TEAMCENTER

Structure Indexing Using Smart Discovery

Teamcenter 2412

SIEMENS

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

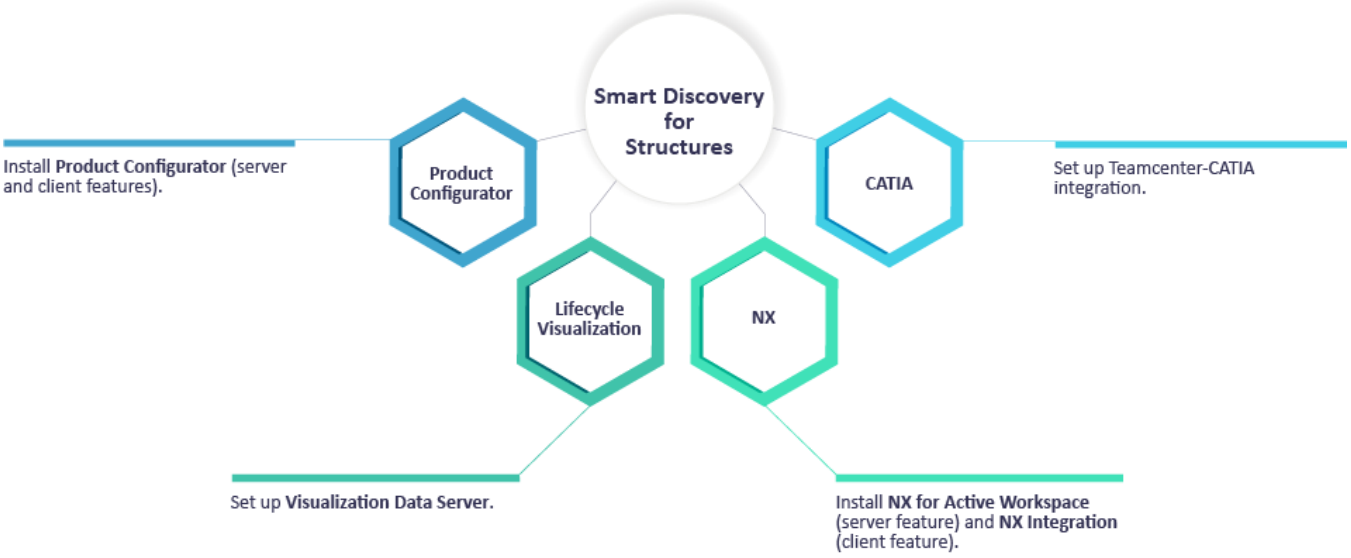
1. About Smart Discovery for Structures

With the increase in a product's complexity and variability, the amount of data associated with it is huge. With Smart Discovery for Structures, you can index product structures by using the **Smart Discovery Indexing** functionality so that users can find the correct product definition required to do their work. Indexing structures makes the search faster, and provides advanced structure filtering and configuration capabilities.

> Example:
>
> To fix a design issue, an engineer wants to work with the engine assembly of a car. When the car structure is indexed, the engineer can search within the structure to find the engine assembly without the need to expand the entire structure. The engineer can further filter the search result to locate all elements that are at a proximity of 10 mm from the engine block.

Users can save an indexed structure within a session, which retains the filters and configurations applied on it. You can configure a session as per your site requirements. For example, you can choose to load a session with static data instead of the latest data or you can restrict users from overwriting a session.

A session can be opened in different applications such as NX, CATIA, and Lifecycle Visualization when these are integrated with Teamcenter.Additionally, the integration with Product Configurator allows users to configure a structure on which variability is set by using Product Configurator variants in case they do not wish to use the classic variants for configuring the structure. Ensure that you have set up these applications.



Where do I go from here?

Business User	See Structure Management on Active Workspace — Usage.

	With Smart Discovery for Structures, you can filter structures. You can also apply some additional configurations to structures.
Administrator	
I am new to Smart Discovery Indexing.	Smart Discovery Indexing is the new method to index structures. Cacheless search and Active Content Structure Indexing are the older methods. Learn more about **Smart Discovery Indexing**.
How do I index structures by using Smart Discovery Indexing?	Look at the **Smart Discovery Indexing process flow**.
I am facing some issues while setting up Smart Discovery Indexing.	Refer how to **troubleshoot the Smart Discovery Indexing issues.**.
How do I know that I have set up Smart Discovery Indexing correctly?	You can perform certain **verification steps** to ensure that you have set up Smart Discovery Indexing correctly.
Where can I get a list of preferences that I need to set for Smart Discovery for Structures?	There are several sources from which you can retrieve a list of preferences. You can view the list in Administration data report, Teamcenter rich client, Active Workspace, and raw XML file.

2. About Smart Discovery Indexing

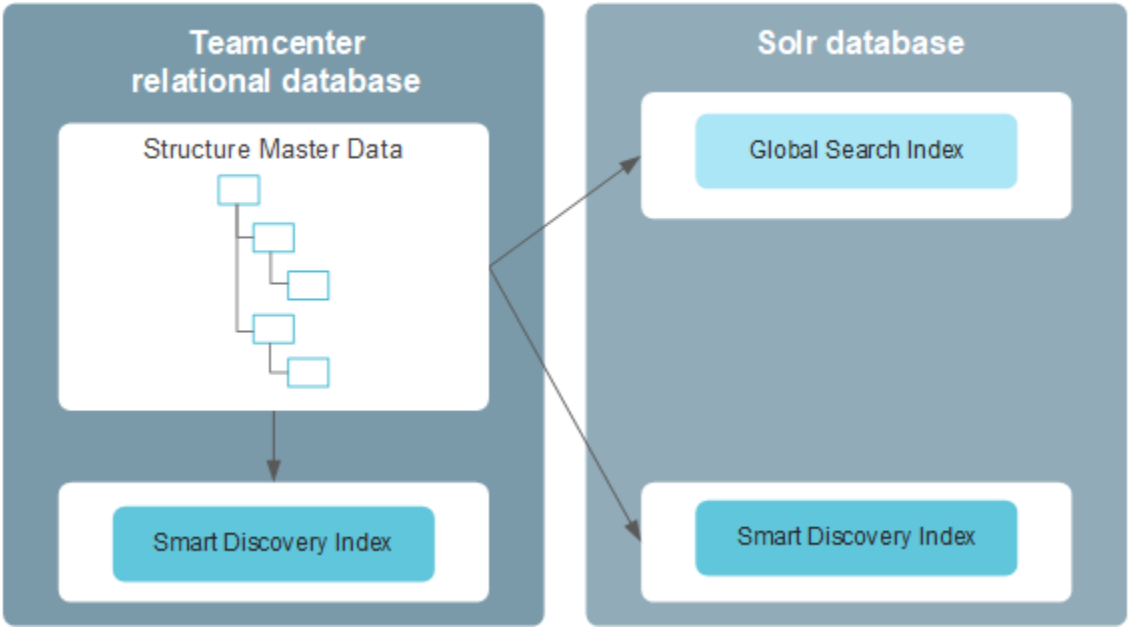
You index structures for faster search, filtering, and configuration of structures. Users can search for elements within a structure by entering a keyword, by specifying certain attributes, or by performing a spatial search.

For any structure, indexes are stored within:

- A global search index used for keyword and attribute search of item revisions.
- A structure index used for finding elements within a structure. Structure indexing can be performed either by using the existing *Active Content Structure Indexing* or *Cacheless Search* functionalities or by using the new *Smart Discovery Indexing* functionality.

  > Note:
  >
  > The Smart Discovery Indexing functionality provides advanced filtering and configuration capabilities such as filter by proximity, filter by volume, configure by proximity, and configure by selection. To use these advanced capabilities, index the structures by using Smart Discovery Indexing. However, it does not support Massive Model Visualization (MMV). For MMV, you must continue to index structures by using the existing Active Content Structure Indexing functionality.
  >
  > The Teamcenter database may contain a mix of structures indexed using cacheless search and Smart Discovery Indexing. However, each structure can have only one type of index, cacheless or smart discovery.

For structures indexed using the Smart Discovery Indexing functionality, the indexes are stored in the global search index and the Smart Discovery index. The global search index is maintained in the Solr search engine, while the Smart Discovery index is maintained in both Solr as well as in the Teamcenter relational database.



Database	Index	Description
Solr	Global search index	Indexes item revision data such as attributes and documents.
Relational database and Solr	Smart Discovery index	Indexes structure related data such as occurrence attributes and spatial data.

