Teamcenter 13.3

Search and Query Builder

Teamcenter 13.3
PLM00031 - 13.3

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a leading global provider of product life cycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens Digital Industries Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens Digital Industries Software products and services, visit **www.siemens.com/plm**.

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

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

Contents

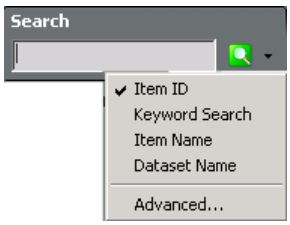
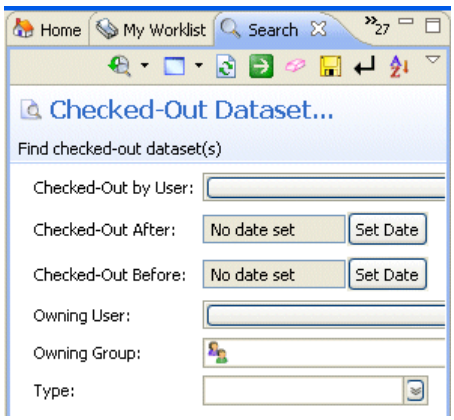
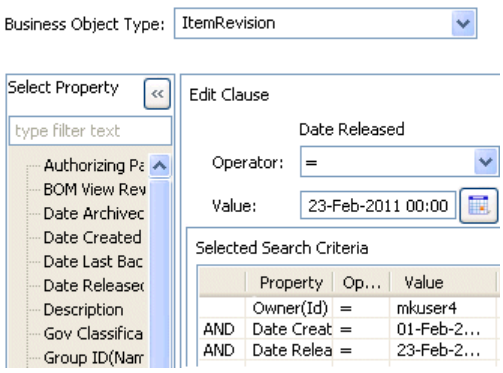
Geolus integration with Teamcenter

1. Configuring Teamcenter searches

Teamcenter search mechanisms

Teamcenter provides three search mechanisms enabled by default: *Quick*, *Advanced*, and *Simple*. You can modify the behavior of each mechanism.

Configuration is required to make full use of *Advanced Search*.

Quick Search	Advanced Search	Simple Search (rich client only)
Displays a single-field search form, providing a quick input method for users to search the Teamcenter database. 	Displays preconfigured search forms allowing users to specify multiple search criteria relevant to the type of information or object being sought. 	Displays the relevant properties for the selected business object, allowing users to create searches without an in-depth knowledge of the Teamcenter POM schema. 
Generates searches based on a single criterion (such as item ID, item name, or dataset name) selected from a shortcut menu. Administrators can set preferences to determine which criteria display in the menu.	Generates searches based on preconfigured search forms. The search forms are derived from search queries. Hundreds of search queries are shipped with Teamcenter. You can create additional search queries (also called saved searches) using Query Builder.	Generates searches based on one or more property values of selected business objects. Users select an object type, and then build a search form by selecting properties and specifying criteria. Users do not need to understand the placement of attributes within the POM schema.
Provides quick searching of items and datasets. It requires users to know the item ID, item name, or the dataset name.	Distributes in-depth, customized search forms throughout the site and/or throughout a global enterprise.	Provides a tool for users to create their own customized searches to search the local Teamcenter database, without a strong knowledge of the POM schema.

Quick Search	Advanced Search	Simple Search (rich client only)
Quick searches are limited to a single user. They cannot be stored by the user, nor shared with other users.		These searches are limited to a single user. They cannot be stored by the user, nor shared with other users.
Quick search is available by default in the rich client. No configuration is required to use this search with its default search queries. However, you can add custom queries.	Advanced search is available by default in the rich client. No configuration is required for this search functionality to appear. However, making full use of this search method requires creating search queries in Query Builder.	Simple search is available in the rich client only. It appears by default. No configuration is required for this search functionality, although you can set preferences to modify its behavior.
These techniques are used to **implement and configure Quick Search**. You configure which objects appear in the selection list.	This technique **implements and configures Advanced Search**. • **Create custom queries**	These techniques are used to **implement and configure Simple Search**. • Modify default display settings. • Filter the business object types available for search.

Implementing quick search

What is Quick search?

Quick Search is available by default in the rich client. No configuration is required to use this search with its default search queries.

You can add custom search items to the **Quick Search** menu by:

1. Building the query.

2. Updating the two **Quick Search preferences** with the name and attribute of the custom query.

Quick search preferences

You must update the following preferences when adding custom queries to the **Quick Search** menu:

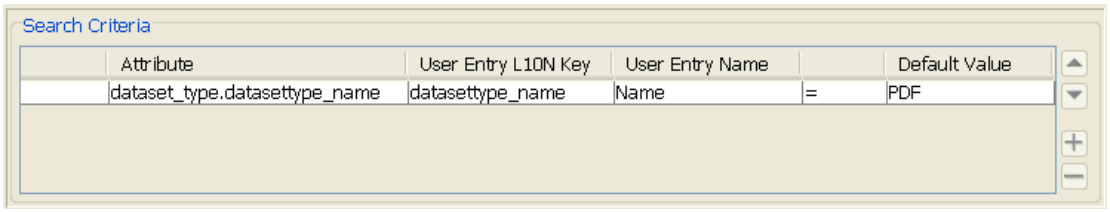
- **Quick_Access_Queries**
  Specifies which queries appear in the **Quick search** menu. Valid values are query names defined in the **qry_text_locale.xml** locale file.
  By default, the following queries are defined:

- **General**
- **Item**
- **Item Revision**
- **Remote**
- **Checked-Out Dataset**

- **Quick_Access_Queries_Attribute**
  Specifies the criteria attribute displayed for a query in the **Quick Search** menu using the following format:

  *internal-query-name*_**SearchAttribute=***L10N-key*

  The internal query name is specified in the **qry_text_locale.xml** locale file. The L10N key is the **User Entry L10N Key** value specified for the given query.
  In the following example, a custom query named **Find PDFs** is created. The value of the L10N key is set to **datasettype_name**.



- **Default_Quick_Access_Query**
  Specifies the default quick search name, for example, **Item ID**.

Implementing simple search

What is simple search?

Simple Search is available by default in the rich client. This powerful search tool does not require an in-depth knowledge of the Teamcenter POM schema. Users can create business object searches based on one or more property values, with only the following limitations:

- Only **WorkspaceObjects** can be specified for this type of search.
  To allows users to search for other POM objects, you must build queries containing these objects using Query Builder. Such queries are accessed from **Advanced Search**.

- Only attribute properties and typed referenced properties can be specified for this type of search. **Simple Search** and Query Builder do not support run-time properties and compound properties.

- Scope is limited to the users local Teamcenter database.
  To allow users to search remote Teamcenter databases, use the **Remote** saved query, or build a custom query using Query Builder. Such queries are accessed from **Advanced Search**.

- Ad hoc and classification searches are not supported.

- Search results display in the **Search Results** view. Users can refresh, compare, save, and assign these search results. However, the search criteria cannot be saved or shared.

No configuration is required before users can begin using this search mechanism. However you can modify its behavior by setting Simple search preferences.

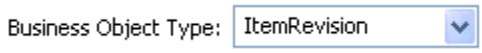
Simple search preferences

You can use the following preferences to modify the behavior of **Simple Search**:

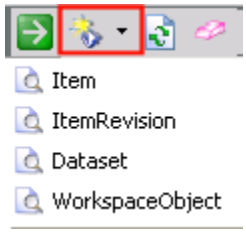
- **Default_Business_Object_To_Search**
  Specifies which business object displays when the **Simple Search** view is opened.

- **Favorite_Business_Objects_To_Search**
  Specifies which business object types appear in the:

  Business Object Type list

  Business Object Type button

- **Searchable_Business_Objects**
  Filters the business object types that display in the **Business Object Type** list, limiting the list to the business objects specified by this preference. Limit the list to only those business objects used at your site, allowing users to locate business object types quickly.
  This preference can be set as a site preference by an administrator and as a group or role preference by a group administrator.

Implementing advanced search

What is advanced search?

Advanced Search is available by default in the rich client. No configuration is required to use its default search queries.

- Display hundreds of saved queries available by default. Users access the saved queries as search forms in which they can type search criteria.
- Create additional **custom saved queries**.
- Distribute saved queries throughout your local site or throughout a global enterprise.

Configuring Advanced Search preferences

You can use the following preferences to modify the behavior of **Advanced Search**:

- **Change_Search_Default**
  Determines which saved query displays in the search panel by default.
  This preference accepts a single string as a value. The value must be a valid saved query.
- **QRY_dataset_display_option**
  Determines whether the latest version or all versions of a dataset object are displayed when query results are returned.
  Set this preference to **1** to display all versions of a dataset object. Set to **2** to display only the latest version of the dataset object.
- **QRY_***query-name***_SortKeys**
  Determines the class attributes used to sort the query results. This preference must be used in conjunction with the **QRY_***query-name***_SortOrder** preference, which determines sort order.
  The value type of the class attribute must be a primitive value type. For example, char, int, double, or date. If a query name contains one of the following characters, this character must be replaced with the underscore (_) character:
  - Space character
  - Return character (**\n**)
  - Tab character (**\t**)
- **QRY_***query-name***_SortOrder**
  Determines the sort order of query results. This preference must be used in conjunction with the **QRY_***query-name***_SortKeys** preference, which determines the class attributes used to sort query results.
  Set this preference to **1** to sort query results in ascending order. Set to **2** to sort query results in descending order.
- **QRY_search_type_hierarchy**
  Determines whether subtypes are included in query results. The system uses type hierarchy functionality to query types and subtypes.
  Administrators can add this as a site preference in the **Options** dialog box, which can be accessed from the **Edit** menu, or in the **tc_preferences.xml** preferences file. Users can set this, as a user preference, to **true** from the search interface:

In the rich client, choose **Edit→Options** to display the **Options** dialog box. Select the **Search** folder from the tree on the left. Select the **Query Options** tab. Then select the **Enable Hierarchical Type Search** check box. Checking this check box automatically adds this preference to the database, and sets it to **true**.

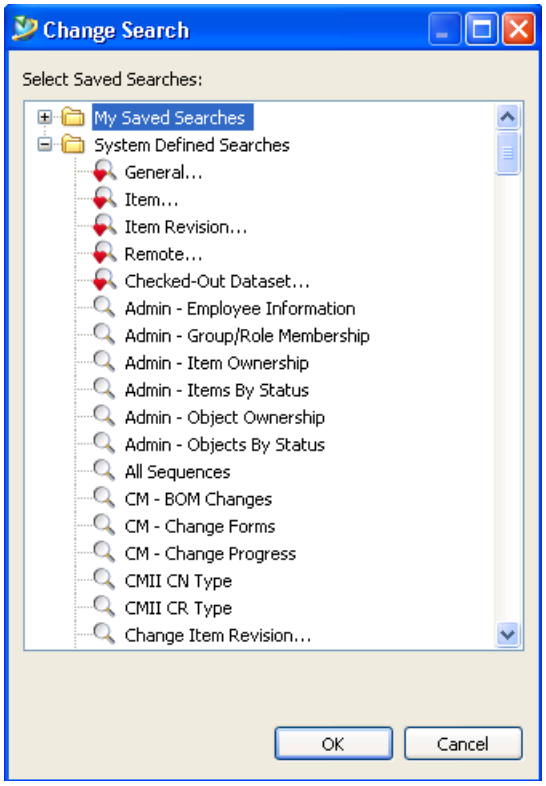
Creating queries for Advanced Search

Query Builder enables you to create complex queries based on the Teamcenter data schema, a hierarchical arrangement of types, subtypes, and properties. Query Builder provides hints to assist you in navigating the schema. The hints present a relationship as a starting point, for example, the relationship between an item and its item revision, and then provide you with the steps to build that relationship into your search definition.

Queries created in Query Builder display in the **Saved Queries** tree. Saved queries can be used for searches and to generate reports.

In the context of **Advanced Search**, these queries also appear in the **System Defined Searches** tree, accessed by clicking the **Select a Search** button on the **Advanced Search** view to display the **Change Search** dialog box.

