Authors:
BOPF Team

Technical documentation

BOPF Enabling for Enterprise Search

Document Version	Status	Date

Template Version: AC 2.2

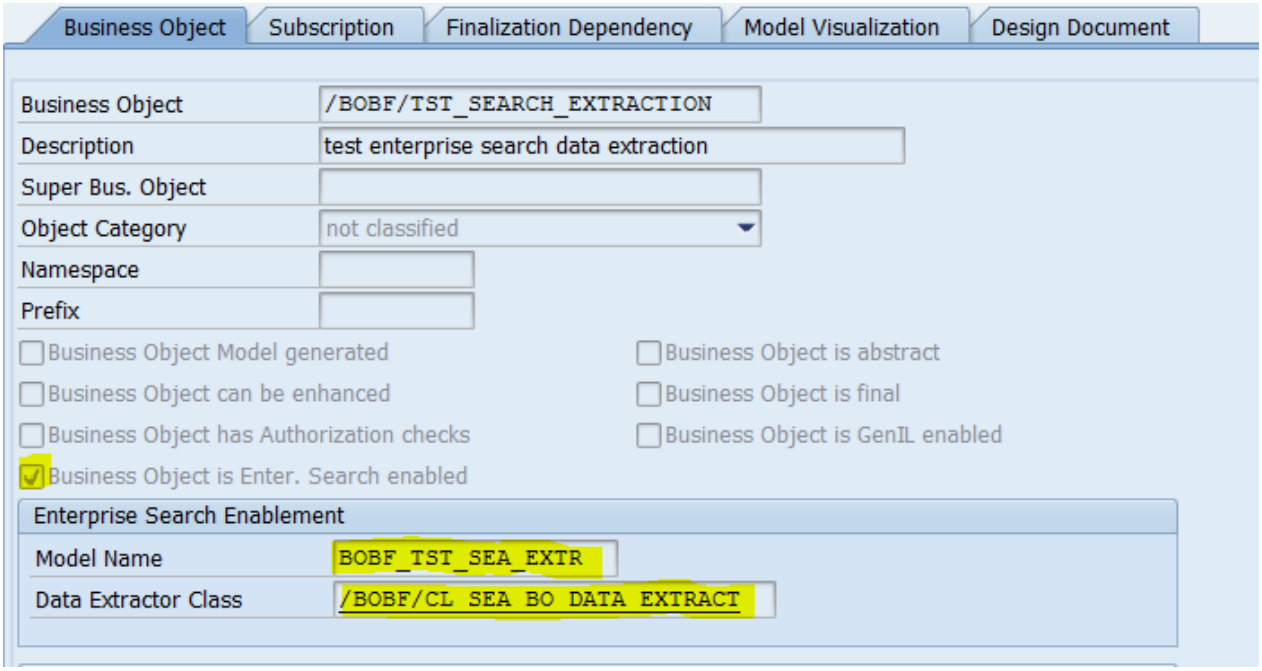
Contents

1 Overview

1.1 Design time

To enable a business object for enterprise search, the following actions have to be performed:

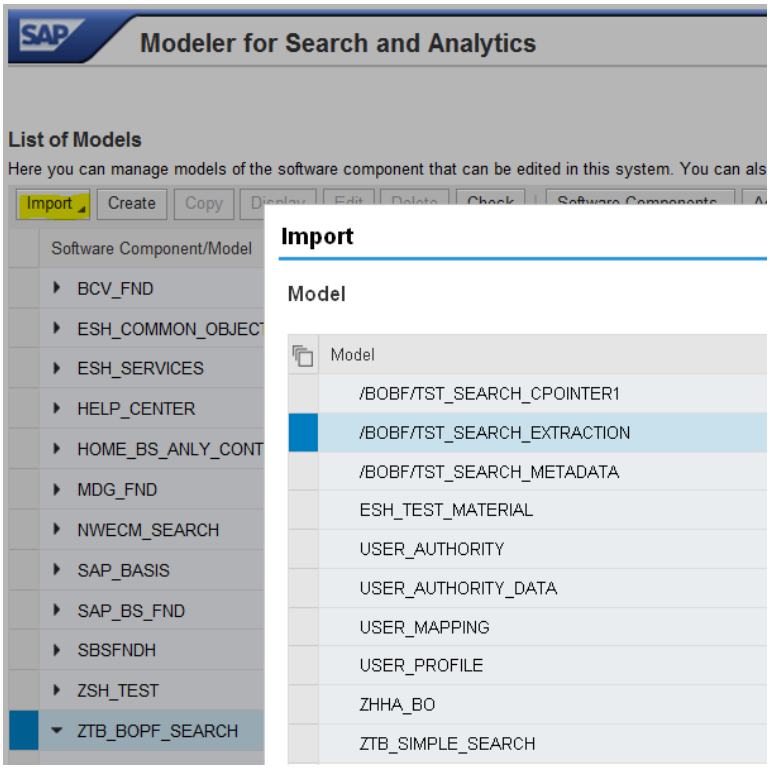
1.1.1 Enable BO for enterprise search in BOPF design time



