I. WHAT DOES THE OBLIGATIONS MODULE DO?

- This function enables users to enter obligations against Purposes for future periods.

- For example, you may want to obligate $75,000 of a professor’s salary on a grant/contract. Entering a record into this module will result in an encumbrance on that grant/contract’s Purpose for the amount specified.

- These encumbrances exist only in the obligations module, and to see them, the obligations records must be queried. The UOD_BALANCES and UOD_TRANS_DTL records do NOT contain obligations from the obligations module.

- Queries have been written that join the information in UOD_BALANCES with information in the obligations module so that you can see both the remaining cash balance and remaining obligations for a purpose. Those queries are described in this training.

Security issue note: The Obligations function is secured by Purpose:
If you are authorized for ANY Purpose involved in an obligation, you can see the whole obligation (even if it contains rows that obligate against Purposes for which you are not authorized).
However, you can only enter obligations on Speedtypes that point to Purposes for which you are authorized (as viewer, approver or statement recipient).
II. ENTERING OBLIGATIONS

The obligations module, like the labor allocation module, is in the production instance of the UD financial system (also known as FIPRD). To access it, go to www.udel.edu/LAM (which will automatically route you to https://turvy.nss.udel.edu:4480/psp/fiprd/).

From Menu, navigate to UD Obligations:
UD Processes
   UD Obligations

![Image of UD Obligations module]

**Exercise 1: Enter an Obligation**

To enter a new obligation:
2. Enter SetID. It must be UOD01.
3. You will be asked for UD Encumbrance ID. This will be a searchable item – give it a name meaningful to you. For example, if you are encumbering the salary of a faculty member Jones, and your initials are ZQD, you may want to call it ZQDJones01. The UD Encumbrance ID field has only 10 characters. **[We’ve requested that this be increased to at least 50; action pending.]**

   **PLEASE NOTE:** if you choose an Encumbrance ID that has already been used with the same effective date, you will get a message saying that this has already been used and you should choose something else. However, if you choose an Encumbrance ID that has already been used, and a different effective date, you WILL be allowed to add it WITHOUT ANY WARNING MESSAGE. This is important to understand, because there may be multiple...
obligations with the same encumbrance ID, and it is possible that different people created them. This can lead to confusion.

You will then see a screen that looks like this:

Note the box for “Description.” This is another searchable field – you can enter any information in here that you would want to query on later.

Let’s say you wanted to encumber $50,000 of this person’s salary on Purpose FREC110000, with $25K of that for Instructional workload and $25K of that for Departmental (unsponsored) Research workload, and another $10,000 on Purpose FREC352114.

In this case, the Speedtypes you want are
FREC110000
FREC11DRES
FREC352114

1. Since you are encumbering salary, you will want to fill in the EMPLID field; Name will be displayed.
2. In the first row for data entry, enter $25,000 under “Original Amount.”
3. You must enter values for Account and Class Field. Here, select Account equal to 121100 and Class equal to 103.
4. Enter FREC into the Speedtype Key box. Click the magnifying glass next to Speedtype key, which will bring up all the Speedtypes beginning with FREC. Select the top one, FREC110000. Your screen will now look like this:
5. Now you want to add the other two obligation rows. Click the + button on the right side of the screen to get another row.
6. Repeat the above two steps to fill in a $25,000 obligation on Speedtype FREC11DRES. Click the + button again, and fill in a $10,000 obligation on Speedtype FREC352114. In both cases, use the same Account and Class Fields.
7. In the description field, give the obligation a name that means something to you, for example “Sample faculty salary obligat’n.”

**Important NOTE!** If you want your obligation to be automatically deobligated when a matching expense occurs (see section “Understanding the Deobligation Process”), certain chartfields of the obligation that you enter must match the chartfields of the expense. This is described in detail later in this document. If there is not a match as described later, the obligation will NOT be automatically deobligated when an expense occurs.

8. Click the Save button.
---End of Exercise---
III. VIEWING OBLIGATIONS

To view obligations, you can look at the screens themselves (as in prior section) or use a query for this purpose. These queries are in the reporting instance of the UD financial system (also known as FIRPT), which is at this URL: www.udel.edu/financials (this automatically routes you to https://boletus.nss.udel.edu:4450/psp/firpt/).

