Commitment Control 101

Julie Thompson
Agenda

• What is Commitment Control?
• Budget Structure
• Project Budgeting
• Basic KK processing
• Control vs Track
• Budget Ref vs Budget Period
• ENCUMB ledger
Commitment Control

• Commitment Control is the PeopleSoft module that enables users to control expenditures actively against predefined, authorized budgets.

• Responsible for budget checking source transactions and updating the corresponding detail ledgers and ledger groups.

• Data posts to the KK ledgers once a valid budget check is received.

• For financial reporting
  • DETAIL_EN and ENCUMB are used to report encumbrance balances
  • ACTUALS is used to report expenditures
Commitment Control

• Pre-encumbrance: Amount the institution is expected to spend, but has no legal obligation. Created when a Requisition is budget checked. Noted by _PR

• Encumbrance: Amount that institution has legal obligation to spend. Created when a Purchase Order is budget checked. Noted by _EN

• Expense: Amount expended. Should mirror amounts in ACTUALS ledger for revenue or expense accounts. Created when a voucher, expense report, or journal is budget checked. Noted by _EX
Budget Structure
Budget Structure

• **Appropriation (APPROP)**
  - Highest level of budgeting
  - Control spending – transactions will not pass budget checking if funds are not available
  - Parent of the ORG Budget
  - Configured to allow Personal Service Transactions (5xxxxx) to pass budget checking even if funds are not available.

• **Organization (ORG)**
  - Department or operating budget
  - Track without budget – transactions will be processed and a warning message logged
  - Child of the APPROP budget
  - Cannot exceed APPROP budget for same chartstring

• **DETAIL**
  - Contains all revenue and expenditure transactions
  - No budget is entered at this level
  - Used for reporting and reconciliation purposes only
  - Captures all chartfield values at level entered
Budget Structure

• **Project Master Expense (PRMST_EXP)**
  • Cumulative project budget
  • Can cross fiscal and budget years
  • Only project id and budget amount are entered
  • Parent of PROJ_GRT

• **Project Grant (PROJ_GRT)**
  • Entered at same level as ORG budget
  • All chartfields are included
  • Child of PRMST_EXP
Budget Structure

• **REVENUE**
  - Track revenues recognized against estimated revenues
  - Georgia*FIRST* makes no distinction between recognized and collected revenue.
  - Recognized and collected balances are updated during budget checking

• **Project Master Revenue (PRMST_REV)**
  - Budget for expected revenue
  - Can cross fiscal and budget years
  - Only project id and budget amount are entered
  - Sibling of PRMST_EXP
Budget Structure

- 700000 level
  - APPROP
  - PRMST_EXP
  - Carries only project id
- 714000 level
  - ORG
  - PROJ_GRT
  - 714000 level
- 714100 level
  - DETAIL
  - Funds 10000, 10500, 10600, 10900, 12xxx, 13000, 14000, 14100, 15000, 16000, 50000
  - Grants Only – determined by Project/Grant flag in set up pages
  - Funds 2xxxx
Budget Rollup

• Commitment Control relies on the hierarchical structure in the account tree to determine how budget lines are set up and where transactions should look for funds.

• Budgets are created and maintained at higher levels than transactions.

• The levels are recorded in the Account Budget Translation Tree.

• Budgetary accounts must appear on this tree (4xxxxx-8xxxxx).
Budget Rollup

• Appropriation Level
  • 400000, 500000, 600000, 700000, and 800000

• Summary Level would be Organization, Project and Revenue
  • Ex. 502000, 714000, etc.

• Detail level is what is entered on Purchase Orders, Vouchers, and Journal lines and is the lowest level at which accounting transactions post.
  • Ex. 502100, 714100 etc.
Rollup Tree
## Budget Rollup

- Example of account roll-up for Fund 10000:

<table>
<thead>
<tr>
<th>Ledger Group</th>
<th>Ledger</th>
<th>Account</th>
<th>Fund</th>
<th>DeptID</th>
<th>Program</th>
<th>Class</th>
<th>Bud Ref</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPROP</td>
<td>APPROP_EN</td>
<td>700000</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>2020</td>
<td>4,620.00</td>
</tr>
<tr>
<td>ORG</td>
<td>ORG_EN</td>
<td>714000</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>2020</td>
<td>2,000.00</td>
</tr>
<tr>
<td>ORG</td>
<td>ORG_EN</td>
<td>715000</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>2020</td>
<td>2,620.00</td>
</tr>
<tr>
<td>DETAIL</td>
<td>DETAIL_EN</td>
<td>714100</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>2020</td>
<td>1,000.00</td>
</tr>
<tr>
<td>DETAIL</td>
<td>DETAIL_EN</td>
<td>714210</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>2020</td>
<td>1,000.00</td>
</tr>
<tr>
<td>DETAIL</td>
<td>DETAIL_EN</td>
<td>715110</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>2020</td>
<td>2,620.00</td>
</tr>
</tbody>
</table>
Budget Rollup