Flag the Business Object as Enterprise Search enabled. Enter the model name and the name of the extraction class.

1.1.2 Create the enterprise search model from the BOPF BO model

Import the BO model in transaction ESH_MODELER as shown below.



The import transfers the persistency view of the BO model into the enterprise search model. That means, the database tables are used as a DDIC reference. As the BOPF BO model doesn't contain search specific metadata, the model has to be adjusted in the Enterprise Search modeler. When re-importing the data from BOPF, changes in the enterprise search model are overwritten.

1.2 Data extraction

Data extraction is needed to replicate the data to TREX. For that reason, interface /BOBF/IF_SEA_BO_DATA_EXTRACT needs to be implemented. The default implementation is /BOBF/CL_SEA_BO_DATA_EXTRACT, which extracts the persistency view of the BO model and fills corresponding nodes and fields of the enterprise search model. Data extraction can be separated into the following parts:

1.2.1 Full indexing

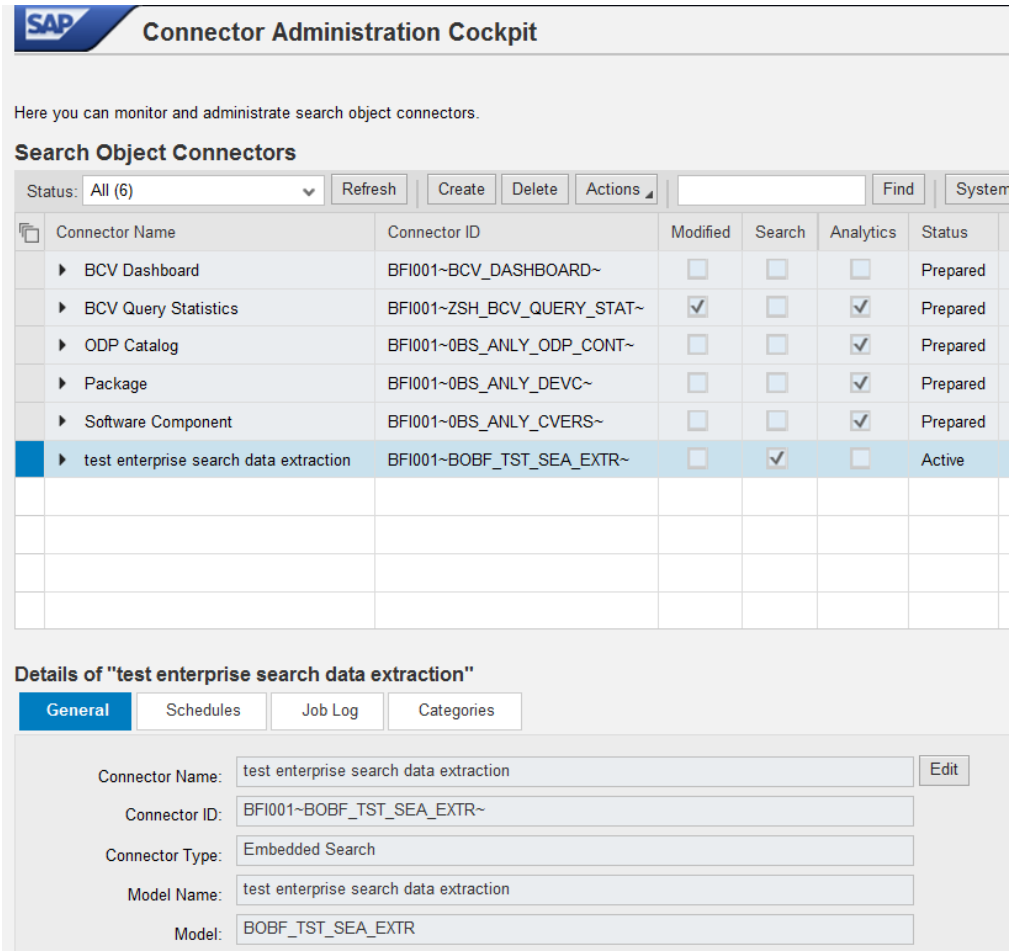
For an initial load of the TREX, the full indexing is needed. This step doesn't need to be repeated at a later point in time, as long as there are no inconsistencies (e.g. caused by a wrong implementation of the extraction class).