Both the indexes are used for searching and filtering in the context of a structure.

3. Install Smart Discovery Indexing through TEM

Before you install Smart Discovery Indexing, ensure that you have set up Indexing Engine and Indexer. Additionally, if your business requires to find content within files, you must set up File Content Indexing. For detailed information on how to set these up, see *Understanding the indexing process and determining your indexing strategy*.

To install Smart Discovery Indexing:

1. Launch the installed (TEM) for the corporate server.

2. In the **Maintenance** panel, select **Configuration Manager** and click **Next**.

3. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration** and click **Next**.

4. In the **Old Configuration** panel, select the corporate server configuration and click **Next**.

5. If Indexing Engine and Indexer are not installed in your Teamcenter setup:

   a. In the **Feature Maintenance** panel, select **Add/Remove Features** and click **Next**.

   b. In the **Features** panel, select the following features and click **Next**.

      - **Base Install → Active Workspace → Server Extensions → Active Content Structure**.
      - **Base Install → Active Workspace → Indexing Server → Active Workspace Indexing Engine**.
      - **Base Install → Active Workspace → Indexing Server → Active Workspace Indexer**.

   c. Enter the required values in the subsequent panels for Indexing Engine and Indexer and click **Next**.

6. If Indexing Engine and Indexer are installed in your Teamcenter setup:

   In the **Feature Maintenance** panel, select **Update Active Workspace Indexer settings** under **Active Workspace Indexer** and click **Next**.

7. In the **Active Workspace Indexer Type Selection** panel, select the following and click **Next**:

   - **Active Workspace Object Data Indexer**

- **Smart Discovery Indexer**

8. In the **Confirmation** panel, click **Start** to being the installation.

After the installation is complete, you can **index structures** for a faster search and for advanced structure filtering and configuration capabilities.

4. Install Smart Discovery Indexing using Deployment Center

Add Smart Discovery Indexing to your existing Teamcenter environment through a series of tasks from selecting the application and entering configuration parameters to generating and running deployment scripts.

Prerequisites

Before you install Smart Discovery Indexing, ensure that you have set up Indexing Engine and Indexer. Additionally, if your business requires to find content within files, you must set up File Content Indexing. For detailed information on how to set these up, see *Understanding the indexing process and determining your indexing strategy*.

Procedure

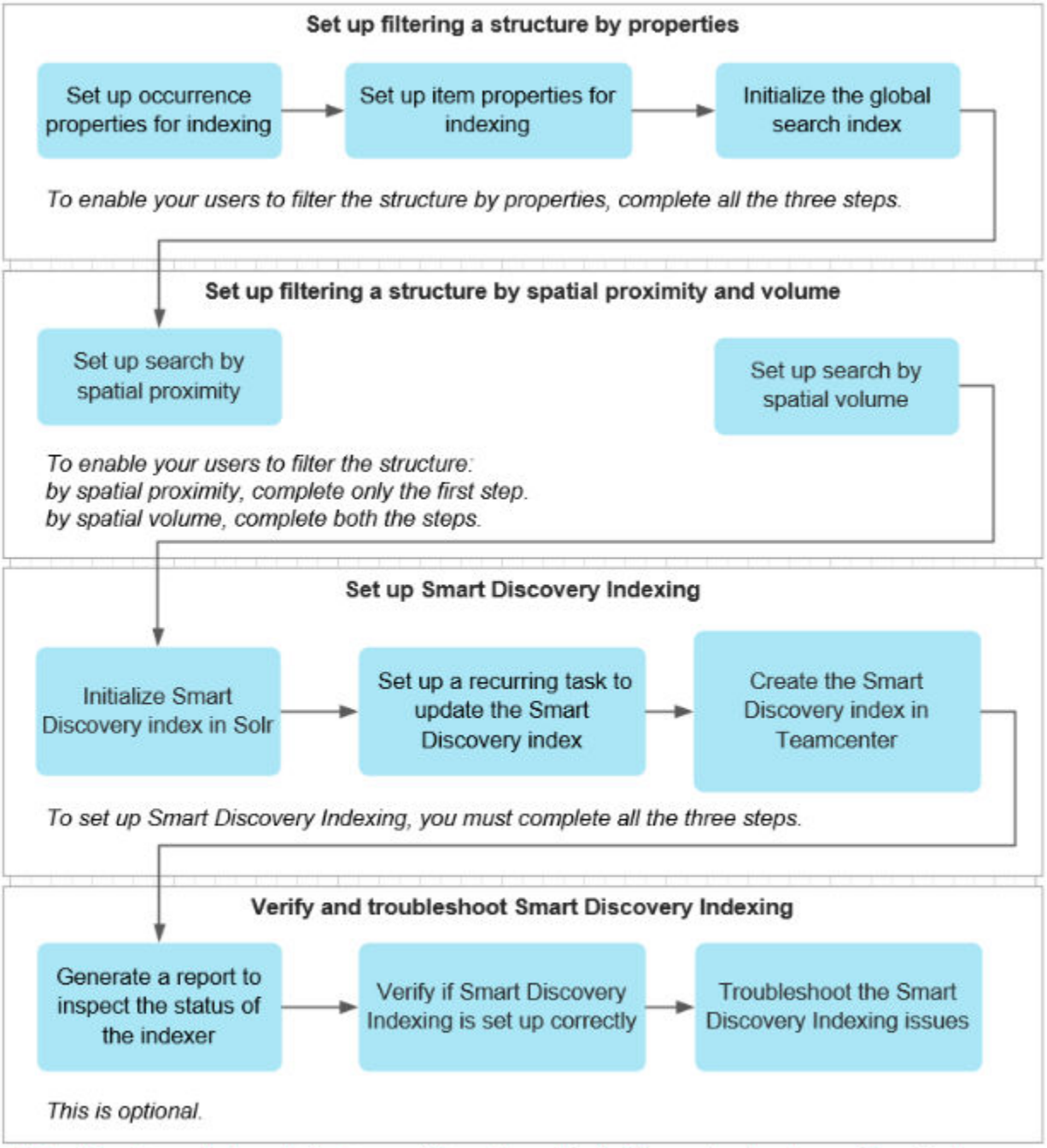
1. Log on to Deployment Center.

2. In the **Applications** task, select **Teamcenter→Active Workspace→Active Content** from the list of available applications.

3. In the **Components** task, fill in the required information so that the state of all components is **100%**. For details on filling up information related to Indexing Engine, Indexer,and file content indexing, see *Install Indexing Engine (Solr)*, *Install Indexer (TcFTSIndexer)*, and *Install asynchronous file content indexing*.

4. In the **Deploy** task, generate the deployment scripts and proceed with the deployment based on the instructions displayed on the screen.