• Example of account roll-up for Restricted Funds:

<table>
<thead>
<tr>
<th>Ledger Group</th>
<th>Ledger</th>
<th>Account</th>
<th>Fund</th>
<th>DeptID</th>
<th>Program</th>
<th>Class</th>
<th>Project ID</th>
<th>Bud Ref</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRMST_EXP</td>
<td>PRMST_EX</td>
<td>714000</td>
<td>20000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>GRT123</td>
<td>2020</td>
<td>4,620.00</td>
</tr>
<tr>
<td>PROJ_GRT</td>
<td>PROJ_GR_EX</td>
<td>714100</td>
<td>20000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>GRT123</td>
<td>2020</td>
<td>2,000.00</td>
</tr>
<tr>
<td>PROJ_GRT</td>
<td>PROJ_GR_EX</td>
<td>715000</td>
<td>20000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>GRT123</td>
<td>2020</td>
<td>2,620.00</td>
</tr>
<tr>
<td>DETAIL</td>
<td>DETAIL_EX</td>
<td>715110</td>
<td>20000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>GRT123</td>
<td>2020</td>
<td>2,620.00</td>
</tr>
<tr>
<td>DETAIL</td>
<td>DETAIL_EX</td>
<td>714210</td>
<td>20000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>GRT123</td>
<td>2020</td>
<td>1,000.00</td>
</tr>
<tr>
<td>DETAIL</td>
<td>DETAIL_EX</td>
<td>714100</td>
<td>20000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>GRT123</td>
<td>2020</td>
<td>1,000.00</td>
</tr>
</tbody>
</table>
Project Budgeting
Project Budgeting

• Projects are special initiatives or other business that need to be tracked with a Project ID, but are not restricted/sponsored funds

• Institutions have two options to handle project budgeting
  • APPROP Budgets will not have to be entered with a Project ID in order for a transaction with a Project ID to pass budget checking at the APPROP level
  • APPROP Budget will have to be entered with a Project ID for transactions with a Project ID to pass budget checking at the APPROP level

• If making a change, the strong recommendation is to make it effective with a Budget Reference and to do it at the beginning of a new fiscal year.
## Project Budgeting

- Example with Project ID as optional on APPROP:

<table>
<thead>
<tr>
<th>Account</th>
<th>Fund Code</th>
<th>Department</th>
<th>Program</th>
<th>Class</th>
<th>Project ID</th>
<th>Budget Ref</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Budget:</strong></td>
<td>700000</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>2020</td>
<td>50,000</td>
</tr>
<tr>
<td><strong>Transactions:</strong></td>
<td>714100</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>2020</td>
<td>22,000</td>
</tr>
<tr>
<td></td>
<td>714100</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>PRJ123</td>
<td>2020</td>
</tr>
<tr>
<td><strong>Remaining Spending Authority</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16,700</td>
</tr>
</tbody>
</table>
## Project Budgeting

- **Budgeting with Project ID on APPROP**

### Transaction with Project ID:

<table>
<thead>
<tr>
<th>Account</th>
<th>Fund Code</th>
<th>Department</th>
<th>Program</th>
<th>Class</th>
<th>Project ID</th>
<th>Budget Ref</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Budget:</td>
<td>700000</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>PRJ123</td>
<td>2020</td>
</tr>
<tr>
<td>Transactions:</td>
<td>714100</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>PRJ123</td>
<td>2020</td>
</tr>
</tbody>
</table>

Remaining Spending Authority 8,700

### Transaction without Project ID:

<table>
<thead>
<tr>
<th>Account</th>
<th>Fund Code</th>
<th>Department</th>
<th>Program</th>
<th>Class</th>
<th>Project ID</th>
<th>Budget Ref</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Budget:</td>
<td>700000</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>2020</td>
<td>30,000</td>
</tr>
<tr>
<td>Transactions:</td>
<td>714100</td>
<td>10000</td>
<td>1010100</td>
<td>11100</td>
<td>11000</td>
<td>2020</td>
<td>22,000</td>
</tr>
</tbody>
</table>

Remaining Spending Authority 8,000
Budget Checking
BUDGET CHECKING

Processing Source Transactions against Control Budgets
Budget Checking

- **KK_SOURCE_HDR**
  - When budget checking is run against a transaction, the budget processor will record the information in the **KK_SOURCE_HDR** table.
  - A unique KK_TRAN_ID is assigned to each transaction header being budget checked.
  - Data will remain in the table unless the transaction is deleted.

<table>
<thead>
<tr>
<th>KK_SOURCE_HDR</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KK_TRAN_ID</strong></td>
<td><strong>KK_TRAN_DT</strong></td>
<td><strong>KK_SOURCE_TRNS</strong></td>
<td><strong>KK_PROCESS_STATUS</strong></td>
<td><strong>KK_PROC_INSTANCE</strong></td>
<td><strong>BUSINESS_UNIT</strong></td>
<td><strong>PO_ID</strong></td>
</tr>
<tr>
<td>0010798861</td>
<td>6/27/2019</td>
<td>PO_POENC</td>
<td>W</td>
<td>12983592</td>
<td>53000</td>
<td>527897</td>
</tr>
</tbody>
</table>
Budget Checking

• **KK_ACTIVITY_LOG**
  • When source transactions are budget checked, the corresponding detail ledgers and ledger groups are updated. The system creates activity lines and stores them in the **KK_ACTIVITY_LOG** table.
  • Lines are identified by a KK_TRAN_ID, KK_TRAN_DT, and KK_TRAN_LN.

• Example: Purchase Order

<table>
<thead>
<tr>
<th>KK_ACTIVITY_LOG</th>
</tr>
</thead>
<tbody>
<tr>
<td>KK_TRAN_ID</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>0010798861</td>
</tr>
<tr>
<td>0010798861</td>
</tr>
<tr>
<td>0010798861</td>
</tr>
</tbody>
</table>
Budget Checking

• **KK_LIQUIDATION** table tracks open balances for document.
• Once a transaction passes budget checking, KK will log a row in this table for each transaction line and ledger group.
• **KK_POSTED_AMT** represents the total source transaction amount.
• **ACTIVITY** represents the activity against total amount.
• **MONETARY_AMOUNT** represents the remaining open balance.

<table>
<thead>
<tr>
<th>KK_TRAN_ID</th>
<th>KK_TRAN_DT</th>
<th>KK_TRAN_LN</th>
<th>LEDGER_GROUP</th>
<th>BASE_CURRENCY</th>
<th>KK_POSTED_AMT</th>
<th>MONETARY_AMOUNT</th>
<th>ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>10798861</td>
<td>6/27/2019</td>
<td>1</td>
<td>APProp</td>
<td>USD</td>
<td>30,308.00</td>
<td>15,661.60</td>
<td>14,646.40</td>
</tr>
<tr>
<td>10798861</td>
<td>6/27/2019</td>
<td>1</td>
<td>DETAIL</td>
<td>USD</td>
<td>30,308.00</td>
<td>15,661.60</td>
<td>14,646.40</td>
</tr>
<tr>
<td>10798861</td>
<td>6/27/2019</td>
<td>1</td>
<td>ORG</td>
<td>USD</td>
<td>30,308.00</td>
<td>15,661.60</td>
<td>14,646.40</td>
</tr>
</tbody>
</table>
Budget Checking

- **KK_TRANS_LOG** stores the budget checking history of a document
- Similar to **KK_ACTIVITY_LOG**
- The SEQNBR is the count for each time the document is budget checked. A new row is inserted for each budget check.

<table>
<thead>
<tr>
<th>KK_TRANS_ID</th>
<th>KK_TRNS_D</th>
<th>KK_TRNS_DN</th>
<th>SEQNBR</th>
<th>DT_TIMESTAMP</th>
<th>CLOSED_V</th>
<th>KK_SOURCE_TRNS</th>
<th>ACCOUNT</th>
<th>DEPTID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010798861</td>
<td>6/27/2019</td>
<td>1</td>
<td>1</td>
<td>6/27/19 12:49 PM</td>
<td>N</td>
<td>PO_POENC</td>
<td>715123</td>
<td>8021958</td>
</tr>
<tr>
<td>0010798861</td>
<td>6/27/2019</td>
<td>1</td>
<td>2</td>
<td>9/16/19 9:39 AM</td>
<td>N</td>
<td>PO_POENC</td>
<td>715123</td>
<td>8021958</td>
</tr>
<tr>
<td>0010798861</td>
<td>6/27/2019</td>
<td>2</td>
<td>2</td>
<td>9/16/19 9:39 AM</td>
<td>N</td>
<td>PO_POENC</td>
<td>715100</td>
<td>5910000</td>
</tr>
</tbody>
</table>
Budget Checking

• **KK_EXCPTN_TBL** stores Commitment Control transaction budget checking exceptions

• **KK_OVERRIDE_TBL** stores the user id that overrode the budget checking for a document.

• Budget Check Exception Status:
  1. Warning: Transaction posted and no further action needed
  2. In Process: Transaction is stuck and will be ignored by process
  3. Error: Transaction failed budget check and must be corrected before it can be posted
## Budget Checking – Liquidation

- **Transaction liquidation example:**

<table>
<thead>
<tr>
<th></th>
<th>PreEncumbrance</th>
<th>Encumbrance</th>
<th>Expense</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requisition</strong></td>
<td>400.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Purchase Order</strong></td>
<td>(400.00)</td>
<td>400.00</td>
<td></td>
</tr>
<tr>
<td><strong>Voucher</strong></td>
<td>(400.00)</td>
<td></td>
<td>400.00</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>400.00</td>
</tr>
</tbody>
</table>
Control versus Track

• Setting a ledger to Control will cause transactions to fail budget checking if the chartfield combination does not have enough spending authority and prevent processing until addressed.
  • Users can either add funds to the budget or override the transaction.

