TEAMCENTER

Structure Partitions — Deployment and Administration

Teamcenter 2412

SIEMENS

Unpublished work. © 2025 Siemens

This Documentation contains trade secrets or otherwise confidential information owned by Siemens Industry Software Inc. or its affiliates (collectively, “Siemens”), or its licensors. Access to and use of this Documentation is strictly limited as set forth in Customer’s applicable agreement(s) with Siemens. This Documentation may not be copied, distributed, or otherwise disclosed by Customer without the express written permission of Siemens, and may not be used in any way not expressly authorized by Siemens.

This Documentation is for information and instruction purposes. Siemens reserves the right to make changes in specifications and other information contained in this Documentation without prior notice, and the reader should, in all cases, consult Siemens to determine whether any changes have been made.

No representation or other affirmation of fact contained in this Documentation shall be deemed to be a warranty or give rise to any liability of Siemens whatsoever.

If you have a signed license agreement with Siemens for the product with which this Documentation will be used, your use of this Documentation is subject to the scope of license and the software protection and security provisions of that agreement. If you do not have such a signed license agreement, your use is subject to the Siemens Universal Customer Agreement, which may be viewed at **https://www.sw.siemens.com/en-US/sw-terms/base/uca/**, as supplemented by the product specific terms which may be viewed at **https://www.sw.siemens.com/en-US/sw-terms/supplements/**.

SIEMENS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS DOCUMENTATION INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. SIEMENS SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, LOST DATA OR PROFITS, EVEN IF SUCH DAMAGES WERE FORESEEABLE, ARISING OUT OF OR RELATED TO THIS DOCUMENTATION OR THE INFORMATION CONTAINED IN IT, EVEN IF SIEMENS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TRADEMARKS: The trademarks, logos, and service marks (collectively, "Marks") used herein are the property of Siemens or other parties. No one is permitted to use these Marks without the prior written consent of Siemens or the owner of the Marks, as applicable. The use herein of third party Marks is not an attempt to indicate Siemens as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A list of Siemens’ Marks may be viewed at: **www.plm.automation.siemens.com/global/en/legal/trademarks.html**. The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow’s idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: **support.sw.siemens.com**

Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

Contents

© 2025 Siemens

© 2025 Siemens

1. Set up Structure Partitions

BOM engineers, designers, and downstream and upstream consumers mostly work with specific structures (assemblies) rather than the entire product itself. In the case of complex and large products, locating the required components of a product becomes tedious. To help users easily find the required structure in Teamcenter, BOM architects can divide a large product into smaller, manageable sections called *partitions*. Partitions can belong to different *partition schemes* such as functional, physical, or spatial. Every structure can be organized within one or more partitions as per the partition scheme without having the need to duplicate the structure. For example, the engine structure can be assigned to partitions in the physical and the functional partition schemes.

You, as an administrator, must set up Structure Partitions so that BOM architects can create the partition schemes and partitions.

Where do I go from here?

Business User	See *Structure Partitions — Usage*.
Administrator	
Deploy partition management	Install the **Partitions for Structure** feature either through Teamcenter Environment Manager (TEM) or through Deployment Center.
Change the partition behavior	Change the **change the default behavior of partitions** to suit your business needs by setting certain BMIDE constants.
Specify access privileges for partitions	**Grant access rights** to users in order to create partition schemes, partitions, and add or remove structure elements to or from partitions.
Create partitions from existing partition templates	First **add partitions** to a product structure from partition templates already available in Teamcenter. Next, **add structure elements** to the partitions.

© 2025 Siemens

© 2025 Siemens

2. Install Partitions for Structure using Teamcenter Environment Manager (TEM)

The **Partitions for Structure** feature is required to get the partitions that are in turn required to organize the contents of a product structure. Before installing this feature, ensure that Teamcenter is set up (Windows or Linux). Next, update your Teamcenter setup to install the following features through Teamcenter Environment Manager (TEM):

- **Base Install** > **Active Workspace** > **Client** > **Partitions for Structure**
- **Base Install** > **Active Workspace** > **Server Extensions** > **Partitions for Structure**
- **Extensions** > **Partitions for Structure**

© 2025 Siemens

© 2025 Siemens

3. Install Partitions for Structure using Deployment Center

Add **Partitions for Structure** to your existing Teamcenter environment through a series of tasks from selecting the application and entering configuration parameters to generating and running deployment scripts.

Procedure

1. Log on to Deployment Center and select the environment to which you want to add **Partitions for Structure**.

2. Go to the **Applications** task. Click **Add or Remove Selected Applications**.

3. In the **Available Applications** panel, use the web browser search to find the **Partitions for Structure** and **Partitions for Structure for Active Workspace** applications. Select the applications, and then click **Update Selected Applications**.

   Deployment Center automatically selects any additional dependent applications.

4. Go to the **Components** task.

5. In the **Selected Components** list, make sure that all components in the environment show a configuration status of **100%**.

   When all components are fully configured, the **Deploy** task is enabled.

6. Go to the **Deploy** task. Click **Generate Install Scripts** to generate deployment scripts you will use to update affected machines.

   When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

7. Locate deployment scripts, copy each script to its target machine, and then run each script on its target machine.

   For more information about running deployment scripts, see *Deployment Center — Usage*.

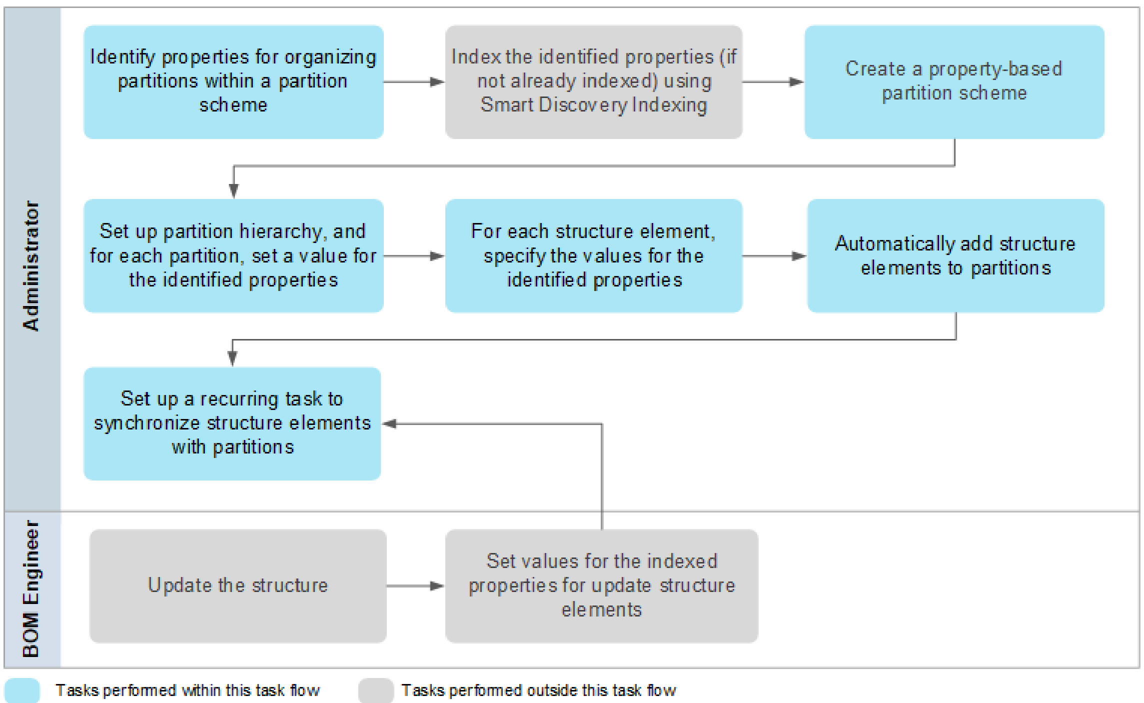
© 2025 Siemens

© 2025 Siemens

4. Automatically add structure elements to partitions based on indexed properties

Task flow to automatically add structure elements to partitions

Instead of users manually creating the partition hierarchy, you can automate the process to create and populate partitions with structure elements based on certain structure properties, such as item revision properties or occurrence properties. After identifying the properties, you must index the structure and the properties by using Smart Discovery Indexing, if not already indexed. Also, before proceeding for the property-based setup, ensure that the properties are displayed on the Filter panel and are filtering the content.



Example — Consider that for the product structure, *Crosskart*, you identify that the properties *PTY10* and *PTY20* will be used for populating the structure elements into different partitions. The property values for the structure elements are set as follows:

© 2025 Siemens

Structure element	PTY10 value	PTY20 value
Glue Gun	10B	6C
Knockdown Tool	10B	6E
Car Shampoo	10B	7D
Rear Speaker	11B	8C
Front Speaker	11B	8E
Steering Wheel	12G	9G
Rear Suspension	13G	14F
Front Suspension	13G	14N

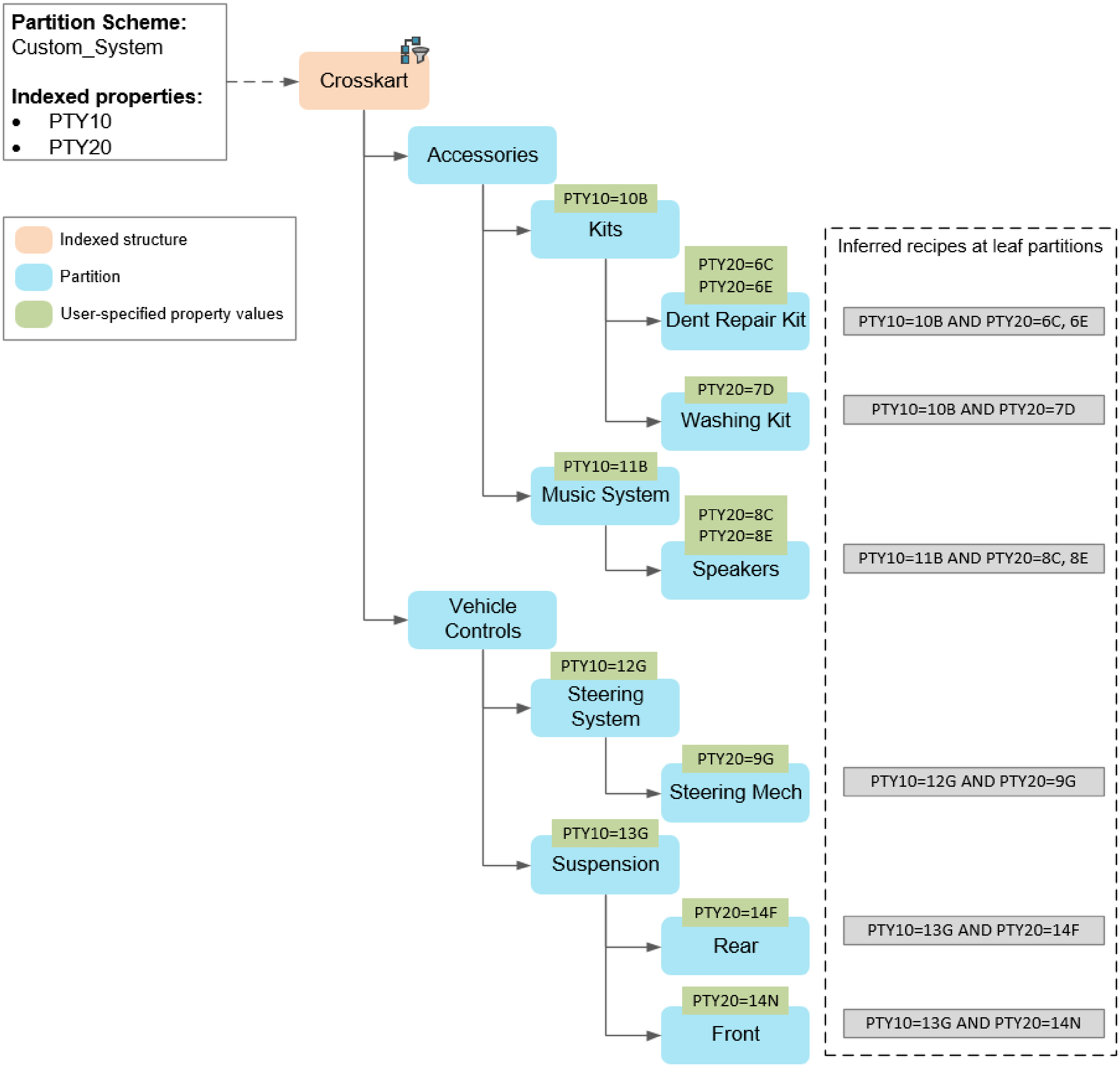
After identifying the properties, you must ensure that these properties are indexed using Smart Discovery Indexing.

Next, you **create different partition schemes** to include certain elements within specific partitions as per the business requirements. For example, you create a *Custom_System* partition scheme to include different partitions in a hierarchical manner. You can set up the partition hierarchy either on Active Workspace or **through a *.csv* file**. For each partition, you specify values for the *PTY10* and *PTY20* properties so that structure elements can be later on populated within correct partitions.

In this example, for the *Kits* partition, you enter the value of *PTY10* as *10B*, and for *Dent Repair Kit*, you enter the value of *PTY20* as *6C* and *6E*.

For leaf partitions, a recipe is automatically inferred based on the specified values. For example, the inferred recipe for *Dent Repair Kit* is *PTY10=10B AND PTY20=6C, 6E*.

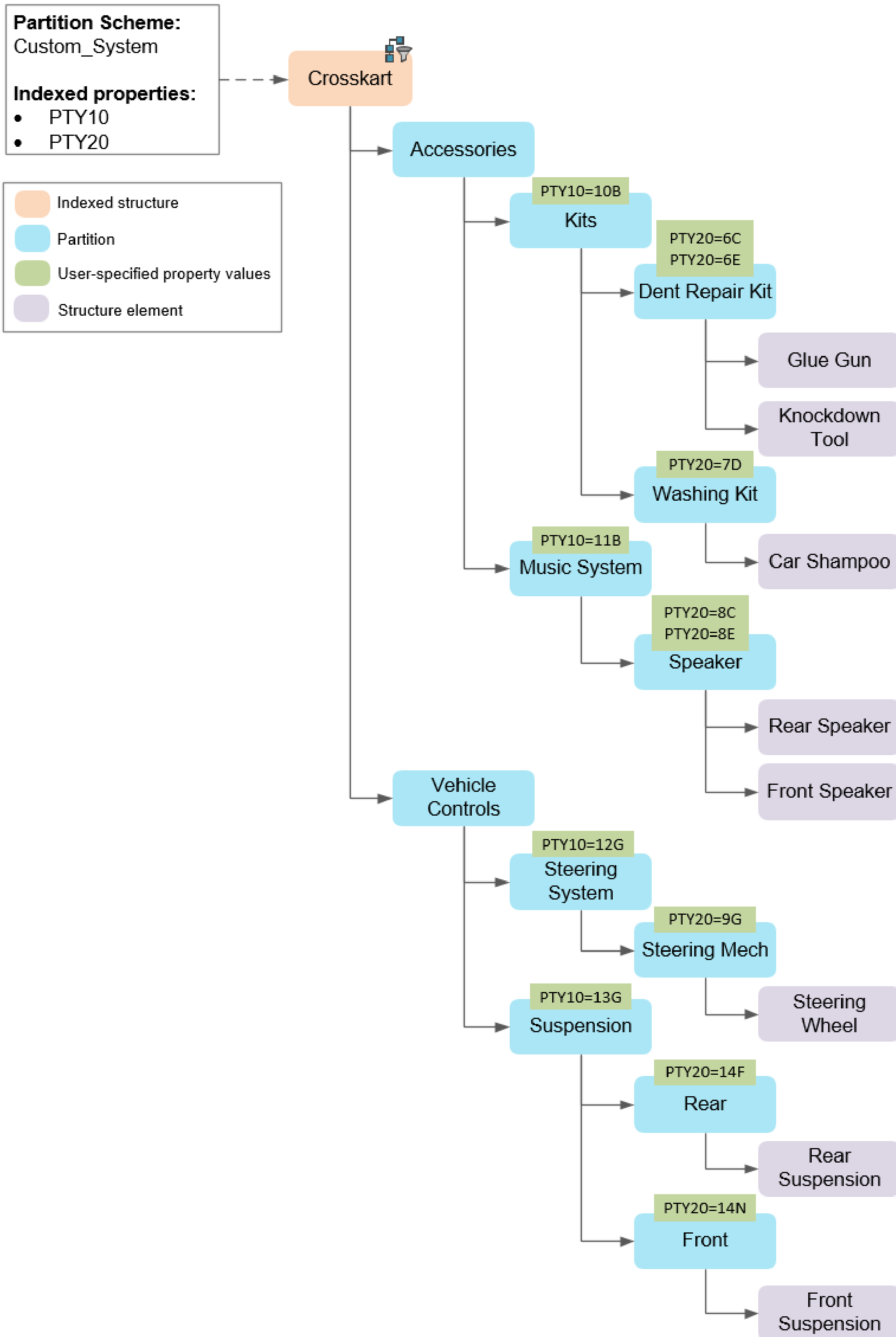
© 2025 Siemens



Before **populating the partitions with structure elements**, ensure that you have set the values of the indexed properties on partitions and structure elements correctly. These properties are matched with the inferred recipe, and the elements are added to the respective partitions. Once the elements are organized within partitions, you cannot move the partitions or change the values of the indexed properties set on the partitions. However, you can update the values of the indexed properties for the structure elements.

In this example, the inferred recipe for *Dent Repair Kit* is *PTY10=10B AND PTY20=6C, 6E*. As the *PTY20* property of *Glue Gun* and *Knockdown Tool* have the values *6C* and *6E*, these elements are included in the *Dent Repair Kit* partition.

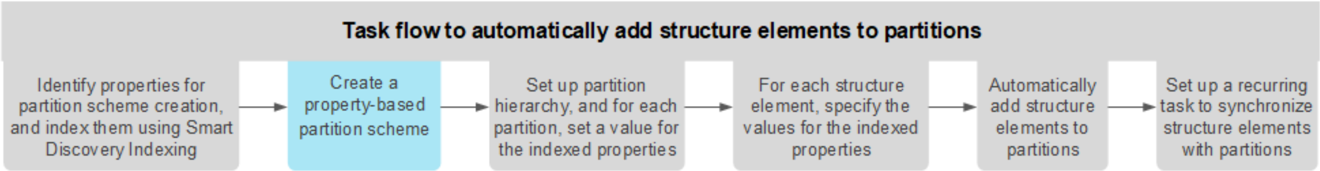
© 2025 Siemens



© 2025 Siemens

After the partitions are created and are populated with structure elements, a BOM engineer can update the structure or can update the values of *PTY10* or *PTY20*. The synchronization process that you set up updates the structure regularly with such changes.

Create a property-based partition scheme



You create different partition schemes for a structure to include certain structure elements within them as per the business requirement.

> Example:
>
> You create a *functional* partition scheme, and divide it into different partitions as per various functions such as electrical, mechanical, and so on. You include all the electrical parts of *Crosskart* within the electrical partition. Now, if a BOM engineer identifies that an issue with the brake assembly is due to some electrical component connected to it, the BOM engineer can work with the parts that are within the electrical partition instead of the entire *Crosskart* structure.

Creating a persistent partition scheme is a one-time setup. As part of this setup, you must perform the following tasks:

a. Create a new partition scheme.

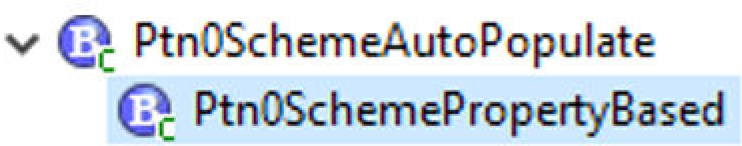
b. Specify what type of partition this new scheme can use.

   The partition type should be partition-owned.

c. Create a new partition.

d. Make this partition partition-owned.

To simplify the preceding steps, you can use the following sample partition scheme:



For any partition scheme that you create, you must set the value of the **Ptn0SchemeAllowedTypes** business object constant to **Ptn0PropertyBased**.

© 2025 Siemens

Set the value of the **Ptn0IsOwningMemberships** business object constant in Business Modeler IDE (BMIDE) to **True** (partition-owned).

You use the sample condition called **Ptn0IsPartitionableCond**.

If you need to create any other persistent partition scheme, perform the following steps:

1. Identify properties for partition scheme creation, and index them using Smart Discovery Indexing. You can index both item revision and occurrence properties. However, before indexing an item revision property, you must index at least one occurrence property.

   Structures are constantly updated. After you populate partitions of a structure with structure elements, elements can be added, removed, or modified. To ensure that the structure elements are correctly placed within appropriate partitions, you must set the **Awp0SearchIsStored** property constant to **true** for the workspace object properties that you have indexed (for example, PTY10), and which will be used as the search recipe.

   Caution:

   To assign members to partitions automatically, you must mark at least one occurrence property for Smart Discovery indexing.

2. In BMIDE, locate the **Ptn0SchemeAutoPopulate** business object, and create a new business object under it, for example, **Ptn0Custom_System**.

3. In the newly added business object, go to the **Business Object Constant** tab, and edit **Ptn0SchemeOrgPropertyNames**.

   In the **Modify Business Object Constant** dialog box, include the indexed properties as **Value** in the following format:

   TypeName1.PropertyName1,TypeName2.PropertyName2

© 2025 Siemens

Here, *TypeName* is a business object and *PropertyName* is the index property that you are using for the scheme creation.

Make sure that the indexed properties that you include are available under the **AWS_Discovery_FullTextSearch_FacetFilters** preference.

> Example:
>
> **Awb0DesignElement.PTY10,Awb0DesignElement.PTY20**
>
> OR
>
> **Awb0PartElement.PTY10,Awb0PartElement.PTY20**
>
> OR
>
> **ItemRevision.IndexedPropertyName1,ItemRevision.IndexedPropertyName2**

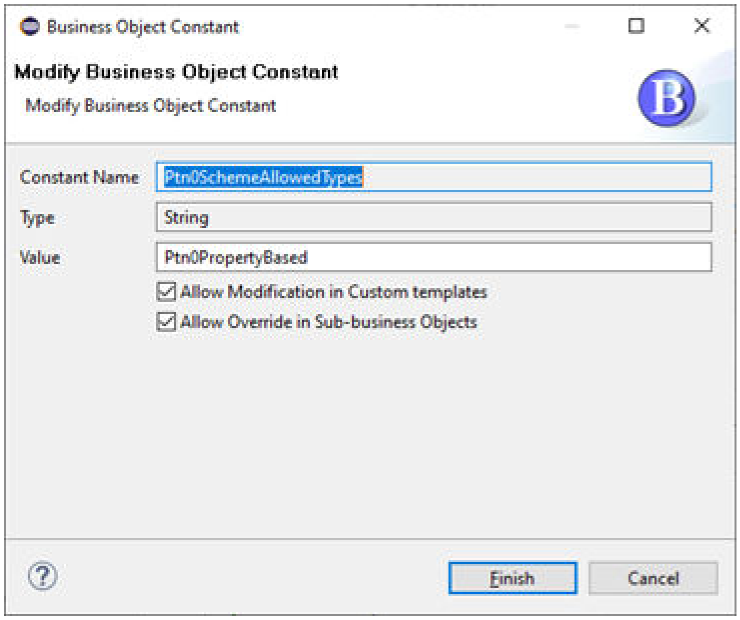
Ensure that the indexed property is not of the type **Date**.

> Caution:
>
> In the **Ptn0SchemeOrgPropertyNames** business object constant for a scheme type, you designate the properties that you want to use for automatically populating a partition scheme. After you deploy the changes in BMIDE, these properties are also saved in the partition scheme instance that a business user creates.
>
> After the business user creates a partition scheme instance, you cannot change those properties in this partition scheme instance even if you change the properties in the **Ptn0SchemeOrgPropertyNames** business object constant.
>
> Therefore, to add, remove, or replace any properties, you must create a new scheme type in BMIDE.
>
> We recommend that you start by creating a scheme with a name that represents some number and then create another scheme with a new version in it. After that, you add any additional properties to the new scheme.
>
> Example:
>
> Consider that you have an existing partition scheme called **SchemeA_v01** that has two properties associated with it. Now, to include an additional property, you create a new scheme called **SchemeA_v02** and associate the additional property with this scheme. Then, you choose the **SchemeA_v02** scheme and follow the next steps to create a persistent partition scheme.

4. In the **Business Object Constant** tab, edit **Ptn0SchemeAllowedTypes**.

   In the **Modify Business Object Constant** dialog box, you specify the types of partitions that can be created in the context of the partition scheme. It is recommended that the partitions in these schemes should have partition-owned membership.

© 2025 Siemens



5. Save and deploy the data model changes.

6. First, export the existing column arrangement in Active Workspace by running the **export_uiconfig** utility on Teamcenter command prompt. Then, include the indexed properties here so that the properties appear under the **Partition Recipe Props** column arrangement in Active Workspace.

   While including the properties in this file, you must add **PS_** as a prefix to the property names. For example, **PS_PTY10**.

   The following is a sample code snippet for your reference to include the object types and property names as per your requirement:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Import>
  <Client abbreviation="AWClient" name="AWClient">
     .<ClientScope hostingClientName=""
name="Awb0OccurrenceManagement" uri="Awb0OccurrenceManagement">
      <ColumnConfig columnConfigId="contentColConfig"
columnConfigName="Partition Search Recipe Properties" sortBy="-1"
sortDirection="Ascending">

      <ColumnDef objectType="Fgf0PartitionElement"
propertyName="REF(awb0UnderlyingObject,Ptn0Partition).PS_PTY10"
width="100"/>
      <ColumnDef objectType="Fgf0PartitionElement"
propertyName="REF(awb0UnderlyingObject,Ptn0Partition).PS_PTY20"
width="100"/>

<!-- Please Add other Fgf0PartitionElement/Awb0DesignElement/
Awb0PartElement properties if needed-->
```


© 2025 Siemens

```
......<ColumnDef objectType="AwbDesignElement" propertyName="PTY10"
width="100"/>
......<ColumnDef objectType="Awb0DesignElement" propertyName="PTY20"
width="100"/>
<!-OR-->
.....<ColumnDef objectType="Awb0PartElement" propertyName="PTY10"
width="100"/>
.....<ColumnDef objectType="Awb0PartElement" propertyName="PTY20"
width="100"/>
.....<columDef objectType="Awb0DesignElement"
propertyName="REF(awb0UnderlyingObject,ItemRevision).IndexedPropertyN
ame>
      </ColumnConfig>
    </ClientScope>
  </Client>
</Import>
```

If the indexed properties are part properties, set the **name**, **uri**, and **columnConfigId** as **Awb0OccurrenceManagement.Part Revision** in the *partitionCots.xml* file. For design properties, set these as **Awb0OccurrenceManagement.Design Revision**.

Lastly, import the column arrangement in Active Workspace by running the **import_uiconfig** utility on Teamcenter command prompt.

Define rules to specify the structure elements to be included in a partition

You can define rules to specify which structure elements must be included within partitions. For example, you can define a rule that specifies that an element must be included within a partition only if it matches a specific type or has a property set to a specific value.

To define the rules, you create a BMIDE condition. For detailed information about creating a condition, see Add a condition.

The following are some examples of conditions:

- **Example 1**

  Partition all components where the parent type is **Installation Assembly**, but the child type is *not* **Installation Assembly**.

  This means that Teamcenter allows all children whose parent type is **Installation Assembly** to be partitioned except those that are of the same type as that of their parent.

- **Example 2**

© 2025 Siemens

Partition all components where the parent type is **Installation Assembly**, but the child type is *not* **Installation Assembly**, and where the child type is **Design** and **Design.object name** does not begin with 'GEO'.

This means that Teamcenter allows all children whose parent type is **Installation Assembly** to be partitioned except those that are of the same type as that of their parent. In addition, Teamcenter excludes the children of type **Design** whose names do not start with 'GEO'.

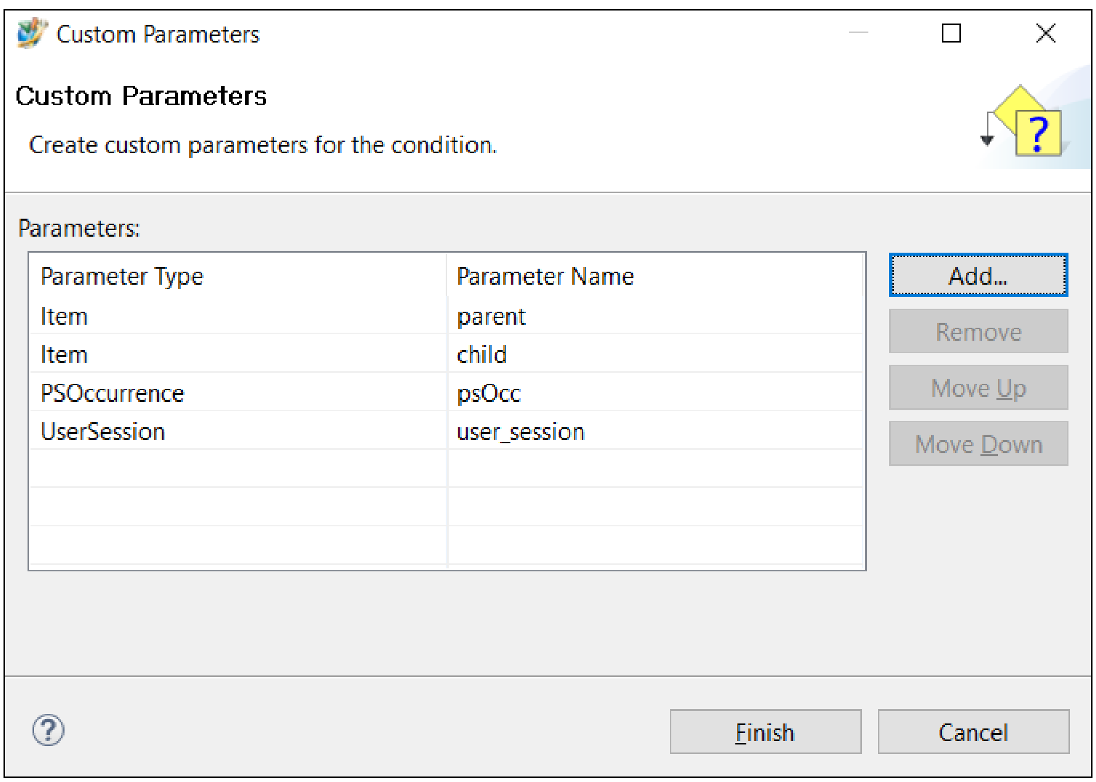
- **Example 3**

  Partition all components where the parent type is **Installation Assembly**, and the child type is **Design** and **Design.object name** begins with 'GEO'.

  This means that Teamcenter allows all children whose parent type is **Installation Assembly** to be partitioned. In addition, it includes the children whose child type is **Design** and whose names start with 'GEO'.

To define the rules:

1. While creating a condition, set **Input Parameters** as **Custom**, and specify the following parameters in the **Custom Parameters** dialog box:



© 2025 Siemens

Caution:

In the **Signature** box, the four parameters listed in the preceding table must be added in the same order. However, the parameter names can be different.

The condition expression is limited to these four parameters only. In the expression, apart from type, you can use item type for parent, item type for child, and **PSOccurrence** properties. Also, you can read the name of the parent or name of the child.

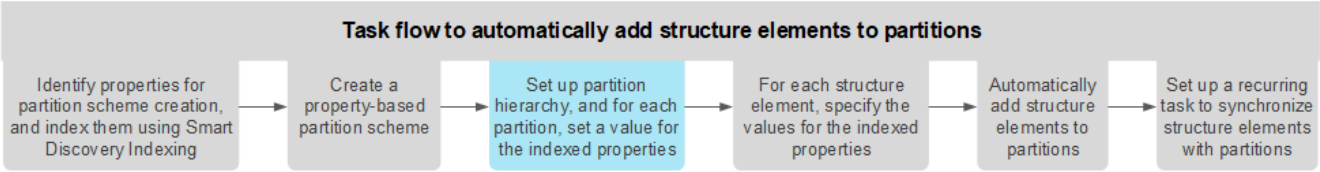
The following image shows a sample condition in BMIDE:

2. Include the condition in the **Ptn0isPartitionableCondition** business constant.

Note:

Consider that the business user has a partition scheme where the top-level and the lower-level structure elements are members of two different partitions. If they have variability on partitions and the partitions are being configured out, they might have a mismatch between the structure and the 3D viewer. To avoid this situation, we recommend you to use an appropriate condition in the **Ptn0IsPartitionableCondition** business object constant. Thus, under the same scheme, children of an assembly should not be the direct and indirect members of different partitions.

Set up partition hierarchy



© 2025 Siemens

You can either create a partition hierarchy manually in Active Workspace or run the **create_partitions** utility as follows:

```
create_partitions -u=admin_user -p=admin_user_password -g=dba
-product_id=item_id_of_top_item -file=path_to_input_file
```

OR

```
create_partitions -u=admin_user -p=admin_user_password -g=dba
-template_id=template_id -file=path_to_input_file
```

You can create partitions for a product or within a partition template. If you create partitions within a partition template and do not specify the `template_id`, a new template is created with the specified ID. Partitions are created based on the details provided in the the input *.CSV* file. Subsequently, to create additional partitions for the same product or partition template, you may run the **create_partitions** utility with a new or updated *.CSV* file.

For detailed description of each argument of the **create_partitions** utility, refer to its help using the Teamcenter command prompt:

```
create_partitions -h
```

If you choose to run the **create_partitions** utility with a *Product ID*, use a *.CSV* file similar to the one shown in the following sample:

© 2025 Siemens

#==								
# Date Name Description of Change								
# HISTORY$								
# HELP:								
#								
# NOTE:DO NOT USE DOUBLE QUOTE IN THIS FILE								
#								
# The input file to c	Type	ptn0partition_scheme_type	index	parent_inde	object_type	object_name	object_desc	ptn0partition_id
# Description of the columns:								
# 1)Type - This column contains the type of object that wanted to create.								
#It can be								
# a)PartitionScheme								
# b)Partition								
# 2)ptn0partition_scheme_type - This column contains the scheme type under which user wants to create the partitions.								
# 3)index - The index is use to create the partition hierarchy.								
# 4)parent_index - The parent_index is use to create the partition hierarchy.								
# 5)Type - This column contains the type of object that wanted to create.								
# 6)object_type - Type of partition scheme/partition.								
# 7)object_name - Name for the partition scheme/partition.								
# 8)object_desc - Description for the partition scheme/partition.								
# 9)ptn0partition_id - Partition Id for the partition.								
# 10)Awb0DesignElement.bl_occ_seq_no - This column is for runtime properties. Similarly user can define multiple runtime properties which are indexed.								
#a) The multiple runtime property values can be define by "|" seperator. example: 10|20|30								
#==								
#								
#Type	index	ptn0partition_scheme_type	parent_index	object_type	object_name	object_desc	ptn0partition_id	Awb0DesignElement.bl_occ_seq_no
PartitionScheme	0	NA	NA	C2SchemeW	AutoScheme	SchemeDes	N	
Partition	1	C2SchemeWithValidProps	0	Ptn0Functio	FunPartition1	PtnDesc	PTN100001	10|12
Partition	2	C2SchemeWithValidProps	0	Ptn0Functio	FunPartition2	PtnDesc	PTN100002	40
Partition	3	C2SchemeWithValidProps	0	Ptn0Functio	FunPartition3	PtnDesc	PTN100003	20|30
Partition	4	C2SchemeWithValidProps	0	Ptn0Functio	FunPartition4	PtnDesc	PTN100004	15
Partition	5	C2SchemeWithValidProps	0	Ptn0Functio	FunPartition5	PtnDesc	PTN100005	25
Partition	6	C2SchemeWithValidProps	1	Ptn0Functio	FunPartition6	PtnDesc	PTN100006	10|60
Partition	7	C2SchemeWithValidProps	1	Ptn0Functio	FunPartition7	PtnDesc	PTN100007	30
Partition	8	C2SchemeWithValidProps	1	Ptn0Functio	FunPartition8	PtnDesc	PTN100008	10
Partition	9	C2SchemeWithValidProps	2	Ptn0Functio	FunPartition9	PtnDesc	PTN100009	
Partition	10	C2SchemeWithValidProps	6	Ptn0Functio	FunPartition1	PtnDesc	PTN100010	100
Partition	11	C2SchemeWithValidProps	6	Ptn0Functio	FunPartition1	PtnDesc	PTN100011	60
Partition	12	C2SchemeWithValidProps	6	Ptn0Functio	FunPartition1	PtnDesc	PTN100012	
Partition	13	C2SchemeWithValidProps	7	Ptn0Functio	FunPartition1	PtnDesc	PTN100013	90
Partition	14	C2SchemeWithValidProps	7	Ptn0Functio	FunPartition1	PtnDesc	PTN100014	40|90
Partition	15	C2SchemeWithValidProps	10	Ptn0Functio	FunPartition1	PtnDesc	PTN100015	

If you choose to run the **create_partitions** utility with a *Template ID*, use a *.CSV* file similar to the one shown in the following sample:

© 2025 Siemens

#===								
# Date Name Description of Change								
# HISTORY$								
# HELP:								
#								
# NOTE:DO NOT USE DOUBLE QUOTE IN THIS FILE								
#								
# The input file to c	Type	ptn0pa	index	parent_inde	object_type	object_name	object_des	ptn0partition_id
# Description of the columns:								
# 1)Type - This column contains the type of object that wanted to create.								
#It can be								
# a)PartitionScheme								
# b)Partition								
# 2)ptn0partition_scheme_type - This column contains the scheme type under which user wants to create the partitions.								
# 3)index - The index is use to create the partition hierarchy.								
# 4)parent_index - The parent_index is use to create the partition hierarchy.								
# 5)Type - This column contains the type of object that wanted to create.								
# 6)object_type - Type of partition scheme/partition.								
# 7)object_name - Name for the partition scheme/partition.								
# 8)object_desc - Description for the partition scheme/partition.								
# 9)ptn0partition_id - Partition Id for the partition.								
#===								
#								
#Type	ptn0partition_scheme_type	index	parent_index	object_type	object_name	object_desc	ptn0partition_id	
PartitionTemplate	NA	NA	NA	Ptn0Partitio	PartitionTemp	TemplateDesc		
PartitionScheme	0	0	0	Ptn0Scheme	FunScheme	PhySchemeD	NA	
PartitionScheme	0	0	0	Ptn0Scheme	PHYScheme	FunSchemeD	NA	
Partition	Ptn0SchemeFunctional	1	0	Ptn0Functio	FunPartition1	PtnDesc	PTN0001	
Partition	Ptn0SchemeFunctional	2	0	Ptn0Functio	FunPartition2	PtnDesc	PTN0002	
Partition	Ptn0SchemeFunctional	3	2	Ptn0Functio	FunPartition3	PtnDesc	PTN0003	
Partition	Ptn0SchemePhysical	4	0	Ptn0Design	PhyPartition1	PtnDesc	PTN0004	
Partition	Ptn0SchemePhysical	5	4	Ptn0Design	PhyPartition1	PtnDesc	PTN0005	

Caution:

The column headers in the *.CSV* file must be the internal names of the properties.

For partitions created manually, you must set values for the indexed properties. These values will be used later to populate the partitions correctly with structure elements. For this:

1. In Active Workspace, open the structure.
2. From **Partition Scheme**, select the persistent partition scheme that you added, for example, *System*.
3. Select the **Partition Recipe Props** column arrangement from **Table Settings** > **Arrange**.
4. For each leaf-level partition, enter the values in the column that shows the indexed properties (for example, *PS_PTY10* and *PS_PTY20*). For example, for *Dent Repair Kit*, enter the value as **10B** for **PS_PTY10** and **6C, 6E** for **PS_PTY20**. You must enter the values for all indexed properties.

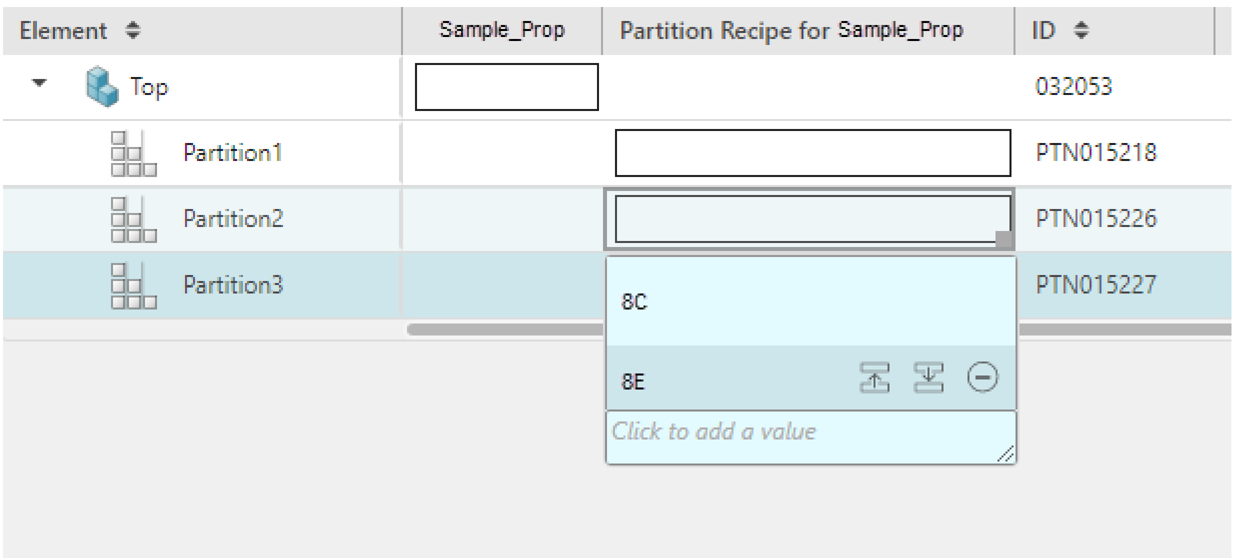
© 2025 Siemens

You can populate the recipes on partitions in one of the following ways:

- While using the **create_partitions** utility, in the input CVS file, you can separate multiple recipes by using a '|'.

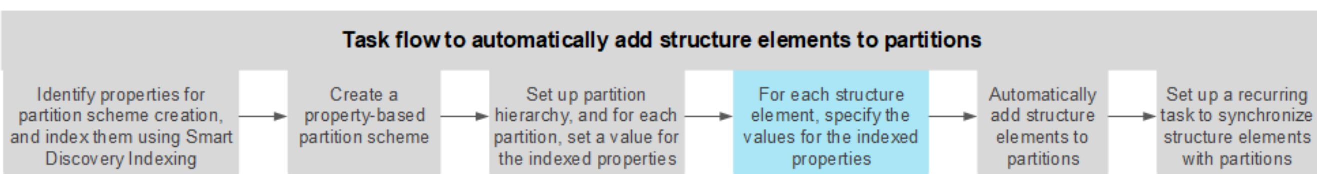
- While using the Active Workspace UI, the properties that are added on the partitions to input recipes are array properties and they accept multiple recipe entries. If you populate the recipes interactively with '|' separated values, they are considered as one recipe.

  For example, in the following image, **Partition Recipe for Sample_Prop** is an array property that accepts multiple recipe entries:



To view the inferred recipe, select a partition, and go to its **Overview** tab. The criteria are listed under **AGGREGATE PARTITION RECIPE**. Only the structure elements that meet this criteria are added to the selected leaf partition.

Set property values to include structure elements into partitions



For each structure element to be included within a partition, verify that the values for the indexed properties are specified. For a newly added or update structure element, specify the property values as follows:

1. In Active Workspace, open the structure.

© 2025 Siemens

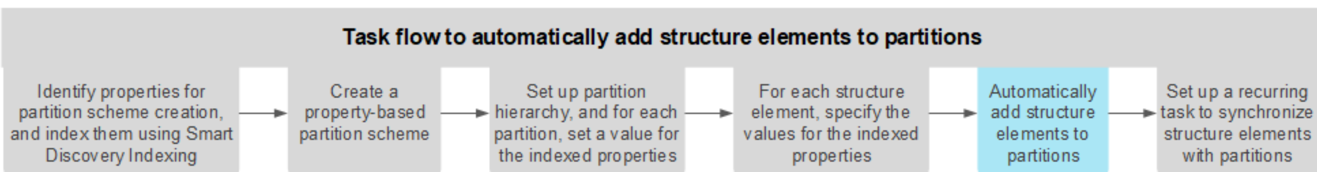
2. Set **Partition Scheme** to **None**.

3. Select the **Partition Recipe Props** column arrangement from **Table Settings** > **Arrange**.

4. For a structure element, in the column that shows the indexed properties, enter the values. For example, for *Car Shampoo*, enter the value as **10B** in **PTY10** and **7D** in **PTY20**.

   Later, when you set up the automatic population of partitions, these values are matched with the property values of the partitions. And, accordingly the elements are populated within different partitions.

Automatically add structure elements to partitions



Before populating the partitions with structure elements, ensure that you have set the values of the indexed properties on partitions and structure elements correctly. Once the elements are organized within partitions, you cannot change the values of these properties.

To add the structure elements to the respective partitions, run the **populate_property_based_partition_members** utility on Teamcenter command prompt:

```
populate_property_based_partition_members -u=admin_user
-p=admin_user_password -g=dba -product=item_id_of_top_item
```

For detailed information about the different argument of this utility, see its help on the Teamcenter command prompt:

```
populate_property_based_partition_members -h
```

When added to partitions, the structure elements are called *partition members*.

> Tip:
>
> It is recommended that while creating partitions based on the indexed properties, you choose partition-owned memberships.

When you create a partition scheme that has properties specified, memberships are created automatically based on these property values.

© 2025 Siemens

When you add structure elements to partitions, the business user can create new partitions, partition hierarchy, and set or update partition recipes. Teamcenter automatically synchronizes the memberships based on the new partition recipes.

For this, it is critical that the user fully populates the recipes by specifying a value for each supported term or property in the recipe. If they do not specify all values in the recipe, the memberships are not created for the partition. Once they populate all the values, Teamcenter automatically synchronizes the memberships based on the new partition recipe.

> Note:
>
> Once the structure elements have been added to the partitions, the users cannot move or delete a partition.

Partition-owned membership vs. member-owned membership

For member-owned membership, the value of the **Ptn0IsOwningMemberships** business object constant in Business Modeler IDE (BMIDE) is set to **false**. This is applicable for the *System* and *Functional* partitions. Conversely, for a partition-owned membership, the value of the **Ptn0IsOwningMemberships** business object constant in BMIDE is set to **true**. This is applicable for the *Zone* and *Manufacturing* partitions.

If you are the owner of an assembly, you create *member-owned* memberships while organizing your data in a partition hierarchy. Thus, you have the access privileges to perform various tasks on your data, such as create, copy, and paste.

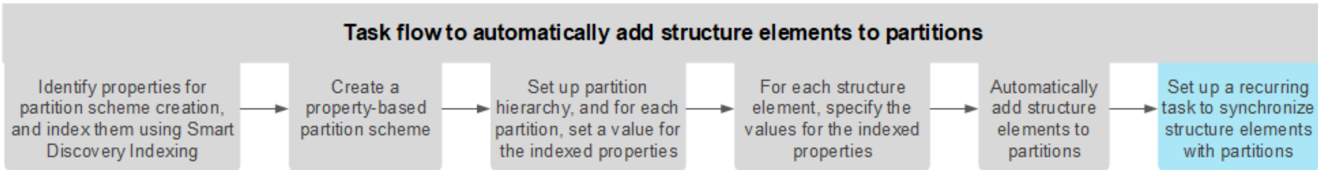
If other users need to organize your data in a partition hierarchy, they create a *partition-owned* memberships for your data. This is because they do not modify the data, they just organize the data. Thus, partition-owned memberships do not require you to have any access privileges on the data.

When you create partitions based on the indexed properties, the partitions own the data. Therefore, it is recommended to choose partition-owned memberships while creating partitions. If you choose member-owned memberships and the **populate_property_based_partition_members** utility is run, for your data, the **Last Modifying User** and the **Last Modified Date** fields get updated to the Admin user who has run the utility. Instead, when you choose partition-owned memberships and the utility is run, the data is not modified in any way. In this case, the owner of the data, the Last Modifying Use, and the Last Modified Date remain as they are.

Essentially, for member-owned memberships, Teamcenter checks the privileges against the data. Conversely, for partition-owned memberships, Teamcenter checks the privileges against the partition.

© 2025 Siemens

Set up synchronization of structure elements with partitions



To start the synchronization process, perform the following tasks:

Task	Action
Run Active Workspace indexing service	In Services, check if the status is **Running**. Else, start the service.
Run TcFTSIndexer	Start TcFTSIndexer as a service: *TC_ROOT*/TcFTSIndexer/bin/runTcFTSIndexer -background
Start object indexing	*TC_ROOT*/TcFTSIndexer/bin/runTcFTSIndexer -task=objdata:sync -interval=<time_interval_in_seconds>
Start Smart Discovery index synchronization	*TC_ROOT*/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:sync -interval=<time_interval_in_seconds> This is a recurring task to regularly update the Smart Discovery index in Solr and Teamcenter.
Start partition synchronization	*TC_ROOT*/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:updatepartition -interval=<time_interval_in_seconds> This is a recurring task to regularly update the partitions with structure elements (partition members).

© 2025 Siemens

5. Change the default partition behavior

You can change the default behavior of partitions as per your site requirements. To do so, set the following business object constants in Business Modeler IDE (BMIDE):

Business Object Constant	Description
Ptn0IsOwningMemberships	If set to **true**, the membership, that is, the access privileges to perform various tasks, such as create, copy, and paste, are owned by partitions. Else, the membership are owned by the structure elements (BOM lines) within the partitions. The actual value of this constant may be overridden for each partition subtype. By default, this is set to **true** for manufacturing and zone partitions. For other partitions, it is set to **false**.
Ptn0SchemeAllowedTypes	Define the partition types allowed for different partition schemes. For example, **Ptn0Functional** and **Ptn0System** are allowed in the functional partition scheme.
Ptn0IsMultipleBomlineMembershipAllowedInScheme	By default, a structure element can be added to multiple partitions within the same partition scheme. Set this constant to **false** if you want an element to be added only to one partition within a scheme.
Ptn0ClosureRulesToRestrictMemberAuthoring	Specify which structure elements can be added within a partition by listing closure rule names (separated by comma) that are available in the **Ptn0ClosureRulesForBOMPartitionAuthoring** preference. A structure element is not included in a partition if even one of the closure rules has the value set to **SKIP**. Note: The business object constant **Ptn0ClosureRulesToRestrictMemberAuthoring** is obsolete.
Ptn0IsPartitionable	Specifies if a structure element is partitionable. Takes one or more conditions as an input. If the evaluation of the conditions is **true**, a BOM line can be added as a child to a partition. If the evaluation is **false**, a BOM line cannot be added to a partition.

© 2025 Siemens

© 2025 Siemens

6. Set a primary partition scheme

Set a primary partition scheme for a business object at the site level by using the **PTN_DefaultPrimaryScheme** preference.

The format of this preference is:

```
Type1 = Scheme1
Type2 = Scheme2
```

For example:

```
Item = Ptn0SchemeFunctional
Design = Ptn0SchemePhysical
```

By default, when a user opens a product of a specific type in Active Workspace, it opens the product within the partition scheme set as its primary scheme. However, the user can override this preference so that products open in a different scheme. However, the primary scheme is still the one that you have set.

© 2025 Siemens

© 2025 Siemens

7. Create partitions from a partition template

You can create partitions for a product from a partition template that is available in Teamcenter by using the **populate_partitions** utility.

Example:

```
populate_partitions -u=tcadmin -p=tcadmin -g=dba -action=clone
-srcModel=PT000001 -trgtItemId=000035
```

Here, **srcModel** is the partition template ID, and **trgtItemId** is the product structure ID. On executing the **populate_partitions** utility, partitions are created for the product structure with the ID as **000035** from the partition template **PT001001**.

Currently, you can only use the value **clone** in the **-action** argument. Run `populate_partitions -h` in the Teamcenter command prompt for detailed information regarding this utility.

After creating partitions, you can **add structure elements to the partitions**.

© 2025 Siemens

© 2025 Siemens

8. Populate partitions with structure elements

After **creating partitions from a partition template**, you can add structure elements to the partitions. A structure element added to a partition is called its *member*. To add partition members, run the **populate_partition_members** utility.

Example:

Consider the following structure where you set the partition schemes and partitions in the **Template Action** field of the structure elements that you want to add to the partitions. You can also set the partition schemes and partitions in any other field that is of the type **string**.

- Crosskart
  - Chassis
    - Foot board (Template Action > Functional=Ptn123, Spatial=Ptn456)
    - Front bumper
    - Rear bumper
    - ...
  - Wheel assembly
    - Wheel (Template Action > Spatial=Ptn456)
    - Rim

If you run the **populate_partition_members** utility for *Crosskart* as follows:

```
populate_partition_members -u=tcadmin -p=tcadmin -g=dba
-item=Crosskart -partition_member_property="Template Action"
```

Foot board is added to the partition *Ptn123* under the *Functional* partition scheme and to the partition *Ptn456* under the *Spatial* partition scheme. *Wheel* is added to the partition *Ptn456* under the *Spatial* partition scheme.

Run `populate_partition_members -h` in the Teamcenter command prompt for detailed information regarding this utility.

© 2025 Siemens

© 2025 Siemens

9. Specify access privileges for partitions

For a user to create partition schemes and partitions, and to do the various tasks related to a partition, you must provide the user the **ADD_CONTENT** and **REMOVE_CONTENT** access privileges to add or remove *partition members*. The structure elements (BOM lines) added to a partition are called its members. To modify other partition properties, you must grant the users the **WRITE** access.

Depending on the access privileges of a user, Teamcenter grants or denies access while the user tries to perform a task. The **Ptn0IsOwningMemberships** business object constant in Business Modeler IDE (BMIDE) decides whether the membership is owned by a partition or by a partition member. If set to **true**, the membership, that is, the access privileges to perform various tasks, such as create, copy, and paste, are owned by partitions. Else, the membership privileges lie with the partition members.

The following table lists the checks that Teamcenter performs for access privileges when the user performs certain actions:

User action	Membership owned by	Access checked on	Required privileges	Access privilege checked on	Grant if
Partition schemes					
Create partition schemes	NA	Product structure ownership with current site	**ADD_CONTENT**	The BOM view belonging to the opened product	TRUE
Partitions					
Create root partitions	NA	Product structure	**ADD_CONTENT**	The BOM view belonging to the opened product	TRUE
Add child partitions	NA	Product structure and child partitions	**ADD_CONTENT** for product structure and **WRITE** for child partitions	The **ADD_CONTENT** access privilege is checked on the BOM view belonging to the opened product. The **WRITE** access privilege is checked on the child partition.	TRUE
Remove child partitions	NA	Product structure and child partitions	**REMOVE_CONTENT** for product structure and **WRITE** for	The **REMOVE_CONTENT** access privilege is checked on the BOM view	TRUE

© 2025 Siemens

User action	Membership owned by	Access checked on	Required privileges	Access privilege checked on	Grant if
			child partitions	belonging to the opened product. The **WRITE** access privilege is checked on the child partition.	
Partition members					
Add partition member	Partition	Product structure and parent partition	**ADD_CONTENT**	The **ADD_CONTENT** access privilege is checked on the BOM view belonging to the opened product.	TRUE
	Member	Product structure and member	**ADD_CONTENT** for product structure and **WRITE** for member	The **ADD_CONTENT** access privilege is checked on the BOM view belonging to the opened product. • For member-owned partitions, the **WRITE** access is checked on the BOM View Revision (BVR) of the parent item when an item is added as a member of a partition. • For partition-owned partitions, the **WRITE** access check is not performed.	TRUE
Remove partition member	Partition	Product structure and parent partition	**REMOVE_CONTENT**	The **REMOVE_CONTENT** access privilege is checked on the BOM view	TRUE

© 2025 Siemens

User action	Membership owned by	Access checked on	Required privileges	Access privilege checked on	Grant if
				belonging to the opened product.	
	Member	Product structure and member	**REMOVE_CONTENT** for product structure and **WRITE** for member	The **REMOVE_CONTENT** access privilege is checked on the BOM view belonging to the opened product. • For member-owned partitions, the **WRITE** access is checked on the BOM View Revision (BVR) of the parent item when an item is removed as a member of a partition. • For partition-owned partitions, the **WRITE** access check is not performed.	TRUE

© 2025 Siemens

© 2025 Siemens

10. Set up partition configuration

You can add a valid revision rule and then set it as the value for the **PTN_Default_Partition_RevRule** preference. This preference specifies the default revision rule applied to configure a partition. The specified revision rule must be valid for structure elements.

For the revision rule to be valid, you must not add any clauses other than **Working**, **Status**, **Date**, and **Unit Number**. Thus, you must add a revision rule that contains clauses only from this list and then set this revision rule as a value for the preference.

If there is no valid revision rule available when you try to configure a list of partitions to be displayed in the **Filter** panel, the Teamcenter server might crash.

© 2025 Siemens