5. Tasks to set up Smart Discovery Indexing

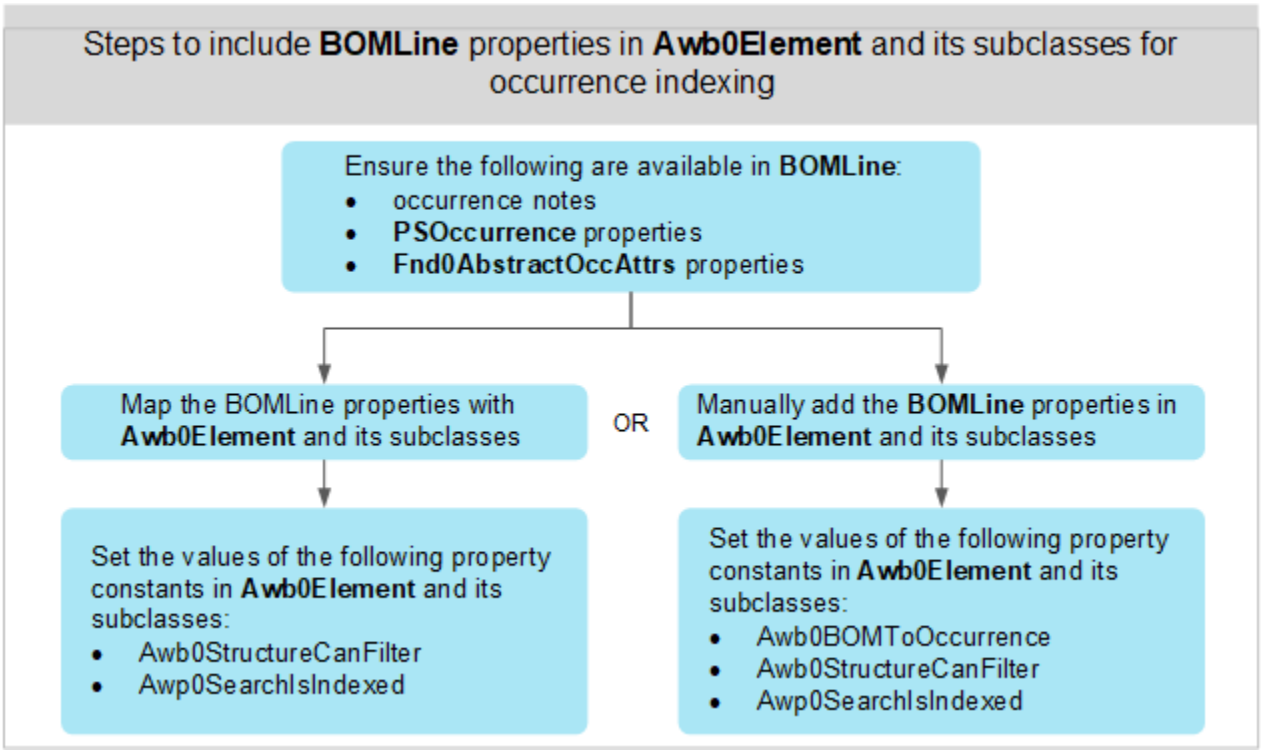


Note: *If you have deployed structure partitions, the option to filter a structure by partitions is always available to the user.*

6. Set up occurrence properties for indexing

For users to filter structures by selecting information related to occurrence, you set up properties of the following for indexing:

- Occurrence notes
- Properties of the **PSOccurrence** business object. The properties can be of the type **Integer**, **String**, **Array**, **Typed Reference**, **Compound**, **Float**, and **Boolean**.
- Properties of the **Fnd0AbstractOccAttrs** business object. The properties can be of the type **Integer**, **String**, **Array**, **Compound**, **Float**, and **Boolean**.



To set up occurrence properties for indexing:

1. Launch Business Modeler IDE (BMIDE).
2. Open an existing BMIDE template project or create a new one, for example, **a5occurrenceproperties**.

   Ensure that **aws2** and **activeworkspacebom** are selected as the dependent templates.

3. Set up required properties for indexing.

   - For occurrence notes:

     a. Click **Extensions** > **Options** > **List of Note Types** to view the list of available occurrence notes.

     b. Locate the **BOMLine** business object, and verify that the occurrence note that you want to index is listed as one of the properties of **BOMLine**. You will later map these properties with the **Awb0Element** and its sub-classes.

   - For **PSOccurrence** properties:

     Go to the **Properties** tab of **PSOccurrence**, and verify that the property you want to index is available here. If is not there, you can add a new **Persistent** or **Compound** property.

     > Note:
     >
     > You cannot set a property of the type **Array** for indexing if the Teamcenter version is 13.2 or lower.

     The newly added properties are automatically added to **BOMLine**.

   - For **Fnd0AbstractOccAttrs** properties:

     a. Create a child business object under **Fnd0AbstractOccAttrs**.

     b. In the **Properties** tab of the newly created child business object, create **Persistent** properties of the type **String**, **Integer**, **Double**, or **Boolean**.

        The newly added properties are automatically added to **BOMLine**.

4. Map the **BOMLine** properties to a subclass of **Awb0Element**:

   a. By default, the subclass **Awb0DesignElement** is already marked as indexable. To map the **BOMLine** properties to a subclass other than **Awb0DesignElement**, open the subclass, for example, **Awb0PositionedElement**, and set the values of the following business object constants:

Business object constant	Value
Awb0BOMToOccurrence	**BOMLine**
Awp0SearchIsIndexed	**true**

   b. The following steps show how to map a **BOMLine** property with **Awb0DesignElement**. You may follow the same steps to map a property with other subclass of **Awb0Element**.

A. In the **Properties** tab of **BOMLine**, select the property, and click **Map Property**.

B. In **Map To Type**, select **Awb0DesignElement**.

C. Click **OK** first and then click **Finish**.

D. Open **Awb0DesignElement** and in the **Property Constants** tab, set the following values:

Property constant	Value
Awb0StructureCanFilter	**true**
Awp0SearchIsIndexed	**true**
Awp0SearchFilterPriority	Integer The integer value is used for sorting the property. The lower the value, the higher is its precedence and it will be displayed higher on the item revision filter list on the **Filter** panel.

- OR -

Manually add the configured runtime properties of **BOMLine** for the following as runtime properties of **Awb0DesignElement**:

- Occurrence note
- **PSOccurrence** property
- A property of a subclass of **Fnd0AbstractOccAttr**

Select each property in **Awb0DesignElement**, go to its **Property Constants** tab, and set the following values:

Property constant	Value
Awb0BOMToOccurrence	Name of the configured runtime property of **BOMLine** for occurrence note, **PSOccurrence** property, or property of subclass of **Fnd0AbstractOccAttr**.
Awb0StructureCanFilter	**true**
Awp0SearchIsIndexed	**true**

5. To index **Typed Reference** properties of the **PSOccurrence** business object, select each property in all those business objects where you want to make the following properties available.

For example, to make the properties available for **Awb0DesignElement**, go to its **Property Constants** tab, and set the following values:

Property constant	Value
Awb0BOMToOccurrence	Name of the selected **Configured Runtime** or **Compound** property of **BOMLine**.These properties can be **Typed Reference** properties of **PSOccurrence**, for example, **bl_occ_occ_type**.
Awb0StructureCanFilter	**true**
Awp0SearchPropFromRefType	**name**
Awp0SearchRefTypeNames	**PSOccurrenceType**

You can only include the first-level persistent properties of **PSOccurrence** that are directly located under **Typed Reference**. If a property is flagged for localization, the **Filter** panel will display the data of the master locale even if the user is logged on using a different locale.

6. Click **BMIDE** > **Save Data Model**.

7. Click **BMIDE** > **Generate Software Package**.

   Make a note of the path in **Target folder**.

8. Navigate to the folder on your computer where you have installed Teamcenter, and launch TEM.

9. Click **Next** until you reach the **Features** panel.

10. Click **Browse** and navigate to the location of the **Target folder** that you had noted down earlier.

11. Select the custom template under **Extensions**, for example, **a5occurrenceproperties**.

    Additionally, select **Extensions** > **Occurrence Attribute Object** in the **Features** panel.

12. To deploy the changes, run the **bmide_modeltool** utility as follows:

    bmide_modeltool -u=*admin_user_name* **-p|-pf=***admin_user_password***-g=dba -tool=all -mode=install -target_dir=***TC_DATA*

13. Clicking **Next** until you reach the **Confirmation** panel and then click **Start**.

14. Update and merge the schema file.

15. (Optional) Import the properties that you set up for indexing as columns, and merge the columns with an existing column configuration.

16. Run `TC_ROOT/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:sync`.

    This step can take time depending on the size of the database.

17. Restart all the required services to launch Active Workspace.

18. On Active Workspace, search for a structure that is already indexed using Smart Discovery. A Smart Discovery indexed structure has an indicator next to it.

    Next, click **Filter**. Verify if the newly added runtime property is displayed on the **Filter** panel. If the properties are not listed, follow the **troubleshooting steps** to fix the issue.

    > Note:
    >
    > If no structures are indexed yet, first **create** the Smart Discovery index and then set up the **recurring** task to update the index.

At this point, you can choose to **set the item revision properties for indexing**, if not already set, after which you can close BMIDE.

7. Set up item properties for indexing

The item revision properties that you set up for indexing appear as filters in the **Filter** panel of global search and in-context search. For this, you first set the properties to appear as filters in the global search **Filter** panel. Next, set up the item revision properties of the type **Persistent** or **Compound** for indexing so that the properties are listed as filters for in-context search. For a **Compound** property, ensure that it is not a dynamic compound property (DCP) and that it does not contain a **Runtime** property as its source.

To set up item properties for indexing:

1. In BMIDE, create a new template project, for example, **a5occurrenceproperties**.

   Ensure that **aws2** and **activeworkspacebom** are selected as the dependent templates.

