

OLE Location Technical Document

Table of Contents

[Purpose](#)

[Dependencies \(db tables\)](#)

[Logical Data Model \(Class Structure\)](#)

[Physical Data Model \(Database Schema\)](#)

[Service Interface Design \(Java\)](#)

[Business Rules Class](#)

[Service Interface Design \(REST if applicable\)](#)

[User Interface Design](#)

[Data Importing](#)

[Data Exporting \(if applicable\)](#)

[Workflow](#)

[System Parameters](#)

[Roles and Permissions](#)

Purpose

The Location eDoc is a simple KRAD maintenance document with provisions to edit and create existing and new location record, respectively. The location record in OLE is designed such that the physical location can be represented here electronically and to achieve this the location level is introduced. Location levels help in modeling relationships existing between parts of the institution.

Location information is part of items or holdings information helping patrons to identify their exact physical presence in the library. It is also linked to the circulation desks in a library based on which circulation policies are defined.

Dependencies (db tables)

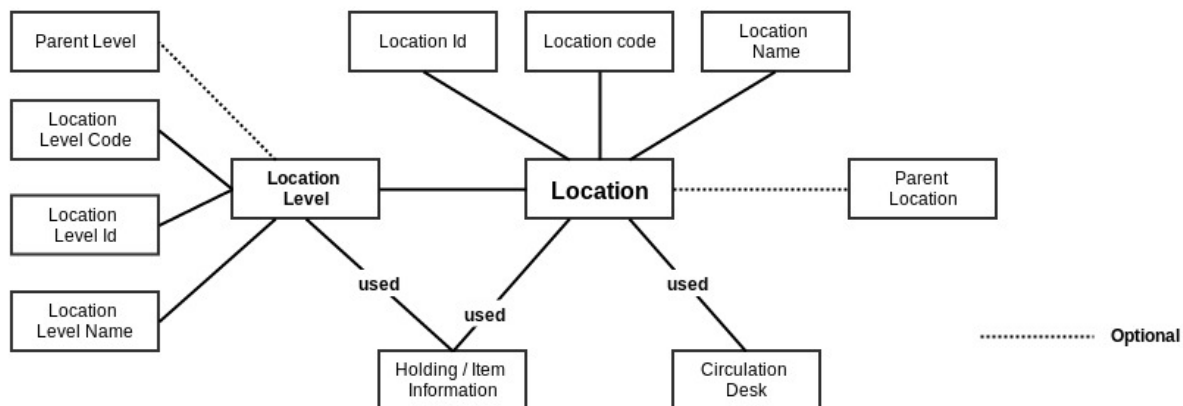
Tables used in Location creation

ole_locn_level_t	OLE Location Level
ole_locn_t	OLE Location

Tables where Location is used/referenced

ole_ds_item_t	OLE Item information
ole_ds_holdings_t	OLE Holdings information
ole_crcl_dsk_locn_t	Circulation Desk Location

Logical Data Model (Class Structure)



Location is made up of the location id, code, name, level and parent. The parent can be null for locations with no parent. Typically, Institution level locations don't have a parent. The location level

is in turn made up of id, code, name and a parent level. The top most level, typically institution, has the parent level field blank. OLE by default comes with five pre-defined location levels viz. Institution, Campus, Library, Collection, Shelving Location. The shelving location is where the physical items are shelved. Though there are five location levels available, the institution, depending on their need, can use only those that are required.

For example, consider a university ABC having 3 campuses namely camp1, camp2 and camp3. Camp1 has two libraries - lib1 and lib2. Lib1 houses books on Computer Science and Astronomy with each topic having five shelves, totalling 10, named with alphabets A to J. In this case it makes sense to use all the available location levels

Institution	-	ABC
Campus	-	Camp1
Library	-	lib1
Collection	-	CompSci
Shelving	-	B

Consider another example, university XYZ having only one campus (camp1) with one library housing books on all subjects. Here using all location levels is unnecessary

Institution	-	XYZ
Campus	-	camp1
Library	-	<Not needed in this scenario>
Collection	-	<Not needed in this scenario>
Shelving	-	D

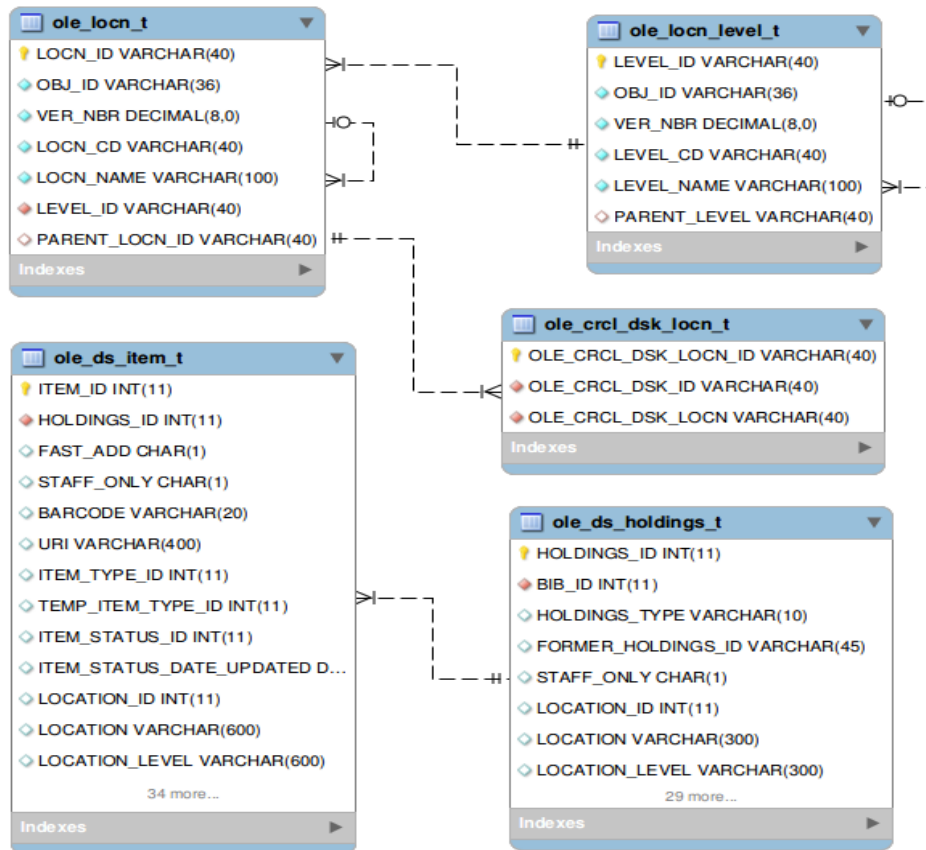
NOTE:

- The Location Level Names are not hard coded and are free to be edited. Location Level Code, SHELIVING, is hard coded and hence changing it might throw errors. Other Location Level Codes can be changed, albeit, through the back end. There is no provision to edit the Location Level Codes from the edit screen in the application.
- There are five location levels by default and a new location level cannot be created through the application. However, from the back end new location levels can be added as modifications to the table data.
- Though multiple location levels can be added (only through back end), retrieved and edited, it will not serve any purpose as the Location must have a Level Id as 5 or less, 5 corresponding to SHELIVING. Any level above 5 though can exist will not show up in the list of locations and hence cannot be used.

The location information is part of Holding or Item information. This aids the user to search copies based on their locations. Locations are also linked to circulation desks to help in circulation policy.

For example, consider an university ABC with campus, CAMP1 and Library, BLOCK1. The Library houses various Shelves, A, B, C, D and E. The library can choose to have policies under a single Library (BLOCK1) or decide to have independent policies for the different shelves (A, C and E). The policies under the parent location trickles down to its child locations. So a policy for BLOCK1 applies to shelves A, B, C, D and E unless they have different policies defined. A more detailed coverage on how locations are used in circulation policy will be done in the OLE KRMS (Kuali Rules Management System) documentation.

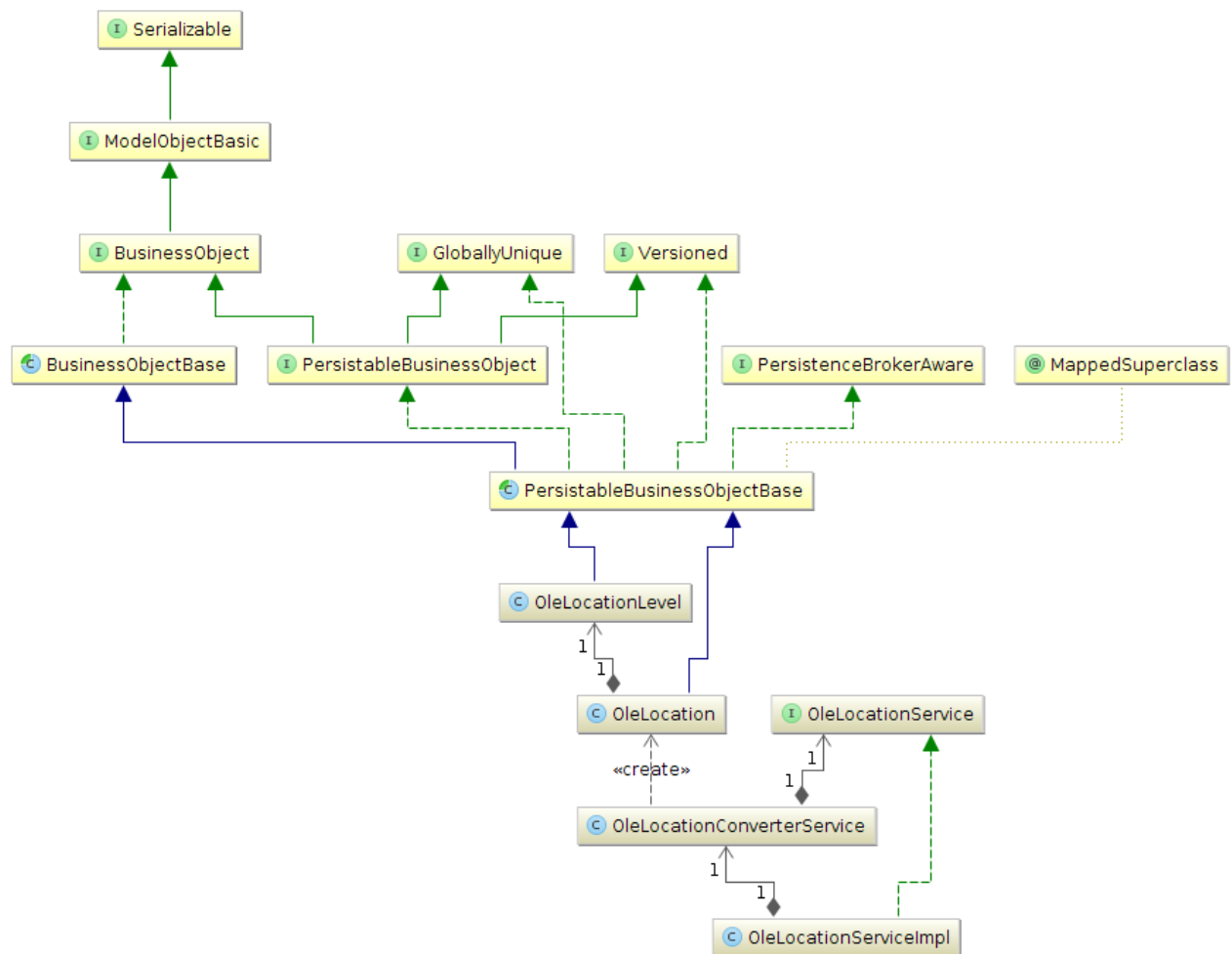
Physical Data Model (Database Schema)



The location data in OLE gets into the *ole_locn_t* table and the location level data into the *ole_locn_level_t* table. Both *ole_locn_t* and *ole_locn_level_t* tables are also self referenced for the PARENT_LOCN_ID and PARENT_LEVEL respectively. The *ole_crcl_dsk_locn_t* links the circulation desk (OLE_CRCL_DSK_ID) with location (OLE_CRCL_DSK_LOCN) with a unique circulation desk location id as primary key. Location is also part of the item and holding record information. Both item record in *ole_ds_item_t* table and holdings record in *ole_ds_holdings_t* table hold location (LOCATION_ID, LOCATION) and location level (LOCATION_LEVEL) data.

The tables *ole_locn_t* and *ole_locn_level_t* contains two properties by default – Version Number and Object Id. This is in order to take advantage of KRAD features. More information can be found in Rice Documentation [here](#).

Service Interface Design (Java)



Powered by yFiles

Maintenance documents are used for creating and modifying Location and Location Level information. The *OleLocationServiceImpl* class uses the *OleLocation* and *OleLocationLevel* business objects to create and update location and location level respectively.

Business Rules Class

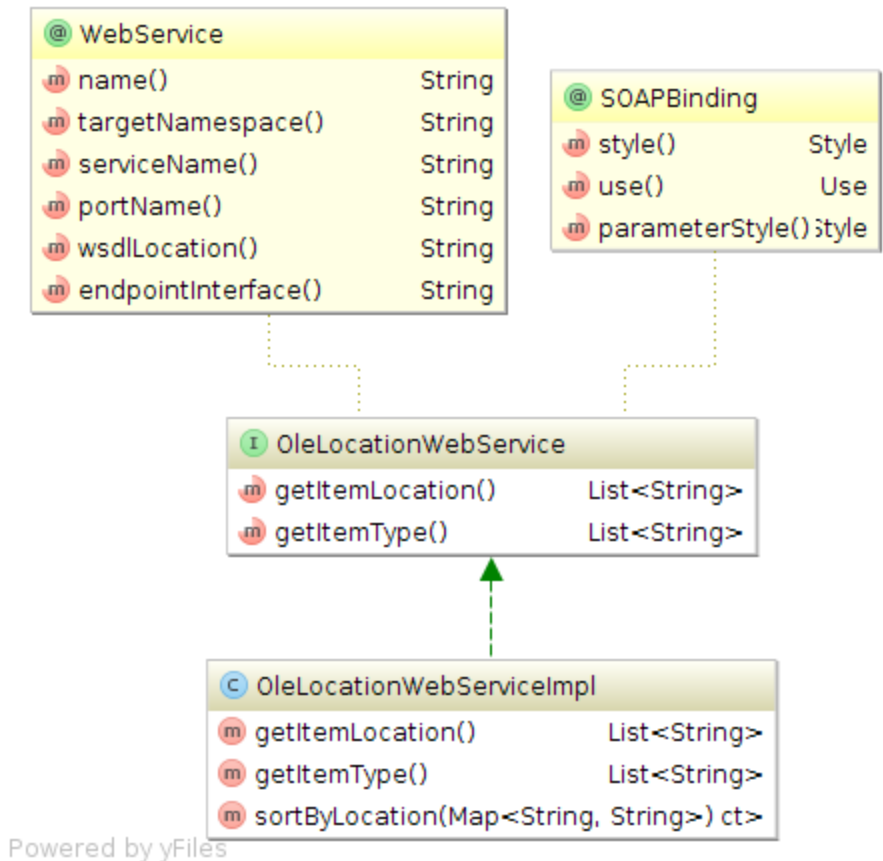
The *OleLocationRule* class extends the *MaintenanceDocumentRuleBase* class of Rice to validate data in the the *Ole location* eDoc.

For more information, Javadocs can be found [here](#).

Service Interface Design (REST if applicable)

OLE doesn't support creation or modification of Location data or Location Level data through REST APIs. However, they have methods to return list of item location and item types.

The WSDL link is <Location URL>/remoting/oleLocationWebService?wsdl (say in case of staging, it is, <http://staging.ole.kuali.org/remoting/oleLocationWebService?wsdl>)



User Interface Design

The OLE Location document uses KRAD's UIF (User Interface Framework). A very good guide on this can be found [here](#). The Spring Beans XML used in the case of Location is the *OleLocationDocument.xml* and in the case of Location Level is the *OleLocationLevelDocument.xml* under the *datadictionary* folder.

Data Importing

Location data can be imported through Batch Processes. A more detailed document on Batch Process can be found [here](#).

Data Exporting (if applicable)

OLE uses a RDBMS backend and hence any data can be exported using simple SQL queries. Currently there is no direct way in which Locations can be exported directly from OLE.

Workflow

Location document doesn't have any workflow defined by default. However, if needed it can be defined at *OlePatronDocType.xml* file. The document type name is OLE_LLMD for OLE Location Level and OLE_LMD for OLE Location.

System Parameters

No System Parameters are currently defined related to Location or Location Level.

Roles and Permissions

Permissions are linked to roles which are in turn linked to Users to give them access to screens and functions.

Permission ID	Permission Name
KRLLMD1	Initiate Location Level Maintenance Document
KRLMD1	Edit Location Maintenance Document
KRLMD2	Edit Location Maintenance Document for Shelving Location Administrator
OLE10133	Edit Location Level Name Field
OLE10134	View Location Level

Role ID	Role Name	Permissions
OLE10049	Location Administrator	OLE10133, OLE10134, KRLMD1, KRLLMD1
OLE10050	Shelving Location Administrator	OLE10134, KRLMD1