and is required. It is set to **None** and not available by default when creating a new **Life Parameter** Item or when a numeric **Life Unit** Item (its **Is Date** property value is **false** (**cleared**)) is specified in the **Unit of Measure** property. It becomes available and set to **Recertifiable** automatically when a **Life Unit** Item of the date type is specified in the **Unit of Measure** property. If it is not **None**, the **Life Parameter** Item demands a **Physical Part** Item (asset) to have an expiry date, until which the represented asset will remain available for sale or use. The list value options are the following:

    a. **None**: the **Life Parameter** Item does not control the shelf life of the **Physical Part** Items where it applies: it does not set the expiry date. This option is not available when a **Life Unit** Item of the date type is specified in the **Unit of Measure** property.

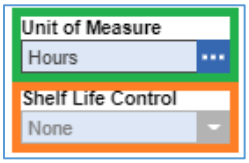


Figure 334.

    b. **Recertifiable**: the **Life Parameter** Item controls the shelf lives of the **Physical Part** Items where it applies, and their shelf lives can be extended—the asset must have an expiration date, and this date can be reassigned under some operational Quality Assurance circumstances. For example, manufacturing is short on a consumable asset, and there are some in stock with one day over their expiration date. Another example is when the QA department issues a concession that extends the expiration date. This option is available only when a **Life Unit** Item of the date type is specified in the **Unit of Measure** property.

    The **Recertifiable** setting of the **Shelf Life Control** property could also be used for a **Life Parameter** that initiates an asset review or reposition on a specific date to keep this asset in service and available for use. For example, some governed avionics devices with sensitive gimbals and bearings require periodic inspection and rotation to ensure that they do not fail to wear or brinelling. The discussed setting is an approach for initiating such procedures.

    c. **Not Recertifiable**: the **Life Parameter** Item controls the shelf lives of the **Physical Part** Items, and their shelf lives cannot be extended—the asset must have an expiry date, and this expiry date can be assigned only once. This option is available only when a **Life Unit** Item of the date type is specified in the **Unit of Measure** property.

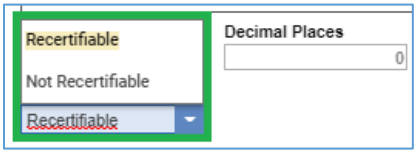


Figure 335.

7.4 Behavior of Life Parameter properties

The built-in business logic of a **Life Parameter** Item ensures that its properties can be set only in the combinations valid for any industry. Depending on a specific value of a property, the system sets some other properties to the valid values and disables assigning the invalid ones.

Both the **Hard Life Indicator** and **Shelf Life Control** properties control an asset life: they set a limit when the asset should be withdrawn from service. However, they are mutually exclusive in a **Life Parameter** Item as they do the same job differently:

- **Hard Life Indicator** defines that a given **Life Parameter** Item sets a *numeric* limit for a **Physical Part** Item and *accumulates* numeric values during the asset service till that limit.
- **Shelf Life Control** defines that a given **Life Parameter** Item sets a *date* limit for a **Physical Part** Item and *waits* for that date.

The **Physical Part** Item can have **Life Parameter** Items of both types.

If the **Unit of Measure** property is populated with a **Life Unit** Item that has its **Is Date** set as **true** (**selected**), other properties behave as follows:

- **Hard Life Indicator** is set to **false** (**cleared**) and frozen.
- **Reset Trigger** is set to **None** and frozen.
- **Shelf Life Control** can be set to **Recertifiable** or **Not Recertifiable**.
- The rest properties can be set to any value that their logic allows.

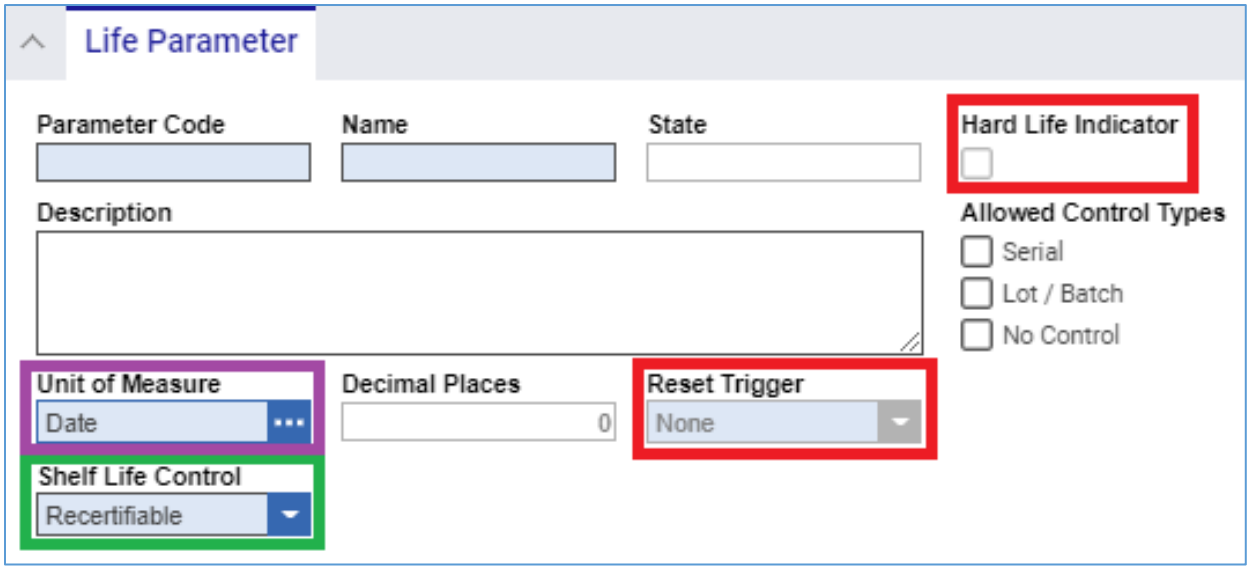


Figure 336.

If the **Unit of Measure** property is populated with a **Life Unit** Item that has its **Is Date** set as **false** (**cleared**), other properties behave as follows:

- **Shelf Life Control** is set to **None** and frozen.
- The rest properties can be set to any value that their logic allows.

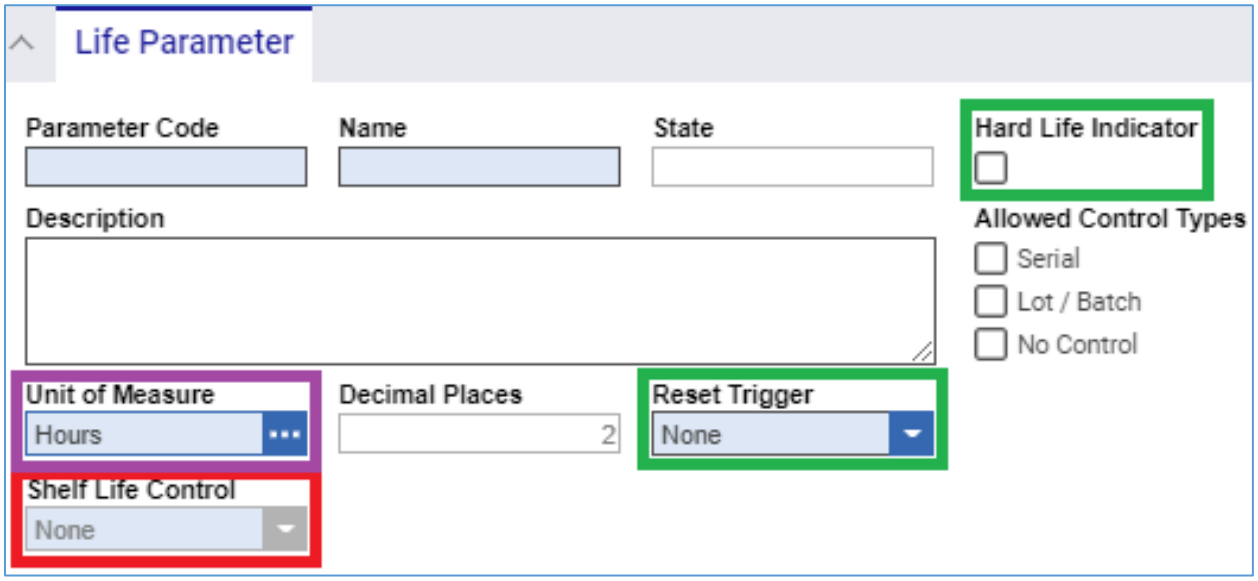


Figure 337.

If the **Hard Life Indicator** property is **selected** (**true**), other properties behave as follows:

- **Reset Trigger** is set to **None** and frozen.
- The rest properties can be set to any value that their logic allows.

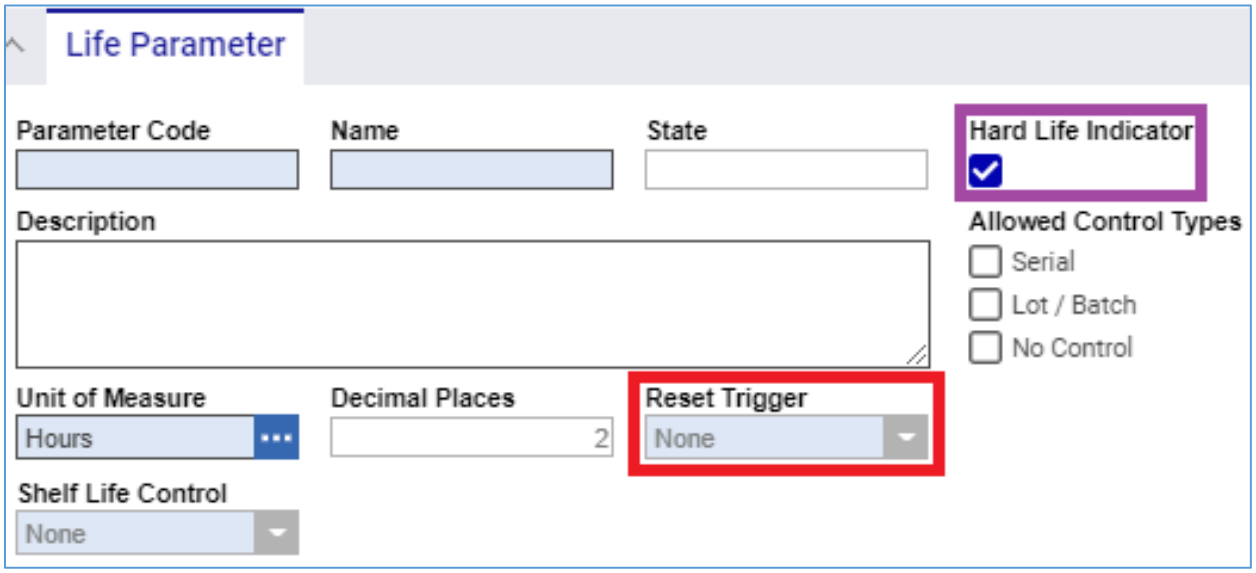


Figure 338.

7.5 Life Parameter Relationships and usage

The **Life Parameter** Item has no **Relationships** accordion because it is a related Item in the **Life Policy Life Parameter** Relationship Item.

To represent an asset life variable, a **Life Parameter** Item should be applied to a **Physical Part** Item representing this asset as follows:

1. This **Life Parameter** Item is included in a **Life Policy** Item. It is shown on the **Life Controls** Relationships tab there. One **Life Parameter** Item can be in many **Life Policy** Items. See section 8.4 Life Policy Relationships and Usage.
2. This **Life Policy** Item is attached to a **Part Policy** Item. The **Life Parameter** Item is shown on the **Life Controls** Relationships tab there. One **Life Policy** Item can be in many **Part Policy** Items. See section 9.4 Part Policy Relationships.
3. This **Part Policy** governs all generations of a **Part** Item. One **Part Policy** Item can be in a relationship with only one **Part** Item. See section 9.4 Part Policy Relationships.
4. The given **Physical Part** Item is related to this **Part** Item. Depending on whether the **Life Parameter** Item is a numeric variable or date, this **Life Parameter** Item is shown on the **Current Life Values** or **Current Date Values** Relationships tab of this **Physical Part** Item. See section 4.7 Tracking lives of Physical Parts.

Use the **Where Used** browser to explore the usage of a **Life Parameter** Item. See section 13 The Where Used Browser and DTC Items.

7.6 Life Parameter Life Cycle and promotion

Note: Only an **Asset Admin** can promote **Life Parameter** Items.

The **LifeParameter** Life Cycle Map defines the following States for the **Life Parameter** Items:

1. **Preliminary**
2. **Active**

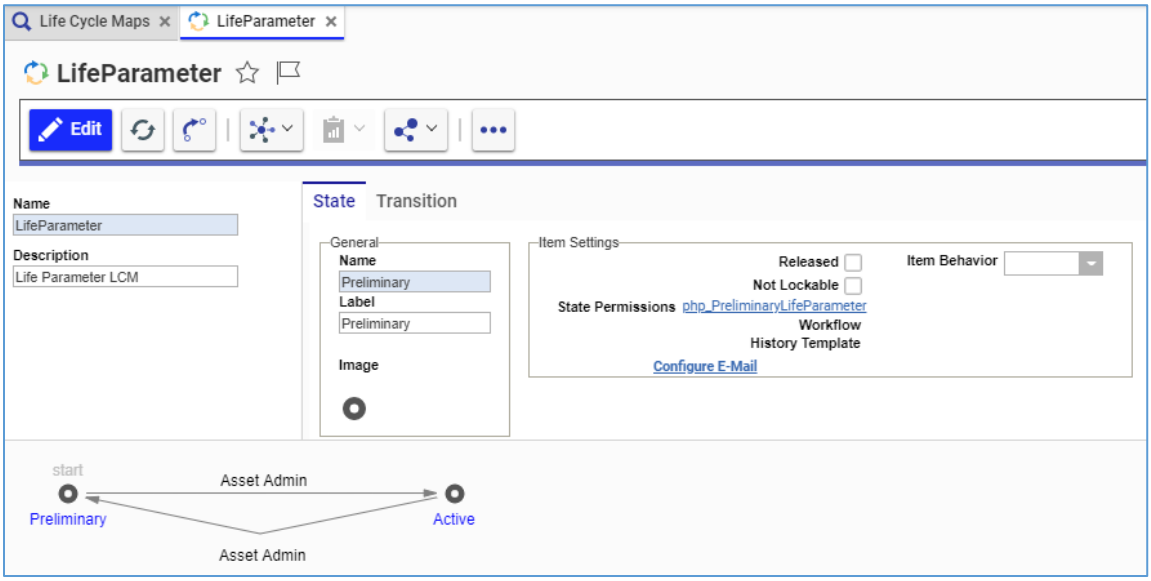


Figure 339.

Use the standard Item promotion procedure to promote the **Life Parameter** Items.

Once created, a new **Life Parameter** Item is in the **Preliminary** State, where it is a draft: a member of the **Asset Admin** Identity has full editing rights for its properties: the **Asset Admin** can create, edit, and delete the **Life Parameter** Item.

Being in the **Preliminary** State, the **Life Parameter** Item cannot be used in a **Life Policy** Item. To become usable, the **Life Parameter** Item must be promoted to the **Active** State when it is ready for usage and not supposed to be changed.

Editing and deleting a **Life Parameter** Item in the **Active** State is not allowed for any Identity.

The **Life Parameter** Item can be promoted back to the **Preliminary** State only it if it is not used in a **Life Policy** Item. For details on the **Life Parameter** Item usage, refer to section 7.5 Life Parameter Relationships and usage.

An error is raised if promoting a used **Life Parameter** Item back to the **Preliminary** State.

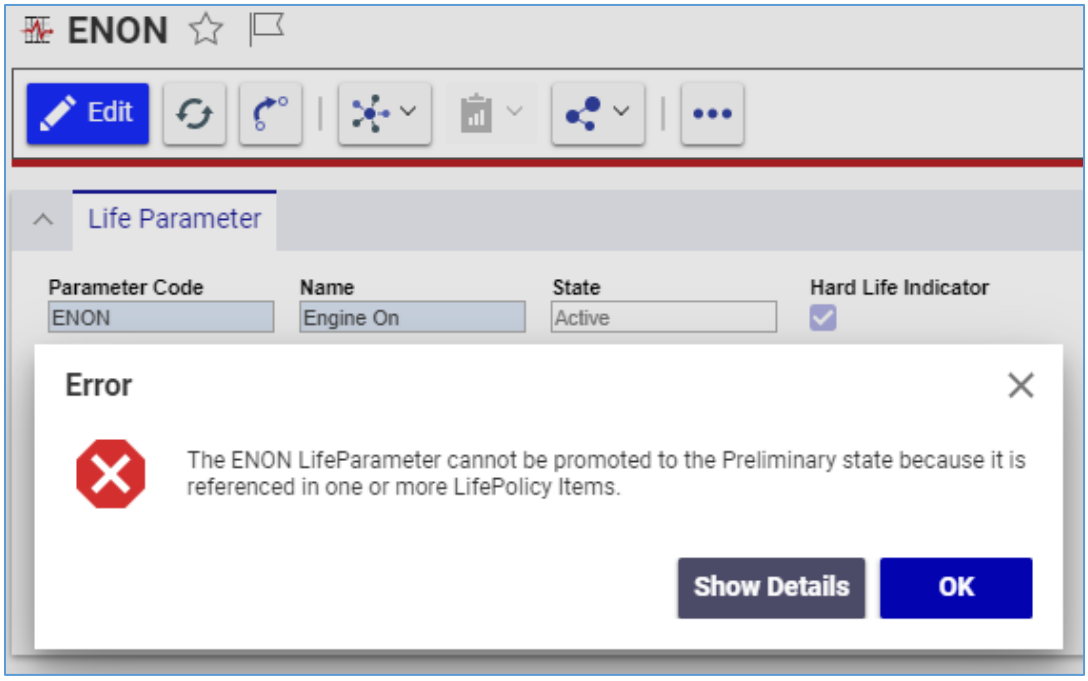


Figure 340.

7.7 Managing Life Parameters

Your ability to manage a **Life Parameter** Item depends on your Asset Identity membership, current Item State, and its Relationships. This section assumes that you are familiar with the **Life Parameter** Item properties and their behavior; discussed in sections 7.3 Life Parameter Properties and 7.4 Behavior of Life Parameter properties.

7.7.1 Creating Life Parameters

Note: Only an **Asset Admin** can create the **Life Parameter** Items.

To create a new **Life Parameter** Item:

1. Go to **Contents** --> **Assets** --> **Life Parameters**.

2. Click **Create Life Parameter**. The **Life Parameter** # Item view appears.

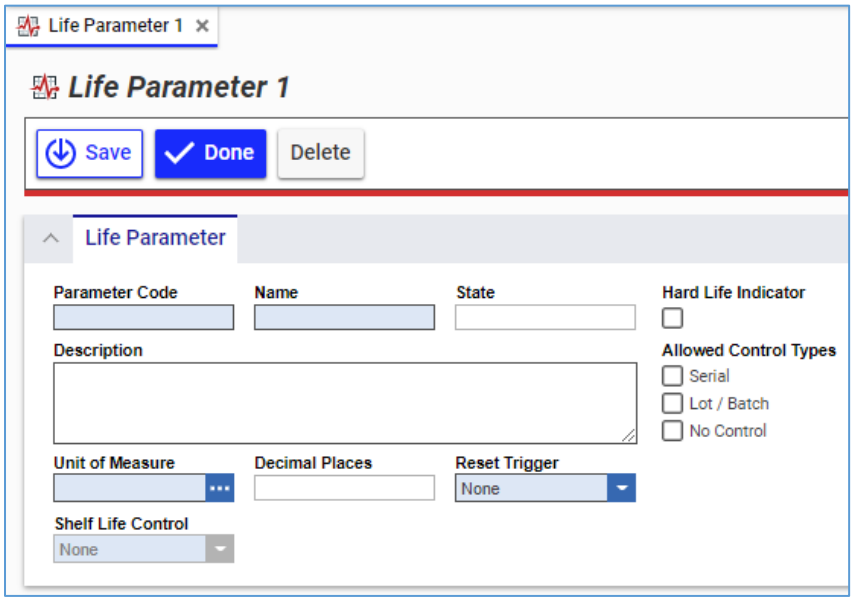


Figure 341.

3. In the **Parameter Code** field, enter a unique identifier for the **Life Parameter** Item.

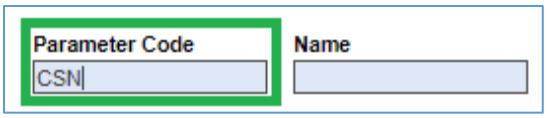


Figure 342.

4. In the **Name** field, enter a name for the **Life Parameter** Item.

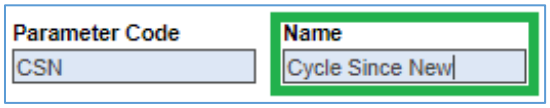


Figure 343.

5. If the **Life Parameter** Item sets a hard life limit, select the **Hard Life Indicator** check box. For the **Hard Life Indicator** property availability, see section 7.4 Behavior of Life Parameter properties.

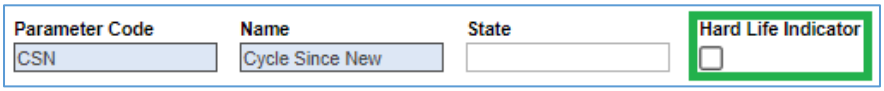


Figure 344.

6. In the **Description** field, enter additional or explanatory information on the **Life Parameter** Item.

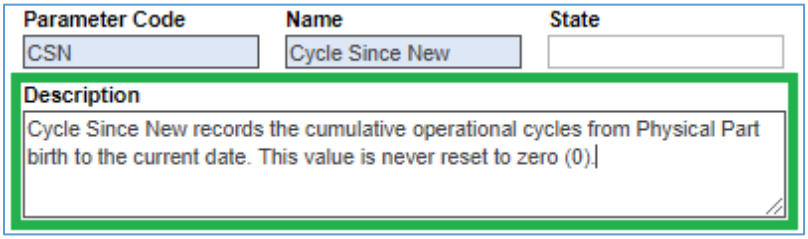


Figure 345.

7. In the **Allowed Control Type** group, select check boxes for the control types to which the **Life Parameter** Item will be applicable.

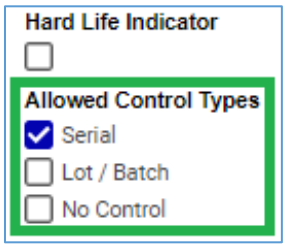


Figure 346.

8. Click the **ellipsis** button in the **Unit of Measure** field. The **Select Items – Life Units** dialog appears.

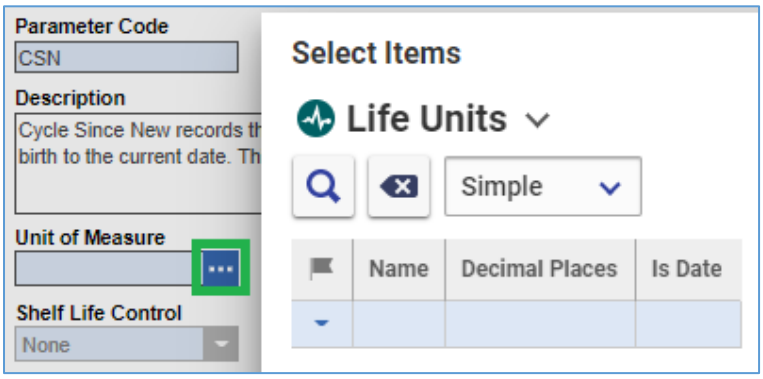


Figure 347.

9. Using the standard search procedure, search for the **Life Unit** Item by which the **Life Parameter** Item will be tracked.

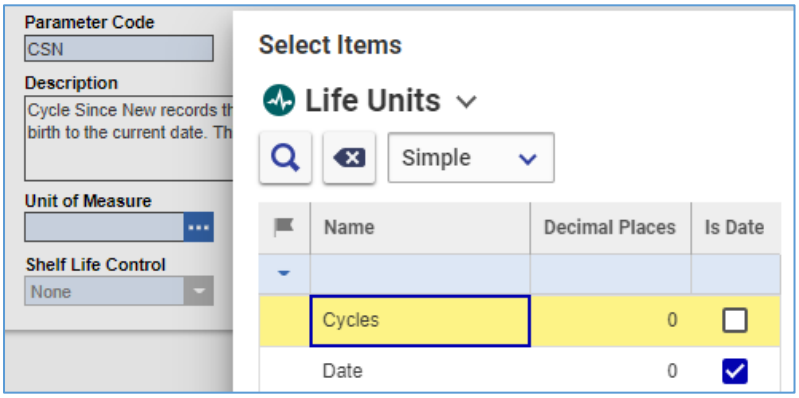


Figure 348.

10. Select this **Life Unit** Item and click **OK**. The selected **Life Unit** Item populates the **Unit of Measure** field. The **Decimal Places** field on the **Life Parameter** Item form is automatically sourced from the **Decimal Places** property of the **Life Unit** Item.

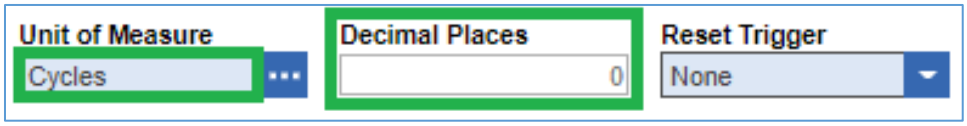


Figure 349.

Note: You can also specify the **Life Unit** Item by typing in the **Unit of Measure** field.

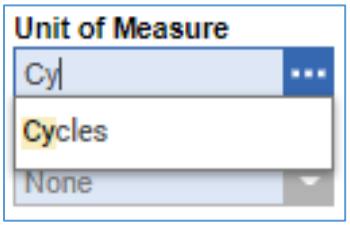


Figure 350.

11. If the **Reset Trigger** field is available, select whether the current value of the **Life Parameter** Item can be reset (**Yes**) or not (**None**). For the **Reset Trigger** availability, see section 7.4 Behavior of Life Parameter properties.

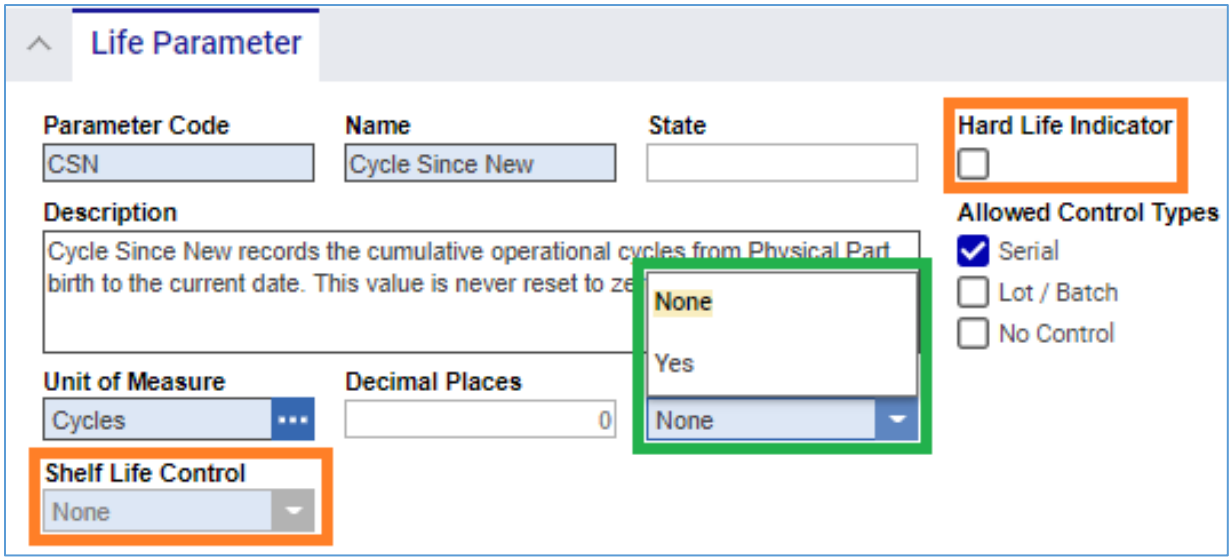


Figure 351.

12. If the **Shelf Life Control** field is available, select there whether an expiry date defined by the **Life Parameter** Item can be reset (**Recertifiable**) or not (**Not Recertifiable**). For the **Shelf Life Control** availability, see section 7.4 Behavior of Life Parameter properties.

Figure 352.

13. Click either **Save** or **Done** on the **Life Parameter** Item toolbar. The new **Life Parameter** Item is created and in the **Preliminary** State.

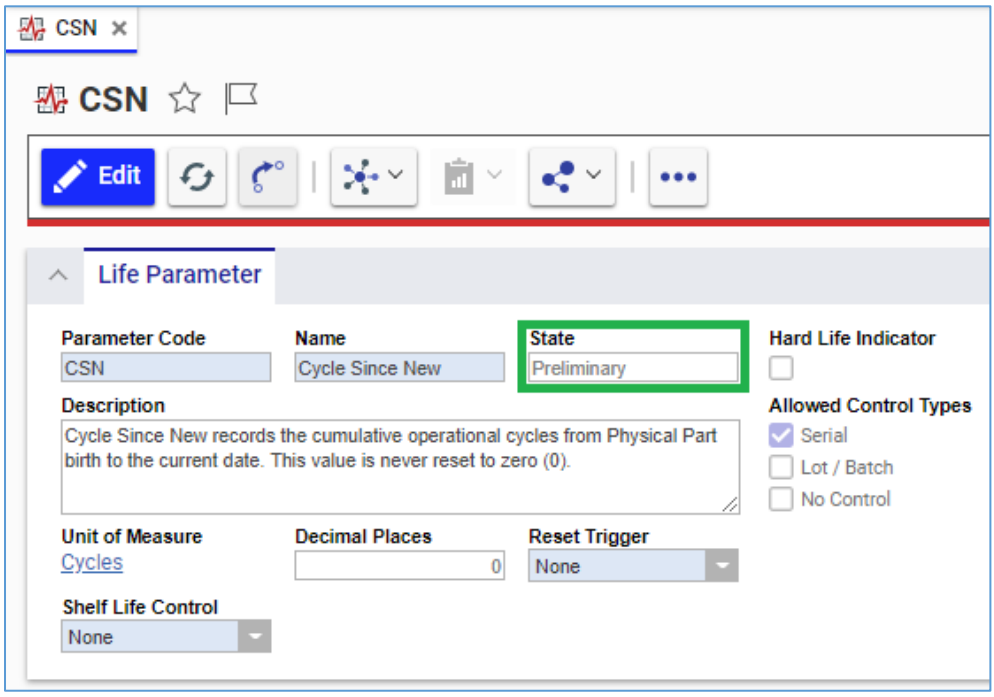


Figure 353.

The **Life Parameter** Item **Innovator tab** and **Page Title** bar show the Item **Parameter Code** value.

Figure 354.

The **Life Parameter** Item is now available via search, subject to the **Asset** Permissions.

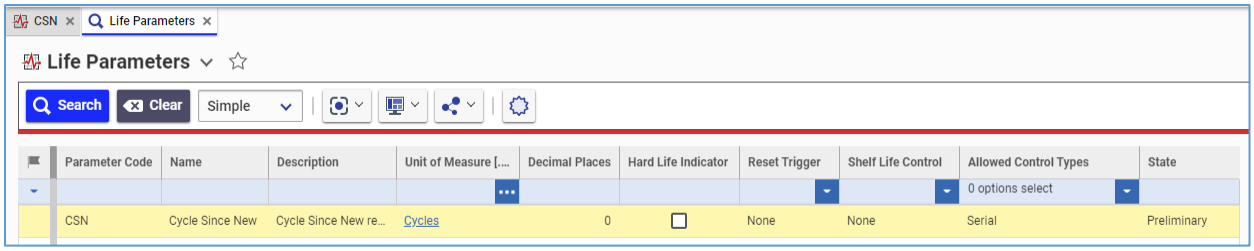


Figure 355.

Upon saving a **Life Parameter** Item, the system validates this Item and raises an intelligent error if the validation fails, for example, if the Item has a blank field for a required property or a duplicating **Parameter Code** value, discussed in section 7.7.4 Validation of Life Parameters.

7.7.2 Modifying Life Parameters

Note: Only an **Asset Admin** can modify properties of a **Life Parameter** Item and only when this Item is in the **Preliminary** State. If otherwise, clicking the **Edit** button on the **Life Parameter** Item toolbar raises an error.

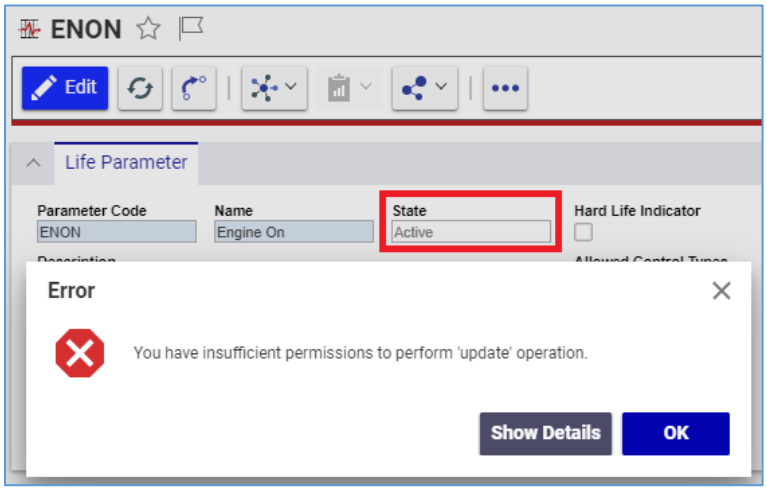


Figure 356.

To modify a **Life Parameter** Item, use the standard modifying procedure.

Upon saving a **Life Parameter** Item, the system validates this Item and raises an intelligent error if the validation fails, for example, if the Item has a blank field for a required property or a duplicating **Parameter Code** value, discussed in section 7.7.4 Validation of Life Parameters.

7.7.3 Deleting Life Parameters

Note: Only an **Asset Admin** can delete a **Life Parameter** Item and only when this Item is in the **Preliminary** State. If otherwise, an error is raised.

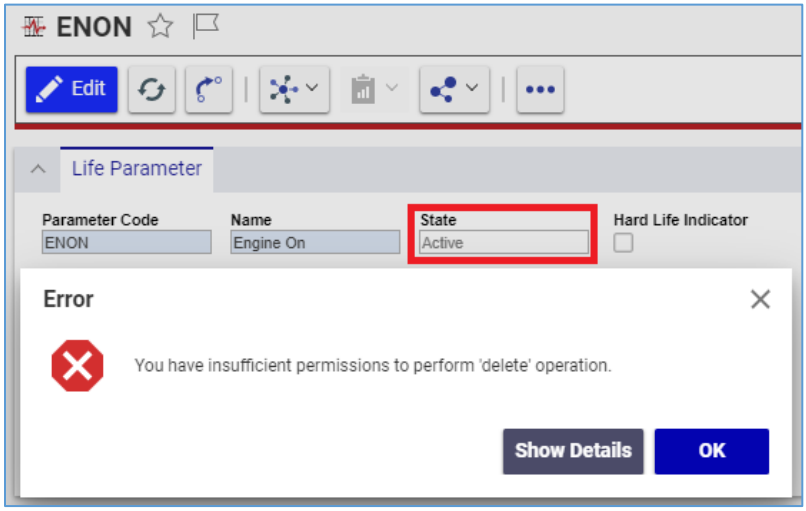


Figure 357.

To delete a **Life Parameter** Item, use the standard deletion procedure.

7.7.4 Validation of Life Parameters

Once you click **Save** or **Done** on a **Life Parameter** Item toolbar, the system validates this Item and raises an appropriate error if the Item:

- Is missing a **Parameter Code** value.

Figure 358.

- Is missing a **Name** value.

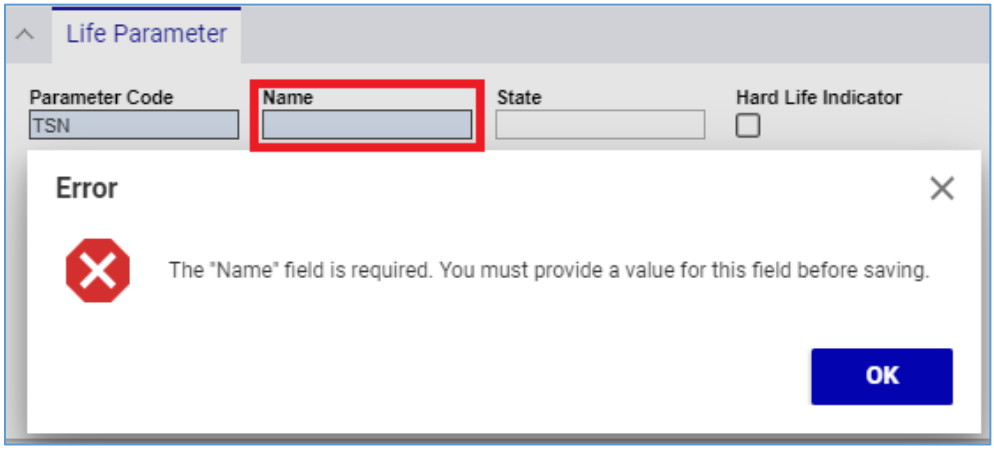


Figure 359.

- Is missing a **Unit of Measure** value.

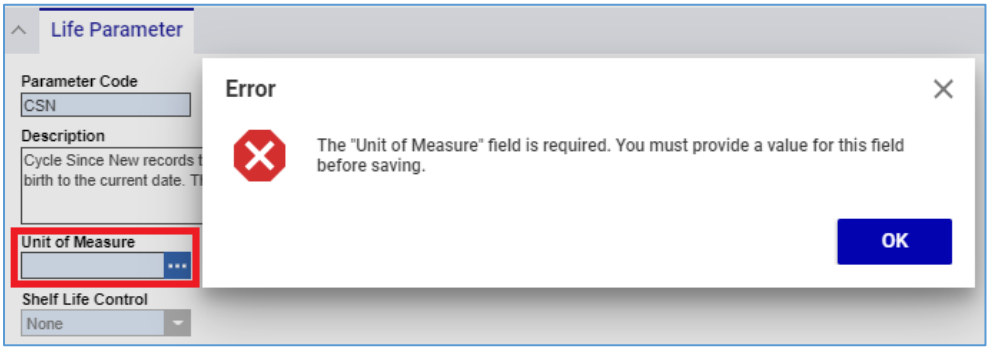


Figure 360.

- Is missing an **Allowed Control Type** value.

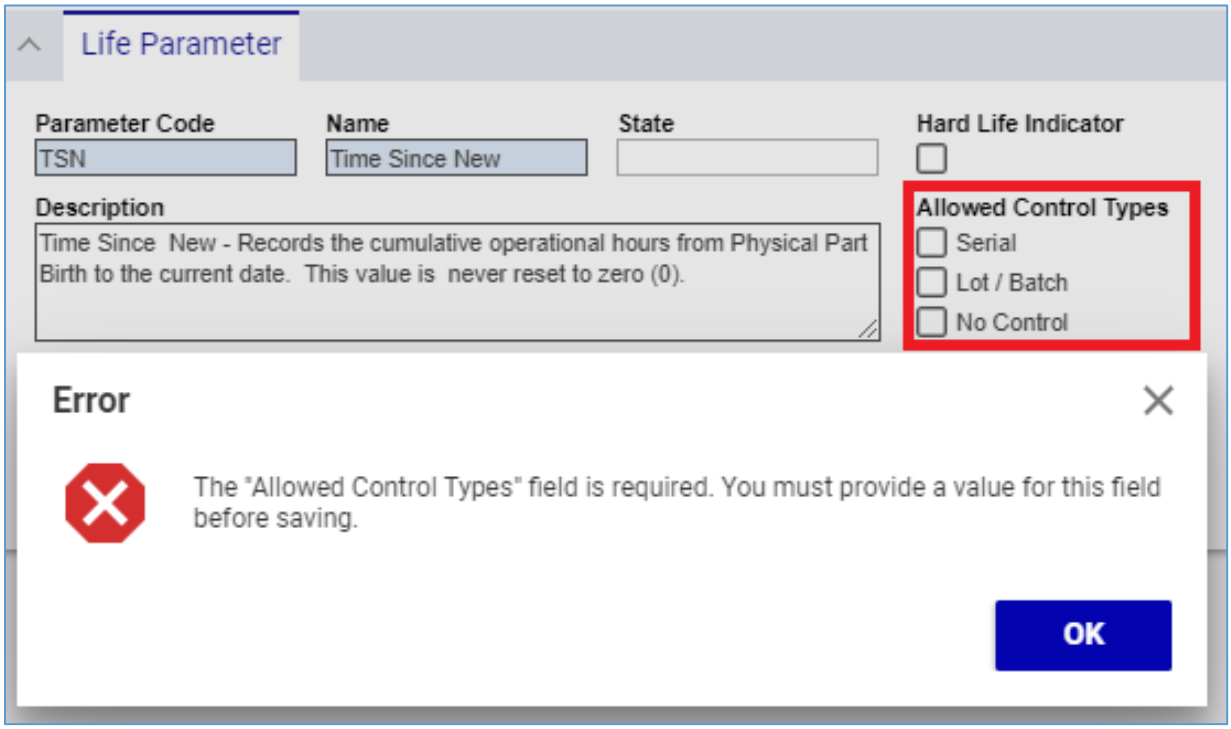


Figure 361.

- Has **Parameter Code** value that is not unique: already given to an existing **Life Parameter**.

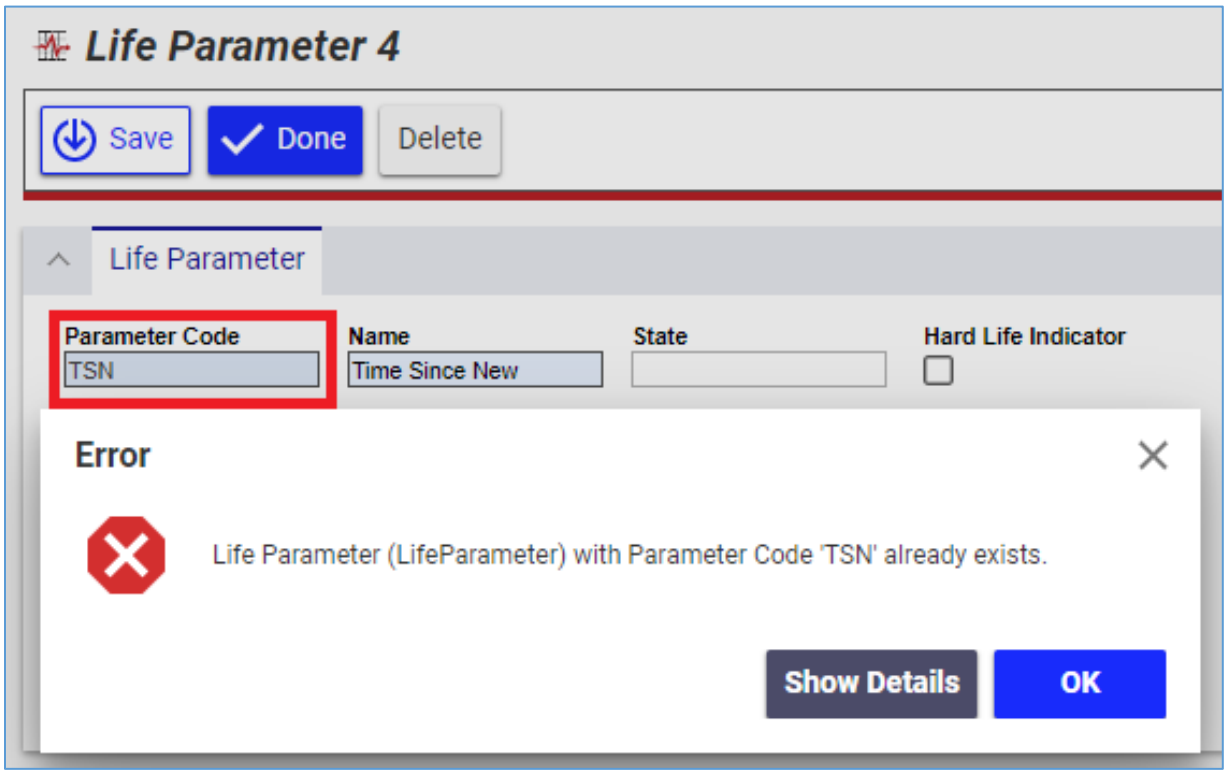


Figure 362.

- Has a numeric **Life Unit** Item set in its **Unit of Measure** property and the **Lot / Batch** or **No Control** option selected in its **Allowed Control Types** property.

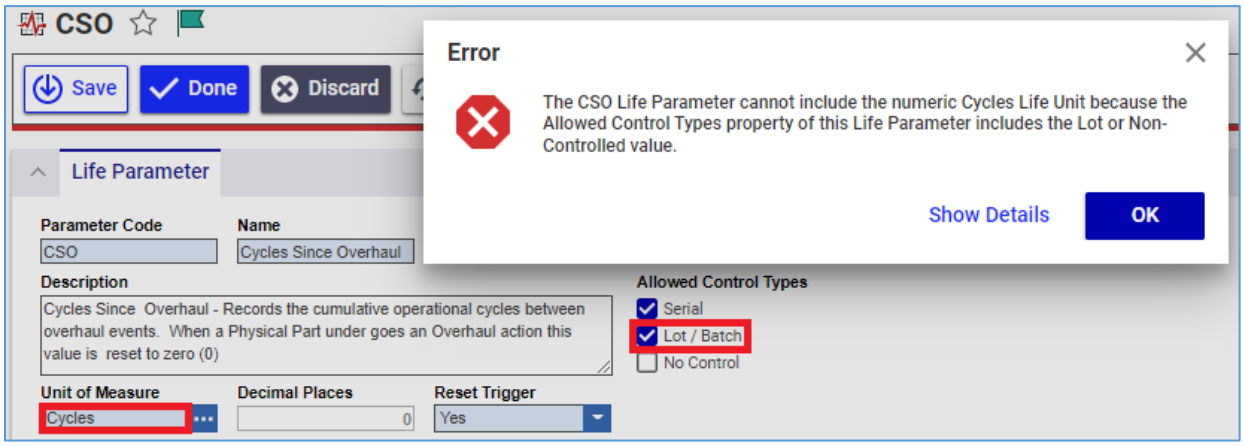


Figure 363.

8 Life Policies

Having all necessary Life Parameters, it is time to create Life Policies. Once they are in place it is time to create the Part Policies.

8.1 Understanding Life Policies

An organization may require all its real-word assets of the same kind, series, part number, or some specific characteristic to be tracked similarly. More than one life variable of an asset may change while the asset is operating in the field, and each variable may require tracking.

A **Life Policy** Item defines a set of **Life Parameter** Items that will track corresponding current values of the **Physical Part** Items where this **Life Policy** Item applies; see section 7.5 Life Parameter Relationships and usage. The **Life Policy** Item enables tracking of the **Physical Part** Items of a given kind in the same manner across an organization. For example, the **Standard Serial Control** Life Policy determines tracking of the serial-controlled **Physical Part** Items by the **TSN** (**Time Since New**), **TSO** (**Time Since Overhaul**), and **TSR** (**Time Since Repair**) Life Parameters.

Note: The DTC application is shipped with no **Life Policy** Items out of the box. This section uses some industry-agnostic **Life Policy** Items as illustrations.

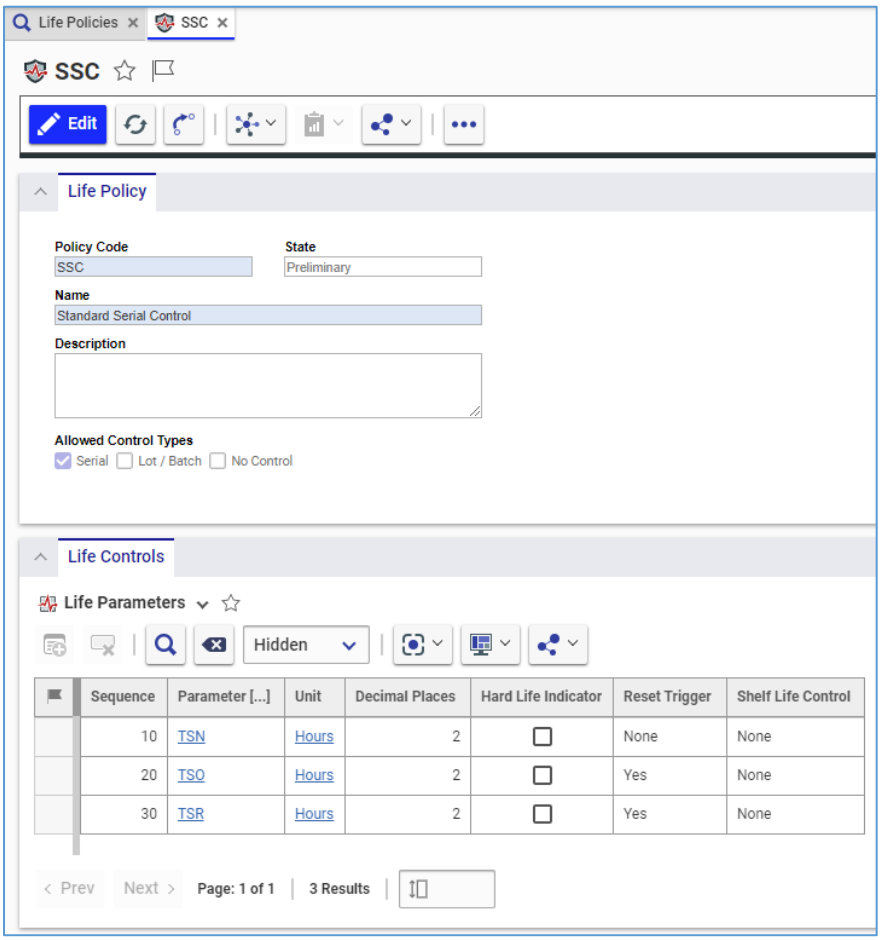


Figure 364.

The **Life Policy** Item gathers the life variables to be tracked on its **Life Control** tab grid as Relationships with the akin **Life Parameter** Items. For details, see section 8.4 Life Policy Relationships and usage.

For details on the **Life Policy** Item properties, see section 8.3 Life Policy Properties.

It is not possible to modify or delete an applied **Life Policy** Item. For details on using Life Policies, refer to section 8.4 Life Policy Relationships and usage.

8.2 Life Policies Search Grid

The **Life Policies** Search Grid has the standard features and the following columns:

- **Policy Code**
- **Name**
- **Description**
- **Allowed Control Types**
- **State**

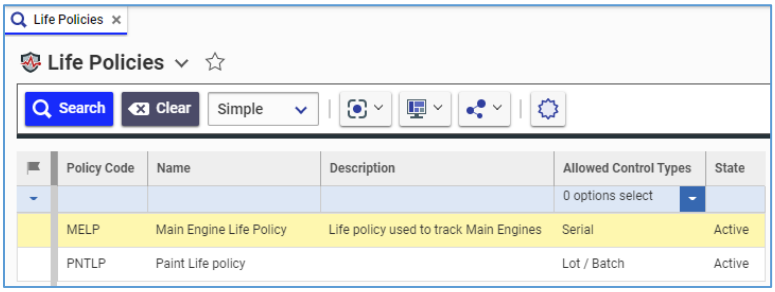


Figure 365.

Each column shows a matching **Life Policy** Item property. For the property details, see section 8.3 Life Policy Properties.

The **Life Policy** Items are sorted in the Search Grid and other grids by their **Policy Code** values in the ascending order (from A to Z).

8.3 Life Policy Properties

The following properties of a **Life Policy** Item are available in the **Form** accordion of this **Life Policy** Item view:

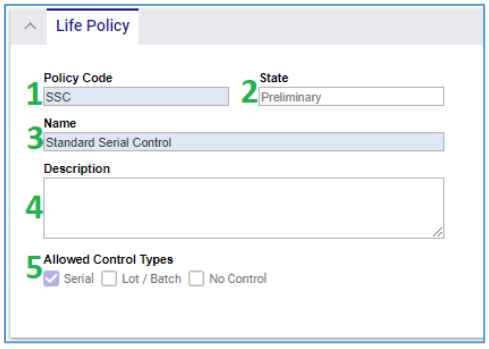


Figure 366.

1. **Policy Code**: the unique identifier for a given **Life Policy** Item among other Life Policies. This property is required and must have a unique value of up to 32 characters long. The characters can be of any type. It is recommended to use an abbreviation or shortened version of the Item **Name** property as its **Parameter Code**.
2. **State**: the current State of the **Life Policy** Item. See section 8.5 Life Policy Life Cycle and promotion.
3. **Name**: the full name of the **Life Policy** Item. This property is required and must have a value of up to 64 characters long. The characters can be of any type. The **Name** should briefly explain where or how this **Life Policy** Item should apply.
4. **Description**: detailed or additional information about the **Life Policy** Item. This property can have a value of up to 256 characters long. The characters can be of any type.
5. **Allowed Control Types**: a group of check boxes where a selected (**true**) check box defines a control type to which the given **Life Policy** Item is applicable. This property is required. For details, see section 8.4 Life Policy Relationships and Usage.

8.4 Life Policy Relationships and Usage

A **Life Policy** Item has the **Life Controls** Relationships tab for **Life Parameter** Items that will track the current matching values of the **Physical Part** Items related to this **Life Policy** Item. This tab is a single-level Relationships editor that enables members of dedicated Asset Identities to view and manage the tracking **Life Parameter** Items. For more details, see section 8.6 Managing Life Policies.

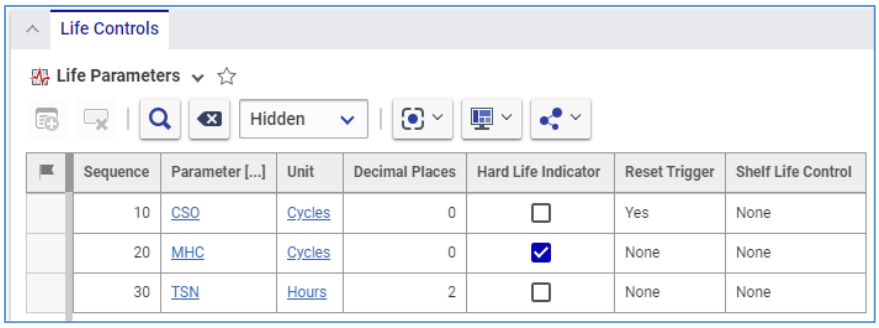


Figure 367.

This tab has the **Life Controls Life Parameters** Grid that shows included **Life Parameter** Items as follows:

- **Sequence**: a sort order number of a given **Life Parameter** Item in the Grid.
- **Parameter**: a view-only **Parameter Code** property value of the **Life Parameter** Item. It is also a link to this **Life Parameter** Item.
- **Unit**: a view-only **Life Unit** Item given in the **Unit of Measure** property of the **Life Parameter** Item. It is also a link to this **Life Unit** Item.
- **Decimal Places**: a view-only **Decimal Places** property value of the given **Life Unit** Item.
- **Hard Life Indicator**: a view-only **Hard Life Indicator** property value of the **Life Parameter** Item.
- **Reset Trigger**: a view-only **Reset Trigger** property value of the **Life Parameter** Item.
- **Shelf Life Control**: a view-only **Shelf Life Control** property value of the **Life Parameter** Item.

In the grid, the **Life Parameter** Items are sorted by their **Sequence** values in the ascending order (from 0 to 9).

The **Allowed Control Types** property of a **Life Policy** Item defines which **Life Parameter** Items this **Life Policy** Item can include and to which **Part Policy** Items it can be applied. One **Life Policy** Item can be applied to many **Part Policies**, thus **Physical Part** Items, within its **Allowed Control Types**.

Only the **Life Parameter** Items of all the **Life Policy** Item control types can populate the **Life Controls** tab grid. For example, **Life Parameter** Items must have both **Serial** and **Lot/Batch** specified in their **Allowed Control Types** properties to be added to a **Life Policy** Item with **Serial** and **Lot/Batch** selected in its **Allowed Control Types** property. The **Life Parameter** Items may or may not have **No Control** specified in such a case.

When adding **Life Parameter** Items to the **Life Policy** Item, the **Select Items – Life Parameters** dialog shows and allows to select only **Life Parameter** Items that include all **Life Policy** Item **Allowed Control Types** even if this **Life Policy** Item is not saved to the system.

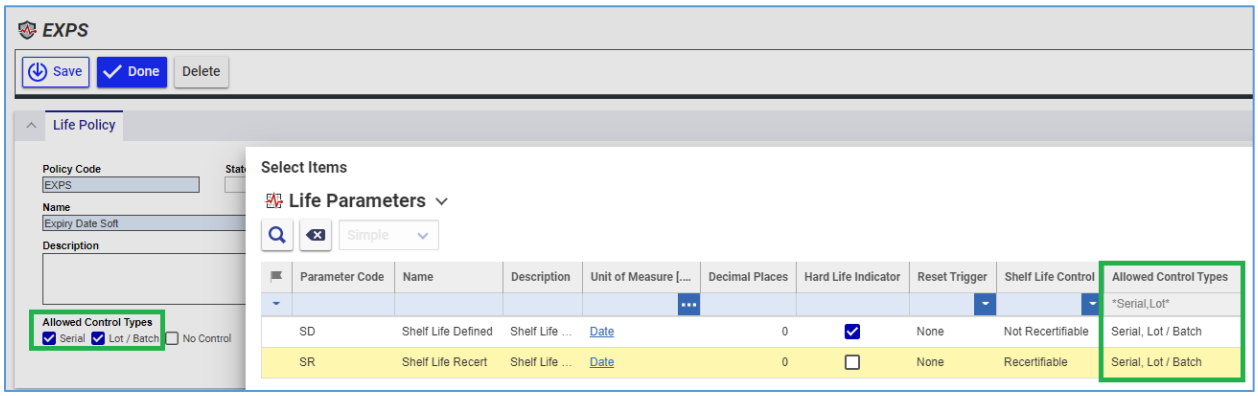


Figure 368.

If the **Allowed Control Types** property of the **Life Policy** Item has been modified after the **Life Parameter** Items had populated its **Life Controls** tab grid, the system validates these **Life Parameter** Items against the current **Allowed Control Types** property setting and does not save the **Life Policy** Item if there are one or more **Life Parameter** Items of a currently unallowed control type.

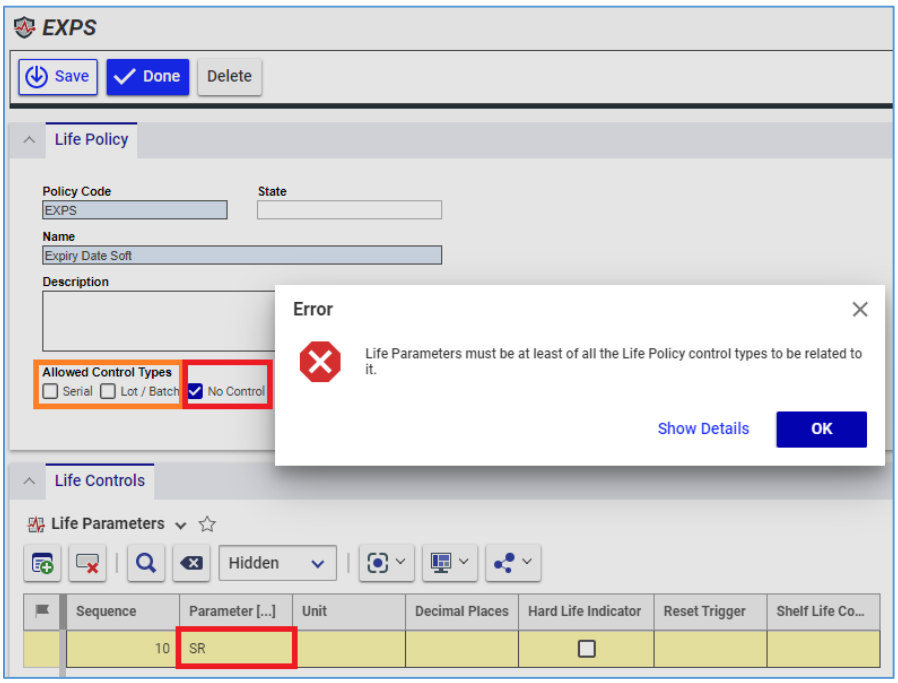


Figure 369.

The **Life Policy** Item should be attached to one or more **Part Policy** Items to track **Physical Part** Items by the included **Life Parameter** Items. See section 9.4 Part Policy Relationships.

A **Life Policy** Item can be applied only to a **Part Policy** Item for a **Part** Item of any control type allowed in this **Life Policy** Item. For example, if a **Life Policy** Item has **Serial** and **Lot/Batch** selected in its **Allowed Control Types** property, it can be related to a **Part Policy** Item where a serial-controlled or lot-controlled **Part** Item is given. It cannot be related to a **Part Policy** Item for a non-controlled **Part** Item. Consequently, the **Life Policy** is applicable only to serial-controlled or lot-controlled **Physical Part** Items.

Use the **Where Used** browser to explore the usage of a **Life Policy** Item. See section 13 The Where Used Browser and DTC Items. Also, you can filter the **Part Policy** Item Search Grid by the given **Life Policy** Item.

8.5 Life Policy Life Cycle and Promotion

Note: Only an **Asset Admin** can promote **Life Policy** Items.

The **LifePolicy** Life Cycle Map defines the following States for the **Life Policy** Items:

1. **Preliminary**
2. **Active**

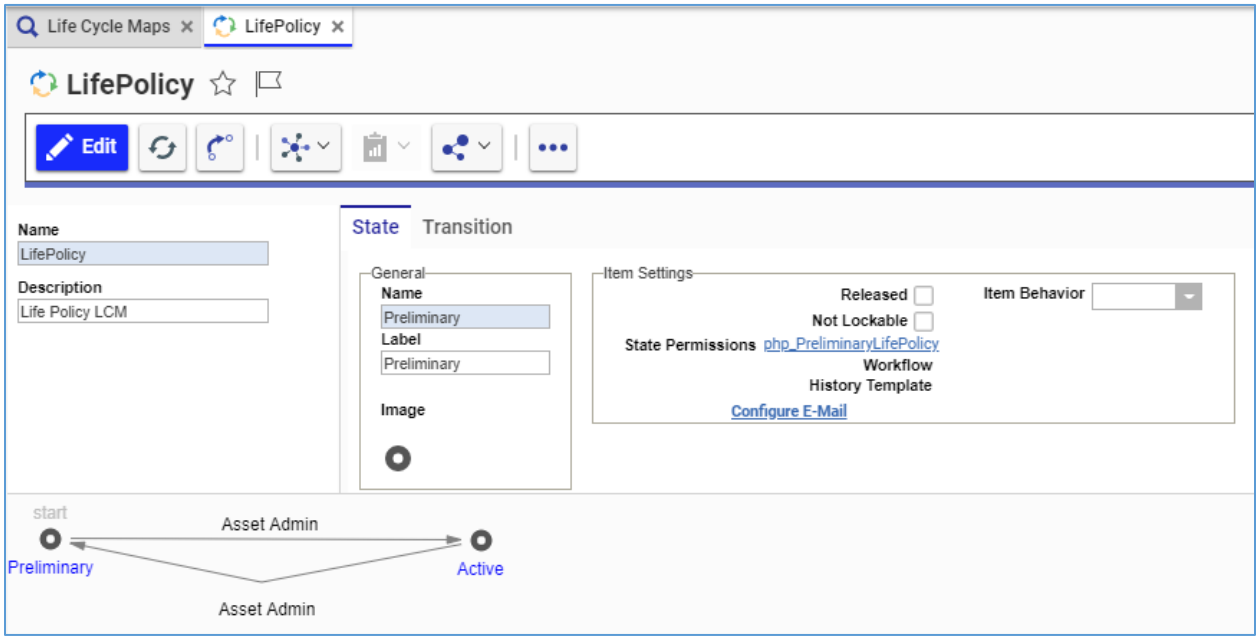


Figure 370.

Use the standard Item promotion procedure to promote the **Life Policy** Items.

Once created, a new **Life Policy** Item is in the **Preliminary** State, where it is a draft: a member of the **Asset Admin** Identity has full editing rights for its properties: the **Asset Admin** can create, populate with **Life Parameter** Items, edit, and delete the **Life Policy** Item.

Being in the **Preliminary** State, the **Life Policy** Item cannot be used in a **Part Policy** Item. To become usable, the **Life Policy** Item must be promoted to the **Active** State when it is ready for usage and not supposed to be changed.

Editing, populating, and deleting a **Life Policy** Item in the **Active** State is not allowed for any Identity.

The **Life Policy** Item can be promoted back to the **Preliminary** State only if it is not used in a **Part Policy** Item. For details on the **Life Policy** Item usage, refer to section 8.4 Life Policy Relationships and Usage.

An error is raised if promoting a used **Life Policy** Item back to the **Preliminary** State.

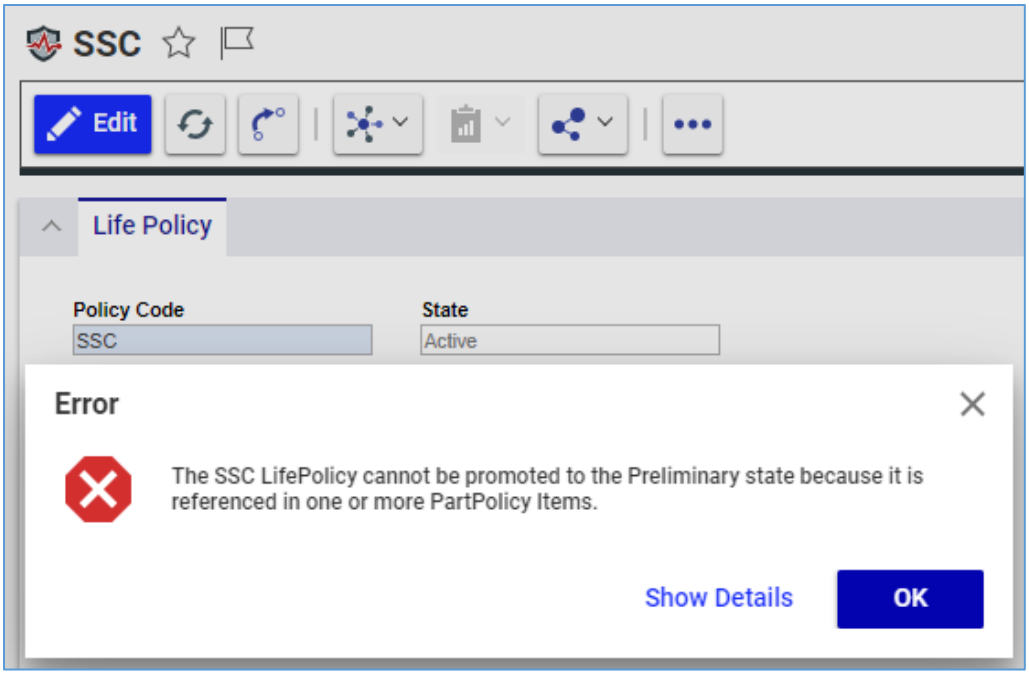


Figure 371.

8.6 Managing Life Policies

Your ability to manage a **Life Policy** Item depends on your Asset Identity, current Item State, and Item Relationships.

8.6.1 Creating Life Policies

Note: Only an **Asset Admin** can create the **Life Policy** Items.

To create a new **Life Policy** Item:

1. Go to **Contents** --> **Assets** --> **Life Policies**.
2. Click **Create Life Policy**. The **Life Policy #** Item view appears.

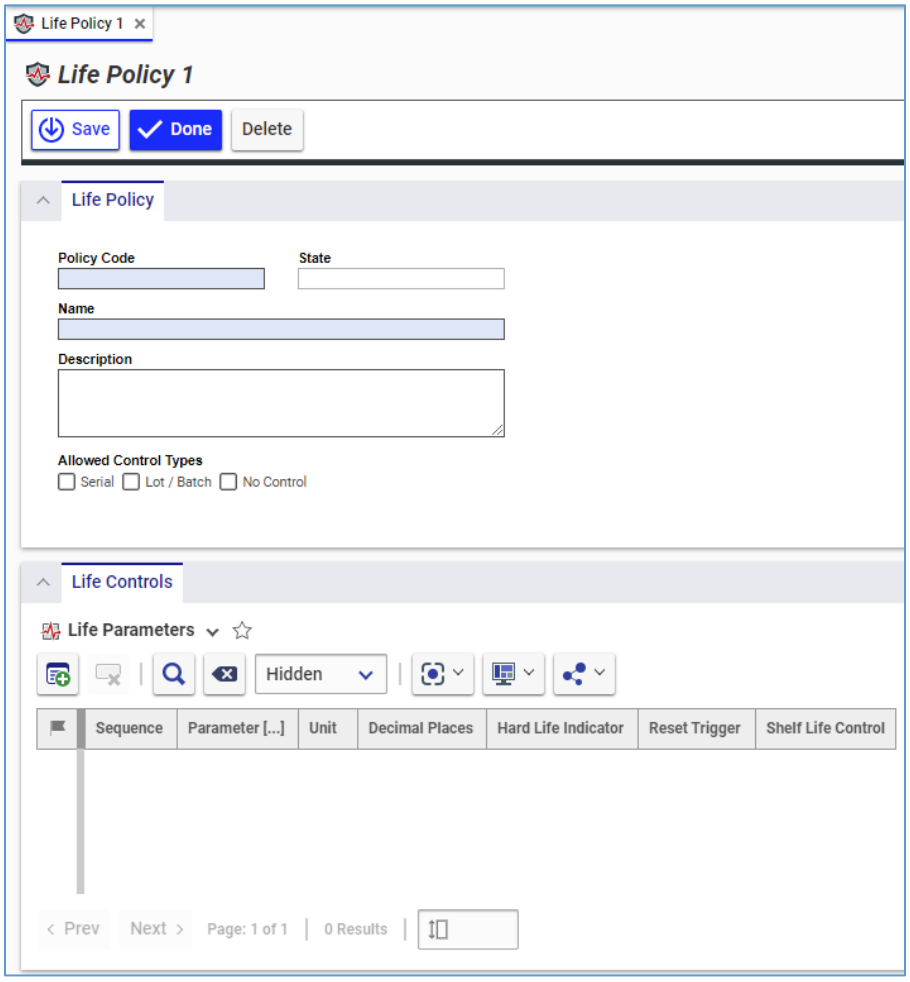


Figure 372.

3. In the **Policy Code** field, enter a unique identifier for the **Life Policy** Item.

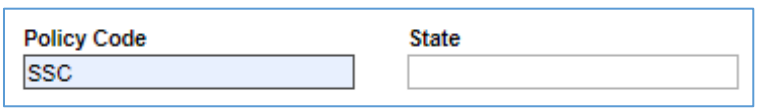


Figure 373.

4. In the **Name** field, enter a name for the **Life Policy** Item.

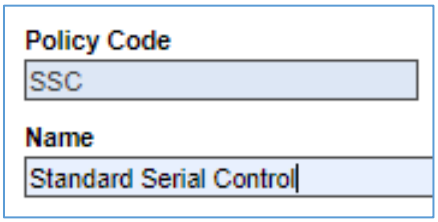


Figure 374.

5. In the **Description** field, enter additional or explanatory information on the **Life Policy** Item.

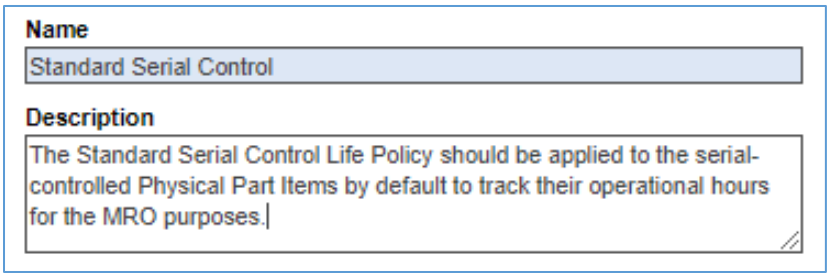


Figure 375.

6. In the **Allowed Control Type** group, select check boxes of the control types to which the **Life Policy** Item will be applicable.

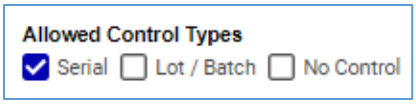


Figure 376.

7. Go to the **Life Controls** Relationships accordion tab.
8. Click **Add Life Parameters** on the **Life Controls Life Parameters** Grid toolbar.

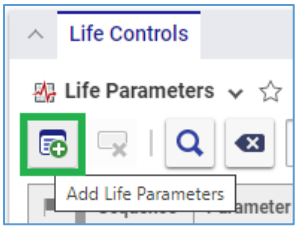


Figure 377.

The **Select Items – Life Parameters** dialog pops up.

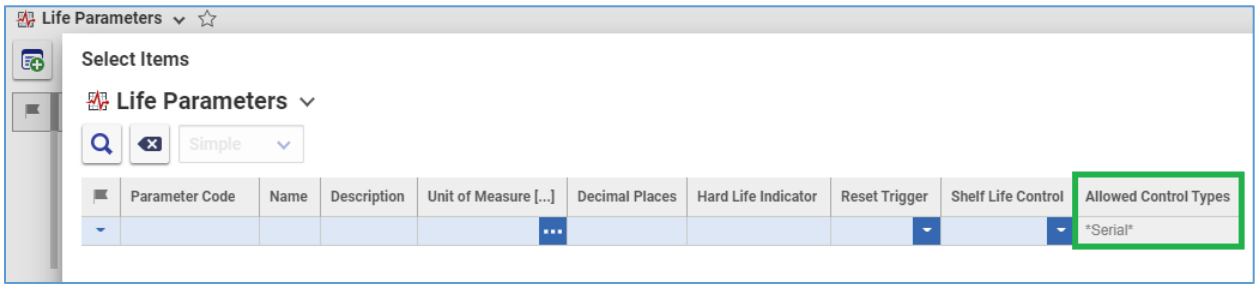


Figure 378.

9. Using the standard search procedure, search for, and select all the necessary **Life Parameter** Items. The selected Items populate the **Life Controls Life Parameters** Grid.

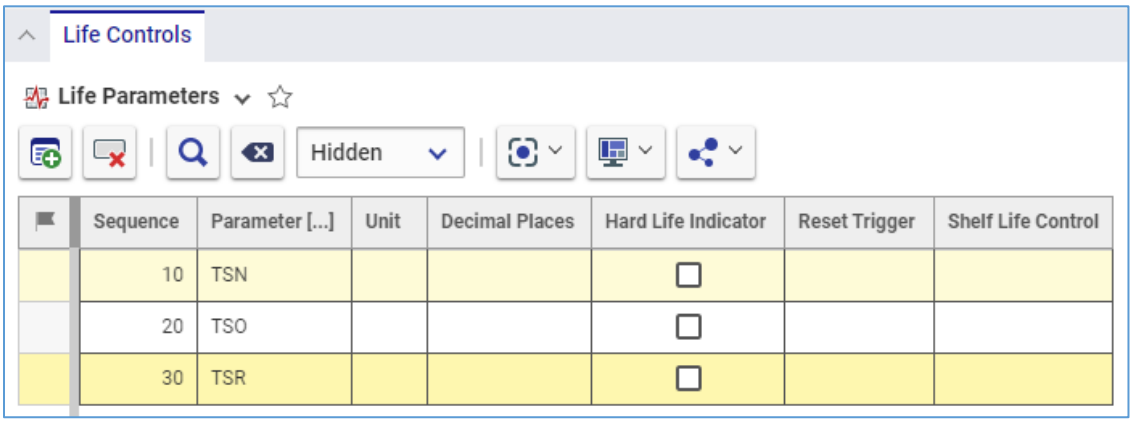


Figure 379.

Note: In the **Select Items – Life Parameters** dialog, you may see new, unsaved **Life Parameter** Items that are being created by you or someone else and have no **Allowed Control Types** specified yet. Please do not select these **Life Parameter** Items and remove them from the **Life Controls** Grid if you have already added them because they can stop saving the new **Life Policy** Item.

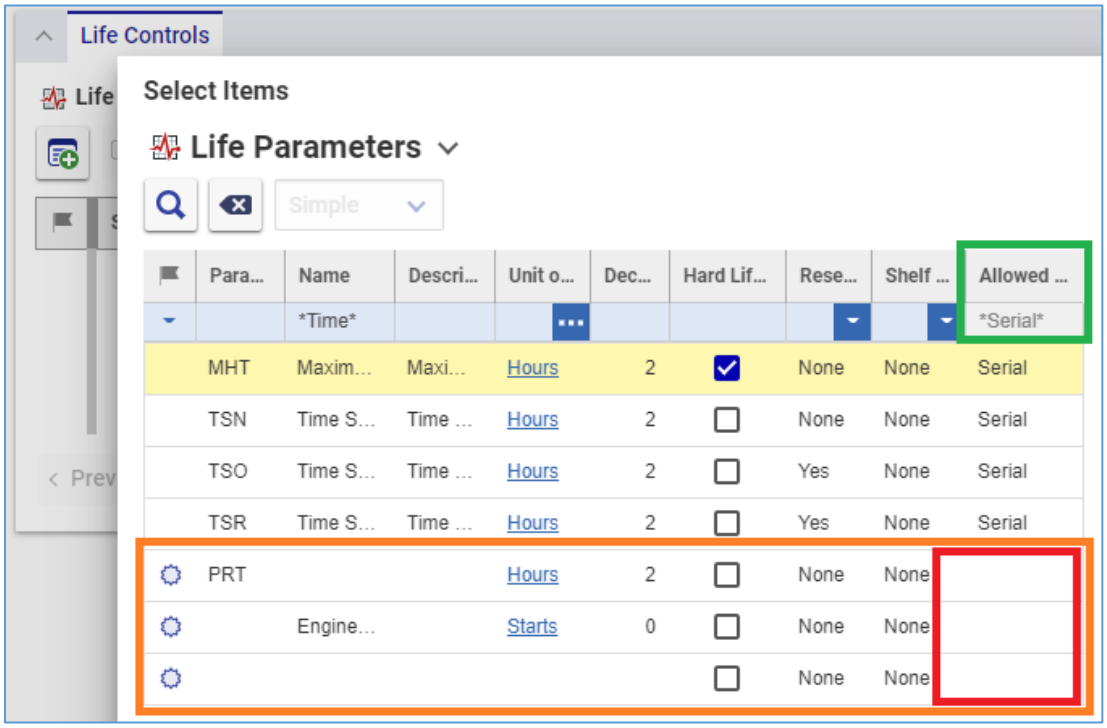


Figure 380.

10. If it is necessary, click the **Sequence** cell and correct its value by typing.

Note: You should assign the **Sequence** values as numbers divisible by ten. For more details on this topic, refer to section 3.6 Sequence Values in the DTC Relationship Grids.

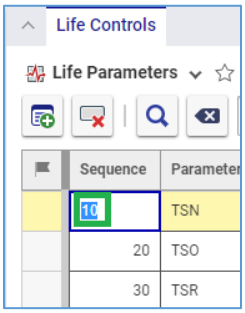


Figure 381.

11. Click either **Save** or **Done** on the **Life Policy** Item toolbar. The new **Life Policy** Item is created and in the **Preliminary** State.

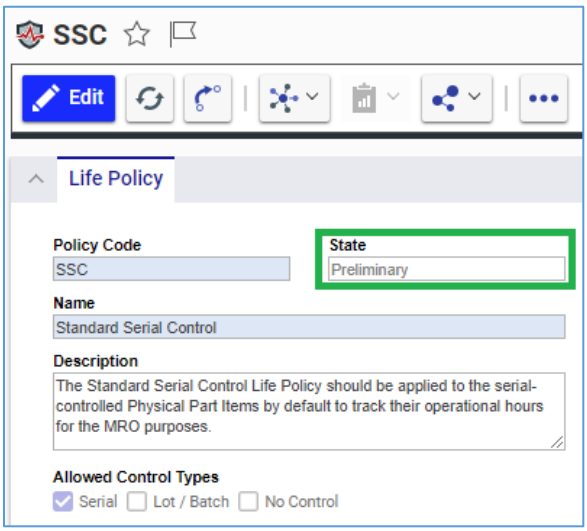


Figure 382.

The **Decimal Places**, **Hard life Indicator**, **Reset Trigger**, and **Shelf Life Control** values of the added **Life Parameter** Items automatically populate the **Life Controls** Grid.

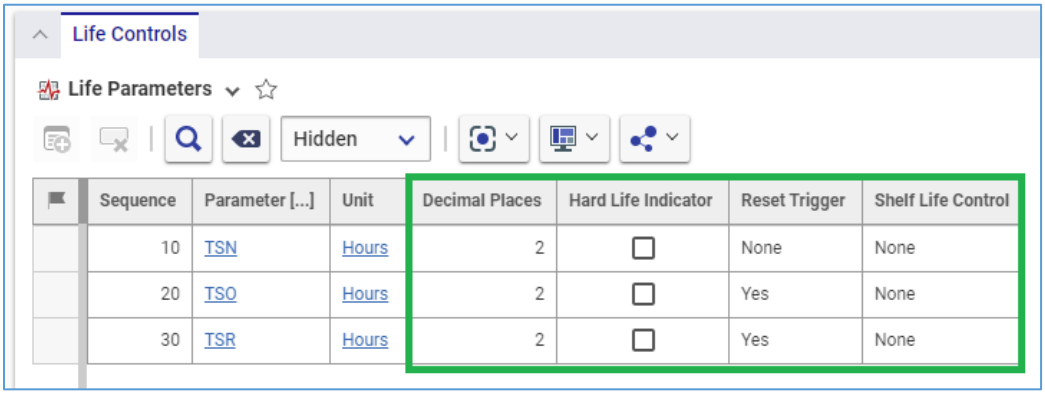


Figure 383.

The **Life Policy** Item **Innovator tab** and **Page Title** bar show the Item **Policy Code** value.

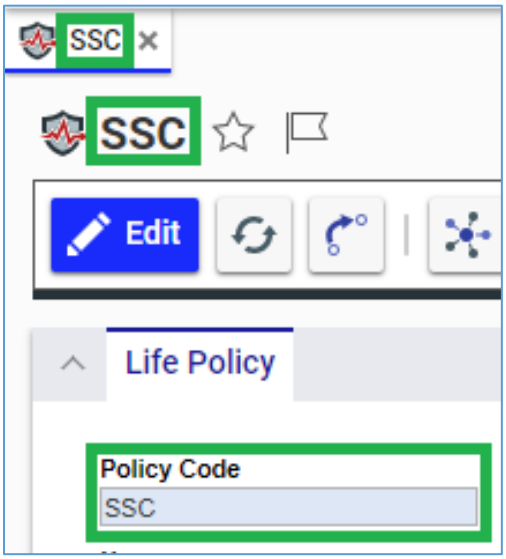


Figure 384.

The **Life Policy** Item is now available via search, subject to the **Asset** Permissions.

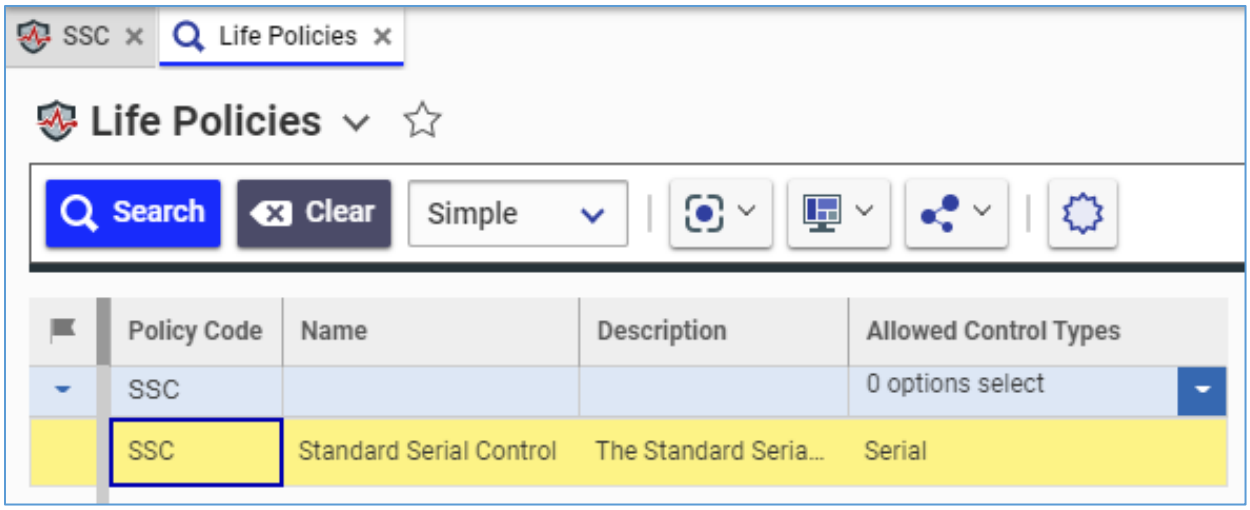


Figure 385.

Upon saving a **Life Policy** Item, the system validates this Item and raises an intelligent error if the validation fails, for example, if the Item has a blank field for a required property or a duplicating **Policy Code** value, discussed in section 8.6.4 Validation of Life Policies.

8.6.2 Modifying Life Policies

Note: Only an **Asset Admin** can modify properties of a **Life Policy** Item and only when this Item is in the **Preliminary** State. If otherwise, clicking the **Edit** button on the **Life Policy** Item toolbar raises an error.

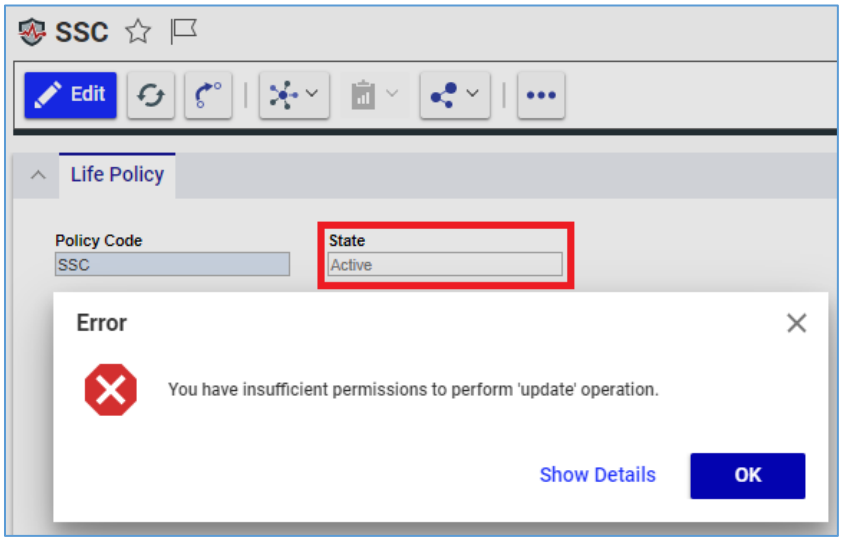


Figure 386.

To modify a **Life Policy** Item, use the standard modifying procedure.

If the **Policy Code** value has been changed, the **Life Policy** Item **Innovator tab** and **Page Title** bar show its new value.

Figure 387.

Upon saving a **Life Policy** Item, the system validates this Item and raises an intelligent error if the validation fails, for example, if the Item has a blank field for a required property or a duplicating **Policy Code** value, discussed in section 8.6.4 Validation of Life Policies.

8.6.3 Deleting Life Policies

Note: Only an **Asset Admin** can delete a **Life Policy** Item and only when this Item is in the **Preliminary** State. If otherwise, an error is raised.

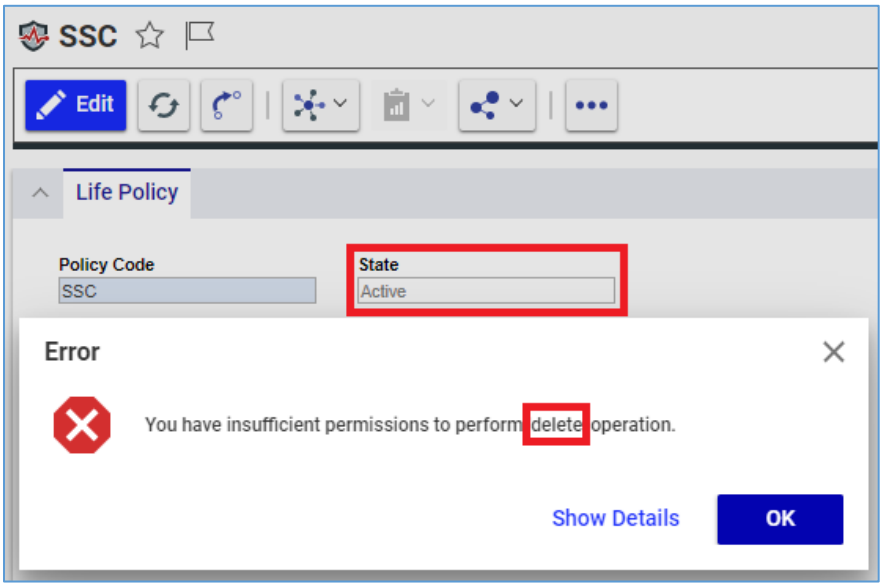


Figure 388.

To delete a **Life Parameter** Item, use the standard deletion procedure.

8.6.4 Validation of Life Policies

Once you click **Save** or **Done** on a **Life Policy** Item toolbar, the system validates this Item and raises an appropriate error if the Item:

- Is missing a **Policy Code** value.

Figure 389.

- Is missing a **Name** value.

Figure 390.

- Is missing an **Allowed Control Type** value.

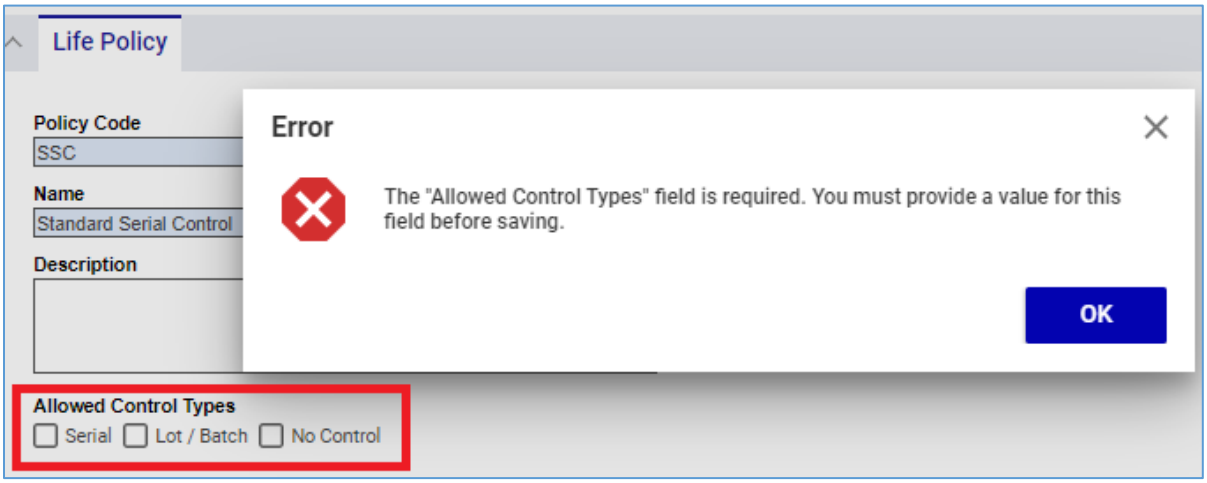


Figure 391.

- Has a **Policy Code** value that is not unique: already given to an existing **Life Policy**.

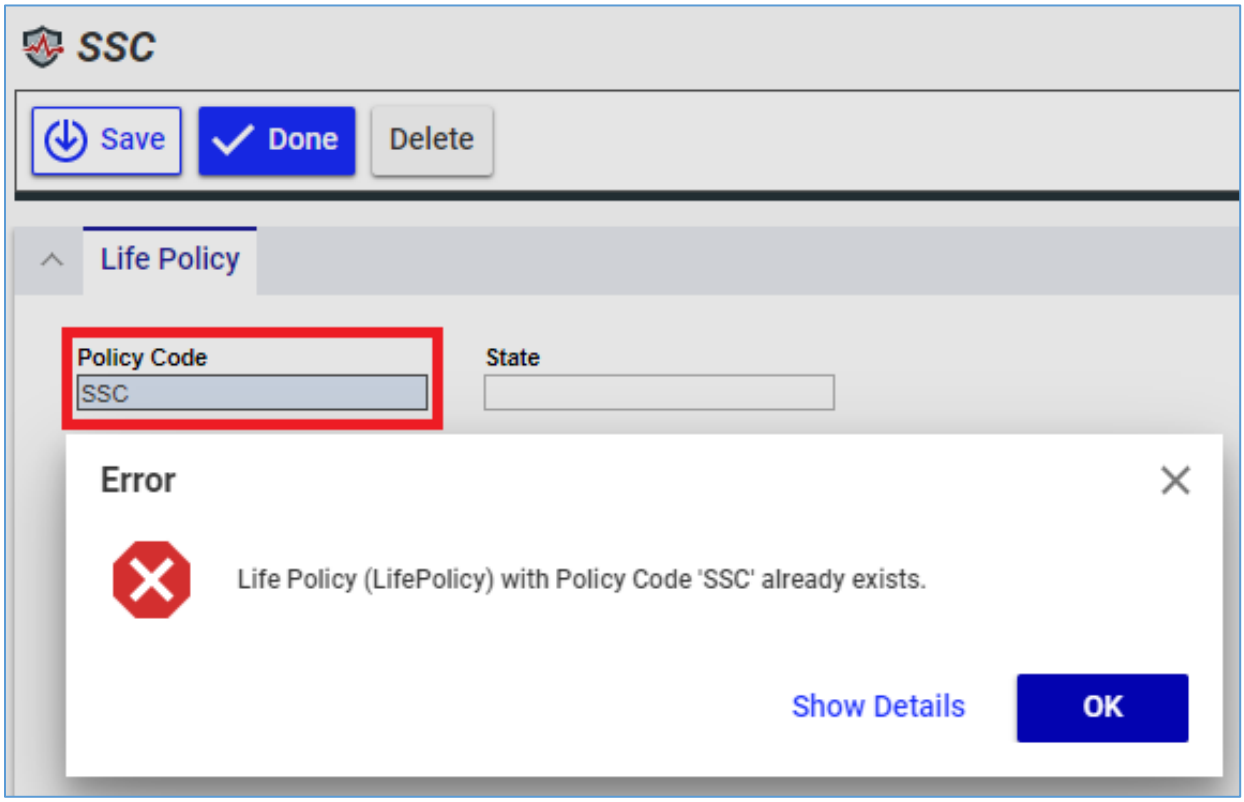


Figure 392.

- Has a **Life Parameter** Item in the **Preliminary** State.

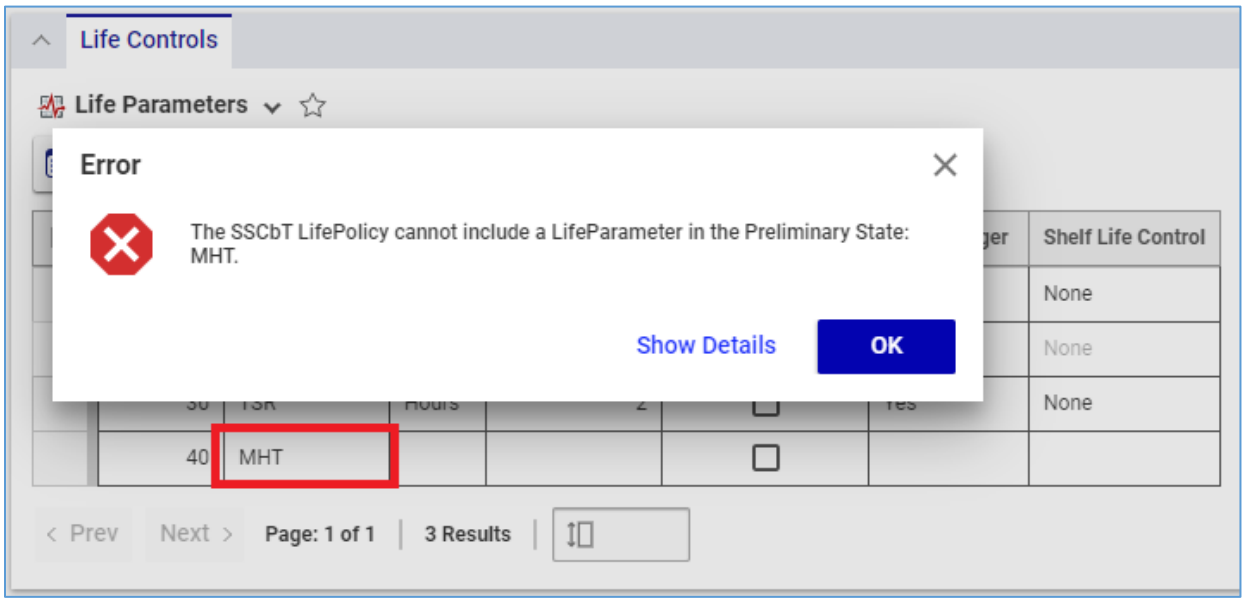


Figure 393.

9 Part Policies

Since the DTC R2 application, a **Part Policy** Item is a prerequisite for a **Physical Part** Item. It must exist for a **Part** Item from which the **Physical Part** Item will be created. For details on the **Physical Part** Items, refer to section 4 Physical Parts.

If it is necessary to track the lives of some **Physical Part** Items, needed **Life Unit**, **Life Parameter**, and **Life Policy** Items should exist before creating the **Part Policy** Items.

9.1 Understanding Part Policies

A set (collection) of rules, constraints, policies, and other properties may manage the lives (behavior) of assets in the real world. Assets of the same part number or some other characteristic must be governed similarly.

A **Part Policy** Item defines a policy set that governs **Physical Part** Items rooted in a **Part** Item specified in this **Part Policy** Item.

Note: The DTC application is shipped with no **Part Policy** Items out of the box. This section uses some **Part Policy** Items as illustrations.

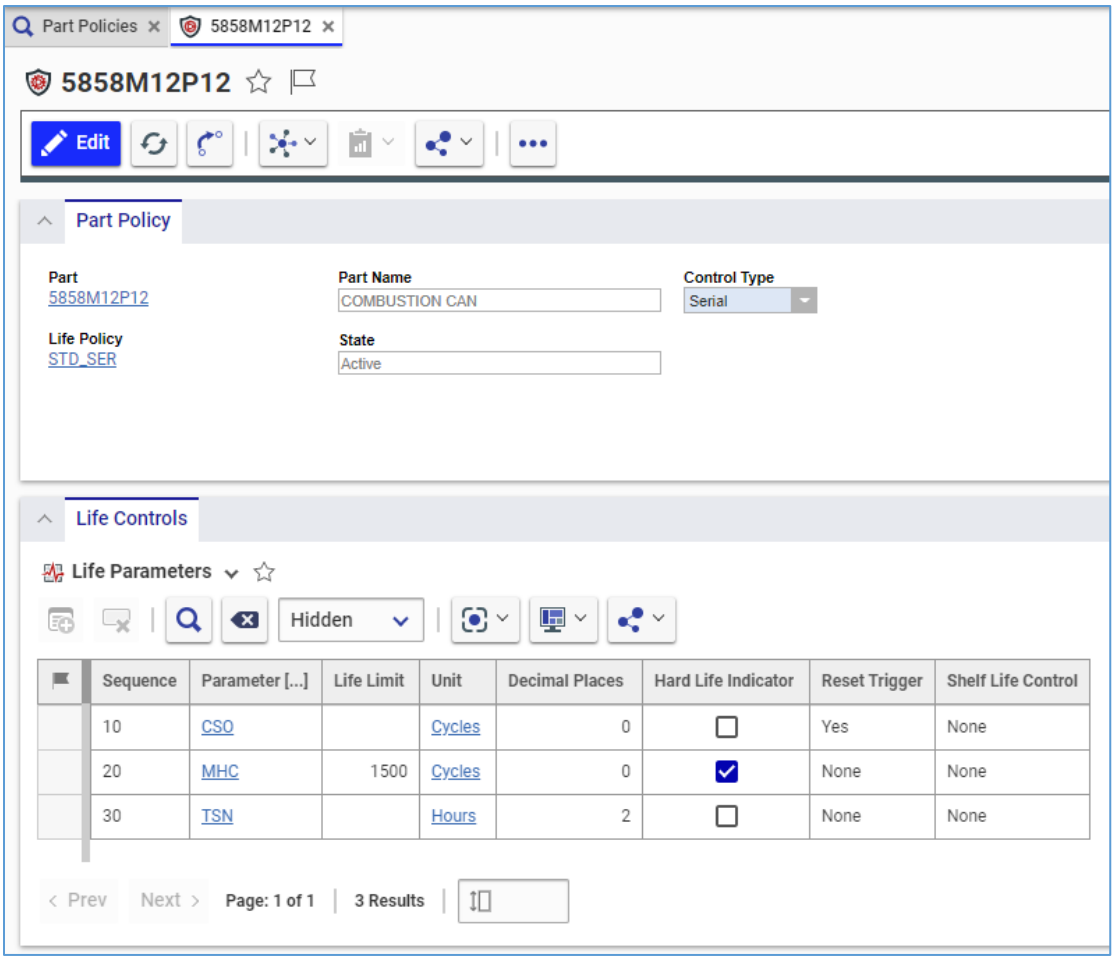


Figure 394.

Since the DTC R2 application, a **Part Policy** Item defines the inventory control type for the governed **Physical Part** Items: **Serial**, **Lot/Batch**, or **No Control**.

The **Part Policy** Item can connect a **Part** Item with a **Life Policy** Item so that the **Physical Part** Items that derive from this **Part** Item can be tracked by the **Life Parameter** Items that this **Life Policy** Item includes. For details, see section 9.4 Part Policy Relationships.

If a **Physical Part** Item does not require tracking its life values, its **Part Policy** Item can have no **Life Policy** Item, thus **Life Parameter** Items.

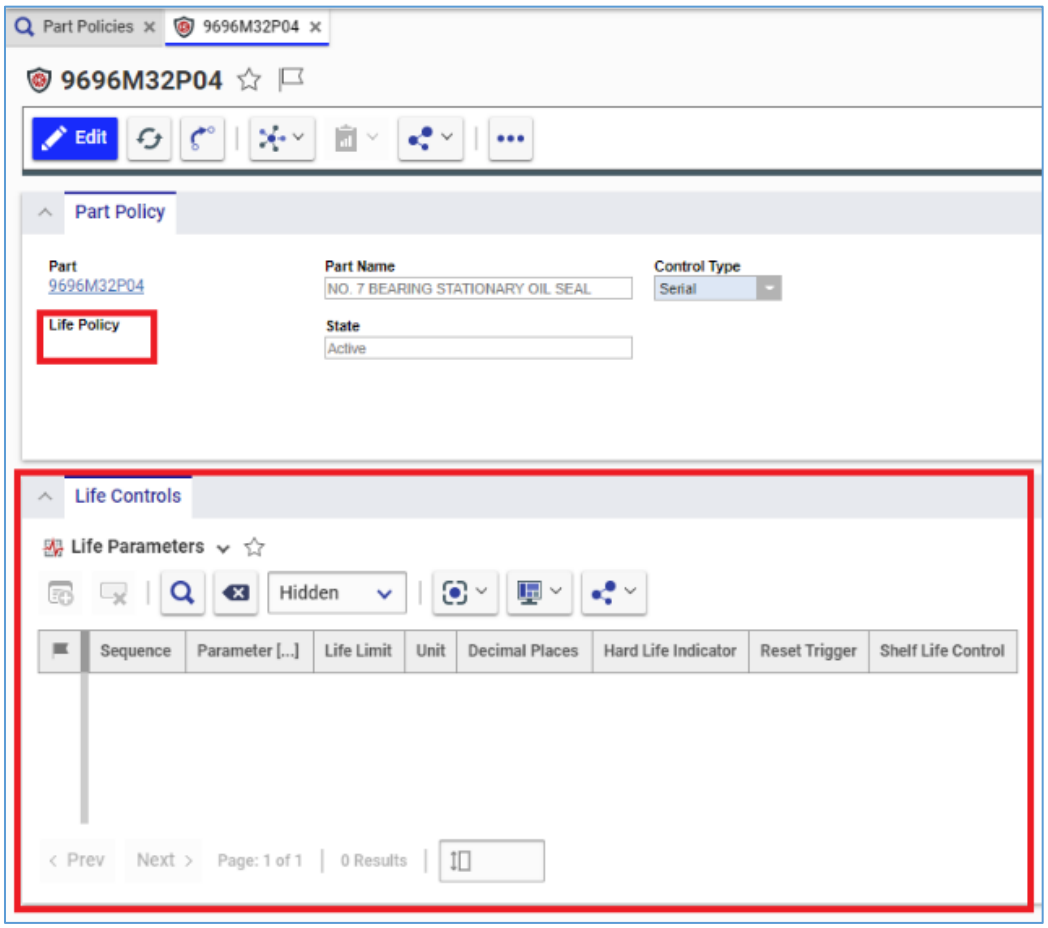


Figure 395.

For details about the **Part Policy** Item properties, see section 9.3 Part Policy Properties.

It is not possible to modify or delete a **Part Policy** Item if there are existing **Physical Part** Items originated from a **Part** Item related to this **Part Policy** Item. For details on using the **Part Policy** Items, refer to section 9.4 Part Policy Relationships.

9.2 Part Policies Search Grid

The **Part Policies** Search Grid has the standard features and the following columns:

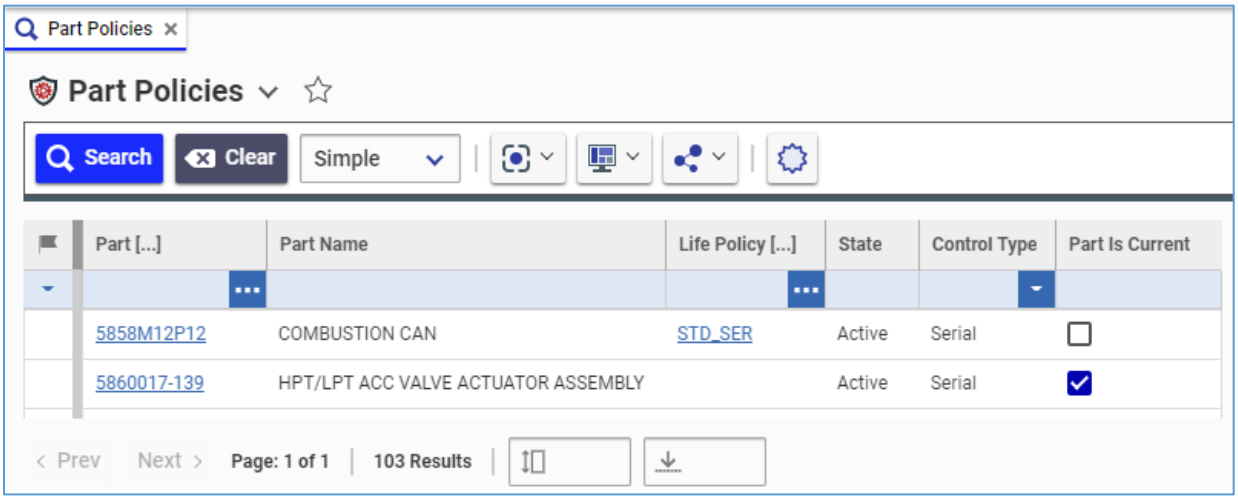


Figure 396.

- **Part**
- **Part Name**
- **Life Policy**
- **State**
- **Control Type**
- **Part Is Current**

Each column shows a matching **Part Policy** Item property. For the property details, see section 9.3 Part Policy Properties.

If **selected** (**true**), the **Part Is Current** property informs that the Generation of a **Part** Item specified in the given **Part Policy** Item is current (latest). When a newer Generation is created, the **Part Is Current** property will become automatically **cleared** (**false**).

The **Part** column provides a link to a related **Part** Item, not to the **Part Policy** Item.

Note: In the **Part** column, the **Part Policy** Items are sorted by globally unique identifiers (GUID) of the **Part** Items, not by the **Part Numbers**.

The **Life Policy** column provides a link to a related **Life Policy** Item, not to the **Part Policy** Item.

9.3 Part Policy Properties

The following properties of a **Part Policy** Item are available at the **Form** accordion of this **Life Policy** Item view:

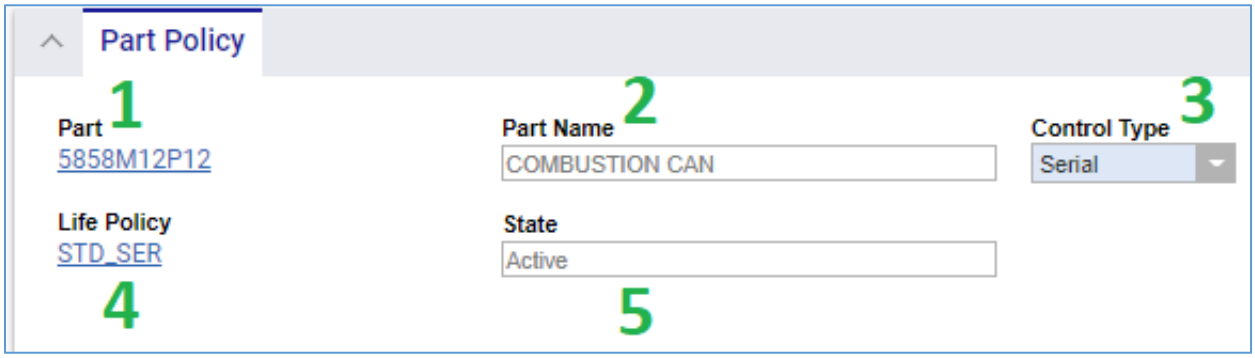


Figure 397.

1. **Part**: a governed **Part** Item showed by its **Part Number**. It is also a link to this **Part** Item. This property is required and unique for a **Part Policy** Item. See section 9.4.1 Part Item Related to Part Policy Item.
2. **Part Name**: a view-only **Name** of a given **Part** Item (1).
3. **Control Type**: a Control Type for a given **Part** Item (1) and **Physical Part** Items derived from it. See section 9.4.1 Part Item Related to Part Policy Item. This property is required for a **Part Policy** Item. It can be any of the following values:
   - **Serial**
   - **Lot / Batch**
   - **No Control**

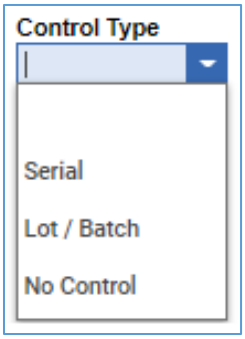


Figure 398.

4. **Life Policy**: an applied **Life Policy** Item showed by its **Policy Code**. It is also a link to this **Life Policy** Item. See section 9.4.2 Life Policy Item related to Part Policy Item.
5. **State**: the current State of the **Part Policy** Item. See section 9.5 Part Policy Life Cycle and promotion.

9.4 Part Policy Relationships

In the case of using **Physical Part** Items without tracking their life variables, a **Part Policy** Item is just a prerequisite for a **Physical Part** Item—it must exist for a **Part** Item from which the **Physical Part** Item will be created. In this case, the **Part Policy** Item should not have a **Life Policy** Item because tracking **Life Parameter** Items are unnecessary.

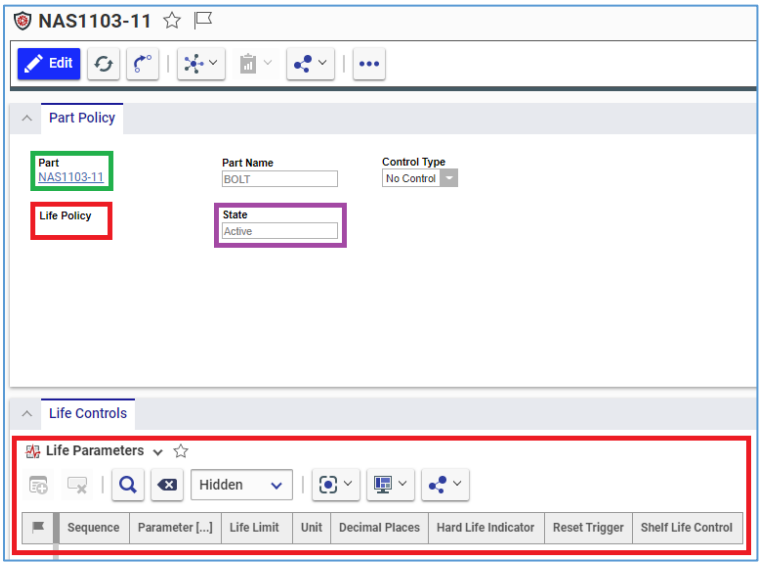


Figure 399.

In the case of tracking life variables of **Physical Part** Items, a **Part Policy** Item establishes a connection between a **Life Policy** Item and a **Part** Item. Through this connection, all **Life Parameter** Items that the given **Life Policy** Item includes are applied to all **Physical Part** Items that originate from all Generations (Versions) of the given **Part** Item.

In this decoupling architecture, a set of one or more **Life Parameter** Items is one **Life Policy** Item.

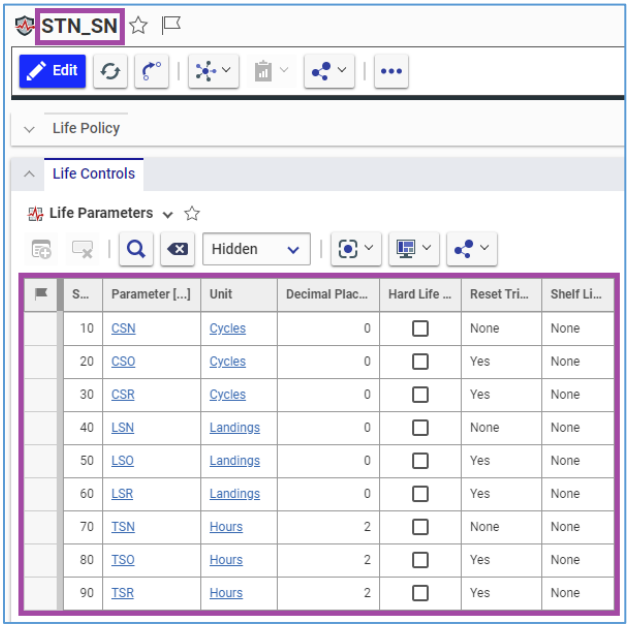


Figure 400.

One **Life Policy** Item, with the same set of **Life Parameter** Items, is applied to one or more **Part Policy** Items, thus **Part** Items with different **Part Numbers**.

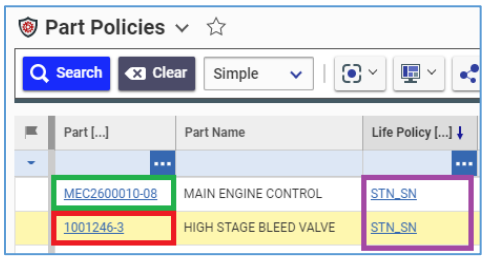


Figure 401.

One **Part Policy** Item, with the same set of **Life Parameter** Items, governs one **Part** Item, thus all **Physical Part** Items with the same **Part Number** and different **Serial** or **Lot/Batch Numbers**.

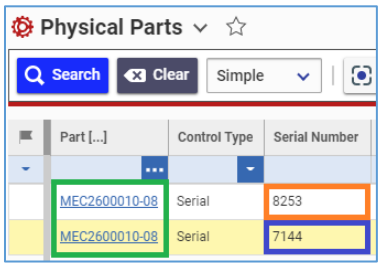


Figure 402.

9.4.1 Part Item Related to Part Policy Item

The **Part** property of a **Part Policy** Item defines a **Part** Item governed by the given **Part Policy** Item. This property can be initially set and then reset to another **Part** Item only when this **Part Policy** Item is in the **Preliminary** State.

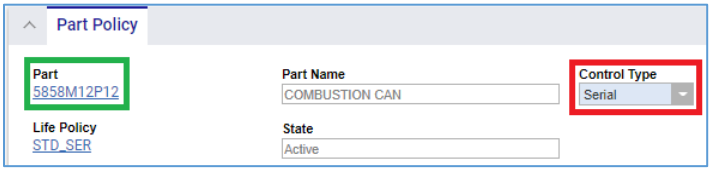


Figure 403.

There is a one-to-one relationship between **Part** and **Part Policy** Items: one **Part** Item can be governed only by one **Part Policy** Item, and one **Part Policy** Item can govern only one **Part** Item. An error is raised, and a **Part Policy** Item cannot be saved if a **Part** Item specified in this **Part Policy** Item already has a governing **Part Policy** Item.

Figure 404.

A **Part** Item should have its **Control Type** property set before being specified in a **Part Policy** Item.

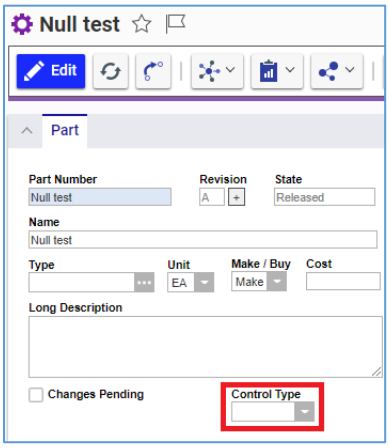


Figure 405.

As soon as the **Part** Item is specified or respecified in the **Part** field of the **Part Policy** Item, the **Control Type** property of this **Part Policy** Item is automatically set to a **Control Type** value specified in the current (latest) Generation of this **Part** Item.

If the current **Part** Item Generation has no **Control Type** specified, the **Part Policy** Item **Control Type** field remains blank.

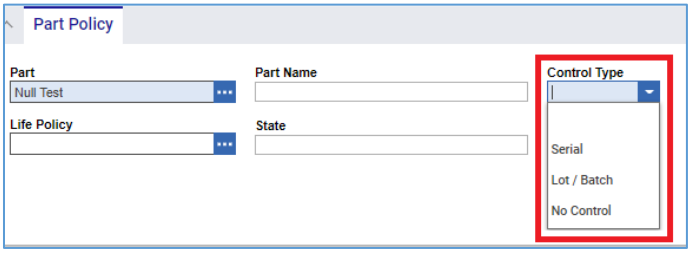


Figure 406.

Regardless of the **Part** Item **Control Type** value, the **Part Policy** Item **Control Type** property can be set or reset to any value on its list when this **Part Policy** Item is in the **Preliminary** State.

Upon saving a **Part Policy** Item, an error is raised, and this **Part Policy** Item cannot be saved if its **Control Type** property is not set.

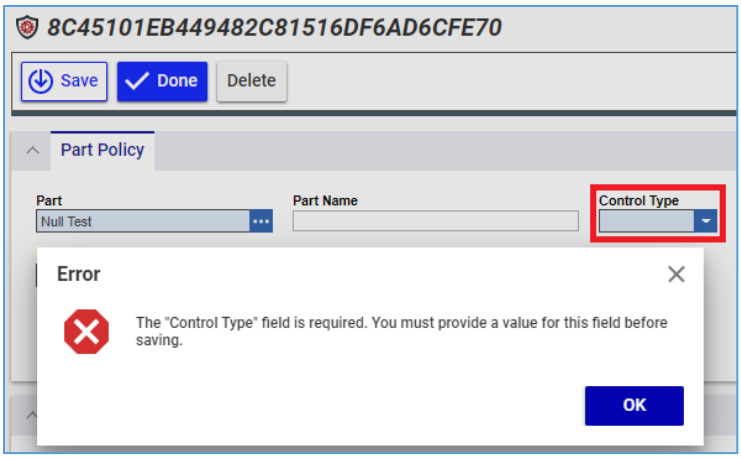


Figure 407.

A **Control Type** value of the **Part Policy** Item defines the Control Type of **Physical Part** Items that originate from the **Part** Item related to this **Part Policy** Item. A **Control Type** value of this **Part** Item does not affect the Control Type of these **Physical Part** Items.

Note: In DTC 12.0R1, the **Part Policy** Items do not exist, and the control type definition is on the **Part** Item. To start using **Physical Part** Items imported from 12.0R1, see section 14 Migrating from DTC 12.0R1 to 12.0R2.

If the **Part** Item gets a newer Generation, including a new **Control Type** value, when already related to the **Part Policy** Item, this **Part** Item Generation does not affect the given **Part Policy** Item. The **Part Policy** Item **Control Type** property is not reset with the new **Part** Item Generation **Control Type** value.

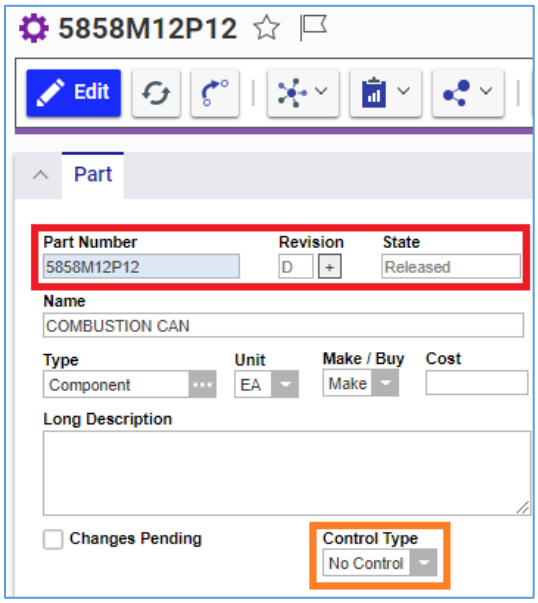


Figure 408.

The **Part Policy** Item remains related to the **Part** Item Generation that was current during this **Part Policy** Item creation. The **Part** property of the **Part Policy** Item is a link to this Generation.

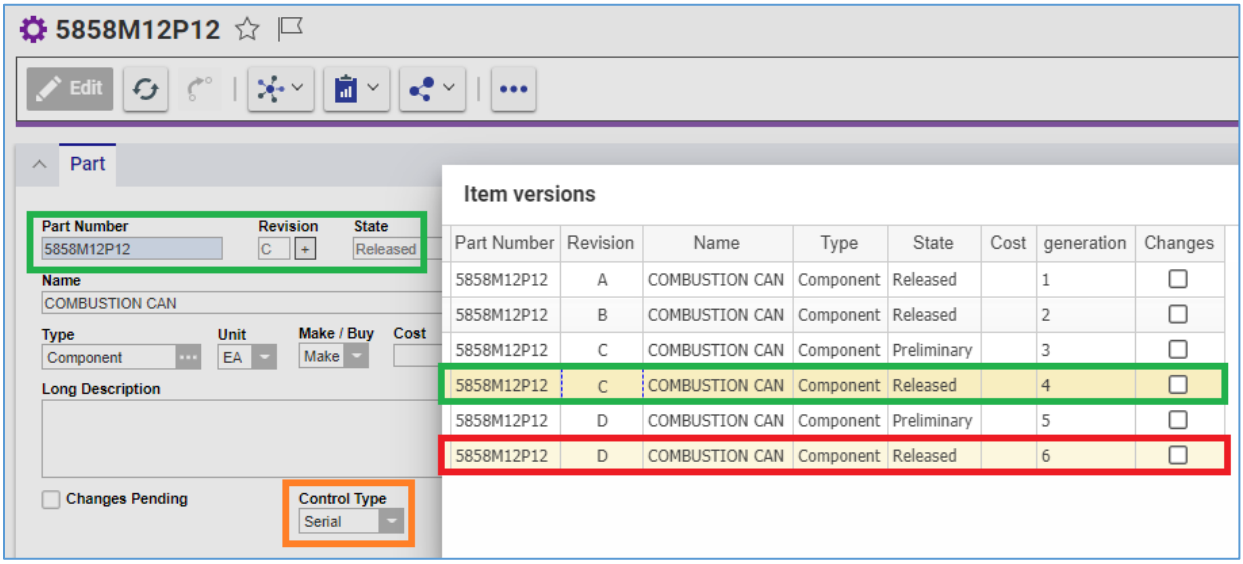


Figure 409.

In this case, the **Part Is Current** property of the **Part Policy** Item is **cleared** (**false**).

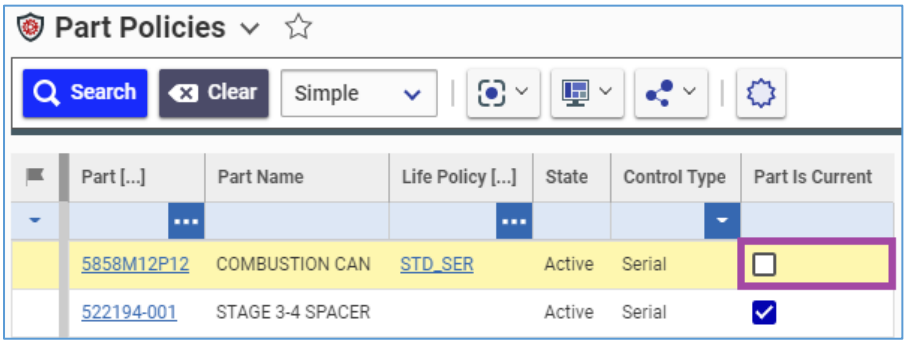


Figure 410.

The **Part** Item can be in any State when specified in a **Part Policy** Item.

The **Part Policy** Item must be in the **Active** State to allow creation of **Physical Part** Items from the **Part** Item related to this **Part Policy** Item.

9.4.2 Life Policy Item Applied to Part Policy Item

The **Life Policy** property of a **Part Policy** Item defines a **Life Policy** Item applied to the given **Part Policy** Item. This property can be initially set and then reset to another **Life Policy** Item only when this **Part Policy** Item is in the **Preliminary** State.

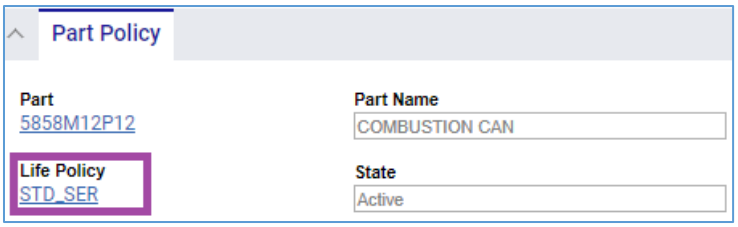


Figure 411.

The **Control Type** property value of the **Part Policy** Item defines which **Life Policy** Item is allowed for this **Part Policy** Item.

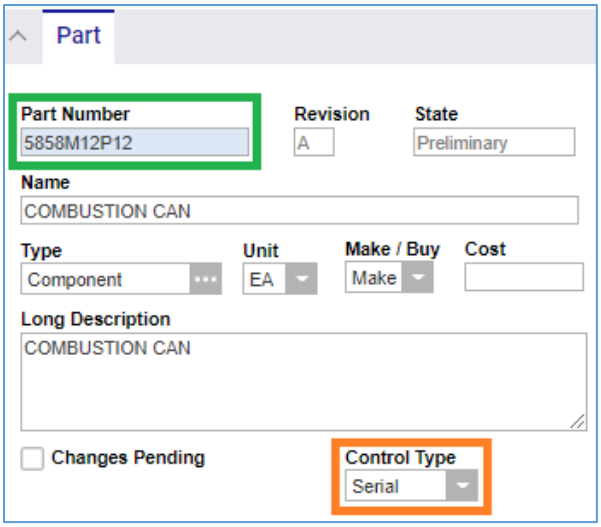


Figure 412.

To be allowed, a **Life Policy** Item should include the **Control Type** value of the discussed **Part Policy** Item in its **Allowed Control Types**.

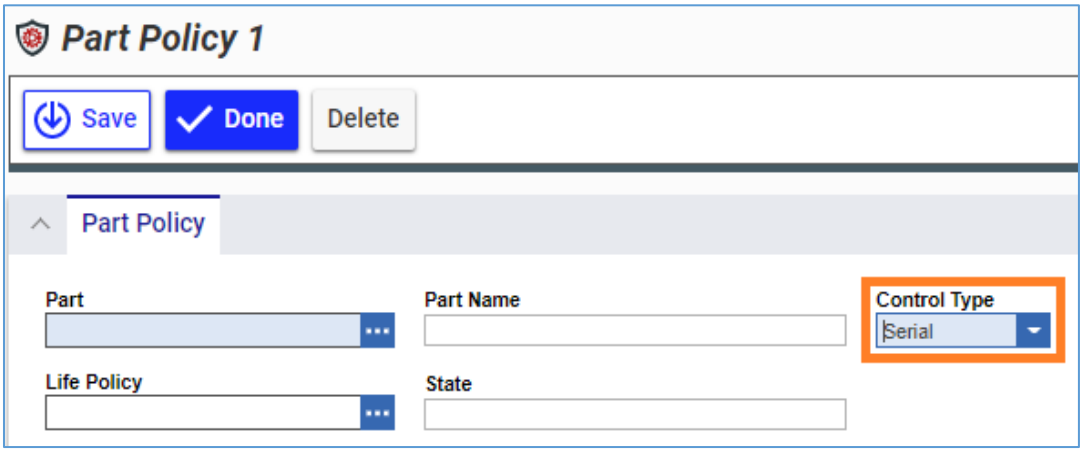


Figure 413.

Otherwise, upon saving a **Part Policy** Item, an error is raised, and this **Part Policy** Item cannot be saved.

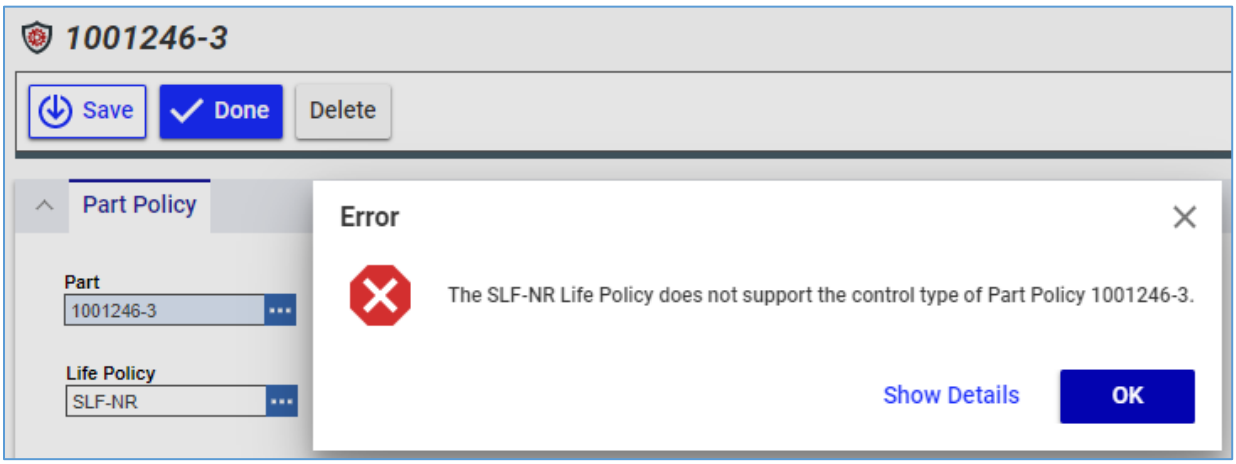


Figure 414.

The **Life Policy** Item must be in the **Active** State to be allowed for the **Part Policy** Item. Otherwise, upon saving this **Part Policy** Item, an error is raised, and this **Part Policy** Item cannot be saved.

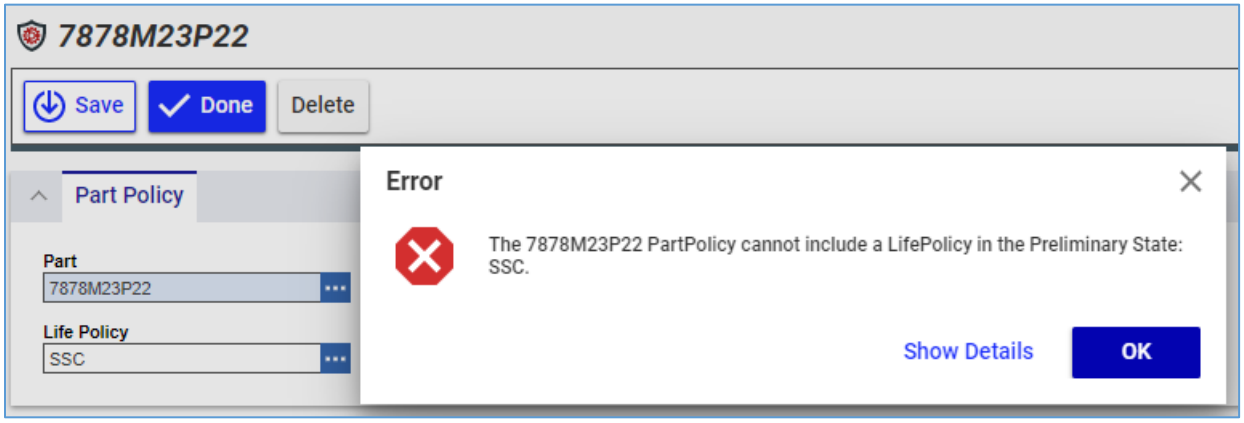


Figure 415.

9.4.3 Life Parameter Items Related to Part Policy Items

A **Part Policy** Item has the **Life Controls** Relationships tab for the **PartPolicy LifeParameter** Relationship Items defining **Life Parameter** Items that will track matching life variables of **Physical Part** Items originated from the **Part** Item governed by this **Part Policy** Item.

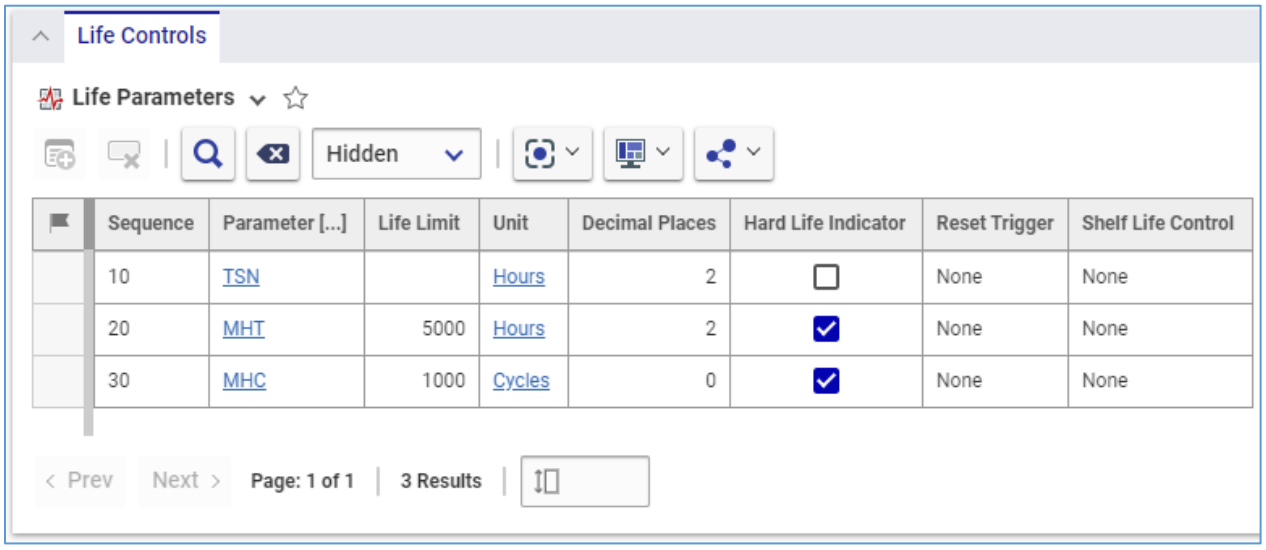


Figure 416.

This tab has the **Life Controls Life Parameters** Relationships Grid that shows a given **PartPolicy LifeParameter** Relationship Item as follows:

- **Sequence**: a view-only sort order number of the given **PartPolicy LifeParameter** Relationship Item in the Grid. It is the same as in the sourcing **Life Controls Life Parameter** Relationships Grid of an applied **Life Policy** Item.
- **Parameter**: a view-only **Parameter Code** property value of a related **Life Parameter** Item. It is also a link to this **Life Parameter** Item.
- **Life Limit**: a numeric value of the maximum failure (hard) limit for a corresponding life-limiting variable. This property should be set only for a **Life Parameter** Item that limits the **Physical Part** Item lives: its **Hard Life Indicator** value is **true**.
- **Unit**: a view-only **Life Unit** Item given in the **Unit of Measure** property of the related **Life Parameter** Item. It is also a link to this **Life Unit** Item.
- **Decimal Places**: a view-only **Decimal Places** property value of a given **Life Unit** Item.
- **Hard Life Indicator**: a view-only **Hard Life Indicator** property value of the related **Life Parameter** Item.
- **Reset Trigger**: a view-only **Reset Trigger** property value of the related **Life Parameter** Item.
- **Shelf Life Control**: a view-only **Shelf Life Control** property value of the related **Life Parameter** Item.

In the Grid, the **PartPolicy LifeParameter** Relationship Items are sorted by their **Sequence** values in the ascending order (from 0 to 9).

Except for the **Life Limit** cells, the **Life Parameters** Relationships Grid is not editable.

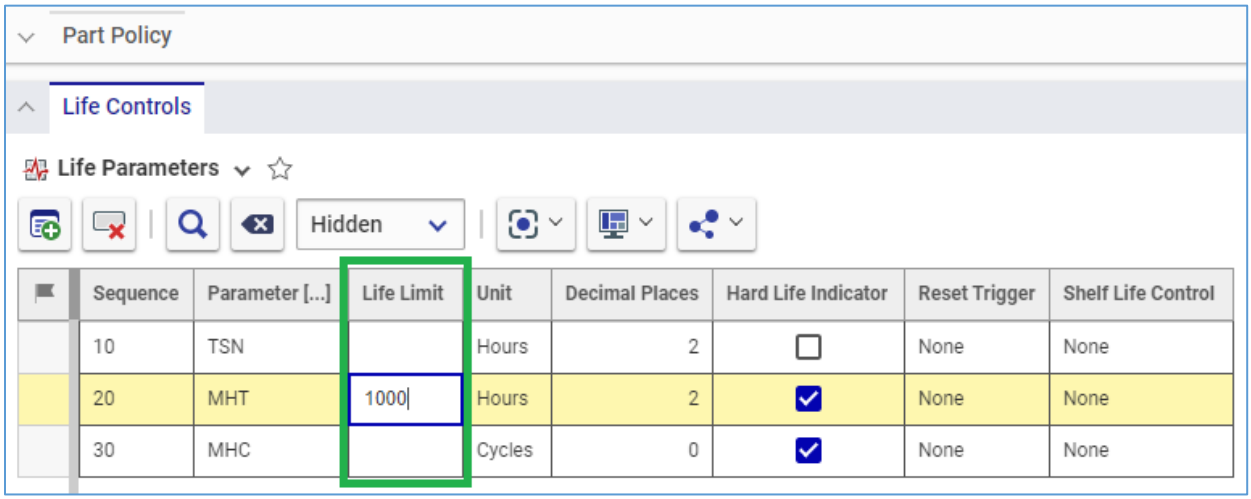


Figure 417.

The **Life Limit** cells should be populated only for the related **Life Parameter** Items with the **Hard Life Indicator** set to **true**. Upon saving a **Part Policy** Item, an error is raised, and this **Part Policy** Item cannot be saved if there is a **Life Limit** cell populated for a related **Life Parameter** Item with the **Hard Life Indicator** set to **false**.

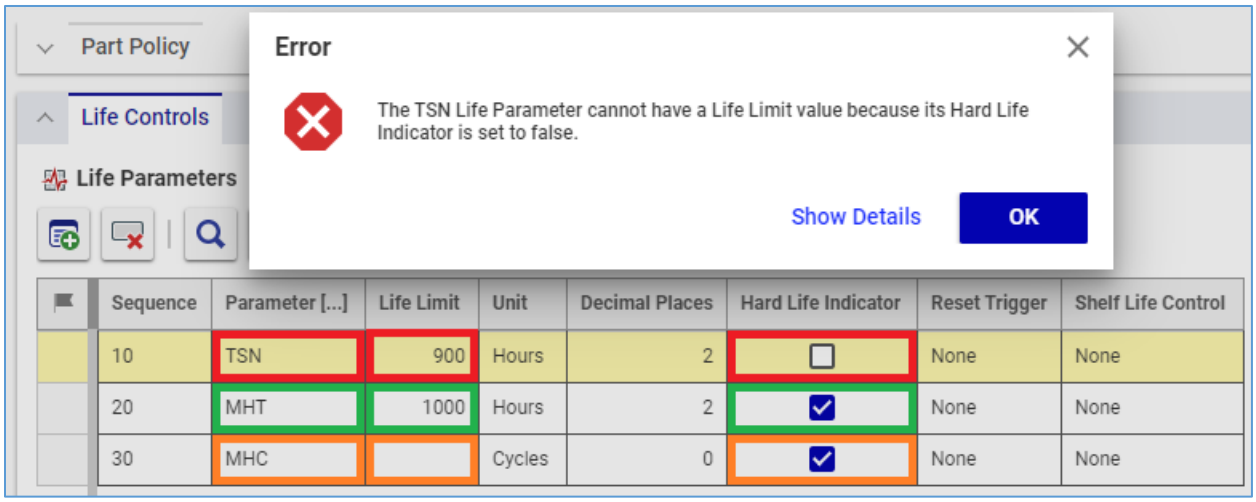


Figure 418.

When the **Part Policy** Item is in the **Preliminary** State, the **Life Limit** cells can be empty where required. To promote the **Part Policy** Item to the **Active** State, they must be populated. Otherwise, an error is raised during the promotion.

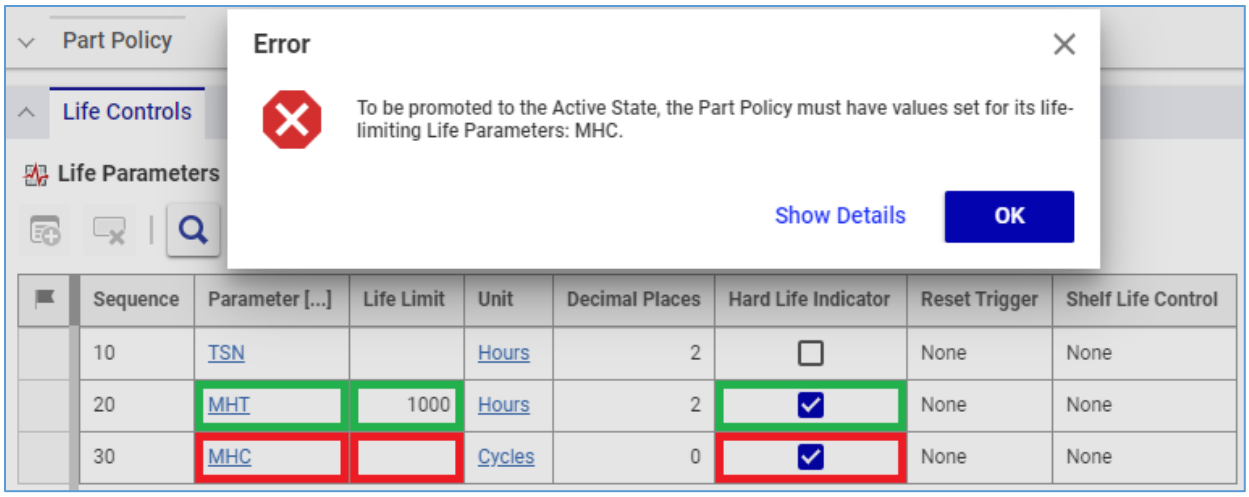


Figure 419.

When editing a **Part Policy** Item, the **Life Controls Life Parameters** Relationships Grid has its toolbar and action menu enabled. But still, the manual management of this Grid is forbidden—an error is raised upon saving the **Part Policy** Item after adding, editing, or removing a **PartPolicy LifeParameter** Relationship Item with any toolbar or action menu button. For details, see section 3.7 Automatically managed DTC Relationships Grids.

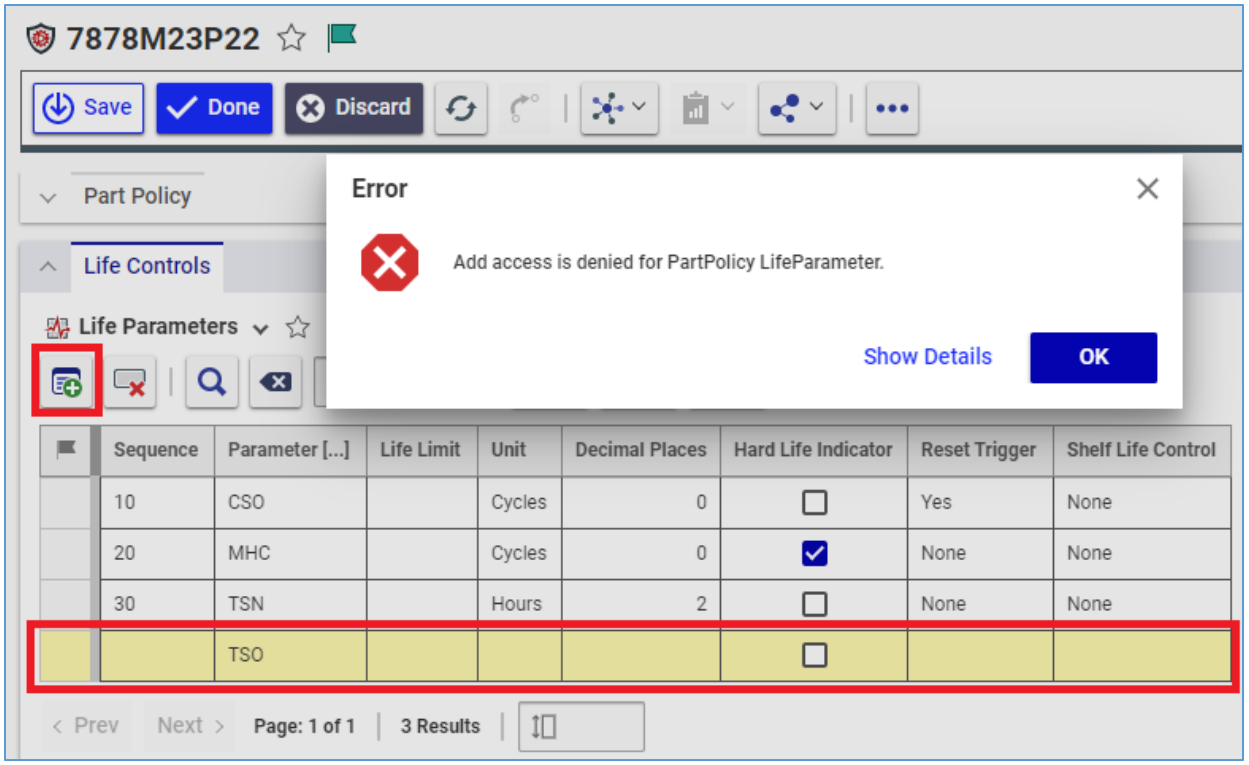


Figure 420.

A **Life Policy** Item specified in a **Part Policy** Item automatically sources the **Life Controls Life Parameters** Relationships Grid with its **Life Parameter** Items.

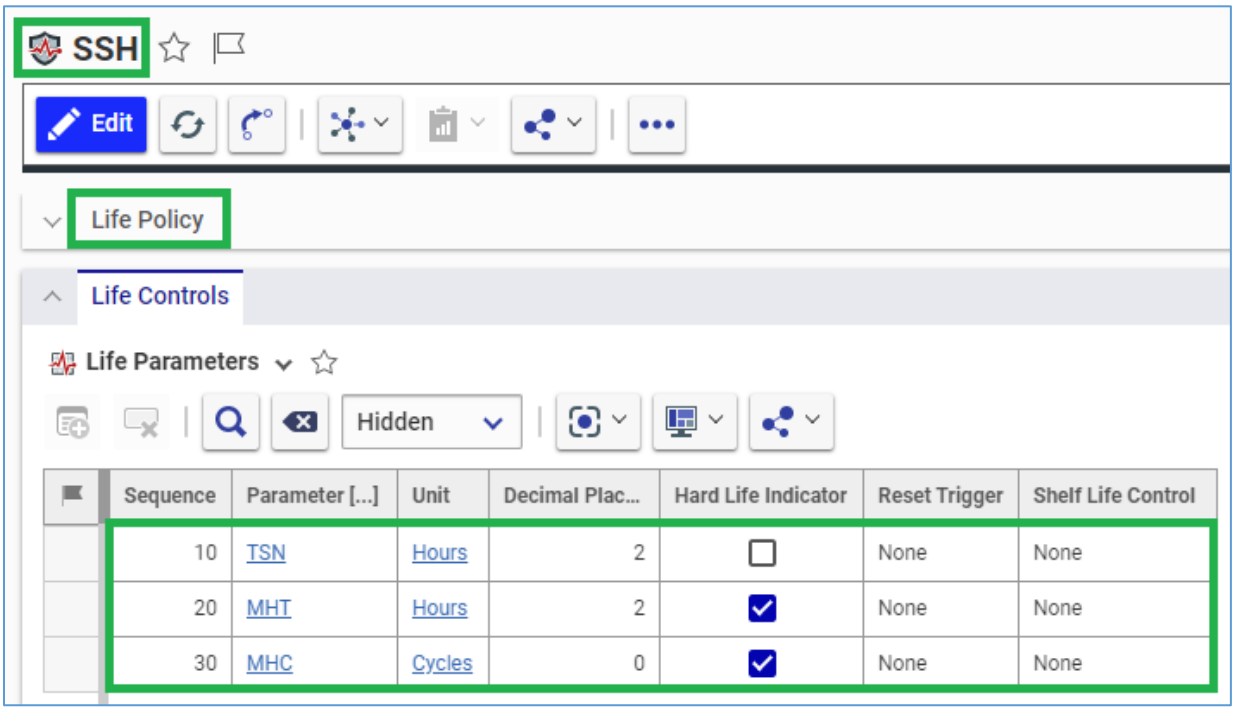


Figure 421.

Initially, the **Part Policy** Item Grid is empty, even after a **Life Policy** Item had been specified.

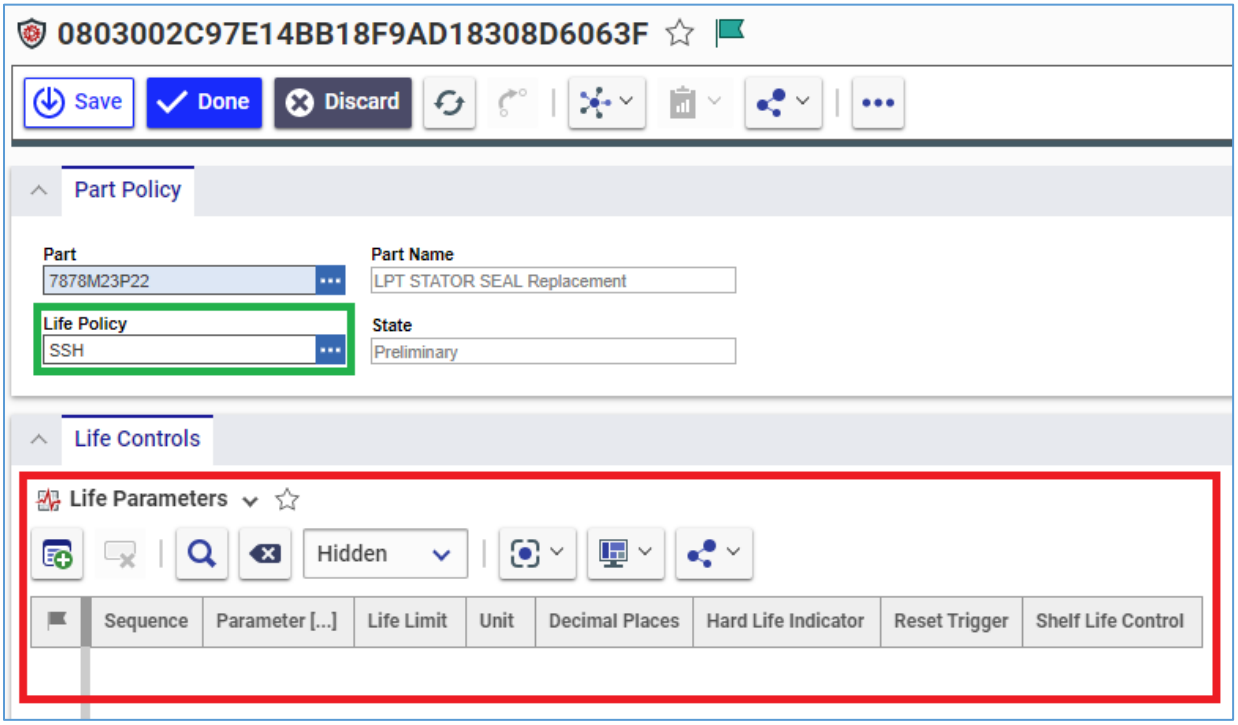


Figure 422.

Once the **Part Policy** Item is saved (2) after the **Life Policy** Item had been specified for the first time (1), the empty Grid is automatically populated with the **PartPolicy LifeParameter** Relationship Items, which related Items are **Life Parameter** Items given in this **Life Policy** Item (3). **Life Limit** values are absent.

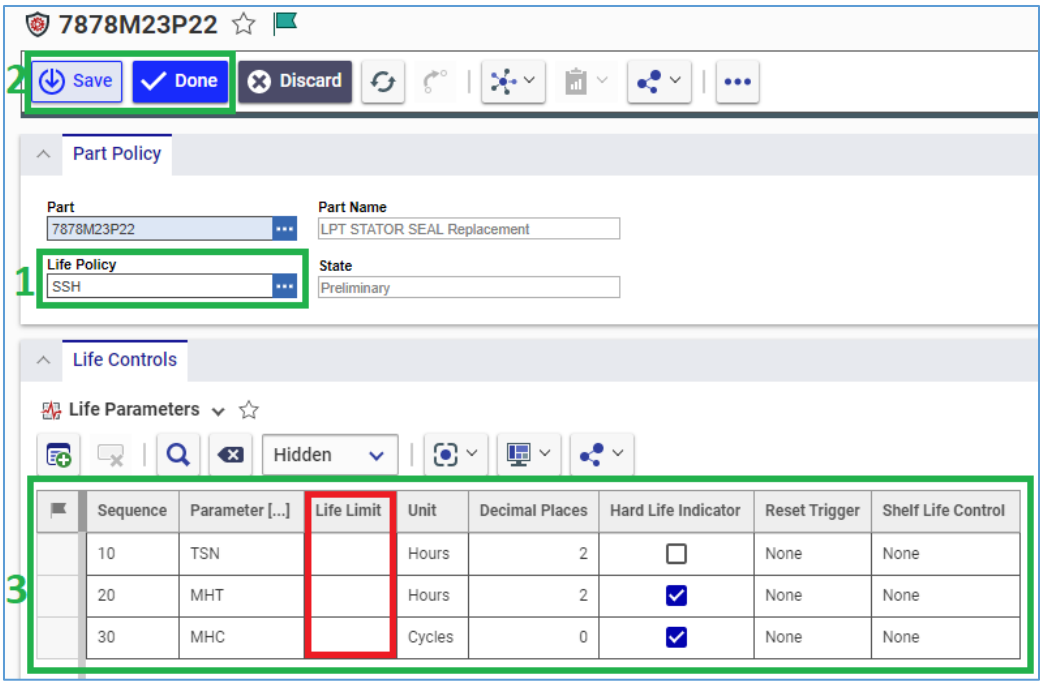


Figure 423.

When the **Part Policy** Item is saved after another **Life Policy** Item had been specified, the Grid is automatically repopulated with **Life Parameter** Items sourced from the new **Life Policy** Item. The **Life Limit** values become absent if they have been populated for the old **Life Policy** Item.

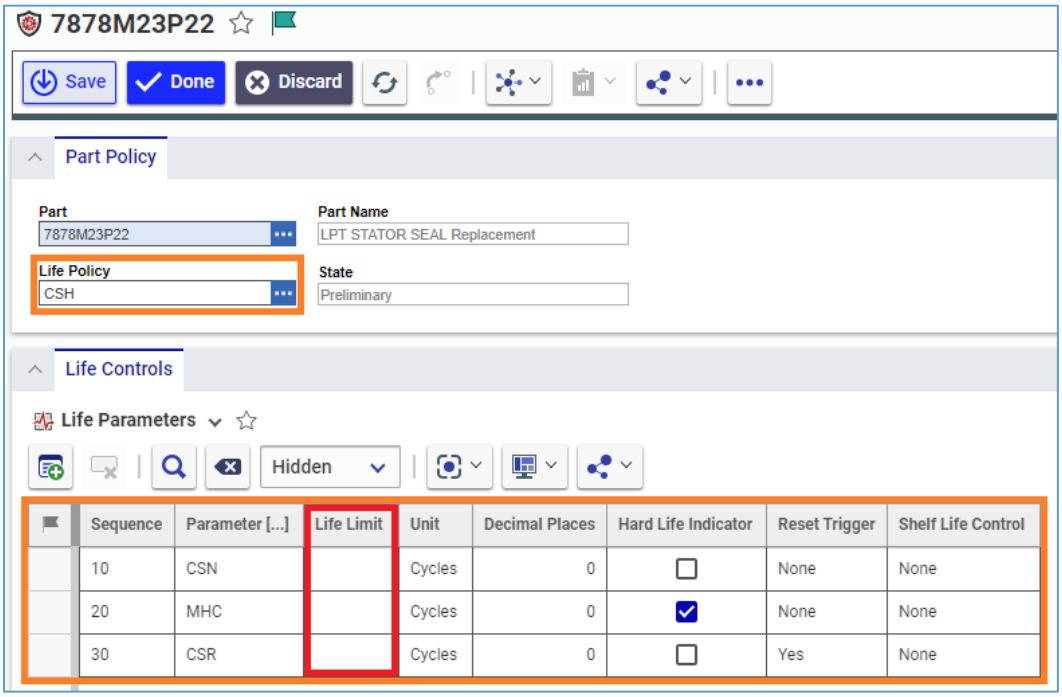


Figure 424.

The Grid has the **Life Parameter** Items exactly as they are in the new **Life Policy** Item.

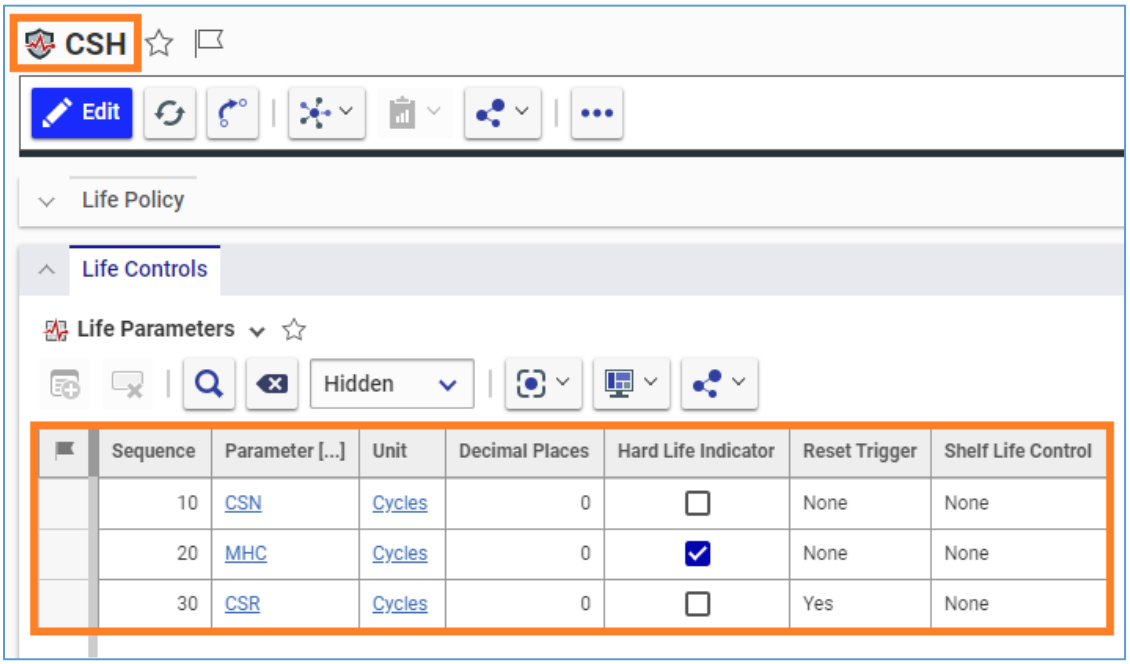


Figure 425.

When the **Part Policy** Item is saved after the **Life Policy** Item had been removed from it, the Grid becomes blank automatically.

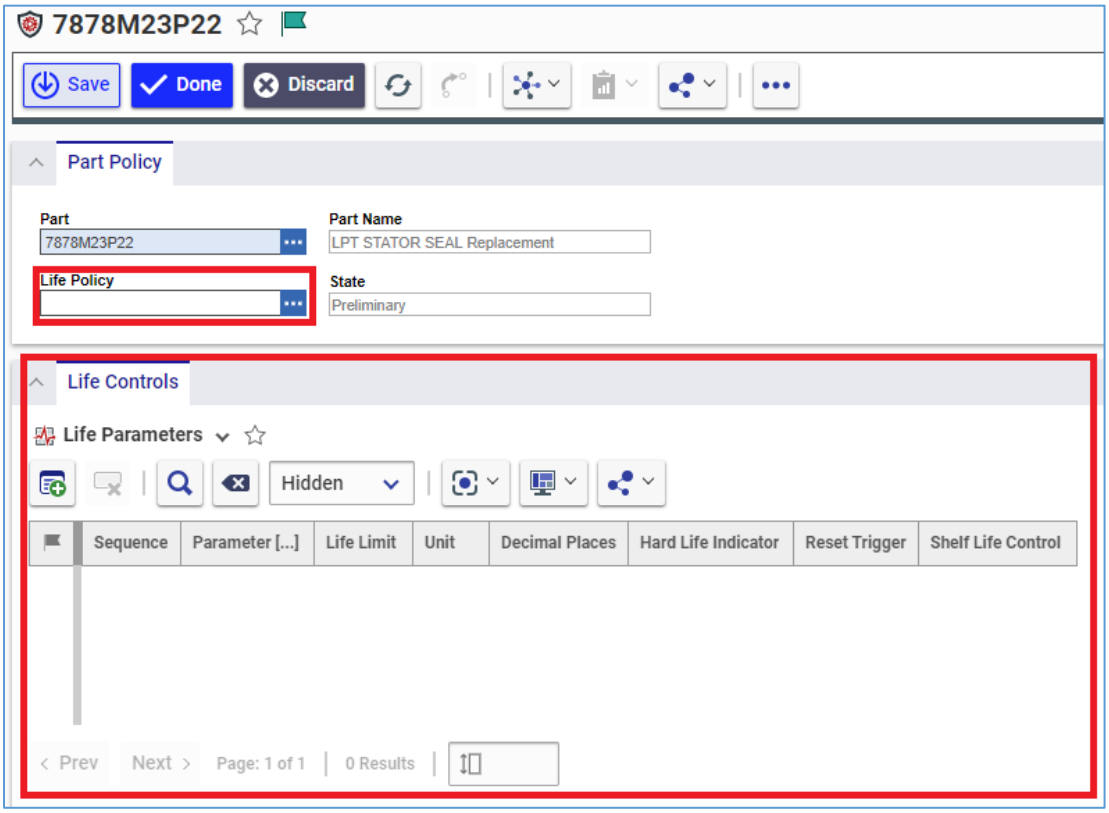


Figure 426.

9.5 Part Policy Life Cycle and Promotion

Note: Only an **Asset Editor** or **Asset Admin** can promote **Part Policy** Items.

The **PartPolicy** Life Cycle Map defines the following States for the **Part Policy** Items:

1. **Preliminary**
2. **Active**

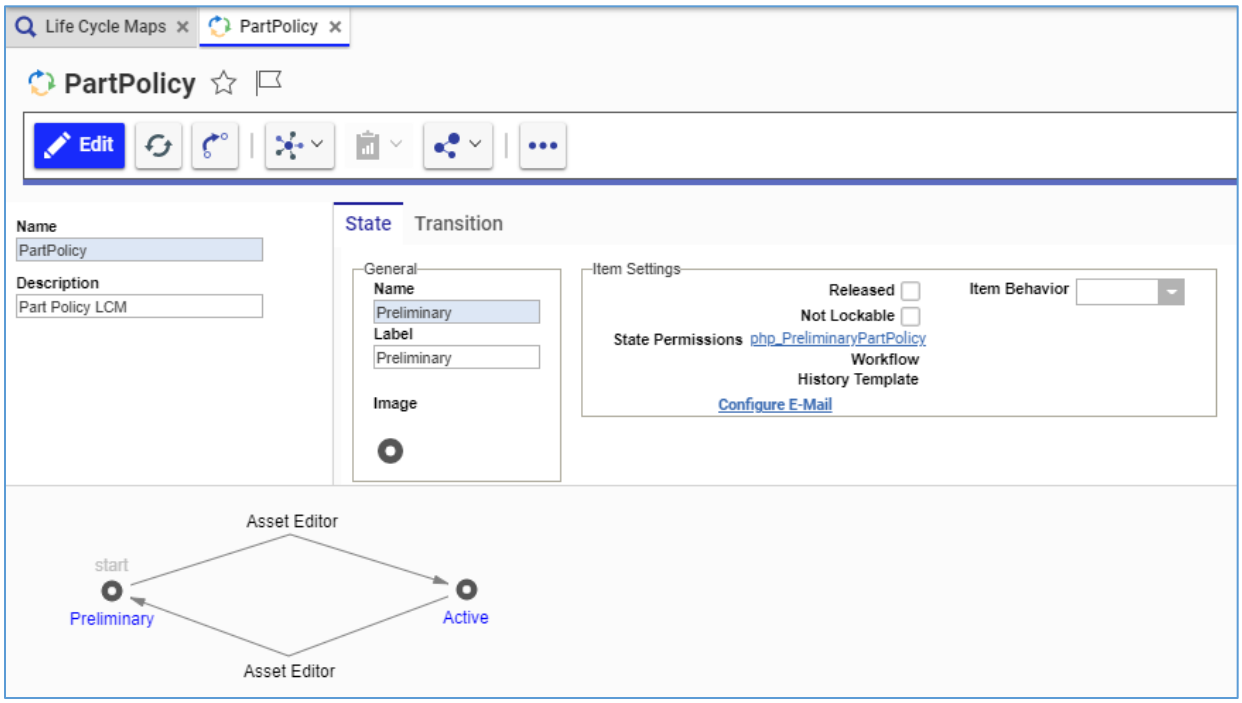


Figure 427.

Use the standard Item promotion procedure to promote the **Part Policy** Items.

Once created, a new **Part Policy** Item is in the **Preliminary** State, where it is a draft: a member of the **Asset Editor** or **Asset Admin** Identity has full editing rights for its properties: they can create, edit, and delete the **Part Policy** Item.

Being in the **Preliminary** State, the **Part Policy** Item cannot govern **Physical Part** Items. To become allowed, the **Part Policy** Item must be promoted to the **Active** State when it is ready for usage and not supposed to be changed.

The **Part Policy** Item cannot be promoted to the **Active** State if its **Life Controls Life Parameters** Relationships Grid has a **Life Limit** cell missing a value where it is required. For details, see section 9.4.3 Life Parameter Items Related to Part Policy Item.

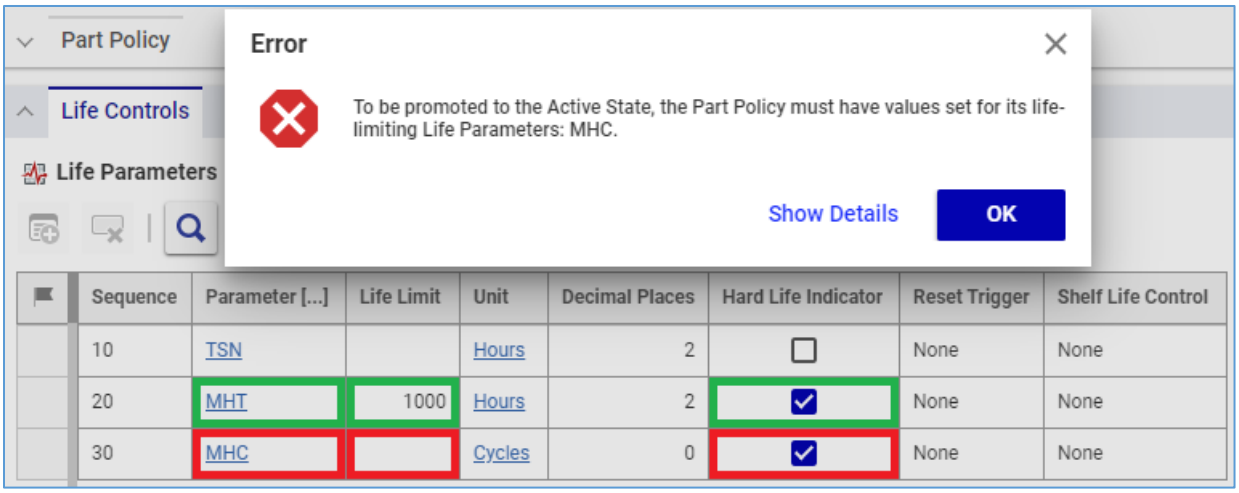


Figure 428.

Editing, populating, and deleting the **Part Policy** Item in the **Active** State is not allowed for any Identity.

The **Part Policy** Item can be promoted back to the **Preliminary** State only if there are no existing **Physical Part** Items originated from a **Part** Item governed by this **Part Policy** Item. Otherwise, an error is raised during such promoting. All **Physical Part** Items in question should be deleted first to allow this promotion. For details on the **Part Policy** Item relationships, refer to section 8.4 Life Policy Relationships and Usage.

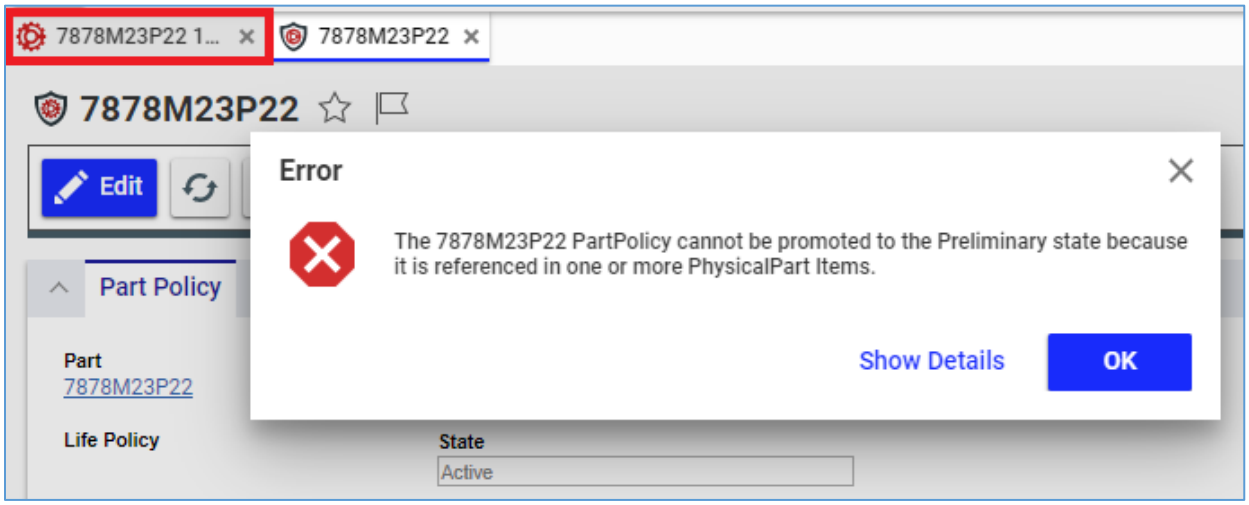


Figure 429.

9.6 Managing Part Policies

Your ability to manage a **Part Policy** Item depends on your Asset Identity, current Item State, and Item Relationships.

9.6.1 Creating Part Policies

Note: Only an **Asset Admin** or **Asset Editor** can create **Physical Part** Items.

To create a new **Part Policy** Item:

1. Go to **Contents** --> **Assets** --> **Part Policies**.
2. Click **Create Part Policy**. The **Part Policy** # Item view appears.

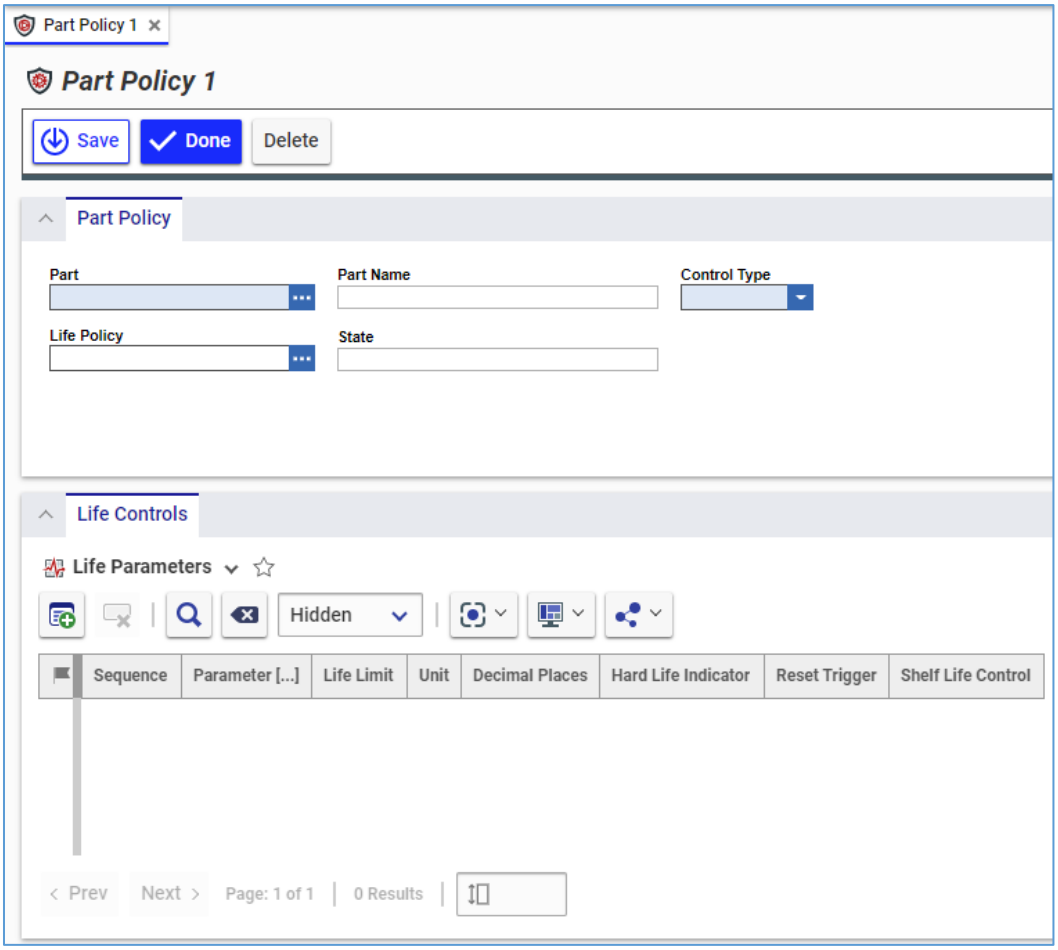


Figure 430.

3. Click the **ellipsis** button in the **Part** field. The **Select Items – Parts** dialog appears.

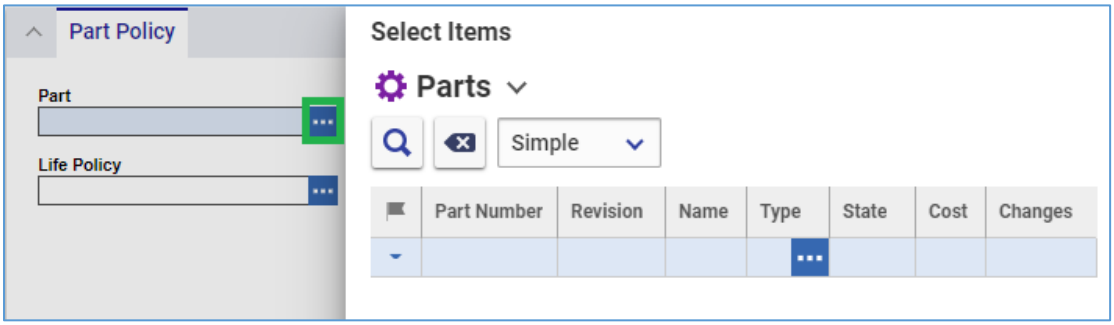


Figure 431.

4. Using the standard search procedure, search for the **Part** Item that the given **Part Policy** Item will govern.

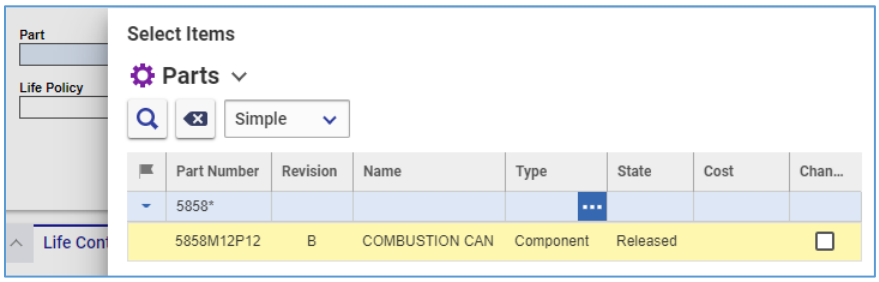


Figure 432.

5. Select this **Part** Item and click **OK**. The **Part** Item **Control Type** property value automatically populates the **Part Policy** Item **Control Type** property.

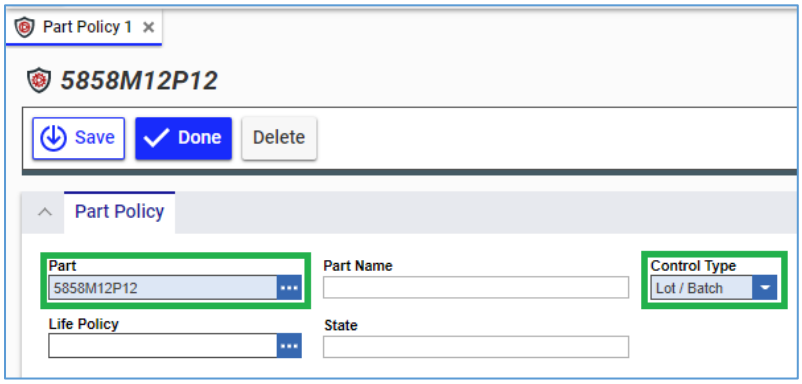


Figure 433.

Note: You can also specify the governed **Part** Item by typing in the **Part** field.

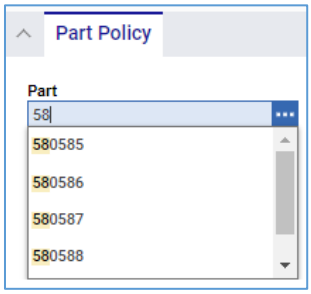


Figure 434.

6. If it is necessary, correct the **Control Type** property of the **Part Policy** Item.

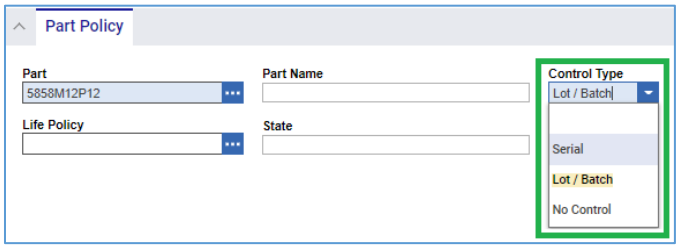


Figure 435.

7. In the case of not tracking life variables of **Physical Part** Items that will originate from the related **Part** Item, go to step 13. Otherwise, take the next step and continue this procedure to the end.
8. Click the **ellipsis** button in the **Life Policy** field. The **Select Items – Life Policies** dialog appears.

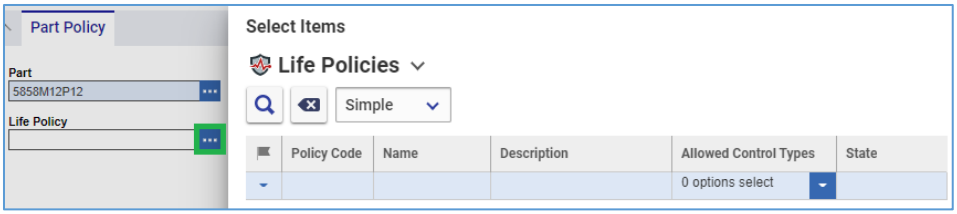


Figure 436.

9. Using the standard search procedure, search for the **Life Policy** Item that will apply to the given **Part Policy** Item.

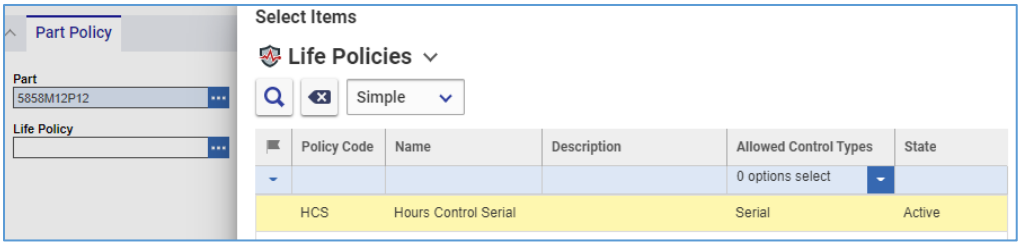


Figure 437.

10. Select this **Life Policy** Item and click **OK**. The selected **Life Policy** Item populates the **Life Policy** property. The **Life Controls Life Parameters** Relationships Grid remains empty.

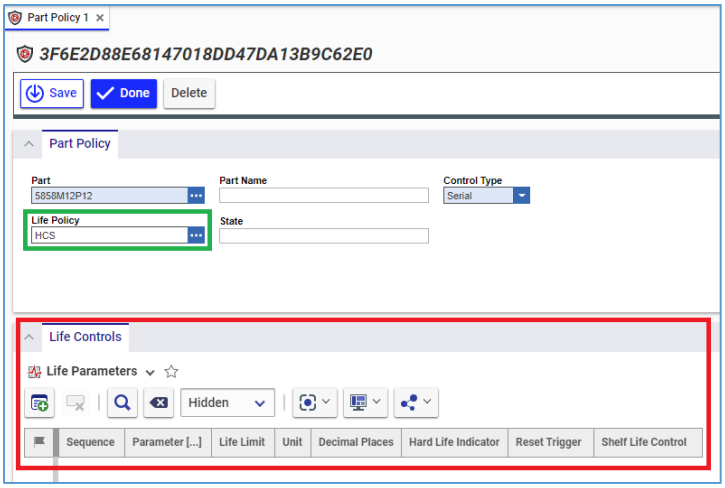


Figure 438.

Note: You can also specify the applied **Life Policy** Item by typing in the **Life Policy** field.

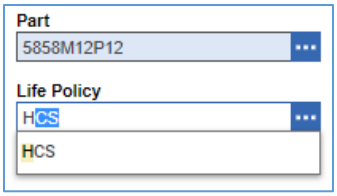


Figure 439.

11. Click **Save** on the **Part Policy** Item toolbar. The **Life Controls Life Parameters** Relationships Grid is automatically populated with the **Life Parameter** Items from the applied **Life Policy** Item.

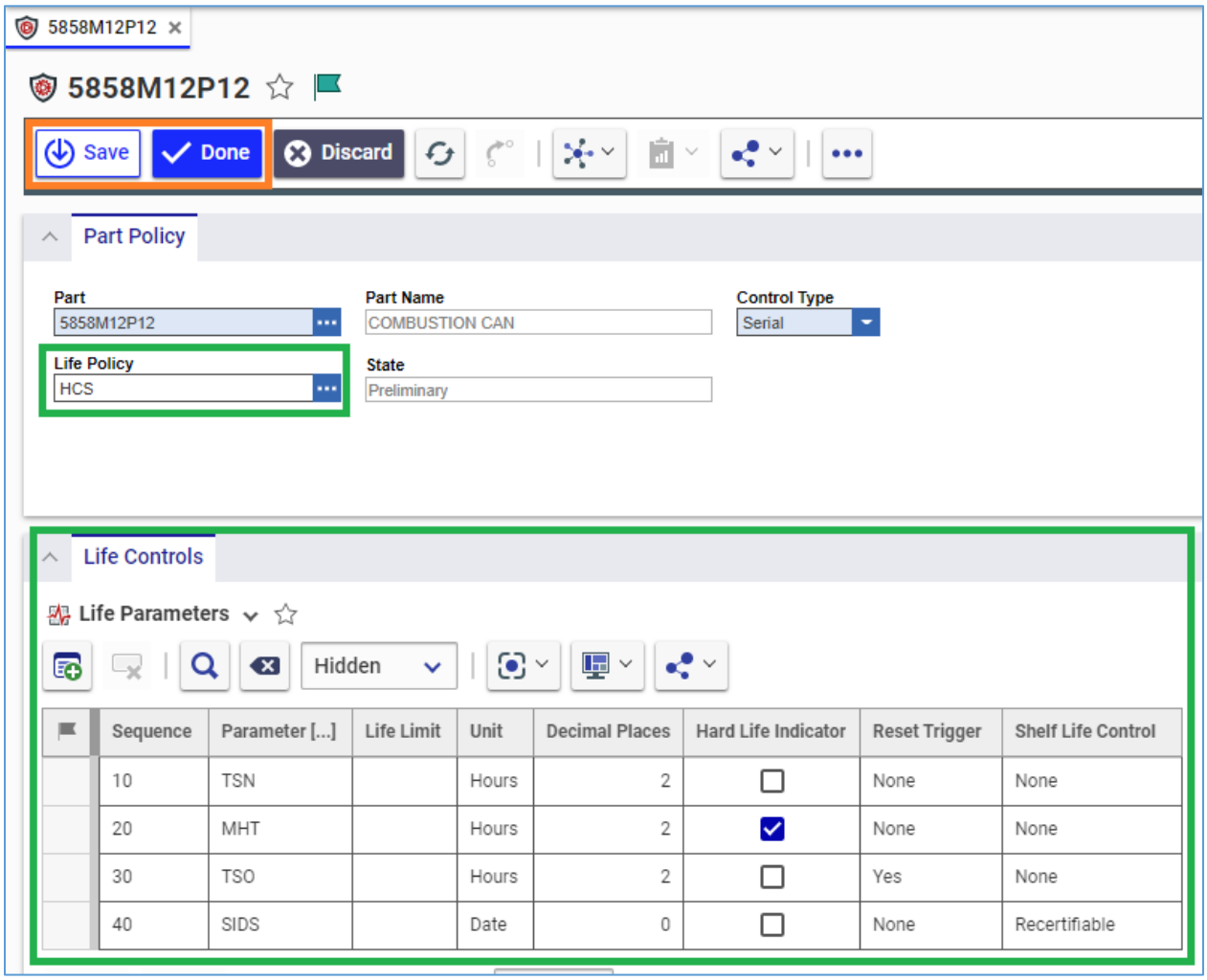


Figure 440.

12. In the **Life Controls Life Parameters** Relationships Grid, specify **Life Limit** values only for the **Life Parameter** Items which have the **Hard Life Indicator** property **selected**.

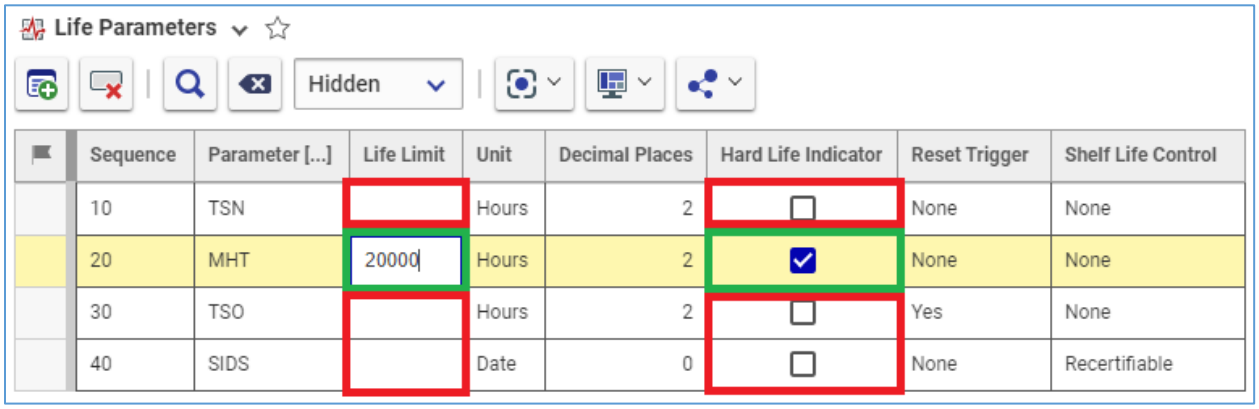


Figure 441.

Note: You may skip steps 11-12 and specify **Life Limit** values where necessary later. These values are required for the **Part Policy** Item promotion to the **Active** State.

13. Click either **Save** or **Done** on the **Part Policy** Item toolbar. The new **Part Policy** Item is created and in the **Preliminary** State.

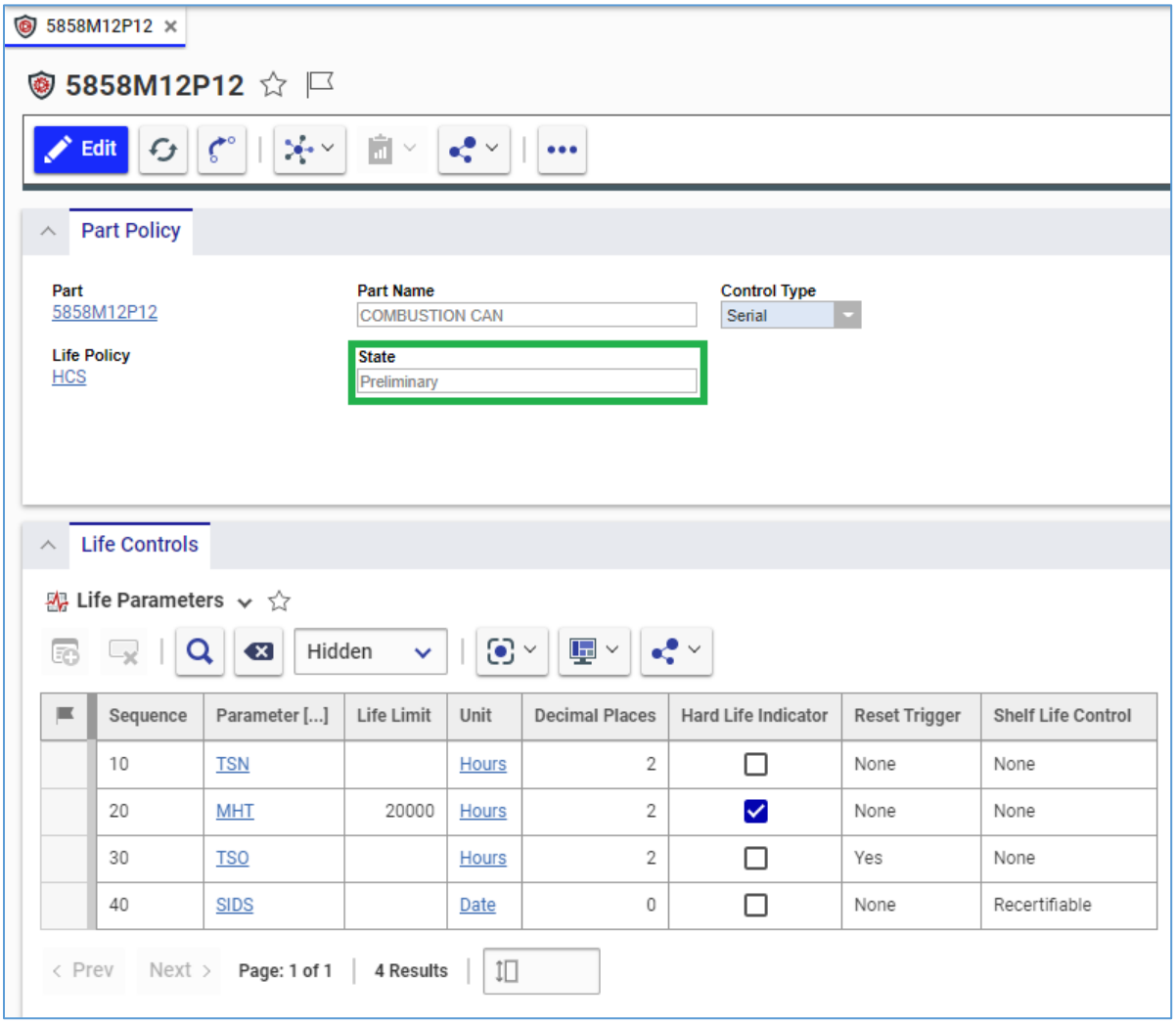


Figure 442.

The **Part Policy** Item **Innovator tab** and **Page Title** bar show the Item **Part** value.

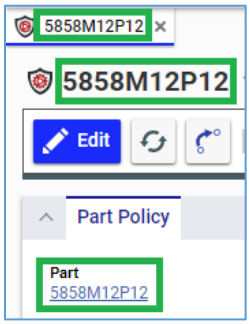


Figure 443.

The **Part Policy** Item is now available via search, subject to the **Asset** Permissions.

If the **Part** Item specified in the **Part Policy** Item is current (latest), the **Part Is Current** property of this **Part Policy** Item is **selected** in the **Part Policy** search Grid.

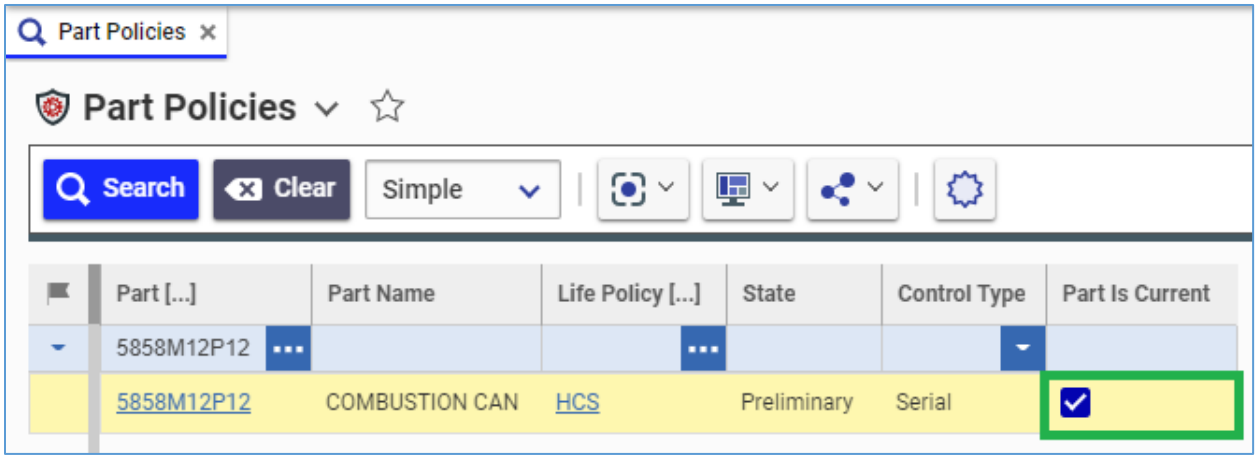


Figure 444.

Upon saving a **Part Policy** Item, the system validates this Item and raises an intelligent error if the validation fails, for example, if the Item has a blank field for a required property or a duplicating **Part** value, discussed in section 9.6.4 Validation of the Part Policies.

9.6.2 Modifying Part Policies

Note: Only an **Asset Admin** or **Asset Editor** can modify a **Part Policy** Item and only when this Item is in the **Preliminary** State. If otherwise, clicking the **Edit** button on the **Part Policy** Item toolbar raises an error.

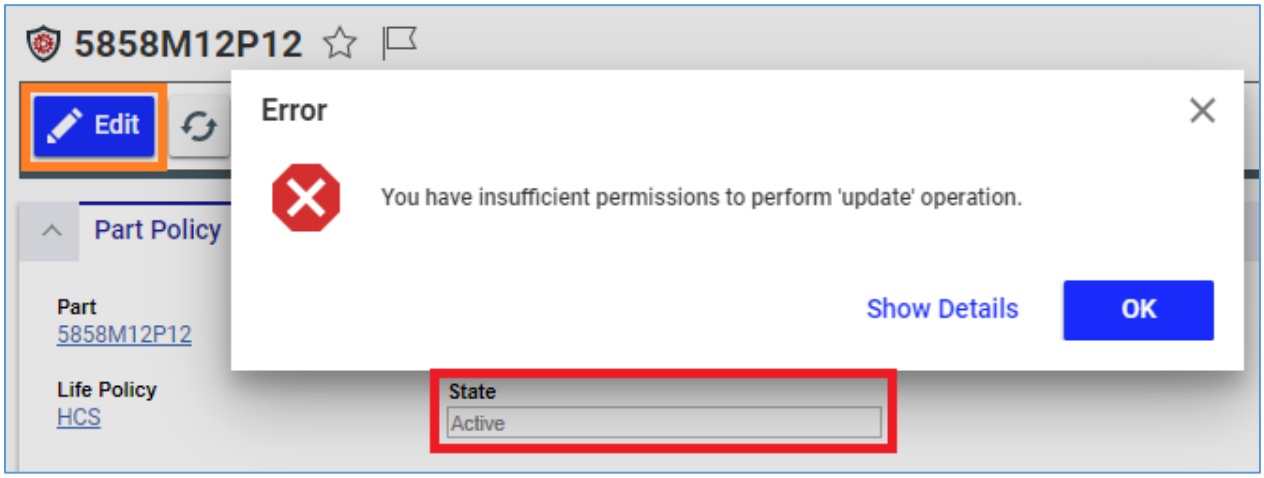


Figure 445.

To modify a **Part Policy** Item, use the standard modifying procedure.

If the **Part** value has been changed, the **Part Policy** Item **Innovator tab** and **Page Title** bar show its new value.

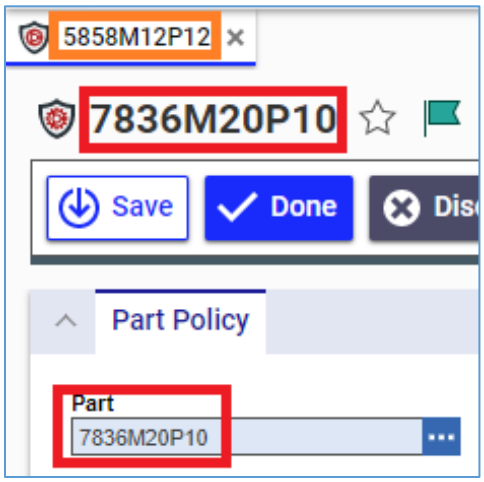


Figure 446.

Upon saving a **Part Policy** Item, the system validates this Item and raises an intelligent error if the validation fails, for example, if the Item has a blank field for a required property or a duplicating **Part** value, discussed in section 9.6.4 Validation of the Part Policies.

9.6.3 Deleting Part Policies

Note: Only an **Asset Admin** or **Asset Editor** can delete a **Part Policy** Item and only when this Item is in the **Preliminary** State. If otherwise, an error is raised.

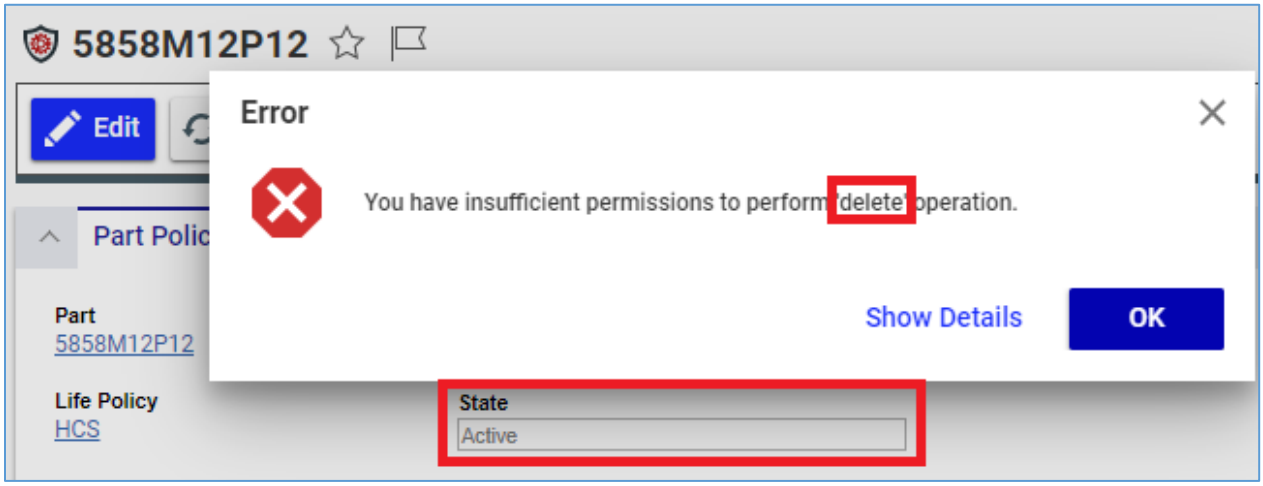


Figure 447.

To delete a **Life Parameter** Item, use the standard deletion procedure.

9.6.4 Validation of Part Policies

Once you click **Save** or **Done** on a **Part Policy** Item toolbar, the system validates this Item and raises an appropriate error if the Item:

- Is missing a **Part** value.

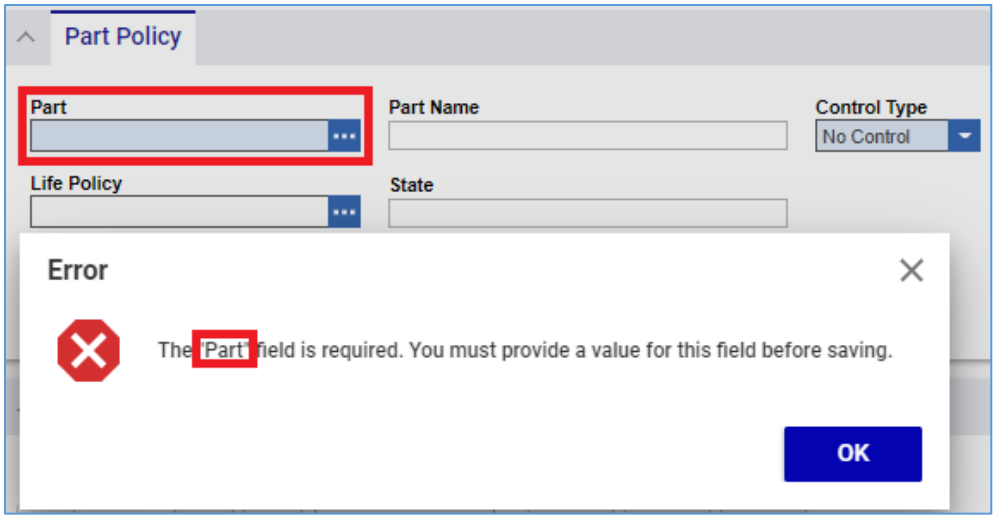


Figure 448.

- Is missing a **Control Type** value.

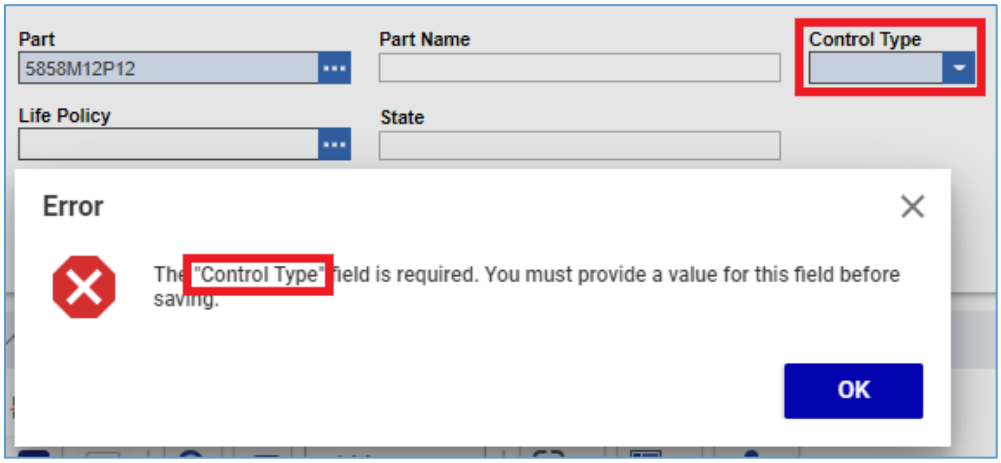


Figure 449.

- Has a **Part** value that is not unique: the given **Part** Item already has an existing **Part Policy** Item. See section 9.4.1 Part Item related to Part Policy Item.

Figure 450.

- Has a specified **Life Policy** Item in the **Preliminary** State. See section 9.4.2 Life Policy Item applied to Part Policy Item.

Figure 451.

- Has a **Control Type** value that a specified **Life Policy** Item does not include. See section 9.4.2 Life Policy Item applied to Part Policy Item.

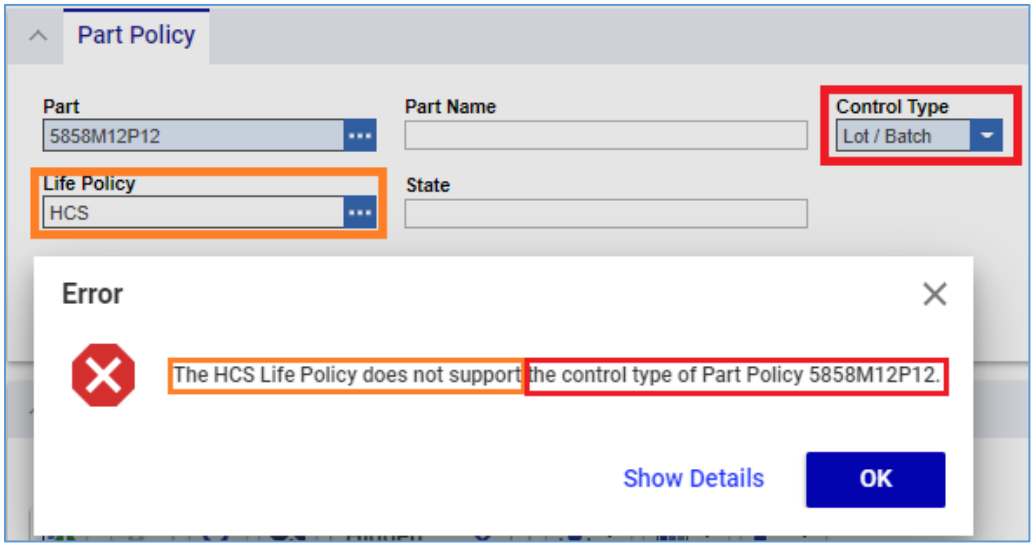


Figure 452.

- Has **Life Limit** values for one or more **Life Parameter** Items in the **Life Controls Life Parameters** Relationships Grid which have the **Hard Life Indicator** property **cleared**. See section 9.4.3 Life Parameter Items related to Part Policy Item.

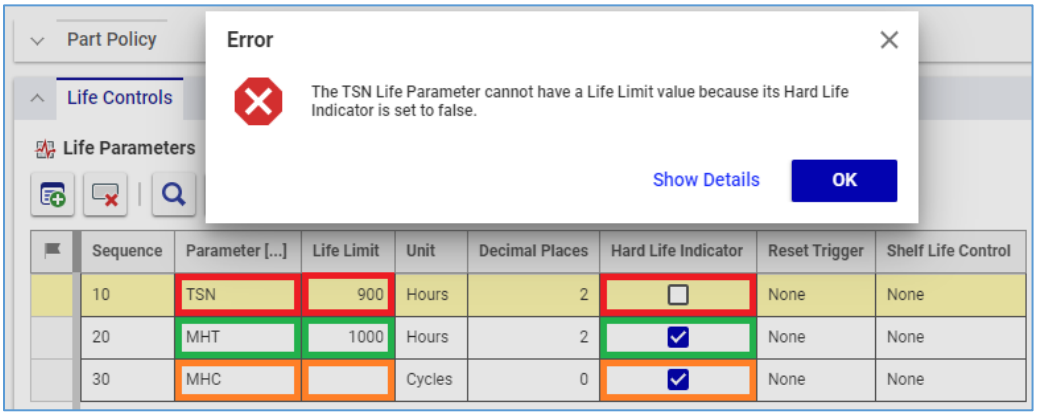


Figure 453.

10 Operational Event Types

DTC 12.0R3 application introduces the **Operational Event Type** and **Operational Event** Items for tracking operational activities and updating life variables of physical assets.

The **Operational Event Type** Items are a prerequisite for using the **Operational Event** Items.

10.1 Understanding Operational Event Types

An **Operational Event Type** Item is a type of an operational activity that a physical asset can perform. For example, the **Flight** Operational Event Type represents the flight activity type: air travel of an aircraft from take-off to landing.

Note: The DTC application is shipped with no **Operational Event Type** Items out of the box. This section uses some industry-agnostic **Operational Event Type** Items as illustrations.

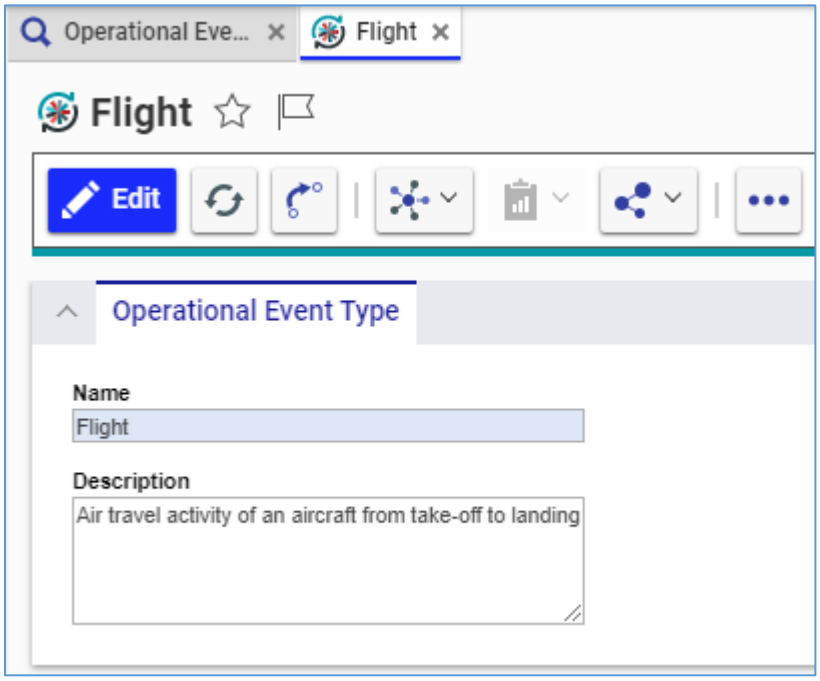


Figure 454.

The **Operational Event Type** Items have no Life Cycle map, thus States.

For the **Operational Event Type** property details, see section 10.3 Operational Event Type properties.

10.2 Operational Event Types Search Grid

The **Operational Event Types** Search Grid has the standard features and the following columns:

- **Name**
- **Description**

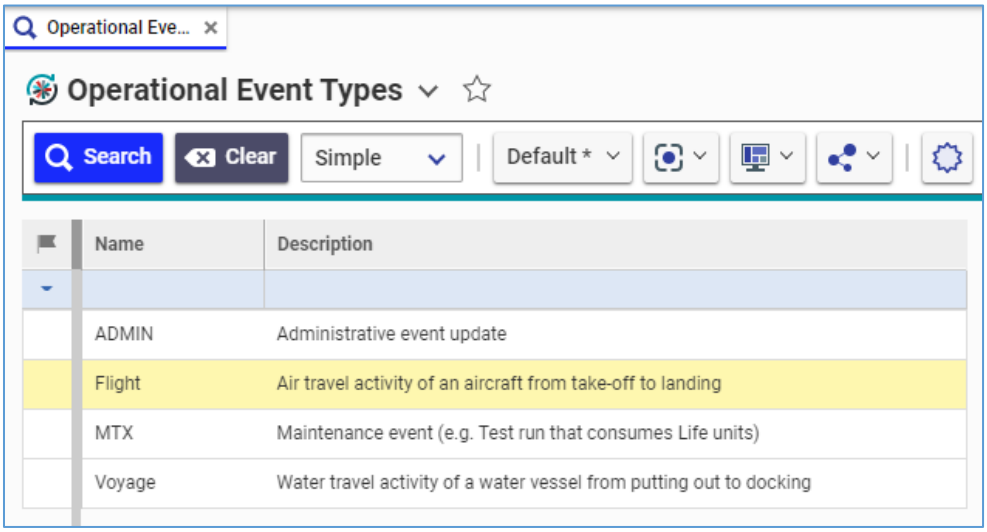


Figure 455.

Each column shows a corresponding **Operational Event Type** Item property. For the property details, see section 10.3 Operational Event Type properties.

The **Operational Event Type** Items are sorted in the Search Grid and other grids by their **Name** values in the ascending order (from A to Z).

10.3 Operational Event Type properties

The following properties of a **Operational Event Type** Item are available at the **Form** accordion of this **Operational Event Type** Item view:

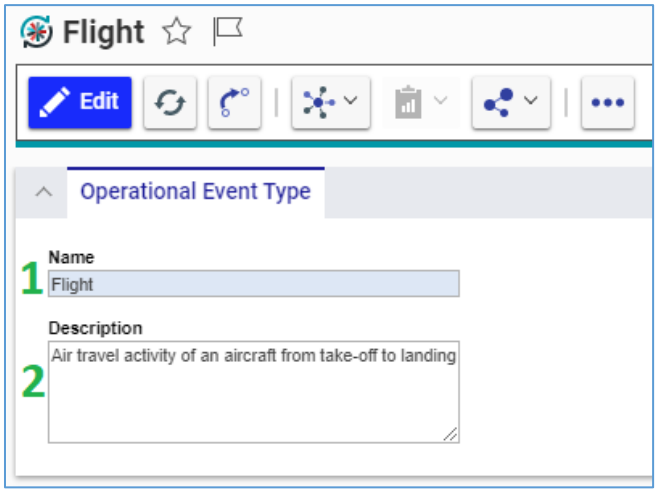


Figure 456.

1. **Name**: a unique name for a given **Operational Event Type** Item among other **Operational Event Type** Items. This property is required and must have a unique text value of up to 32 characters long. The characters can be of any type. The **Name** should briefly explain the **Operational Event Type** Item as a generic or specific type of a physical activity. For example, the **Flight** Operational Event Type Item is generic because it represents the flight of any flying asset. The **Helicopter Flight** Operational Event Type Item is specific because it represents the flight of helicopters.
2. **Description**: detailed or additional information about the **Operational Event Type** Item. This property can have a value of up to 512 characters long. The characters can be of any type.

10.4 Operational Event Type Relationships and usage

The **Operational Event Type** Item has no **Relationships** accordion because it is used in an **Operational Event** Item as a required, foreign property.

Use the **Where Used** browser to explore the usage of a given **Operational Event** Item. See section 12 The Where Used Browser and DTC Items. Also, you can filter the **Operational Event** Item Search Grid by the given **Operational Event** Item.

10.5 Managing Operational Event Types

Your ability to manage an **Operational Event Type** Item depends on your Operational Event Identity membership and Item usage. This section assumes that you are familiar with the **Operational Event Type** Item properties; discussed in section 10.3 Operational Event Type properties.

10.5.1 Creating Operational Event Types

Note: Only a member of the **Operational Event Reviewer** Identity can create the **Operational Event Type** Items.

To create a new **Operational Event Type** Item:

1. Go to **Contents** --> **Assets** --> **Operational Event Type**.
2. Click **Operational Event Type**. The **Operational Event Type #** Item view appears.

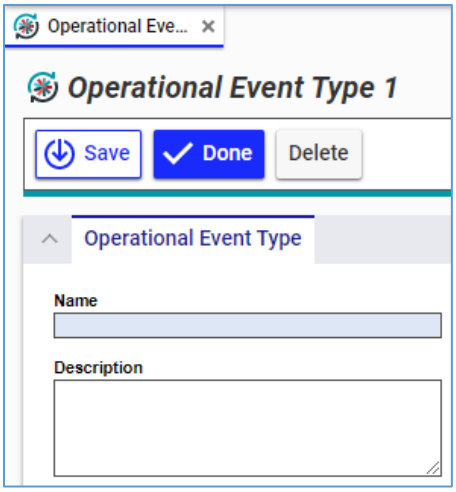


Figure 457.

3. In the **Name** field, enter a unique name for the **Operational Event Type** Item.

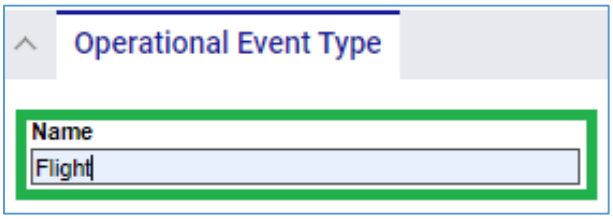


Figure 458.

4. In the **Description** field, enter additional or explanatory information on the **Operational Event Type** Item.

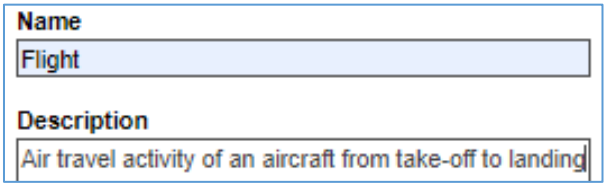


Figure 459.

5. Click either **Save** or **Done** on the **Operational Event Type** Item toolbar. The new **Operational Event Type** Item is created. The **Operational Event Type** Item **Innovator tab** and **Page Title** bar show the Item **Name** value.

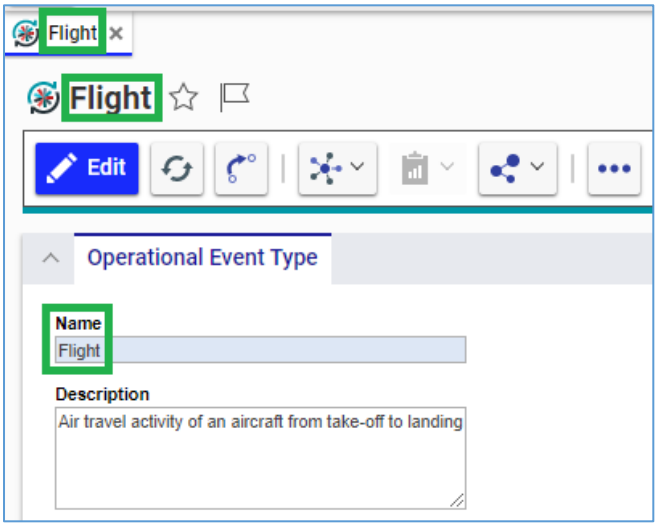


Figure 460.

The **Operational Event Type** Item is now available via search, subject to the **Asset** Permissions.

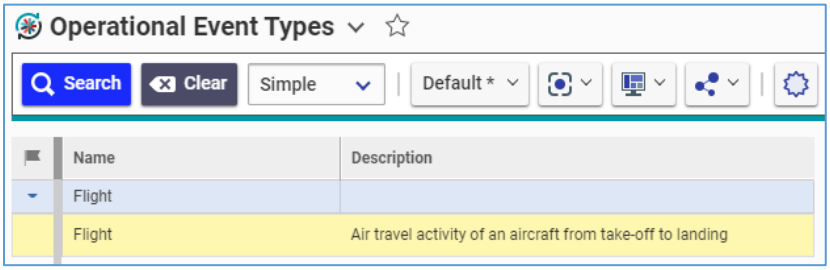


Figure 461.

An error is raised when trying to save an **Operational Event Type** Item with a blank field for the **Name** property or a duplicating **Name** property value. For details, see section 10.5.4 Validation of Operational Event Types.

10.5.2 Modifying Operational Event Types

Note: Only a member of the **Operational Event Reviewer** Identity can modify the properties of a **Operational Event Type** Item and only when this Item is not used. If otherwise, clicking the **Edit** button on the **Operational Event Type** Item toolbar raises an error.

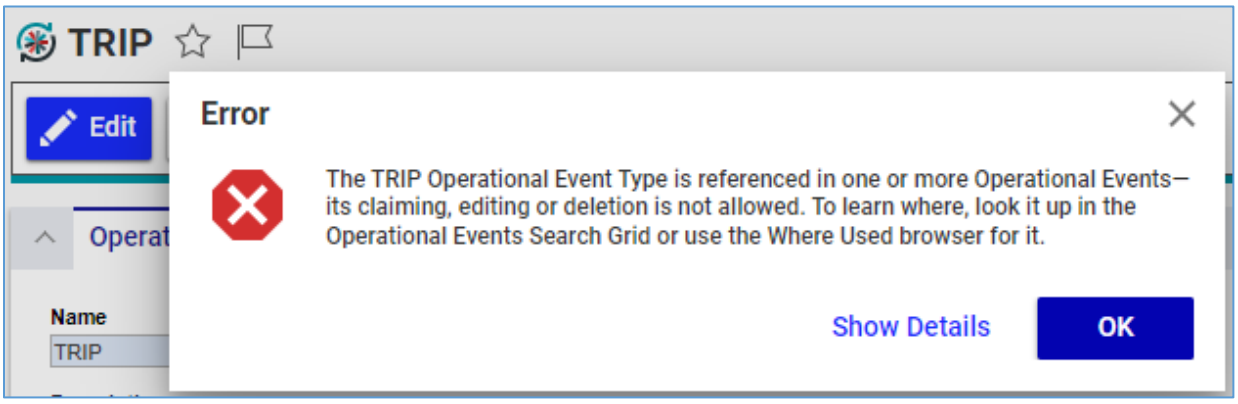


Figure 462.

To modify an **Operational Event Type** Item, use the standard modifying procedure.

If the **Name** value has been changed, the **Operational Event Type** Item **Innovator tab** and **Page Title** bar show its new value.

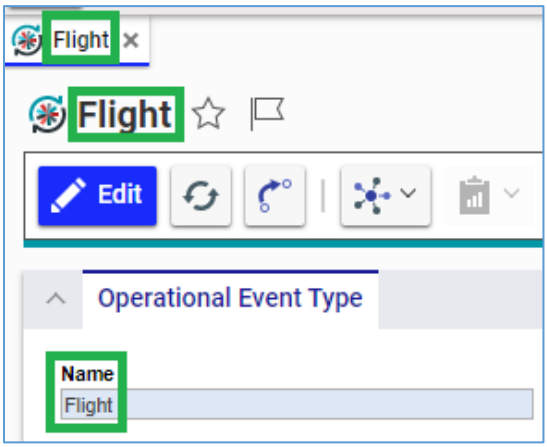


Figure 463.

An error is raised when trying to save an **Operational Event Type** Item with a blank field for the **Name** property or a duplicating **Name** property value. For details, see section 10.5.4 Validation of Operational Event Types.

10.5.3 Deleting Operational Event Types

Note: Only a member of the **Operational Event Reviewer** Identity can delete a **Life Unit** Item and only when this Item is not used. If otherwise, an error is raised.

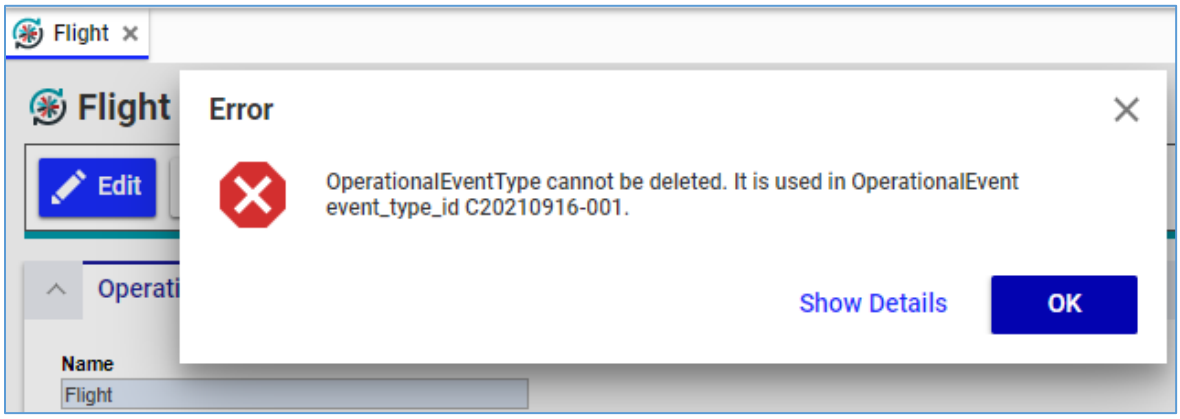


Figure 464.

To delete an **Operational Event Type** Item, use the standard deletion procedure.

10.5.4 Validation of Operational Event Types

Once you click **Save** or **Done** on a **Life Unit** Item toolbar, the system validates this Item and raises an appropriate error if the Item:

- Is missing a **Name** value.

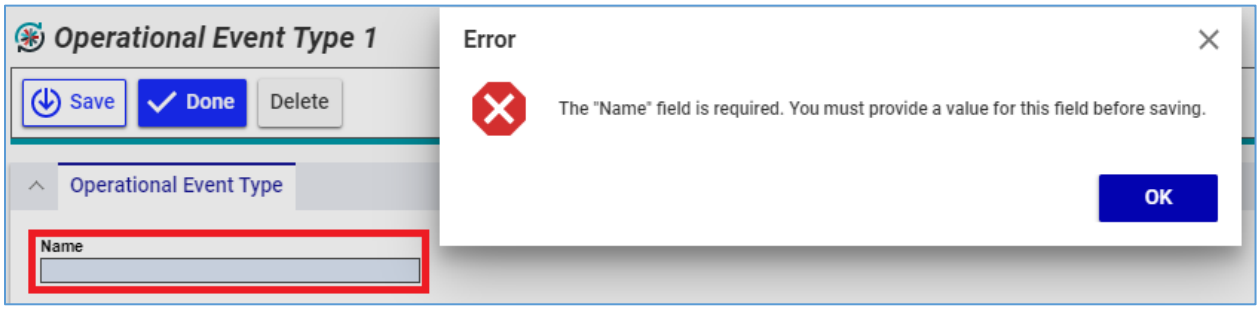


Figure 465.

- Has a **Name** value that is not unique: already given to an existing **Life Unit**.

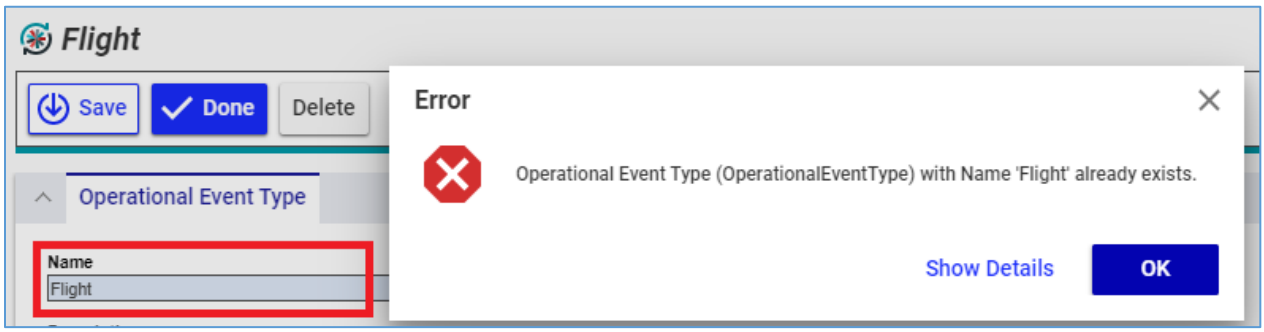


Figure 466.

11 Operational Events

Having all necessary Operational Event Types, you can create and use Operational Events to track operational activities and update Life Parameters of Physical Parts.

11.1 Understanding Operational Events

Life characteristics of an asset are constantly changing while the asset is operating. An operational activity of the asset consumes some piece of the asset life, changing asset life variables. For example, one flight costs one landing, one engine start, some flying (working) hours of an airplane, and so on. The values of airplane life variables should be adjusted accordingly after each flight to track the current airplane life state. Each such adjustment is an increment of a life variable.

An **Operational Event** Item digitally represents a discrete in-field operational activity that a single real-world asset has performed over some time. For example, an **Operational Event** Item can be a regular passenger airplane flight. The **Operational Event** Item includes the following data:

- Operational activity parameters, like a flight number, start and end time, departure and arrival airports in our example.
- Increments of life variables that a given single asset has accumulated with the operational activity completion, like the number of starts, landings, cycles, and working hours during the given flight in our example.

Note: The DTC application is shipped with no **Operational Event** Items out of the box. This section uses some industry-agnostic **Operational Event** Items as illustrations.

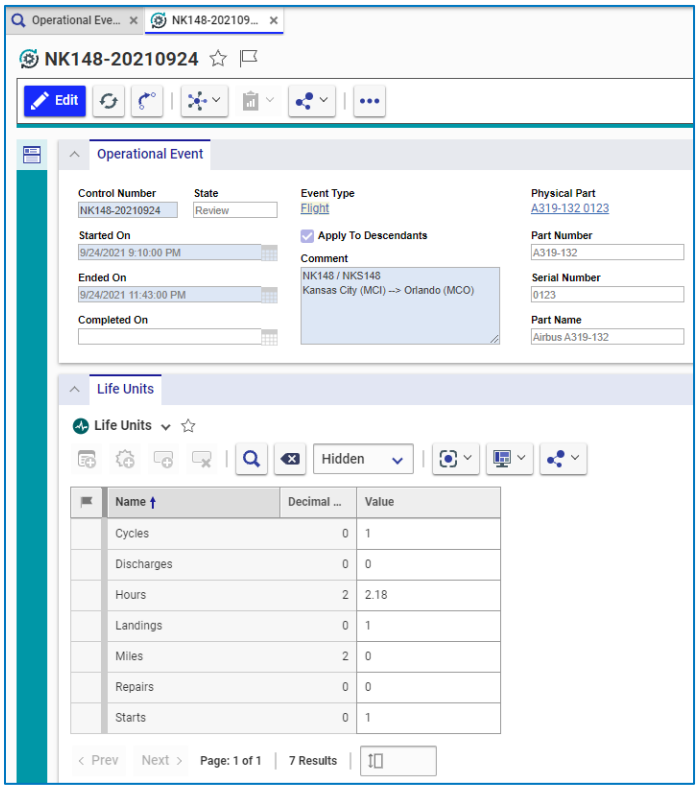


Figure 467.

The **Operational Event** Item gathers the asset life variable increments on its **Life Units** tab as **OperationalEvent LifeUnit** Relationship Items with the **Life Unit** Items; see section 11.4.3 OperationalEvent LifeUnit Relationships.

The system always automatically updates the numeric **Life Parameters** of a given **Physical Part** Item upon the **Operational Event** Item promotion to the **Complete** State. See section 11.5 Operational Event Life Cycle and promotion.

By default setting, the system also updates the numeric **Life Parameters** of the given **Physical Part** Item children. You can set the **Operational Event** Item to update only the given **Physical Part** Item without propagating down its Physical Parts BOM structure.

For the **Operational Event** property details, see section 11.3 Operational Event Properties.

Operational Event Item deletion or update is conditionally limited. For details, refer to section 11.6 Managing Operational Events.

11.2 Operational Events Search Grid

The **Operational Events** Search Grid has the standard features and the following columns:

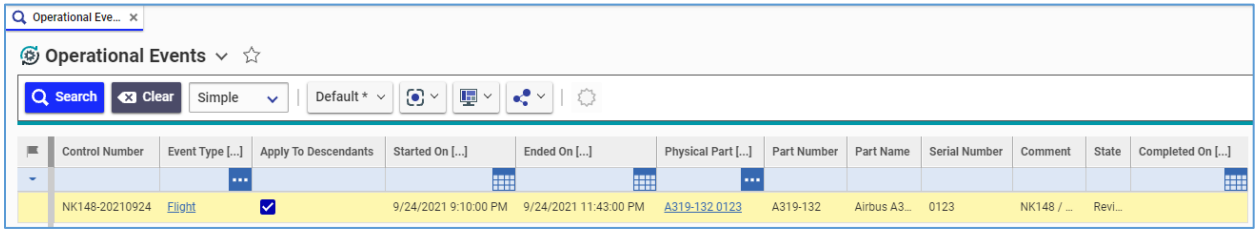


Figure 468.

- **Control Number**
- **Event Type**
- **Apply To Descendants**
- **Started On**
- **Ended On**
- **Physical Part**
- **Part Number**
- **Part Name**
- **Serial Number**
- **Comment**
- **State**
- **Completed On**

Each column shows a matching **Operational Event** Item property. For the property details, see section 11.3 Operational Event Properties.

The **Operational Event** Items are sorted in the Search Grid and other grids by their **Control Number** values in the ascending order (from A to Z).

11.3 Operational Event Properties

The following properties of an **Operational Event** Item are available in the **Form** accordion of this **Operational Event** Item view:

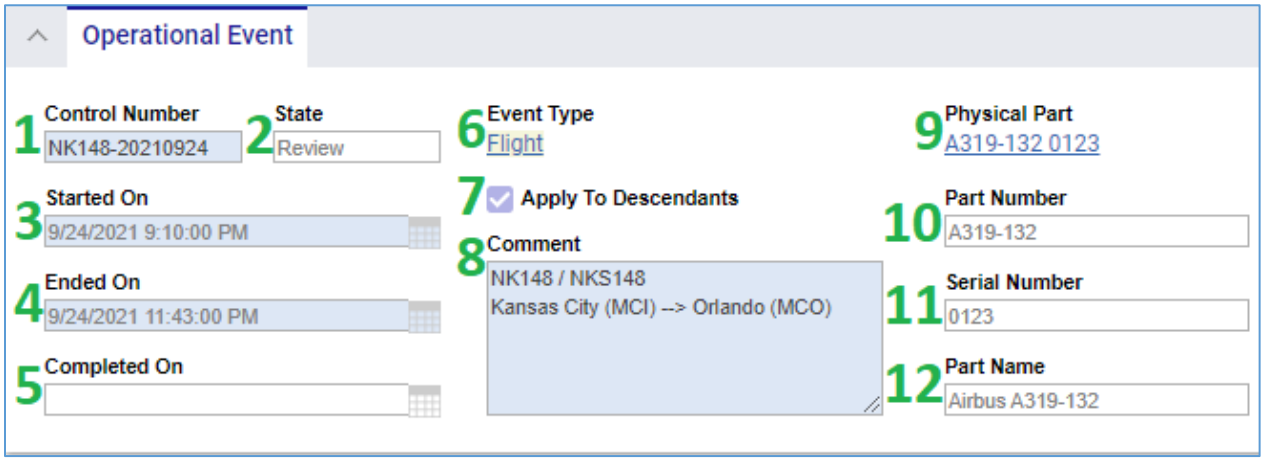


Figure 469.

1. **Control Number**: a unique identification number of a given **Operational Event** Item among other **Operational Event** Items. This property is required and must have a unique value of up to 64 characters long. The characters can be of any type, except white space characters: the system automatically clears an entered value from white spaces upon saving an **Operational Event** Item.
2. **State**: a current State of this **Operational Event** Item. This property is view-only. See section 11.5 Operational Event Life Cycle and promotion.
3. **Started On**: date and time when a represented real-world operational activity started. This property is required. A value legal for this property should be equal to or earlier than the one given in the **Ended On** property (4).
4. **Ended On**: date and time when the represented real-world operational activity ended. This property is required. A value legal for this property should be equal to or later than the one given in the **Started On** property (3).
5. **Completed On**: date and time when this **Operational Event** Item is completed: promoted to the **Complete** State. This property is view-only. The system populates this property upon the discussed promotion. See section 11.5 Operational Event Life Cycle and promotion.
6. **Event Type**: a type of the represented real-world operational activity defined as an **Operational Event Type** Item. This property is required and also a link to this **Operational Event Type** Item. See section 11.4.1 Operational Event Item and Operational Event Type Item connection.
7. **Apply To Descendants**: a check box that defines to update or not the numeric **Life Parameter** values of the **Physical Part** Items that are the qualified children of a **Physical Part** Item given in the **Physical Part** property (9):
   a. Selected (**true**): to update the children.
   b. Cleared (**false**) not to update the children.

   This property is required and set to **true** by default when creating a new **Operational Event** Item.
8. **Comment**: detailed or additional information about the represented real-world operational activity. This property is required and can have a value of up to 512 characters long. The characters can be of any type.

9. **Physical Part**: a **Physical Part** Item representing a real-world asset that has performed the given operational activity. This property is required and also a link to this **Physical Part** Item. See section 11.4.2 Operational Event Item and Physical Part Item connection
The system always updates the numeric **Life Parameters** of this **Physical Part** Item upon the **Operational Event** Item promotion to the **Complete** State. See section 11.5 Operational Event Life Cycle and promotion.
10. **Part**: a **Part Number** property value of the **Physical Part** Item given in the **Physical Part** property (9). This property is foreign and view-only.
11. **Serial Number**: a **Serial Number** property value of the **Physical Part** Item given in the **Physical Part** property (9). This property is foreign and view-only.
12. **Part**: a **Part Name** property value of the **Physical Part** Item given in the **Physical Part** property (9). This property is foreign and view-only.

11.4 Operational Event Relationships and Usage

An **Operational Event** Item requires the following Items:

- **Operational Event Type**
- **Physical Part**

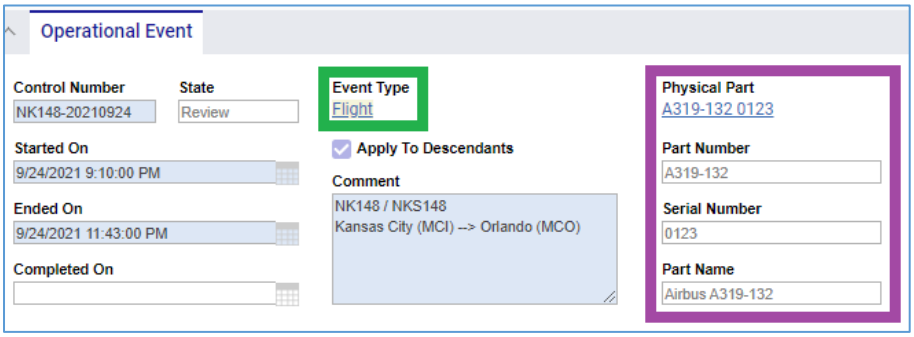


Figure 470.

The **Operational Event** Item also includes Relationship Items with the **Life Unit** Items.

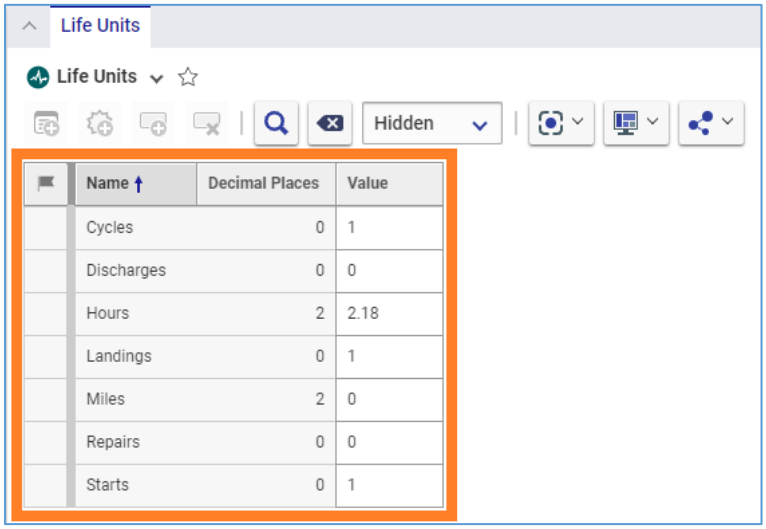


Figure 471.

11.4.1 Operational Event Item and its Operational Event Type Item

An **Operational Event Type** Item set in the **Event Type** property of a given **Operational Event** Item defines an operational activity type for this **Operational Event** Item; see 10.1 Understanding Operational Event Types.

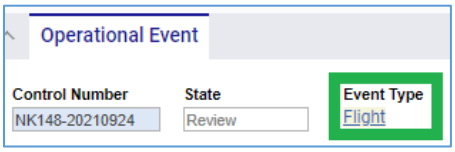


Figure 472.

There is a one-to-many connection between **Operational Event Type** and **Operational Event** Items:

- One **Operational Event Type** Item can be given to zero, one, or more than one **Operational Event** Items.
- One **Operational Event** Item can have only one **Operational Event Type** Item given.

The **Event Type** property is required. Depending on the current **Operational Event** Item State, a Member of a dedicated Identity can reset this property to another **Operational Event Type** Item as follows:

- **Preliminary**: **Operational Event Creator**
- **Review**: **Operational Event Reviewer**
- **Complete**: **Operational Event Admin**

11.4.2 Operational Event Item and Physical Part Item connection

The **Physical Part** property of a given **Operational Event** Item defines a **Physical Part** Item expressing a real-world asset that has performed an operational activity denoted by the given **Operational Event** Item. This **Operational Event** Item will be applied to this **Physical Part** Item when promoted to the **Complete** State; see section 11.5 Operational Event Life Cycle and Promotion.

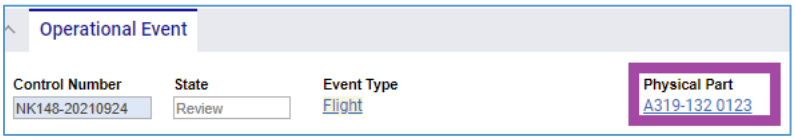


Figure 473.

There is a one-to-many connection between **Physical Part** and **Operational Event** Items:

- One **Physical Part** Item can have zero, one, or more than one **Operational Event** Items applied to it.
- One **Operational Event** Item can be applied only to one **Physical Part** Item.

The **Physical Part** property is required. Depending on the current **Operational Event** Item State, a Member of a dedicated Identity can reset this property to another **Physical Part** Item as follows:

- **Preliminary**: **Operational Event Creator**
- **Review**: **Operational Event Reviewer**
- **Complete**: nobody

An **Operational Event** Item can be applied only to a serial-controlled **Physical Part** Item in the **Active** State because an **Operational Event** Item represents a discrete operational activity completed in the field by a single asset. We cannot track each lot- or non-controlled asset (**Physical Part** Item) due to its

nature. And a **Physical Part** Item in the **Active** State stands for an asset commissioned into service. An error is raised if trying to save an **Operational Event** Item that violates these rules; for the error messages, see section 11.6.4 Validation of Operational Events.

11.4.3 OperationalEvent LifeUnit Relationships

An **Operational Event** Item has the **Life Units** Relationships tab for **OperationalEvent LifeUnit** Relationship Items. Each such Relationship Item is a record of a single life variable increment gained by a given asset with a given operational activity.

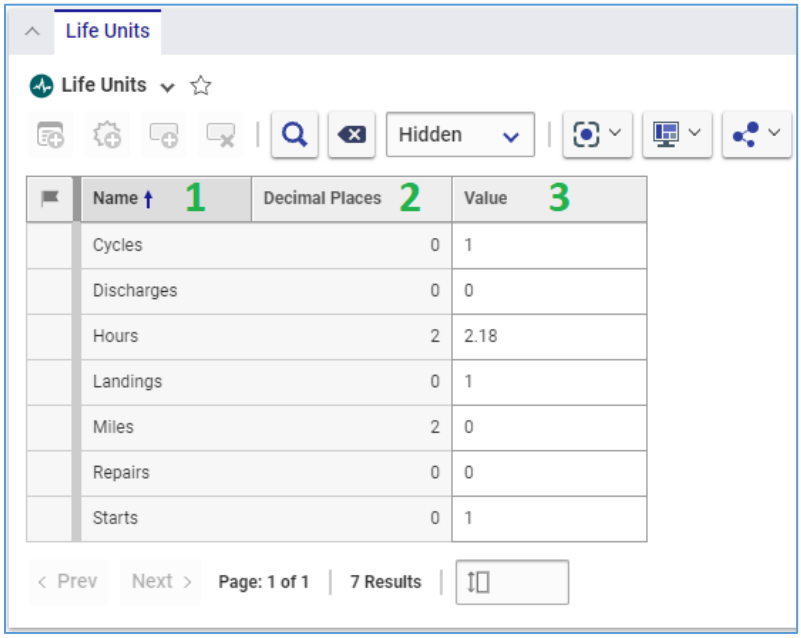


Figure 474.

This tab has the **Life Units** Relationships Grid that shows a given **OperationalEvent LifeUnit** Relationship Item as follows:

1. **Name**: a view-only **Name** of a given related **Life Unit** Item.
2. **Decimal Places**: a view-only maximum number of digits after a decimal point for a number in the corresponding **Value** property (3).

   If trying to save the **Operational Event** Item that has a number in a **Value** property (3) with more digits after a decimal point that the given **Decimal Places** property value allows, an error dialog appears. For the error message, see section 11.6.4 Validation of Operational Events.
3. **Value**: a numeric value of a corresponding life variable increment. Its default value is **0** (zero). This property is editable and required: it cannot be empty in any State of the **Operational Event** Item.

   If trying to save the **Operational Event** Item with no value a **Value** cell, an error pop-up dialog appears, and the default **0** value is set in these cells. For the error message, see section 11.6.4 Validation of Operational Events.

   The **0** value means no increment for a given life variable. It should be set for the **Life Unit** Items describing asset life characteristics which the **Operational Event** Relationship Item has not consumed.

   You should populate a **Value** cell only with real, true data for a given **Life Unit**: the **Operational Event** Item has consumed some asset life characteristic measured by this **Life Unit** Item. That data will be added to all corresponding **Current Value** property values of a **Physical Part** Item to which

this **Operational Event** Item will be applied when promoted to the **Complete** State; see section 11.5 Operational Event Life Cycle and Promotion.

In the Grid, the **OperationalEvent LifeUnit** Relationship Items are sorted by their **Name** values in the ascending order (from A to Z).

The **Life Units** Relationships Grid is managed automatically. Manual management is prohibited. It is initially empty during the creation of a new **Operational Event** Item.

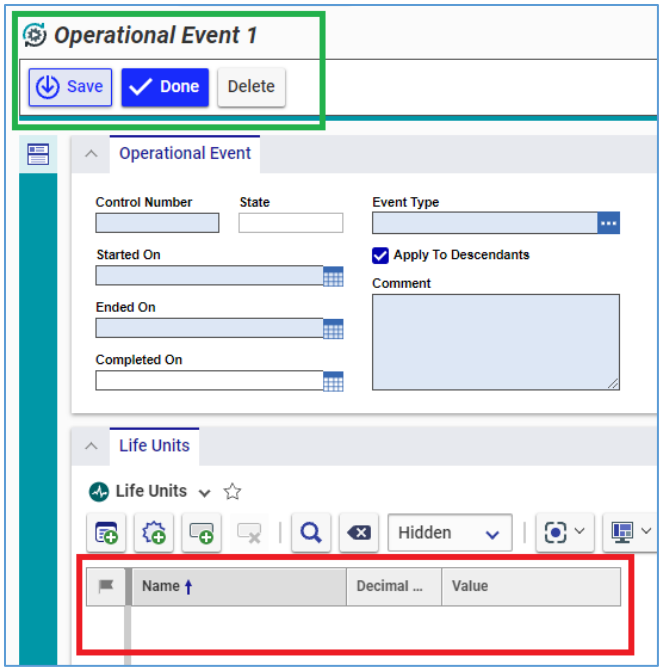


Figure 475.

Once you populate all required properties (1) and save the new **Operational Event** Item (2), the system automatically populates the **Life Units** Relationships Grid.

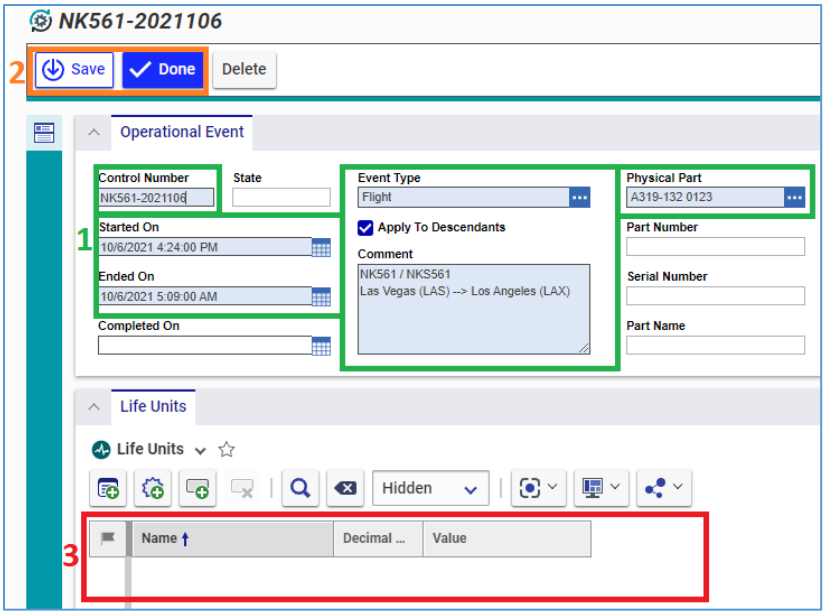


Figure 476.

All **Value** properties of the automatically added **OperationalEvent LifeUnit** Relationship Items have the default value of **0**.

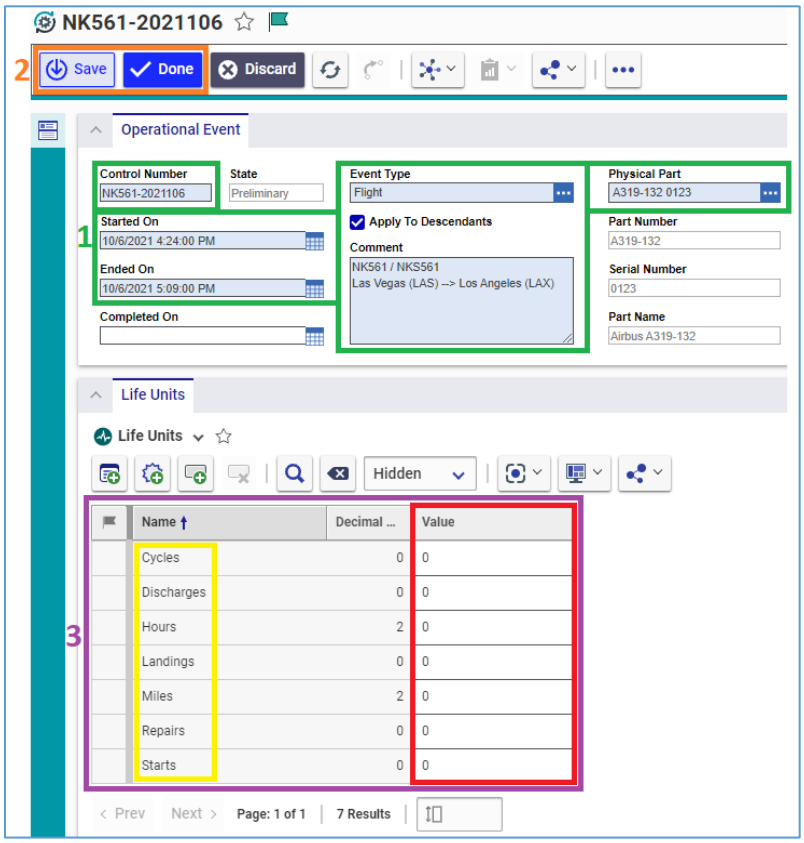


Figure 477.

The system populates the **Life Units** Relationships Grid with Relationship Items for all **Life Unit** Items that both:

- Exist in the database at the moment of saving the **Operational Event** Item.
- Are not dates.

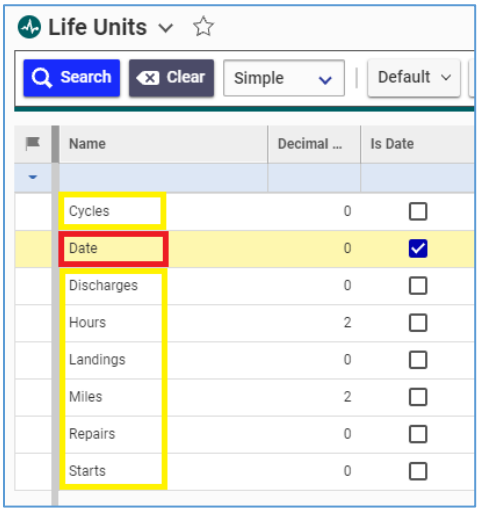


Figure 478.

A **Life Unit** Item can be created after the **Operational Event** Item.

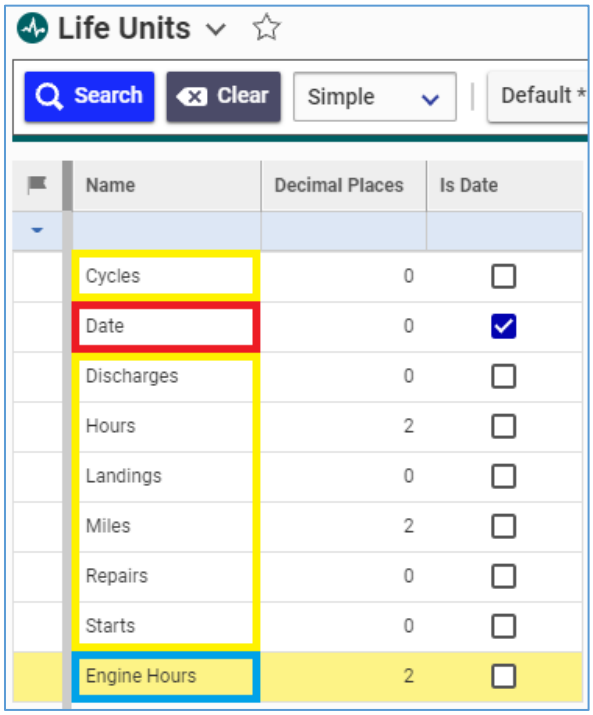


Figure 479.

In such a case, the **Operational Event** Item does not include a Relationship Item with this new **Life Unit** Item.

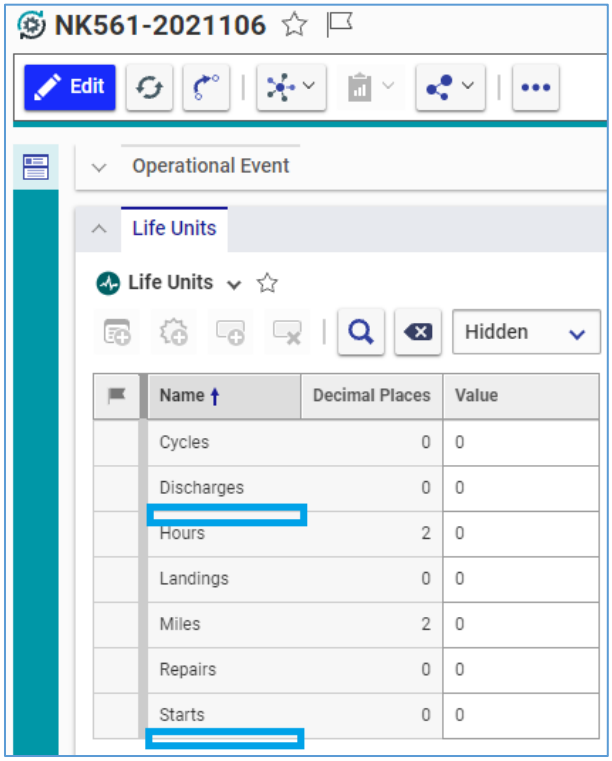


Figure 480.

If the existing **Operational Event** Item should include a Relationship Item with the later added **Life Unit** Item, you just need to resave the **Operational Event** Item.

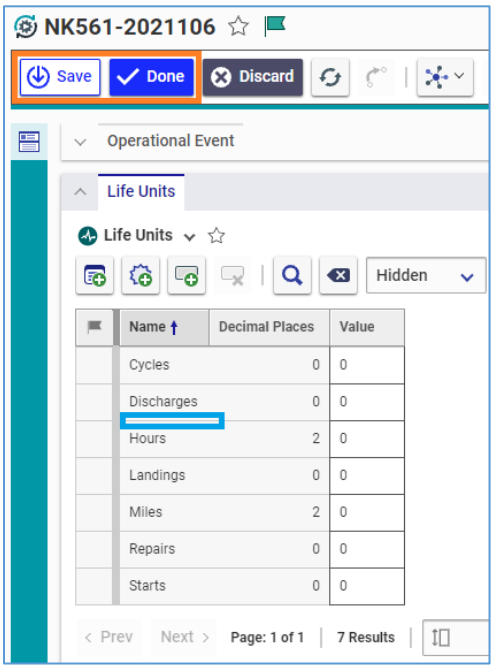


Figure 481.

And, the system will add a Relationship Item with this **Life Unit** Item to the **Life Units** Grid at a position given according to current sorting rules.

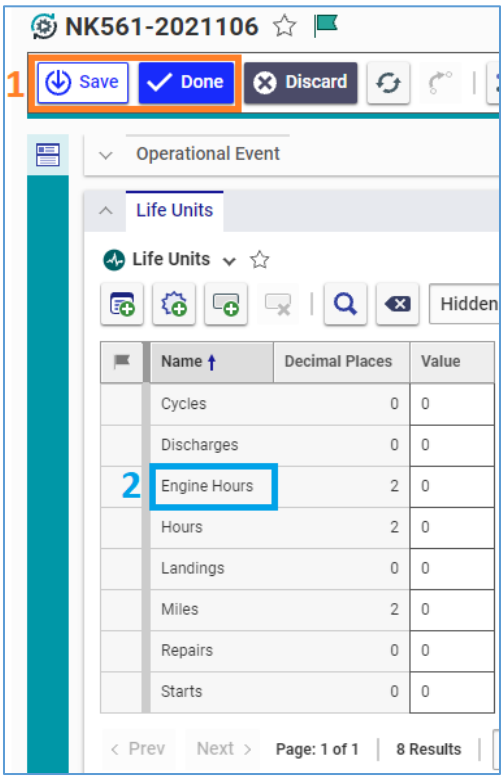


Figure 482.

Once the **Life Units** Relationships Grid is populated, it cannot be blank again. Its **OperationalEvent LifeUnit** Relationship Items cannot be removed, replaced, or deleted.

When editing an **Operational Event** Item, the **Life Units** Relationships Grid has its toolbar and action menu enabled. But still, the manual management of this Grid is forbidden—an error is raised upon saving the **Operational Event** Item after adding, editing, or removing an **OperationalEvent LifeUnit** Relationship Item with any toolbar or action menu button. For details, see section 3.7 Automatically managed DTC Relationships Grids.

11.5 Operational Event Life Cycle and Promotion

Note: A current State of an **Operational Event** Item determines which Operational Event Identity can promote this Item and to which State. See this section for details.

Use the standard Item promotion procedure to promote the **Operational Event** Items.

The **OperationalEvent** Life Cycle Map has the following States for the **Operational Event** Items:

1. **Preliminary**
2. **Review**
3. **Complete**

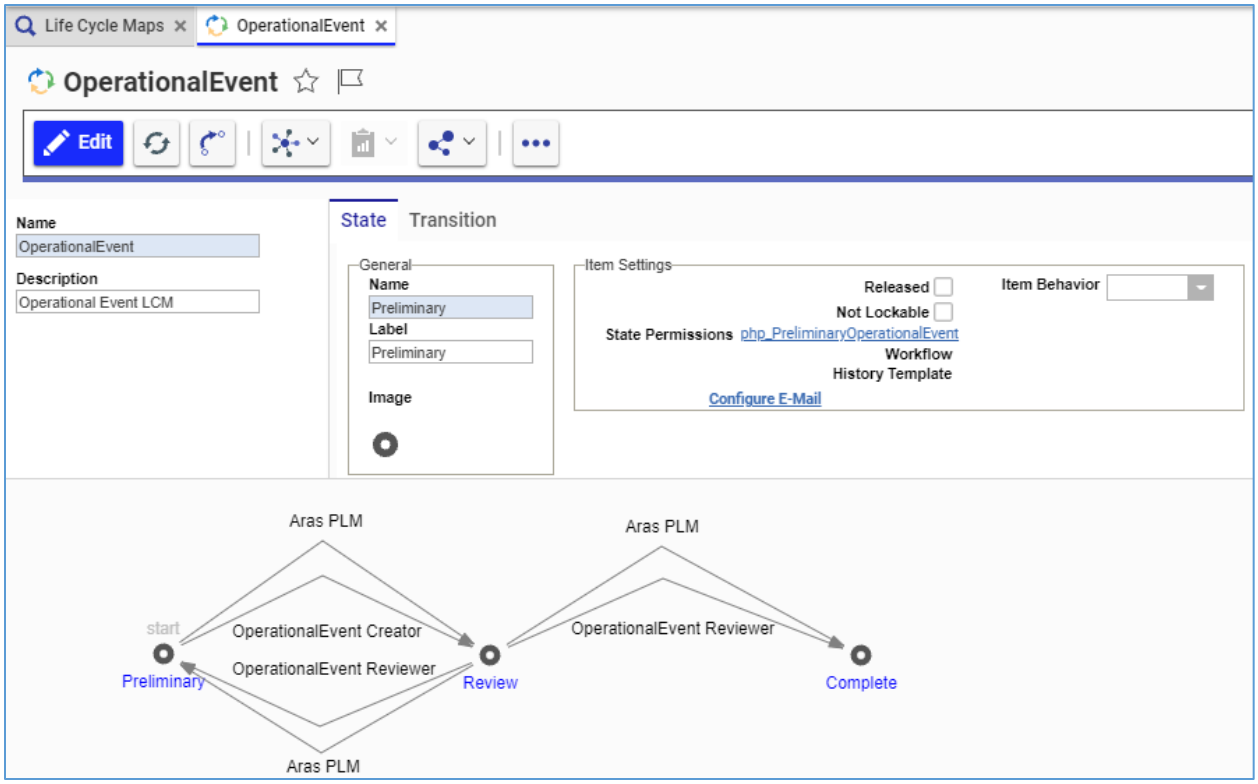


Figure 483.

The **OperationalEvent** Life Cycle Map defines the approval workflow for an **Operational Event** Item:

1. Having required information about a real-world operational activity completed by an asset, a Member of the **Operational Event Creator** Identity creates a new **Operational Event** Item as a digital representation of this activity. The newly created **Operational Event** Item is in the

Preliminary State, where it is a draft. The **Operational Event Creator** Identity can modify or delete the **Operational Event** Item.

Once the **Operational Event** Item includes all valid data about the operational activity, the **Operational Event Creator** Identity promotes the **Operational Event** Item from the **Preliminary** to **Review** State.

2. In the **Review** State, a Member of the **Operational Event Reviewer** Identity verifies that the **Operational Event** Item is free of errors and does not miss pieces of the real-world activity data. The **Operational Event Reviewer** Identity can modify or delete the **Operational Event** Item.

   Once verification is completed, the **Operational Event Reviewer** Identity promotes the **Operational Event** Item from the **Preliminary** to one of the following States:

   a. **Preliminary**—to return the **Operational Event** Item to the **Operational Event Creator** Identity for correction.

   b. **Complete**—to approve the **Operational Event** Item and to add its increments to life variables of one or more given **Physical Part** Items.

3. In the **Complete** State, the **Operational Event** Item is approved, finalized, and its life increments are added to the life variables of the given **Physical Part** Items. No one can delete or promote the **Operational Event** Item to other States. It cannot be rolled back or reapplied to given or other **Physical Part** Items.

   A member of the **Operational Event Admin** Identity can correct some errors in the **Operational Event** Item properties.

During the **Review** to **Complete** promotion, the system always applies the **Operational Event** Item to a **Physical Part** Item given in the **Physical Part** property of this **Operational Event** Item.

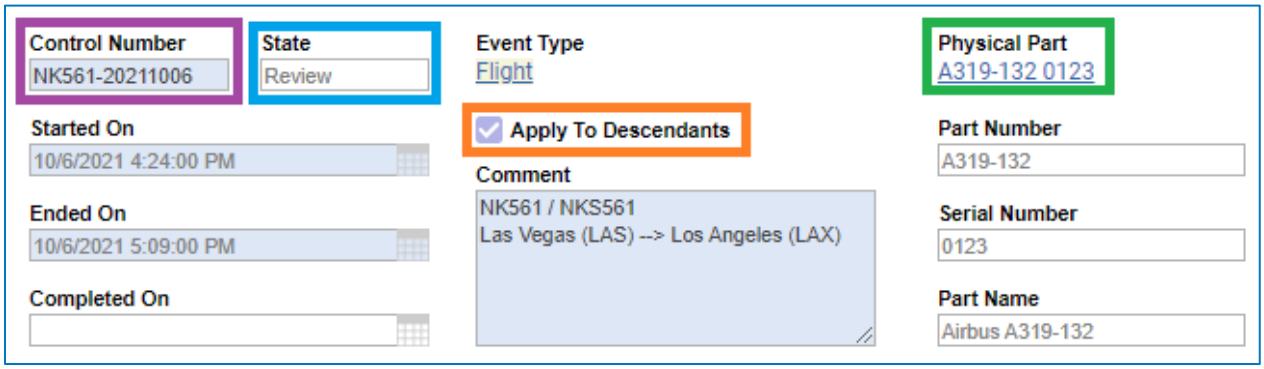


Figure 484.

If the **Operational Event** Item has its **Apply To Descendants** property set to **true**, the system also traverses down the multi-level Physical Part BOM structure where the given **Physical Part** Item is a root Item (parent) and applies the **Operational Event** Item to qualified serial-controlled **Physical Part** Items in this structure (children). For details on the children eligibility and **Operational Event** Item applying, refer to section 4.7.2.3 Operational Event update of life values.

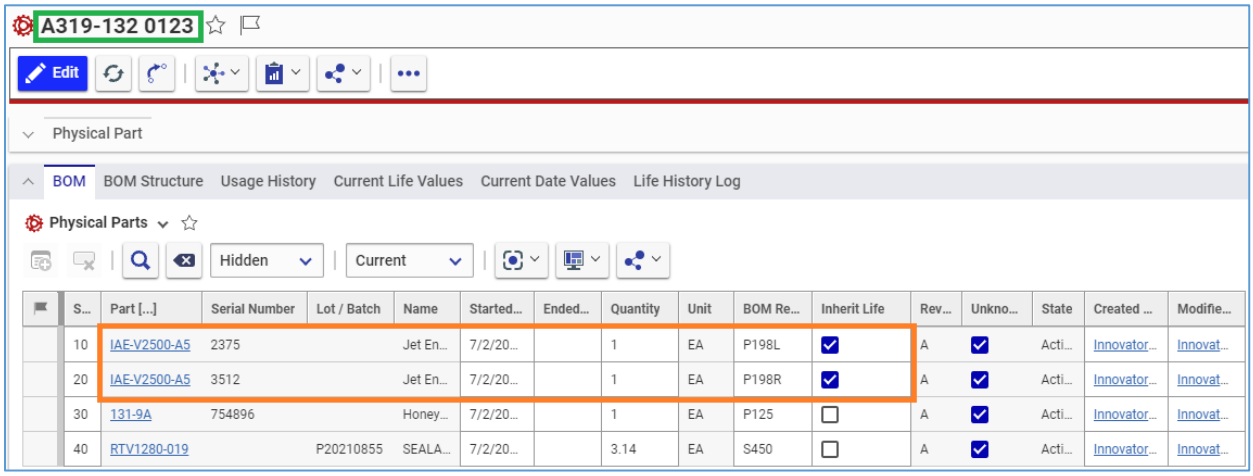


Figure 485.

If you are going to use a **Physical Part** Item created before DTC 12.0R3, ensure that all its **PhysicalPart LifeValue** Relationship Items have values in their **Current Value** properties. For successful promotion of an applied **Operational Event** Item to the **Complete** State, all the **Current Value** properties (cells) of each eligible **Physical Part** Item in the Physical Part BOM structure must have a value. If a given related **Life Parameter** has no initial value recorded, **0** (zero) should be set. The **Current Value** property has become required only since DTC 12.0R3.

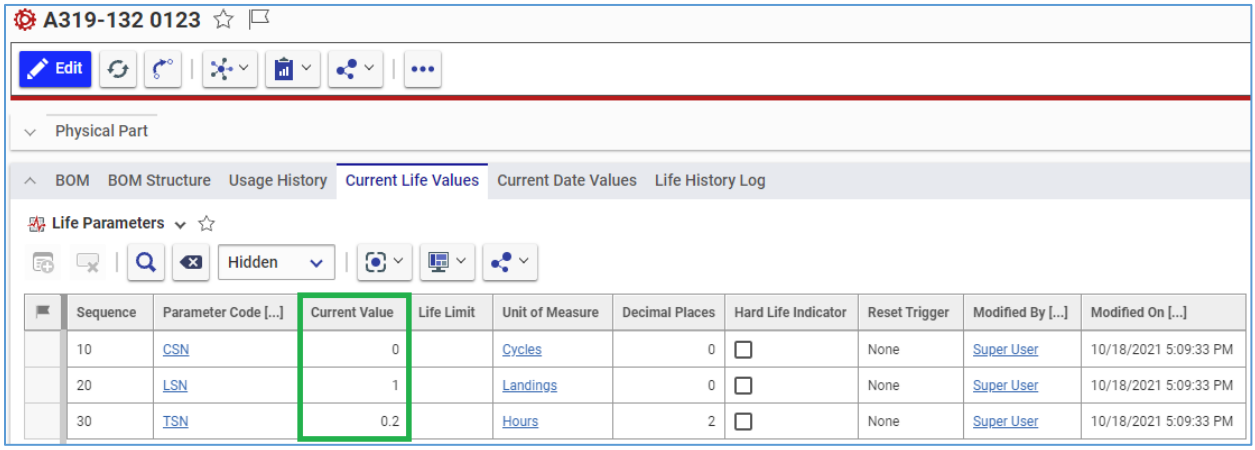


Figure 486.

One or more **Current Value** properties (cells) of one or more **Physical Part** Items in the Physical Part BOM structure can be empty in a legacy database.

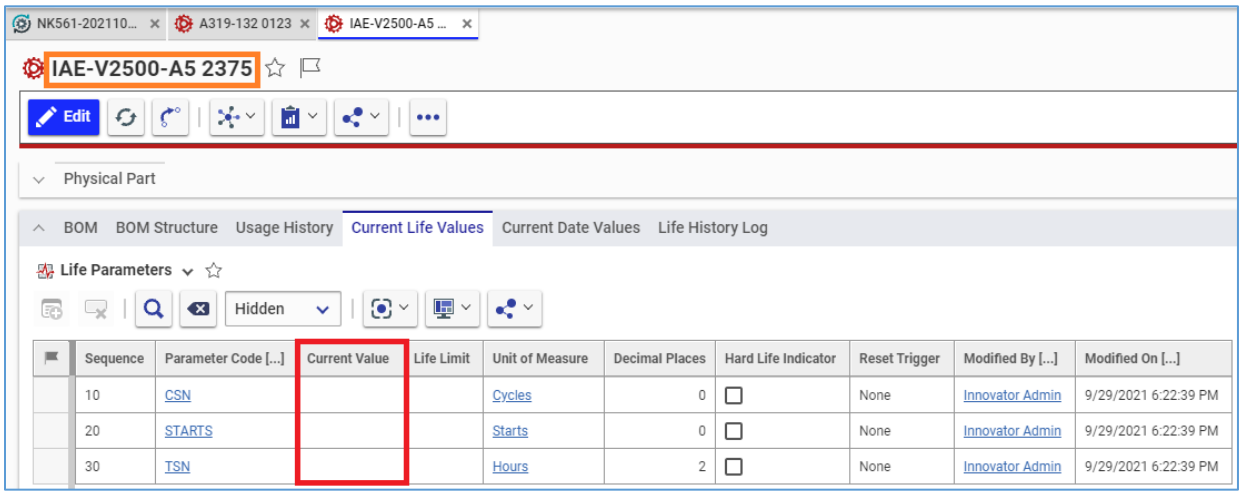


Figure 487.

In this case, an **Operational Event** Item that should be applied to such a **Physical Part** Item cannot be promoted to the **Complete** State, and an error message is raised.

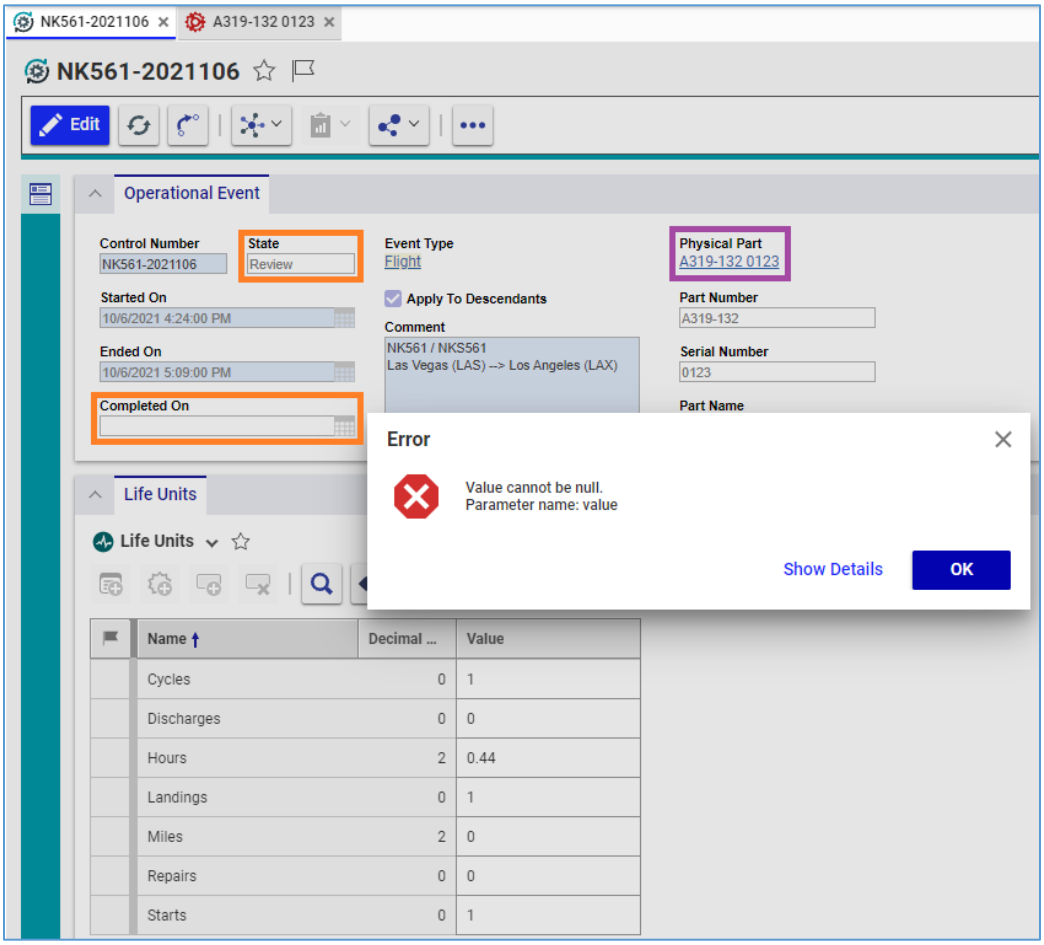


Figure 488.

To enable the promotion, all empty **Current Value** properties (cells) of the involved **PhysicalPart LifeValue** Relationship Items must get values either manually or automatically by resaving the Item.

If promotion to the **Complete** State fails, the system does not apply the **Operational Event** Item to given **Physical Part** Items. The **Current Value** properties (cells) of the involved **PhysicalPart LifeValue** Relationship Items remain unchanged.

If promotion to the **Complete** State succeeds, and the system applies the **Operational Event** Item to given **Physical Part** Items. The **Current Value** properties (cells) of the involved **PhysicalPart LifeValue** Relationship Items are updated. Also, the system populates the **Completed On** property with the promotion date and time. For details, see section 4.7.2.3 Operational Event update of life values.

If an **Operational Event** Item has been promoted to the **Complete** state with one or more wrong values in the **Value** properties of its **OperationalEvent LifeUnit** Relationship Items, there are two ways to correct the appropriate, affected **Current Value** property values of the involved **PhysicalPart LifeValue** Relationship Items:

1. To create and promote a new **Operational Event** Item with values that will produce correct **Current Value** property values.

Note: A correctional **Operational Event** Item should be of a specifically dedicated **Operational Event Type**, like **Admin** or **Correction**.

2. To correct wrong **Current Value** property values manually; see section 4.7.2.2 Manual update of life values.

11.6 Managing Operational Events

Your ability to manage an **Operational Event** Item depends on your Operational Event Identity membership and current Item State.

11.6.1 Creating Operational Events

Note: Only an **Operational Event Creator** can create the **Operational Event** Items.

To create a new **Operational Event** Item:

1. Go to **Contents** --> **Assets** --> **Operational Events**.
2. Click **Create Operational Event**. The **Operational Event #** Item view appears.

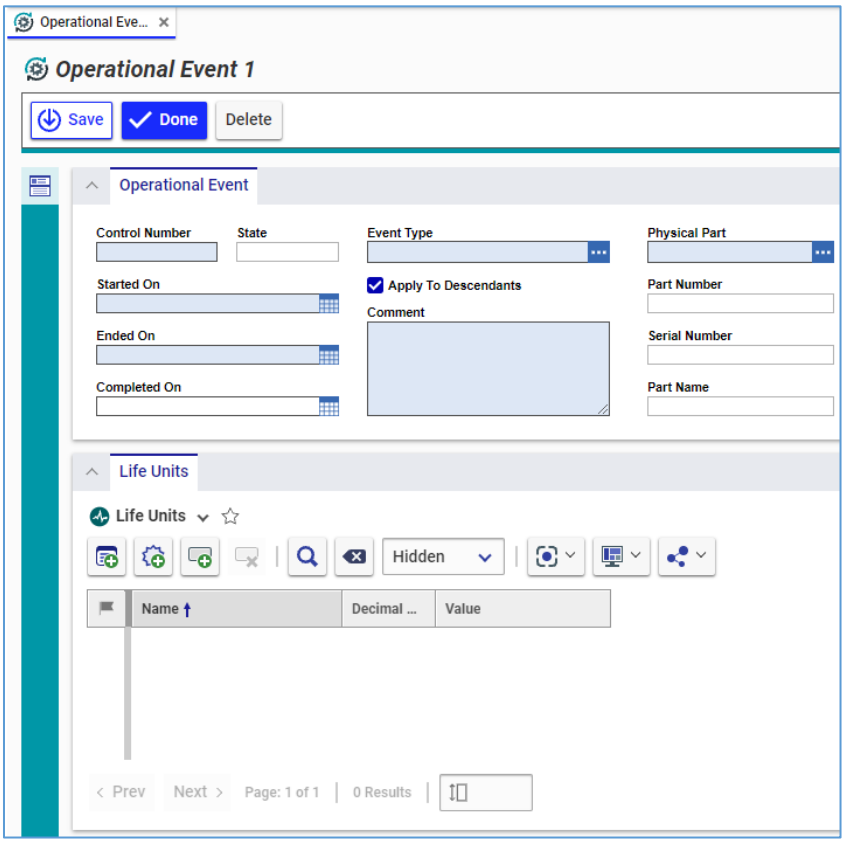


Figure 489.

3. In the **Control Number** field, enter a unique identifier of the **Operational Event** Item.

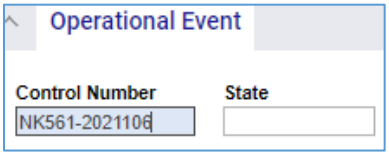


Figure 490.

Note: The white space characters are not allowed in the **Control Number** property values: the system clears the entered value from them upon saving the **Operational Event** Item.

4. In the **Started On** field, specify the **date** and **time** when the represented operational event has started in the field. See section 3.5 Specifying Date and Time values.

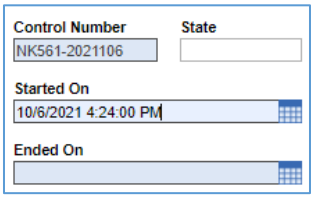


Figure 491.

5. In the **Ended On** field, specify the **date** and **time** when the represented event has ended in the field.

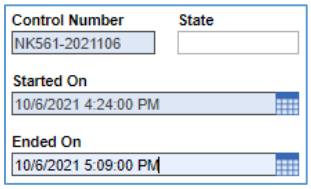


Figure 492.

6. Click the **ellipsis** button in the **Event Type** field. The **Select Items – Operational Event Types** dialog appears.

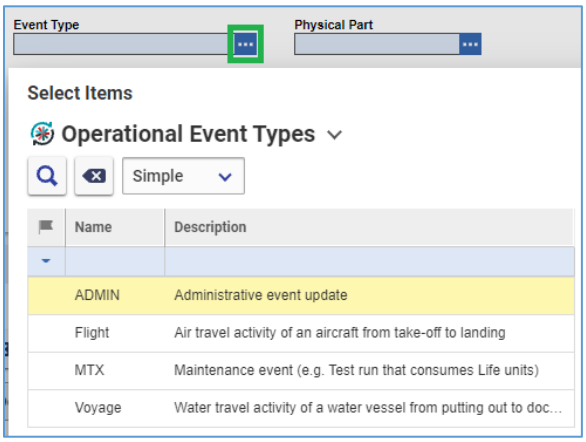


Figure 493.

7. Using the standard search procedure, search for an **Operational Event Type** Item which stands for a type of the represented real-world operational event.

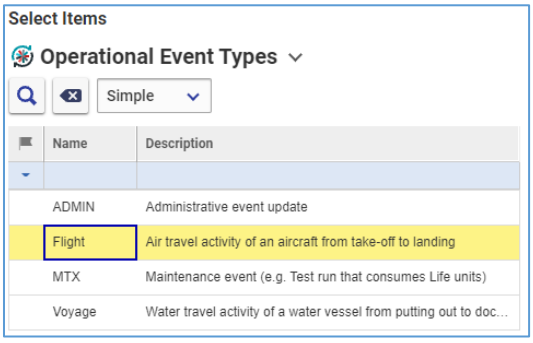


Figure 494.

8. Select this **Operational Event Type** Item and click **OK**. The selected **Operational Event Type** Item populates the **Event Type** field.

Figure 495.

Note: You can also specify the **Operational Event Type** Item by typing in the **Event Type** field.

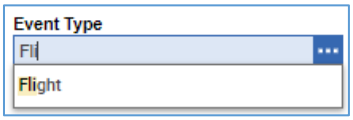


Figure 496.

9. Set the **Apply To Descendants** check box depending on whether this **Operational Event** Item should update or not the children of a **Physical Part** Item given in steps 11-13 by either:
   a. Keeping it selected if it should.
   b. Clearing it if it should not.

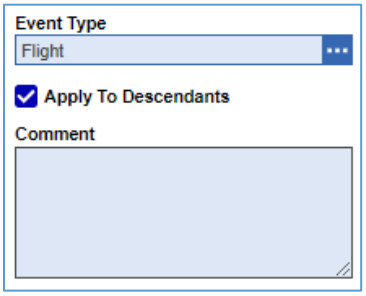


Figure 497.

10. In the **Comment** field, enter detailed or additional information on the represented real-world operational activity.

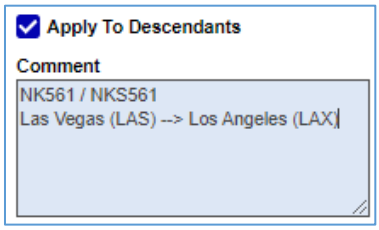


Figure 498.

11. Click the **ellipsis** button in the **Physical Part** field. The **Select Items – Physical Parts** dialog appears.

Figure 499.

12. Using the standard search procedure, search for a **Physical Part** Item representing a real-world asset that has performed the operational activity in the field.

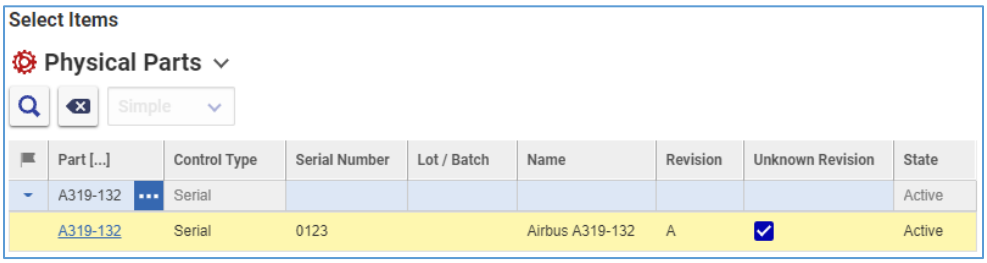


Figure 500.

13. Select this **Physical Part** Item and click **OK**. The selected **Operational Event Type** Item populates the **Physical Part** field.

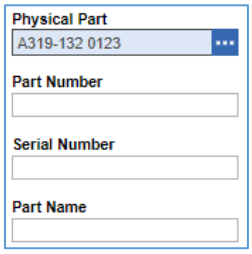


Figure 501.

Note: You can also specify the **Physical Part** Item by typing in the **Physical Part** field.

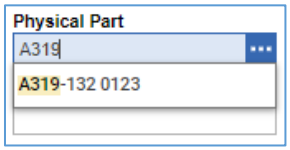


Figure 502.

14. Click **Save** on the **Operational Event** toolbar.

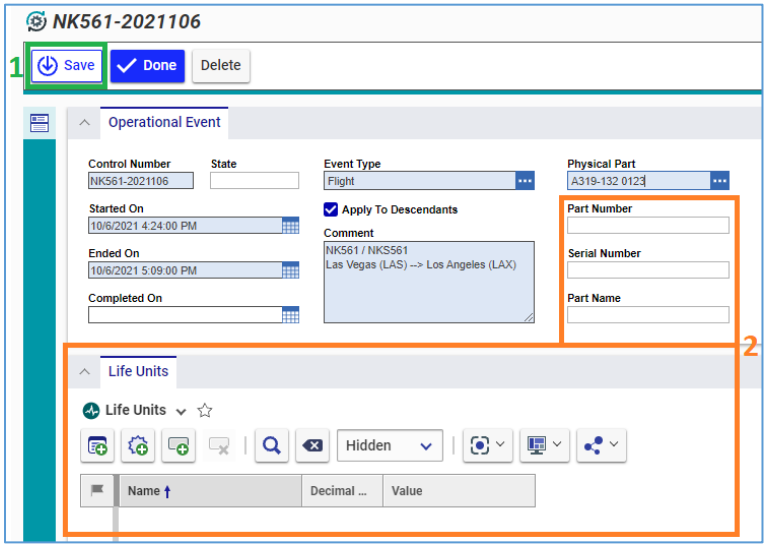


Figure 503.

The system automatically populates:

- The **Part Number**, **Serial Number**, and **Part Name** fields from the given **Physical Part** Item.

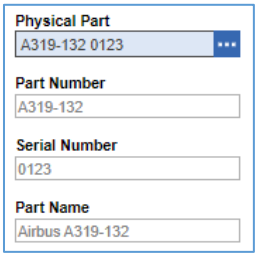


Figure 504.

- The **Life Units** Relationships grid with all existing numeric **Life Unit** Items.

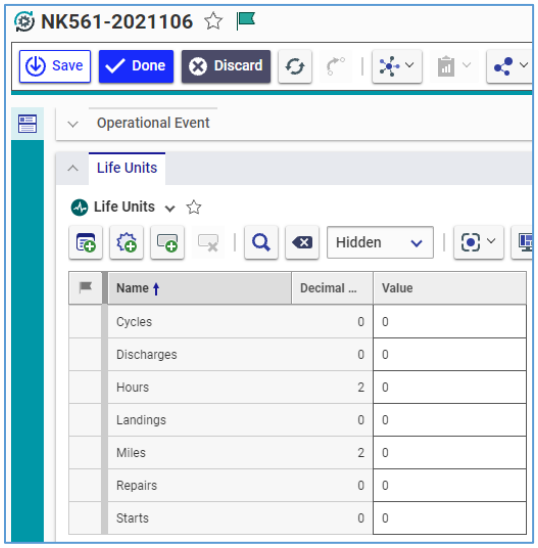


Figure 505.

15. Go to the **Life Units** Relationships grid.
16. Enter real, true numeric data in the **Value** cells only for those **Life Unit** Items which operational activity has consumed.

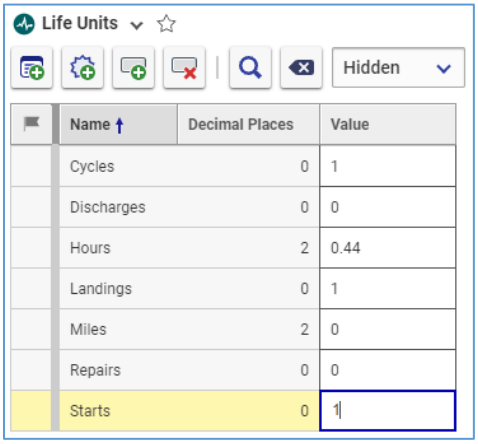


Figure 506.

17. Click either **Save** or **Done** on the **Operational Event** Item toolbar. The new **Operational Event** Item is created and in the **Preliminary** State.

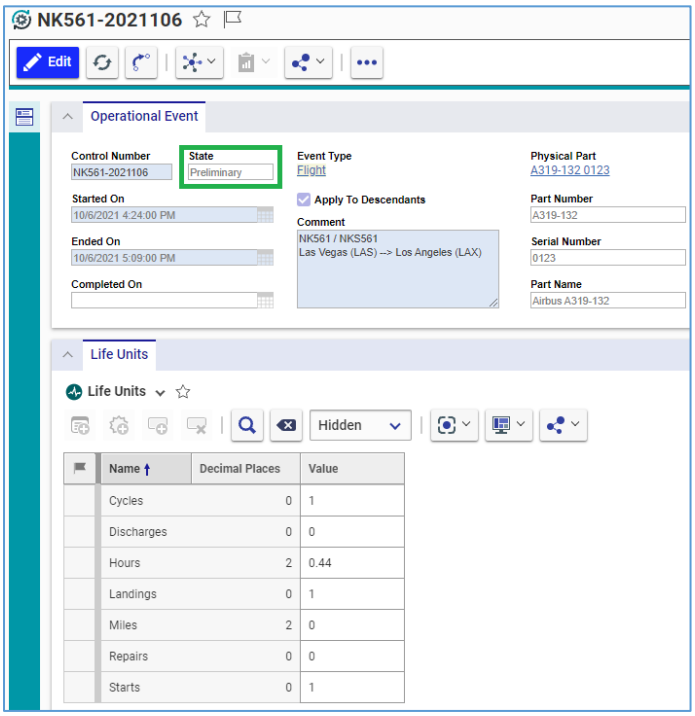


Figure 507.

The **Completed On** property remains empty.

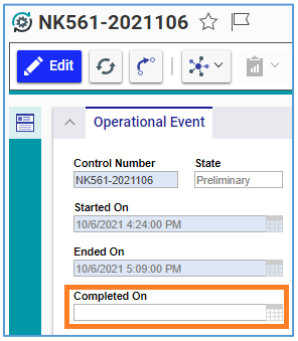


Figure 508.

The **Operational Event** Item **Innovator tab** and **Page Title** bar show the Item **Control Number** value.

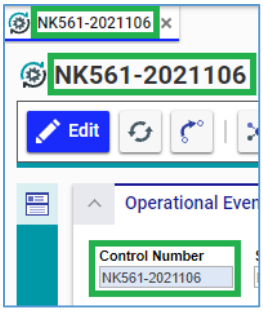


Figure 509.

The **Operational Event** Item is now available via search, subject to the Operational Event Permissions.

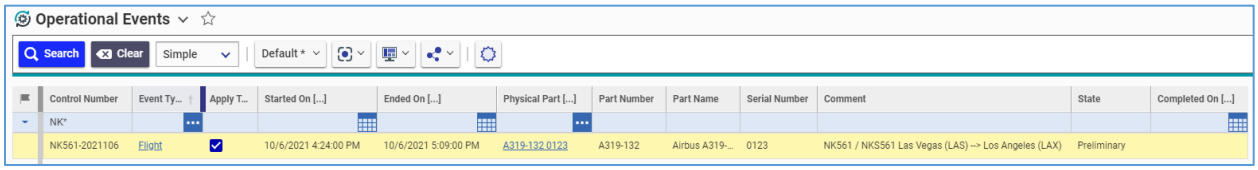


Figure 510.

Upon saving an **Operational Event** Item, the system validates this Item and raises an intelligent error if the validation fails, for example, if the Item has a blank field for a required property or a duplicating **Policy Code** value, discussed in section 11.6.4 Validation of Operational Events.

11.6.2 Modifying Operational Events

Note: A current State of an **Operational Event** Item determines who can update this Item. See this section for details. If having view-only rights, clicking the **Edit** button on the **Life Policy** Item toolbar raises an error.

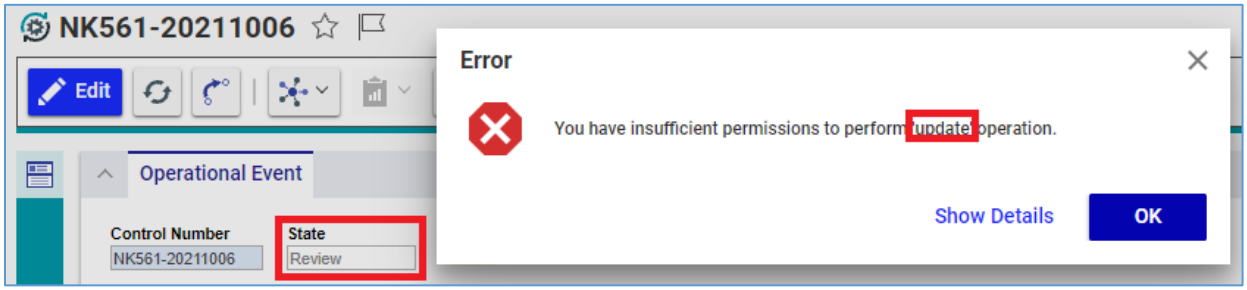


Figure 511.

Depending on a current State of an **Operational Event** Item, a Member of a given Identity can do with this Item the following:

- **Preliminary**:
  - **Operational Event Creator**:
    - Edit all its native properties, except **Completed On**.
    - Edit the **Value** properties of its **OperationalEvent LifeUnit** Relationship Items.
  - **Asset Viewer**:
    - View only.
- **Review**:
  - **Operational Event Reviewer**:
    - Edit all its native properties, except **Completed On**.
    - Edit the **Value** properties of its **OperationalEvent LifeUnit** Relationship Items.
  - **Operational Event Creator**:
    - View only.
  - **Asset Viewer**:
    - View only.

- **Complete**:
  - **Operational Event Admin**:
    - Edit its native properties, except **Physical Part** and **Completed On**.
  - **Operational Event Creator**:
    - View only.
  - **Operational Event Reviewer**:
    - View only.
  - **Asset Viewer**:
    - View only.

If the **Control Number** value has been changed, the **Operational Event** Item **Innovator tab** and **Page Title** bar show its new value.

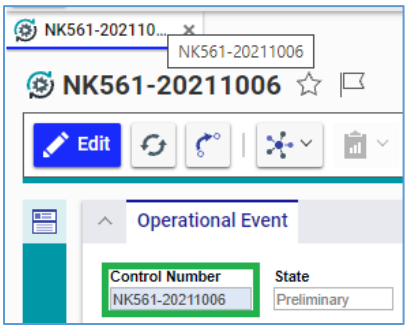


Figure 512.

Upon saving an **Operational Event** Item, the system validates this Item and raises an intelligent error if the validation fails, for example, if the Item has a blank field for a required property or a duplicating **Policy Code** value, discussed in section 11.6.4 Validation of Operational Events.

11.6.3 Deleting Operational Events

Note: A current State of an **Operational Event** Item determines who can delete this Item. See this section for details. If having no rights, an error is raised.

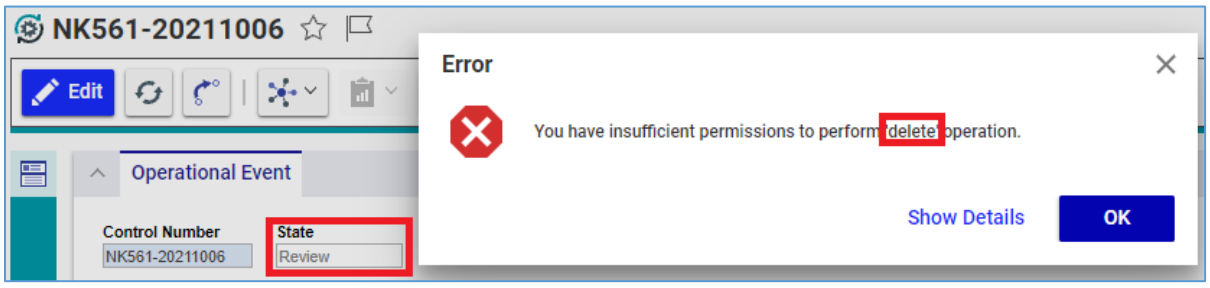


Figure 513.

To delete a **Life Parameter** Item, use the standard deletion procedure.

A current State of an **Operational Event** Item defines who can delete this Item:

- **Preliminary**:
  - **Operational Event Creator**

- **Review**:
  - **Operational Event Reviewer**
- **Complete**:
  - Nobody

11.6.4 Validation of Operational Events

Once you click **Save** or **Done** on an **Operational Event** Item toolbar, the system validates this Item and raises an appropriate error if the Item:

- Is missing a **Control Number** value.

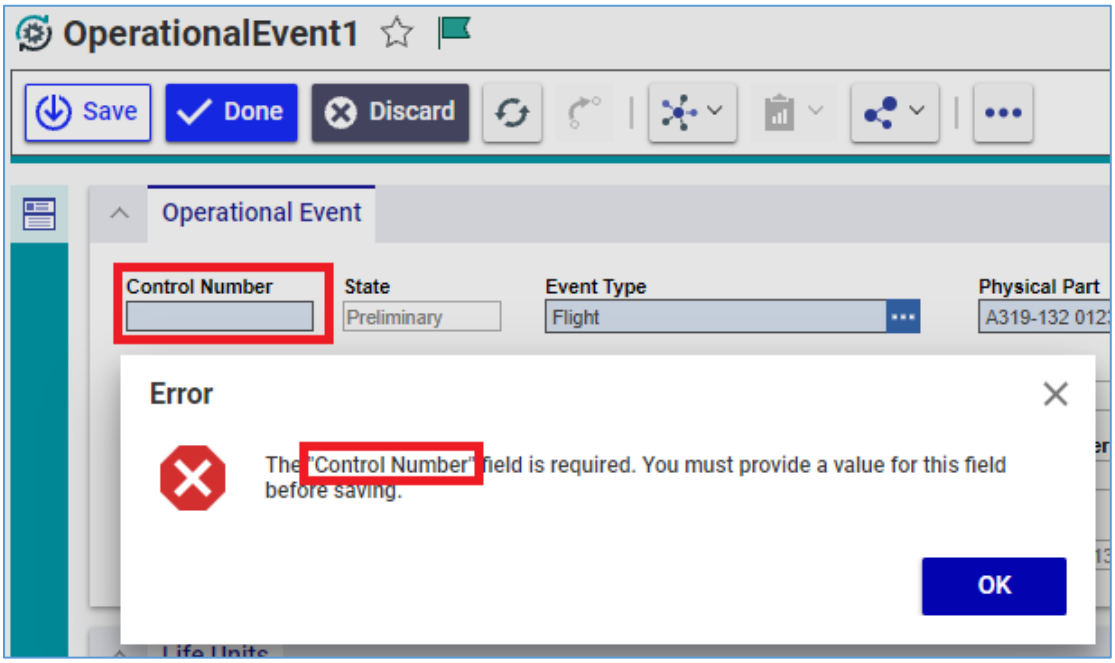


Figure 514.

- Is missing a **Started On** value.

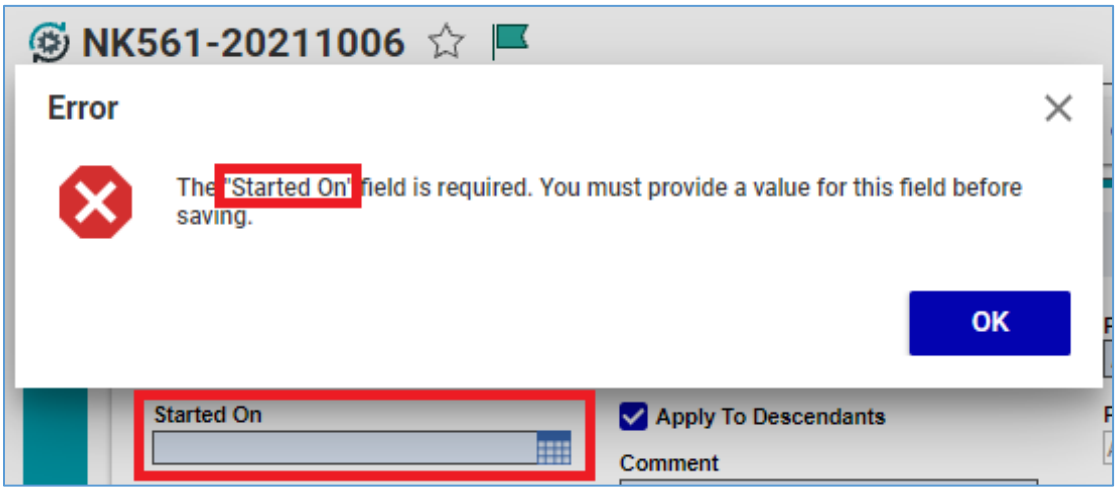


Figure 515.

- Is missing an **Ended On** value.

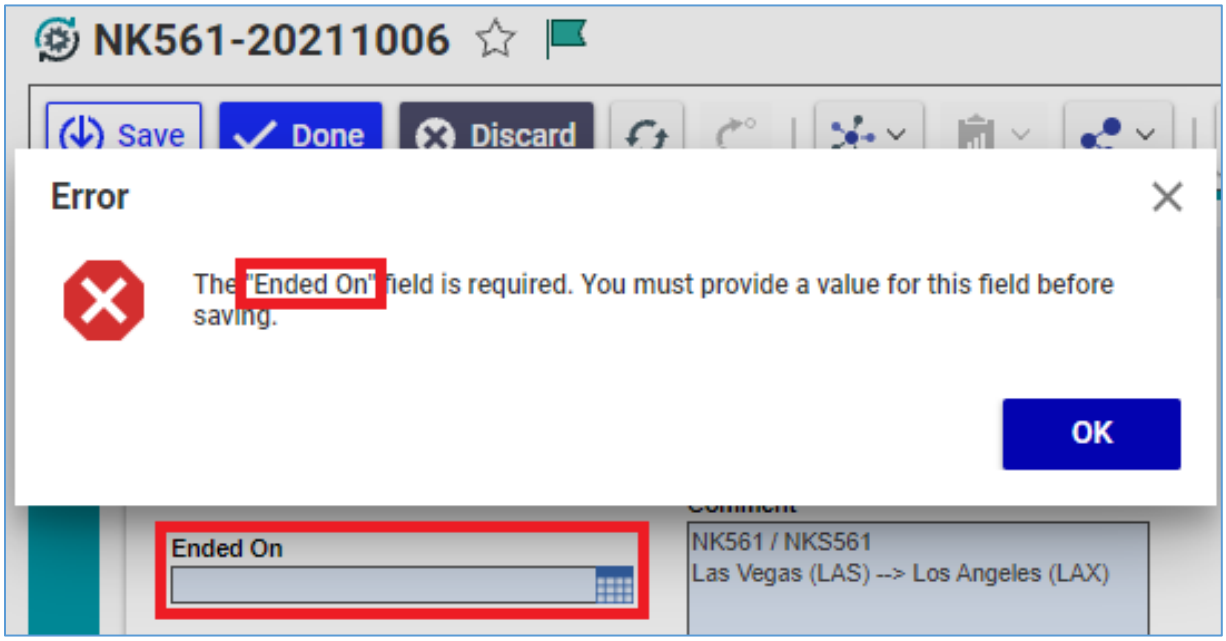


Figure 516.

- Is missing an **Event Type** value.

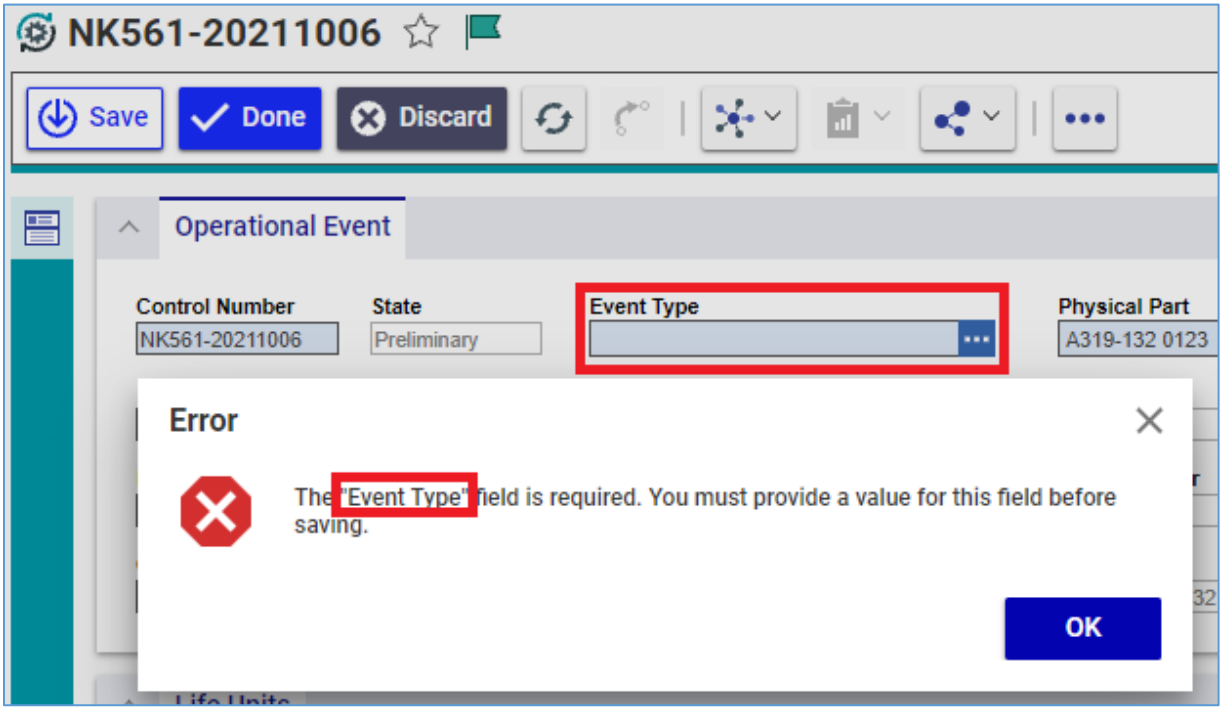


Figure 517.

- Is missing a **Comment** value.

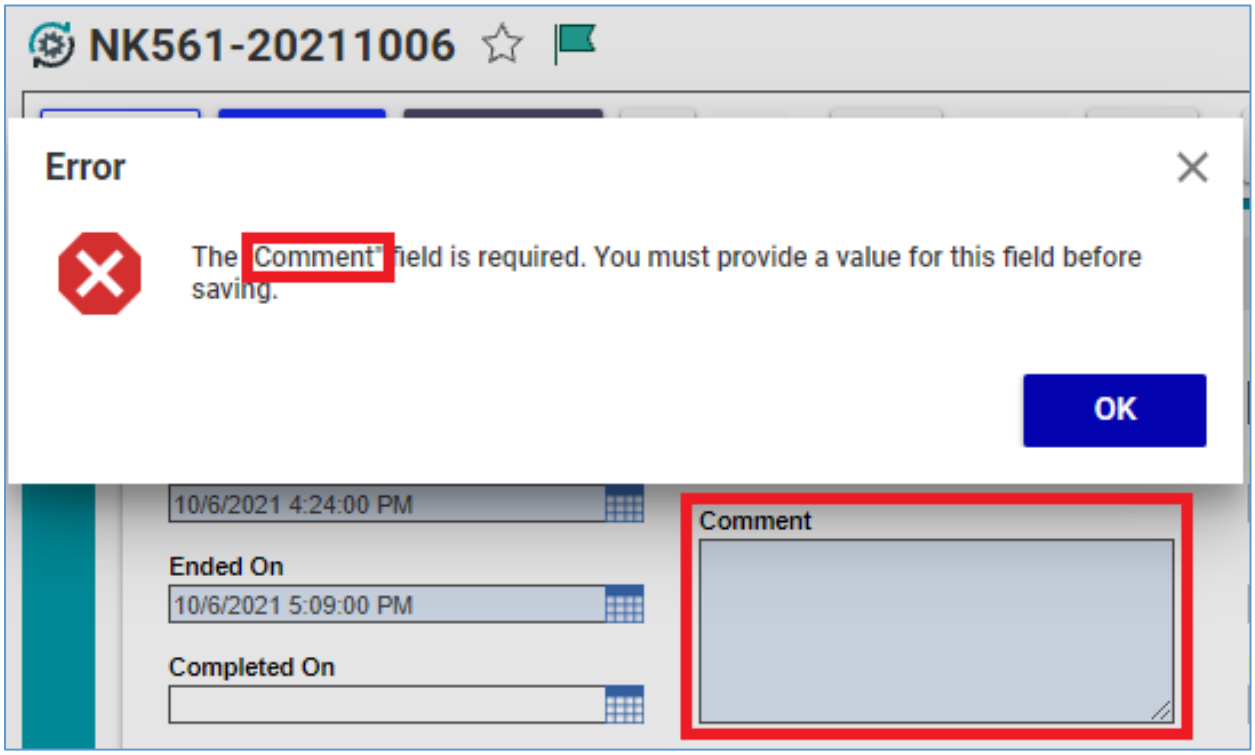
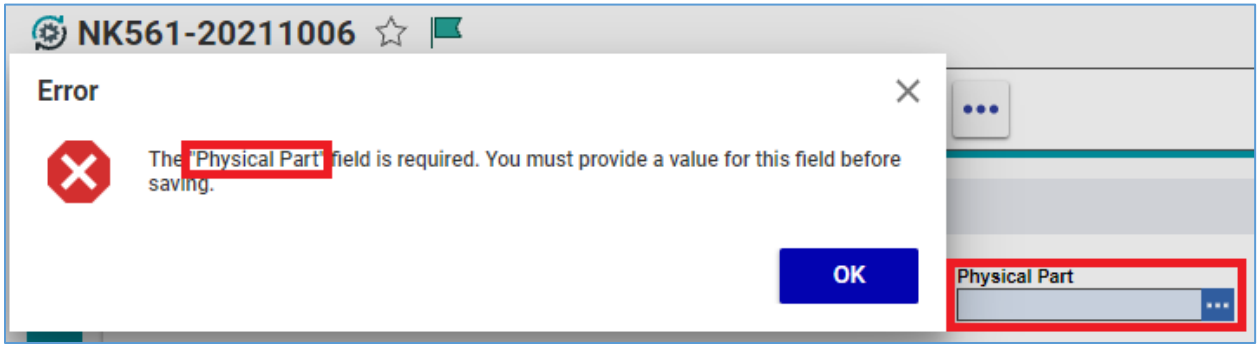


Figure 518.

- Is missing a **Physical Part** value.



- Has a **Control Number** value that is not unique: already given to an existing **Operational Event** Item.

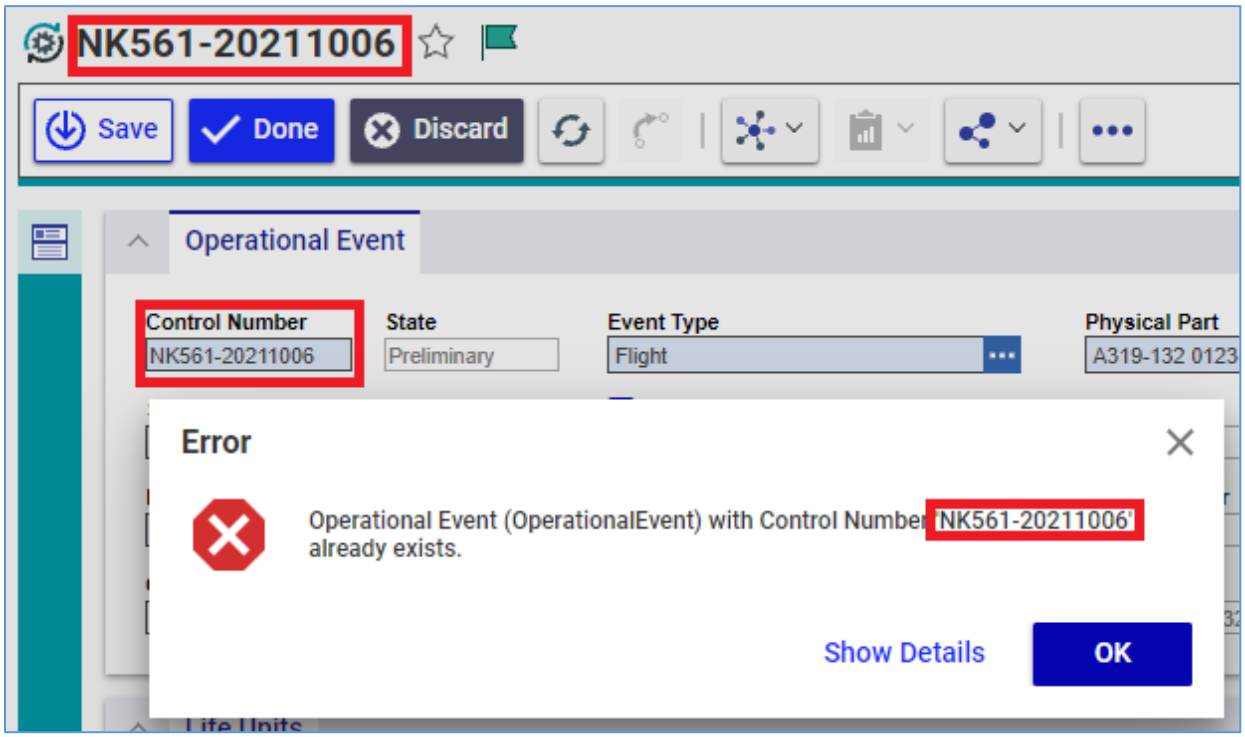


Figure 519.

- Has a **Started On** value later than an **Ended On** value.

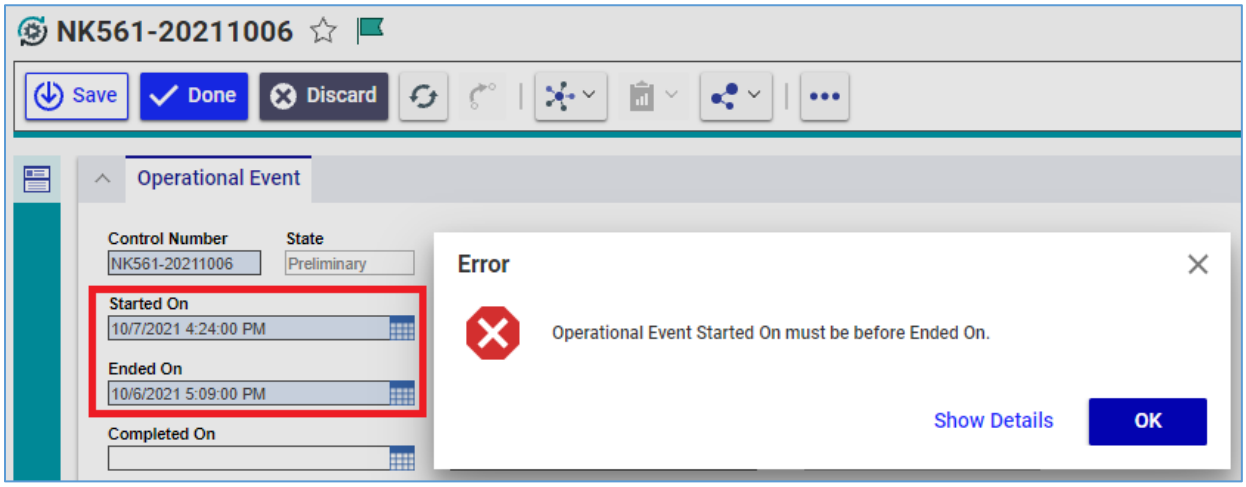


Figure 520.

- Has a **Physical Part** Item in the **Preliminary** State.

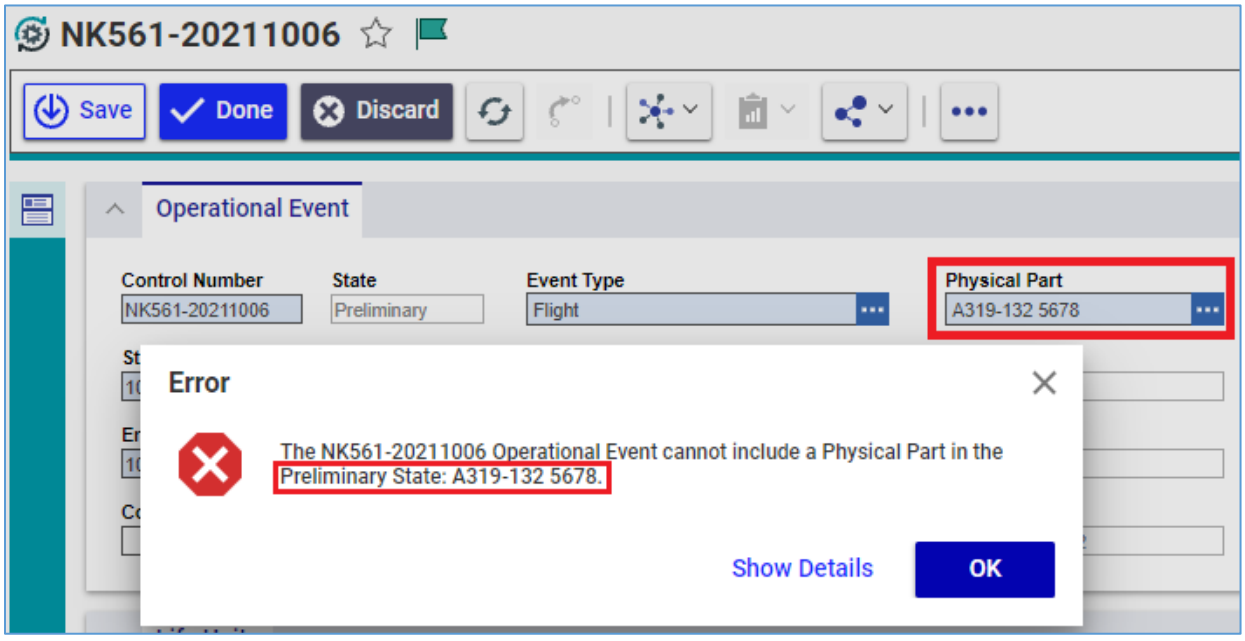


Figure 521.

- Has a **Physical Part** Item that is not serial-controlled.

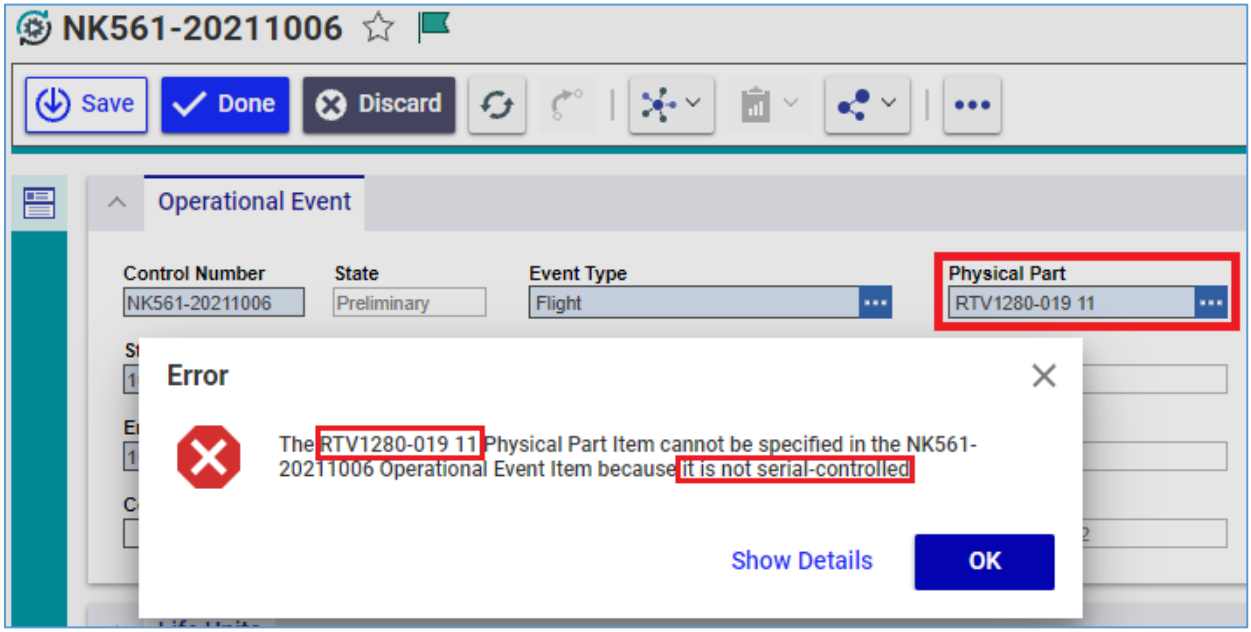


Figure 522.

- Has an **OperationalEvent LifeUnit** Relationship Item whose **Value** property is empty.

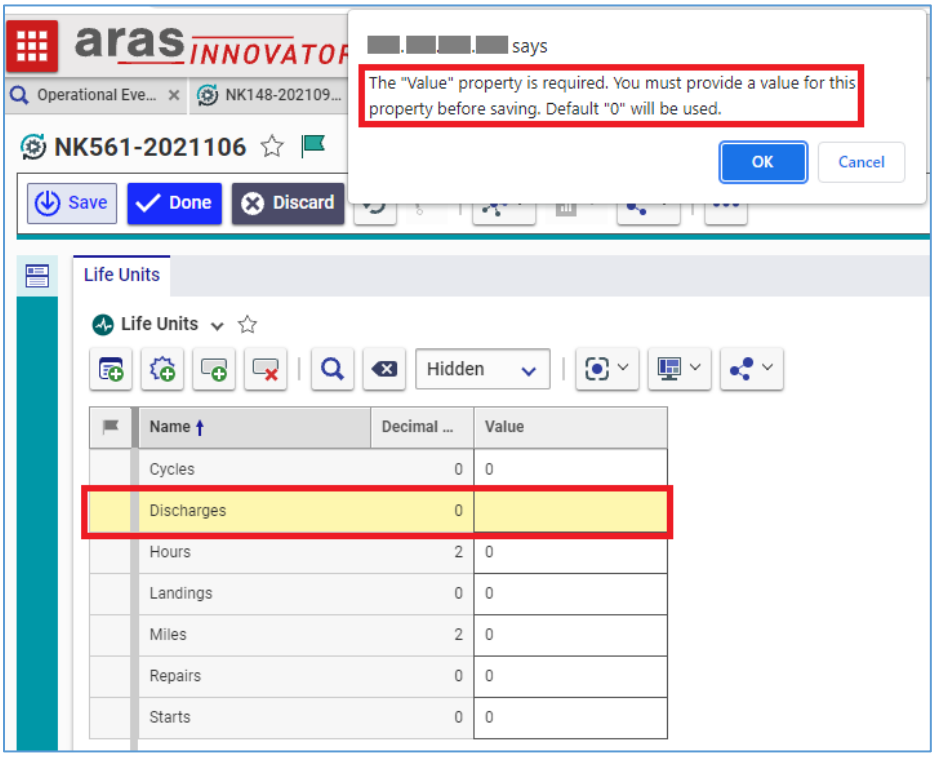


Figure 523.

- Has an **OperationalEvent LifeUnit** Relationship Item whose **Value** property has a number with more digits after a decimal point than the given **Decimal Places** property value allows.

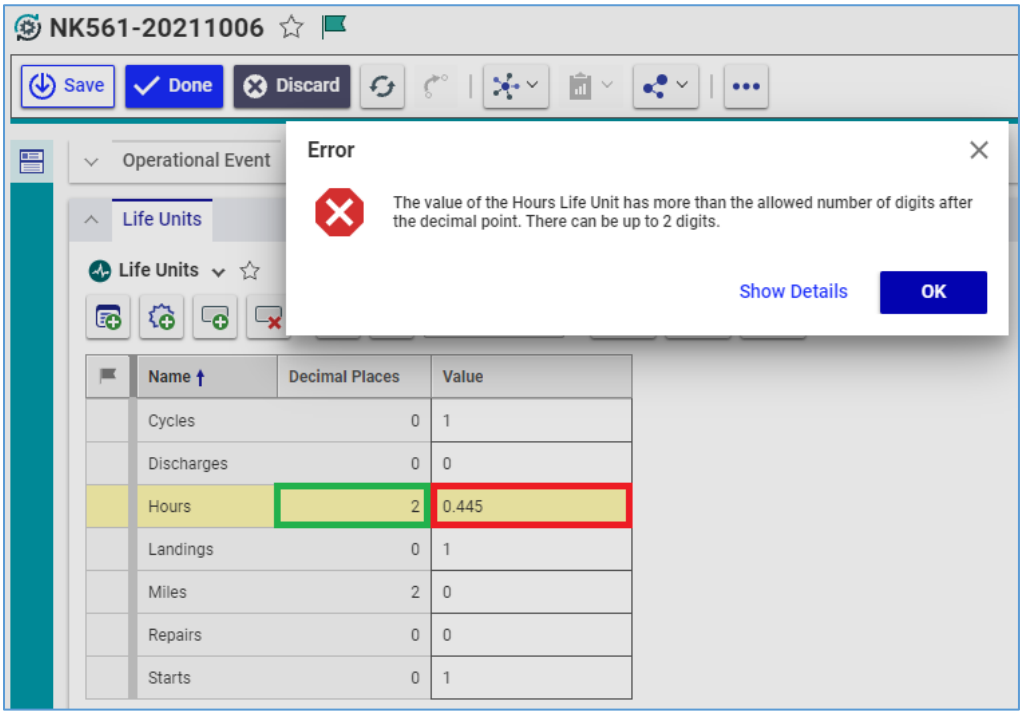


Figure 524.

12 The Structure Browser and DTC Items

To compare side-by-side simple single-level structures or complex multi-level hierarchies of Items, Aras Innovator has a standard feature: Structure Browser. You can use it for DTC application Items that have Relationship Grids, like the **Physical Part** Items, **Part Policy** Items, and so on.

There are several ways to access the Structure Browser:

- Go the ***Item Type*** Search Grid Page, right-click an ***Item*** to be compared, and then click **Navigate** --> **Structure Browser**.

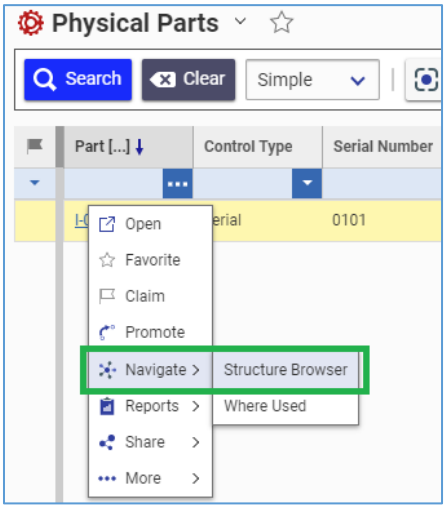


Figure 525.

- Open the view of an ***Item*** to be compared and then click **Navigate** --> **Structure Browser** on the ***Item*** toolbar.

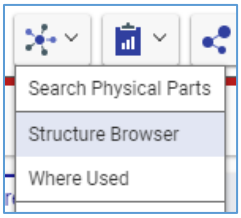


Figure 526.

Later in this section, it is assumed that you have accessed the Structure Browser.

When you access the Structure Browser, it appears as a new Aras Innovator tab view that shows the hierarchy (structure) tree of a given ***Item***.

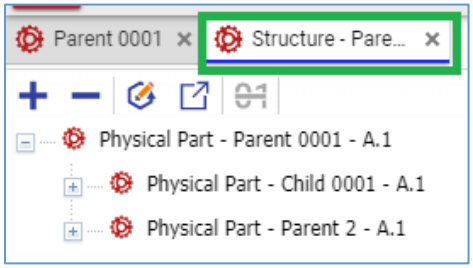


Figure 527.

In addition to the standard tree navigation features, there are the following buttons in the **Structure Browser** toolbar:

- **Expand All** expands the whole hierarchy tree.

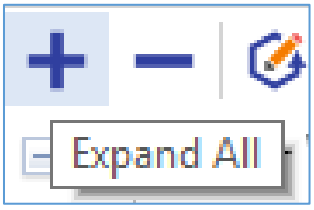


Figure 528.

- **Collapse All** collapses the whole hierarchy tree.

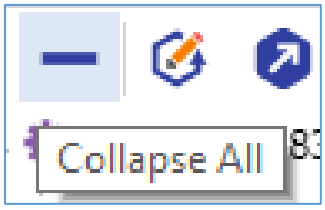


Figure 529.

To quit the Structure Browser, close its view.

Use the following procedure to compare the structures or hierarchies of two different ***Items***:

1. Click the **Side-by-Side with other *Item Type*** button on the **Structure Browser** toolbar. The **Select Items – *Item Type*** dialog pops up.

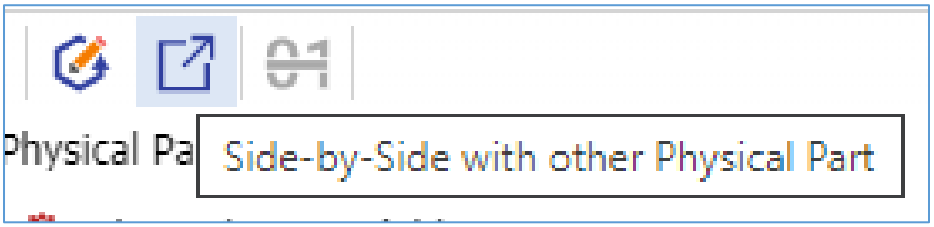


Figure 530.

2. Using the standard search procedure, search for and select another ***Item*** to be compared. The tree of the selected ***Item*** structure appears on the right side of the Structure Browser, and the **Highlight** button becomes available on the **Structure Browser** toolbar.

Figure 531.

3. Click **Highlight** in the **Structure Browser** toolbar. The **trees** are expanded with highlighted differences.

Figure 532.

4. View the comparison of the ***Item*** structure, where each Item is color-highlighted:
   - Black means that both structures use the same ***Item***.
   - Blue means that this ***Item*** does not exist in the opposite structure.

If it is necessary to view detailed information on an ***Item***, double-click it to open its view.

Repeat this procedure if you need to compare the given ***Item*** with another one.

Note: You cannot compare more than two ***Item*** structures in one Structure Browser tab.

The below figure shows an example of comparing the **Life Parameter** Items included in different **Life Policy** Items.

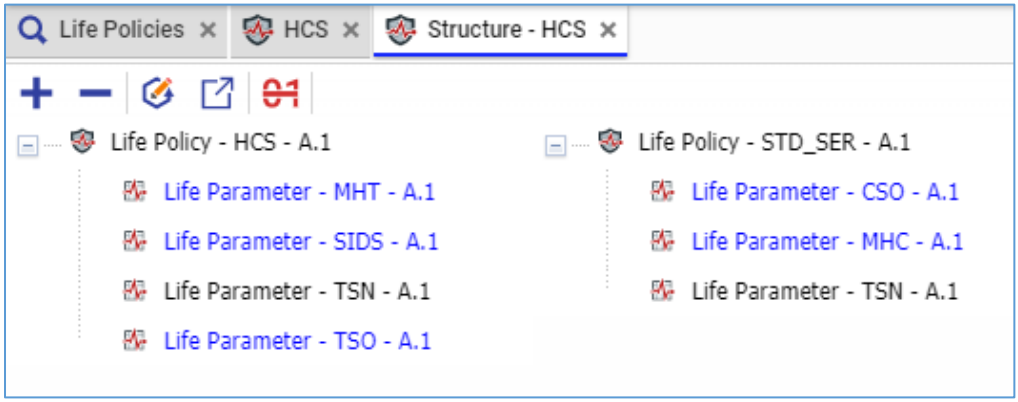


Figure 533.

13 The Where Used Browser and DTC Items

The **Where Used** browser is an out-of-the-box tool for many Aras Innovator Items that provides general information on Item usage. You can use it for DTC application Items that are used by other Items: the **Life Unit** Items, **Life Parameter** Items, and so on.

If you need usage information specific for the **Physical Part** Items, use the **Usage History** feature dedicated for this ItemType as discussed in section *5.4 Physical Part Usage History*.

The **Where Used** browser shows a bottom-up hierarchy tree, where an explored ***Item*** is on the top, and its usage is beneath. Each branch represents a separate ***Item*** that uses the given ***Item***. The branch can be shown to the top ***Item*** user.

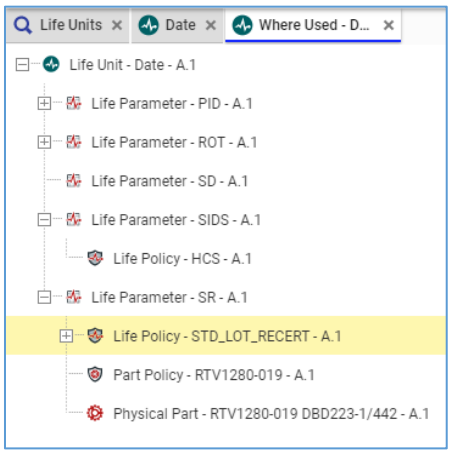


Figure 534.

To access the **Where Used** browser for an ***Item***, either:

- Go to the ***Item Type*** Search Grid page, right-click the **Item**, and then click **Navigate** --> **Where Used**.

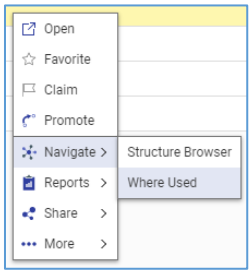


Figure 535.

- Open the ***Item*** view, and on its toolbar, click **Navigate** --> **Where Used**.

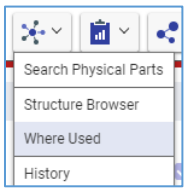


Figure 536.

When you access the **Where Used** browser, it appears as a new Aras Innovator tab view.

To quit the **Where Used** Browser, close its tab view in Aras Innovator.

14 Migrating from earlier DTC versions to 12.0R3

14.1 Migrating from DTC 12.0R1 to 12.0R3

Between the DTC application releases, the Control Type definition for the **Physical Part** Items differs:

- In 12.0R1, it is on the **Part** Items. Many **Physical Part** Items with the same **Part Number** value can have different **Control Type** values because many Revisions with different **Control Type** values can exist for one **Part** Item.
- Since 12.0R2, it is on the **Part Policy** Items. All **Physical Part** Items with the same **Part Number** value always have the same **Control Type** value because only one **Part Policy** Item can exist for all Revisions of one **Part** Item.

To avoid data corruption and erroneous application behavior in 12.0R3, the following actions are required before using a **Physical Part** Item imported from 12.0R1:

1. Create and promote to the **Active** State, a **Part Policy** Item for the **Part** Item sourcing the given **Physical Part** Item. Attach **Life Policy** if necessary. This **Part Policy** Item must have its **Control Type** value be the same as this **Part** Item has in 12.0R1. Otherwise, an error is raised if you are trying to create, use, or edit a **Physical Part** Item rooted in the given **Part** Item.

Warning If one **Part** Item has many Revisions with different **Control Type** values, and **Physical Part** Items exist for these Revisions in DTC 12.0R1, only **Physical Part** Items of one **Control Type** value can be used in DTC 12.0R3 among such **Physical Part** Items. The **Physical Part** Items that are rooted in this **Part** Item and have other **Control Type** values should be deleted.

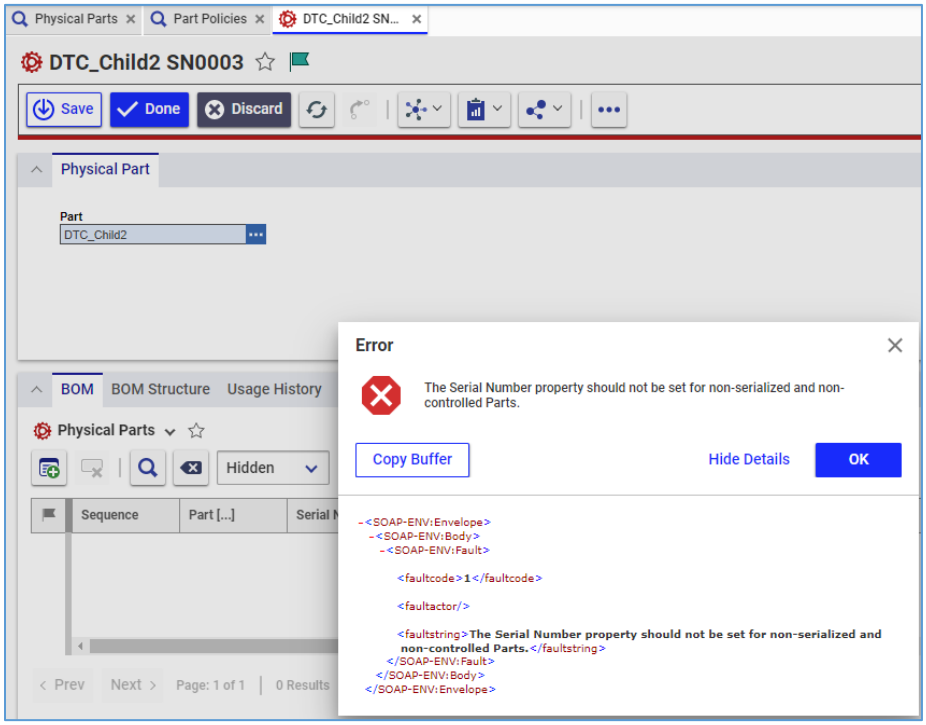


Figure 537.

2. Update the **Physical Part** Item as follows:
   a. Open the **Physical Part** Item form.
   b. Click **Edit** --> **Done** on the **Physical Part** Item toolbar.
   c. Close the **Physical Part** form.

Note: If a **Physical Part** Item is not updated, its Item form shows only the **Part** property.

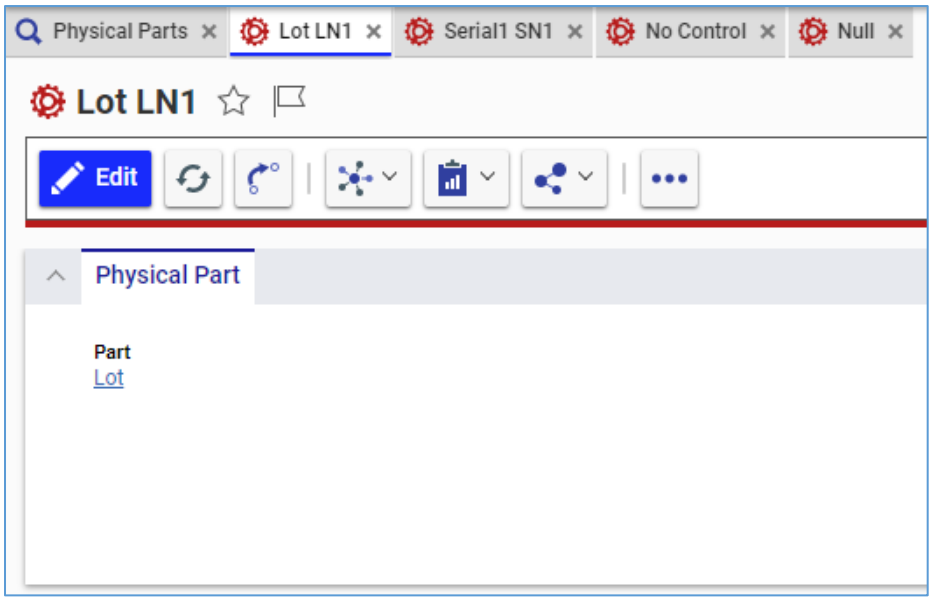


Figure 538.

The **Physical Part** Item imported from DTC 12.0R1 is ready for use in 12.0R3. All its properties are visible. No unexpected errors are raised.

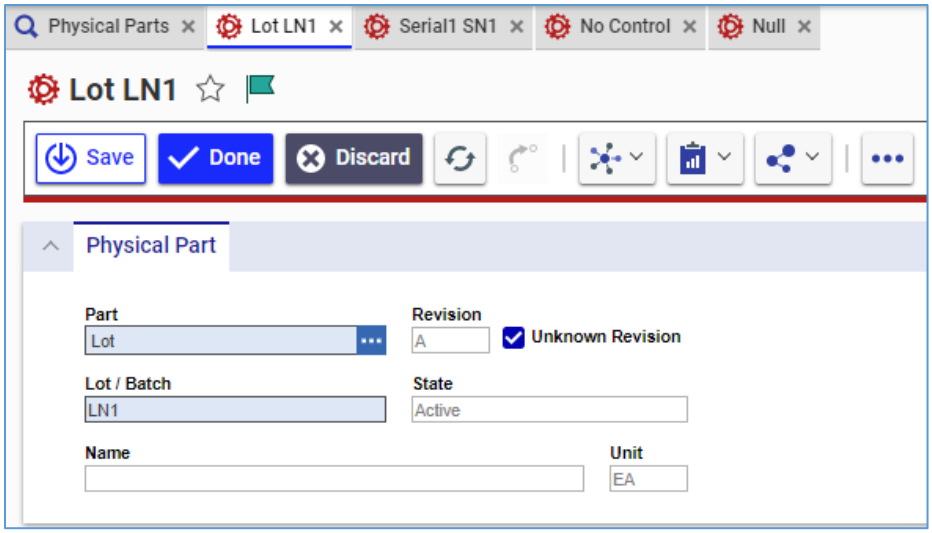


Figure 539.

14.2 Migrating from DTC 12.0R2 to 12.0R3

Since 12.0R2, the DTC application features manual tracking of the current values of asset life variables. The 12.0R3 release added the update of life variables automatically by operational events of an asset. To enable such automatic update, this release changed the existing tracking functionality as follows:

- Imposed new Control Type constraints for a **Life Parameter** Item depending on its **Unit of Measure**; see section 7.1 Understanding Life Parameters.
- Made the **Current Value** property of the **PhysicalPart LifeValue** Relationship Items required; see section 4.7.2.1 Current Life Values Relationships Grid.

This section discusses what you should do if you want to use in DTC 12.0R3 a database created in DTC 12.0R2. If that legacy database has no life-tracking Items (**Life Unit**, **Life Parameter**, **Life Policy**), no actions are required. If there are such Items, you should follow the instructions given in this section.

14.2.1 New constraints for Life Parameters

Since 12.0R3, a **Life Parameter** Item that includes a non-serialized Control Type cannot have a **Unit of Measure** that is a numeric **Life Unit** Item.

As 12.0R2 has no such constraints, you should update the **Allowed Control Types** properties of all numeric **Life Parameter** Items created in that release to include only the **Serial** Control Type before migrating these Items to 12.0R3.

Besides, you should update accordingly the non-serialized **Life Policy**, **Part Policy**, and **Physical Part** Items that are created in 12.0R2 and include the pertinent **Life Parameter** Items. When lot-controlled or non-controlled, such Items must not include numeric **Life Unit** and **Life Parameter** Items.

The application will not behave properly if the abovementioned updates are not carried out before the migration.

The necessary updates in question require special Aras Innovator and database privileges because the application server side does not allow updating Items used in other Items and some Items in the **Active** State.

14.2.2 Values are required for current numeric life variables

Since 12.0R3, the **Current Value** property of the **PhysicalPart LifeValue** Relationship Items is required and has the default value of **0**.

The **PhysicalPart LifeValue** Relationship Items created in 12.0R2 may have empty **Current Value** properties because the **Current Value** property is not required in that release version.

If migrating data with empty **Current Value** properties, the system will automatically populate them with the default value of **0** during the upgrade process, without raising any warnings or errors. The populated **Current Value** properties will carry forward their original values, unchanged.