Budget Construction/Adjustment Technical Document

Last Published January 29, 2016

Purpose

Budget Construction and Adjustment helps a library to construct and subsequently adjust the base budget for the coming fiscal year. It is used to designate a base fiscal commitment.

This document will provide an overview about Budget construction and adjustment in OLE, the associated tables, classes and configurations.

Dependencies (db tables)

Tables part of Budget in OLE

gl_acct_balances_t	General Ledger Account Balances
gl_acct_balances_hist_t	General Ledger Account Balances History
gl_balance_t	General Ledger Balance
gl_balance_hist_t	General Ledger Balance History
gl_sf_balances_t	General Ledger Sufficient Fund Balances

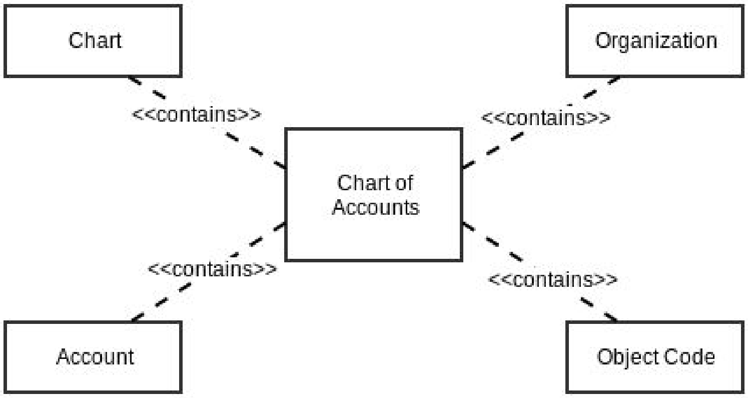
Tables where Budget is used/referenced in OLE

fp_bal_by_cons_mt	Financial Processing Balance by Consolidation
fp_bal_by_level_mt	Financial Processing Balance by Level
fp_bal_by_obj_mt	Financial Processing Balance by Object Code
fp_interim1_cons_mt	Financial Processing interim 1 consolidated
fp_interim1_level_mt	Financial Processing interim 1 level
fp_interim1_obj_mt	Financial Processing interim 1 Object code
fp_interim2_cons_mt	Financial Processing interim 2 consolidated
fp_interim2_level_mt	Financial Processing interim 2 level
fp_interim2_obj_mt	Financial Processing interim 2 Object code

NOTE: Tables ending with '_mt' are not empty tables. They are temporary tables. The abbreviation "MT" means "Move Table".

Logical Data Model (Class Structure)

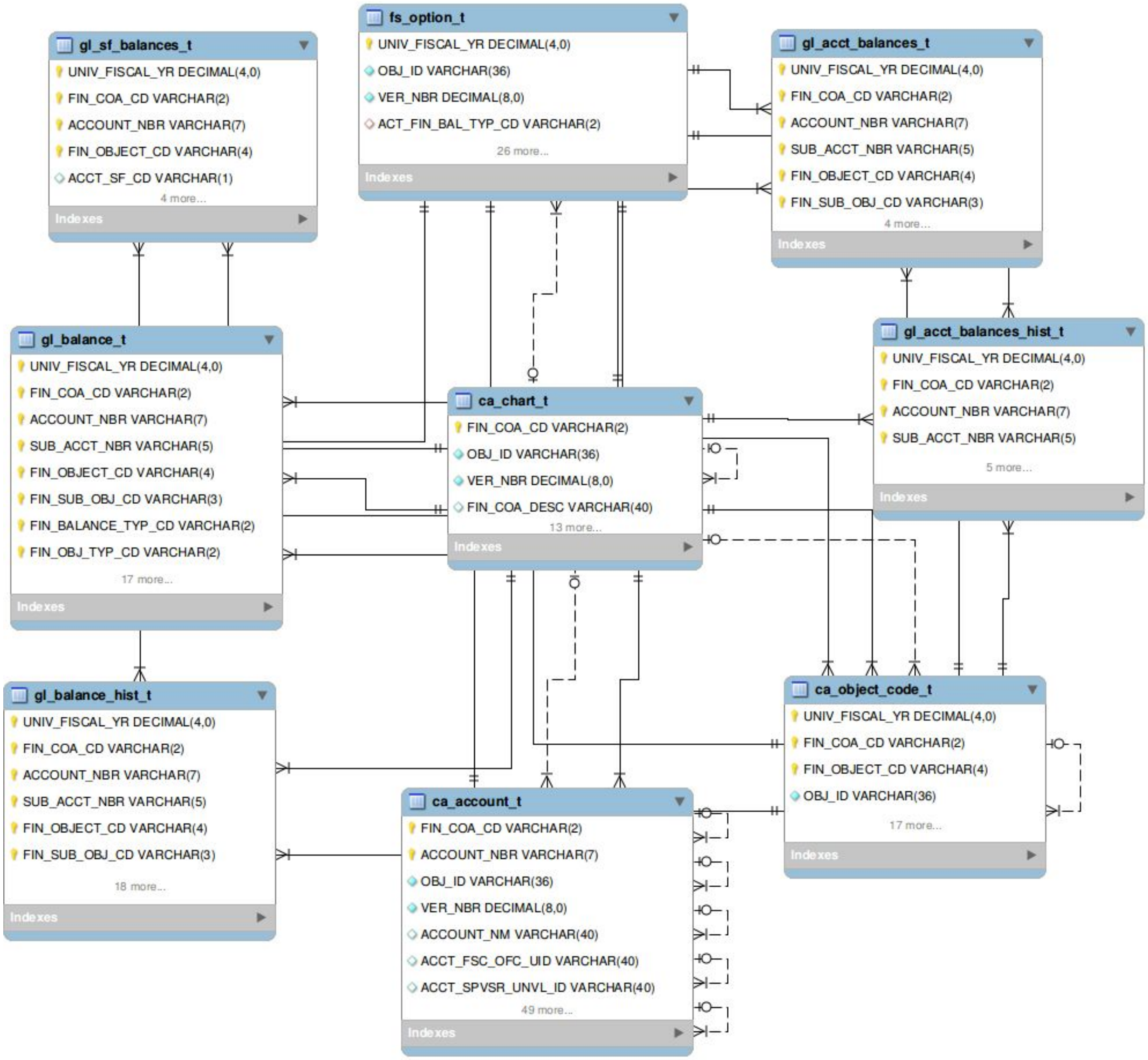
Budget allocation is the process of committing a specific sum of fund for a certain purpose. This is achieved in OLE by allocating the fund to a certain account which has been created to hold funds for a specific purpose. However, Accounts alone cannot record a valid transaction, it will have to include a Chart and Object Code for a specific Fiscal year to record a valid financial transaction. Similarly budgeting also has to be done for the combination rather than for an Account. Budget adjustment is used to move budgeted funds between various combination as the need arises.