1.2.2 Delta indexing

After full indexing is completed, delta indexing is required only to keep the data on TREX up to date. BOPF notifies enterprise search automatically about BO changes, as soon as BO data is newly persisted. It is therefore important that the enterprise search model name is maintained in the BOPF model, even if the enterprise search model is not created via the import function.

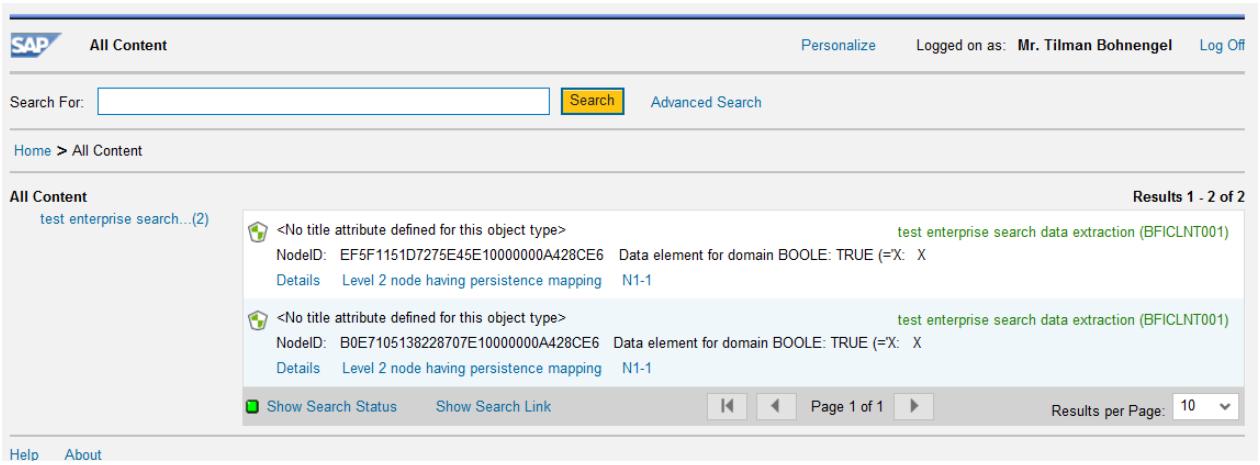
1.2.3 Administration

In the Enterprise Search administration cockpit, a connector has to be created for the enterprise search model. Afterwards, the connector has to be scheduled for indexing.

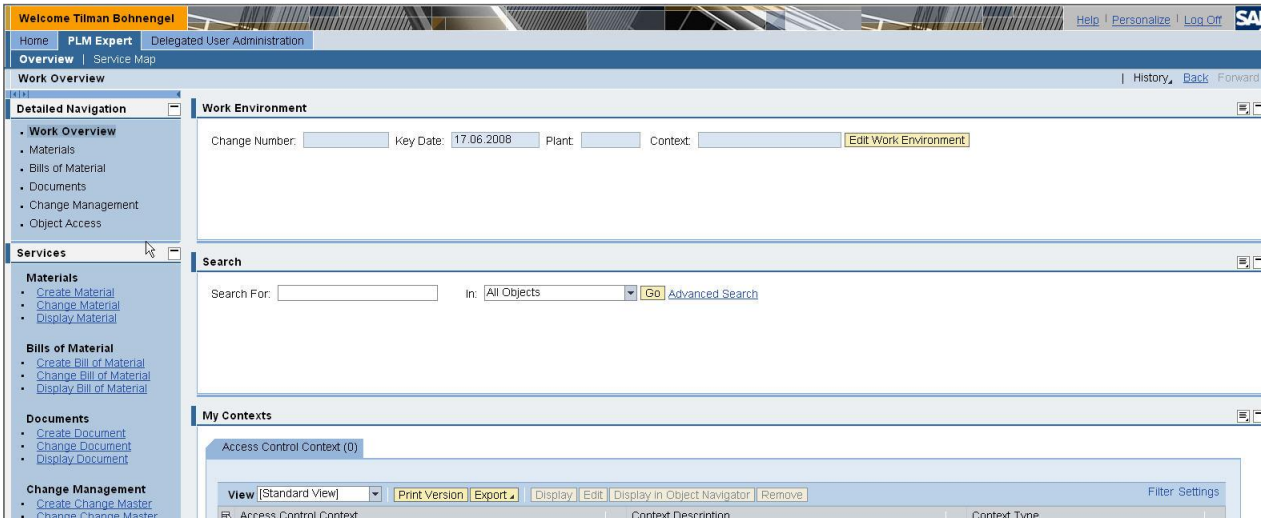


1.3 Search

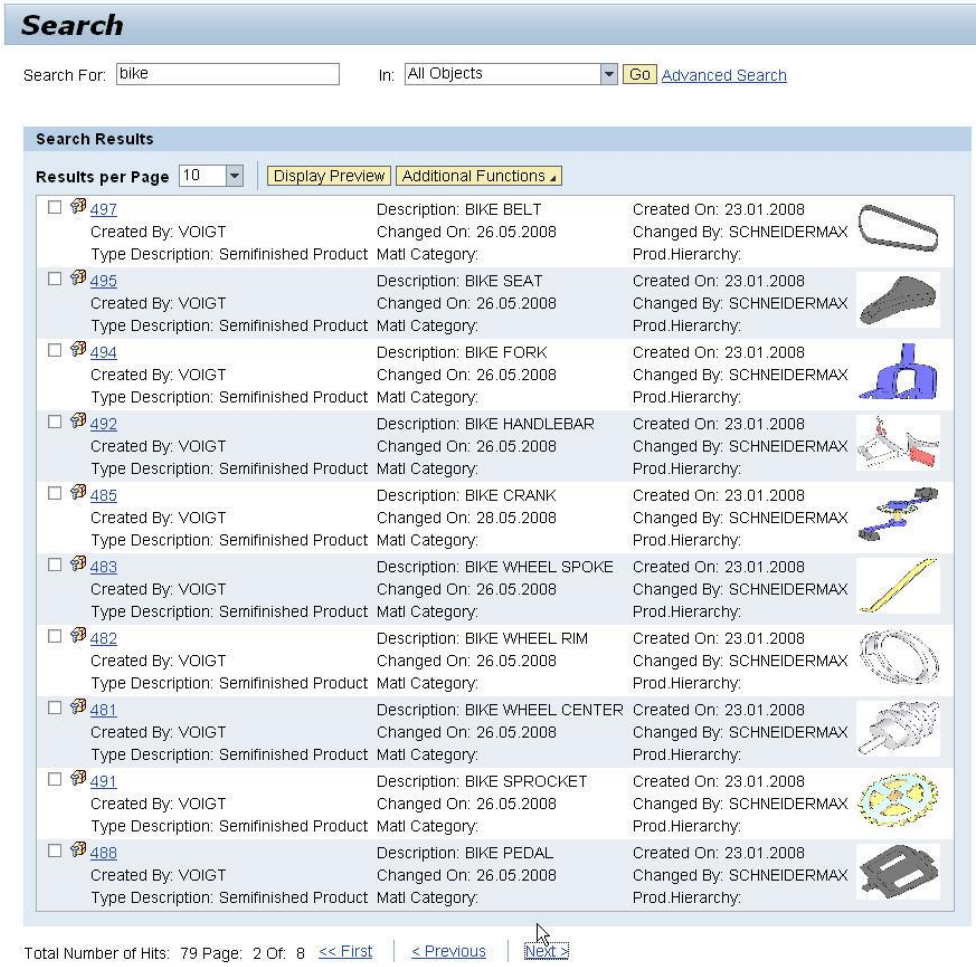
The BO is searchable, if its connector has the status Active. Enterprise Search delivers several generic UIs, which can be used to search the BO. For the local system search (embedded search), a web dynpro application exists that can be embedded into other web dynpro applciations. It can also be started with transaction ESH_SEARCH.