2. Set the properties of item revisions:

   a. Locate the **WorkspaceObject** business object.

   b. In the **Properties** tab, select a property (for example, **object_name**) and set the value of the:

      - **Awb0StructureCanFilter** property constant to **true** so that the property is included in the global search.
      - **Awp0SearchIsIndexed** property constant to **true** so that the property is displayed on the **Filter** panel.
      - **Awp0SearchFilterPriority** property constant to an integer value to sort the property. The lower the value, the higher is its precedence and it will be displayed higher on the item revision filter list on the **Filter** panel.
      - **Awp0SearchIsStored** property constant to **true** if the property is used for automatically adding structure elements to partitions. For automatic partition creation, before indexing an item revision property, index at least one **occurrence property**.
      - **Awp0IsStored** property constant to **true**.

   c. Index at least one **occurrence property**.

   d. Regenerate Global Index.

   e. Regenerate Product Index.

3. Click **BMIDE** > **Save Data Model**.

4. Click **BMIDE** > **Generate Software Package**.

   Make a note of the path in **Target folder**.

5. Navigate to the folder on your computer where you have installed Teamcenter, and launch TEM.

6. Click **Next** until you reach the **Features** panel.

7. Click **Browse** and navigate to the location of the **Target folder** that you had noted down earlier.

8. Select the custom template under **Extensions**, for example, **occnotes**.

9. Clicking **Next** until you reach the **Confirmation** panel and then click **Start**.

10. To deploy the changes, run the **bmide_modeltool** utility as follows:

    bmide_modeltool -u=*admin_user_name* **-p|-pf=***admin_user_password***-g=dba -tool=all -mode=install -target_dir=***TC_DATA*

11. Update and merge the schema file.

12. Restart all the required services to launch Active Workspace.

13. Verify if the indexed properties appear as filters in the **Filter** panel of global search. Additionally, open an indexed structure, and verify if the properties are listed in the **Filter** panel of-the context search. If the properties are not listed, follow the **troubleshooting steps** to fix the issue.

At this point, you can choose to **set the occurrence properties for indexing**, if not already set, after which you can close BMIDE.

8. Initialize the global search index

The global search index in Solr is used for keyword and attribute search of elements (item revisions) within a structure. It contains the properties of item revisions that are required to find and filter structures. The global search index is also required to set up the creation of the Smart Discovery index.

Initialize the global search index only if it is not already deployed at your site. To set it up, see *Data Indexing and Search Configuration*.

> Note:
>
> Smart Discovery only supports SolrCloud configuration with a single shard and one or more replicas.

If you set up any new item revision properties for indexing after initializing the global search index, you must set up the global search index again and run the indexer.

9. Set up spatial search by generating bounding boxes and TruShape files

About bounding boxes and TruShape files

Bounding boxes and TruShape (TSO) files are required to perform spatial search within structures.

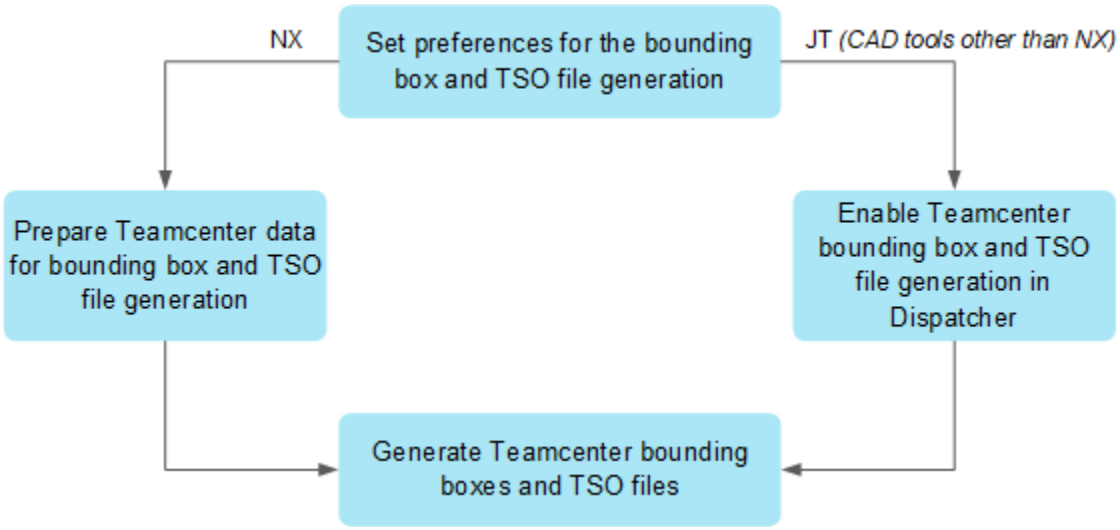
A bounding box is a box around the geometry of a part (item revision). When a user performs a proximity search by selecting the part, Teamcenter limits the search result by identifying bounding boxes of other parts that are within or ones that intersect the bounding box of the selected part.

To make the search result more accurate (relative to the geometric shape of the part), TSO files are required. A TSO file contains the voxel representation of each part that has geometry. With TSO files, Teamcenter simplifies the geometry of the intersecting parts into small cubes that provide a simplified representation of the actual shapes.

If cacheless search is deployed at your site, the process of generating bounding boxes and TSO files is already set up. If cacheless search is not deployed, you must **set up the process of generating bounding boxes and TSO files**.

Process flow to generate bounding boxes and TruShape files

Follow this process to set up bounding boxes and TruShape files if cacheless search is not deployed at your site.



Set preferences to generate bounding boxes and TruShape files

Process flow to generate Teamcenter bounding boxes and TruShape files for NX datasets				
1. Set preferences to generate bounding boxes and TruShape files	2. Upgrade legacy transforms to PLM XML transforms	3. Generate NX bounding boxes	4. Generate TSO files for NX datasets	5. Generate Teamcenter bounding boxes from NX bounding boxes

Process flow to generate Teamcenter bounding boxes and TruShape files for JT datasets		
1. Set preferences to generate bounding boxes and TruShape files	2. Enable the generation of bounding boxes and TSO files in Dispatcher for new data	3. Generate Teamcenter bounding boxes and TruShape files for the existing data

Before generating bounding boxes and TruShape (TSO) files for the structures in the Teamcenter database, you must set the value of the following preferences to **True**:

- For NX and JT:

  QS_SPATIAL_ENABLED

- For NX:

  - **QS_NX_RELATION_POST_ACTION_ENABLED**

    You must set the value of this preference to **True** because the **Tc Bounding Box** property is not exported through Multi-Site Collaboration, PLM XML, or TCXML. You need to create this property locally for the dataset object in the Teamcenter database. The value of this property is rolled up from the dataset to the item revision, and in turn, to the item business object. The Teamcenter database might not contain every item revision, so the **Tc Bounding Box** property must be calculated locally. A dispatcher process is required for the data import.

  - **QS_BBOX_GENERATION_FROM_NX_ENABLED**

    You must set the value of this preference to **True** so that when you save an item in NX, the UGBBOX Form is created in the Named References of the dataset. Additionally, this information is copied into the **Tc Bounding Box** property, which is used for spatial searching in Teamcenter.

- For JT:

- **QS_TRUSHAPE_GENERATION_ENABLED**
- **QS_BBOX_GENERATION_ENABLED**

Additionally, to specify the unit of measure for a proximity search in a Smart Discovery-indexed structure, set the **RDV_user_defined_units_of_measure** site preference to **METERS**, **MILLIMETERS**, **INCHES**, or **UNKNOWN**. If you specify **UNKNOWN**, the search is performed in the units of the **assy_units** property set on the top level BVR. If this property is not set, the search is performed in the units specified in the **PS_assume_legacy_transform_units** site preference.

For more information on accessing these preferences, see Where can I get a list of preferences? in *Teamcenter Administration*.

Generate Teamcenter bounding boxes and TruShape files for NX datasets

Process flow to generate Teamcenter bounding boxes and TruShape (TSO) files if the CAD application is NX				
1. Set preferences to generate bounding boxes and TSO files	2. Upgrade legacy transforms to PLM XML transforms	3. Generate NX bounding boxes	4. Generate TSO files for NX datasets	5. Generate Teamcenter bounding boxes from NX bounding boxes

The NX bounding box is stored in Teamcenter in a **UGPART-BBOX** file, and the TSO data is stored in the **Trueshape-Data** file. Both the files are attached to the **UGMASTER** dataset.