Physical Data Model (Database Schema)

The chart related data is held in the chart table, *ca_chart_t.* The chart code, *FIN_COA_CD,* of the *ca_chart_t* table is referenced as a foreign key in all the other main elements of the COA - Account (*ca_account_t)* and Object Code (*ca_object_code_t)*. The *ca_object_code_t* table contains the Object code related data and has a fiscal year (*UNIV_FISCAL_YR)* and chart code referenced as foreign keys. The General Ledger related tables (*gl_sf_balances_t, gl_balance_t, gl_balance_hist_t, gl_acct_balances_t, gl_acct_balances_hist_t*) is where budget related data is stored. And the General Ledger tables refer to *ca_chart_t, ca_account_t, ca_object_code_t* and *fs_option_t* through their primary keys FIN_COA_CD, ACCOUNT_NBR, FIN_OBJECT_CD and UNIV_FISCAL_YR, respectively.

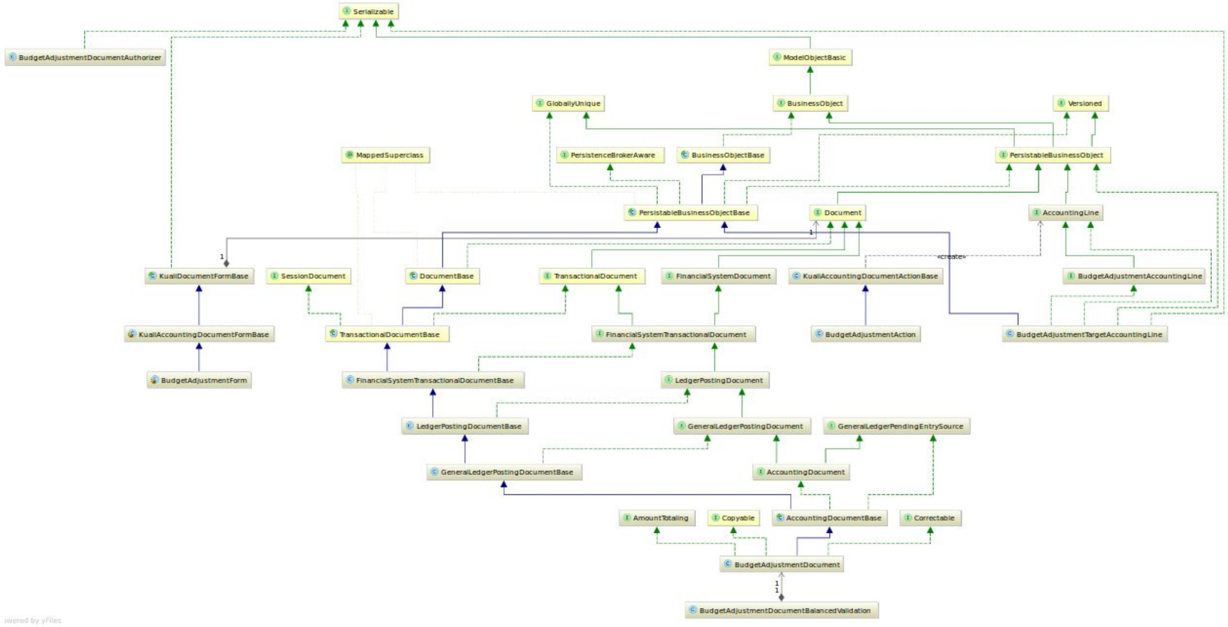
All tables with persistable data contains two properties by default – Version Number and Object Id. This is in order to take advantage of KRAD features. More information can be found in Rice Documentation here.



Service Interface Design (Java)

The Budget part in OLE is inherited from KFS. However, considering that the Budget need not be as exhaustive for a library as that of a financial system, the Budget construction part is kept aside. Hence, the budget figures for the very first time needs to be loaded as part of the backend data load. Budget Construction doesn’t get a separate document in OLE. Budget Adjustment, however, is available to move funds between various Accounts to satisfy needs as it arises over a fiscal year.

The Budget Adjustment document is initiated from the *BudgetAdjustmentAction* class. The business object class is *BudgetAdjustmentDocument* where some validations are also carried out. The *BudgetAdjustmentForm* class takes care of overrides needed for the document. Other validations are carried out by other classes such as *BudgetAdjustmentDocumentPreRules* and *BudgetAdjustmentDocumentBalancedValidation*.


Serializable
BudgetAdjustmentDocumentAuthorizer
ModelObjectBasic
GloballyUnique
BusinessObject
Versioned
MappedSuperclass
PersistenceBrokerAware
BusinessObjectBase
PersistableBusinessObject
PersistableBusinessObjectBase
Document
AccountingLine
1
1
«create»
KualiDocumentFormBase
SessionDocument
DocumentBase
TransactionalDocument
FinancialSystemDocument
KualiAccountingDocumentActionBase
BudgetAdjustmentAccountingLine
KualiAccountingDocumentFormBase
TransactionalDocumentBase
FinancialSystemTransactionalDocument
BudgetAdjustmentAction
BudgetAdjustmentTargetAccountingLine
BudgetAdjustmentForm
FinancialSystemTransactionalDocumentBase
LedgerPostingDocument
LedgerPostingDocumentBase
GeneralLedgerPostingDocument
GeneralLedgerPendingEntrySource
GeneralLedgerPostingDocumentBase
AccountingDocument
AmountTotaling
Copyable
AccountingDocumentBase
Correctable
BudgetAdjustmentDocument
1
1
BudgetAdjustmentDocumentBalancedValidation

OLE also inherits the COA structure from KFS. The COA data comprising, Chart, Account, Organization and Object Code are loaded into OLE through maintenance documents. Maintenance documents are primarily used for creating and maintaining data in OLE.

Chart, *Account, Organization* and *ObjectCode* are business object classes extending the *PersistableBusinessObject* of the Kuali framework, Rice. Each element has their own maintainable implementation class which extends the *FinancialSystemMaintainable* class which in turn extends the *KualiMaintainableImpl* class. The validation, loading and saving of the data is done in the *validateBusinessObjectForSave* and *linkAndSave* methods of the *BusinessObjectServiceImpl* class.

Initializing new budget related data while implementing OLE from scratch will have to done through the back end. There are no available OLE screens for this purpose. More information on the tables involved are available here.

For more information, Javadocs can be found here.

Service Interface Design (SOAP/REST)

No Web services are available currently.

Last Published January 29, 2016

User Interface Design

The Budget adjustment documents uses KRAD’s UIF (User Interface Framework). A very good guide on the framework can be found here.

Data Importing

It is currently not possible to import the data into OLE through any batch processes. However, considering OLE uses a RDBMS backend any data can be inserted using simple SQL queries.

Data Exporting (if applicable)

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

Workflow

Budget Adjustment documents doesn’t have any workflow defined by default. However, if needed it can be defined in files under olefs-> src-> main-> resources-> org.kuali-> workflows.

System Parameters

Budget construction or adjustment documents currently don’t have any associated system parameters in OLE.

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
OLE80164	Initiate Budget Document
OLE80165	Edit Budget Document
OLE80172	Upload Budget

Role ID	Role Name	Permissions
OLE10042	Financial-AQ4	OLE80164, OLE80165
OLE10043	Financial-AQ5	OLE80164, OLE80165, OLE80172
OLE54	User	OLE80164
OLE10000	OLE_User	OLE80164