Exercise 2: View Obligations

Navigate to Query Manager on your menu: Reporting Tools>Query>Query Manager (or, navigate to Query Viewer on your menu: Reporting Tools>Query>Query Viewer)

1. Run EZQ_OBLIGATIONS_VIEW by clicking on the query name (this will run the query if you are in Query Viewer), then clicking on the “Preview Tab” in Query Manager. Please note that this query may not run correctly if you try to use the “Run” button.

You will see several prompts:

![EZQ_OBLIGATIONS_VIEW](image)

2. The first prompt - “REQD: Eff Date On or After” – is required. You must click the calendar icon to select a date. Select February 1, 2003.
3. Note that the other prompts are optional; for now, leave them blank. Click OK. Your results will look something like this:

The Query results show the obligations existing in the purposes for which you are secured.

----End of Exercise----
IV. UNDERSTANDING THE AUTOMATIC DEOBLIGATION PROCESS

A valuable feature of the Obligations module is the automatic deobligation process. This means:

- Under certain conditions (described below), an obligation will be deobligated by an automatic process when an expense “matching” the obligation occurs.
- The “remaining amount” in the screen below would be adjusted to subtract actual expenses that match the obligation (“deobligations to date”) from the “original amount.”

Q. What triggers a deobligation?

A. In order for an obligation to be automatically “deobligated,” the following conditions must be met:

- Any active deobligation will be deobligated if PS Account is in the Salary (not salary and wage) node of the PS Account tree, and if a transaction occurs where Business Unit, Chartfield1, Account, Class, and Fund all match [NOTE: Source and Program do NOT have to match].
- Also, we’re requested automatic deobligations under these circumstances, but this is NOT IMPLEMENTED YET: If PS Account is NOT in the salary node of the PS Account tree, and UDChartfield is NOT blank/null, then any active deobligation will deobligate if all of the above chartfields AND UDChartfield all match. This allows users to have non-salary obligations deobligate, but not unintentionally.
- Obligations which are marked as “inactive” will NOT keep deobligating.
- Your use of UDChartfield may evolve over time. Remember that if you populate this field one way in the transaction, you must populate it in the same way in the UD Obligations screen if you want your obligation to be automatically deobligated.

As noted, an obligation is deobligated under certain conditions, resulting in a decreasing “remaining amount.” What happens when the obligation is met? That is, what happens when the full amount of an obligation is deobligated? The answer is that the deobligation
process continues, leading eventually to a negative “remaining amount,” as long as the obligation is “active.”

**For example:**

Jane obligates $40K of Dr. Jones's salary on a given purpose/account/fund with class 102, and $50K of Dr. Smith's salary on the same purpose/account/fund, but with class 103.

The deobligation process runs every time these two people are paid from their corresponding purpose/account/class/fund. Let's say they are each paid $7K per pay period from this source for 5 pays.

**Initial:**

<table>
<thead>
<tr>
<th>Class</th>
<th>Original Amount</th>
<th>Deobligations To Date</th>
<th>Remaining Obligation</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>40,000</td>
<td>0</td>
<td>40,000</td>
</tr>
<tr>
<td>103</td>
<td>50,000</td>
<td>0</td>
<td>50,000</td>
</tr>
</tbody>
</table>

**After the first pay:**

<table>
<thead>
<tr>
<th>Class</th>
<th>Original Amount</th>
<th>Deobligated Now</th>
<th>Deobligations To Date</th>
<th>Remaining Obligation</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>40,000</td>
<td>7,000</td>
<td>7,000</td>
<td>33,000</td>
</tr>
<tr>
<td>103</td>
<td>50,000</td>
<td>7,000</td>
<td>7,000</td>
<td>43,000</td>
</tr>
</tbody>
</table>

...after five pays:

<table>
<thead>
<tr>
<th>Class</th>
<th>Original Amount</th>
<th>Deobligated Now</th>
<th>Deobligations To Date</th>
<th>Remaining Obligation</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>40,000</td>
<td>7,000</td>
<td>35,000</td>
<td>5,000</td>
</tr>
<tr>
<td>103</td>
<td>50,000</td>
<td>7,000</td>
<td>35,000</td>
<td>15,000</td>
</tr>
</tbody>
</table>