It is also possible to use the APIs provided by enterprise search to seamlessly embed search into the applications. The figures below show embedding of enterprise search into PLM solution with an application specific UI.



Full text search across all relevant PLM Objects embedded into PLM control center



Result list as a modal dialog window

2 Design

All enterprise search relevant objects can be found in package /BOBF/SEARCH.

2.1 Data provider

The data provider /BOBF/CL_SEA_DATA_PROVIDER represents the central link between enterprise search and BOPF. It is called by enterprise search for design time actions (import function in the modeler) as well as for the extraction of data. It works also as a factory for:

- The object type provider
- The Enterprise Search metadata descriptor
- The BO data extractor

2.2 Metadata descriptor

The abstract metadata descriptor /BOBF/CL_SEA_BO_METADATA contains the general mapping rules between the BOPF BO and the Enterprise Search Model. Besides that, the class /BOBF/CL_SEA_BO_METADATA_PERS, which inherits from that, adds the details, which are specific to the persistency view of the BO model. These are

- The data base table on the node as DDIC reference
- The association binding of compositions

2.3 Writer of change pointers

Writing of change pointers is necessary for delta indexing in order to keep the TREX indexes up to date.

The class /BOBF/CL_SEA_CHANGE_POINTER notifies changes during the save phase to enterprise search. It is realized as a plugin of the transaction manager. In case of synchronous update, the plugin directly calls method CL_ESH_IF_DATA_PROVIDER_TOOLS=>SET_CHANGEPOINTERS. In case of asynchronous update, the function module /BOBF/SEA_SAVE_CPOINTERS is called in update task in order to do that job. Changes are aggregated on BO level, as data extraction is always executed for complete BOs. This ensures consistency at least within the BO. Change pointers are only written for BOs, which are marked as enterprise search enabled.

2.4 BO data extractor

The BO data extractor /BOBF/CL_SEA_BO_DATA_EXTRACT is implemented to generically extract the persistency view of a business object. That means, the enterprise search model must also reflect the persistency view, if this extractor should be used. The class is the default implementation of interface /BOBF/IF_SEA_BO_DATA_EXTRACT. Another class can be defined in the BO configuration. Full indexing is done in packages. One call of method ESH_IF_OBJECT_ITERATOR~NEXT extracts one package during the full indexing, whereas IF_ESH_IF_OBJECT_DATA_BY_ID~GET_DATA retrieves the BO data for a list of keys during the delta indexing.

3 Further hints and sources of information

3.1 Important transactions

ESH_MODELER	Design time
ESH_COCKPIT	Connector Administration
ESH_SEARCH	Generic search UI
TREX_ADMIN	Trex administration

3.2 Wikis

Enterprise Search @ Business Suite:

https://wiki.wdf.sap.corp/wiki/display/BSExcellence/Enterprise+Search

NW Enterprise Search

https://wiki.wdf.sap.corp/wiki/display/EPSEARCH/Home

TREX

http://trexweb.wdf.sap.corp:1080/wiki/index.php/Main_Page