The following graphic illustrates a few of the saved searches available by default.



2. Learning about Query Builder

What is Query Builder?

Query Builder is a feature that allows administrators to create predefined queries for users searching Teamcenter databases. Query Builder offers users a list of **Saved Queries** in My Teamcenter.

A set of **Saved Queries** are defined by default, and you can define additional queries. Defining a query requires knowledge of the Teamcenter POM (persistent object manager) schema, which is a hierarchical arrangement of types, subtypes, and properties.

> Tip:
>
> When creating objects for your data model in Business Modeler IDE, be sure that you create related queries that allow users to find them.
>
> Because the queries list is not searchable, be sure to use descriptive query names that users can recognize by purpose or use. Consider grouping similar queries using common prefixes.

Query Builder is selected during Teamcenter installation, which makes it available with no further configuration. Use of **Saved Queries** is subject to user security access rules. Query Builder does not support run-time or compound properties.

Features that can enhance query creation

- **Reuse an existing definition**.
- **import or export query definitions** to share with other Teamcenter sites. The XML contents are parsed and verified before the data is imported.
- **Use query hints** from the **queryHint.xml** file.
- **Use type properties in a query**
- **Use subtype queries on a type reference**
- **Use referenced-by properties in a query**
- **Use IS_NULL or IS_NOT_NULL operators**

Using Query Builder

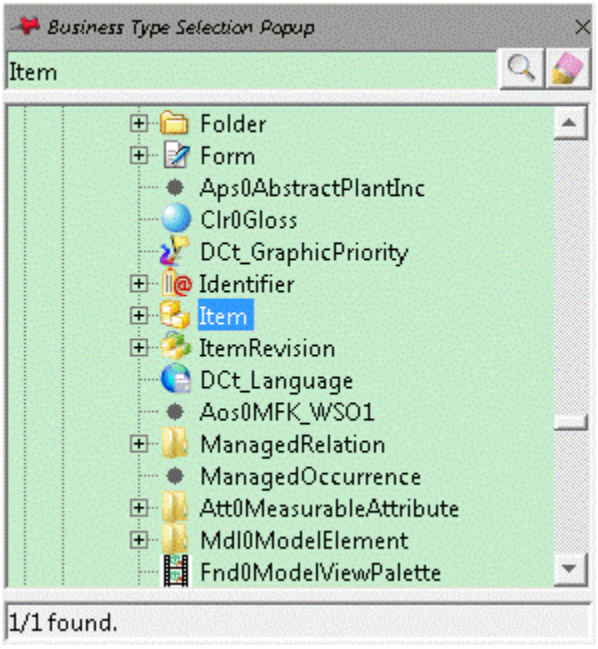
To begin using **Query Builder**, click  in the navigation pane. **Saved Queries** display a list of predefined queries.

Saved Queries tree	Displays all saved queries in the database. When you select a saved query in the tree, the details are displayed in the panes on the right side. For example, a saved query might find all items that have been shipped.
Saved query properties	Displays the name, description, query type, and search type of the selected **Saved Queries**. You can modify these properties and then create or modify a saved query. You can also delete a saved query.
Localization 	**Language Translations** lists existing translation values. • View and modify the existing localization text. • Add or remove a translation value for a locale without altering the master value of the property.
Search Type	Displays the **Business Type Selection** dialog box that lists the **Query** types.
Property Selection pane	Displays the attributes of the selected type and either all inherited types or only the direct attributes of the type, depending on the display setting you select.
Search Criteria pane	Defines the search criteria clauses using attributes, user entry keys, operators, and default values. Boolean operators process multiple search criteria clauses.

Selecting a search type

Click **Search Type** to display the **Business Type Selection** dialog box. You can navigate and search the Teamcenter POM schema and select types of attributes to use in query definitions.

Type a name (and an optional wildcard) in the box at the top and click **Search**  . The number of types matching your search are displayed at the bottom, and the first result is highlighted in the tree.



Selecting properties

To create query definitions, **select the type of property** to use in the search criteria.

Finding the type's properties

Related properties are displayed in **Property Selection pane**. Displayed properties show all inherited types or only the direct attributes of the type.

Search type	All properties are displayed.
Parent type	When all properties are displayed, inherited properties are included.
Reference type	When related properties are displayed, reference properties are included.
Referenced-by type	When properties related to the **Search Type** are not displayed, **Referenced By** can locate additional referenced types.

Display Settings

Click **Display Settings** to choose:

- Properties defined on the type or all types (including inherited properties).
- Property display names or the property database names.

Add type properties to search criteria

You can add a property to the **Search Criteria** table.

- Properties without the plus symbol can be directly set when searching for objects.
- Properties with the plus symbol ⊞ refer to another type in the type structure, which may have direct properties or more properties with plus symbols.

1. Select a type as the search type.
2. Set **Display Settings** to **All Properties** and **Real Names**.
3. Populate the **Property Selection** list with the properties required to build the search criteria. Properties are either type properties, parent type properties, or reference type properties.

   Double-click attributes without the plus symbol ⊞ to add them directly to the **Search Criteria** table.
4. You can add reference types, which are types related to the search type. Navigating reference type requires knowledge of the Teamcenter data model.

   To list reference type properties:

   - Double-click references with the plus symbol ⊞ to list more attributes or to open **Property Selection**. Double-click a type from the **Property Selection** to add the type and its properties to the **Property Selection** list.
   - Double-click **Referenced By** at the bottom of the **Property Selection** list to open **Property Selection**. Double-click the **Search Type** box to search for a type. Select a reference and click **OK** to add the reference and its attributes to the **Property Selection** list.

Property Selection

Property Selection displays the properties of the selected type and either all inherited type properties or only the direct properties of the type.

Symbol	Property type
C	Character
[C]	Character array
	Date
	Date array

Symbol	Property type
d	Double
[d]	Double array
f	Float
[f]	Float array
i	Integer
[i]	Integer array
b	Logical
[b]	Logical array
sh	Short
{sh}	Short array
S	String **LongString** attributes cannot be used in queries.
[s]	String array
t→	Typed reference
{t→}	Typed reference array
→	Untyped reference
[→]	Untyped reference array
e→	External reference
{e→}	External reference array
	Note
{}	Note array
t→	Typed relation
t←	Untyped relation
	Class

Symbol	Property type
	External link default
	vi overlay

Understanding search clauses

When you perform your search, Teamcenter examines your search clauses and looks for values that match your search. Search clauses are defined in the **Search Criteria** pane.

Elements of a search criteria clause

Boolean rules	The Boolean rules (**AND**/**OR**) are used to combine clauses to create a custom query. When you use **AND** clauses together, both must be satisfied to return a match. When you use **OR** clauses together, either can be satisfied to return a match. The indented search feature only supports **AND** clauses.
Attribute	The selected database attribute displays in this box.
User Entry L10N Key	Specifies the localization key used to look up user entry names. The localization key-value pairs are defined in **qry_user_entry_names_locale.xml**. The value in this column can be modified and must be unique within the search criteria definition.
User Entry Name	Displays the query box names as they appear in the search form. The user name is the value of the localization key entered in the **User Entry L10N Key** column. If the key-value pair is not defined in the **qry_user_entry_names_locale.xml** file, the user entry name is the same as the key entered in the **User Entry L10N Key** column. The value in this column cannot be modified.
Logical operators	You can specify a range of values using **>**, **>=**, **<**, and **<=**, or invert search criteria using **!=**. Specify one of the following logical operators in each search clause. **=** Equal to **!=** Not equal to **>** Greater than **>=** Greater than or equal to **<** Less than **<=** Less than or equal to **IS_NULL** Reference property value is not set (blank). You may also specify **= $NULL**. **IS_NOT_NULL** Reference property must have a value. You may also specify **!=$NULL**.

	CONTAINS Reference property contains the default value.
Default Value	You can specify a default values for a search clause. Enter a default value as a text string or select it from the associated list of values. This value is required only when you do not specify the user entry name or if the logical operator is **IS_NULL** or **IS_NOT_NULL**. If you make any change to the default Teamcenter queries, the modified values are displayed unless you explicitly enter the variable name over its displayed value. **Default Teamcenter query variables** • Use **$USERID**, **$USERNAME**, and **$GROUP** variables as default values for the end user who is running the query. • The **$TODAY** variable displays the current date. The **$NULL** variable can be a default value when the logical operator is **=** or **!=**. The **User Entry L10N Key** is not required. To use a list of values (LOV) for a property on a form business object, the LOV must be attached to the same property on the form's parent business object. The form parent is the storage class for the properties. Otherwise, the LOV is not attached to the property and does not display in Teamcenter clients.

In the **Search Criteria** pane:

- Make certain the **User Entry L10N Key** value is unique for each clause.
- Use the **AND** operator (rather than **OR**) to place multiple default values, each separated by a semicolon, in the **Default value** column.
  Using **AND** and **OR** clauses together can display unexpected results.

Example

	Attribute	User Entry L10N Key	User Entry Name		Default Value
	object_name	object_name	Name	=	*
AND	object_type	object_type	Type	=	UGMASTER;DirectModel

Sort order

Default sort attributes are defined on the **Order By** tab. In My Teamcenter, use the **Search** view **Sort** dialog box to specify sort order or override the default sort order for a saved query.

The **Order By** tab contains the following search criteria elements.

Attribute	The selected database attribute appears in this box.
User Entry L10N Key	Specifies the localization key used to look up user entry names. The localization key-value pairs are defined in the **qry_user_entry_names_locale.xml** file. The value in this column can be modified and must be unique within the search criteria definition.
User Entry Name	Displays the query box names as they appear in the search form. The user name is the value of the localization key entered in the **User Entry L10N Key** column. If the key-value pair is not defined in the **qry_user_entry_names_locale.xml** file, the user entry name is the same as the key entered in the **User Entry L10N Key** column. The value in this column cannot be modified.
Order By	Specifies the sort order or overrides a default sort order. Order can be either **Ascending** or **Descending**.

1. Create a query.

2. Next to **Search Criteria**, click **Order By** and select an attribute to display the **Order By** tab.

3. In the **Order By** tab, select an attribute to move it up or down in the **Sort By** precedence order list. Specify whether to arrange in **Ascending** or **Descending** order.

Defining search criteria

Specify search criteria by defining statements called clauses in the **Search Criteria** pane. Each clause searches the type and examines a specific property for that type. Each clause can examine only one property. If you want to build a complex query that examines multiple properties, you must add a search clause for each property you want to search.

Search Criteria table

When you perform your search, Teamcenter examines each search clause for matching values. The **Search Criteria** pane defines the elements in the **Search Criteria** table. Double-click an attribute in the **Property Selection** box to add it to the **Search Criteria** table.

You can move clauses ▲ or ▼, or remove a clause using —.

Your changes are committed to the database when you click **Create** or **Modify**.

> Example:
>
> This **Search Criteria** table finds users that meet the person name and user ID search criteria.

	Attribute	User Entry L10N Key	User Entry Name		Default Value
	person.user_name	PersonName	Person Name	=	
AND	user_id	UserId	User Id	=	

Search criteria display

The combination of the **User Entry L10N Key** and **Default Value** elements control how the search criteria is displayed.

1. The **User Entry L10N Key** has a value and the **Default Value** is blank.
   Result: The attribute displays in the saved query for the user to populate.

2. The **User Entry L10N Key** has a value and the **Default Value** has a value.
   Result: The attribute displays in the saved query with the default value. The user can change the default value in the saved query pane.

3. The **User Entry L10N Key** is blank and the **Default Value** has a value.
   Result: The attribute does not display in the saved query. The value is evaluated in the query.

   Note:

   When the **User Entry L10N Key** is blank, the **Default Value** must have a value.

Example

Saved query: **Admin – Object Ownership**, search type: **Workspace Object**

		Attribute	User Entry L10N Key	User Entry Name		Default Value
1		owning_user.user_id	OwningUser	Owning User	=	$USERID
2	AND	owning_group.name	OwningGroup	Owning Group	=	$GROUP
3	AND	object_type	Type	Type	=	ItemRevision

1. The user ID of the **Workspace Object** owning user is the user logged on and running the query.

   Result: The attribute displays the user ID of the current user in the saved query. The query looks for workspace objects owned by the user ID.