After the next pay, the total amount deobligated for Dr. Jones (class 102) will be 42,000, which EXCEEDS the original obligation. The system will deobligate the full amount even when it exceeds the obligation. This leads to a result that may be misleading if it is not fully understood by the user:
...after six pays:

<table>
<thead>
<tr>
<th>Class</th>
<th>Original Obligation</th>
<th>Amount Deobligated Now</th>
<th>Deobligations To Date</th>
<th>Remaining Obligation</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>40,000</td>
<td>7,000</td>
<td>42,000</td>
<td>-2,000</td>
</tr>
<tr>
<td>103</td>
<td>50,000</td>
<td>7,000</td>
<td>42,000</td>
<td>8,000</td>
</tr>
</tbody>
</table>

The advantage of having the deobligation process continue even after the original obligation is fully “met” is you can see that the obligation for Dr. Jones (class 102) was "overspent."

The disadvantage is that the data can be misleading if you are not familiar with the detail. If you subtotal all the obligations on this purpose, you'll see a remaining obligation of $6,000 (=-2K+8K). This could be misleading if indeed the obligation for class 103 is correct. Therefore, it is important that you know whether there is a negative remaining obligation against a purpose. More about this will be in the section on “Seeing Balances with Remaining Obligations.”
V. MANUAL DEOBLIGATION

As noted, automatic deobligation only occurs during certain conditions. You may want to manually enter information that an obligation has been partly or fully met, and have the remaining balance reflect the expense that you know has occurred. For example, you may have entered an obligation for $10,000 worth of widgets to be charged to a certain purpose code, and you know that $7,000 of those widgets have already been purchased. You know that there is a $7,000 actual charge to the purpose code, so your remaining obligation is $3,000. Here is how you would manually deobligate the $7,000, resulting in the correct remaining obligation.

**Exercise 3: Manual Deobligation**

To manually deobligate part of all of an existing obligation, go to the obligations module in FIPRD: [www.udel.edu/LAM](http://www.udel.edu/LAM).

From Menu, navigate to UD Obligations:
UD Processes
   UD Obligations

1. Select “Find an Existing Value” tab. As you will see, we will be changing an existing obligation, so you must check the “Correct History” box as shown below.

2. In this example, we will select an obligation called LINDATEST1. Enter LINDATEST1 in the UD Encumbrance ID search box, and click “Search.” This will take you to the following screen:
3. You now want to indicate that $6,000 of the widgets have been purchased. You know that a $6,000 charge has made against this purpose, so that the remaining obligation is really now $9,000. You can do this in two ways.

4. The simplest way is to enter $6,000 in the “Deobligations to Date” field. Enter 6000 into that field, and press the tab key. Your screen now looks like this:

   Note that the remaining amount changed to 9,000, which is what you know to be correct. While this method is very simple, it has two disadvantages:
   a) At “press time” for this training document, any manually entered deobligation prevents the automatic deobligations process from working. So, if you are in a situation where an obligation would normally be automatically deobligated, entering information in the “deobligations to date” field may prevent this process from happening.
   b) Because all the information is on one row, if you later spend another $2,000 on widgets, you will see that $8,000 has been deobligated, but you won’t see that it happened two different times. You may not care about this detail, so this may not be a disadvantage to you.

5. Now, we’ll do this using a second method. Delete the value you put in the “Deobligations to Date” field, and press the tab key. Your remaining amount will be back to $15,000.

6. Click the yellow “+” button at the right of your screen to get an additional row.
7. This time, we will treat the 6,000 deobligation as a negative obligation. Enter -6000 is the “Original Amount” field of the second row. Enter the same values in Account and Speedtype Key that were in row 1. Your screen now looks like this:

8. Note that the sum of the remaining amount for this obligation is 9,000 (15,000-6,000), but the information is on two rows. If you later spent another $2,000 on widgets, you could enter a third row for this information, thus retaining the detail information that the deobligation happened in multiple stages. Again, this detail may or may not be important to you.

9. Click the Save button.

----End of Exercise----
VI. INACTIVATING AN OBLIGATION

In an earlier example, a salary obligation was made for Dr. Jones, and after that obligation was fully met, the automatic deobligation process continued, resulting in a negative “Remaining Obligation.” If an obligation is fully met, and you want the deobligation process to stop, you can do this by making the obligation “inactive.”

