

# PO Bulk Amend Technical Document

## Table of Contents

- [Purpose](#)
- [Dependencies \(DB tables\)](#)
- [Logical Data Model](#)
- [Physical Data Model](#)
- [Service Interface Design \(Java\)](#)
- [Service Interface Design \(REST if applicable\)](#)
- [User Interface Design](#)
- [Data Importing](#)
- [Data Exporting \(if applicable\)](#)
- [Workflow](#)
- [System Parameters](#)
- [Roles and Permissions](#)

## Purpose

OLE allows for bulk editing of Purchase Order (PO) documents. The Purchase Order Bulk Amendment document in OLE (Select/Acquire -> Others -> Purchase Order Bulk Amendment) allows for editing of PO data. It achieves this through exposing data to be edited through a CSV file where the data to be edited can be modified by the users.

This document details the technical details involved in this process.

## Dependencies (DB tables)

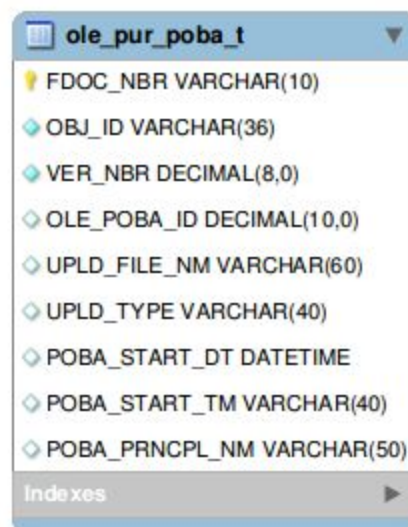
Tables part of the PO bulk amend in OLE

ole_pur_poba_t	OLE Purchasing Order Bulk Amend
----------------	---------------------------------

## Logical Data Model

The PO bulk amend document retrieves the existing PO details and takes in the modified data and replace it with the older data. Hence it doesn't involve any new tables and write only to existing PO data. One table is maintained to record the upload process with values such as the file name, start time, end time, etc. The data is just for logging the upload process details and is not referred to by other tables.

## Physical Data Model

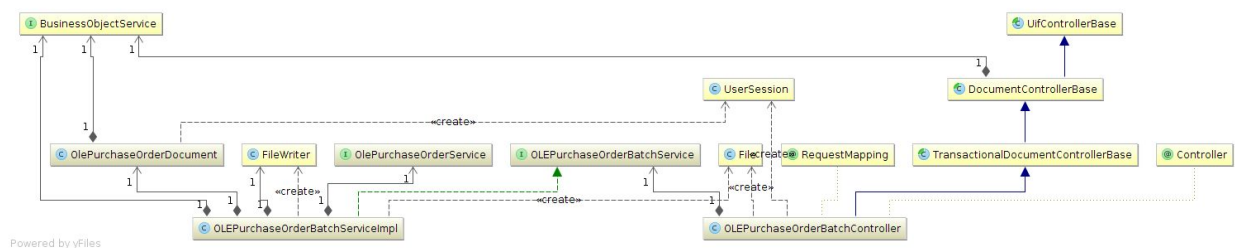


ole_pur_poba_t	
FDOC_NBR	VARCHAR(10)
OBJ_ID	VARCHAR(36)
VER_NBR	DECIMAL(8,0)
OLE_POBA_ID	DECIMAL(10,0)
UPLD_FILE_NM	VARCHAR(60)
UPLD_TYPE	VARCHAR(40)
POBA_START_DT	DATETIME
POBA_START_TM	VARCHAR(40)
POBA_PRNCPL_NM	VARCHAR(50)
Indexes	

The *ole\_pur\_poba\_t* table stores the purchase order bulk amend document data to record the job details. The FDOC\_NBR is the primary key and records the document number. The OLE\_POBA\_ID records an id, UPLD\_FILE\_NM records the file name that has been uploaded and the UPLD\_TYPE records the upload type. The POBA\_START\_DT contains the start date while the POBA\_START\_TM contains the start time. The POBA\_PRNCPL\_NM records the principal name of the user who initiated the bulk amendment document.

All tables with persistable data 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)



The Purchase Order Bulk Amendment (POBA) starts with the uploading of the first file with the PO ids. The file is a simple text file with the Purchase Order (PO) numbers. In cases where multiple POs are to be amended, the file contains the PO numbers one after another (each row has one PO number). The *createFileForPOBA* method of the *OLEPurchaseOrderBatchServiceImpl* class retrieves the PO ids from the text file. This list of PO ids are sent to the *downloadCSV* method of the same class. This method retrieves the PO details from the database for the PO ids and writes them to a CSV (Comma Separated Values) file. The file names generated by the method ends with *'\_PurchaseOrderBulkAmendmentFile.csv'* and will be listed for download under the 'Batch File' screen under Admin.

After the user makes the necessary changes to the CSV file, it is uploaded in the POBA page. The *readFile* method of *OLEPurchaseOrderBatchServiceImpl* class calls the *readCSVAndPreparePOs* method. It uses the CSVParser to parse the data in the file and returns a list of POs. The list is iterated and the PO details retrieved. It is passed to *validatePOFields* method to validate the various incoming modified values of the PO. After validation, the *createPurchaseOrderAmendmentDocument* method is called. In this method, different parts of the PO is updated through individual methods, *updateDocument* method updates the purchase order related data, *updateVendor* method updates the vendor related data in the PO, *updateItem* method updates the item related information in the PO, *updateAccount* method updates the Accounting Line information in the PO.

## **Service Interface Design (REST if applicable)**

The POBA services are not exposed as REST APIs.

## **User Interface Design**

The PO Bulk Amend 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 Patron is the *OLEPurchaseOrderBatchDocument.xml* under the *datadictionary* folder.

## **Data Importing**

The POs that are to be amended can be imported as CSV file in the POBA document.

## **Data Exporting (if applicable)**

OLE uses a RDBMS backend and hence any data can be exported using simple SQL queries.

## **Workflow**

Workflow is not currently available.

## **System Parameters**

System Parameters are not used.

## **Roles and Permissions**

Roles and Permissions are currently not used.