2. The name of the **Workspace Object** owning group is the current group of the user logged on and running the query.

Result: The attribute displays the group name of the current user in the saved query. The query looks for workspace objects owned by the group name.

3. The object type of **Workspace Object** is **Item Revision**.

   Result: The attribute displays **Item Revision** in the saved query. The query looks for workspace objects with a matching type.

The resulting saved query:

3. Understanding and managing queries

Create and manage queries

Query requirements

You may create custom queries that search for objects in Teamcenter databases. When you create a query, you must provide the following:

- The search type for the query
- At least one search criteria clause

Create a query

Clear existing information from Query Builder.

1. Type a unique **Name** for the query. You may provide a **Description**.

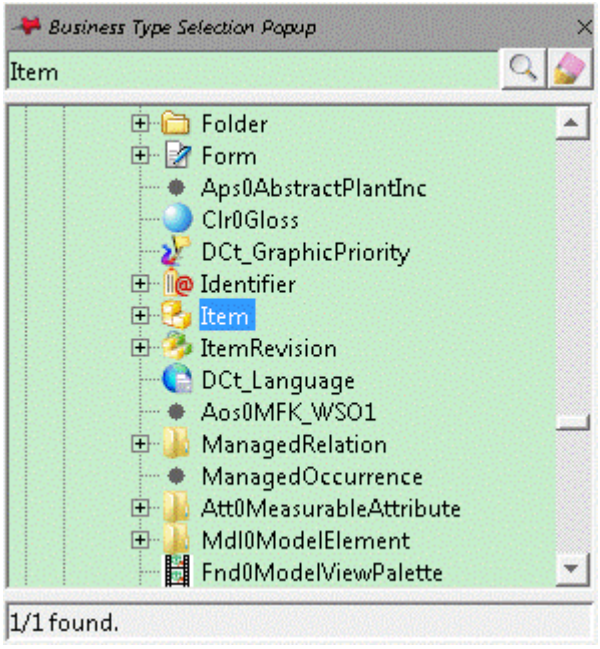
   If you create a query using "__" preceding the name, the query will only be visible to the **dba** group.

2. Select **Local Query** from the **Modifiable Query Types** list.

3. Click **Search Type** .

   The **Business Type Selection** dialog box appears.

   Specify a search type by selecting an entry.



4. You can expand or narrow the focus depending on the type you choose. Limit the search to the lowest possible type in the hierarchy.

   To locate a type, type the type name (or partial name and wildcards) in the box at the top, and click **Search** . The number of types matching your search are displayed at the bottom, and the first result is highlighted in the tree.

5. To display the search results in an indented or hierarchical form, select **Show Indented Results**.

6. Select an applicable revision rule from the **Revision Rule** list.

7. **Select at least one of the properties** in the **Property Selection** pane.

   Direct properties of the type are displayed in the tree. Reference types and properties can be accessed by expanding **Referenced By** in the tree.

   Double-click the property to add it to the **Search Criteria** table.

8. Specify the desired **search criteria**. The following are required:

   - **Attribute**
   - **User Entry L10N Key**
   - **Logical operators**

9. Click **Create** .

The query name appears in the **Saved Queries** tree of Query Builder and in **System Defined Searches** in My Teamcenter.

Modify a query

1. Select a query from **Saved Queries**. Query information is displayed on the right side of the window.

2. Change the information in **Name**, **Description**, or **Search Type**, and/or **Search Criteria** table columns.

3. Click **Modify** to save your changes.

Delete a query

1. Select a query from **Saved Queries**. Query information is displayed on the right side of the window.

2. Click **Delete** and confirm the deletion.

3. Click **Yes** to delete the query from the database and remove it from the **Saved Queries** tree.

Create a query using hints

Query hints help navigate the Persistent Object Manager (POM) schema by presenting a relationship to traverse (for example, item-to-item master form) and build that relationship into your query definition.

1. Type a unique **Name** for the query, and, optionally, a **Description**.

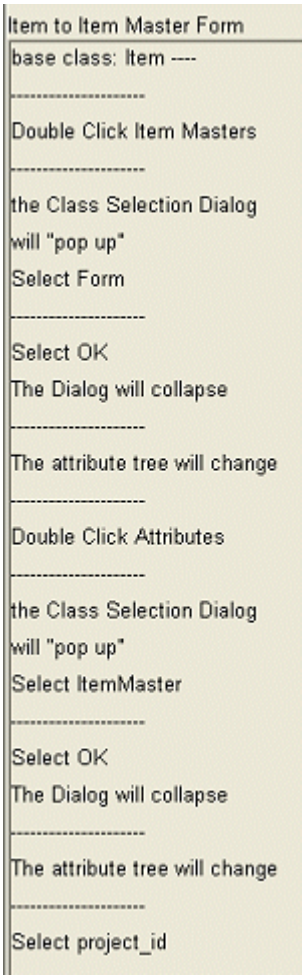
2. Select **Local Query** from the **Modifiable Query Types** list.

3. Click **Show Hints** next to **Search Type** to display the hints pane.

4. Click **Choose Hint** to display the directory of query hints.

5. For this example, expand the **Item Queries** folder, select **Item to Item Master Form**, and click **OK**.

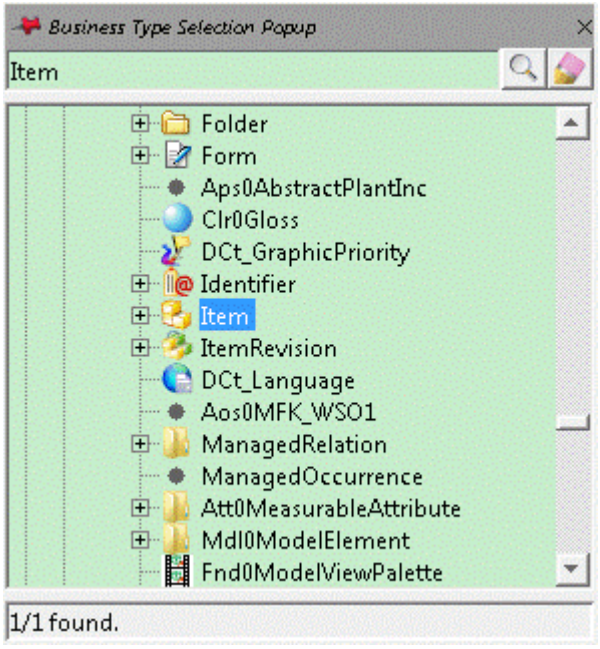
   The selected hint displays in the hint pane. The first entry in the hint is the base type, followed by the traversal steps, ending with the property of the item master form type that is added to the query clause.



6. Click **Search Type**  to select the base type.

7. In **Property Selection**, search  for **Item**.

   The **Item** type is highlighted in the schema tree of **Business Type Selection**.

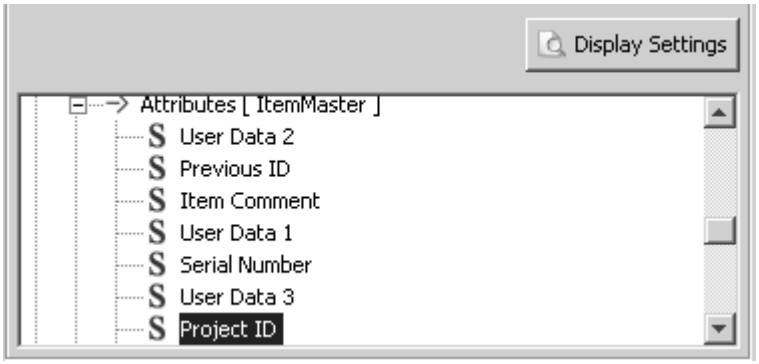


Close **Business Type Selection**.

8. The item type and its properties display in **Property Selection**. The **Find in Tree** and **Add Clause to Table** buttons are available.

9. The **Item to Item Master Form** hint traverses the schema to the **project_id** property from the Item Master Form. To add this property to the query definition, click **Add Clause to Table**.

10. You can also search for the **project_id** property of the Item Master Form. Click **Find in Tree** to display the **Property Selection** pane, and search for the **Project ID** property.



11. Double-click any of the other properties to add them to the clause table.

    You can use other hints to add more clauses to the query table. Select a new hint and repeat the previous steps.

12. Modify the search criteria and clause order as needed.

13. Click **Create** to save the query definition.

The query form is also available in **System Defined Searches** in My Teamcenter.

Create a new query based on an existing definition

Reuse an existing query by modifying it.

1. Select an existing query from the **Saved Queries** tree and enter a unique **Name**.
2. Change the information in the **Description** box, **Search Type** box, and/or **Search Criteria** table columns.
3. Click the **Create** button .

The system adds the query to **Saved Queries**. The query form is also available in **System Defined Searches+** in My Teamcenter.

Create a query using IS_NULL or IS_NOT_NULL

Create queries to find objects with null or non-null property values. For example, you can find items that do not have descriptions or object properties containing any value. Clauses that use the **IS_NULL** and **IS_NOT_NULL** operators are treated as a fixed value; therefore, there is no need to enter a name and default value for the clause.

Example

Create a null-value query to find released **Item Revision** objects

1. Type a unique **Name** for the query, and, optionally, a **Description**.
2. Select **Local Query** from the **Modifiable Query Types** list.
3. Click **Search Type**  to select the target type for the query.

   Type Selection displays the POM schema in tree format.
4. Expand the **POM Application Object** type. Find and expand the **Workspace Object** type.
5. Select **Item Revision** and then close. **Item Revision** is now displayed on **Search Type** and **Item Revision** and its properties are displayed in the **Property Selection** pane.

6. Click **Display Settings** and select **All Properties**. Select the **Release Status** property in the tree, and then click **Add**  .

   The **Release Status** property and default operator (**IS_NULL**) are displayed in **Search Criteria** table.

7. Click the right corner of the operator cell and change **IS_NULL** to **IS_NOT_NULL**.

8. Click the **Create** button  to create the query.

The system adds the query to the **Saved Queries** tree. The query form is also available in **System Defined Searches** in My Teamcenter.

Create a referenced-by query

You can create queries using clauses on a reversed-reference relationship. In the following example, the purpose of the query is to find dataset objects that are referenced, through an **IMAN_specification** relationship, by an item revision with a specific name.

Example

1. Type a unique **Name** for the query, and, optionally, a **Description**.

2. Select **Local Query** from the **Modifiable Query Types** list. Click **Search Type**  to select the target type for the query.

   Property Selection displays the POM schema in tree format.

3. Expand the **POM_application_object** type, then find and expand the **WorkspaceObject** type.

   Double-click the **Dataset** type to display the type and its attributes in the **Property Selection** box.