**Exercise 4: Inactivating an Obligation**

To inactivate an obligation, thus stopping the deobligation process, go to the obligations module in FIPRD: [www.udel.edu/LAM](http://www.udel.edu/LAM).

From Menu, navigate to UD Obligations:
- UD Processes
  - UD Obligations

1. Select “Find an Existing Value” tab. As in exercise 3, you must check the “Correct History” box because you are going to change information about an existing, saved obligation.
2. Enter the UD Encumbrance ID of the obligation you want to inactivate. In this example, you will inactivate the obligation you created in Exercise 1, “ZQDJONES01.”
3. In the drop-down box next to “status,” select inactive as shown:

4. Click the Save button. Your obligation is now inactive, and the deobligation process will not continue.

---End of Exercise---
VII. SEEING BALANCES WITH REMAINING OBLIGATIONS

Queries have been written to allow you to view the remaining active obligations against a purpose along with the actual transactions and balances of that purpose. In this section, you will learn how to use these queries to see both aggregate and detail data on obligations so that you have useful information about your future commitments, and so that you will not be misled by any negative remaining obligations.

These queries look similar to the EQZ_BAL queries, but they have additional rows where there are obligations.

As with all public queries, it is important that you **DO NOT MODIFY THESE QUERIES IN ANY WAY**. If you want to make any changes to the queries, first save them under your own initials, and then make changes.

There are six queries that marry the information on actual transactions and balances with remaining obligations. These queries are:

1. `OBQ_BAL_TOTAL_W_OBLIGATN` (summary)
2. `OBQ_BAL_BY_CLASS_W_OBLIGATN` (detail)
   - Use these two queries for purposes that have no budgets that is, for fiscal “OPBAL” fund purpose. Examples of such purposes are those with 17 or 41 in digits 5 and 6 of the purpose code, e.g. CHEM17S000.
3. `OBQ_BAL_BGT_TOTAL_W_OBLIGATN` (summary)
4. `OBQ_BAL_BGT_BY_CLASS_W_OBLIG` (detail)
   - Use these two queries for purposes that have fiscal budgets, that is, for basic budget (a.k.a. 1-book or OPBAS) and self-supporting (a.k.a. 2-book or OPSS) purposes. Examples of such purposes are those with 11 or 21 in digits 5 and 6 of the purpose code, e.g. CHEM110000.
5. `OBQ_BAL_PROJ_TOTAL_W_OBLIGATN` (PTD OPBAL; summary)
6. `OBQ_BAL_PROJ_BY_CLASS_W_OBLIG` (PTD OPBAL; detail)
   - Use these two queries for purposes that have project-to-date budgets, that is, for contract and grant purposes, e.g. DSEC322147.

There is a seventh query that you may use to see all negative remaining obligations. As noted earlier, negative “remaining obligations” can lead to potentially-misleading results, so this query may be useful in helping you identify the cause of unexpected results.

7. `OBQ_OBLIG_NEG_REMAINING`
Note that all of these queries start with OBQ. These queries are in the reporting instance of the UD financial system (also known as FIRPT), which is at this URL: www.udel.edu/financials.

To use these queries, navigate to query manager (or query viewer), and enter “OBQ” in the search box:

Examples of each query’s prompt page and results follow.
OBQ_BAL_TOTAL_W_OBLIGATN (fund OPBAL; summary)

Use this query for purposes with fund equal to OPBAL (operating balance), such as those with 17 or 41 in digits 5 and 6 of the purpose code. This query will return summary information: the first row will be the same information you would see if you ran EZQ_BAL_TOTAL; the second row will show the remaining obligations (if any) for the purpose. If the purpose has no obligations entered against it, you will not see a second row.

Prompts:

Results:

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Descr</th>
<th>Year</th>
<th>Period</th>
<th>BegBal</th>
<th>Expense</th>
<th>Revenue</th>
<th>Expenses or Oblig</th>
<th>EndBalance</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARSC175123</td>
<td>OF SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td></td>
<td>41667</td>
<td>43</td>
<td>0.00</td>
<td>1590.24</td>
<td>49009.19</td>
</tr>
<tr>
<td>ARSC175123</td>
<td>REMAINING OBLIGATIONS</td>
<td>2005</td>
<td></td>
<td>888</td>
<td>0.00</td>
<td>0.00</td>
<td>1619.41</td>
<td>0.00</td>
</tr>
</tbody>
</table>
OBQ_BAL_BY_CLASS_W_OBLIGATN (fund OPBAL; detail)

Use this query for purposes with fund equal to OPBAL (operating balance), such as those with 17 or 41 in digits 5 and 6 of the purpose code. This query will return detail information: it will return both actual data for each account/class combination in that purpose, and any remaining obligations by account and class. The rows for “remaining obligations” have “remaining obligations” in the “Descr” column, as shown below.

