



a division of ciena

Blue Planet Inventory Metadata Modeller User Guide

Table of Contents

| | |
|---|----|
| Overview | 4 |
| Create a New Library | 5 |
| Library Details | 5 |
| Library History | 7 |
| Import Library | 8 |
| Create Equipment | 9 |
| Action Buttons | 13 |
| Modifying an Equipment Model | 14 |
| Create and edit metadata | 15 |
| Creating metadata | 15 |
| Accessing the Create Metadata Page | 15 |
| Metadata Summary | 16 |
| Type | 16 |
| Discovery Attributes | 16 |
| Discovery Position Matching | 17 |
| Discovery Ignore Object | 17 |
| Discovery Convert Object | 18 |
| Discovery Connection Mapping | 18 |
| Discovery Interface Mapping | 19 |
| Callout Properties | 19 |
| Additional Properties | 20 |
| Metadata Relationships | 21 |
| Creating a new metadata and building its relationship | 21 |
| Viewing a summary of metadata | 22 |
| Copying existing metadata | 23 |
| Deleting existing metadata | 24 |
| Viewing the metadata report | 25 |
| Position Compatibility Between Equipment Types | 28 |
| Family | 29 |
| Create or Select a Master Group | 29 |
| Select Parent and Child Compatible Equipment | 30 |
| Action Buttons | 30 |
| Define Position Groups | 30 |
| Action Buttons | 32 |
| Define Position Group Combinations | 33 |
| Direct Compatibility Between Archetypes | 34 |
| Accessing the Direct Compatibility Page | 34 |
| Adding the Compatibility Between Archetypes | 35 |
| Using the Compatible Archetype in the Blue Planet Inventory | 37 |
| Metadata Modeller Workflow | 39 |
| Library Fragments | 51 |
| Working with library fragments | 52 |
| Extracting fragments of a library | 53 |

| | |
|--|------------|
| Adding a fragment to another library..... | 57 |
| Metamodel Management | 58 |
| Accessing the Metamodel Management Page | 58 |
| Create Metamodel..... | 58 |
| MetaModel Relationships..... | 59 |
| Adding a new metamodel..... | 60 |
| Removing a metamodel | 61 |
| Modifying a metamodel | 62 |
| Adding a new metamodel relationship | 62 |
| Removing a metamodel relationship..... | 63 |
| Deploying the newly added metamodel and its relationship | 63 |
| Using the newly added metamodel | 66 |
| Metadata Permission | 68 |
| Accessing the Metadata Permission Page..... | 68 |
| Creating permissions for metadata..... | 68 |
| Removing permission for metadata | 70 |
| Using the created equipment | 70 |
| Equipment Diagram Editor | 72 |
| Accessing the Equipment Diagram Editor | 72 |
| Structure of the Equipment Diagram Editor | 73 |
| Options | 74 |
| Scenario | 75 |
| Preview | 75 |
| Metadata..... | 76 |
| Structure of the Tabbed Panes..... | 77 |
| Diagram Concepts | 78 |
| Layout | 79 |
| Layout Expressions | 82 |
| Expression Data Types | 83 |
| Layout Case Study | 84 |
| Raise By..... | 87 |
| "Raise By" Case Study | 87 |
| Palette | 89 |
| SVG Definitions | 90 |
| CSS Stylesheets..... | 91 |
| Summary | 92 |
| Creating an Equipment Diagram of a Device | 93 |
| Using the Equipment Diagram | 94 |
| Copying existing equipment..... | 97 |
| Library Processes | 99 |
| Creating a Project Library | 99 |
| Library Delta Tool..... | 106 |
| Library Lights Out Tool | 106 |
| LEGAL NOTICES..... | 108 |

Overview

The *Metadata Modeller* assists in creating, modifying, and deleting metadata for equipment. It also allows you to define the compatibility for created metadata. Using the Metadata Modeller, you can export the selected library, import it to a different environment and also deploy changes done via that library up to target activity into live. To access the Metadata Modeller:

1. Go to the Main Menu.
2. Select **Metadata Modeller**.

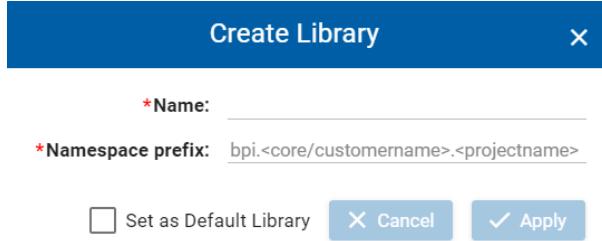
There are various options available in the drop-down, including:

- [Create Library](#)
- [Import Library](#)
- [Create Equipment](#)
- [Create Metadata](#)
- [Position Compatibility](#)
- [Metamodel Management](#)
- [Metadata Permission](#)

Create a New Library

Creating a new library in Blue Planet Inventory enables you to manage metadata centrally. To create a new library:

1. Go to the Main Menu.
2. Select **Metadata Modeller > Create Library**.



3. Fill out the **Name** and **Namespace prefix** in the **Create Library** dialog box.

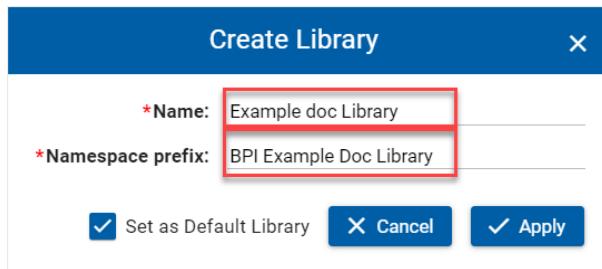


Name and **Namespace prefix** must be unique.

If you want to set the newly created library as the default library, check the **Set as Default Library** checkbox. An example to fill the required fields is displayed in the figure below.



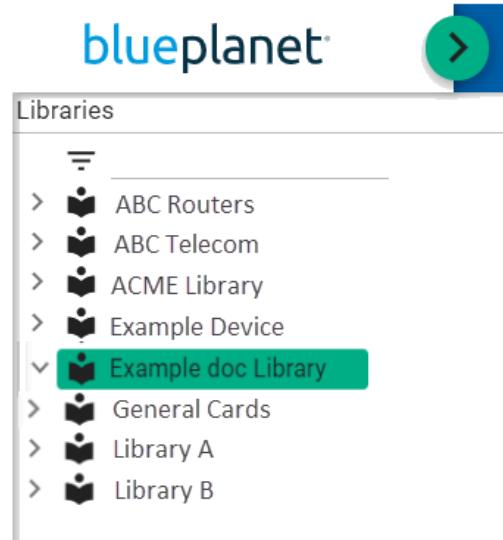
A red asterisk (*) indicates a required field.



4. Click **Apply**.

Library Details

After creating the library, the library moves to the Libraries tree group.



To see the details of the created library:

1. In the Libraries tree, right-click the library.
2. Select **Library Details**.

The Library Details page for the selected library opens.

The screenshot shows the 'Showcase' interface with the title 'Library Details - Packaging Metadata_Library'. The main area displays the following details:

- Name:** Packaging Metadata_Library
- Created By:** Normal
- Namespace prefix:** bpi.test.metadata.example
- Created Date:** Nov 10, 2019
- Perspective:** 900
- Revision Number:** 0

Below these details are buttons for 'Set as Default Library', 'Cancel', and 'Apply'. The interface also includes sections for 'Associated activities' and 'Dependencies'.

| Activity Name | Created Date | Revision Number | Tag Name |
|---|--------------|-----------------|--------------------|
| Activity for sample Library to package Metadata | Nov 10, 2019 | 1573389038392 | Latest(Not Tagged) |

| Library | Tag | CreatedDate | Revision |
|------------------|-----|-------------|----------|
| No Records Found | | | |

For more information about Associated activities section, see [Deploying created equipment](#).

For more information about Dependencies section, see [Adding Library Dependency](#).

Library History

To see the entire history of the library:

1. In the Libraries tree, right-click the library you want to see the history.
2. Select **History**.

The History page for the selected library opens.

| Object | Activity | Who By | Date | Attribute Change | From | To | Relationship Change | Related Object | Related Object Type | Order | Current Order Status | Perspective |
|--------|----------|--------|-------------------------|------------------|------|-----------------------------------|---------------------|----------------|---------------------|-------|----------------------|-------------|
| Test | Test | Elena | Feb 1, 2021 16:03:52 | Acronym | test | bpitest.111.2558 5706849421857 | | | | | | |
| | | | | BPID | | test | | | | | | |
| | | | | Description | | test | | | | | | |
| | | | | Height in Inches | | 1 | | | | | | |
| | | | | Model Number | | [test] | | | | | | |
| | | | | Name | | Test | | | | | | |
| | | | | Position Used | | 1 | | | | | | |
| | | | | Vendor | | test | | | | | | |
| | | | | Width in Inches | | 1 | | | | | | |

Import Library

The Import Library feature allows you to import a library that was exported on another environment (Blue Planet Inventory provides some standard libraries to import). These standard libraries can be used as a central source (for example, you can import a library provided by Blue Planet that your project can use as a central resource).

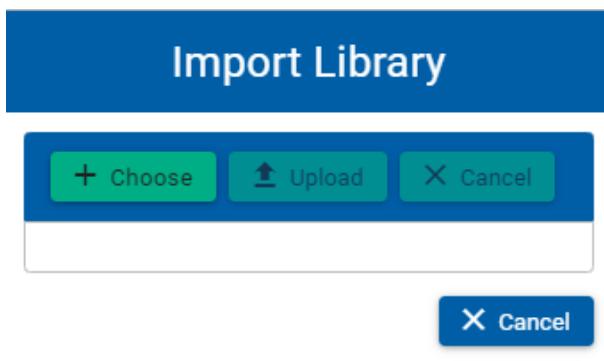
There is a Dependencies panel on the Library Details page to make a library dependent on the one of the imported ones. For more information about library dependency, see [Adding Library Dependency](#).



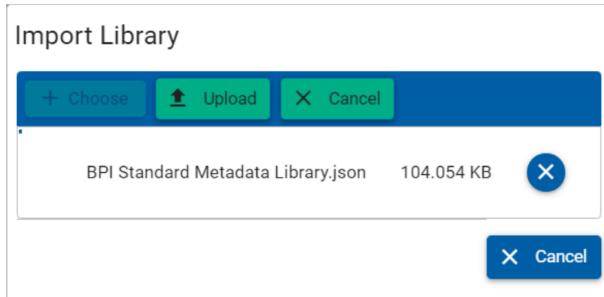
You cannot edit the metadata in an imported library.

To import the library:

1. Go to the Main Menu.
2. Select **Metadata Modeller > Import Library**.



3. Click **Choose** to select a .json file of a library.



4. Click **Upload** to upload the library.



Importing a library allows models in libraries to be moved between Metadata Modeller environments.

Create Equipment

The procedures in this section describe how to create and modify equipment types, including Racks, Devices, Shelves, Cards and so on. To create equipment:

1. Go to the Main Menu.
2. Select **Metadata Modeller > Create Equipment**.

The Create Equipment page opens.

| | Parent | Type | Family | Archetype Name | Archetype Instance Name | Actions | Primary | Acronym | Category | Description | Model Number List | Position Used | Vendor Used |
|--------------------------|------------------|------------------|------------------|----------------|-------------------------|---------|----------------------------------|---------|----------|-------------|-------------------|---------------|-------------|
| <input type="checkbox"/> | Please Select... | Please Select... | Please Select... | | | + ↴ ↑ ↵ | <input checked="" type="radio"/> | | | | | | |



The default library must be dependent on the other libraries (for example, Packaging Standard Metadata_Library – bpi.standard.metadata) that includes Metamodels to possibly create equipment.

3. Fill out the *Create Equipment* fields, then click **Apply**.

The following table lists all the column headers of the table displayed in the *Create Equipment* page.



In the *Create Equipment* page, you can implement all the changes based on the default library, which is displayed on the bottom-right side of the page.

| FIELD | PROCEDURES |
|----------------|---|
| Default Family | <p>Family refers to a vendor specific group.</p> <p>Select a family name from the drop-down list to create equipment for a specific family name. This field also allows you to create a new family name.</p> <p> When a new row is added in the table, Default Family name populates automatically in the Family field.</p> <p>Creating a New Family Name</p> <ol style="list-style-type: none"> 1. Select New from the drop-down list. 2. Fill out the Name and Notes in Create Family dialog box. 3. Click Apply. |
| Parent | Defines a parent name of equipment. |
| Type | Select a type of equipment from the drop-down list, such as Rack , Device , Shelf , Slot , Card , Pluggable and Physical Port . |
| Family | Similar to Default Family , select a family name or create a new family from the drop-down list. |
| Archetype Name | <p>Specifies the equipment type's metamodel. For example, each device type has its Archetype.</p> <p>Creating a New Archetype Name</p> <ol style="list-style-type: none"> 1. Select New from the drop-down list. 2. Enter the Name in Set Archetype Name dialog box. 3. Click Apply. |

| FIELD | PROCEDURES |
|-------------------------|---|
| Archetype Instance Name | <p>Defines the composition of archetypes. Allows you to enter a formula in the Current Value field (available next to the Default Family) to define the Archetype Instance Name. There are different formulas that is set by default in this field, you can also modify the default formula:</p> <ul style="list-style-type: none"> • [Archetype Name] for Rack, Device, Shelf, Card, Physical Termination Position, and Pluggable. • [n] for Rack Position. • 'Shelf Pos '+([n]+1)" for Shelf Position. • 'Slot '+([n]+1) for Slot Position. • ''([n]+1) for Physical Port. <p> After the Archetype Name is defined, it populates automatically in the Archetype Instance Name field. However, you can modify the auto-populated value.</p> |
| n | Provides the child equipment sequence within type. Numeration starts from 0 for each type. For example, if there are 2 Slot Position and 2 Shelf Position children, these children will have 0, 1 and 0,1 as n. |
| Actions | <p>There are four action buttons available, such as:</p> <ul style="list-style-type: none"> • Add Child (+) – Enables you to create a child row. For example, if you want to create a shelf position for a device, click the Add Child button. • Copy Below (↓) – Copies a row and pastes downward. • Copy Above (↑) – Copies a row and pastes upward. • Remove – Enables you to remove a row. You can undo the deletion by clicking the Undo icon. |

| FIELD | PROCEDURES |
|-------------------|--|
| Primary | <p>Click the radio button to indicate the equipment, which is available for selection in the UI. For example, if you create a device, shelf, slots, and shelf position, then an enabled radio button is displayed for the device to indicate the parent equipment among all the types of equipment.</p> <p> This is available only for top rows (archetype instances which don't have parent).</p> |
| Acronym | Defines an abbreviation of the equipment name. |
| Category | <p>Select a device category from the drop-down list.</p> <p>Active</p> <ul style="list-style-type: none"> • Must have a network role. • Device name is not mandatory but if device name is not set, it gets default. <p>Passive</p> <ul style="list-style-type: none"> • Does not need a network role. • Device name is mandatory. <p>Generic</p> <ul style="list-style-type: none"> • Does not need to be placed in a rack. <p>Other</p> <ul style="list-style-type: none"> • Does not need a network role. • Device name is not mandatory but if device name is not set, it gets default. |
| Description | Defines a description of the equipment. |
| Model Number List | Defines a model number list of equipment. Example of a list: NTK456T,NTH555X. |

| FIELD | PROCEDURES |
|-------------------|---|
| Position Used | Defines a position of the equipment. |
| Vendor | Defines a vendor for the equipment. |
| Width In Inches | Defines the equipment's width in inches. |
| Position Sequence | Populates automatically with the sequential position of equipment. You can also enter a formula after populated to change a position sequence of equipment. |
| Part Number | Defines a part number of the equipment. |
| Height In Inches | Defines the equipment's height in inches. |
| Port Number | Defines a port number of a card. |
| Port Sequence | Populates automatically with the port sequence of a card. |
| Port Type Name | Defines a port type name of a card. |
| Logical Interface | List of logical interface archetype instances available in the default library. This column is meaningful for Physical Ports. It creates a relationship <i>RELATIONSHIP_DELEGATES_CONSISTS_OF</i> between Physical Port Archetype Instance and selected Logical Interface Archetype Instance. |



When you click Add Child (+) in the Actions column to create child equipment, then only the compatible types are displayed on the drop-down menu in the Type column for child equipment.

Action Buttons

On the *Create Equipment* page, you see different action buttons, including:

| NAME | DESCRIPTION |
|----------|---------------------------|
| Checkbox | Selects the required row. |

| NAME | DESCRIPTION |
|-------------|---|
| Edit | <ul style="list-style-type: none"> • Edit icon in the row is an indicator if you modify the row. <p> Edit icon is not seen in the row if there is no modification in the row.</p> <ul style="list-style-type: none"> • If you want to edit the bulk rows, check the checkboxes to select the required rows, then click the Bulk Edit button available on the bottom-left of the screen. <p> All selected rows must be the same type.</p> |
| Bulk Remove | If you want to remove the bulk rows, check the checkboxes to select the required rows then click Bulk Remove . |
| Change Name | Allows you to change the name of the Archetype. |



An angular Gate ($\langle \rangle$) allows you to expand or collapse the column.

Modifying an Equipment Model

To modify the created equipment:

1. In the Metadata Modeller tree, navigate to the equipment you want to modify.
2. Right-click the equipment.
3. Select **Equipment Modeller**.

The Equipment Modeller page opens.

4. Modify the values in the given fields.
5. Click **Apply**.

Create and edit metadata

Creating metadata

The Create Metadata page adds support to create metadata nodes, relates them to Metamodel, Archetype and Archetype Instances, modify their configuration, and report on assigned Metadata.

Accessing the Create Metadata Page

1. Go to the Main Menu.
2. Select **Metadata Modeller** and then select **Create Metadata**.

The Create Metadata page opens.

This page consists of four sections:

- [Metadata Summary](#)
- [Callout Properties](#)
- [Additional Properties](#)
- [Metadata Relationships](#)

Metadata Summary

The Metadata Summary section includes the following mandatory fields to create a Metadata in the default Library:

Metadata Summary fields

| FIELD | DESCRIPTION |
|--------------|---|
| Family | Family refers to a vendor specific group. Select the family from the drop-down list. |
| Name | Static name value used for descriptive purposes on metadata items or as a default name for Blue Planet Inventory. |
| Priority | Determines the ordering of metadata when multiple levels are applied and allows rules to apply a chain of command pattern. For example, this field allows you to set the priority between Metadata nodes to other nodes, such as Archetype, Metamodel, or Archetype Instance. |
| Type | Select the type of metadata as a specific discovery type (see below) or Other for all other metadata. The selection identifies the metadata type to be configured which affects the panels for data entry. |
| Object Class | Appears when you select Type . You must define it to configure the Common Properties . |

Type

You can manage the different metadata entries by selecting the following metadata types:

Discovery Attributes

When you select **Discovery Attributes**, you need to choose from the **Object Class**. You can choose Device, Card, Pluggables, etc. from the Object Class.

The screenshot shows the 'Create Metadata' interface with a red border around the main content area. At the top, there are tabs for 'My Orders', 'Create Metadata' (which is active), and a delete icon. Below the tabs is a 'Metadata Summary' section with fields for 'Family' (DiscoveryTestMetadata), 'Name' (My Metadata Item), 'Priority' (1001), 'Type' (DiscoveryAttributes), and 'Object Class' (Device). Underneath is a 'Common Properties' section containing two properties: 'bpuaa_PlaceObjectInNetwork' and 'bpuaa_PlaceDeviceInLocation', both set to 'com.blueplanet.inventory.bpuua.BPUAADeviceExampleCalloous.placeDeviceInNetwork'. Below that is an 'Additional Properties' section with a 'Add New Property' button and a table with columns for Property, Data Type, and Value. The final section is 'Properties per Metadata Item', which has a table for adding new archetypes with columns for Archetype and Discovered Part Number List. At the bottom right are 'Default Library: Library#1234' and 'Cancel/Apply' buttons.

The two **Common Properties** are displayed when you select 'Device' in the Object class.

When a new Archetype is added, the part number property list defaults to those defined for the equipment. You can then edit the list for discovery.

Each row in the **Properties per Metadata Item** corresponds to a metadata node.

Discovery Position Matching

This screenshot shows the 'Create Metadata' interface with a red border. The 'Properties per Metadata Item' section is highlighted. It contains a table for adding new archetypes with columns for Archetype, Child Type, Sub Slot, Regex, and Offset. The 'Default Library' is set to 'Library#1234'. At the bottom right are 'Cancel/Apply' buttons.

Choose the **Object Class** from the provided list.

You can fill the metadata item in the **Properties per Metadata Item** section.

Discovery Ignore Object

Select the **Object Class** from the provided list.

Discovery Convert Object

Pick from list or add to list for Convert From.

Each row in the bottom panel table corresponds to a metadata node.

Discovery Connection Mapping

Pick from list or add to list for Layer Rate, Network Role and Directionality.

Logical Connection archetype is optional.

Relate Complement, Ignore and Ignore Subgraph are by default null.

Each row in the bottom panel table corresponds to a metadata node.

Discovery Interface Mapping

Pick from list or add to list for Layer Rate and Layer Rate Qualifier.

Layer Rate Qualifier and Capacity is optional.

Each row in the bottom panel table corresponds to a metadata node.

Callout Properties

The Callout Properties section allows you to specify the callout properties of a metadata.

There are various fields available in this section, including:

Callout Properties fields

| FIELD | DESCRIPTION |
|-----------------------------------|--|
| core_preRelationshipCreateParent | Callout before creating a new relationship (has parent and child variations) |
| core_preRelationshipCreateChild | |
| core_preRelationshipDestroyParent | Callout before destroying a new relationship (has parent and child variations) |
| core_preRelationshipDestroyChild | |

| FIELD | DESCRIPTION |
|------------------------------------|--|
| core_postRelationshipCreateParent | Callout after creating a new relationship (has parent and child variations) |
| core_postRelationshipCreateChild | |
| core_postRelationshipDestroyParent | Callout after destroying a new relationship (has parent and child variations) |
| core_postRelationshipDestroyChild | |
| core_postNodeCreate | Callout after node creation |
| core_postActivity | Callout after finishing the activity |
| core_postNodePropertiesSet | Callout after setting properties on a node |
| postActivityNode | Callout after core_PostActivity |
| core_preCopyToNewPerspectiveParent | Callout before copying a node to a new perspective (has parent and child variations) |
| core_preCopyToNewPerspectiveChild | |

Additional Properties

The Additional Properties section allows you to specify the additional properties of a metadata.

There are various fields and action button available in this section, including:

Additional Properties fields

| FIELD | DESCRIPTION |
|-----------|--|
| Property | Define the property name of a metadata. |
| Data Type | Allows you to select the type of the data from the drop-down list, such as String, Boolean, json, number, etc. |

| FIELD | DESCRIPTION |
|------------------|---|
| Value | Validates value of specified type, such as String, Boolean, json, number, etc.  You can also modify the value if the data type is in json format. |
| Add New Property | Adds a new row. |
| Delete icon | Enables you to delete a row. You can undo the deletion by clicking the Undo icon. |

Metadata Relationships

This section allows you to define the relationship between metadata nodes to other nodes, such as Metamodel, Archetype, or Archetype Instance.

Creating a new metadata and building its relationship

To add a new metadata:

1. Create a new library and set that as the default library. For more information, see [Create Library](#).
2. Add the dependency library. For more information, see [Adding Library Dependency](#).



Click the radio button of standard Metadata in the Tag column to add standard metadata library into the library. Adding standard metadata library allows you to create the equipment and set its compatibility.

3. Launch the Create Metadata page to create a new metadata. For more information, see [Accessing the Create Metadata Page](#).
4. Fill out the required fields in the Create Metadata page. For more information on how to fill out the fields, see [Metadata Summary fields](#), [Callout Properties fields](#), and [Additional Properties fields](#).
5. Click **Apply**.

A new metadata is now created.

To build a metadata relationship:

1. Add a new related node in the search box to build a relationship with metadata and then click **Add Relationship** in the Metadata Relationships section.

2. After adding a new metadata and its relationship, you can deploy it to Blue Planet Inventory. For more information on how to deploy metadata, see [Deploying created equipment](#).

The new added metadata and its relationship is now deployed to the Blue Planet Inventory.

Viewing a summary of metadata

This section describes how to view the summary of metadata.

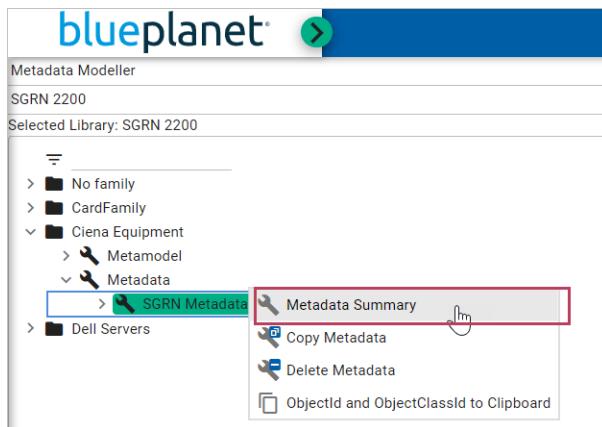
To view the summary of metadata:

1. In the Metadata Modeller tree, right-click the metadata you want to view summary.



The Metadata Modeller tree is extended to browse the Metamodel and Metadata.

2. Select **Metadata Summary**.



Displays Metadata Summary page.

3. You can view or modify the metadata nodes using this page.

4. Click **Apply** if you modified the fields.

Copying existing metadata

This section describes how to copy the existing metadata.

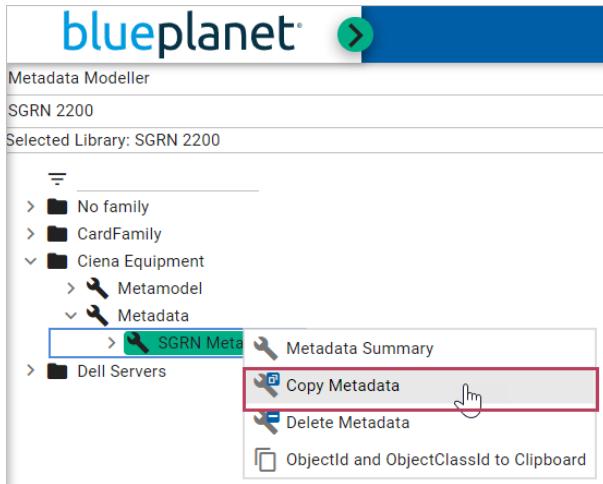
To copy the existing metadata:

1. In the Metadata Modeller tree, right-click the metadata you want to copy.



The Metadata Modeller tree is extended to browse the Metamodel and Metadata.

2. Select **Copy Metadata**.



Displays Copy Metadata page.

3. On the Metadata Summary section, select the Family name from the drop-down menu to add the copied metadata.

You can also modify the metadata nodes.

4. Click **Apply**.

The metadata is copied to the selected Family.

Deleting existing metadata

This section describes how to delete the existing metadata.

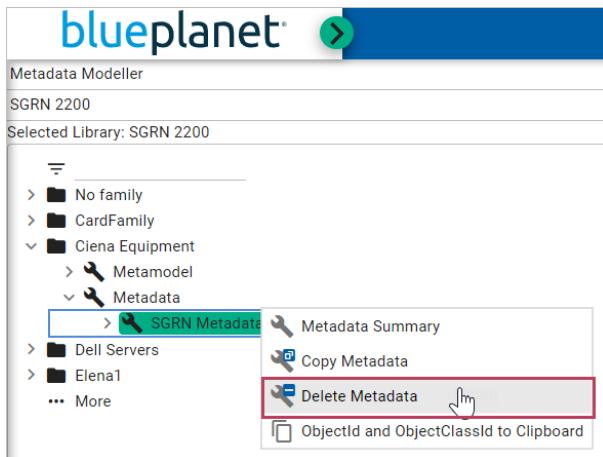
To delete the existing metadata:

1. In the Metadata Modeller tree, right-click the metadata you want to delete.

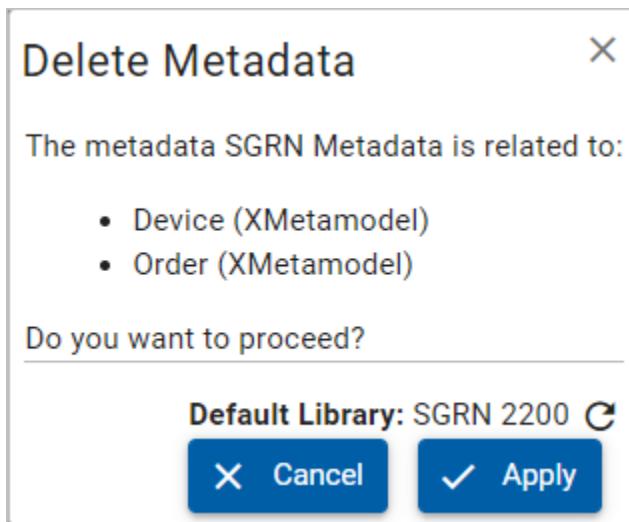


The Metadata Modeller tree is extended to browse the Metamodel and Metadata.

2. Select **Delete Metadata**.



A confirmation pop-up opens.



3. Click **Apply**.

The metadata is deleted from Blue Planet inventory.



You can not delete a metadata if it has any association with Metamodel, Archetype, or Archetype Instance.

Viewing the metadata report

This section describes how to see all the Metadata associated with a Metamodel, Archetype, or Archetype Instance (directly or indirectly via some parent level assignment).

The report also allows you to add and remove Metadata from the launched object.

To view the metadata report:

1. In the Metadata Modeller tree, right-click the Metamodel, Archetype, or Archetype Instance you want to view the report.

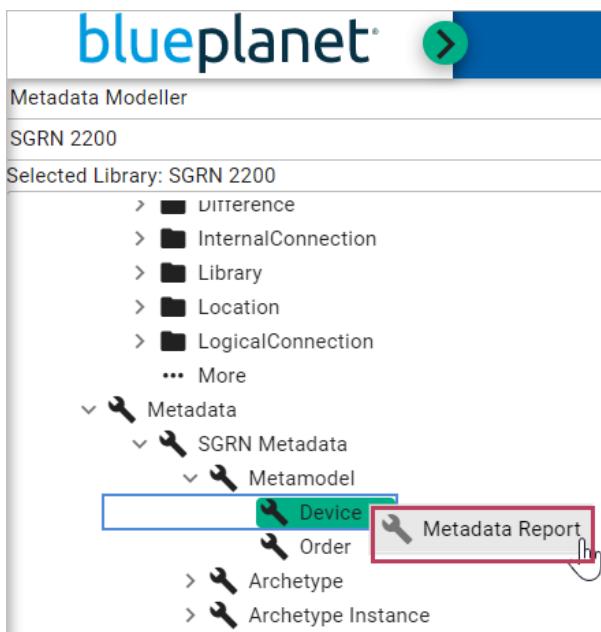
If the report is launched on a Metamodel node then you can only add or remove Metamodel scope.

If the report is launched on an Archetype node then you can only add or remove Archetype scope.

Metamodel items includes the read-only fields.

If the report is launched on an Archetype Instance node then you can only add or remove Archetype Instance scope. Metamodel and Archetype items includes the read-only fields.

2. Select **Metadata Report**.



Displays Metadata Report page.

The screenshot shows the 'Metadata Report - Device' page. At the top, there's a header 'Metadata Report - Device' and a 'Global Search' bar. Below it, a section titled 'My Associated Metadata' shows a table with one row. The table has columns for 'Name', 'Priority', and 'Scope'. The row contains 'SGRN Metadata', '1000', and 'Metamodel'. There are 'Cancel' and 'Apply' buttons at the bottom right.

| | Name | Priority | Scope |
|-----------------------|---------------|----------|-----------|
| <input type="radio"/> | SGRN Metadata | 1000 | Metamodel |

3. In the My Associated Metadata section, click the radio button to display the Metadata Details that includes Callout Properties, Additional Properties, and Metadata Relationships sections.



These three sections, such as Callout Properties, Additional Properties, and Metadata Relationships includes read-only fields.

The screenshot shows the 'blueplanet' application interface. In the top navigation bar, there is a 'Global Search' field and a gear icon. Below the header, a sidebar on the left shows 'Metadata Report - Device'. The main content area has a title 'My Associated Metadata'. At the top of this section is a table with columns 'Name', 'Priority', and 'Scope'. A red callout with the text 'Click Radio Button to display Metadata Details' points to the radio button in the first row of this table. Below the table, there are three expandable sections: 'Callout Properties', 'Additional Properties', and 'Metadata Relationships'. Each section contains a table with specific details. At the bottom of the page, there is an 'Inspector' button and a copyright notice: 'Copyright© 2021 Ciena® Corporation, Inc. All Rights Reserved.'

- a. You can add new metadata nodes from the Inventory tree and then click **Add Metadata**.

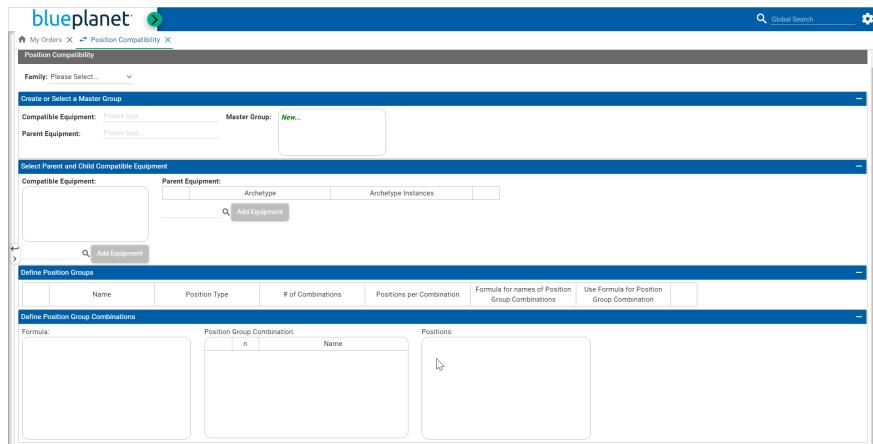


Delete icon enables you to delete a row. You can undo the deletion by clicking the Undo icon.

- b. Click **Apply** to add a metadata.

Position Compatibility Between Equipment Types

The steps in this section describe how it is possible to put pieces of equipment within another equipment piece, and into which positions. For example, compatibility between devices and rack positions in a rack; and compatibility between cards and slot positions in a shelf.



In this section, you can define a set of child equipment that can go into the same positions defined within a set of parent equipment. For example, you might want to determine that a set of line card types can go into some slots in many of the shelves for a particular equipment manufacturer. For this, you'll need to do the following:

1. Specify all card types as the child equipment and all shelves as the parent equipment.
2. To specify the exact positions within each shelf, select each shelf and define one or more position groups in them, which can take all the cards. Position groups generally define ranges of slots that can take all the child cards. So, if the cards can go into positions 1-5 and then 8-10, you could define two position groups for each range of positions. You can also define one position group with those positions individually identified.

To position compatibility of equipment:

1. Go to the Main Menu.
2. Select **Metadata Modeller > Position Compatibility**.

The **Position Compatibility** page opens.

3. Fill out the **Position Compatibility** page, and click **Apply**.

There are different tabs available on the **Position Compatibility** page, including:

- [Create or Select a Master Group](#)
- [Select Parent and Child Compatible Equipment](#)
- [Define Position Groups](#)
- [Define Position Group Combinations](#)



In the **Position Compatibility** page, you can implement all the changes based on the default library, which is displayed on the bottom-right side of the page.

Family

Family refers to a vendor-specific group.

Select a family name from the drop-down list to create equipment compatibility for a specific family name.

Create or Select a Master Group

Create or Select a Master Group consists of different fields, including:

| FIELD | DESCRIPTION |
|----------------------|--|
| Master Group | Defines a name for the configuration of compatible child and parent equipment pieces. |
| Compatible Equipment | Defines an equipment that is compatible. For example, card types. |
| Parent Equipment | Defines the parent equipment. For example, shelf types that contain slots into which cards can be added. |



Compatible Equipment and *Parent Equipment* are filters. When compatible equipment is entered, all the Master Groups are filtered to also include the entered compatible equipment.

Select Parent and Child Compatible Equipment

Select Parent and Child Compatible Equipment consists of different fields, including:

| FIELD | PROCEDURE |
|----------------------|--|
| Compatible Equipment | <p>Add a child equipment type for which you want to create compatibility with the parent equipment. For example, card is the child equipment for a shelf.</p> <p> Search suggestions are displayed based on the compatible type for the parent equipment.</p> |
| Parent Equipment | <p>Add a parent equipment type for which you want to create compatibility with the child equipment. For example, shelf is the parent equipment for a card.</p> |

Action Buttons

| FIELD | DESCRIPTION |
|------------------|---|
| Remove | Enables you to remove an equipment. You can undo the deletion by clicking the Undo icon. |
| Radio button | Displays the Parent Equipment details in Define Position Groups . |
| Angular Gate (↔) | Expands or collapses the column. |
| Search | <p>Populates the relevant search suggestion when you start typing characters in the field.</p> <p> The search suggestion is based on the selected Family.</p> |

Define Position Groups

For the selected parent equipment, you can define one or more set of positions within that parent where all the compatible child equipment can go into. You can also define a position group for each contiguous set of positions, and determine the positions within that group that the child equipment occupy.

For example, if you want to determine the cards that can go into different slots in a parent shelf, the parameters of the position group allow you to define ‘double width’ cards. However, you can also determine position groups that represent non-regular shaped cards, or to indicate where a shelf has both a front and a back, where a card type might poke through and occupy both the front and back of a shelf.

Here are the different possibilities to make it easier to define how card types can fit in the shelf’s slots:

1. Define how many different possible positions the cards can fit in (given the number of slots each card occupies and the number of slots available in this run).
2. Either define a formula to exactly determine which slots the cards occupy (use this for regular-sized cards), or for each position, you can manually pick the exact slots that the cards occupy (use this for irregular cards).
3. Where a gap is required, you can either use two position groups (one for each contiguous run of slots), or use direct compatibility and define different slot types for the ‘gap’.
4. Lastly, you need to determine a ‘name’ for each of the positions that the cards occupy. For example, double-sized cards might fit in “slots 1 and 2”, and “slots 2 and 3”. The slots can also be named as “Slot 1-2”, “Slot 2-3” and so on, based on your preference. You can either use a formula to name the positions, or manually name the positions individually.

Define Position Groups consists of different fields, including:

| FIELD | DESCRIPTION |
|---------------|---|
| Name | Defines the name. If you do not set a name then the name field displays the default name. |
| Position Type | Defines the position type of the parent equipment. |

| FIELD | DESCRIPTION |
|--|---|
| # of Combinations | <p>Automatically populates with the number of combinations for the parent equipment. For example, if you create 14 shelf positions while creating equipment for a shelf, then the number in the field displays as 14.</p> <p>The value in this field also depends on the Position type. For example, if the parent equipment is a card which consists of 2 Physical Termination Positions and 3 Physical Ports:</p> <ul style="list-style-type: none"> When <i>Physical Termination Positions</i> is selected in Position type, then the value in # of Combinations is displayed as 2. When <i>Physical Ports</i> is selected in Position type, then the value in # of Combinations is displayed as 3. |
| Positions per Combination | You can adjust the positions as required. |
| Formula for names of Position Group Combinations | Allows you to enter a formula to display a position in the Position Group Combination . When there is no formula, then you see the position with an empty name. It allows you to auto-populate and update the name of the positions, using the same formula. For example, if all the positions are named as My position 0, My position 1 and so on, then the formula is set as " My positon " + [n]. |
| Use Formula for Position Group Combination | Select the checkbox to use the formula for Position Group Combination . If you uncheck the checkbox, you can manually change the position. |

Action Buttons

| FIELD | DESCRIPTION |
|-------------------|--|
| Angular Gate (<>) | Expands or collapses the column. |
| Radio button | Displays the Position Groups details in Define Position Group Combinations tab. |
| Remove | Enables you to remove the selected row. |

Define Position Group Combinations

Define Position Group Combinations consists of different fields, including:

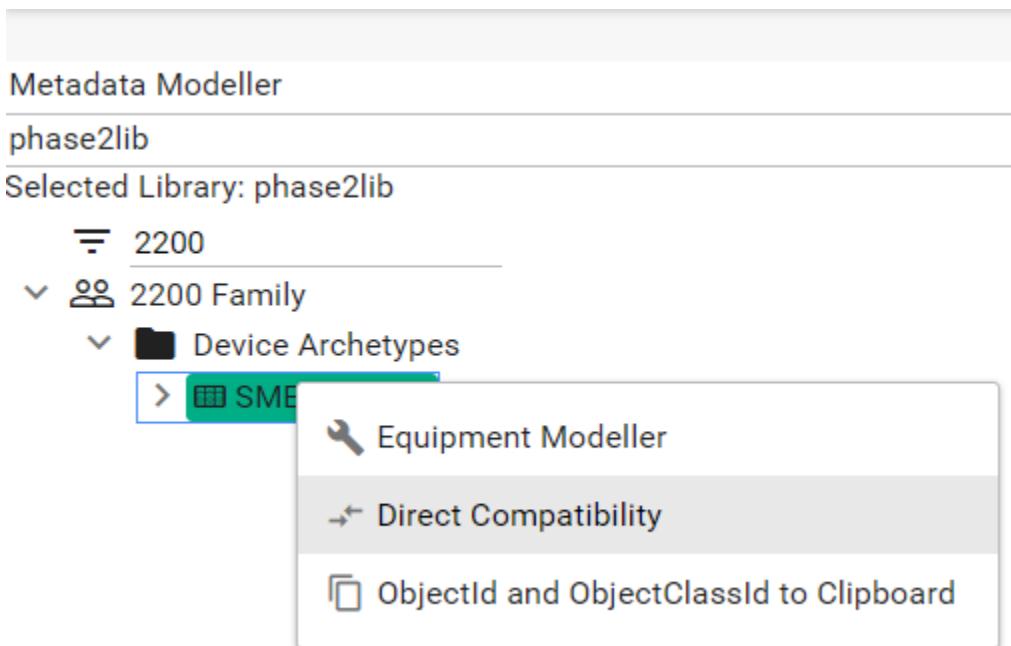
| FIELD | DESCRIPTION |
|----------------------------|--|
| Formula | <p>Displays the formula as defined in Define Position Groups. For example:</p> <ul style="list-style-type: none">• If the value of Positions per Combination is 1, then Formula is displayed as [n].• If the value of Positions per Combination is 2, then Formula is displayed as [n] and [n]+1• If the value of Positions per Combination is 3, then Formula is displayed as [n], [n]+1, [n]+2. |
| Position Group Combination | Displays the position group combination as defined in Define Position Groups . |
| Positions | Displays the positions as defined in Define Position Groups . |

Direct Compatibility Between Archetypes

The procedures in this section describe how to directly add the compatibility between archetypes. For example, you can add compatibility between a device and shelf.

Accessing the Direct Compatibility Page

1. Set a library as a default library. For more information, see [default library](#).
2. In the Metadata Modeller tree, select the default library and then navigate to the archetype.
3. Right-click the archetype.



4. Select **Direct Compatibility**.

The Direct Compatibility page opens.

The screenshot shows the 'Direct Compatibility:SME 2200' page. In the 'Associated Types' section, there is a table with two rows:

| | Name | Type |
|-----------------------|-----------|---------------|
| <input type="radio"/> | device 4 | Device |
| <input type="radio"/> | shelf p 4 | ShelfPosition |

In the 'Compatible Types' section, there is a table with one row:

| | Name | Type |
|--------------------------|---------|-------|
| <input type="checkbox"/> | shelf 4 | Shelf |

Below the tables are 'Add Type' and 'Search' buttons. At the bottom right are 'Cancel' and 'Apply' buttons, with 'Default Library: phase2lib C' above them.

Adding the Compatibility Between Archetypes

1. On the Direct Compatibility page, select the family you want to set the compatibility from the **Family** drop-down list.

The screenshot shows the 'Direct Compatibility:SME 2200' page with the 'Family' dropdown open. The options are:

- Please Select...
- No Family
- 2200 Family** (highlighted in green)
- A Property
- A Demo Family

2. In the Compatible Types section, add the archetype in the **Add Type** field. Relevant archetype suggestions of the selected family appear when you start typing characters in the field.



You can also drag and drop an archetype to the **Add Type** field.

Compatible Types

Hide types which can not be compatible with currently selected Associated Type

| | Name | Type |
|--------------------------|---------|-------|
| <input type="checkbox"/> | shelf 4 | Shelf |

s

- device 4 (Device)
- net4 (Network)
- rack 4 (Rack)
- shelf p 4 st (ShelfPosition)**
- shelf p4 (ShelfPosition)

3. Click **Add Type** to add the archetype in the table of the Compatible Types section.
4. In the Associated Types section, select the radio button of an archetype that you want to set the compatibility.



In Compatible Types section, the selected checkbox indicates the compatible type with the selected Associated Type.

5. Select the archetype checkbox in the Compatible Types section and click **Apply** to add the compatibility between archetypes.

The screenshot shows the 'Direct Compatibility' dialog in the Blue Planet Inventory Metadata Modeller. The 'Associated Types' section lists 'device 4' (Device) and 'shelf p 4' (ShelfPosition). The 'shelf p 4' row is selected and highlighted with a red border. The 'Compatible Types' section lists 'rack 4' (Rack) and 'shelf 4' (Shelf). The 'shelf 4' row is selected and highlighted with a red border. At the bottom right, there are buttons for 'Default Library: phase2lib' (with a 'C' icon), 'Cancel', and 'Apply'.

Using the Compatible Archetype in the Blue Planet Inventory

After you added the compatibility between archetypes, open the Device Summary page to use the created compatibility.



After setting direct compatibilities via Metadata Modeller, the changes need to be deployed first. Only after deployment it becomes a real model and then can be used on the Device Summary page.

1. On the Device Summary page, click **Add Shelf** in the Device Structure section.
2. Select the created compatible archetype from the drop-down list in Shelf Type column.
3. Add the order name in the **Changes applied to** field.

The screenshot shows the 'Device Structure' tab in the 'Device Summary - SME 2200_HAMILTON_0947' section. A new row is being added to the table, indicated by a red box around the 'Add Shelf' button in the top-left corner of the grid. The first column of the new row contains a checkbox and the text 'shelf 4'. Another red box highlights this cell. The rest of the columns in this row show 'dfcsd34355' in the 'Shelf Type' dropdown, 'Planned' in the 'Status' dropdown, '1' in 'Height (RUs)', '22' in 'Width (in)', and 'None-' in both 'Floor' and 'Room' dropdowns. The 'Rack' and 'Rack Position' columns also show 'None-' and 'None-' respectively. On the far right of the table, there are two 'Display Rack' buttons, one for each rack entry.

4. Click **Apply**.

Metadata Modeller Workflow

Creating a library

1. Create a first Library on the **Create Library** page. For more information, see [Create Library](#).

The screenshot shows the 'Create Library' dialog box. It has a blue header bar with the title 'Create Library'. Below the header, there are two input fields: one for 'Name' containing 'Deva_Library_1' and another for 'Namespace prefix' containing 'bpi.deva.library'. At the bottom of the dialog are three buttons: a checked checkbox for 'Set as Default Library', a 'Cancel' button with a red 'X', and an 'Apply' button with a green checkmark.

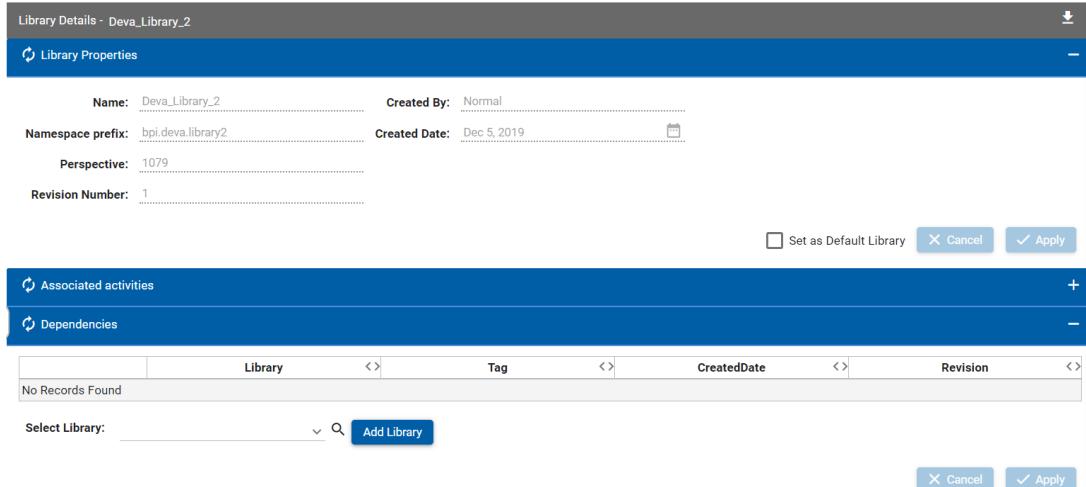
2. After you create a library, you can see the details of the created library. For more information see [Library Details](#).

The screenshot shows the 'Library Details - Deva_Library_1' page. It has a dark header bar with the title 'Library Details - Deva_Library_1'. Below the header, there is a section titled 'Library Properties' with fields for 'Name' (Deva_Library_1), 'Namespace prefix' (bpi.deva.library), 'Created By' (Normal), 'Created Date' (Dec 5, 2019), 'Perspective' (1078), and 'Revision Number' (5). At the bottom of this section are three buttons: a checked checkbox for 'Set as Default Library', a 'Cancel' button with a red 'X', and an 'Apply' button with a green checkmark. Below this section, there are two expandable sections: 'Associated activities' and 'Dependencies'.

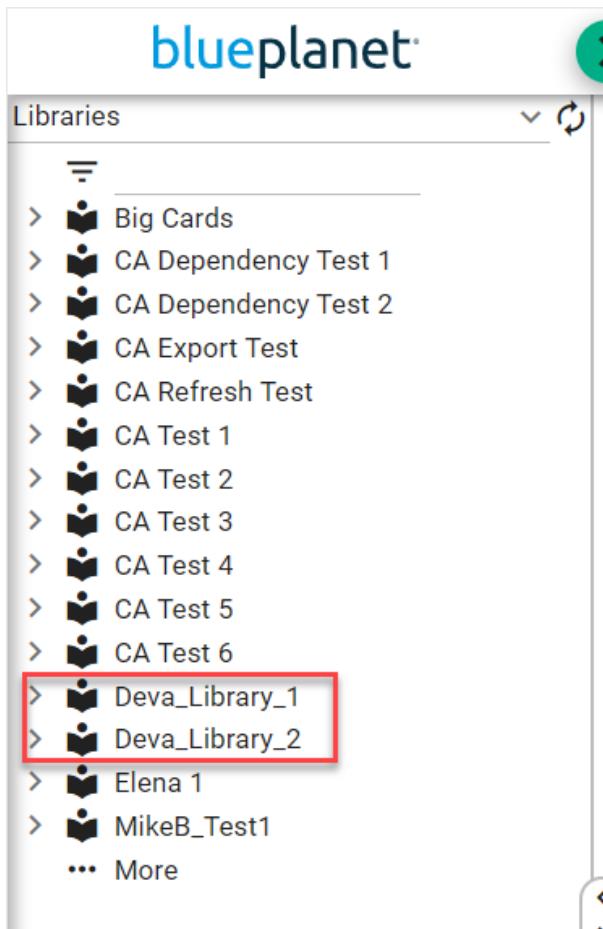
3. Create a second Library on the **Create Library** page.

The screenshot shows the 'Create Library' dialog box again. This time, the 'Name' field contains 'Deva_Library_2' and the 'Namespace prefix' field contains 'bpi.deva.library2'. The buttons at the bottom are the same: a checked checkbox for 'Set as Default Library', a 'Cancel' button with a red 'X', and an 'Apply' button with a green checkmark.

4. After you create a library, you can see the details of the created library.



5. After creating both libraries, they appear in the **Libraries** tree.



Creating Equipment

1. Create a Rack on *Create Equipment* page. For more information, see [Create Equipment](#)

The screenshot shows the 'Create Equipment' page with the following details:

- Default Family:** Deva_Family
- Current value:** 23
- Table Headers:** Parent <>, Type <>, Family <>, Archetype<> Name, Archetype<> Instance, Actions <>, Primary, Vendor, Width In Inches <>, Position Sequence <>
- Table Data:**

| | | | | | | | | | |
|-------------------------------------|------|-------------|-------------|-------------|----------|----------------------------------|------|----|--|
| <input checked="" type="checkbox"/> | Rack | Deva_Family | Deva_Rack_2 | Deva_Rack_2 | + ↓ ↑ 🗑️ | <input checked="" type="radio"/> | Deva | 23 | |
|-------------------------------------|------|-------------|-------------|-------------|----------|----------------------------------|------|----|--|

2. Create a Rack Position for the rack on the *Create Equipment* page.

The screenshot shows the 'Create Equipment' page with the following details:

- Default Family:** Deva_Family
- Current value:** 23
- Table Headers:** Parent <>, Type <>, Family <>, Archetype<> Name, Archetype<> Instance, Actions <>, Primary, Vendor, Width In Inches <>, Position Sequence <>
- Table Data:**

| | | | | | | | | | |
|-------------------------------------|---------------|-------------|---------------|-------------|----------|----------------------------------|------|----|---|
| <input checked="" type="checkbox"/> | Rack | Deva_Family | Deva_Rack_2 | Deva_Rack_2 | + ↓ ↑ 🗑️ | <input checked="" type="radio"/> | Deva | 23 | |
| <input checked="" type="checkbox"/> | Rack Position | Deva_Family | Deva_Rack_Pos | 0 | 0 | + ↓ ↑ 🗑️ | | | 0 |
| <input checked="" type="checkbox"/> | Rack Position | Deva_Family | Deva_Rack_Pos | 1 | 1 | + ↓ ↑ 🗑️ | | | 1 |
| <input checked="" type="checkbox"/> | Rack Position | Deva_Family | Deva_Rack_Pos | 2 | 2 | + ↓ ↑ 🗑️ | | | 2 |

3. Create a Device on the *Create Equipment* page.

The screenshot shows the 'Create Equipment' page with the following details:

- Default Family:** Deva_Family
- Current value:** 20
- Table Headers:** Parent <>, Type <>, Family <>, Archetype<> Name, Archetype<> Instance, Actions <>, Primary, Acronym, Category, Description, Model Number, Position Used, Vendor, Width In Inches
- Table Data:**

| | | | | | | | | | | | | | |
|-------------------------------------|--------|-------------|-------------|-------------|----------|----------------------------------|----|--------|--------|------|---|------|----|
| <input checked="" type="checkbox"/> | Device | Deva_Family | Deva_Router | Deva_Router | + ↓ ↑ 🗑️ | <input checked="" type="radio"/> | DR | Active | router | DR-1 | 4 | Deva | 20 |
|-------------------------------------|--------|-------------|-------------|-------------|----------|----------------------------------|----|--------|--------|------|---|------|----|

4. Create a Shelf Position for the device on the *Create Equipment* page.

The screenshot shows the 'Create Equipment' page with a table of equipment items. The table has the following columns:

| | Parent | Type | Family | Archetype Name | Archetype Instance | Actions | Primary | Acronym | Category | Description | Model Number | Position Used | Vendor | Width in Inches | Position Sequence |
|-------------------------------------|--------|------------------|-------------|----------------|--------------------|-----------|----------------------------------|---------|----------|-------------|--------------|---------------|--------|-----------------|-------------------|
| <input checked="" type="checkbox"/> | | Device | Deva_Family | Deva_Router | Deva_Router | + ↓ ↑ 🗑 | <input checked="" type="radio"/> | DR | Active | router | DR-1 | 4 | Deva | 20 | |
| <input checked="" type="checkbox"/> | | > Shelf Position | Deva_Family | Deva_Shelf Pos | Shelf Pos 1 | 0 + ↓ ↑ 🗑 | | | | | | | | | 0 |
| <input checked="" type="checkbox"/> | | Shelf Position | Deva_Family | Deva_Shelf Pos | Shelf Pos 2 | 1 + ↓ ↑ 🗑 | | | | | | | | | 1 |
| <input checked="" type="checkbox"/> | | Shelf Position | Deva_Family | Deva_Shelf Pos | Shelf Pos 3 | 2 + ↓ ↑ 🗑 | | | | | | | | | 2 |
| <input checked="" type="checkbox"/> | | Shelf Position | Deva_Family | Deva_Shelf Pos | Shelf Pos 4 | 3 + ↓ ↑ 🗑 | | | | | | | | | 3 |

5. Create a Shelf in Shelf Position on the *Create Equipment* page.

The screenshot shows the 'Create Equipment' page with a table of equipment items. The table has the same columns as the previous screenshot. A row for a 'Shelf' has been selected and highlighted with a red box. The selected row contains the following data:

| | | | | | | | | | | | | | | |
|-------------------------------------|--|-------|-------------|------------|------------|-----------|----------------------------------|----|-------|------|---|------|----|--|
| <input checked="" type="checkbox"/> | | Shelf | Deva_Family | Main Shelf | Main Shelf | 0 + ↓ ↑ 🗑 | <input checked="" type="radio"/> | SP | shelf | DS-1 | 2 | Deva | 12 | |
|-------------------------------------|--|-------|-------------|------------|------------|-----------|----------------------------------|----|-------|------|---|------|----|--|

6. Create a Slot Position for the Shelf on the *Create Equipment* page.

| | Parent | Type | Family | Archetype Name | Archetype Instance | n | Actions | Primary | Acronym | Catégorie | Description | Modèle | Position Used | Vendeur | Width in Inches | Position Sequence |
|--|--------|----------------|-------------|----------------|--------------------|---|---------|---------|---------|-----------|-------------|--------|---------------|---------|-----------------|-------------------|
| | | Device | Deva_Family | Deva_Router | Deva_Router | 0 | + ↓ ↑ 🗑 | ⊕ | DR | Active | router | DR-1 | 4 | Deva | 20 | |
| | | Shelf Position | Deva_Family | Deva_Shelf Pos | Shelf Pos 1 | 0 | + ↓ ↑ 🗑 | | SP | shelf | DS-1 | 2 | Deva | 12 | 0 | |
| | | Shelf | Deva_Family | Main Shelf | Main Shelf | 0 | + ↓ ↑ 🗑 | | | | | | | | 0 | |
| | | Slot Position | Deva_Family | slot-position | Slot 1 | 0 | + ↓ ↑ 🗑 | | | | | | | | 1 | |
| | | Slot Position | Deva_Family | slot-position | Slot 2 | 1 | + ↓ ↑ 🗑 | | | | | | | | 2 | |
| | | Slot Position | Deva_Family | slot-position | Slot 3 | 2 | + ↓ ↑ 🗑 | | | | | | | | 3 | |
| | | Slot | Deva_Family | slot- | Slot 4 | 3 | + ↓ ↑ 🗑 | | | | | | | | | |

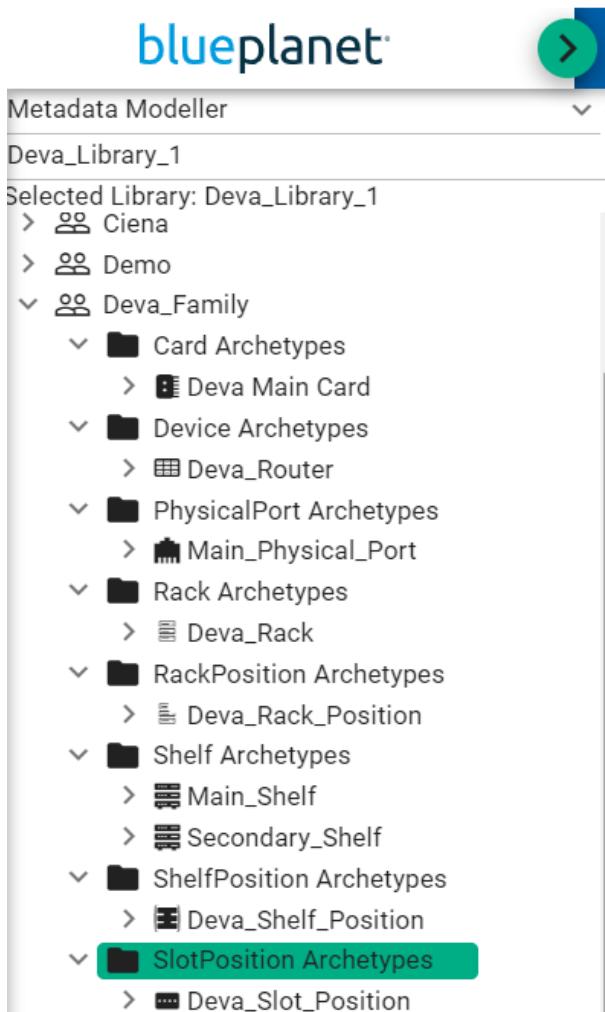
7. Create a Card on the *Create Equipment* page.

| | Parent | Type | Family | Archetype Name | Archetype Instance | n | Actions | Primary | Acronym | Catégorie | Description | Modèle | Position Used | Vendeur | Width In Inches |
|--|--------|------|-------------|----------------|--------------------|---|---------|---------|---------|-----------|-------------|--------|---------------|---------|-----------------|
| | | Card | Deva_Family | Main card | Main card | 0 | + ↓ ↑ 🗑 | ⊕ | MC | | doc card | DC-1 | 2 | Deva | 10 |
| | | | | | | | | | | | | | | | |

8. Create a Physical Port for the Card on the *Create Equipment* page.

| | Parent | Type | Family | Archetype Name | Archetype Instance | n | Actions | Primary | Acronym | Catégorie | Description | Modèle | Position Used | Vendeur | Width In Inches |
|--|--------|---------------|-------------|----------------|--------------------|---|---------|---------|---------|-----------|-------------|--------|---------------|---------|-----------------|
| | | Card | Deva_Family | Main card | Main card | 0 | + ↓ ↑ 🗑 | ⊕ | MC | | doc card | DC-1 | 2 | Deva | 10 |
| | | Physical Port | Deva_Family | Main Physical | Main Physical | 1 | + ↓ ↑ 🗑 | | | | | | | | |
| | | Physical Port | Deva_Family | Main Physical | Main Physical | 2 | + ↓ ↑ 🗑 | | | | | | | | |
| | | Physical Port | Deva_Family | Main Physical | Main Physical | 3 | + ↓ ↑ 🗑 | | | | | | | | |
| | | Physical Port | Deva_Family | Main Physical | Main Physical | 4 | + ↓ ↑ 🗑 | | | | | | | | |

9. After creating all equipment, they appear in the **Metadata Modeller** tree.



Adding Library Dependency

1. If Library1 is dependent on equipment in Library2, then you can add Library2 as a dependency for Library1 from the Library Details page. For more information about the Library Details page, see [Library Details](#).



To add a library as a dependency, at least one activity in that library should be tagged.



Click the radio button of standard Metadata in the Tag column to add standard metadata library into the library. Adding standard metadata library allows you to create the equipment and set its compatibility.

Library Details - Deva_Library_1

Library Properties

| | | | |
|-------------------|------------------|---------------|---|
| Name: | Deva_Library_1 | Created By: | Normal |
| Namespace prefix: | bpi.deva.library | Created Date: | Dec 5, 2019 <input style="width: 20px; height: 20px;" type="button" value="..."/> |
| Perspective: | 1078 | | |
| Revision Number: | 5 | | |

Set as Default Library Apply

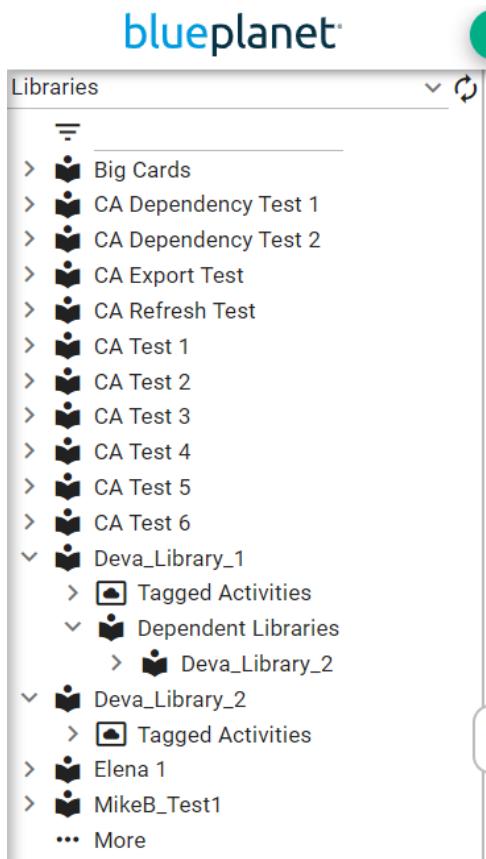
Associated activities

Dependencies

| | Library | Tag | CreatedDate | Revision |
|---|----------------|---------------|--------------------------|----------|
| . | Deva_Library_2 | Deva Rack 1.0 | Dec 5, 2019, 12:29:16 PM | 1 |

Select Library: Dev Apply

2. The added dependency appears in the **Libraries** tree.



After you add Library2 as a dependency in Library1, you can now modify metadata available in Library2, as part of Library1.

Position Compatibility

1. You can create compatibility between a Rack and Device on the *Position Compatibility* page. For more information, see [Position Compatibility](#).

The screenshot shows the 'Position Compatibility' page. In the 'Create or Select a Master Group' section, 'Family: Deva_Family' is selected. Under 'Master Group:', 'Deva Card Group' is chosen. In the 'Select Parent and Child Compatible Equipment' section, 'Deva_Router' is listed under 'Compatible Equipment' and 'Deva_Rack' is listed under 'Parent Equipment'. Both entries have a red error icon next to them. A blue 'Add Equipment' button is visible.

The screenshot shows the 'Define Position Groups' page. A table row for 'Deva_Rack_To_Device_Cor' is selected, showing 'RackPosition' as the position type, '41' as the '# of Combinations', and '4' as 'Positions per Combination'. The formula '(n)+*(n+1)' is entered in the 'Formula for names of Position Group Combinations' field, and the 'Use Formula for Position Group Combination' checkbox is checked.

The screenshot shows the 'Define Position Group Combinations' page. The 'Formula' dropdown contains '=n', '=n+1', '=n+2', and '=n+3'. The 'Position Group Combination' table shows four rows: 0, 1, 2, and 3. The 'Positions' list on the right shows combinations from 0-0 to 5-5.

- If the **# of Combination** and **Positions per Combination** is incorrectly entered, the Position Group Combination displays an error, which is highlighted in red.



For example, If you have 3 rack position and your device occupies 2 rack position, then **# of Combination** and **Positions per Combination** must be 2.

The screenshot shows the 'Define Position Groups' page. The '# of Combinations' field is set to '45', while 'Positions per Combination' is still '4'. The formula '(n)+*(n+1)' is entered in the 'Formula for names of Position Group Combinations' field, and the 'Use Formula for Position Group Combination' checkbox is checked.

The screenshot shows the 'Define Position Group Combinations' page. The 'Formula' dropdown contains '=n', '=n+1', '=n+2', and '=n+3'. The 'Position Group Combination' table shows four rows: 41, 42, 43, and 44. The entire table is highlighted in red, indicating an error. The 'Positions' list on the right shows combinations from 0-0 to 5-5.

2. You can also create compatibility between a Shelf and Card on the *Position Compatibility* page.

The screenshot shows the 'Position Compatibility' page. At the top, it says 'Family: Deva_Family'. Below that, there's a section titled 'Create or Select a Master Group' with a sub-section 'Select Parent and Child Compatible Equipment'. It shows 'Compatible Equipment' (Deva Main Card) and 'Parent Equipment' (Main_Shelf). There are 'Archetype' and 'Archetype Instances' columns. A button 'Add Equipment' is visible. Below this, the 'Define Position Groups' section is shown with a table:

| | Name | Position Type | # of Combinations | Positions per Combination | Formula for names of Position Group Combinations | Use Formula for Position Group Combination |
|----------------------------------|---------|---------------|-------------------|---------------------------|--|--|
| <input checked="" type="radio"/> | Default | SlotPosition | 3 | 2 | 'Position '<[n]+1) | <input checked="" type="checkbox"/> |

A 'New...' button is at the bottom left. The 'Define Position Group Combinations' section follows, with a table:

| n | Name |
|---|------------|
| 0 | Position 1 |
| 1 | Position 2 |
| 2 | Position 3 |

On the right, a 'Positions:' list shows: 0 - Slot 1, 1 - Slot 2, 2 - Slot 3, 3 - Slot 4.

Deploying created equipment

1. In the Associated activities section of the Library Details page, click **Validate** to see the report of a library in the Validation section to check if deployments impact existing models.



Validate the work only if you have made some changes within library and wants to deploy them.

2. Click **Tag** to add a tag name.

The screenshot shows the 'Library Details - Deva_Library_2' page. It has sections for 'Library Properties' (Name: Deva_Library_2, Created By: Normal, Namespace prefix: bpi.deva.library2, Created Date: Dec 5, 2019, Perspective: 1079, Revision Number: 2) and 'Associated activities'. The 'Associated activities' table has columns: Activity Name, Created Date, Revision Number, Tag Name, and a 'Tag' button. Two rows are shown:

| | Activity Name | Created Date | Revision Number | Tag Name | |
|-----|---------------|--------------|-----------------|--------------------|---------------|
| • > | Deva_Rack_2 | Dec 5, 2019 | 2 | Latest(Not Tagged) | Tag |
| • > | Deva_Rack | Dec 5, 2019 | 1 | Deva Rack 1.0 | Deploy |

Red arrows point to the 'Tag' button in the first row and the 'Tag' column header.

3. Fill in the **Name** in the **Add Tag Name** pop-up window and click **Apply**.



4. Click **Deploy** to deploy the equipment to the inventory.

| | Activity Name | Created Date | Revision Number | Tag Name | |
|-----|---------------|--------------|-----------------|---------------|---------------|
| • > | Deva_Rack_2 | Dec 5, 2019 | 2 | Deva Rack 2.0 | Deploy |
| • > | Deva_Rack | Dec 5, 2019 | 1 | Deva Rack 1.0 | Deploy |

5. Equipment is deployed to the inventory.

| | Activity Name | Created Date | Revision Number | Tag Name | |
|-----|---------------|--------------|-----------------|---------------|-----------------|
| • > | Deva_Rack_2 | Dec 5, 2019 | 2 | Deva Rack 2.0 | Deployed |
| • > | Deva_Rack | Dec 5, 2019 | 1 | Deva Rack 1.0 | Deployed |

Validating deployment of created equipment

After you create compatibility between two equipment types, open the *Device Properties* page to verify the created compatibility.

Device Summary - DEVA_ROUTER_1

Planned

Device Properties

| | | | |
|-----------------|-------------|--|---------------|
| *Device Type: | Deva_Router | Device Name: | DEVA_ROUTER_1 |
| *Part Number: | DR-1 | IP Address: | |
| Vendor: | Deva | Status: | |
| Height(RUs): | 4 | *Network: | Other 0001 |
| Width(inches): | 12 | <Select> | |
| *Role: | Ethernet | NMS: | |
| <Select> | | | |
| *Building: | MyBuilding | | |
| Floor/Room: | -None- | | |
| *Rack: | Deva_Rack_1 | <input type="checkbox"/> Only list racks with space. | |
| *Rack Position: | 0-1 | <input type="button" value="Display Rack"/> *Changes applied to: <order name> <input type="button" value=""/> | |



When device models (archetypes) are created using the Metadata Modeller, the rack diagram metadata must be configured separately and attached to the device archetype to describe how an instance of the device is displayed in the rack positions it occupies.

Export the Metadata Modeller Library as a ^ .JSON File

1. In the Library Details page, click the Export icon.



The library must have at least one tag to use the export feature.

Library Details - Deva_Library_1

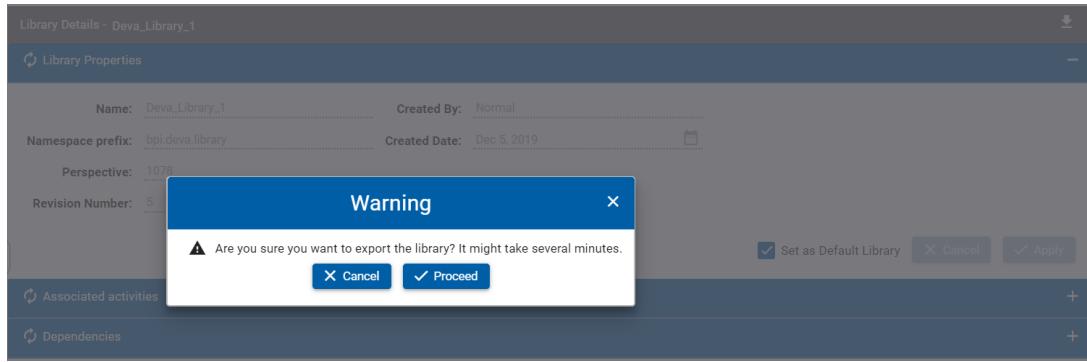
Library Properties

| | | | |
|-------------------|------------------|--|---|
| Name: | Deva_Library_1 | Created By: | Normal |
| Namespace prefix: | bpi.deva.library | Created Date: | Dec 5, 2019 <input type="button" value=""/> |
| Perspective: | 1078 | | |
| Revision Number: | 5 | <input checked="" type="checkbox"/> Set as Default Library <input type="button" value=""/> Cancel <input checked="" type="button" value=""/> Apply | |

Associated activities

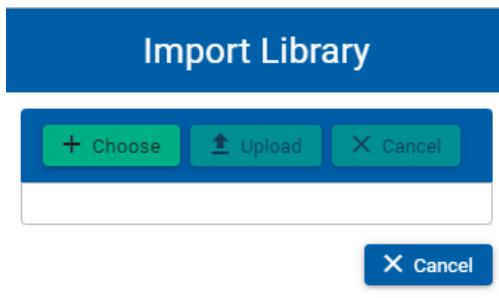
Dependencies

2. Click **Proceed** to download the Metadata Modeller information as a .json file.



Import Library

You can import a created Library from the *Import Library* page. For more information see [Import Library](#).



Library Fragments

When working with libraries in Metadata Modeller, you may want to export fragments of your library to other BPI environments without exporting the entire library. Metadata Modeller allows you to fragment your library by extracting a set of activities between two tags from your library into another library. You can add your library as a dependent to a new library and specify the point from where you want the changes from your library to apply to the new library.

In the Metadata Modeller UI, you can extract a set of activities between two tags from your library into another library by checking "Add selected tag as a dependency without applying the changes for its activities to this library":

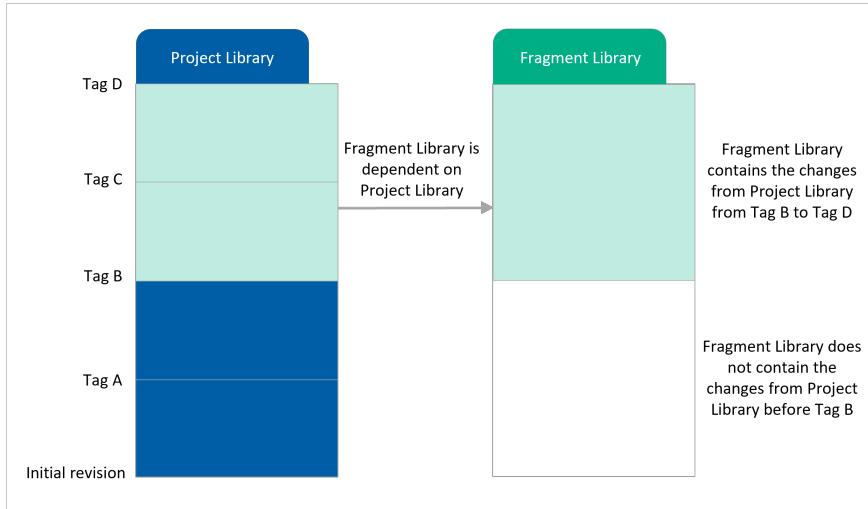
The screenshot shows the 'Dependencies' section of the Metadata Modeller UI. At the top, there is a table with columns: Library, Tag, Created Date, and Revision. The table lists several tags: Tag B (selected), Tag D, Tag C, Tag B (selected again), Tag A, Delta Tag, and Std Lib Tag. Below the table, there is a 'Select Library' dropdown set to 'Project Library', a search icon, and a 'Add Library' button. At the bottom right, there is a checkbox labeled 'Add selected tag as a dependency without applying the changes for its activities to this library', which is checked.

| | Library | Tag | Created Date | Revision |
|----|-----------------|-------------|--------------------------|----------|
| .. | Project Library | Tag B | | |
| .. | | Tag D | Mar 11, 2023, 3:17:50 PM | 490 |
| .. | | Tag C | Mar 11, 2023, 3:17:04 PM | 489 |
| .. | | Tag B | Mar 11, 2023, 3:16:19 PM | 488 |
| .. | | Tag A | Mar 11, 2023, 3:15:24 PM | 487 |
| .. | | Delta Tag | Mar 11, 2023, 3:14:06 PM | 486 |
| .. | | Std Lib Tag | Mar 11, 2023, 1:48:39 PM | 422 |

Select Library: Project Library Add selected tag as a dependency without applying the changes for its activities to this library

Working with library fragments

The following diagram describes an example of how library fragments work:



This diagram shows:

- You extract fragments of Project Library into a new library, named Fragment Library. To create the Project Library, see [Creating a Project Library](#).
- Project Library includes four tagged activities for each revision, named Tag A, Tag B, Tag C, and Tag D.
- Project Library is added as a dependency for Fragment Library.
- A fragment of Project Library is extracted from Tag B to Tag D without applying the activity changes before Tag B.
- Fragment Library contains the changes from Project Library, but only from Tag B to Tag D.

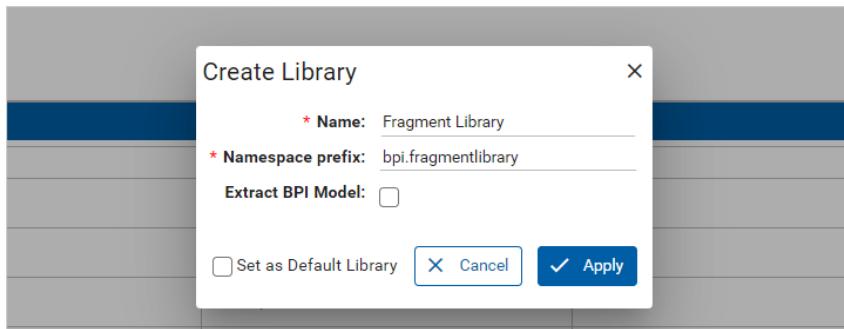
Extracting fragments of a library

This section provides the steps to extract fragments of your project library into a new library.

To extract the fragments into a new library:

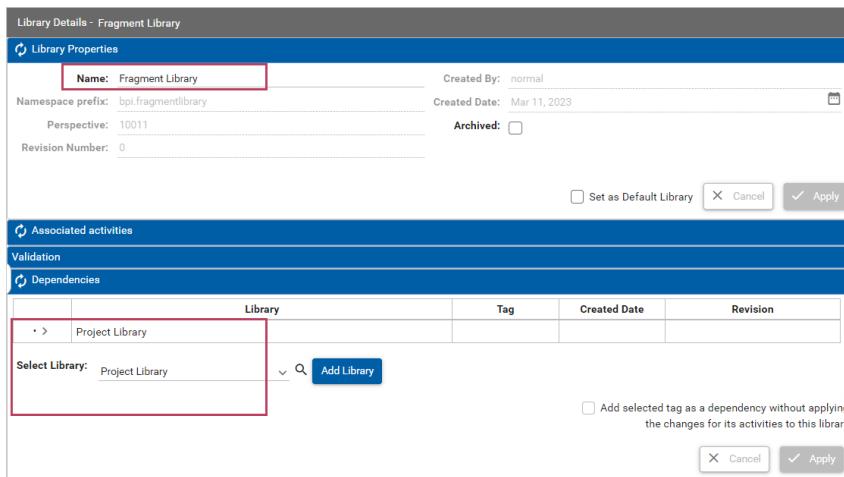
1. Go to the Metadata Modeller menu, and select **Create Library**.

In this example, you create an empty library, named **Fragment Library**:



2. Click **Apply**.

3. Add Project Library as a dependency for Fragment Library:



4. Because you have added Project Library as a dependency for Fragment Library, select a tag of Project Library to apply to Fragment Library:

| Library | Tag | Created Date | Revision |
|-----------------|-----------------------------------|--------------------------|----------|
| Project Library | <input type="radio"/> Tag D | Mar 11, 2023, 3:17:50 PM | 490 |
| .. | <input type="radio"/> Tag C | Mar 11, 2023, 3:17:04 PM | 489 |
| .. | <input type="radio"/> Tag B | Mar 11, 2023, 3:16:19 PM | 488 |
| .. | <input type="radio"/> Tag A | Mar 11, 2023, 3:15:24 PM | 487 |
| .. | <input type="radio"/> Delta Tag | Mar 11, 2023, 3:14:06 PM | 486 |
| .. | <input type="radio"/> Std Lib Tag | Mar 11, 2023, 1:48:39 PM | 422 |

Select Library: Project Library

Add selected tag as a dependency without applying the changes for its activities to this library

In this example, you want to extract a fragment of Project Library from Tag B to Tag D without applying the activity changes before Tag B.

5. Select the **Tag B** radio button.
6. Check **Add selected tag as a dependency without applying the changes for its activities to this library**:

| Library | Tag | Created Date | Revision |
|-----------------|--|--------------------------|----------|
| Project Library | <input checked="" type="radio"/> Tag B | Mar 11, 2023, 3:17:50 PM | 490 |
| .. | <input type="radio"/> Tag D | Mar 11, 2023, 3:17:04 PM | 489 |
| .. | <input type="radio"/> Tag C | Mar 11, 2023, 3:16:19 PM | 488 |
| .. | <input checked="" type="radio"/> Tag B | Mar 11, 2023, 3:15:24 PM | 487 |
| .. | <input type="radio"/> Delta Tag | Mar 11, 2023, 3:14:06 PM | 486 |
| .. | <input type="radio"/> Std Lib Tag | Mar 11, 2023, 1:48:39 PM | 422 |

Select Library: Project Library

Add selected tag as a dependency without applying the changes for its activities to this library

This adds Tag B as a dependency for Fragment Library without applying the activity changes before Tag B.

7. Click **Apply**.

You have added Tag B as a dependency, and specified the point from where you want the changes from Project Library to apply to Fragment Library:

| Library | Tag | Created Date | Revision |
|-----------------|-------------|--------------------------|----------|
| Project Library | Tag B | Mar 11, 2023, 3:16:19 PM | 488 |
| | Tag D | Mar 11, 2023, 3:17:50 PM | 490 |
| | Tag C | Mar 11, 2023, 3:17:04 PM | 489 |
| | Tag B | Mar 11, 2023, 3:16:19 PM | 488 |
| | Tag A | Mar 11, 2023, 3:15:24 PM | 487 |
| | Delta Tag | Mar 11, 2023, 3:14:06 PM | 486 |
| | Std Lib Tag | Mar 11, 2023, 1:48:39 PM | 422 |

Select Library: Project Library Add selected tag as a dependency without applying the changes for its activities to this library

8. Refresh and expand the **Associated activities** panel of Fragment Library.

The dependency on Project Library is updated to Tag B without applying the activity changes before Tag B:

| Activity Name | Created Date | Revision | Tag Name |
|--|--------------|----------|--------------------|
| Updating dependency on Project Library to tag Tag B without applying the changes | Mar 11, 2023 | 1 | Latest(Not Tagged) |

Validation

| Library | Tag | Created Date | Revision |
|-----------------|-------------|--------------------------|----------|
| Project Library | Tag B | Mar 11, 2023, 3:16:19 PM | 488 |
| | Tag D | Mar 11, 2023, 3:17:50 PM | 490 |
| | Tag C | Mar 11, 2023, 3:17:04 PM | 489 |
| | Tag B | Mar 11, 2023, 3:16:19 PM | 488 |
| | Tag A | Mar 11, 2023, 3:15:24 PM | 487 |
| | Delta Tag | Mar 11, 2023, 3:14:06 PM | 486 |
| | Std Lib Tag | Mar 11, 2023, 1:48:39 PM | 422 |

Select Library: Project Library Add selected tag as a dependency without applying the changes for its activities to this library

9. Because you want to apply the activity changes from Tag B to Tag D, select the **Tag D** radio button:

| Library | Tag | Created Date | Revision |
|-----------------|-------------|--------------------------|----------|
| Project Library | Tag D | Mar 11, 2023, 3:16:19 PM | 488 |
| | Tag D | Mar 11, 2023, 3:17:50 PM | 490 |
| | Tag C | Mar 11, 2023, 3:17:04 PM | 489 |
| | Tag B | Mar 11, 2023, 3:16:19 PM | 488 |
| | Tag A | Mar 11, 2023, 3:15:24 PM | 487 |
| | Delta Tag | Mar 11, 2023, 3:14:06 PM | 486 |
| | Std Lib Tag | Mar 11, 2023, 1:48:39 PM | 422 |

Select Library: Project Library Add selected tag as a dependency without applying the changes for its activities to this library



The checkbox is unchecked because you want to extract Project Library from Tag B to Tag D by applying the activity changes.

10. Click **Apply**.

11. Refresh and expand the **Associated activities** panel of Fragment Library.

The dependency on Project Library is updated to Tag D:

| Associated activities | | | | |
|-----------------------|--|--------------|----------|--------------------|
| | Activity Name | Created Date | Revision | Tag Name |
| .. | Updating dependency on Project Library to tag Tag D for library related nodes | Mar 11, 2023 | 3 | Latest(Not Tagged) |
| .. | Updating dependency on Project Library to tag Tag D - 0 | Mar 11, 2023 | 2 | |
| .. | Updating dependency on Project Library to tag Tag B without applying the changes | Mar 11, 2023 | 1 | |

| Validation | | | | |
|--------------|-----------------|-------|--------------------------|----------|
| Dependencies | | | | |
| | Library | Tag | Created Date | Revision |
| .. > | Project Library | Tag D | Mar 11, 2023, 3:17:50 PM | 490 |

Select Library: Project Library

You have extracted a set of changes between two tags (Tag B to Tag D) from Project Library into Fragment Library.

You can now export the Fragment Library.

12. Tag the latest activity, and then export the Fragment Library:

| Associated activities | | | | |
|-----------------------|--|--------------|----------|----------|
| | Activity Name | Created Date | Revision | Tag Name |
| .. | Updating dependency on Project Library to tag Tag D for library related nodes | Mar 11, 2023 | 3 | Tag 1 |
| .. | Updating dependency on Project Library to tag Tag D - 0 | Mar 11, 2023 | 2 | |
| .. | Updating dependency on Project Library to tag Tag B without applying the changes | Mar 11, 2023 | 1 | |



You can use fragments to move specific changes from one environment to the other. You cannot use a fragment library on its own to create your models.

Adding a fragment to another library

This section describes the steps to add a fragment to another library.

To add a fragment:

1. Create a library using the steps in [Creating a Project Library](#).
2. Import the fragment library.
3. Add the fragment library as a dependency in your library.



To add a fragment to another library, it's not mandatory to create a new library every time. If you have a library already created using the steps in [Creating a Project Library](#), you can just import the fragment library, and add the fragment library as a dependency in your library.

Metamodel Management

The **Metamodel Management** page allows you to manage a metamodel and define the relationship between metamodels.

Accessing the Metamodel Management Page

1. Go to the Main Menu.
2. Select **Metadata Modeller > Metamodel Management**.

The Metamodel Management page opens.

The screenshot shows the Metamodel Management interface. At the top, there's a navigation bar with 'My Orders' and 'Metamodel Management'. Below it is a 'Create Metamodel' section titled 'Add new Metamodel' containing a table with five rows of metamodel definitions. The columns are 'Display Relationships', 'Type', 'Metamodel Name', 'Preferred DnId', and 'Actions'. The rows show relationships like 'Equipment' to 'Rack', 'Address' to 'RackPosition', 'Equipment' to 'Shelf', 'Address' to 'ShelfPosition', and 'Address' to 'SlotPosition'. Below this is a 'MetaModel Relationships' section titled 'Add new Relationship' with a table showing 'No records found'. Both sections have 'Cancel' and 'Apply' buttons at the bottom right. The footer of the page includes copyright information: 'Copyright ©Blue Planet Software, Inc. 2020. All rights reserved.'

This feature consists of two sections:

- [Create Metamodel](#)
- [MetaModel Relationships](#)

Create Metamodel

Use the *Create Metamodel* section to create, read, delete, and modify the metamodel. There are various column headers available in this section, including:

Create Metamodel fields

| FIELD | DESCRIPTION |
|-----------------------|---|
| Display Relationships | Click the radio button to display the relationships of the selected metamodel in the MetaModel Relationships section. |
| Type | Select the type from the drop-down list to define a type. |
| Metamodel Name | Defines the name of a metamodel. |
| Preferred Drnild | Defines the id, which is used during deployment and become drnild of a real Metamodel. |
| Actions | Enables you to delete a row. You can undo the deletion by clicking the Undo icon. |



For more information about metamodel deployment, see [Deploying the newly added metamodel and its relationship](#).

MetaModel Relationships

The *MetaModel Relationships* section allows you to add, delete, and modify the relationships between metamodels.

When you click the radio button in the Create Metamodel section, the relationships of the selected metamodel is displayed in this section. There are various column headers available in this section, including:

MetaModel Relationships fields

| FIELD | DESCRIPTION |
|--------|--|
| Source | Defines the source metamodel. Automatically generates the metamodel name from the Create Metamodel section. For example, if the Metamodel Name is "card", then that name is populated in this column. |

| FIELD | DESCRIPTION |
|--------------------|--|
| Direction | <p>Defines the direction of relationship between the source and target.</p> <p>Select the direction from the drop-down list, which includes the right and left arrow, and the self-relation icon. For example, if you select the right arrow, then the relationship is defined from source to target.</p> |
| Target | Select the metamodel from the drop-down list to define the target metamodel. |
| Relationship Types | <p>Select the checkbox from the drop-down list to define the type of relationship between the source and target. For example, if you want to build a relationship between the source and target based on "IP range", then select the "IP range" checkbox from the drop-down list.</p> <p>If you want to add a new relation type:</p> <ol style="list-style-type: none"> 1. Click Add new relationship types (+) in the Relationship Types column. 2. Define the name in Add New Relation dialog box. 3. Click Apply. |
| Actions | Enables you to delete a row. You can undo the deletion by clicking the Undo icon. |

Adding a new metamodel

To add a new metamodel:

1. Create a new library and set that as the default library. For more information, see [Create Library](#).

If you want to add a new metamodel in the existing library, then set that library as the default library.

2. Add the dependency library. For more information, see [Adding Library Dependency](#).



Click the radio button of standard Metadata in the Tag column to add standard metadata library into the library. Adding standard metadata library allows you to create the equipment and set its compatibility.

3. Launch the Metamodel Management page to add a new metamodel. For more information, see

Accessing the Metamodel Management Page.

- In the [Create Metamodel](#) section, click **Add new Metamodel**.
- Define the Type, Metamodel Name, and Preferred DrnId.

See the [Create Metamodel fields](#) table as a guide. The below figure shows an example of a metamodel that has been added.

| Display Relationships | Type | Metamodel Name | Preferred DrnId | Actions |
|----------------------------------|--------------------|----------------------|-----------------------|---------|
| <input checked="" type="radio"/> | * Equipment | * DocCard | * 123456789 | |
| <input type="radio"/> | * Please Select... | * | * | |
| <input type="radio"/> | * Activity | * | * 121 | |
| <input type="radio"/> | * Equipment | * Card | * 11011 | |
| <input type="radio"/> | * Abstraction | * Contact | * 202 | |
| <input type="radio"/> | * Abstraction | * Customer | * 119 | |
| <input type="radio"/> | * Equipment | * Device | * 108 | |
| <input type="radio"/> | * Abstraction | * Difference | * 127 | |
| <input type="radio"/> | * Connection | * InternalConnection | * 1232112345678123456 | |
| <input type="radio"/> | * Abstraction | * Library | * 126 | |

| < 1 2 3 > |

Default Library: phase2lib

Cancel Apply

Copyright ©Blue Planet Software, Inc. 2020. All rights reserved.

- Click **Apply**.

Removing a metamodel

To remove a metamodel:

- Launch the Metamodel Management page and go to the [Create Metamodel](#) section. For more information, see [Accessing the Metamodel Management](#).
- Click the radio button in the Display Relationships column for the metamodel you want to remove.
- In the Actions column, click the delete icon.
- Click **Apply**.

Modifying a metamodel

To modify a metamodel:

1. Launch the Metamodel Management page and go to the [Create Metamodel](#) section. For more information, see [Accessing the Metamodel Management](#).
2. Click the radio button in the Display Relationships column for the metamodel you want to modify
3. Edit the name in the Metamodel Name column.

You can also modify the information in the Preferred Drnild column.

4. Click **Apply**.

Adding a new metamodel relationship

The following procedure explains how to add a new metamodel relationship:

1. Launch the Metamodel Management page. For more information, see [Accessing the Metamodel Management](#).
2. In the Display Relationships column, click the radio button for the metamodel you want the relationships to be displayed in the [MetaModel Relationships](#) section:
 - a. Click **Add new Relationship**.
 - b. Define the Source, Direction, Target, and Relationship Types.

See the [MetaModel Relationships fields](#) table as a guide. The below figure shows an example of a metamodel relationship that has been added.



You cannot add the same relationship which is already defined.

The screenshot shows two main sections of the Metamodel Management interface. The top section is a table listing various metamodel entities. A red arrow points from the 'Source' field in the relationship editor below to the 'DocCard' row in the list above. The bottom section is a 'MetaModel Relationships' editor with fields for Source, Direction, Target, Relationship Types, and Actions. It shows a single relationship entry: DocCard to Device.

| | Abstraction | Customer | 119 |
|----------------------------------|-------------|--------------------|---------------------|
| ○ | Equipment | Device | 108 |
| ○ | Abstraction | Difference | 127 |
| <input checked="" type="radio"/> | Equipment | DocCard | 123456789 |
| ○ | Connection | InternalConnection | 1232112345678123456 |
| ○ | Abstraction | Library | 126 |
| ○ | Address | Location | 101178888789787878 |

| Source | Direction | Target | Relationship Types | Actions |
|---------|-----------|--------|--------------------|---------------------------------------|
| DocCard | → | Device | 4 items selected | <input type="button" value="Delete"/> |

c. Click **Apply**.

Removing a metamodel relationship

To remove a metamodel relationship:

1. Launch the Metamodel Management page and go to the [MetaModel Relationships](#) section. For more information, see [Accessing the Metamodel Management](#).
2. In the Actions column, click the delete icon to remove the specific metamodel relationship.
3. Click **Apply**.

Deploying the newly added metamodel and its relationship

After adding a new metamodel and its relationship, you can deploy it to Blue Planet Inventory:

1. Right-click the default library on the bottom right-hand side of the page.

The screenshot shows the 'Create Metamodel' section of the Metamodel Management page. A table lists ten metamodel entries, each with a radio button, a type (Activity, Equipment, Abstraction, Address), a name field (e.g., *Activity, *Card, *Contact, *Customer, *Device, *Difference, *DocCard, *InternalConnection, *Library, *Location), and a preferred DrnId field. The last entry, *Location, has its DrnId set to *123456789. A context menu is open over this entry, containing options: *1232112345678123456, *126, Library Details, History, and ObjectId and ObjectClassId to Clipboard. At the bottom right of the menu, it says 'Default Library: phase2lib'. Below the table are navigation buttons (back, forward, 1, 2, 3, >, >>).

| Display Relationships | Type | Metamodel Name | Preferred DrnId | Actions |
|----------------------------------|---------------|---------------------|-----------------------|--|
| <input checked="" type="radio"/> | * Activity | *Activity | * 121 | █ |
| <input type="radio"/> | * Equipment | *Card | * 11011 | █ |
| <input type="radio"/> | * Abstraction | *Contact | * 202 | █ |
| <input type="radio"/> | * Abstraction | *Customer | * 119 | █ |
| <input type="radio"/> | * Equipment | *Device | * 108 | █ |
| <input type="radio"/> | * Abstraction | *Difference | * 127 | █ |
| <input type="radio"/> | * Equipment | *DocCard | * 123456789 | █ |
| <input type="radio"/> | * Connection | *InternalConnection | * 1232112345678123456 | █ |
| <input type="radio"/> | * Abstraction | *Library | * 126 | █ |
| <input type="radio"/> | * Address | *Location | * 1011 | █ |

2. Select **Library Details**.



You can also launch the Library Details from the Libraries tree. See [Library Details](#).

3. In the Associated activities section of the Library Details page, perform the following steps:
 - a. Click **Validate** to see the report of a library in the Validation section to check if deployments impact existing models.
 - b. Click **Tag**.

The screenshot shows the 'Library Details - phase2lib' page in the Metamodel Management section. The 'Associated activities' section contains a table with the following data:

| | Activity Name | Created Date | Revision Number | Tag Name | |
|-----|---|--------------|-----------------|--------------------|-------------------------------|
| • > | Managing XMetamodel Relationships in library phase2lib By User Normalon Apr 07,2020 19:46 | Apr 8, 2020 | 3 | Latest(Not Tagged) | Tag Validate |

Below the table, there are sections for 'Validation' and 'Dependencies'. The 'Dependencies' section shows a table with the following data:

| | Library | Tag | CreatedDate | Revision |
|-----|---|-------------------|---------------------------|---------------|
| • > | Packaging Standard Metadata_Library - bpi:standard:metadata | Standard Metadata | Feb 24, 2020, 11:05:00 PM | 1582565641929 |

At the bottom, there is a 'Select Library:' dropdown and an 'Add Library' button.

- c. Define the tag name in **Add Tag Name** dialog box and click **Apply**.
- d. Click **Deploy** to deploy the metamodel to Blue Planet Inventory, and then click **Yes** to confirm the deployment.

The screenshot shows the 'Library Details - phase2lib' page. At the top, there are fields for Name (phase2lib), Namespace prefix (phase2-check), Perspective (1140), and Revision Number (3). Below these are buttons for 'Set as Default Library', 'Cancel', and 'Apply'. The main content area is titled 'Associated activities' and contains a table with one row:

| | Activity Name | Created Date | Revision Number | Tag Name | |
|-----|--|--------------|-----------------|----------|--|
| • > | Managing XMetamodel Relationships in library phase2lib By User Normalon Apr 07, 2020 19:46 | Apr 8, 2020 | 3 | Tag1 | Deploy Validate |

Below this are sections for 'Validation' and 'Dependencies'.

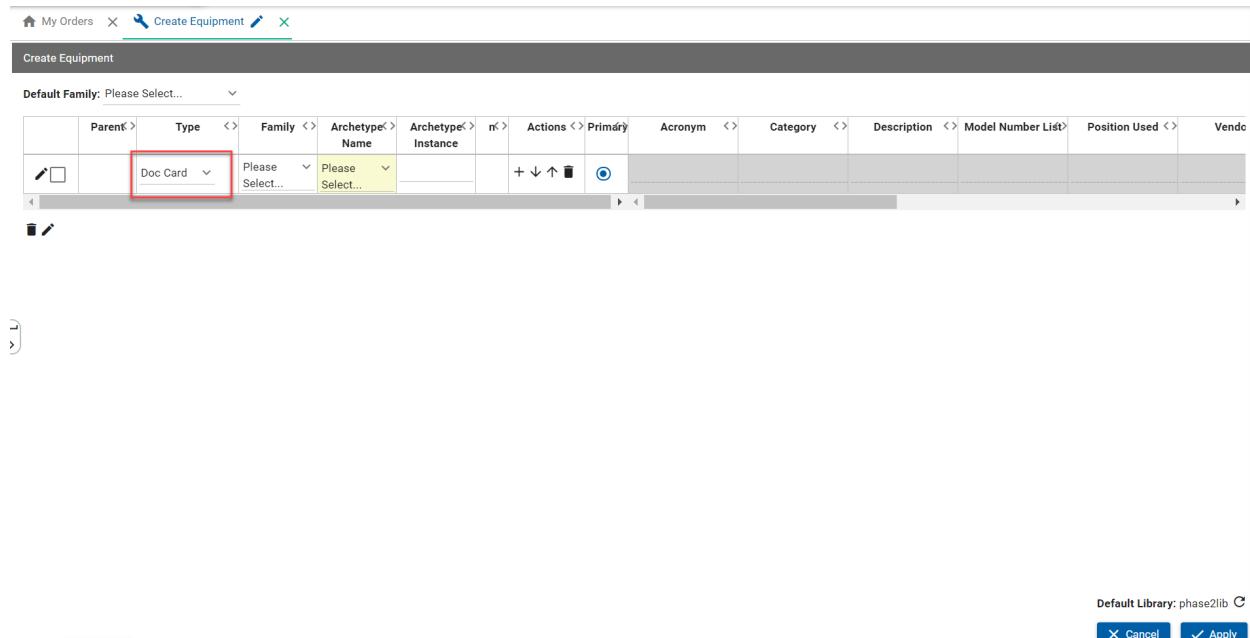
The newly added metamodel and its relationship is now deployed to Blue Planet Inventory. For more examples on deploying to Blue Planet Inventory, see [Deploying created equipment](#).

Using the newly added metamodel

After adding the new metamodel, it is available on the Create Equipment page before deployment. It becomes available in the Blue Planet Inventory only after the deployment. To see the added metamodel:

1. Launch the Create Equipment page, which is available before deployment. For more information, see [Create Equipment](#).
2. Select the newly added metamodel from the drop-down list in the Type column.

The following figure shows an example of the newly added metamodel:



Metadata Permission

The **Metadata Permission** page is where you configure the permissions for metadata and these permissions can be deployed to live via standard deployment process. This page displays metamodels and archetypes in a hierarchical format. You can set the permissions only for archetypes, such as device, network, connection, card, and rack.

Accessing the Metadata Permission Page

1. Go to the Main Menu.
2. Select **Metadata Modeller > Metadata Permission**.

The Metadata Permission page opens.

The screenshot shows the 'Metadata Permission' page. At the top, there are navigation links: 'My Orders' and 'Metadata Permission'. Below the header, a sub-header reads 'Metadata Permission'. A dropdown menu labeled 'Default Family: Please select...' is open, showing a list of metadata archetypes. The list includes: Rack, LogicalPosition, PhysicalTerminationPosition, Order, Plan, PhysicalConnection, Device, SlotPosition, LogicalInterface, RackPosition, Location, and DocCard. To the right of the list is a 'Create Permission' button. At the bottom of the page, there is a 'Default Library: phase2lib' dropdown, a 'Cancel' button, and an 'Apply' button.

This page provides summarized information for the added metadata and permissions.

Creating permissions for metadata

Permissions are created to make available only for a particular user. For example, if you created permission of a rack type for the User1, then that rack type will not be available for User2, whereas it is available for the User1.

See [Create Equipment](#) as a guide to create a new equipment. The following figure is an example showing the newly created equipment.

| Parent | Type | Family | Archetype Name | Archetype Instance | Actions | Primary Position Used | Vendor | Width In Inches | Position Sequence | Part Number | Height In Inches |
|--------------------------|------|-----------|----------------|--------------------|---------|-----------------------|--------|-----------------|-------------------|-------------|------------------|
| <input type="checkbox"/> | Rack | No Family | rack1 | | + ↴ ↑ ↵ | family | 3 | | | | |

To create a permission for the newly added equipment:

1. On the Metadata Permission page, select the family name from the Default Family drop-down list.



Select the same family name that you used while creating the equipment in the Create Equipment Page.

2. Select the newly created equipment checkbox in the Metadata column.



You can select multiple checkboxes to provide bulk permissions.

3. Define the Activity type in the Activity Type field on the bottom-left hand side.

4. Click **Apply**.

The following figure is an example to create the permission.

The screenshot shows the 'Metadata Permission' interface. At the top, it says 'Default Family: No Family'. Below is a table with two columns: 'Metadata' and 'Create Permission'. In the 'Metadata' column, there is a tree view of equipment types. A checkbox for 'rack1' under the 'Rack' category is checked and highlighted with a red border. Other items in the tree include LogicalPosition, PhysicalTerminationPosition, Order, Plan, PhysicalConnection, Device, Shelf, RackPosition, Location, DocCard, and position. At the bottom left, it says 'Activity Type : DocApprove'. On the right, it says 'Default Library: phase2lib' with a refresh icon. Below that are 'Cancel' and 'Apply' buttons.

5. Deploy the newly permitted equipment to Blue Planet Inventory. For more information, see [Deploying the newly added metamodel and its relationship](#).

Removing permission for metadata

To remove a permission for the equipment:

1. On the Metadata Permission page, select the family name from the Default Family drop-down list.



Select the same family name that you used while creating the equipment in the Create Equipment Page.

2. Select the equipment checkbox in the Metadata column.



You can select multiple checkboxes to provide bulk permissions.

3. Click **Apply**.
4. Deploy the removed permission(s) from the Blue Planet Inventory. For more information, see [Deploying the newly added metamodel and its relationship](#).

Using the created equipment

After deploying the created equipment, it is available in Blue Planet Inventory. To use the equipment:

1. In the Locations tree, right-click a building, room, or customer premise.

The Context Menu opens.

2. Select **Operations > Create Rack**.
3. Select the newly created equipment from the Rack Type drop-down list.

The following figure is an example of the newly created equipment after providing permissions:

Create Rack

*Rack Type: Notes:

*Rack Name:

Model Number:

*Location:

Aisle:

Bay:

Equipment Diagram Editor

The *Equipment Diagram Editor* helps you to create a schematic diagram of a device to make it look like a real device. Using this editor, you can design, add color, size and position a label, and set rectangle shape for devices, shelves, slots, cards, pluggables, and ports.

The Equipment Diagram Editor:

- Controls the presentation of the equipment diagram, which displays SVG graphics for devices, shelves, slots, cards, pluggables, and ports.
- Provides a graphical interface for the real-time design of the equipment diagram.

Accessing the Equipment Diagram Editor

This section describes how to access the Equipment Diagram Editor.

To access the Equipment Diagram Editor:

1. Set a library as a default library. For more information, see [default library](#).
2. In the Metadata Modeller tree, select the default library, and then navigate to the specific equipment.

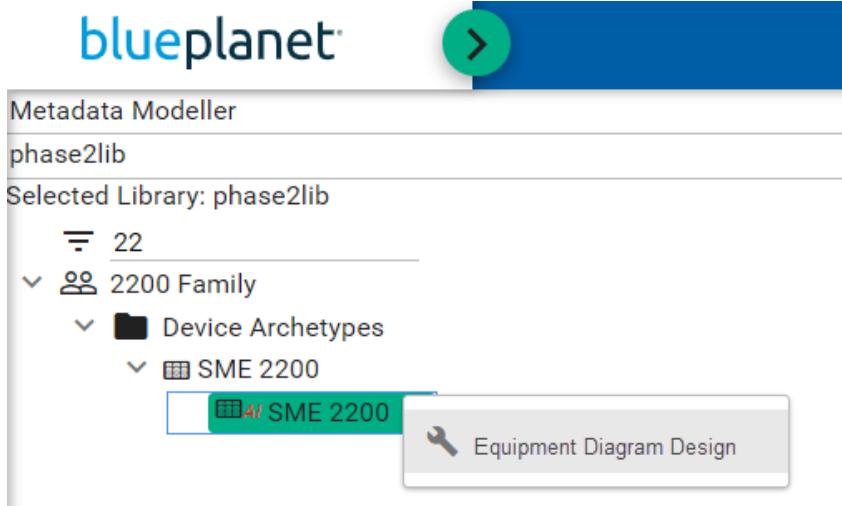


To create and define a new equipment, see the [Create Equipment](#) section.

3. Right-click the equipment (archetype instance).

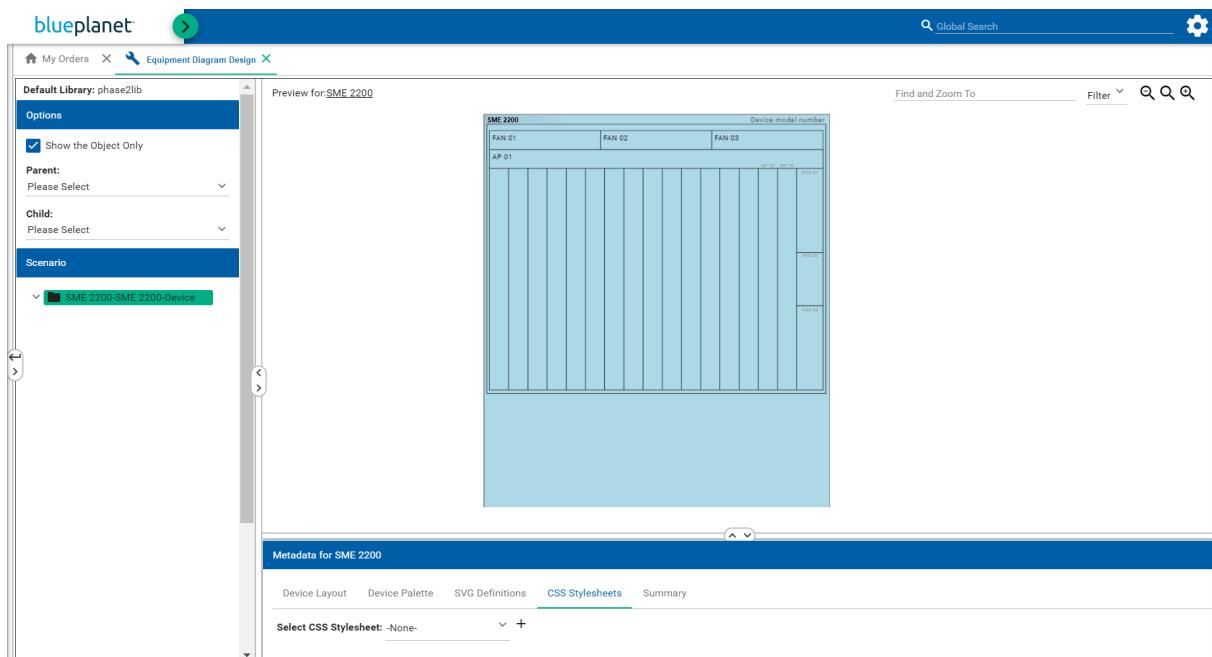


The equipment used to launch the Equipment Diagram Editor is termed as the *target*.



4. Select Equipment Diagram Design.

Displays the Equipment Diagram Editor page.

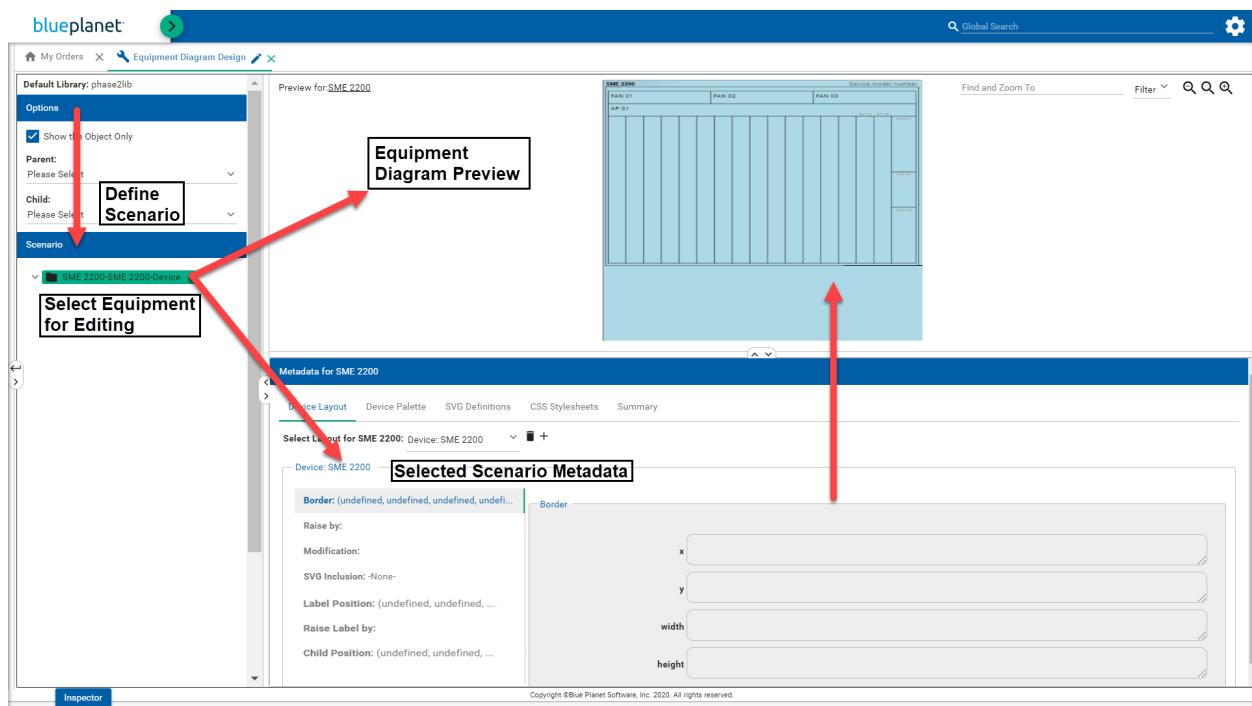


Structure of the Equipment Diagram Editor

The Equipment Diagram Editor page includes the following sections:

- Options
- Scenario

- Preview
- Metadata



Options

Use the *Options* section to define the scenario for the diagram.

This section includes the following fields:

Options fields

| FIELD | DESCRIPTION |
|----------------------|--|
| Show the Object Only | Select the checkbox to remove parent or child relationships and display the scenario based on the target . |
| Parent | <p>Defines the parent equipment of the target.</p> <ul style="list-style-type: none"> • Parent Positions: This field appears when you define the Parent. Use this field to define the position within the parent where the target should appear in the diagram. |

| FIELD | DESCRIPTION |
|-------|--|
| Child | <p>Defines the child equipment that will be inserted in the target.</p> <ul style="list-style-type: none"> • Child Positions: This field appears when you define the Child. Use this field to define the position of the child equipment within the target. ◦ Count of Copies: This field appears when you define the Child Positions. Use this field to create multiple instances of the child for the selected Child Positions. |

Scenario

After defining the *Options* section, the equipment scenario is displayed in the *Scenario* section, which is automatically generated from the equipment compatibility metadata. The *Scenario* is displayed as an expandable tree.

The *Preview* and *Metadata* sections are updated when you select an equipment in the scenario tree:

- The *Preview* section displays the diagram using the latest metadata rules.
- The *Metadata* section provides the metadata for the selected equipment for editing.

Preview

This section shows the preview of the equipment diagram. The diagram is generated based on your selection in the *Scenario* and *Metadata* sections.

For example:

- When you select an equipment in the *Scenario* section, the diagram of that equipment is displayed.
- When you modify any metadata information in the *Metadata* section, the diagram updates accordingly.

This section includes the following fields:

Preview fields

| FEATURE | DESCRIPTION |
|--|--|
| Title (auto-populated title with "Preview for:") | Automatically displays the title (name of the equipment) based on your selection in the Scenario section. |
| Find and Zoom To | Enables you to search for specific equipment in the diagram. The equipment matching the search string is highlighted as specified in the CCS highlighted style. For more information, see CSS Stylesheets . |
| Zoom In or Out | Enables you to zoom in or out the diagram. |
| Reset Zoom | Enables you to reset the zoom level in the diagram. |
| Filter | Use this option to show/hide the group name in the diagram. For example, if you have defined the Group Name in SVG Inclusion in the Layout tab, this name will appear in the Filter drop-down list. Select or clear the checkbox to show or hide the name. |



Press the Alt key on the keyboard to use the magnifier widget to see an enlarged view of the displayed equipment diagram.

Metadata

The **Metadata** section allows you to create, modify, or delete the metadata that controls the equipment diagram. The Preview of the diagram is generated based on the metadata.

In this section, you can customize the metadata of the equipment using the following tabs:

Metadata tabs

| TAB | DESCRIPTION |
|---------------------------------|--|
| Layout | Defines the layout of the selected equipment, such as the height, width, and label positions. |
| Palette | Defines the equipment style, such as colors, shading, line effects, fonts, and transparencies. |
| SVG Definitions | Provides the definition of SVG fragments, which you can use to customize the diagram. |

| TAB | DESCRIPTION |
|-----------------|--|
| CSS Stylesheets | Defines the CSS stylesheets to style the diagram. |
| Summary | Shows all the modifications done and allows you to apply or cancel them. |



Based on the selected equipment in the Scenario section, a prefix is added to the Layout and Palette tabs. For example, if you select a shelf, the tab name changes to Shelf Layout and Shelf Palette.

Structure of the Tabbed Panes

The following figure and table describes the general structure of the tabbed panes in the *Metadata* section.

The screenshot shows the 'Metadata for SME 2200' interface with the 'CSS Stylesheets' tab selected. The 'Editable Name' field contains 'Default Styles'. The 'Selector' dropdown menu is open, showing options like '-None-', '2200 Styles', and 'Default Styles', with 'Default Styles' currently selected. A warning message 'Warning: This Layout is used in other places.' is displayed next to the '+' button. The code editor below shows CSS rules such as 'text.nested-metamodel-112-label { font-size: 3px; }' and 'rect.blue-channel { fill: #CCE6FF }'.

General structure of the tabbed panes

| TAB | DESCRIPTION |
|---------------|---|
| Selector | Enables you to select the metadata from the drop-down. |
| Editable Name | Displays the name of the metadata. Click the name to edit it. |
| Delete icon | Deletes the metadata. |
| Add icon | Adds new metadata. |

| TAB | DESCRIPTION |
|-----------------|---|
| Warning message | Displayed when a selected Layout or Palette is used by other equipment. |
| Form | Displays the selected metadata for editing. The Form varies for each tab. |

Diagram Concepts

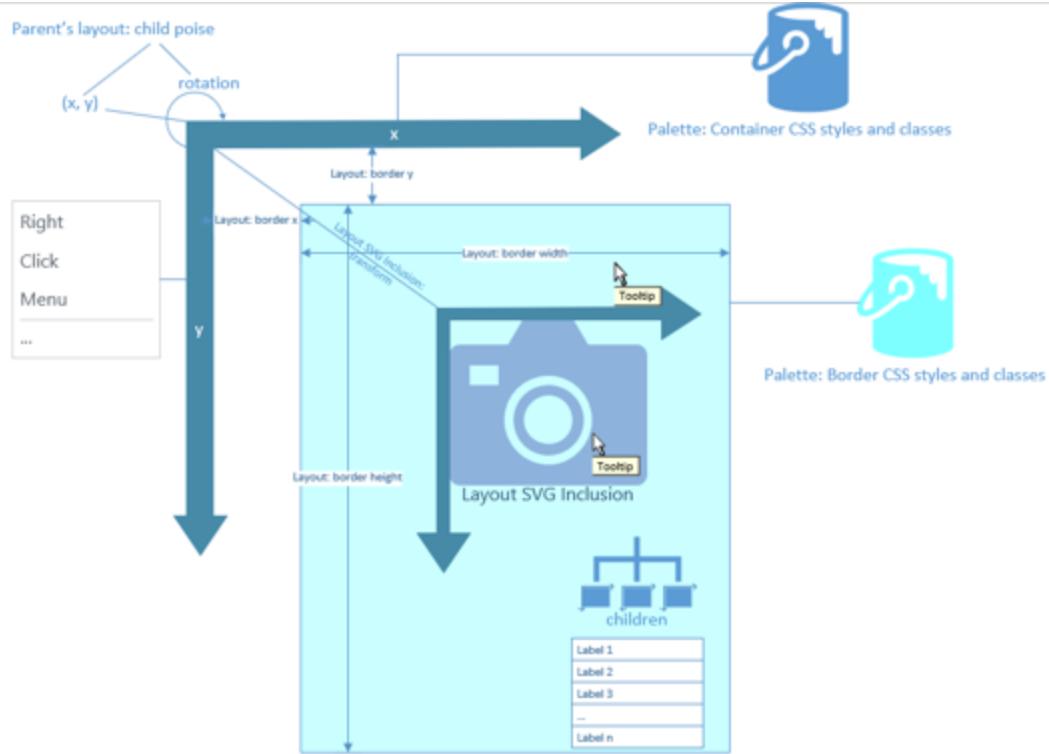
To configure an equipment diagram correctly, you must have a basic understanding of SVG and CSS.

The equipment diagram has a nested SVG structure that corresponds to the containment structure of the equipment that it represents. In the Equipment Diagram Editor, that containment structure is termed as the *Scenario*, which you can define in the *Options* section. The Equipment Diagram Editor retrieves the containment structure of the target equipment from the database.

The layout rules that you define in the Layout tab in the *Metadata* section controls the display of an SVG element. This tab also defines the position of its child elements and CSS styles that are applied to each element.

An SVG is generated for each piece of equipment. An <svg:g> element is created to contain the entity's SVG and its child elements. The following steps are then added to this element:

1. A transform attribute to define a coordinate space for this element.
2. The attributes required to enable the right-click menu for the equipment.
3. A tooltip.
4. CSS styles and classes, as defined in metadata, as well as specific CSS classes of the form 'nested-metamodel-<metamodel type name>' and 'nested-archetype-< BPI UID>'.
5. An <svg:rect> element, which shows the equipment's extent.
6. If an **SVG Inclusion** is configured in metadata for the equipment, it is included as a transformed <svg:use> element with a tooltip.
7. These preceding steps are invoked recursively to generate the SVG for any child entities.
8. Any labels that require clipping are added as <svg:text> elements within an <svg:svg> element that causes the text to be clipped to the extent of the <svg:rect>.
9. Any labels that do not require clipping are added as <svg:text> elements.



Note that the preceding order is significant as later SVG elements are painted on top of earlier ones. For example, a node's labels will be painted over its [SVG Inclusion](#) or any of its child elements. Those child elements are also added within the `<svg:g>` elements of their parents. This implies that:

- The coordinate space of a child element is defined relative to its parent's coordinate space. For example, if a card is rotated, all of its contents are moved and rotated appropriately.
- CSS selectors can take advantage of the nesting structure. For example, a CSS selector of `'.nested-metamodel-device > .nested-metamodel-physicalport'` will only apply styles to physical ports that are the immediate child element of devices.

Layout

Use this tab to create or customize the layout of the selected equipment in the equipment diagram.

The screenshot shows the 'Metadata for SME 2200' tab in the Equipment Diagram Editor. The 'Device Layout' tab is active. The interface includes a navigation bar with links like 'Device Layout', 'Device Palette', 'SVG Definitions', 'CSS Stylesheets', and 'Summary'. Below the navigation bar, there's a dropdown labeled 'Select Layout for SME 2200: Device: SME 2200' with a dropdown arrow, a minus sign, and a plus sign. The main content area is titled 'Device: SME 2200'. It contains several configuration sections:

- Border:** (undefined, undefined, undefined, undefined) - This section contains input fields for 'x', 'y', 'width', and 'height'.
- Raise by:**
- Modification:**
- SVG Inclusion:** -None-
- Label Position:** (undefined, undefined, undefined)
- Raise Label by:**
- Child Position:** (undefined, undefined, undefined)

This tab includes the following fields:

Layout fields

| FIELD | DESCRIPTION |
|--------------|--|
| Border | <p>Determines the position and size of the rectangle that represents the equipment in the diagram:</p> <ul style="list-style-type: none"> • x & y: Defines the offset of the top-left corner of the rectangle, relative to where the equipment's parent SVG element is placed. • width: Defines the rectangle's width. • height: Defines the rectangle's height. |
| Raise by | See Raise by . |
| Modification | Allows you to customize the equipment's data before the diagram is previewed. For example, adding or deleting labels or modifying attributes. |

| FIELD | DESCRIPTION |
|----------------|---|
| SVG Inclusion | <p>Identifies whether an SVG definition (typically a realistic-looking SVG image) is displayed for the specific equipment.</p> <p>SVG Definition: Select the SVG Definition from the drop-down list. You can use SVG Definitions in multiple layouts.</p> <p> The <i>SVG Definitions</i> tab also manages SVG Definitions.</p> <p>When you select the SVG definition from the drop-down list, the below options are displayed:</p> <ul style="list-style-type: none"> • Transform: Add the SVG syntax to scale or rotate the size. For example, <code>scale(0.1) translate(15,5) rotate(45)</code>. • Group Name: You can define the name of the group for the selected SVG. <p>If a group name is defined, this will appear in the Filter drop-down in the <i>Preview</i> section. For more information, see Filter.</p> |
| Label Position | <p>Defines the position of the equipment's label. This field includes the following options:</p> <ul style="list-style-type: none"> • x & y: Defines the horizontal and vertical position of the label, relative to the equipment's SVG element. • rotation: Defines the clockwise rotation of the label in degrees. For example, if you want to rotate the label to 90 degree clockwise, then define the rotation field to 90. |
| Raise Label by | See Raise by . |
| Child Position | <p>Defines the position of a child equipment.</p> <p>This field includes similar options like Label Position and can be defined in the same way.</p> |

It is recommended to use the same scale for the layout of all equipment diagrams. Failure to do so may result in diagrams that will look incorrect because different components are drawn at different scales. A scale of one pixel per mm of physical equipment is recommended. Adopting this simple scale has several

advantages:

- It avoids mistakes due to conversion calculations.
- It means that your diagrams will be the same scale as existing models that use this standard.
- It provides a reasonable balance between resolution and file size for most diagrams when they are exported into a raster format, such as .png.

SVG Inclusions (see below) must be scaled to match your chosen scale.

Layout Expressions

You also have the option to provide JavaScript expressions for the following Layout fields:

Layout expressions

| FIELD | AVAILABLE VARIABLES | EXPRESSION RESULT TYPE | EXAMPLE |
|---------------|--|------------------------|--|
| Border x | children: Equipment[] self: Equipment | Number | 64 |
| Border y | | | |
| Border width | | | |
| Border height | | | |
| Raise by | | | |
| Modification | children: Equipment[] self: Equipment | None | <pre>{self.addLabel((self.placeNumber % 2) == 0 ? 'Out' : 'In', 'Channel Directions')}</pre> |

| FIELD | AVAILABLE VARIABLES | EXPRESSION RESULT TYPE | EXAMPLE |
|---|---|------------------------|---|
| Label Position x Label Position y Label Position rotation Raise Label by | i: number (see Note) self: Label priorSibling: Label priorX: number priorY: number v: Equipment | Number | <pre>self.groupName== 'Model Numbers' ? 243 : 2</pre> |
| Child Position x Child Position y Child Position rotation | n: number (see Note) self: Equipment priorSibling: Equipment i: number (see Note) | Number | <pre>4.5 + 6.75 * Math.trunc((n+3) /6)</pre> |



The variable "i" is the zero-based index of a label or piece of equipment among its child elements. The variable "n" is the equipment's place number (port number, slot position number, etc.). It is an alias for self.placeNumber.

The value of "i" for a piece of equipment will change if its child elements are inserted or removed before it. Therefore, prefer using "n" to calculate the layout values rather than "i".

Expression Data Types

The Equipment data type includes the following members:

Expression data types

| MEMBER | DESCRIPTION |
|-------------|--|
| placeNumber | A numerical field indicating the equipment's place number (port number, slot position number, etc.). This field is not editable. |

| MEMBER | DESCRIPTION |
|---|--|
| addLabel(text: string, groupName: string = null, id: string = null) | <p>A method to add a label to the equipment.</p> <p>(Optional) You can provide a group name to labels to enable the filter option in the diagram. You can also provide an "id" to them to easily identify by Layout or Palette tabs.</p> |

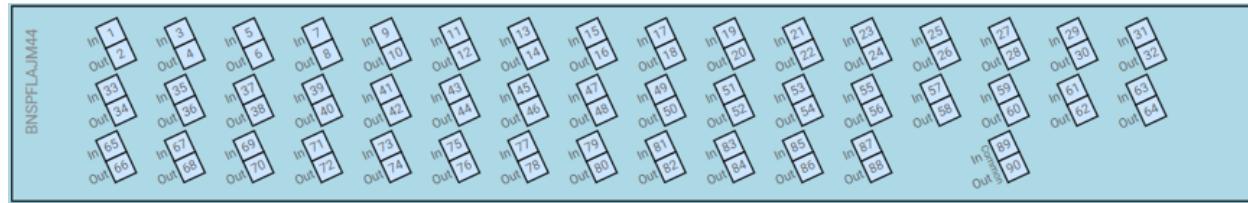
The Label data type includes the following read-only fields:

Label data types

| FIELD | DESCRIPTION |
|-----------|--------------------------------------|
| text | Defines the text of the label. |
| groupName | Defines the group name of the label. |
| id | Defines the id of the label. |

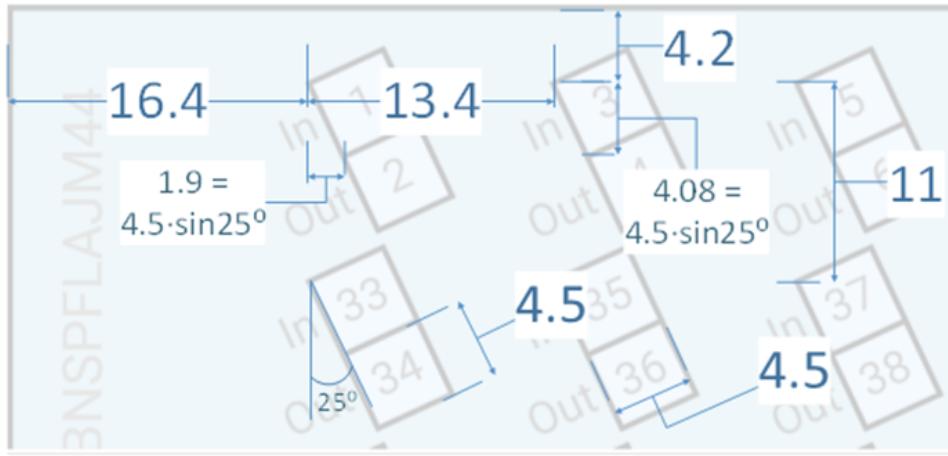
Layout Case Study

The CMD44 shelf contains 90 ports, laid out in a challenging pattern:



To layout these ports:

1. Look at each pair of ports in detail and determine their spacing (in mm) and orientation:



2. Configure the layout for the CMD44 shelf as follows:

Shelf layout configuration

| FIELD | VALUE | DESCRIPTION |
|----------------|--|---|
| Border x & y | | Not provided, so defaults to (0,0). |
| Border width | 242 | Width of the shelf in mm. |
| Border height | 38 | Height of the shelf in mm. |
| Label x | 5 | Horizontal position the shelf's label. |
| Label y | 25 | Vertical position the shelf's label. |
| Label rotation | -90 | Rotational angle of the shelf's label. |
| Child x | $16.4 + 13.4 * (n >= 89 ? 13 : \text{Math.trunc}((n-1)/2) \% 16) + 1.9 * ((n-1) \% 2)$ | Calculates the position of each port on the shelf from its port number, "n", with dimensions in mm. |
| Child y | $4.2 + 4.08 * ((n-1)\%2) + (n < 33 ? 0 : n < 65 ? 11 : 22)$ | Child rotation |

3. Configure the layout for the CMD44 ports as follows:

Ports layout configuration

| FIELD | VALUE | DESCRIPTION |
|---------------|--|--|
| Border x & y | | Not provided, so defaults to (0,0). |
| Border width | 4.5 | Width of the shelf in mm. |
| Border height | 4.5 | Height of the shelf in mm. |
| Modification | { self.addLabel((self.placeNumber % 2) == 0 ? 'Out' : 'In', 'Channel Directions')} | The port is modified: Adds an "In" or "Out" label, based on the port's placeNumber. A group name is provided to the new label to filter these labels from the diagram. |
| Label x | self.groupName=='Channel Directions' ? -0.6 : 2.25 | Positions the port's label. These labels are positioned relative to the port's coordinate space, which is rotatable. A numerical label is retrieved with the port's data and a label is added by the Modification function. Since no rotation is provided, the value is considered as zero. |
| Label y | 3 | Label rotation |
| | | Child x |
| | 16.4 + 13.4 * (n>=89 ? 13 : Math.trunc((n-1)/2) % 16) + 1.9 * ((n-1) % 2) | Calculates the position of each port on the shelf from its port number, "n", with dimensions in mm. |
| Child y | 4.2 + 4.08 * ((n-1)%2) + (n<33 ? 0 : n<65 ? 11 : 22) | Child rotation |
| | -25 | |

Raise By

SVG elements are painted to the screen in their document order, with subsequent elements painted on top of the previously painted elements. This can be incompatible with the equipment nesting model, for example, when a child of equipment E extends into a region which is occupied by one of E's siblings. This problem usually occurs when a card spans multiple slots. Although you may model the card as being a child of the first of its slots, it will be painted in the region occupied by several slots and some of these slots will then be painted over it. A similar situation occurs if you want to float labels around a slot; they may be overpainted by the elements over which you want to float them.

To overcome this, the equipment diagram allows you to "raise" an equipment's SVG up the SVG hierarchy by a specific number of levels and have it appended to one of its containing SVG element instead of being appended to its actual parent.

This behavior is configured by the Raise By (for equipment) and Raise Label By (for the labels) fields.

As a part of the layout process, any equipment or labels for which these fields return a non-zero value will have:

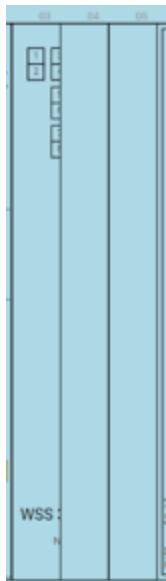
- their SVG relocated to be a child of the appropriate containing SVG element so that it is rendered as the last element of that element.
- the SVG for all their contained SVG elements moved with them (because they are nested within the SVG that is being moved).



Raising elements in this way will affect the SVG document's structure and may require corresponding changes to CSS selectors. See the [CSS Stylesheets](#) tab for information on how to define CSS stylesheets.

"Raise By" Case Study

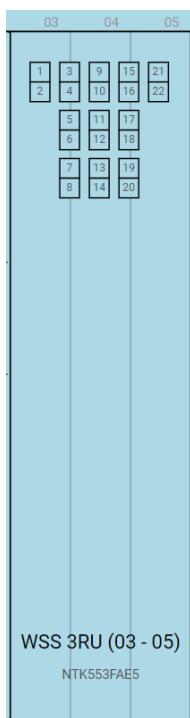
The NTK553FAE5 WSS card spans 3 slots and is plugged into slots 3-5 in a Ciena 6500 shelf. If the SVG for this card is nested inside the SVG for slot 3, then the SVG for slots 4 and slots 5 will be painted over it.



By raising the WSS card by one level (setting the [Raise by](#) field to 1 for the WSS card's layout), you can cause it to be adopted by its shelf, so that it is painted in the same position (but after painting all the slots of that shelf (along with their contents)).



In this example the card's CSS style has been made partially transparent, allowing the slots to be seen through it.



Palette

Use this tab to create or customize the CSS styles that are applied to the SVG elements of the selected equipment in the equipment diagram.

The screenshot shows the 'Device Palette' tab selected in the top navigation bar. Below the tabs, a dropdown menu displays 'Select Palette for SME 2200: Device: SME 2200'. The main area contains three sections: 'Container', 'Border', and 'Label', each with a 'CSS Classes' input field and a 'CSS Styles' input field. A checkbox labeled 'Label clipped' is located at the bottom left of the palette area.

You can set the CSS classes and CSS styles for the equipment's:

- Container (the `<svg:g>` element)
- Border (the `<svg:rect>` element)
- Label (the `<svg:rect>` element).

It is also possible to control if the labels are clipped to the equipment's border.

You also have the option to provide JavaScript expressions for the following fields to allow more control over styling in the Palette fields.



Since these fields are JavaScript expressions, to enter a fixed CSS style or class, the value must be within single or double quotation marks. For example, "font-size: 4px; fill: #EE5555; text-anchor: middle".

Palette fields

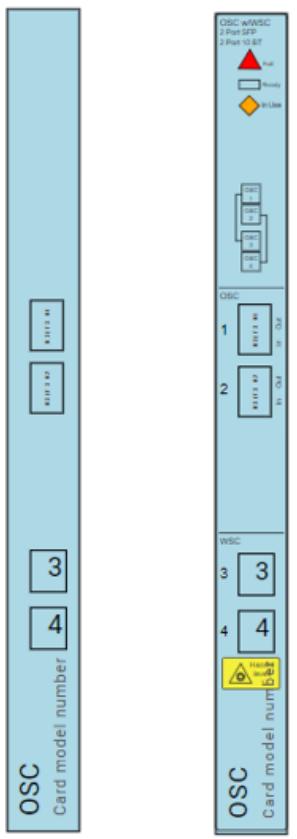
| FIELD | AVAILABLE VARIABLES | EXPRESSION RESULT TYPE | EXAMPLE |
|-----------------------|--------------------------|------------------------|--|
| Container CSS Classes | self: Equipment | String | (self.groupName== 'Model Numbers') ? "text-anchor: end; fill: #555555" : "font-weight: bold" |
| Container CSS Styles | | | |
| Border CSS Classes | | | |
| Border CSS Styles | | | |
| Label CSS Classes | self: Label v: Equipment | String | self.groupName== 'Model Numbers' ? "faded-slightly" : " |
| Label CSS Styles | | | |
| Label clipped | Checkbox | | Checked |



See the [Expression data types](#) section for information about the equipment and label data types.

SVG Definitions

Use this tab to manage SVG definitions, which can be included in the diagram based on the equipment. SVG definitions are used to add a realistic view to the diagram. For example, you can configure a card to layout its child equipment correctly (left), but adding an SVG fragment to this may make it look realistic (right):



When an SVG is defined, you can add it to an equipment by selecting it in the [SVG Inclusion](#) field in the [Layout](#) tab.

CSS Stylesheets

Use this tab to manage CSS Stylesheets and define the CSS rules that will be applied to all equipment diagrams.



- To apply styles on each equipment, use the [Palette](#) tab to define the CSS styles and classes of a piece of equipment.
- All style definitions must start with the CSS Selector `.nestedDiagram` to ensure that they only affect the Equipment Diagram pages.

The following CSS classes are added to SVG elements of the diagram and can be used to style specific elements:

CSS Stylesheets fields

| CSS CLASS | DESCRIPTION | EXAMPLE OF USE |
|----------------------------|--|--|
| nested-metamodel-<type> | Identifies the metamodel of the equipment. | .nested-metamodel-card> rect { fill-opacity:0.8; } Make all cards semi-transparent. |
| nested-archetype-<BPI UID> | Identifies the archetype of the equipment. | .nested-archetype-bpi-project-v1-2022> rect { fill-opacity:0.8; } |
| highlighted | Indicates that the searched equipment is highlighted in the diagram. | .highlighted { fill: yellow !important; } |
| newly-created | Indicates that the equipment is newly-created. | rect.newly-created { stroke: green} |
| newly-destroyed | Indicates that the equipment is newly-destroyed. | rect.newly-destroyed { stroke: red; fill: #FFE6E6 } |
| is-today | Indicates that the equipment has been modified on the current date. | .is-today { stroke-width: 3px; } |
| child-positions-undefined | Identifies the <svg:rect> of equipment that have child equipment, which cannot be displayed in the diagram because it has not been configured. | rect.child-positions-undefined { fill: #eebb00 } Shows a bright orange rectangle where the diagram is not configured. |

Summary

The *Summary* provides the summarized data in a tabular format, which includes new changes made in the Equipment Diagram Editor page.

Metadata for SME 2200

| | Created | Updated | Deleted |
|-------------------|------------------|------------------|---------|
| Archetype Formats | | Device: SME 2200 | |
| Layouts | Device: SME 2200 | | |
| Palettes | Device: SME 2200 | | |
| SVG Definitions | | | |
| CSS Stylesheets | | | |

Default Library: phase2lib C
X Cancel ✓ Apply

In this tab, you can:

- Click **Apply** to submit the new changes.
- Click **Cancel** to reject the new changes.

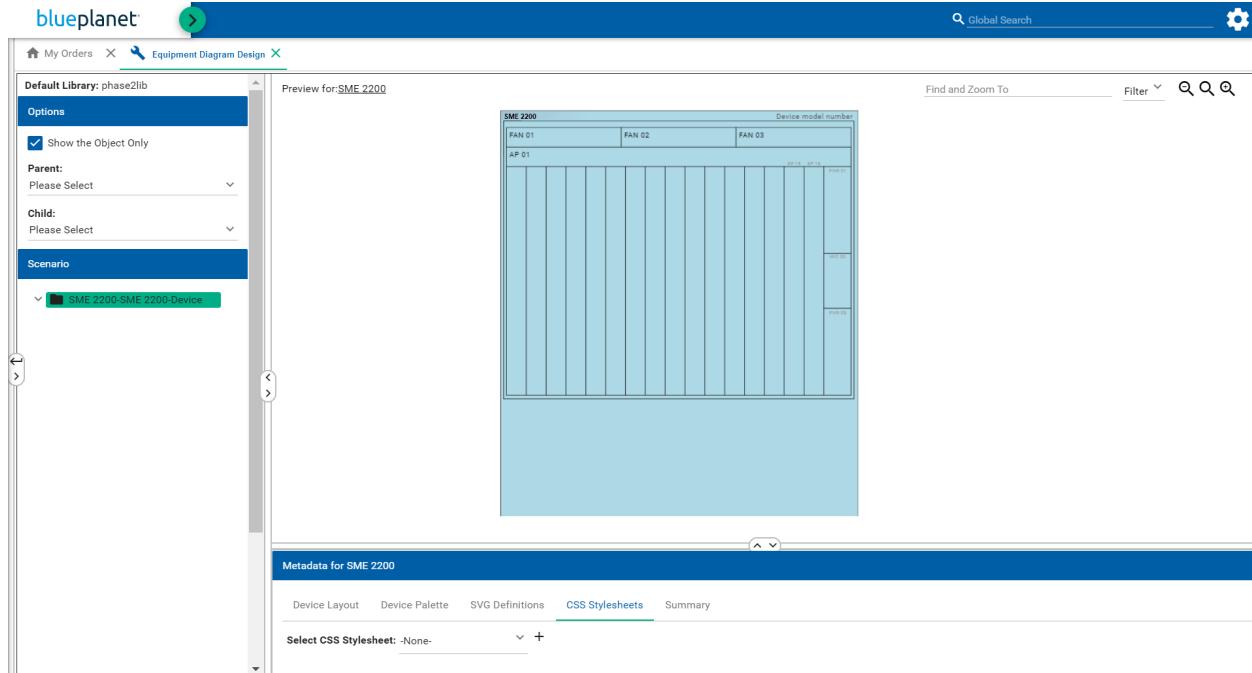
Creating an Equipment Diagram of a Device

1. Launch the Equipment Diagram Editor. For more information, see [Accessing the Equipment Diagram Editor](#).
2. Define the Options section to specify an equipment scenario. For more information, see [Options](#).
3. Select the equipment in the Scenario section that you want to create the equipment diagram for. For more information, see [Scenario](#).
4. Define the equipment's metadata in the Metadata section. For more information, see [Metadata](#).

When you define the equipment's metadata, the diagram in the Preview section updates accordingly. For more information, see [Preview](#).

5. Use the Summary tab to submit the changes made in the Metadata section. For more information, see [Summary](#).

The following figure is an example of the newly created equipment diagram.



Using the Equipment Diagram

After creating the equipment diagram of a device, you can add that device to the inventory and view the Device Diagram.

The following procedure explains how to use equipment diagram:

Create a New Rack

1. In the Locations tree, right-click a building, room, or customer premise.

The Context Menu opens.

2. Select **Operations > Create Rack**.
3. Fill out the required fields (marked with an asterisk).
4. Click **Apply**.

After you create a rack, you can

- Add an equipment to the rack.
- View the Device Diagram.

Add Equipment to the Rack

After you create a rack, you can begin to add equipment to the rack, as shown in the figure below:

1. Select **Operations > Create Equipment**.
2. In the Device Properties section, select the device you created the Equipment Diagram for from the **Device Type** drop-down list.
3. Fill out the required fields (marked with an asterisk).
4. Click **Apply**.

The screenshot shows the 'Device Properties' dialog box. The 'Device Type' field is highlighted with a red border. Other fields include 'Template' (dropdown), 'Device Name' (text input), 'IP Address' (text input), 'Status' (dropdown), 'Network' (dropdown), 'NMS' (text input), 'Management IP Address' (text input), 'Building' (dropdown), 'Floor/Room' (dropdown), 'Rack' (dropdown), and 'Rack Position' (dropdown). A 'Display Rack' button is at the bottom right. A note says 'Only list racks with space.' with a checkbox. At the bottom right are 'Cancel' and 'Apply' buttons, and a status message 'Changes applied to: 801297'.

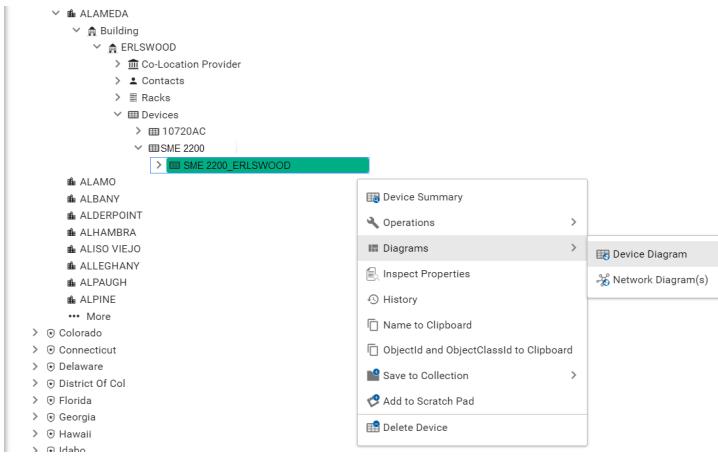
View the Device Diagram

You can launch the Device Diagram through the context menu of a device.

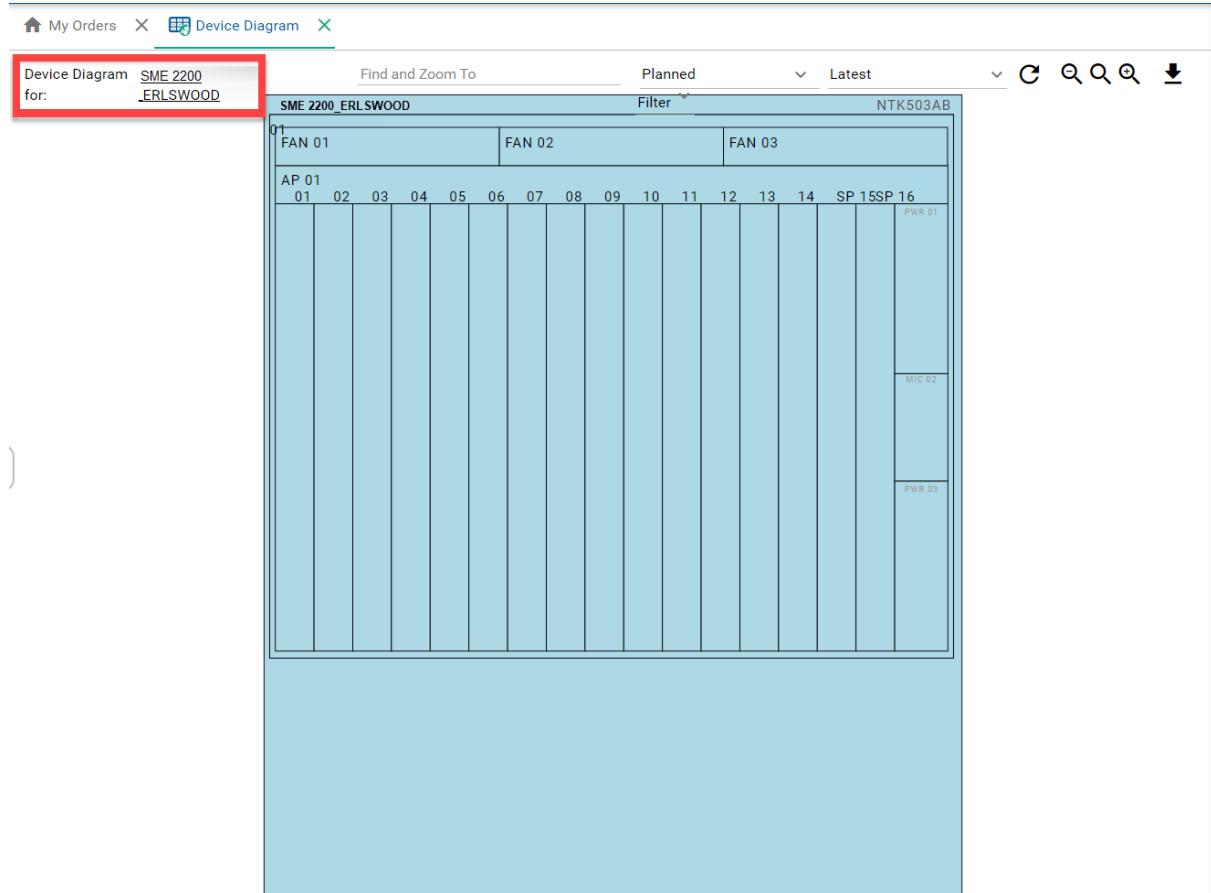
1. Right-click the device.

The Context Menu opens.

2. Select **Diagrams > Device Diagram**.



The Device Diagram opens.



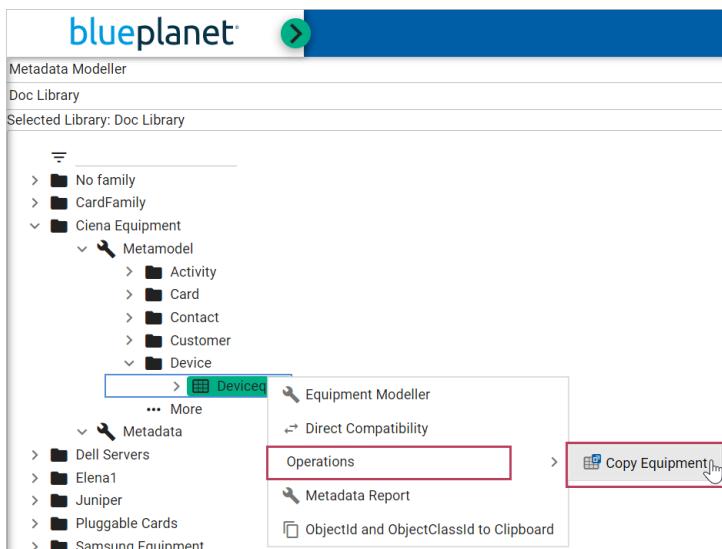
The configuration of the Device Diagram is the same as the diagram that you have created in the Equipment Diagram Editor section.

Copying existing equipment

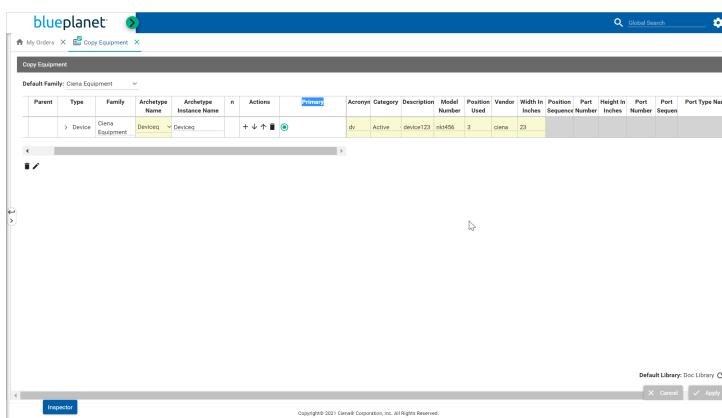
This section describes how to copy the existing equipment.

To copy existing equipment:

1. In the Metadata Modeller tree, select the default library.
2. Right-click the equipment you want to copy.
3. Select **Operations** and then select **Copy Equipment**.



Displays the Copy Equipment page.



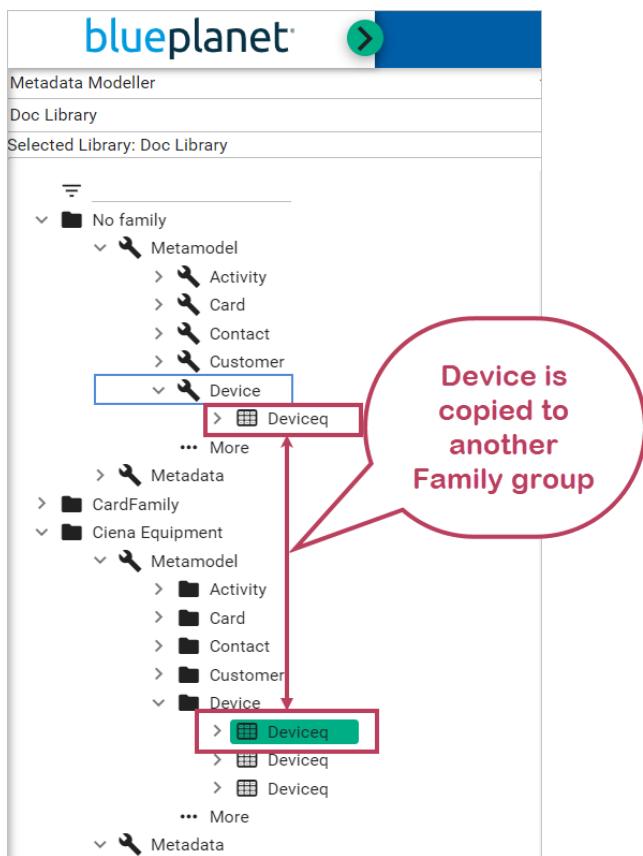
4. In the Copy Equipment page, select the Family name from the drop-down menu to add the copied equipment to the selected Family.



You can also modify the fields based on your necessity.

5. Click **Apply**.

Equipment is now copied to the selected Family.



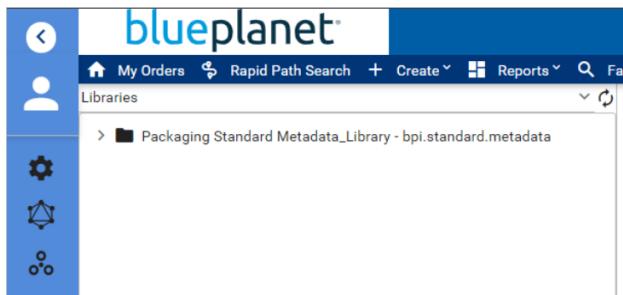
Library Processes

Creating a Project Library

At a high level, the database installation of BPI:

- loads the Hypermodel and Metamodel into BPI through scripts
- imports the Standard Library in Metadata Modeller (containing the majority of the model) and deploys the Standard Library to BPI using the lights out install tool
- executes some extra scripts referred to as Patch scripts that make some updates to the BPI model.

The Standard Library in Metadata Modeller is named `Packaging Standard Metadata_Library - bpi.standard.metadata`:



When deploying the Standard Metadata Library, Metadata Modeller uses the `BPIUUID` to compare and match objects in Metadata Modeller libraries with the BPI objects so that the deployment adds, merges, or deletes the objects as required. If no matching `BPIUUID` is found, Metadata Modeller looks for objects in BPI, where the `preferredDrnId` of the Metadata Modeller objects matches the `drnId` of the BPI objects. If unique matches are found, Metadata Modeller sets the correct `BPIUUID` on the BPI objects.

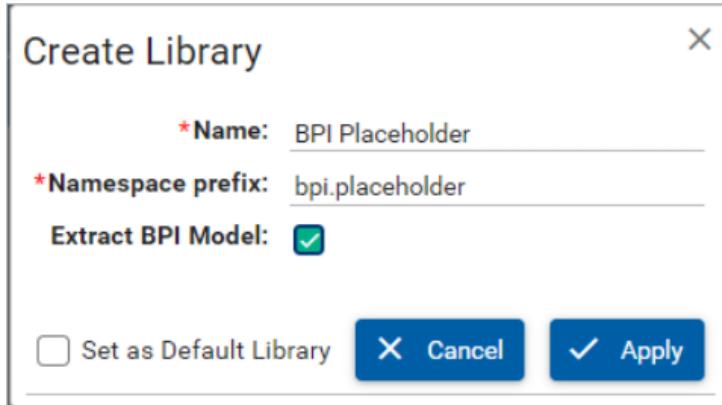
```
inventoryDao.queryForInt("with ? as input "
    + "MATCH (l:Library)-[ ]->(:Plan)-[ ]->(:Activity)-[]->(m: " +
labelMap.getKey() + ") , "
    + "(a:" + labelMap.getValue() + ":Model) "
    + "where l.drnId = input.libraryDrnId "
    + "and a.BPIUUID is null "
    + "and m.name = a.name "
    + "and m.preferredDrnId = a.drnId "
    + "SET a.BPIUUID = m.BPIUUID "
    + "return count(a) ", properties);
```

On a fresh installation, this aligns any content in BPI from earlier steps with the content of the Standard Library.

To create a new library for your project work, follow these steps:

1. Go to the Metadata Modeller menu and select **Create Library**.

The Create Library pop-up opens:



2. Create a library (referred to here as the placeholder library). This is the library where you extract the BPI model.
3. Enter the Namespace prefix. The namespace prefix generates unique identifiers for the metadata in different libraries.

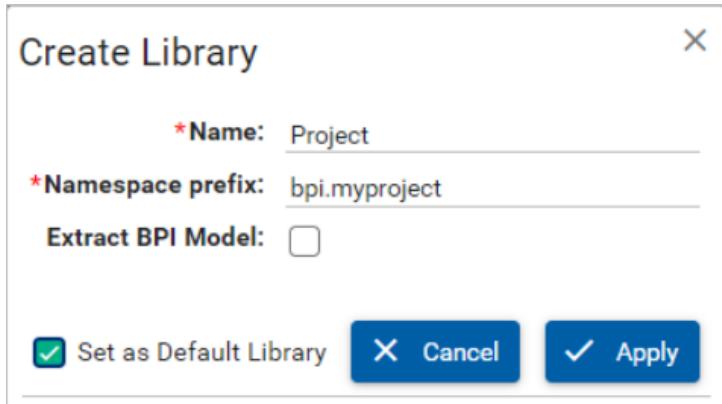
The name and namespace prefix are mandatory and must be unique for each library.

4. Check **Extract BPI Model**.

5. Click **Apply**.

This process may take up to forty minutes to complete depending on database size and deployment.

6. Close the pop-up after the process completes.
7. Create a new library (referred to here as the project library) and set it as the default library for your project.
8. Enter the Namespace prefix.
9. Check **Set as Default Library**:



10. In the Dependencies panel, locate the Standard Metadata Library (Packaging Standard Metadata_Library - bpi.standard.metadata) in the Select Library search field, and add it as a dependent to your project library. As you start typing, the library displays.
11. Select the library and click **Add Library**.
12. Expand the standard metadata library row, and select the latest revision tag. In this example, the latest revision is FINAL_TAG.
13. Click **Apply**.

This process usually takes around five minutes to complete:

The screenshot shows the 'Library Details - Project' page. Under the 'Dependencies' section, there is a table with columns: Library, Tag, CreatedDate, and Revision. One entry shows 'Packaging Standard Metadata_Library - bpi.standard.metadata' as the library, 'FINAL_TAG' as the tag, 'Dec 8, 2020, 12:01:01 PM' as the created date, and '126' as the revision. Below the table, there is a checkbox labeled 'Add selected tag as a dependency without applying the changes for its activities to this library'. At the bottom, there are 'Cancel' and 'Apply' buttons.

| | Library | Tag | CreatedDate | Revision |
|----|---|-----------|--------------------------|----------|
| .. | Packaging Standard Metadata_Library - bpi.standard.metadata | FINAL_TAG | Dec 8, 2020, 12:01:01 PM | 126 |

14. After completion, tag the latest version for future reference:

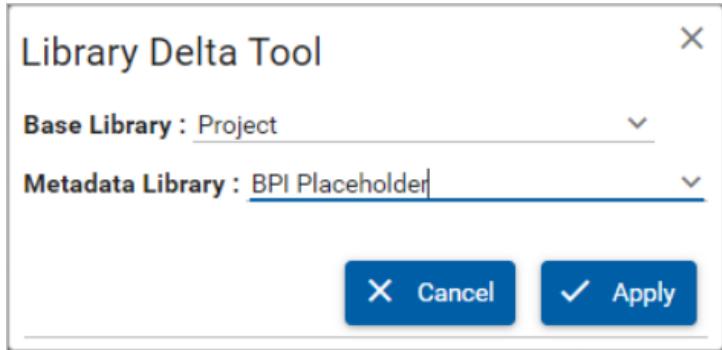
The screenshot shows the 'Library Details - Project' screen. At the top, there are fields for Name (Project), Namespace prefix (bpi.myproject), Perspective (10002), and Revision Number (0). Below these are buttons for 'Set as Default Library', 'Cancel', and 'Apply'. The main area is titled 'Associated activities' and contains a table with the following data:

| | Activity Name | Created Date | Revision Number | Tag Name | Action Buttons |
|----|---|--------------|-----------------|--------------------------------------|--|
| .. | Updating dependency on Packaging Standard Metadata_Library - bpi.standard.metadata to tag FINAL_TAG - 121 | Apr 14, 2021 | 122 | 1. Added Standard Library dependency | Deploy Validate |
| .. | Updating dependency on Packaging Standard Metadata_Library - bpi.standard.metadata to tag FINAL_TAG - 120 | Apr 14, 2021 | 121 | | Tag Validate |
| .. | Updating dependency on Packaging Standard Metadata_Library - bpi.standard.metadata to tag FINAL_TAG - 119 | Apr 14, 2021 | 120 | | Tag Validate |
| .. | Updating dependency on Packaging Standard Metadata_Library - bpi.standard.metadata to tag FINAL_TAG - 118 | Apr 14, 2021 | 119 | | Tag Validate |
| .. | Updating dependency on Packaging Standard Metadata_Library - bpi.standard.metadata to tag FINAL_TAG - 117 | Apr 14, 2021 | 118 | | Tag Validate |

Copyright © 2021 Ciena® Corporation, Inc. All Rights Reserved.

15. Go to the Metadata Modeller menu and select **Library Delta Tool**.

The Library Delta Tool pop-up opens. The Library Delta Tool updates your project library and applies the changes from the extracted library. This aligns your project library with the BPI model.



16. From the Base Library dropdown menu, select the project library.

17. From the Metadata Library dropdown menu, select the placeholder library.

18. Click **Apply**.

This process may time out in the UI, but still execute in the database. This process usually takes around ten minutes to complete.

19. To check the completion status, refresh **Associated activities**. The latest revision is the one that removes relationships from the Baseline Perspective:

Library Details - Project

Library Properties

| Activity Name | Created Date | Revision Number | Tag Name |
|--|--------------|-----------------|--------------------|
| Removing Relationship from Baseline Perspective - 18 | Apr 14, 2021 | 167 | Latest(Not Tagged) |
| Removing Relationship from Baseline Perspective - 17 | Apr 14, 2021 | 166 | |
| Removing Relationship from Baseline Perspective - 16 | Apr 14, 2021 | 165 | |
| Removing Relationship from Baseline Perspective - 15 | Apr 14, 2021 | 164 | |
| Removing Relationship from Baseline Perspective - 14 | Apr 14, 2021 | 163 | |

Associated activities

20. After completion, tag the latest version for future reference:

Library Details - Project

Library Properties

| Activity Name | Created Date | Revision Number | Tag Name |
|---|--------------|-----------------|--------------------------------------|
| Removing Relationship from Baseline Perspective - 18 | Apr 14, 2021 | 167 | 2. After Delta applied |
| Updating dependency on Packaging Standard Metadata_Library - bpi.standard.metadata to tag FINAL_TAG - 121 | Apr 14, 2021 | 122 | 1. Added Standard Library dependency |

Associated activities

21. Deploy this tagged version of the project library. The deployment does not change the BPI model because it's now aligned with the project library. This process may time out in the UI, but still execute in the database. This process usually takes around ten minutes to complete. If the deployment is complete, the latest tag, including the previous tags are set to Deployed:

The screenshot shows the 'Library Details - Project' dialog box. At the top, it displays basic information: Name (Project), Namespace prefix (bpi.myproject), Perspective (10002), and Revision Number (0). Below this, the 'Associated activities' section lists two entries:

| Activity Name | Created Date | Revision Number | Tag Name | Status |
|---|--------------|-----------------|--------------------------------------|----------|
| Removing Relationship from Baseline Perspective - 18 | Apr 14, 2021 | 167 | 2. After Delta applied | Deployed |
| Updating dependency on Packaging Standard Metadata_Library - bpi.standard.metadata to tag FINAL_TAG - 121 | Apr 14, 2021 | 122 | 1. Added Standard Library dependency | Deployed |

At the bottom right of the dialog are three buttons: 'Set as Default Library' (with a checked checkbox), 'Cancel', and 'Apply'.

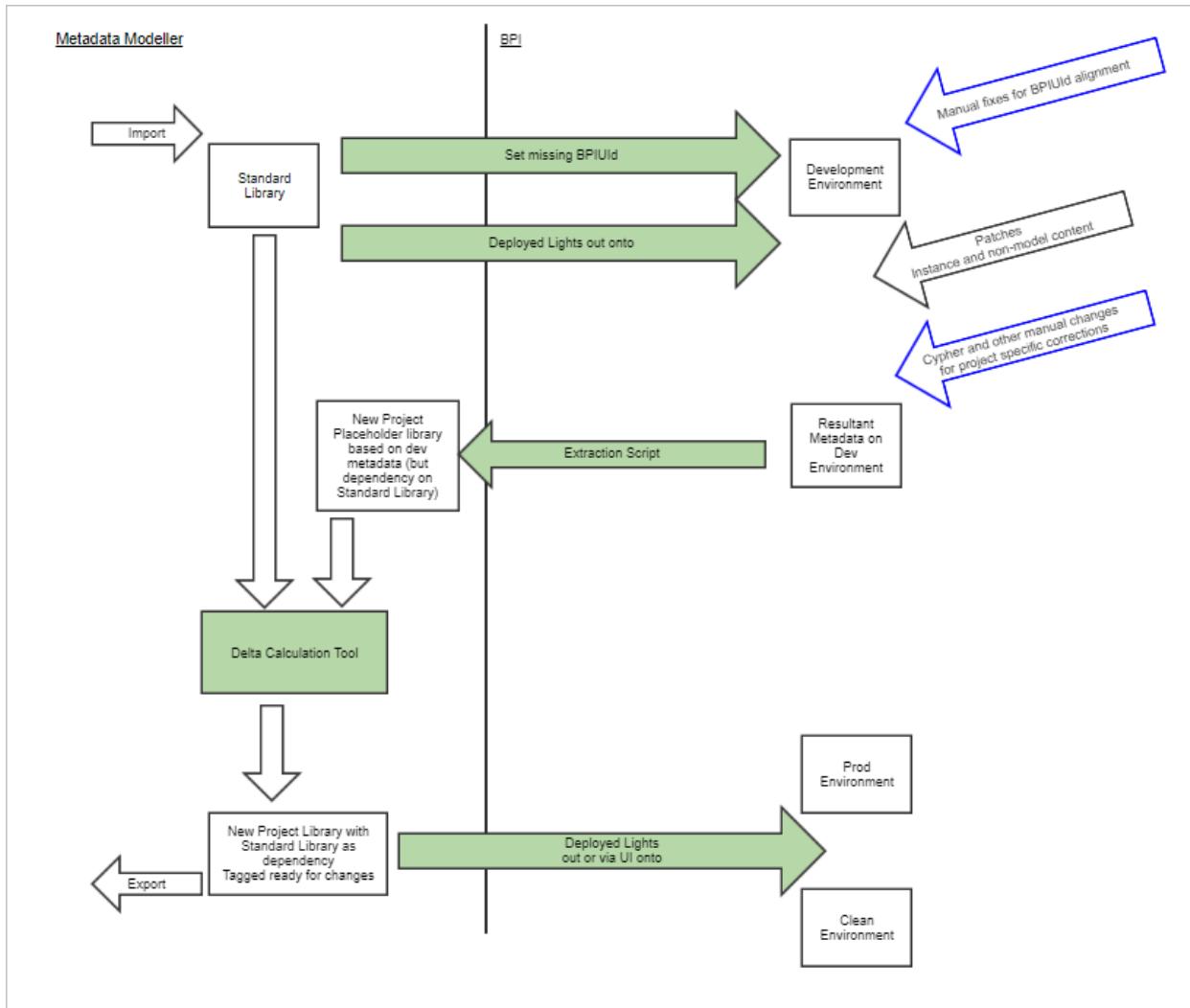
You can now work with your new project library.

In existing deployments that are not new installations, the steps are similar, but may require manual intervention to correct any mismatches in the model or other project specifics. For example:

- The deployment of the Standard Library may fail if there are conflicts with the existing metadata. For example, if the metadata in BPI has the same drnId as the preferredDrnId in the library, but the objects do not match. These issues are fixed by manually correcting BPI. For example, if an object does not match, set the correct BPIUUID on the object in BPI to its matching object in the library. If there are conflicting objects, change the BPI drnId of the object that conflicts with the library.
- The Standard Library deployment may have created some example models that the project wishes to remove.

Make any manual changes to the project metadata in BPI after the installation of the standard library, product patches, and any other product installation.

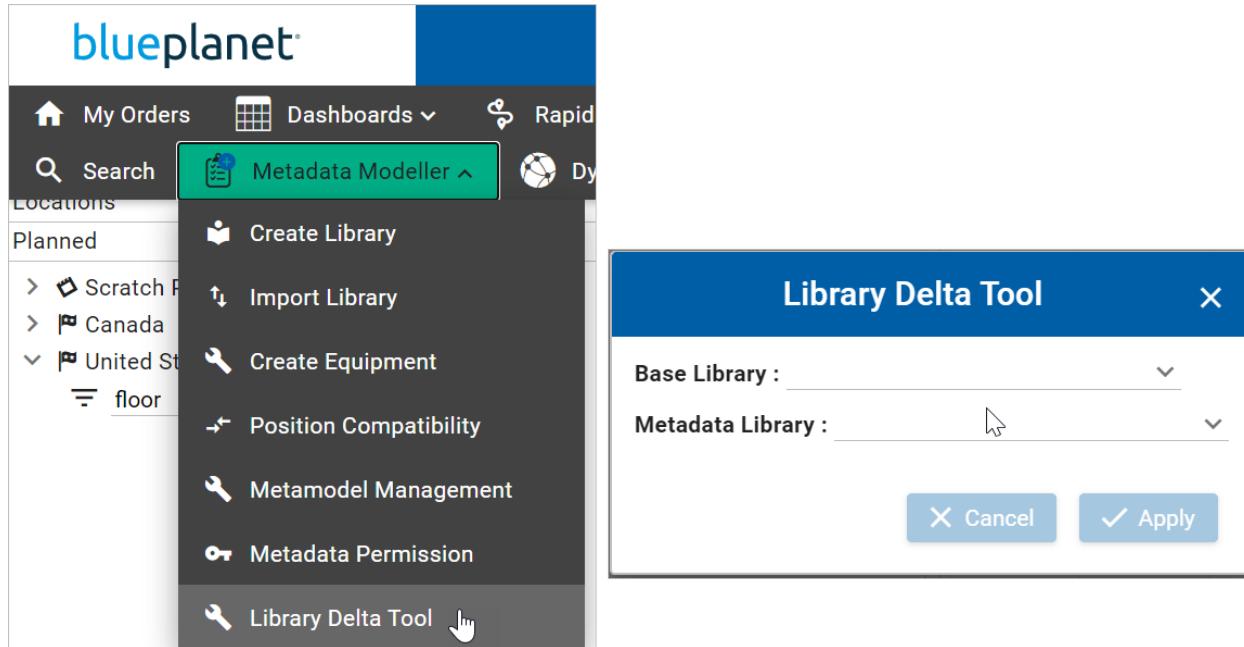
The following diagram illustrates the preceding steps. The arrows in blue are applicable for upgrade scenarios:



Library Delta Tool

Library Delta tool can be accessed via Blue Planet UI, under Metadata Modeller Tab, with option Library Delta Tool.

The pop-up window is shown in the following figure:



The Library Delta tool takes two libraries, the Base Library and the Metadata Library, finds the delta between them and updates the Base Library with those deltas.

Library Lights Out Tool

The Library Lights Out Tool allows the use of the REST APIs for Library Import and Library Tag Deploy with parameters in installations or from the command line. This tool is used as a jar and is available at `db-server-neo4j\blueplanet-inventory-examples\packager\target\blueplanet-inventory-examples-build-xx.x.x-SNAPSHOT\tools\blueplanet-library-lightsout-tool.jar` on successful compilation of modules from `level \db-server-neo4j`

Also, `blueplanet-inventory-examples-build-xx.x.x-SNAPSHOT-bin.zip` must be unzipped at location `db-server-neo4j\blueplanet-inventory-examples\packager\target` for the jar.

The following commands when run on Windows OS may require wrapping filename and Library Name in double quotes instead of single.

```
java -jar blueplanet-library-lightsout-tool.jar -u <username> -p <password> -url <showcase-rest-url> -f <FilenameWithPath> -l <LibraryName> -t <Tag Name>
Example :
java -jar blueplanet-library-lightsout-tool.jar -u normal -p ***** -url 10.78.103.233:8080/blueplanet-inventory-rest-showcase/ -f '.../tdm/packaging/Packaging Project TDM Metadata_Library - bpi.project.tdm.metadata.json' -l 'Packaging Project TDM Metadata_Library - bpi.project.tdm.metadata' -t 'TAGA'
```

The command-line tool has a help option which explains the usage with parameter details as shown in the following figure:

```
$ java -jar blueplanet-library-lightsout-tool.jar -h
Usage : <options>
-u or -U -> <Application Environment's Username>
-p or -P -> <Application Environment's Password>
-url or -URL -> <Rest Base URL>
-f or -F -> <File Path with File Name to be Imported>
-l or -L -> <Library Name for Deployment>
-t or -T -> <Tag Name for Deployment>
```

The use case of generating the JSON file with a set of Metadata can have parameters, including:

- - u / - u: Username for the Target Environment application
- - p / - p: Password for Target Environment application
- - url / - URL: Target Environment REST application URL
- - f / -F: JSON File downloaded from Source Environment or a particular PACK JSON File (Entire Path with Filename)
- - l / -L: Library name (Generally the same as JSON Filename without .json extension)
- - t / -T: TAG Name (While exporting JSON, a TAG is specified which needs to be deployed)

LEGAL NOTICES

THIS DOCUMENT CONTAINS CONFIDENTIAL AND TRADE SECRET INFORMATION OF CIENA CORPORATION, INCLUDING ITS SUBSIDIARY, BLUE PLANET SOFTWARE, INC., AND ITS RECEIPT OR POSSESSION DOES NOT CONVEY ANY RIGHTS TO REPRODUCE OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING THAT IT MAY DESCRIBE. REPRODUCTION, DISCLOSURE, OR USE IN WHOLE OR IN PART WITHOUT THE SPECIFIC WRITTEN AUTHORIZATION OF CIENA CORPORATION OR BLUE PLANET SOFTWARE, INC IS STRICTLY FORBIDDEN. EVERY EFFORT HAS BEEN MADE TO ENSURE THAT THE INFORMATION IN THIS DOCUMENT IS COMPLETE AND ACCURATE AT THE TIME OF PUBLISHING; HOWEVER, THE INFORMATION CONTAINED IN THIS DOCUMENT IS SUBJECT TO CHANGE. While the information in this document is believed to be accurate and reliable, except as otherwise expressly agreed to in writing, BLUE PLANET PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OR CONDITION OF ANY KIND, EITHER EXPRESS OR IMPLIED. The information and/or products described in this document are subject to change without notice. For the most up-to-date technical publications, visit <https://my.ciena.com>.

Copyright© 2022 Ciena® Corporation. All Rights Reserved

The material contained in this document is also protected by copyright laws of the United States of America and other countries. It may not be reproduced or distributed in any form by any means, altered in any fashion, or stored in a data base or retrieval system, without express written permission of Blue Planet. Ciena®, the Ciena logo, Blue Planet®, and other trademarks and service marks of Ciena, Blue Planet, and/or their affiliates appearing in this publication are the property of Ciena and Blue Planet. Trade names, trademarks, and service marks of other companies appearing in this publication are the property of the respective holders.

Security Ciena® cannot be responsible for unauthorized use of equipment and will not make allowance or credit for unauthorized use or access.