• Setting a ledger to Track with Budget will prevent a transaction from passing budget check if a budget does not exist at all. Requires at least a $0 budget for chartstring.

• Setting a ledger to Track will not prevent a transaction from being processed if a budget doesn’t exist. The process will create a warning and still post.
## Control versus Track

### GeorgiaFIRST Ledger Configuration

<table>
<thead>
<tr>
<th>Budget Type</th>
<th>Track versus Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appropriation</td>
<td>Control</td>
</tr>
<tr>
<td>Organization</td>
<td>Track</td>
</tr>
<tr>
<td>Revenue Estimate</td>
<td>Track</td>
</tr>
<tr>
<td>Project/Grant</td>
<td>Track and/or Control, based on Chartfield definition</td>
</tr>
<tr>
<td>Project Master Expense</td>
<td>Track and/or Control, based on Chartfield definition</td>
</tr>
<tr>
<td>Project Master Revenue</td>
<td>Track</td>
</tr>
<tr>
<td>Detail</td>
<td>Track</td>
</tr>
</tbody>
</table>

- This is a typical setup
- There is some institution discretion for ORG and for the Project ledgers
Budget Period vs. Budget Reference

• What is the difference?
  • **Budget Period** is the PeopleSoft delivered field that is intended to track a budget year. However, due to budgetary rules that are unique to Georgia, the Budget Period field did not always work for our purposes. GaFIRST institutions no longer use Budget Period after Budget Period 2015.

  • **Budget Reference** is also a PS delivered field, but was intended to be an additional indicator of a budget that may not fall into the Budget Period Calendars that are defined, such as multiyear overlapping budgets. Budget Reference works better for us because it is subject to chartfield inheritance, and does not get “reset” by PeopleSoft Financials according to accounting date.
ENCUMB Ledger vs DETAIL_EN

• DETAIL_EN is a PeopleSoft delivered Commitment Control ledger
  • Commitment Control ledgers reside in the LEDGER_KK table
  • Commitment Control ledgers do not have an option to close the ledgers
  • So, no period 0 / beginning balances

• ENCUMB ledger is a reporting ledger that resides in the LEDGER table and is based off of DETAIL_EN
  • There is a nightly process that copies the information from DETAIL_EN to ENCUMB with the same chartfields
  • The process then creates an offset to the 381100, which becomes your Encumbrance Payable Liability
  • The ENCUMB ledger can be ledger closed to create period 0 balances for the next fiscal year.
ENCUMB Ledger vs DETAIL_EN

<table>
<thead>
<tr>
<th>Table</th>
<th>Account</th>
<th>Fund Code</th>
<th>Dept ID</th>
<th>Program</th>
<th>Class</th>
<th>Budget Ref</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL_EN</td>
<td>LEDGER_KK</td>
<td>733100</td>
<td>10000</td>
<td>1057409</td>
<td>11100</td>
<td>11000</td>
<td>2018</td>
</tr>
<tr>
<td>ENCUMB</td>
<td>LEDGER</td>
<td>733100</td>
<td>10000</td>
<td>1057409</td>
<td>11100</td>
<td>11000</td>
<td>2018</td>
</tr>
<tr>
<td>ENCUMB</td>
<td>LEDGER</td>
<td>381100</td>
<td>10000</td>
<td>1057409</td>
<td>11100</td>
<td>11000</td>
<td>2018</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table</th>
<th>Account</th>
<th>Fund Code</th>
<th>Dept ID</th>
<th>Program</th>
<th>Class</th>
<th>Budget Ref</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL_EN</td>
<td>LEDGER_KK</td>
<td>733100</td>
<td>10000</td>
<td>1057409</td>
<td>11100</td>
<td>11000</td>
<td>2018</td>
</tr>
<tr>
<td>ENCUMB</td>
<td>LEDGER</td>
<td>342100</td>
<td>10000</td>
<td>1057409</td>
<td>11100</td>
<td>11000</td>
<td>2018</td>
</tr>
<tr>
<td>ENCUMB</td>
<td>LEDGER</td>
<td>381100</td>
<td>10000</td>
<td>1057409</td>
<td>11100</td>
<td>11000</td>
<td>2018</td>
</tr>
</tbody>
</table>

- There are queries based off of both DETAIL_EN and ENCUMB
- Just be aware of that they are two sides of the same “coin”
Questions?
Please remember to fill out the session survey in the app.