Prompts:

![OBQ_BAL_BY_CLASS_W_OBLIGATN form]

Results:

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Account</th>
<th>Class</th>
<th>Descr</th>
<th>Year</th>
<th>Period</th>
<th>Beg Bal</th>
<th>Revenue</th>
<th>Expenses or Oblig</th>
<th>EndBalance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ARS0175123</td>
<td>138100</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>984.50</td>
<td>-984.50</td>
</tr>
<tr>
<td>2</td>
<td>ARS0175123</td>
<td>138100</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>188.20</td>
<td>-188.20</td>
</tr>
<tr>
<td>3</td>
<td>ARS0175123</td>
<td>146000</td>
<td>REMAINING OBLIGATIONS</td>
<td>2009</td>
<td>B000</td>
<td>0.00</td>
<td>0.00</td>
<td>1619.41</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>ARS0175123</td>
<td>146100</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>ARS0175123</td>
<td>145000</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>35.00</td>
<td>-35.00</td>
</tr>
<tr>
<td>6</td>
<td>ARS0175123</td>
<td>145000</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>24.95</td>
<td>-24.95</td>
</tr>
<tr>
<td>7</td>
<td>ARS0175123</td>
<td>145000</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>8</td>
<td>ARS0175123</td>
<td>145000</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>320.59</td>
<td>-320.59</td>
</tr>
<tr>
<td>9</td>
<td>ARS0175123</td>
<td>149100</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>5.00</td>
<td>-5.00</td>
</tr>
<tr>
<td>10</td>
<td>ARS0175123</td>
<td>166501</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>ARS0175123</td>
<td>166700</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>12</td>
<td>ARS0175123</td>
<td>F12300</td>
<td>DF-SOCIAL SCIENCE &amp; HISTORY</td>
<td>2005</td>
<td>7</td>
<td>41807.43</td>
<td>0.00</td>
<td>0.00</td>
<td>41807.43</td>
</tr>
</tbody>
</table>
OBQ_BAL_BGT_TOTAL_W_OBLIGATN (purposes with budgets; summary)

Use this query for purposes with fund equal to OPBAS (basic budget) or OPSS (self-sustaining, a.k.a. 2-book), such as those with 11 or 21 in digits 5 and 6 of the purpose code, or you can use this for other purposes which have budgets. This query will return summary information: the first row will be the same information you would see if you ran EZQ_BAL_BGT_TOTAL; the second row will show the remaining obligations (if any) for the purpose. **If the purpose has no obligations entered against it, you will not see a second row.**

Prompts:

<table>
<thead>
<tr>
<th>OBQ_BAL_BGT_TOTAL_W_OBLIGATN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter FISCAL Year: 2005</td>
</tr>
<tr>
<td>Enter Period: 7</td>
</tr>
<tr>
<td>Enter Purpose or partial: ITNS112123</td>
</tr>
</tbody>
</table>

Results:

Enter FISCAL Year = 2005, Enter Period=7, Enter Purpose or partial=ITNS112123

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Descr</th>
<th>Year</th>
<th>Period</th>
<th>Net Budget Amt</th>
<th>Actual Trans to Date or Oblig</th>
<th>Available Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITNS112123</td>
<td>ADMIN SOFTWARE MAINT</td>
<td>2005</td>
<td>7</td>
<td>597467.000</td>
<td>316874.850</td>
<td>177792.15</td>
</tr>
<tr>
<td>ITNS112123</td>
<td>REMAINING OBLIGATIONS</td>
<td>2006</td>
<td>800</td>
<td>0.000</td>
<td>-36718.150</td>
<td>0.00</td>
</tr>
</tbody>
</table>
OBQ_BAL_BGT_BY_CLASS_W_OBLIG (purposes with budgets; detail)

Use this query for purposes with fund equal to OPBAS (basic budget) or OPSS (self-sustaining, a.k.a. 2-book), such as those with 11 or 21 in digits 5 and 6 of the purpose code, or you can use this for other purposes which have budgets. This query will return detail information: it will return both actual data for each account/class combination in that purpose, and any remaining obligations by account and class. The rows for “remaining obligations” have “remaining obligations” in the “Descr” column, as shown below.