Upgrade legacy transforms to PLM XML transforms

1. Find all legacy transforms by running the **qsearch_process_queue** utility:

```
qsearch_process_queue -u=<user_id> -p=<password> -g=<group>
-list_legacy_transforms -outfile=c:\temp\legacytransforms.txt
```

2. Upgrade the legacy transforms by running the **nxmgr_upgrade_transforms** utility:

```
nxmgr_upgrade_transforms -u=<user_id> -p=<password> -g=<group>
-bypass=yes -i=c:\temp\legacytransforms.txt -upgrade_release=yes
```

3. Ensure that all transforms are upgraded by running the **qsearch_process_queue** utility again with the **-list_legacy_transforms** option.

Generate NX bounding boxes

1. Find all NX datasets that do not have the **UGPARTBBOX** file by running the **create_or_update_bbox_and_tso** utility:

   ```
   create_or_update_bbox_and_tso -u=<user_id> -p=<password>
   -g=<group> -translation_mode=NXBBOXFORM -mode=query
   -output_dir=c:\temp\ugpart_bbox
   ```

   This utility generates a list of file that are saved in the location that you specified in `-output_dir`. Use these files as the input (`-i`) in the **run_tc_publishing_utility.bat** utility.

2. Generate the NX bounding boxes by first setting the **QSEARCH_update_enabled** preference to **False** and then running the **run_tc_publishing_utility.bat** utility, which is located in *UGII_BASE_DIR***NXMANAGER**. Here, *UGII_BASE_DIR* is the folder where NX is installed.

   ```
   run_tc_publishing_utility.bat -u=<user_id> -p=<password> -g=dba -i="!
   file!" -record_pa=yes -log="!file!_refile.txt"
   ```

Generate TSO files for NX datasets

1. In NX, click **File** > **Utilities** > **Custom Defaults**.

2. In the **Customer Defaults** dialog box, click **Assemblies** > **Site Standards**, and select **True Shape** tab.

3. Set the cell size and cell units, and click **Generate Component Shape Representations on Save**.

   TSO files are generated upon every save. The default cell size in NX is 0.0 mm. A cell size of 3 mm is recommended for a more accurate spatial search.

4. Generate the TSO files for the existing NX dataset by running the **run_tc_publishing_utility.bat** utility, which is located in *UGII_BASE_DIR***NXMANAGER**. Here, *UGII_BASE_DIR* is the folder where NX is installed.

   ```
   run_tc_publishing_utility.bat -u=<user_id> -p=<password> -g=dba -i="!
   file!" -publish_tr=yes -record_pa=yes -log="!file!_refile.txt"
   ```

Generate Teamcenter bounding boxes from NX bounding boxes

You can generate bounding boxes for all data in the Teamcenter database. Although this is the recommended method, it is time-consuming. If you do not wish to wait for all data to be indexed, you can proceed and generate bounding boxes and TSO files for a specific structure.

Scenario	Procedure
Generate bounding boxes and TSO files for all data	To generate bounding boxes, first query the Teamcenter database to find out how many bounding boxes must be processed for the entire database. Subsequently, create the bounding boxes by using the NX bounding box data. `create_or_update_bbox_and_tso -u=<user_id> -p=<password> -g=dba -translation_mode=NXBBOXTOBBOX -mode=query+process`

Generate bounding boxes and TSO files for a specific structure	To generate the bounding box, first query the Teamcenter database to find the structure indexed using Smart Discovery Indexing. Subsequently, create the bounding box.
	
	`create_or_update_bbox_and_tso -u=<user_id> -p=<password> -g=dba -translation_mode=NXBBOXTOBBOX -mode=query+process -product=<product_item_id>`
	
	If Multi-Field Key (MFK) is used to uniquely identify a structure, execute the **create_or_update_bbox_and_tso** using the following syntax:
	
	`create_or_update_bbox_and_tso -u=<user_id> -p=<password> -g=dba -translation_mode=NXBBOXTOBBOX -mode=query+process -product=%#MFK#%,=item_id=<product_item_id>,object_type=<item_or_its_sub_type>`

For detailed information about the **create_or_update_bbox_and_tso** utility and its arguments, run `create_or_update_bbox_and_tso -h` on the Teamcenter command prompt.

Generate Teamcenter bounding boxes and TruShape files for JT datasets

Process flow to generate Teamcenter bounding boxes and TruShape (TSO) files for JT datasets		
1. Set preferences to generated bounding boxes and TSO files	2. Enable the generation of bounding boxes and TSO files in Dispatcher for new data	3. Generate Teamcenter bounding boxes and TruShape files for the existing data

Enable the generation of bounding boxes and TSO files in Dispatcher for new data

If you use CAD applications other than NX, you must enable the generation of bounding boxes and TSO files in Dispatcher. If Dispatcher is not deployed in your Teamcenter set up, install it along with the **JtToBboxAndTso** translator. If Dispatcher is available, update it to include this translator.

The **JtToBboxAndTso** translator invokes the process to create the bounding boxes and TSO files for new data that is currently not available in the Teamcenter database. Teamcenter reads the bounding box information from this index and stores the TSO data in a **Trushape-Data** file attached to the **DirectModel** dataset.

Generate Teamcenter bounding boxes and TSO files for existing data

You can generate bounding boxes and TSO files for all existing data in the Teamcenter database. Although this is the recommended method, it is time-consuming. If you do not wish to wait for all data to be indexed, you can proceed and generate bounding boxes and TSO files for a specific structure.

Scenario	Procedure
Generate bounding boxes and TSO files for all data	To generate bounding boxes and TSO files, query the Teamcenter database first and then create the files. `create_or_update_bbox_and_tso -u=<user_id> -p=<password> -g=dba -translation_mode=JTTOBBOX+JTTOTSO -mode=query+process`

Generate bounding boxes and TSO files for a specific structure	To generate bounding boxes and TSO files for a structure indexed using Smart Discovery Indexing, query the Teamcenter database first and then create the files.
	
	`create_or_update_bbox_and_tso -u=<user_id> -p=<password> -g=dba -translation_mode=JTTOBBOX+JTTOTSO -mode=query+process -scope=PRODUCT -product=<product_item_ID> -output_dir=c:\temp\ugpart_bbox`
	
	If Multi-Field Key (MFK) is used to uniquely identify a structure, execute the **create_or_update_bbox_and_tso** using the following syntax:
	
	`create_or_update_bbox_and_tso -u=<user_id> -p=<password> -g=dba -translation_mode=JTTOBBOX+JTTOTSO -mode=query+process -scope=PRODUCT -product=%#MFK#%,=item_id=<product_item_id>,object_type=<item_or_its_sub_type> -output_dir=c:\temp\ugpart_bbox`

For detailed information about the **create_or_update_bbox_and_tso** utility and its arguments, run `create_or_update_bbox_and_tso -h` on the Teamcenter command prompt.

Generating bounding boxes and TSO files for the existing data using the **create_or_update_bbox_and_tso** utility is an one-time activity. You need not run this utility for structures created after Smart Discovery Indexing is set up.

10. Set up volume search by creating box zones

You create box zones for a product so that users can perform a volume search within it. Before creating box zones, you must define zone dimensions. Note down the dimensions as you will use these to create box zones. If a box zone is already created, you can modify it to change the box coordinates, if needed.

Create box zones

1. Log on to Teamcenter rich client.
2. In **Structure Manager**, select the topmost node of the product, and click **File** > **New** > **Item Element**.
3. In the **Item Element** dialog box, click **More** and select **RDV Box Zone Form Type**.
4. Select the **Open on Create** check box and click **OK**.
5. In the **RDV Box Zone** dialog box, click **Check-Out and Edit**.
6. Enter the base point coordinates and edge vectors. For this, you may refer to the zone dimensions that you noted down earlier. You must enter data in every box.
7. Click **Save and Check-In**.

Modify box zones

1. Log on to Teamcenter rich client.
2. In **Structure Manager**, open the topmost node of the product.
3. Select the **RDV Box Zone** form and right click **Check-In/Out** > **Check Out**.

   In the **Check-Out** dialog box, click **Yes**.
4. Right click the checked out form and click **View Properties**.
5. In the **Properties** dialog box, click **All**.

   If **All** is not displayed, in **General**, locate **bl_line_object**, and continue with the steps below.
6. In the **All** properties tab, locate **ItemRevision/GDE Object**, and click the Expand icon ▼, and select **Open**.

7. Edit the base point and edge vector values and click **Save and Check-In**.

11. Initialize and set up a recurring task to update the Smart Discovery index

Indexing structures makes the search faster and provides advanced structure filtering and configuration capabilities. For this, you must set up a recurring task to initialize and update the Smart Discovery index.

In this process:

- The Smart Discovery index in Solr is initialized with the necessary Teamcenter data.
- Items and their occurrences that are regularly updated and the new items that are frequently created in the Teamcenter database are incrementally updated in the Smart Discovery index in Solr and in Teamcenter.

If changes are made to the schema due to the addition of new occurrence properties, then you must first run the `discovery:index` task, and then run the `discovery:sync` task. If you run the `discovery:index` task separately from the `discovery:sync` task, you must first stop the `discovery:sync` task already running, and then run the `discovery:index` task followed by the `discovery:sync` task.

Before populating the Smart Discovery index in Solr, you can **choose which properties of an item revision or an occurrence** must be indexed.

Prerequisites

Given the complexity and time required for initializing the Smart Discovery index, ensure that your enterprise setup considers the following recommendations:

- Schedule the process of creating the Smart Discovery index on non-working days or outside of work hours. Initializing the Smart Discovery index for large structures may take several hours. To estimate the time required, initialize a medium-sized structure and scale the recorded time to match the structure sizes of your production setup.
- Increase the system and environment resources, server pool sizes, and timeout duration of the initializing process. Ensure that there are minimal to no connectivity issues in a wide area network environment as the initialization process is highly communicative and may get interrupted by network restriction or congestion.
- Determine the maximum number of Teamcenter connections to be open at a time for indexing by considering your business requirement and update the value of in the `Tc.maxConnections` parameter in the *%TC_ROOT%\TcFTSIndexer\conf\TcFtsIndexer.properties* file.
- Determine the number of parallel processes required to process the index initially and recurringly, and then update the values of the following tuning parameters in the *%TC_ROOT%\TcFTSIndexer\conf\TcFtsIndexer_discovery.properties* file:

- `Discovery.maxConnectionPercentageForInitialIndex`
- `Discovery.maxConnectionPercentage`

To monitor the progress of indexing, inspect the *TcFTSIndexer.discovery.log* in the *%TC_ROOT%\TcFTSIndexer\logs* folder.

Procedure

To set up the recurring task, use the **runTcFTSIndexer** utility.

```
TC_ROOT/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:sync
-interval=<time_interval_in_seconds>
```

Specify the time interval in *seconds* depending on your environment requirements and the number of structures to be synchronized. A time interval of 120 seconds is ideal but it is recommended not set a time interval below 60 seconds.

At this stage, you can validate your enterprise set up before starting the Smart Discovery index creation process. To validate, you perform a basic test to ensure that **TcFTSIndexer** and Smart Discovery Indexing are set up correctly:

```
TC_ROOT/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:test
```

12. Create the Smart Discovery index in Teamcenter

The Smart Discovery index in Teamcenter stores the structure hierarchy and spatial coordinates for each element in a structure.

Restrictions and limitations

For structures already existing in the Teamcenter database, plan and choose which structures you want to index based on your site requirements. The selected structures must be the topmost nodes. This is a one-time activity. The recurring task will keep the indexed structures up to date.

Every time a user creates a new structure, you must determine if it needs to be indexed and manually index it.

Prerequisites

Before setting up the creation of the Smart Discovery index, ensure that the **recurring task** to update the index is running. Also, ensure that you have the read access to structures (and their contents) that you want to index.

Procedure

1. Perform a basic test to ensure that **TcFTSIndexer** and Smart Discovery Indexing are set up correctly:

   ```
   TC_ROOT/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:test
   ```

2. Create the Smart Discovery index in one of the following ways:

   - **Create the Smart Discovery index automatically for a list of products**

     a. Identify the toplevel item type that represents the root of the products you want to index. And, create saved queries for these items (or for their subclasses). While creating the saved query, ensure that the toplevel item you identified is not an item revision. Additionally, while creating the saved query, ensure that no user input is required to return the items that you intend to index.

        If engineering BOM management is deployed in your Teamcenter setup, the **EBOM Designs** and **EBOM Products** saved queries are created by default for design structures and collaborative product engineering BOMs.

b. Add the saved queries as values of the **QSEARCH_products_to_index_saved_queries** preference. By default, the **EBOM Designs** and **EBOM Products** saved queries are added to this preference.

c. Run the **TcFTSIndexer** utility to index all the products that get listed when you run each of the search query that you added in the **QSEARCH_products_to_index_saved_queries** preference.

```
TC_ROOT/TcFTSIndexer/bin/runTcFTSIndexer
-task=discovery:create -product=#autoindex#
-interval=<time_interval_in_seconds>
```

The **TcFTSIndexer** service automatically runs the **TcFTSIndexer** utility at the time interval you specified. Any product that was not indexed in the earlier time interval is indexed when the utility is run again.

- **Create the Smart Discovery index manually for each structure**

Create the Smart Discovery index manually in one of the following ways, that is, by using either the **runTcFTSIndexer** utility or the **smart_discovery_index** utility:

- **Create the Smart Discovery index manually by using the runTcFTSIndexer utility**

Run the **runTcFTSIndexer** utility for every structure that you want to index. You run this utility on the machine on which Indexer is deployed.

```
TC_ROOT/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:create
-product=<product_item_id>
```

If Mutli-Field Key (MFK) is used to uniquely identify a structure, execute the **runTcFTSIndexer** using the following syntax:

```
TC_ROOT/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:create
-productKey="item_id=<product_item_id>,object_type=<item_or_its_s
ub_type>"
```

The time taken to create the Smart Discovery index is dependent on the size of the structure. Larger structures may take longer time to get indexed. To avoid the time-out during the index creation process, it is recommended that you use the server manager (pool manager) dedicated to **runTcFTSIndexer** for executing the indexing process. It is also recommended that you set **QUERY_TIMEOUT** to **0** in the *serverPool.properties* file so that the server connection does not time out automatically and allows the indexing process to complete.

To check the status of the indexing process, run the following command:

```
TC_ROOT/TcFTSIndexer/bin/runTcFTSIndexer -status
```

- **Create the Smart Discovery index manually by using the smart_discovery_index utility**

  Run the **smart_discovery_index** utility for every structure that you want to index. You run this utility on the machine where Teamcenter is deployed.

  ```
  smart_discovery_index -u=<user_id> -p=<password> -g=<group>
  -create -product=<item_id>
  ```

  For structures with Mutli-Field Key (MFK):

  ```
  smart_discovery_index -u=<user_id> -p=<password> -g=<group>
  -create
  -product=%#MFK#%,=item_id=<product_item_id>,object_type=<item_or_
  its_sub_type>
  ```

  The structure is now submitted for processing the index creation. If indexed successfully, the structure is moved to the list of products for which indexes are *Available*. To verify if the structure is listed in the *Available* list, you must inspect the status of the indexing process by running the following command:

  ```
  smart_discovery_index -u=<user_id> -p=<password> -g=<group>
  -inspect
  ```

Postrequisites

After you index structures, generate a report to **inspect the status** of the indexer. In case of any issue, see *Troubleshoot issues related to administering Smart Discovery Indexing*.

13. Inspect the status of the indexer or a structure

To validate if index creation process is complete and if structures are indexed successfully by using Smart Discovery Indexing, you can generate a report and inspect the status of the indexer. To inspect the status, you can run either the **TcFTSIndexer** utility or the **smart_discovery_index** utility.

Inspect the status of the indexer or a structure by using the TcFTSIndexer utility

Run the **TcFTSIndexer** utility on the machine where Indexer is deployed:

```
TC_ROOT/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:admin
-inspect_indexer -depth=<report_level>
```

Here, the *report_level* specifies the level to which the inspection routine can go into. The value of *report_level* can be **0**, **1**, or **2**. It is recommended that you set the value as **1**. If you set the value as **2**, the report may take longer to execute.

The inspection report is available in the *TC_ROOT/TcFTSIndexer/logs/report/directory* folder. The name of the report is in the format, `inspect_indexer_<yyyy_MM_dd_HH_mm_ss>.txt`.

To generate a report to find object that did not get indexed or updated in Solr, run the **TcFTSIndexer** utility:

```
TC_ROOT/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:admin
-report_failures
```

This report is also available in the *TC_ROOT/TcFTSIndexer/logs/report/directory* folder. And, the name of the report is in the format, `occpropreport_<yyyy_MM_dd_HH_mm_ss>.txt`.

Inspect the status of the indexer or a structure by using the smart_discovery_index utility

Run the **smart_discovery_index** utility Teamcenter is deployed:

```
smart_discovery_index -u=<user_id> -p=<password> -g=<group> -inspect
```

The details of the report are displayed in the command prompt. If any issues are listed, you may **troubleshoot** to fix them.

You can update the following preferences to specify various duration and time limits:

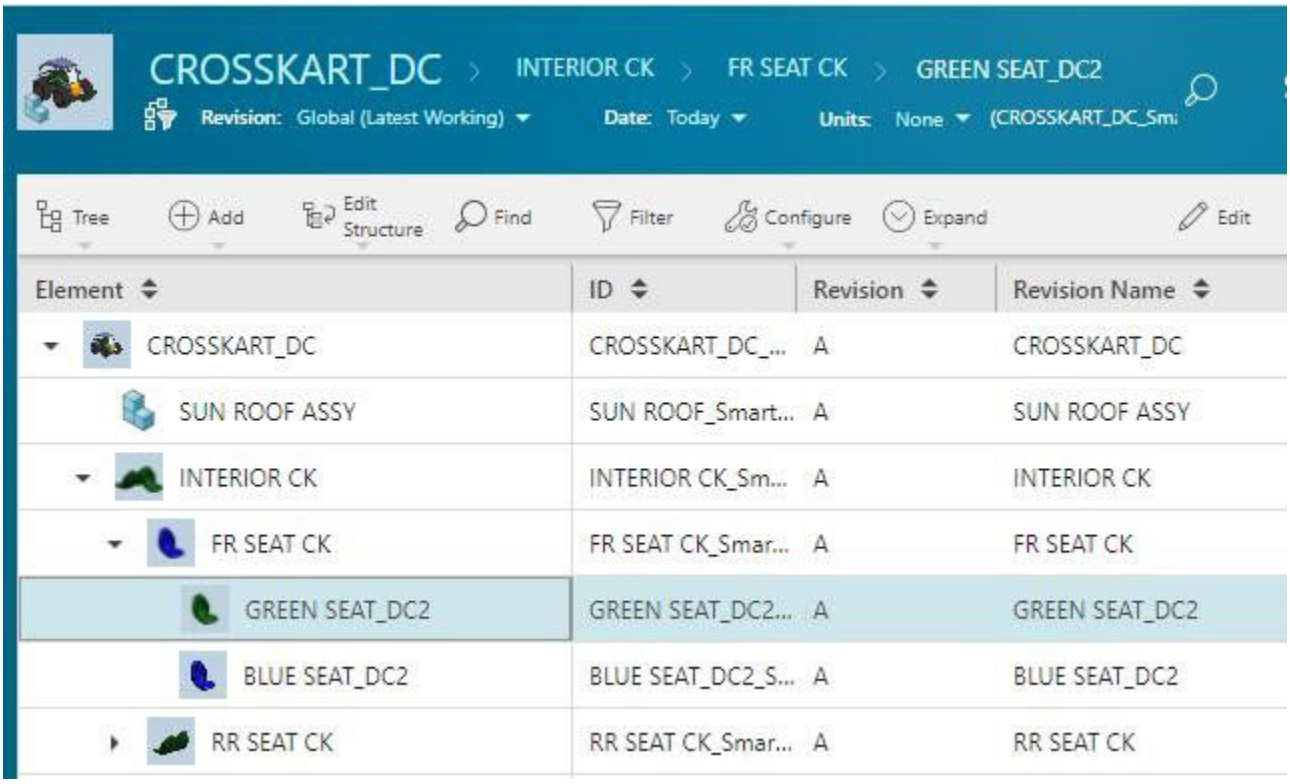
Preference	Description
QSEARCH_ADMIN_inspect_updater_last_processed_date_age_limit	Specify the time limit to determine when the updater is deemed stale with respect to its last processed date.
QSEARCH_ADMIN_inspect_stale_product_age	Specify the time limit to determine when a structure is deemed stale with respect to its last updated date.
QSEARCH_ADMIN_inspect_stale_product_age_delta_updater	Specify the time limit to determine when a structure is deemed stale with respect to its last updated date relative to the last processed date of the index updater.

You can also inspect the status of a specific structure by using a hierarchical path consisting of item IDs or unique IDs.

```
smart_discovery_index -u=<user_id> -p=<password> -g=<group> -inspect
-path="root_ID,child01,child02,target_id"
```

Example:

To inspect the status of **GREEN_SEAT_DC2** in **CROSSKART_DC**:



```
smart_discovery_index -u=<user_id> -p=<password> -g=<group>
-inspect -path="CROSSKART_DC,INTERIOR CK,FR SEAT CK,GREEN_SEAT_DC2"
```

14. Remove deleted occurrences from the Smart Discovery index

When a user removes an occurrence from a structure, the occurrence continues to appear as a facet on the **Filter** panel, by default. This is because the value of the **QSEARCH_index_deletion_mode** site preference is set as **disabled** due to which the occurrence continues to be available in the Smart Discovery index. If you do not want the deleted occurrences to appear as facets, you must change the value of this preference to **delete**. On doing so, any occurrence that the user deletes in the future is removed from the Smart Discovery index when **the index is updated**. Occurrences that were deleted from the structure before you changed the value of the preference to **delete** are not removed from the Smart Discovery index.

15. Delete the Smart Discovery index of a product

Warning:

Deleting the Smart Discovery index of a product for performance improvements or for fixing issues must be performed only if absolutely needed as this disables the Smart Discovery Indexing process for the product, and the product may have to be reindexed. Reindexing may take several days to complete depending on the size of the product.

To delete the Smart Discovery index, run:

```
smart_discovery_index -u=<user_id> -p=<password> -delete
-product=<product_id>
```

Deleting the Smart Discovery index of a product does not modify its structure and the product can be reindexed.

Tip:

Deleting the Smart Discovery index of a product can improve the search and filtering performance of the remaining products. Purge legacy data of a product that is frozen, released, or retired.

16. Verify if Smart Discovery Indexing is set up correctly

After performing all **tasks of Smart Discovery Indexing set up process**, perform the following tasks to verify in the indexing is set up correctly.

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
Enable Smart Discovery Indexing	*TC_ROOT*/bin/smart_discovery_index -u=<user_id> -p=<password> -g=<group> -enable This is a one-time activity.
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 recurring task initially populates the Smart Discovery index in Solr and then regularly updates it in both Solr and Teamcenter.
Perform a basic test to ensure that the global search index is set up correctly	*TC_ROOT*/TcFTSIndexer/bin/runTcFTSIndexer -task=objdata:test
Perform a basic test to ensure that TcFTSIndexer and Smart Discovery Indexing are set up correctly	*TC_ROOT*/TcFTSIndexer/bin/runTcFTSIndexer -task=discovery:test
Create Smart Discovery index	*TC_ROOT*/bin/smart_discovery_index -u=<user_id> -p=<password> -g=<group> -create -product=<product_item_id> For structures with Mutli-Field Key (MFK): smart_discovery_index -u=<user_id> -p=<password> -g=<group> -create

Task	Action
	`-product=%#MFK#%,=item_id=<product_item_id>,object_type=<item_or_its_sub_type>`
Inspect the Smart Discovery index creation status	*TC_ROOT*`/bin/smart_discovery_index -u=<user_id> -p=<password> -g=<group> -inspect`
View how many structures are indexed	*TC_ROOT*`/bin/smart_discovery_index -u=<user_id> -p=<password> -g=<group> -list` The indexed structures (items) are included in the **Available** list.
View the Smart Discovery indexed structure on Active Workspace	Open the top node of the structure and verify if **Filter** is displayed.
Perform a spatial search on Active Workspace	Ensure that you have **generated bounding boxes and TruShape files** for structures in order to perform a spatial search.
Perform a volume search in on Active Workspace	Ensure that you have **created RDV box zone forms** for products in order to perform a volume search.

17. Troubleshoot Smart Discovery Indexing issues

Troubleshoot issues related to administering Smart Discovery Indexing

First, verify if Smart Discovery Indexing is set up correctly. If yes, proceed to troubleshooting:

- Issues found while creating the Smart Discovery index.
- Issues found while populating Solr with the Smart Discovery index.
- Issues found while updating the Smart Discovery index at regular time intervals.

Issues found while creating the Smart Discovery index

Issue	Fix
The Teamcenter relational database is unavailable.	• Inspect the Smart Discovery index to see if there are any issues reported for the structure or for a target item. • Consider deleting the Smart Discovery index only if needed, and index the structure again.
The **smart_discovery_index** utility is used incorrectly or an issue occurs while executing this utility.	• Run `smart_discovery_index -h` to see its help on the correct argument usage. • Check the Teamcenter command prompt to see if there are any errors reported.
The Smart Discovery index creation process is not complete.	• As large structures can take several days to index, periodically inspect if the structure is still in the indexing process. Wait for the indexing to be complete. • Check for suitable system resources. • Consider deleting the Smart Discovery index only if needed, and index the structure again.

Additionally, inspect the tcserver log files for issues. Also, restrict the active users to administrators till the initial Smart Discovery index creation is complete.

Issues found while populating Solr with the Smart Discovery index

Issue	Fix
Solr is unavailable.	• Restart the indexer (**FTSIndexer**) and Solr processes. • **Perform the Solr indexing again**.
The query times out.	• Increase system resources. • Increase the timeout period in Solr and of **FTSIndexer**. • Restart the **FTSIndexer** and Solr processes. • **Perform the Solr indexing again**.

Additionally, inspect the **FTSIndexer** log files for issues.

Issues found while updating the Smart Discovery index at regular time intervals (FTSIndexer synchronization)

Issue	Fix
The Smart Discovery index in Solr is unavailable.	• Ensure that the **synchronization task is set up** and is executed at the specified time interval. The structures in which issues were found while updating the Smart Discovery index are considered again for synchronization.
The product path in the Smart Discovery index is not updated OR There is a long work queue.	• Ensure that the **synchronization task is set up** and is executed at the specified time interval. The structures in which issues were found while updating the Smart Discovery index are considered again for synchronization. The Smart Discovery index in Solr is automatically updated when the product path is updated in the Teamcenter relational database. • Wait for the synchronization task to be complete. Large structures and a longer queue take time to be completed. Check the last-indexed timestamp of a structure. If the structure was indexed after the last synchronization happened, the structure may not be synchronized yet.

Issue	Fix
	• Check if the session settings are correct. • Increase the number of **tcserver** instances in the server pool. • Inspect the **tcserver** log files for issues.
The synchronization task is not running.	• **Inspect** the Smart Discovery index to see the last available update.
The synchronization task is stuck.	• Wait for large structures to get indexed completely. • **Delete unused structure indexes**. • Increase the number of **tcserver** instances in the server pool. • Increase the system resources. • Inspect the **TcFTSIndexer** and **tcserver** log files for issues.
The changes are too large and the synchronization task times out.	• Increase the timeout period. • Inspect the **TcFTSIndexer** and **tcserver** log files for issues.
The Solr schema is not updated.	• Stop the **TcFTSIndexer** synchronization task, **reindex Solr**, and restart the synchronization task. • Verify if the required **item revision** and **occurrence** properties are set up for indexing. Also ensure that after setting up the properties for indexing, you update and merge the schema file.

Troubleshoot issues related to Smart Discovery indexed structures

First, **verify if Smart Discovery Indexing is set up correctly**. If yes, proceed to troubleshooting:

- **The indexed properties of an item revision are not displayed as filters in the Filter panel**.
- **The proximity and volume filter options are not available in the Filter panel**.
- **The filter icon is not enabled**.

- **The changes made to an indexed structure are not updated in the Smart Discovery index**.
- **The indexed occurrence properties are not displayed as filters in the Filter panel**.
- **Unable to get the expected search results**.

> Note:
>
> Filtering with Smart Discovery indexed structures does not support multiple view types. If view types are applied, these are ignored for filtering.

The indexed properties of an item revision are not displayed as filters in the Filter panel

Reason	Fix
The item revision properties are not set up for global search indexing.	Set up the item revisions to appear as filters in the global search.
The item revision properties are not set up for Smart Discovery Indexing. OR The item revision properties are not listed in the **AWS_Discovery_FullTextSearch_FacetFilters** preference.	Set up the item revisions to appear as **filters in the in-context search** of an indexed structure.

The proximity and volume filter options are not available in the Filter panel

Reason	Fix
The structure was indexed using Active Content Structure Indexing.	**Index the structures** again using Smart Discovery Indexing. Users must then select a revision rule from the list below the line separator.
There were problems during indexing.	Follow the troubleshooting steps to fix issues found while **creating the Smart Discovery index** and **populating Solr with the Smart Discovery index**.
The spatial data is not generated.	Generate and verify spatial data for **NX** and **JT** datasets. Wait for the next synchronization to be complete. Lastly, **delete the Smart Discovery index** and index the structure again.

Reason	Fix
The Smart Discovery index in Solr is not updated.	Follow the troubleshooting steps to **fix issues found while updating the Smart Discovery index at regular time intervals**.
The *session*, which stores the filtered and configured structures, is not configured correctly.	Check if the session settings are correct.

Additionally, perform the following validations:

- Check the **tcserver** log files for issues during the creation or import of spatial data.
- Validate the spatial data on a different instance of Active Workspace.
- Increase system resources and **tcserver** pool resources, if needed.

The filter icon is not enabled

Reason	Fix
Correct license is not available.	Verify if the Context Management User license is available in your Teamcenter setup by running the following utility: *TC_ROOT*`\TcFTSIndexer\bin\runTcFTSIndexer.bat -task=discovery:task`
Solr is not set up correctly.	• Check the *solr.log* file at the location where Solr is setup, for example, **TCROOT/Solr**. • Fix Solr issues.
The Smart Discovery index for the product is not created in Solr and Teamcenter relational database.	Follow the troubleshooting steps to **fix issues related to Smart Discovery Indexing**.

The changes made to an indexed structure are not updated in the Smart Discovery index

Reason	Fix
The **FTSIndexer** synchronization task is not running. OR	Follow the troubleshooting steps to **fix issues found while updating the Smart Discovery index at regular time intervals**.

Reason	Fix
The **FTSIndexer** synchronization task is running but the changes are not yet updated.	
A problem during the initial indexing is causing update issues.	Follow the troubleshooting steps to fix issues found while **creating the Smart Discovery index**.
The session is configured incorrectly.	Check if the session settings are correct.

Additionally, increase the system resources and **tcserver** pool resources, if needed.

The indexed occurrence properties are not displayed as filters in the Filter panel

Reason	Fix
The Smart Discovery index in Solr is not updated. OR The Solr schema is not updated.	Follow the troubleshooting steps to **fix issues found while updating the Smart Discovery index at a regular time intervals**.
An error occurred while updating the occurrence property in BMIDE. OR The occurrence property type is not supported.	Check the associated log files and **set up occurrence properties for indexing** again.
The newly added occurrence property does not have any value populated in the target structure.	Create a sample persistent occurrence property and add a value to it. Wait for the synchronization to be complete. In case of any errors, follow the troubleshooting steps to **fix synchronization issues**.
The session is configured incorrectly.	Check if the session settings are correct.
The occurrence properties are not listed in the **AWS_Discovery_FullTextSearch_FacetFilters** preference.	Set up the properties to appear as **filters** in the in-context search for an indexed structure.

Additionally, increase the system resources and **tcserver** pool resources, if needed.

Unable to get the expected search results

There can be several reasons for not getting the expected search results or not seeing the search results that you see in other applications, such as NX and Lifecycle Visualization.

Reason	Fix
The Smart Discovery index in Solr is not updated.	Ensure that your setup meets the recommendations required to create the Smart Discovery index. Follow the troubleshooting steps to fix issues found while updating the Smart Discovery index at a regular time intervals. Check if the session settings are correct. Check the **TcFTSIndexer** log files.
The session is configured incorrectly.	Check if the session settings are correct.

Additionally, increase the system resources and tcserver pool resources, if needed.

View the tcserver log file to troubleshoot Smart Discovery Indexing issues

You can view the **tcserver** log file to troubleshoot several issues related to Smart Discovery Indexing.

Structures in which issues were found while updating the Smart Discovery index are considered again for synchronization, and the issues are automatically resolved. However, certain objects in the structure in which the issues cannot be fixed automatically are quarantined. For example, if the background update of an index has failed and cannot be resolved automatically, it is placed in quarantine. Information related to the failure is recorded in the **tcserver** log file under the quarantine section.

To get a count of the quarantined objects, run:

```
smart_discovery_index -inspect
```

To locate the **tcserver** log file to view the details related to the issues, run:

```
smart_discovery_index -report_update_failures
```

In the **tcserver** log file (for example, *syslog.example.syslog*), search for **quarantine** to locate the quarantined objects and to view the details related to the issues found.