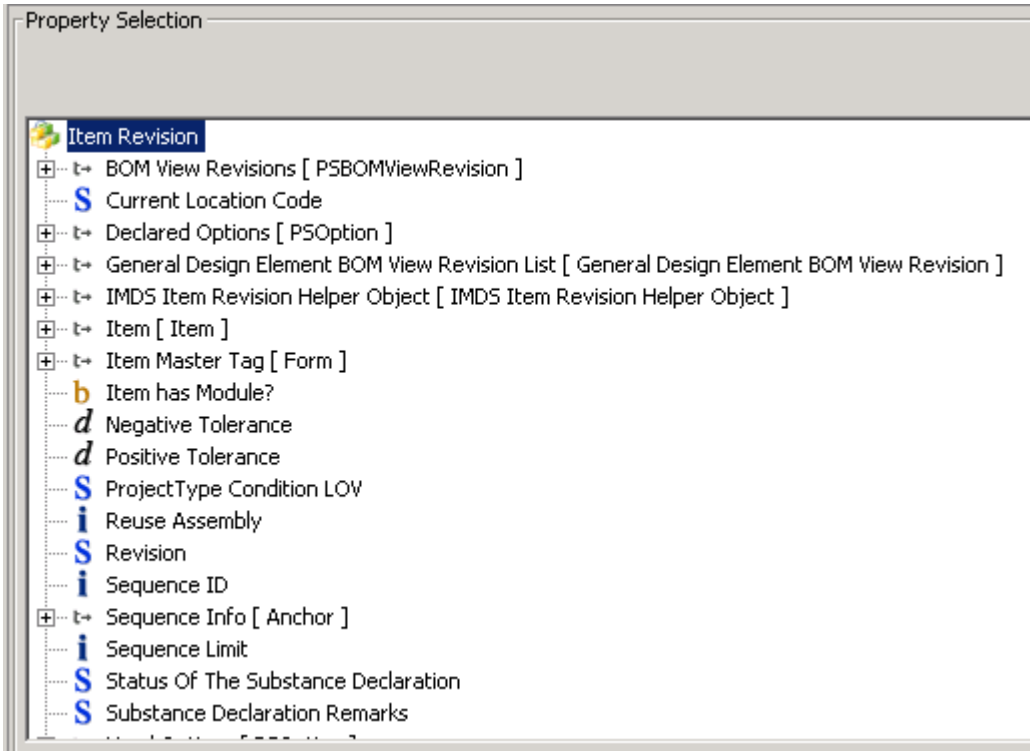
4. Double-click the **Referenced By** node in the **Property Selection** pane to display the **Property Selection** dialog box. Select the type and through which attribute the given object is referenced in that type. You can add selected attributes to the **Search Criteria** table.

5. Click **Search Type**  to select the referencing type for the query.

   Property Selection displays the POM schema in tree format.

6. Expand the **POM_application_object** type and find **WorkspaceObject**. Expand the **WorkspaceObject** type, highlight **ItemRevision** and then close.

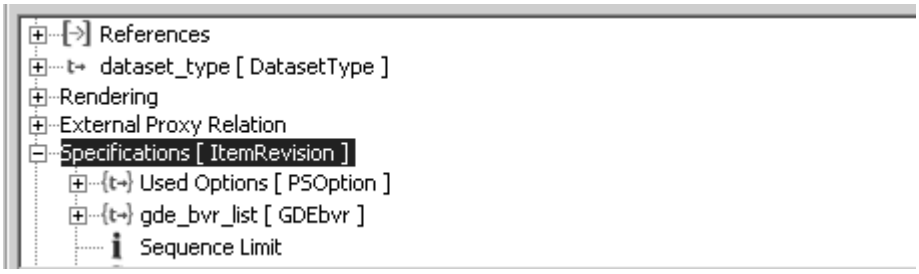
   The referencing type and its attributes are displayed in the **Property Selection** dialog box.



Only those attributes that may reference the objects of the type being queried, in this case the **Dataset** type, are displayed.

7. Double-click a referencing attribute node, in this case **IMAN_specification**.

   Property Selection displays the referencing attribute and type, in this case **Specifications** and **ItemRevision**.



8. Select the attributes of the referencing type on which you want to build query clauses. In this case, find and double-click the **Name** attribute to display it in the **Search Criteria** table. The display name is **ItemRevision←IMAN_specification.object_name**. The ← symbol indicates a reversed-reference relationship.

9. Type a key name in **User Entry Key** to make **User Entry Key** unique when you perform a query from My Teamcenter search. Then click **Create** .

The system adds the query to the **Saved Queries** tree. The query form is also available in **System Defined Searches** in My Teamcenter.

Create a saved query on classification attributes

With this function, and appropriate privileges, you create a saved query on Classification properties using Query Builder. In particular, you can create a saved query that can combine Classification properties and Teamcenter properties. The following examples illustrate a few scenarios:

- Define a query that restricts the search to Bolt and makes all Bolt properties available as the search criteria. In the Find application, the user can enter values for one or more Bolt properties.
- Define a query that restricts the search to Bolt but makes a subset of Bolt properties (for example, Length and Radius) available as the search criteria.
- Define a query that restricts the search to item and makes item properties and bolt properties through the **IMAN_classification** reference be available as the search criteria.

1. Select a **Classification** type from the type schema using the **Type Selection** dialog box, accessed by clicking the **Search Type** button  .

   When creating a saved query, the search type must have at least one of its own properties (a property not inherited from its parent type). Otherwise, the search returns objects from its parent type, rather than the specified type.

2. After selecting a **Classification** type, select Classification properties through the **Property Selection** pane to build query clauses.

The Classification hierarchy used by Query Builder is exactly the same as the Classification hierarchy used in the Classification applications. The highest level of the hierarchy represents groups that do not contain properties and cannot be used in a search. These group nodes display a different icon. The nodes below the group level correspond to a **Classification** type. The type level nodes can be used to build saved queries.

After a Classification property is selected from the **Property Selection** tree, its internal property ID is shown as **Property** name in the **Search Criteria** table while its property display name is shown as **User Entry Name** in the **Search Criteria** table. You can modify the user entry name if required.

A Classification object (ICO) is attached to a Teamcenter object through the **Tc_classification** relationship. This **Tc_classification** relationship is shown as one of the properties for a Teamcenter type whose instances are classified. For such a Teamcenter type (for example, item and item revisions), you can expand its **Tc_classification** property in the **Property Selection** tree to access the Classification properties. In doing so, you build a saved query that combines Classification property search criteria and Teamcenter property search criteria.

Note:

When you double-click the **Tc_classification** property, only the Classification hierarchy is shown in the **Type Selection** dialog box to enable you to select a Classification type and use its properties to build query clauses.

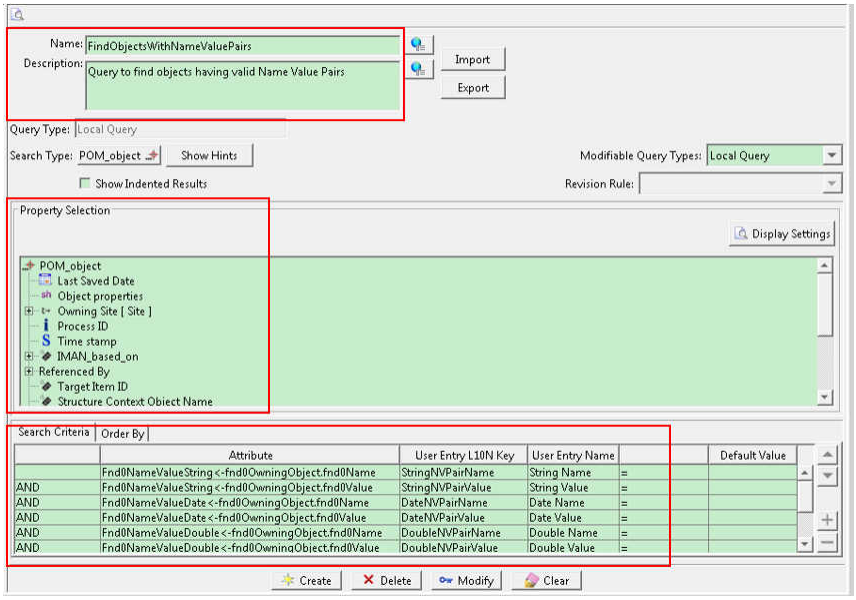
Note:

Classification LOVs do not show up as cascading LOVs when used in saved queries or with extended multi-application search. Instead they are displayed as a flat list with all selectable entries.

Create a saved query for name-value properties

To query for name-value property information on a business object, select **FindObjectsWithNameValuePairs**.

The **Name**, **Description**, **Property Selection** and **Search Criteria** display the information for **FindObjectsWithNameValuePairs** query.



If you prefer, you can create your own **saved query for table properties**.

Create a saved query for table properties

Create a saved query for table properties or name-value properties.

1. Type a unique **Name** for the query, and, optionally, a **Description**.

2. Select **Local Query** from the **Modifiable Query Types** list. Click **Search Type** to select the target type for the query.

3. Expand the **POM_application_object** type and locate the **WorkspaceObject** type.

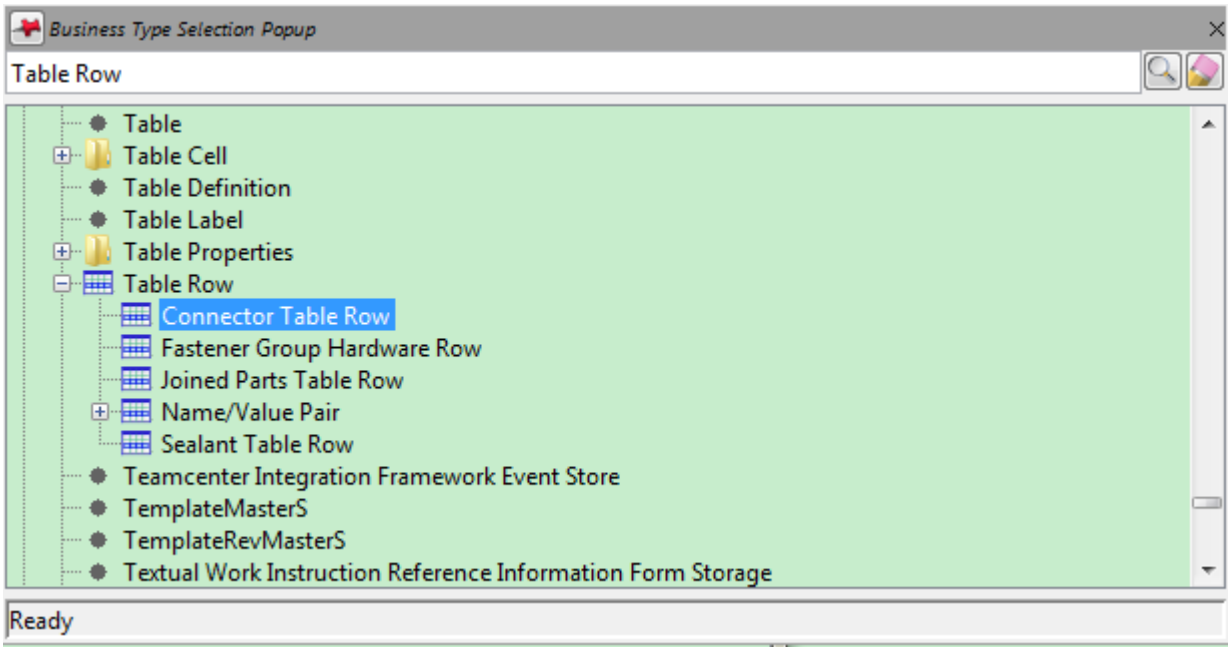
4. Click **Display Settings** and select **All Properties** and **Real Names**.

5. In the **Property Selection** box, double-click the **Referenced_By** node.

   The **Type Property Selection** dialog box appears.

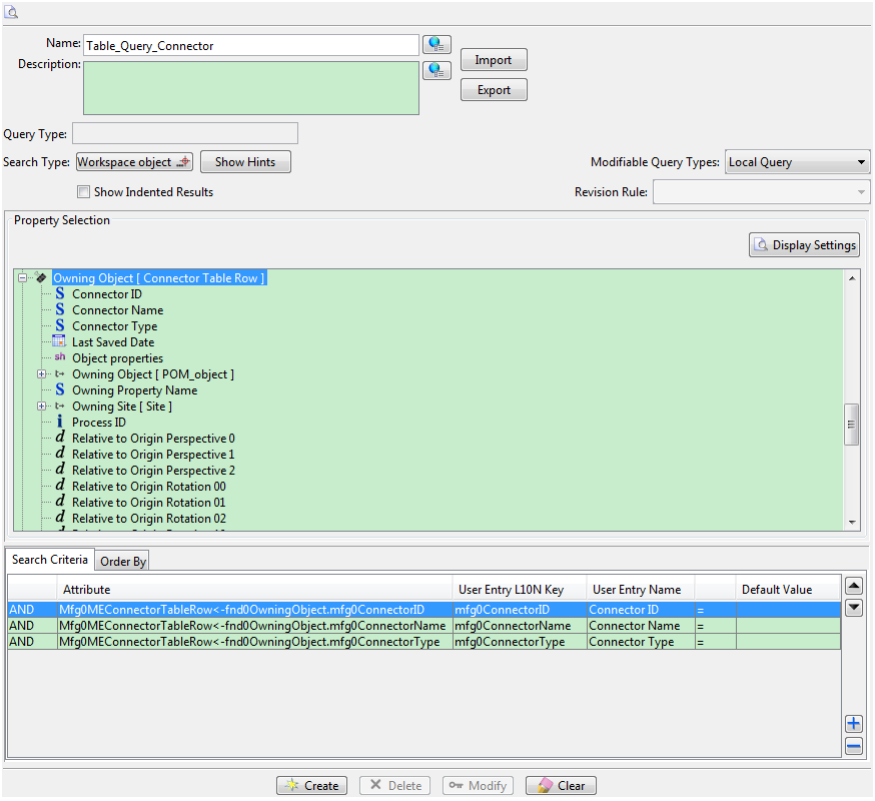
6. Click **Search Type** and search for your table row business object.

   Under the **Fnd0TableRow** node, see:



7. Select the table row business object and open **fnd0OwningObject**, which lists the table row properties for your search.

8. Add the owning object and build your search by selecting the attributes on which you want to build the query clauses. When finished, click **Create** to create the query.



Your new query now appears in the **Saved Queries** tree pane and is available in **Select a Query** in My Teamcenter.

Create a subtype query on a typed reference

You can create saved queries based on a subtype of a typed reference and then build clauses against the properties of the subtype instead of against the properties of the referenced type. For example, the contents property in the **Folder** type is a typed reference of the referenced type **WorkspaceObject**. You can, therefore, select the **Dataset** type as a referenced type of the contents property and build query clauses against the properties of the **Dataset** type, rather than against the **WorkspaceObject** type.

The following steps show how to create a query of a subtype on a typed reference. In this case, the purpose of the query is to find folders containing datasets that are checked out.

1. Type a unique **Name** for the query, and, optionally, a **Description**.

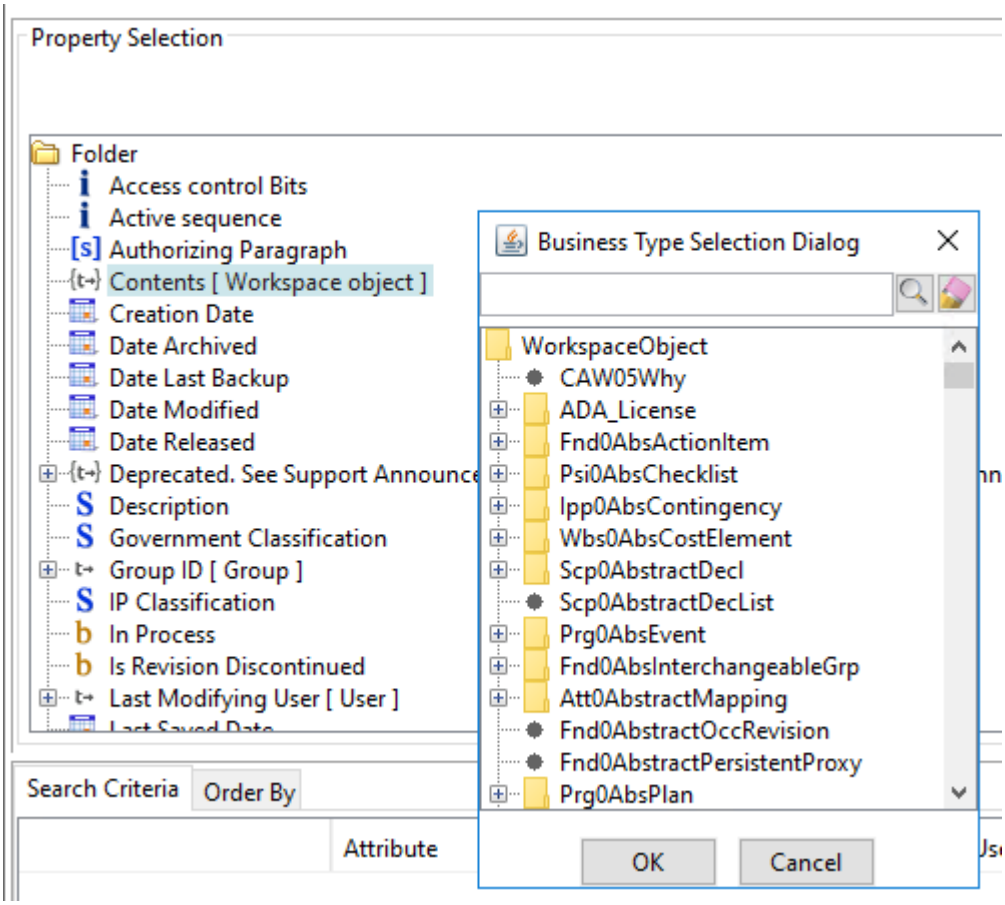
2. Click **Search Type**  to select the target type. The **Business Type Selection Popup** displays the business objects in tree format.

3. Expand the **POM Application Object** type. Find the **Workspace Object** type and expand it. Select the **Folder** type, and close.

Folder is displayed on the **Search Type** button, and the **Folder** type and its properties are displayed in the **Property Selection** pane.

4. In **Property Selection**, double-click the **Contents [Workspace object]** property.

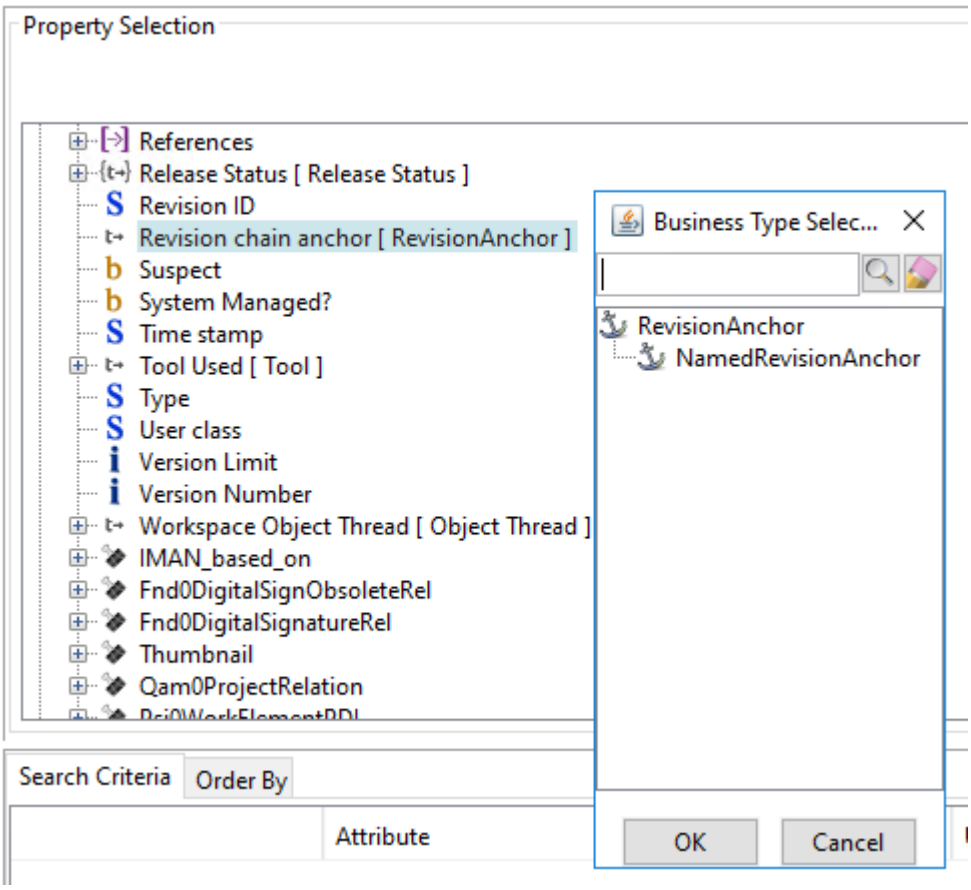
   The **Business Type Selection Dialog** displays the referenced type, in this case, **WorkspaceObject**, and all of its subtypes.



5. Double-click the **Dataset** subtype in the tree.

   The **Property Selection** pane displays the **Contents [Dataset]** properties and its properties of the **Dataset** type.

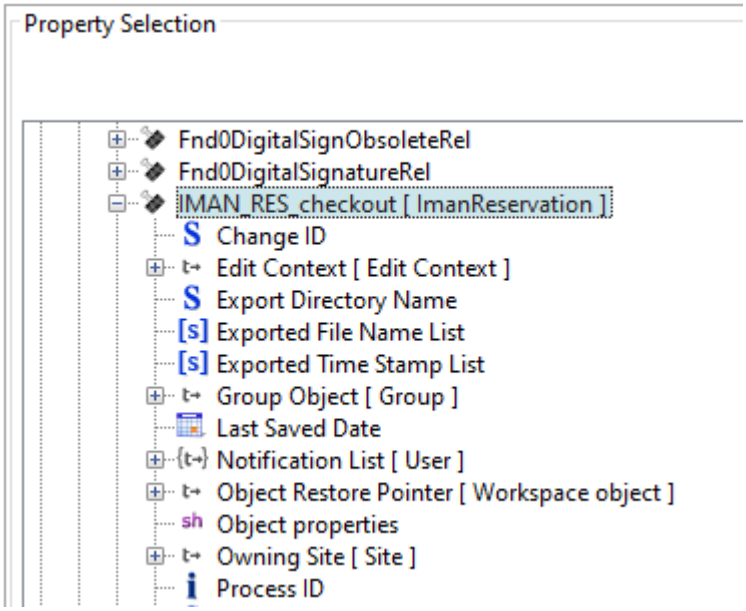
6. Double-click the **Revision chain anchor [RevisionAnchor]** property node. The **Business Type Selection Dialog** displays the **RevisionAnchor** type.



Expand the **RevisionAnchor** in the tree to display the properties in the **Property Selection** pane.

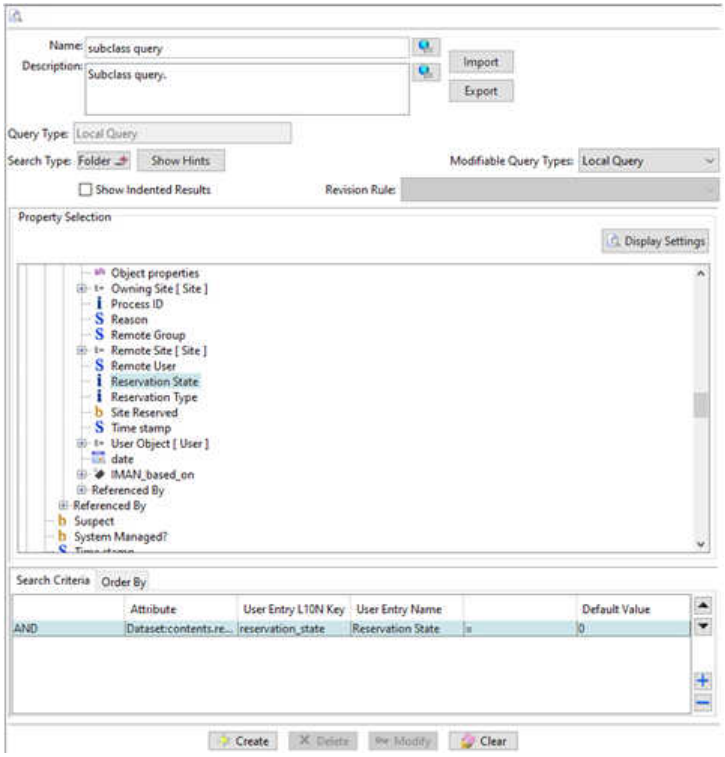
7. Double-click the **IMAN_RES_checkout** type. The **Business Type Selection Dialog** displays the business object schema.

   Expand the **ImanReservation** type to display its properties in the **Property Selection** pane.



Double-click the **Reservation State** property to add it to the **Search Criteria** table.

8. Type a default value **0** for this query clause.



9. Click **Create** to create the query.

The system adds the query to the **Saved Queries** tree. The query form is also available in **System Defined Searches** in My Teamcenter.

Using preferences when querying

When querying, the following preferences can further refine your query:

- **QRY_exception_list**
  Specifies the list of queries to be excluded from using join approach on relations and references.

- **QRY_*query_name*_SortKeys**
  Determines the type property used to sort the query results. This preference must be used with the **QRY_*query_name*_SortOrders** preference, which determines sort order.

- **QRY_*query_name*_SortOrders**
  Determines the sort order of query results. This preference must be used with the **QRY_*query_name*_SortKeys** preference, which determines the type property used to sort query results.

- **QRY_*query_name*_REVRULE**
  Specifies the revision rule applied to search results for searches against **ItemRevision** and its subclasses.

- **TC_QRY_search_by_rev_rule**
  Finds the latest item revisions based on revision rules. For example, when querying for the latest working item revision, you can set this preference to **Latest Working** before running your query.

- **TC_QRY_display_latest_dataset_only**
  Displays the latest version of dataset objects when you set this preference to **1** before running your query.

- **SEARCH_RESULT_LOAD_ALL_LIMIT**
  Specifies the maximum number of loaded search results (default is **400**).

- **Open_previous_open_search**
  Display a previous search result in the search results view when you set this preference to **1**.

Property finder formatter (PFF)

Search results can display property data related to objects returned in the results. Property finder formatter (PFF) objects can locate properties related to objects in **Search Results** without reformatting the query. You can display these properties in the search results using a selected PFF. By navigating object relationships to locate properties of the object, a PFF organizes the display of properties in the search results table.

A set of PFF objects corresponding to the basic search types are part of the Teamcenter installation. Additional PFF objects may be defined by a Teamcenter administrator, as this action requires knowledge of property data storage and relationships between data objects. If you build queries using property finder formatter objects, you cannot include the dot character (**.**) in type names.

Adding a property finder to your query

1. Run the search for which you want to create a PFF.

2. Click  to launch the **Edit** window.

3. Enter a **Name** which reflects the related query and an optional **Description**.

4. Select a type for the PFF criteria. Click **Principle Query Object** and select the type that matches the base type of the query. When you close, the type appears in **Principle Query Object**.

   If you don't know the base type, view the query details in the search criteria definition.

   The type and its properties appear in **Property Selection**.

   You can display all properties, rather than only type properties, by clicking **Display Setting** and selecting **All Properties**.

5. Add a search clause to **Search Criteria**. In **Property Selection**, double-click a property to add it to the **Search Criteria** table.

   If you change display settings after adding search clauses to the **Search Criteria** table, the table is cleared.

6. You may include a secondary type in the PFF property search criteria, and add the related search clauses to the **Search Criteria** table.

   Click **Add Clauses From** and select a secondary type that matches one of the object types returned by the original query.

   When you close, the selected type appears in **Add Clauses From**. The type and its properties appear in **Property Selection**.

   Add a search clause to the **Search Criteria** table for each property of the secondary type in the same manner you added one for the primary type.

7. To create the PFF, click **Create**.

   If you specify criteria for a PFF for an ad hoc query but do not click **Create**, the file size, byte size, and MIME type information is not displayed in the ad hoc results.

8. Verify that the PFF search returns what you expect. In the **Search Results** view, open the PFF list and choose the one you created. The results are applied to the **Search Results** objects.

The PFF object also appears in **Saved Property Formatter Finder** in the **Edit** window. You may open it and make changes from there.

Attribute index

If there are more than 5,000 instances of the search type to be found, you must add an index to the attribute on which you are searching. Doing this helps search performance. However, if the table is very small (that is, the type has less than 500 instances), a full-table scan is more efficient than an index scan.

If you do not create an index on an attribute, a full-table scan of the Oracle table automatically takes place. This affects the System Global Area (SGA) of the Oracle server. SGA is the portion of memory where Oracle caches queries and their execution plan. You can use the **install** utility **-add_index** argument to add an index for an attribute.

4. Managing Query Builder files

Importing and exporting query definitions

Query definitions can be exported and saved as XML files for sharing with other Teamcenter sites. Conversely, query data saved in XML files can be imported into Teamcenter. The XML files are parsed and verified before the data is imported.

Importing query definitions

You can import a query definition from an XML file and create a corresponding query in the Teamcenter database. Correctly formatted data in the XML file may be incompatible with the local database schema, causing an error when creating a query definition from incompatible data.

1. Open **Query Builder** and click **Import**, which displays the last query definition file that was imported.

2. Click **Browse**  to find the XML file containing the definition you want to import.

3. In the **Read Query Definition**, find the XML file and click **Import**, which displays the contents of the XML file.

4. Click **Verify**. If the file format is valid, the query data is displayed in the Query Builder pane.

   Verify validates that the POM class matches existing classes in the database. If there are parser errors, a message describes the nature of the errors.

5. Click **OK** to load the query in the saved query tree. Then close **Import**.

6. Optionally, you may modify the name, description, or query clauses.

7. Click **Create**  .

The system verifies that the definition is compatible with the local database schema. If so, the query is saved in the database. If not, an error message describes the discrepancies.

Exporting query definitions

You can export a query definition to share with other sites:

1. Select the query in the **Saved Queries** tree to display the query definition in the right pane.

2. Click **Export**. **Print** displays the query in XML format.

3. Click **Save**  to specify the path and file where you want to save the definition. Include the **.xml** file extension in the file name.

4. After you click **Save**, close **Print**.

Restore My Saved Searches folder

If you mistakenly delete your **My Saved Searches** folder, you can create a query to restore your saved searches using the following steps.

1. Type a unique **Name** for the query, and, optionally, a **Description**.

2. Click **Search Type**  to select the target type for the query.

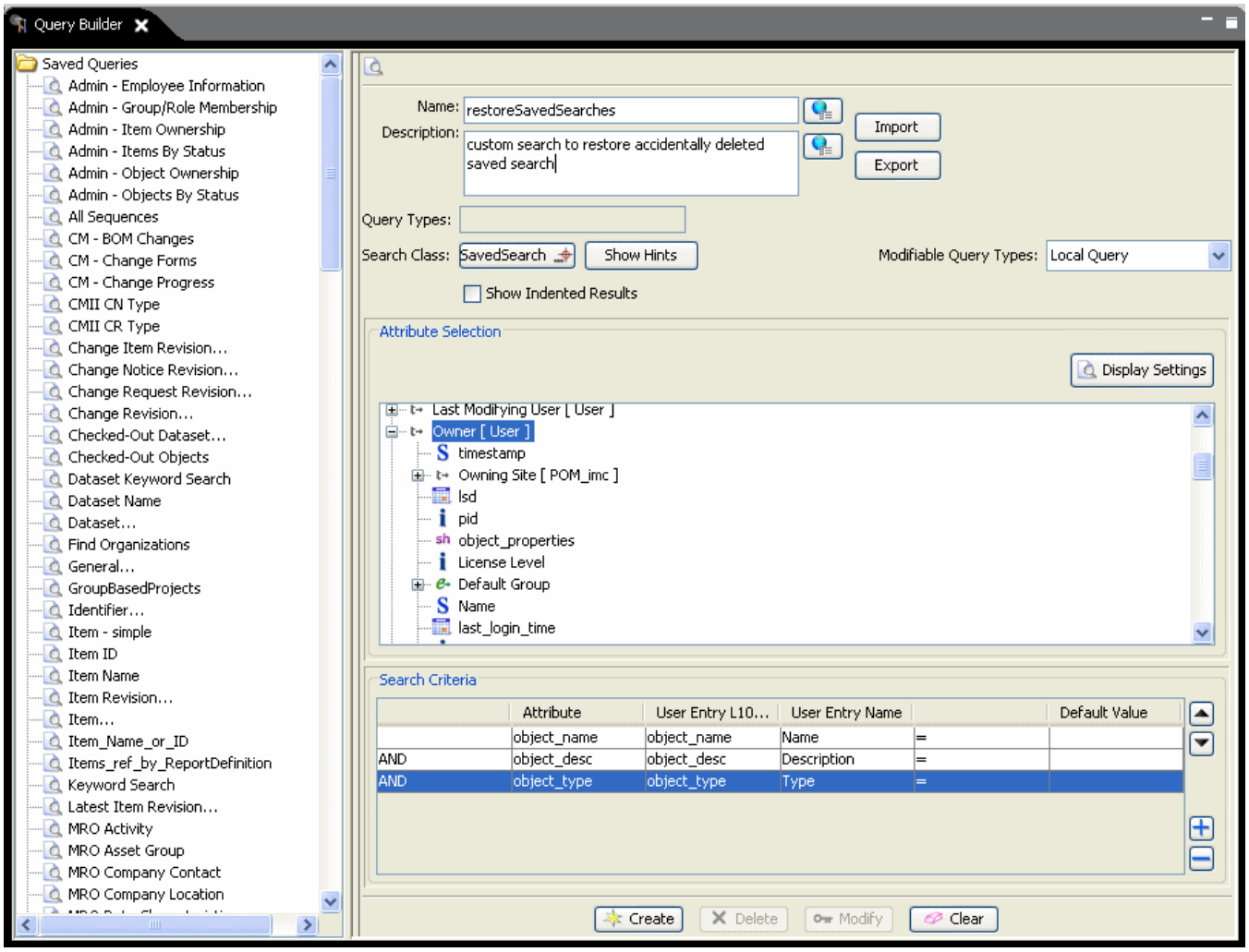
   Type Selection displays the business objects in tree format.

3. Expand the **POM Application Object** type and locate the **Saved Search** type.

4. In **Property Selection**, click **Display Settings**. Select **All Properties** and then **Display Names**.

5. Double-click the **Owner** subtype in the tree. **Property Selection** displays the **Owner [User]** property and its properties of the **Saved Search** type.

6. In the **Search Criteria** pane, specify **Name**, **Description**, and **Query Type**.



7. Click **Create** to create the query.

The system restores the **Saved Queries** tree.

The query hints file

Query hints are defined in the **queryHint.xml** file.

It is stored in the **CrfOutputXml** dataset in **TC OOTB** as **Fnd0SavedQueryHint**.

The XML file contains two element types:

- Folders
  Used to group hints by category. They have two associated attributes: folder name and description. The folder name is displayed in the **Choose Hint** dialog box tree and the description is the tool tip displayed when the user places the mouse over the folder. Folders can be nested to arbitrary depths and can contain both folders and hints.

- Hints
  Contain four fields: **name**, **description**, **search type**, and **query clause**. The hint name is the text you see in the hint tree. The description is the tool tip displayed when the user hovers over the hint. The search type is the origin of the query, and the query clause is the hint that provides the path to the attribute.

The simplest way to obtain the query clause is to copy it from the Query Builder **Search Criteria** table from an existing query and paste it into the query hints XML file.

Example query hints file

If you edit the file in a text editor, convert the angle brackets (>) and (<) characters to the XML equivalents, **>** and **<**. There is no mechanism during an upgrade to preserve customizations to the **Fnd0SavedQueryHint** dataset.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Define dtd for tc query hints
 -->

<!DOCTYPE ImanQueryHintFile [

    <!ELEMENT folder ( (folder | hint)* )>
    <!ATTLIST folder name        CDATA #REQUIRED
                     description CDATA #REQUIRED>
    <!ELEMENT hint EMPTY>
    <!ATTLIST hint name        CDATA #REQUIRED
                   description CDATA #REQUIRED
                   class       CDATA #REQUIRED
                   hint_text   CDATA #REQUIRED>

    <!ELEMENT ImanQueryHintFile ((folder | hint)*)>
]>


<ImanQueryHintFile>
<!-- ============================================== -->
<!-- ========= item hints ========================= -->
<!-- ============================================== -->
<folder name="Item Queries" description="Query Hints for Item Relationships">
    <hint name="Item to Item Revision" description="adding item revision criteria to the
item
     search" class="Item" hint_text="ItemRevision&lt;-items_tag.item_revision_id"/>
    <hint name="Item to Item Master Form" description="adding master form criteria to the
item
     search" class="Item" hint_text="Form:IMAN_master_form.ItemMaster:data_file.project_id"/>
    <hint name="Item to Item Revision Form" description="adding criteria from an item
     revision master form into the item search" class="Item"
     hint_text="ItemRevision&lt;-items_tag.Form:IMAN_master_form_rev.ItemVersionMaster:
     data_file.project_id"/>
    <hint name="Item to IMAN_Specification" description="adding criteria from an underlying
     item revision's dataset to the item search" class="Item"
     hint_text="ItemRevision&lt;-items_tag.Dataset:IMAN_specification.format_used"/>
    <hint name="Item to IMAN_Aliasid" description="adding criteria from an associated alias
     identifier to the item search" class="Item"
     hint_text="Alias:IMAN_aliasid.alias_id"/>
```

```
</folder>
<!-- ================================================ -->
<!-- ========= item revision hints ================== -->
<!-- ================================================ -->
<folder name="Item Revision Queries" description="Query Hints for Item Revision
Relationships">
    <hint name="Item Revision to IMAN_Specification" description="adding dataset criteria
     to the item revision search" class="ItemRevision"
     hint_text="Dataset:IMAN_specification.format_used"/>
    <hint name="Item Revision to Item Revision Master Form" description="adding master
     form criteria to the item revision search" class="ItemRevision"
     hint_text="Form:IMAN_master_form_rev.ItemVersionMaster:data_file.project_id"/>
    <hint name="Item Revision to Item" description="adding item criteria to the item
revision
     search" class="ItemRevision" hint_text="items_tag.item_id"/>
</folder>
<!-- ================================================ -->
<!-- ========= dataset hints ================== -->
<!-- ================================================ -->
<folder name="Dataset Queries" description="Query Hints for Dataset Relationships">
    <hint name="Specification Dataset to owning Item Revision" description="adding item
revision
     criteria to the specification dataset search" class="Dataset"
     hint_text="ItemRevision&lt;-IMAN_specification.object_name"/>
    <hint name="Dataset with unknown relation to Item Revision" description="adding item
     revision criteria to the dataset search" class="Dataset"
     hint_text="ImanRelation&lt;-secondary_object.ItemRevision:primary_object.object_name"/>
</folder>
<!-- ================================================ -->
<!-- ========= folder hints ================== -->
<!-- ================================================ -->
<folder name="Folder Queries" description="Query Hints for Folder Relationships">
    <hint name="Contents to owning Folder" description="adding folder criteria
     to the content object search" class="WorkspaceObject"
     hint_text="Folder&lt;-contents.object_name"/>
    <hint name="Folder to Contents" description="adding content object criteria
     to the folder search" class="Folder" hint_text="contents.object_name"/>
</folder>
<!-- ================================================ -->
<!-- ========= product structure hints ================== -->
<!-- ================================================ -->
<folder name="Product Structure Queries" description="Query Hints for Product
Structure Relationships">
    <hint name="Assembly to child Components (Imprecise Structure)"
     description="adding child component item criteria to the assembly item
     revision search" class="ItemRevision"
     hint_text="structure_revisions.PSOccurrence&lt;-parent_bvr.Item:child_item.item_id"/>
    <hint name="Child Components to Assembly (Imprecise Structure)"
     description="adding assembly item
     revision criteria to the child component item search" class="Item" hint_text=

"PSOccurrence&lt;-child_item.parent_bvr.ItemRevision&lt;-structure_revisions.object_name"/>
    <hint name="Assembly to child Components (Precise Structure)" description="adding child
     component item revision criteria to the assembly item revision search"
class="ItemRevision"

hint_text="structure_revisions.PSOccurrence&lt;-parent_bvr.ItemRevision:child_item.object_nam
e"/>
    <hint name="Child Components to Assembly (Precise Structure)" description="adding
assembly
```

```
     item revision criteria to the child component item revision search"
class="ItemRevision" hint_text
="PSOccurrence&lt;-child_item.parent_bvr.ItemRevision&lt;-structure_revisions.object_name"/>
</folder>
<!-- ================================================ -->
<!-- ========= workspace object hints ============== -->
<!-- ================================================ -->
<folder name="Workspace Object" description="Query Hints for Workspace Objects">
    <hint name="Contents to owning Folder" description="adding folder criteria to the
content
     object search" class="WorkspaceObject" hint_text="Folder&lt;-contents.object_name"/>
    <hint name="Workspace Object owner id" description="adding owner criteria to the
workspace
     object search" class="WorkspaceObject" hint_text="owning_user.user_id"/>
    <!-- note that this hint is repeated below.
         hints can be duplicated under different folder haedings as needed -->
    <hint name="Workspace Object in an EPMJob" description="adding EPMJob criteria
     to the workspace object search" class="WorkspaceObject"
     hint_text="EPMTask&lt;-attachments.EPMJob&lt;-root_task.object_name"/>
</folder>
<!-- ================================================ -->
<!-- ========= EPMJob  hints ======================== -->
<!-- ================================================ -->
<folder name="Workflows" description="Query Hints for EPMJobs">
    <hint name="Job by state" description="adding job state criteria to the EPMJob search"
     class="EPMJob" hint_text="root_task.state_value"/>
    <!-- note that this hint is repeated above.
         hints can be duplicated under different folder headings as needed -->
    <hint name="Workspace Object in an EPMJob" description="adding EPMJob criteria
     to the workspace object search" class="WorkspaceObject"
     hint_text="EPMTask&lt;-attachments.EPMJob&lt;-root_task.object_name"/>
</folder>
<!-- ================================================== -->
<!-- === one of every relationship type  hints ========= -->
<!--                                                    -->
<!-- This block includes an example of each type of     -->
<!-- hint in a relationship by relationship type basis  -->
<!--                                                    -->
<!-- It is not intended that these be made available    -->
<!-- in a production environment. This is why the are   -->
<!-- commented out                                      -->
<!--                                                    -->
<!-- The below hints also demonstrate the sub-folder    -->
<!-- capability of the hint tree. It is possibly to     -->
<!-- organize hints by tc class (as above) or by        -->
<!-- organization - as detailed below.                  -->
<!--                                                    -->
<!--                                                    -->
<!-- ================================================== -->
<!--  commented out...
<folder name="Dept 1 hints" description="hints for joe's department">
    <hint name="simpleName" description="no levels of indirection" class="Item"
     hint_text="item_id"/>
    <hint name="inClass" description="inClass classification... " class="Item"
     hint_text="&lt;Complex_analysis&gt;:IMAN_Classification.1000"/>
    <hint name="inClass2" description="inClass classification...with funny name "
     class="Item" hint_text="&lt;33333&gt;:IMAN_Classification.1003"/>
    <hint name="alias" description="iman alias relationship" class="Item"
     hint_text="Alias:iman_alias"/>
    <hint name="owner id" description="one level of indirection" class="Item"
```

```
     hint_text="owning_user.user_id"/>
    <hint name="item rev reference" description="references example" class="Item"
     hint_text="ItemRevision&lt;-items_tag.owning_user.user_id"/>
</folder>
<folder name="Designer's Hints" description="a list of queries used in the design group">
    <folder name="engine design" description="yes we can nest n-levels deep">
        <hint name="simpleName" description="no levels of indirection" class="Item"
         hint_text="item_id"/>
        <hint name="inClass" description="inClass classification... " class="Item"
         hint_text="&lt;Complex_analysis&gt;:IMAN_Classification.1000"/>
        <hint name="alias" description="iman alias relationship" class="Item"
         hint_text="Alias:iman_alias"/>
        <hint name="owner id" description="one level of indirection" class="WorkspaceObject"
         hint_text="owning_user.user_id"/>
        <hint name="item rev reference" description="references example" class="Item"
         hint_text="ItemRevision&lt;-items_tag.owning_user.user_id"/>
    </folder>
    <hint name="simpleName" description="no levels of indirection" class="Item"
     hint_text="item_id"/>
    <hint name="inClass" description="inClass classification... " class="Item"
     hint_text="&lt;Complex_analysis&gt;:IMAN_Classification.1000"/>
    <hint name="alias" description="iman alias relationship" class="Item"
     hint_text="Alias:iman_alias"/>
    <hint name="owner id" description="one level of indirection" class="WorkspaceObject"
     hint_text="owning_user.user_id"/>
    <hint name="item rev reference" description="references example" class="Item"
     hint_text="ItemRevision&lt;-items_tag.owning_user"/>
</folder>
-->
</ImanQueryHintFile>
```

5. Geolus integration with Teamcenter

Configure the integration

Configure the display of inaccessible results

Some Geolus databases may contain parts relating to Item Revisions that are inaccessible by the rich client user. By default, the Geolus integration with Teamcenter omits inaccessible Item Revisions from the search results displayed. However, there are preferences which can be set enable the display of inaccessible search results and control what information is displayed. The only functionality available to the user for an inaccessible result, when displayed as part of a shape search, is to display some of the textual information held by Geolus associated with that Item Revision and the thumbnail image. Whether to display the thumbnail image or not and what textual information to make available can be configured by the Teamcenter administrator.

A result is classed as inaccessible if it does not exist on the current Teamcenter site (it is owned by a different site and has not been published/exported), or if the currently logged-in Teamcenter user does not have permission to access the Item Revision.

The display of inaccessible results is disabled by default. The preference to enable it (**Geolus_IA_Enabled**) is read at the GROUP scope level so that it can be controlled by the Teamcenter administrator and not by individual users.

Preferences allow the individual elements of inaccessible results to be shown or hidden. There are preferences to control how the thumbnails are faded so that they appear different, and there are preferences which allow the padlock symbol and other images to be customized.

Teamcenter Preference	**Allowed Values** **[default]**	**Notes**
Geolus_IA_Enabled	**ON or OFF** **[OFF]**	The master switch **-ON=**show inaccessible results. Read at **GROUP** level.
Geolus_IA_ShowThumb	**ON or OFF** **[OFF]**	If thumbnails are not shown, the image from **Geolus_IA_ImagesDisabledImage** preference is shown instead. Read at **GROUP** level.
Geolus_IA_ShowName	**ON or OFF** **[OFF]**	**ON/OFF** switch controlling if the name is shown for inaccessible results or not. Accessible results are unaffected. Read at **GROUP** level.

Geolus_IA_ShowSize	**ON or OFF** **[OFF]**	**ON/OFF** switch controlling if the size match value is shown for inaccessible results or not. Accessible results are unaffected. Read at **GROUP** level.
Geolus_IA_InfoFormat	<special>	See below. Read at **GROUP** level.
Geolus_IA_FadeFactor	**0.0 to 1.0** **[0.67]**	Any floating point number between **0.0** and **1.0**. This controls the transparency of the thumbnails. **0.0=**invisible,**1.0=**opaque (the same as accessible results). Read at **USER** level.
Geolus_IA_DesatFactor	**0.0 to 1.0** **[0.67]**	Any floating point number between **0.0** and **1.0**. This controls the saturation of the thumbnails. **0.0=**grey,**1.0=**original color. Read at **USER** level.
Geolus_IA_Dither	**ON or OFF** **[OFF]**	**ON/OFF** switch controlling if the thumbnail is dithered (50% pixels set to transparent, 50% untouched). Read at **USER** level.
Geolus_IA_OverlayImage	File name or URL [notavailable.png]	This is the padlock image which is displayed over the inaccessible result thumbnails. It can be changed to a file name, which must be in the results *plugin .jar* file, or a URL such as **http://server/mypicture.png**. Read at **GROUP** level.
Geolus_IA_ImagesDisabledImage	File name or URL [imagesdisabled.png]	This image is used in place of the inaccessible thumbnails, when the preference **Geolus_IA_ShowThumb** is set to **OFF**. It can be changed to a file name, which must be in the results *plugin.jar* file, or a URL such as **http://server/mypicture.png**. Read at **GROUP** level.
Geolus_Results_NoPictureImage	File name or URL [nopicture.png]	This image is used when thumbnails cannot be accessed (such as when the web server is down). It can be changed to a file name, which must be in the results *plug.jar* file, or a URL

		such as **http://server/mypicture.png**. Read at **GROUP** level.

Any image files specified should be located in the *icons* subdirectory of the *results* JAR file.

The preference **Geolus_IA_InfoFormat** controls the information displayed in the tool-tip box when the mouse hovers over an inaccessible result. The same text can also be copied to the clipboard by right-clicking the result and selecting **Copy Inaccessible Text**.

This preference may contain any text. Special values can also be used so that attribute values from Geolus can be displayed. If any of these special values are put into the preference, they are replaced with the corresponding attribute value for each result.

%00%- The first generic attribute (typically Item Revision UID)
%01%- The second generic attribute (typically Item ID)
%02%- The third generic attribute
...
%19%- and so on up to 19.

Additionally, all occurrences of **\n** are replaced by newlines.

The default value includes the Item ID, Name, Revision and Owning User:

Inaccessible result \n Item ID = %01%\nItem/Rev = %02%/%03%\nOwner = %04%

To assist in setting preferences, the defaults are in an XML file that can be imported directly to the Teamcenter server using the **preferences_manager** utility.

The *geolus_ia_prefs.xml* XML file can be found in the folder:

GEOLUS_HOME\GeolusTeamcenterIntegration\TC_VERSION

It must be copied to the Teamcenter server for import.

The recommended command for importing the preferences to Teamcenter is (enter as a single line):

```
preferences_manager -u=username -p=password -g=group -mode=import -file=geolus_ia_prefs.xml
-scope=GROUP -action=OVERRIDE
```

The **GROUP** scope is recommended but can be changed. If **GROUP** is used, the preferences must be imported for each group of users on the Teamcenter server. The **SITE** scope can be used to specify the preferences once for the entire Teamcenter site.

Configure shape match control

Modifying the search describes how the user controls the degree of shape match in a search. This interface control will be the same in the modify search and the preferences dialogs. By default, this interface control will match the one used on Geolus web pages in the browser interface.

To do this, set the Teamcenter preference **GeolusSimilaritySelector** to **drop-down** or **gradient**.

It is recommended that this is done in the Teamcenter preferences interface, or using the **preferences_manager** tool. Note that this is usually a **SITE** wide preference.

Enter the following on the command line:

```
preferences_manager -mode=import -scope=site
-preference=GeolusSimilaritySelector
    -values=keyword -u=user -p=password -g=group -action=OVERRIDE
```

where the value of **keyword** can be either **drop-down** or **gradient**.

Troubleshoot errors

Any errors that occur during the search will typically be reported on the Geolus **Search Results** page or in a popup.

Error Message	Explanation
Geolus Search can only be performed on Item Revisions that contain a DirectModel DataSet	The Item Revision selected to initiate the search did not contain a DirectModel (or JT) dataset.
Error connecting to the Geolus Server. Please check that the Teamcenter preference **GeolusServer** is specified correctly.	The **GeolusServer** preference has been set incorrectly, or the Geolus Server to which it refers is not available. The Geolus administrator should verify the setting of the **GeolusServer** preference and the status of the Geolus server.
Part not found in Geolus Database. Error occurred during extraction: Model doesn't contain triangles	The JT dataset associated with the part contains no geometry.
Part not found in Geolus Database. Error occurred during extraction: Java heap space	The JT dataset was too large to be indexed in memory.
No Results Available (0)	The Microsoft Visual Studio Redistributables need to be installed. A copy of these will be found in the **redistributables** folder under the

com.ugs.geolus.search.dataExtr.triangle.JTImporter3. getUnit(Ljava/lang/String;Z)S(0)	*GEOLUS_HOME* folder if they have been selected at installation. Install all versions, then restart the Teamcenter Rich Client.
Access to Shape Search denied to user (0)	The Teamcenter Single Sign On token cannot be authenticated. Contact the Teamcenter administrator.
The specified Geolus server is not compatible with Shape Search.	The Geolus database associated with the specified Geolus server does not contain the right attribute names for use with the Geolus Teamcenter Integration. The Geolus administrator should refer to the *Geolus Teamcenter Indexer* manual to ensure the database is appropriately synchronized with the Teamcenter database.
The Geolus Teamcenter Integration plug-in does not appear in the rich client.	If, after installation, the Geolus icon does not appear in the **Send To** menu, it may be necessary to rebuild the cache of plug-ins recognized by the rich client. To achieve this, close the rich client and run the **genregxml** utility located in the *portal/registry* folder within the rich client installation folder.
Shape searches are very slow and the Teamcenter log file contains the message: Cannot download result thumbnail	Check access to thumbnails. Network or configuration issues with thumbnails can impact search performance. Contact the Geolus administrator for thumbnail configuration problems.
The **Send To** action fails with the message: Could not create the view: Plug-in "com.teamcenter.rac.geolus.results" was unable to instantiate class "com.teamcenter.rac.geolus.results.views.ResultsView" And there are messages in the error details that include the string: java.lang.NoClassDefFoundError: org/apache/axis2/AxisFault	Unregister and reregister the plugin. 1. Close the rich client. 2. Delete all Teamcenter Integration plugin files from the **plugins** folder (including the expanded directory). 3. Run the **genregxml** utility in the **registry** folder inside the rich client installation location. 4. Copy the new Teamcenter Integration files and folders into the **plugins** folder.

This may be associated with an upgrade of Geolus software.	5. Run the **genregxml** utility once more.

Using Geolus shape search

Search for similar released item revisions

Shape Search functionality is available from any Teamcenter application that displays selectable Item Revisions. As an example, start **My Teamcenter**, and find an Item Revision which contains a JT dataset. Right-click on the Item Revision, choose **Send To** > **Shape Search**.

Presentation of search results

The results of the Shape Search appear as a grid of thumbnails in the **Shape Search Results** view in the current application or perspective.

Each part listed in the new view will be an Item Revision whose shape has some degree of similarity to the query part. The first part in the array is always the query part and has a gray background area. Parts are ordered in decreasing order of similarity from left to right and down the page.

The results displayed depend on the query part, the contents of the Geolus database, and on how the search has been configured. Results will be excluded if the current Teamcenter user does not have read privileges on the Item Revision(s) unless the display of inaccessible parts has been enabled (see **Display of inaccessible parts**).

For each part, the name and size relative to the query part is usually shown. The information displayed is configurable.

Examine individual items in search results

Parts can be selected from the array: the selected part has a blue border. When in the **My Teamcenter** application, the **Summary**, **Details** and **Viewer** views relate to the current selected part.

Right-click to see the available options.

Option	Action
Send to Shape Search	Initiates a new Shape Search using the selected part
Send to Lifecycle Viewer	Opens the selected part in the Lifecycle Viewer (the Shape Search Results view remains open)

Send to **My Teamcenter**	Opens the selected part in **My Teamcenter** (the Shape Search Results view remains open).
Copy	Allows copy and paste functionality into Teamcenter applications.
Properties	Opens the Teamcenter properties dialog to show the Item Revision.

Display a large set of results

Larger sets of Shape Search results are usually divided into pages. A typical page might consist of 30 items (the exact number that are displayed on each page is configured). Up and down arrows allow navigation through several pages, and you can display all the results on one page.

Before displaying all the search results, check the total number of results indicated at the top of the Shape Search Results view. It can take an appreciable length of time to render the thumbnails if there are more than a hundred results to display at once.

Modify search criteria

Modify the current search by selecting **Modify Search**. The Shape Match control displayed in this dialog box and in **Preferences** can be changed by the Teamcenter administrator. See **Configuring shape match control**.

The relative minimum and maximum sizes are used to filter the search results to include only those parts whose size relative to the query part is within the given bounds.

A common configuration gives a list of three choices:

Shape Match Setting	Returned Results
Identical	Return parts that Geolus considers identical
Very Similar	Return parts that Geolus considers very similar
Similar	Return parts that Geolus considers similar

Ask your Geolus Server administrator to explain other terms that appear in this list.

If the Geolus administrator makes changes to the server configuration that affect the shape match options, these changes will not be available until the rich client is restarted.

A slider can be used to control shape match. The left-most position of the slider corresponds to Identical. Dragging the slider to the right will increase the number of results obtained from a search. The right-most position corresponds to all similar results.

The **Save Options** section controls if and how the modified search is saved for later use.

Option	Action
Add New Search	Save the modified search as a new search, in addition to the original
Replace Current Search	Replace the search which is being modified with the modified version
Do Not Save Modified Search	Do not save the modified version of the search

Information about saving searches is in **accessing previous searches**.

Attribute Filters allows search results to be refined by applying alphanumeric filters on the generic attributes stored in Geolus. The filter labels will depend on the Geolus database and may vary in number.

A search can be modified so that only parts where the attribute **ITEM NAME** starts with `04_s` will be returned.

The characters ***** and **?** are wild card characters, matching any number of characters and any single character respectively.

If the Geolus Server administrator connects to a different database or makes a change to the database which affects the attributes stored in it, the rich client must be restarted. This is a very rare circumstance. If the rich client is not restarted, then the result of any search that includes attribute filters will be unpredictable.

Modify the default search

To modify the default search, select the **Preferences** option.

- The **Search** preferences control the minimum and maximum size filters and shape match parameter used for all subsequent searches. Shape match is set by a slider control but may be replaced by a drop-down.

- The **Results** preferences control what is displayed with each part, how many results are displayed per page (default 30), and how many results are saved with persisted searches (default 30).

Display inaccessible parts

The Teamcenter administrator can enable this functionality which is disabled by default. If this functionality has been enabled, Item Revisions that are not directly accessible by the currently logged in Teamcenter user will be shown in the results of searches. Depending on the configuration, more or less information about these results can be seen, enabling suitable candidate results to be explored.

Configure this functionality by **configuring the display of inaccessible results**. Many of the preferences affect the appearance.

All accessible results are shown with normal thumbnails, the Item/Revision name and the size match information. If objects are inaccessible, they display with a padlock image (this image can be changed), and are slightly faded (the amount of fading can also be changed).When an inaccessible result is selected, the thumbnail will be shown without any fading.

The accessible and inaccessible search results appear in the same order as they would if the same search were performed through the Geolus web interface.

To show more information about an inaccessible result, hover over the thumbnail to show a tool-tip. The text can be configured by the Teamcenter administrator. To copy this text, right-click on the result and select **Copy accessibility text to clipboard**.

Note the rest of the menu entries are disabled.

Saved search queries

Every time a Shape Search is run, the search query parameters are saved for later use, including between Teamcenter sessions. This means that running a saved search against a Geolus server that has changed (in configuration or content) may produce different search results.

Access previous searches

You can see a list of the saved searches in the **Shape Search Results** view. This displays a list of the most recent searches, with the option of showing all saved searches by clicking on **More Open Shape Search Results**.

Hover over a search in the list to display the search parameters.

If a search is saved and the maximum number of saved searches has been reached, the oldest search will be deleted to make space. The maximum number of saved searches can be configured using **Preferences** as explained in **modifying the default search**.

The **More Open Shape Search Results** option opens the **Previous Shape Searches** dialog, allowing the entire list of saved searches to be accessed. Individual searches or the entire list can be deleted. Searches can run again using **Open**.

Selecting the option **Clear All Shape Search Results** removes all saved searches.

Run previous searches

Searches may be run again either by clicking on the entry in the list or by clicking **Open** in the dialog box. If parts were added or removed from the Geolus server since the search was originally created, the new results will reflect this.

In rare situations, a saved search may refer to shape match values that do not correspond to the current Geolus server configuration. A dialog prompts for an alternative shape match value selected from those currently available. Choosing the correct shape match value depends on the type of search required. Contact the Geolus administrator for guidance.

If the Geolus administrator connects it to a different database or makes a change to the database which affects the attributes stored in it, this will affect any saved search which refers to attribute filters. If this happens, restart the rich client and clear the list of saved searches.