Prompts:

**OBQ_BAL_BGT_BY_CLASS_W_OBLIG**

Enter FISCAL Year: 2005
Enter Period: 7
Enter Purpose or partial: ITNS112123

Results:

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Account</th>
<th>Class</th>
<th>Descr</th>
<th>Year</th>
<th>Period</th>
<th>Net Budget Amt</th>
<th>Actual Trans to Date or Oblig</th>
<th>Available Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ITNS112123</td>
<td>148000</td>
<td>ADMIN SOFTWARE MAINT</td>
<td>2005</td>
<td>7</td>
<td>597487.00</td>
<td>0.000</td>
<td>597487.00</td>
</tr>
<tr>
<td>2</td>
<td>ITNS112123</td>
<td>148400</td>
<td>ADMIN SOFTWARE MAINT</td>
<td>2005</td>
<td>7</td>
<td>0.000</td>
<td>406522.48</td>
<td>-406522.48</td>
</tr>
<tr>
<td>3</td>
<td>ITNS112123</td>
<td>148400</td>
<td>REMAINING OBLIGATIONS</td>
<td>2000</td>
<td>2000</td>
<td>133523.90</td>
<td>301546.86</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>ITNS112123</td>
<td>148000</td>
<td>ADMIN SOFTWARE MAINT</td>
<td>2005</td>
<td>7</td>
<td>0.000</td>
<td>133523.90</td>
<td>-133523.90</td>
</tr>
<tr>
<td>5</td>
<td>ITNS112123</td>
<td>148500</td>
<td>ADMIN SOFTWARE MAINT</td>
<td>2005</td>
<td>7</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>ITNS112123</td>
<td>153200</td>
<td>ADMIN SOFTWARE MAINT</td>
<td>2005</td>
<td>7</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
OBQ_BAL_PROJ_TOTAL_W_OBLIGATN (project-to-date purposes; summary)

Use this query for purposes which have project-to-date data, e.g. contracts and grants. This query will return summary information: the first row will be the same information you would see if you ran EZQ_BAL_PROJ_TOTAL; the second row will show the remaining obligations (if any) for the purpose. **If the purpose has no obligations entered against it, you will not see a second row.**

Prompts:

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter FISCAL Year:</td>
<td>2005</td>
</tr>
<tr>
<td>Enter Period:</td>
<td>7</td>
</tr>
<tr>
<td>Enter Purpose or partial:</td>
<td>DSEC322147</td>
</tr>
</tbody>
</table>

Results:

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Descr</th>
<th>Year</th>
<th>Period</th>
<th>Sam Proj to Dt Bud</th>
<th>Revenues</th>
<th>Expenses or Obligs</th>
<th>Project Cash Balance (rev-exp)</th>
<th>Budg Balance (bgl-exp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DSEC322147</td>
<td>2005</td>
<td>7</td>
<td>100000.000</td>
<td>100014.51</td>
<td>100014.51</td>
<td>0.00</td>
<td>-14.51</td>
</tr>
<tr>
<td>2</td>
<td>DSEC322147</td>
<td>2005</td>
<td>7</td>
<td>0.000</td>
<td>0.00</td>
<td>13003.86</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
OBQ_BAL_PROJ_BY_CLASS_W_OBLIG (project-to-date purposes; detail)

Use this query for purposes which have project-to-date data, e.g. contracts and grants. This query will return detail information: it will return both actual data for each account/class combination in that purpose, and any remaining obligations by account and class. The rows for “remaining obligations” have “remaining obligations” in the “Descr” column, as shown below.

Prompts:

Results:
This query shows you all your negative remaining obligations, so you know to look at DETAIL for these purposes. In the example of Dr. Jones and Dr. Smith’s salary given earlier, it was shown that negative remaining obligations can lead to misleading results if you only look at the total remaining obligation for a purpose.

Prompts:

The first prompt is required; the others are optional.

Results: