# 1.  Overview

## 1.1  Introduction

The Recurring Calculation (RC) tool is designed to provide users the ability to define Selection Criteria, Calculation and Posting processes against actual transactions, account balances and budgets.  These definitions are stored within the system such that they may be a 'recurring' process which is available on demand or triggered based on defined activity within IFAS.

(Please Note:  The initial release of Recurring Calculations (RC) only functions within the General Ledger and Endowment Management subsystems.  Future releases will expand the functionality across all subsystems.)

## 1.2  Key Features

The following is a brief listing of the key features currently available

Selection Criteria
> Selection Classes – Ability to define 3 levels of categorization

Activation Controls – Ability to immediately activate or inactivate a definition or be active within given date range
> Ledger Selection – Specific GL and JL Ledgers for analysis
> Execution History – Provides automatic update for 'Last Run' date and status
> Narrative – Provides unlimited text for purpose and use descriptive information

Data Items
> Select data to be utilized from pre-defined information categories
> Item Names provided in both database naming and English wording
> Ability to Optimize the list based on selected items

Analysis Steps
> Step based definitions indicating the order in which actions will be performed
> Individual Steps are documented with a narrative
> Steps may contain both Filter and Calculation rules
> Steps may reference one or more previously defined steps

Output
> User defined output to pre-defined subsystem Transaction Sets for posting
> Visible display of the target fields to be populated in the Transaction Set
> Ability to View the final results prior to posting the transactions to the system

## 1.3  Getting Started

Before defining a Recurring Calculation the user should have an outline of the following information:

What will be the Selection Criteria?

Account(s), Transaction, Date Range, etc. as well as what table(s) this information comes from.


What Calculations are to be performed against the selected information?

For example:

Multiply selection based on a standard rate found in the selected subsystem

(i.e., Standard Overhead rate, Cost Recovery rate, etc.)

Calculate the Average Daily Balance for the selected accounts and date range

Calculate the ending balance of an account and transfer it to the Balance Sheet for Year-end closing purposes


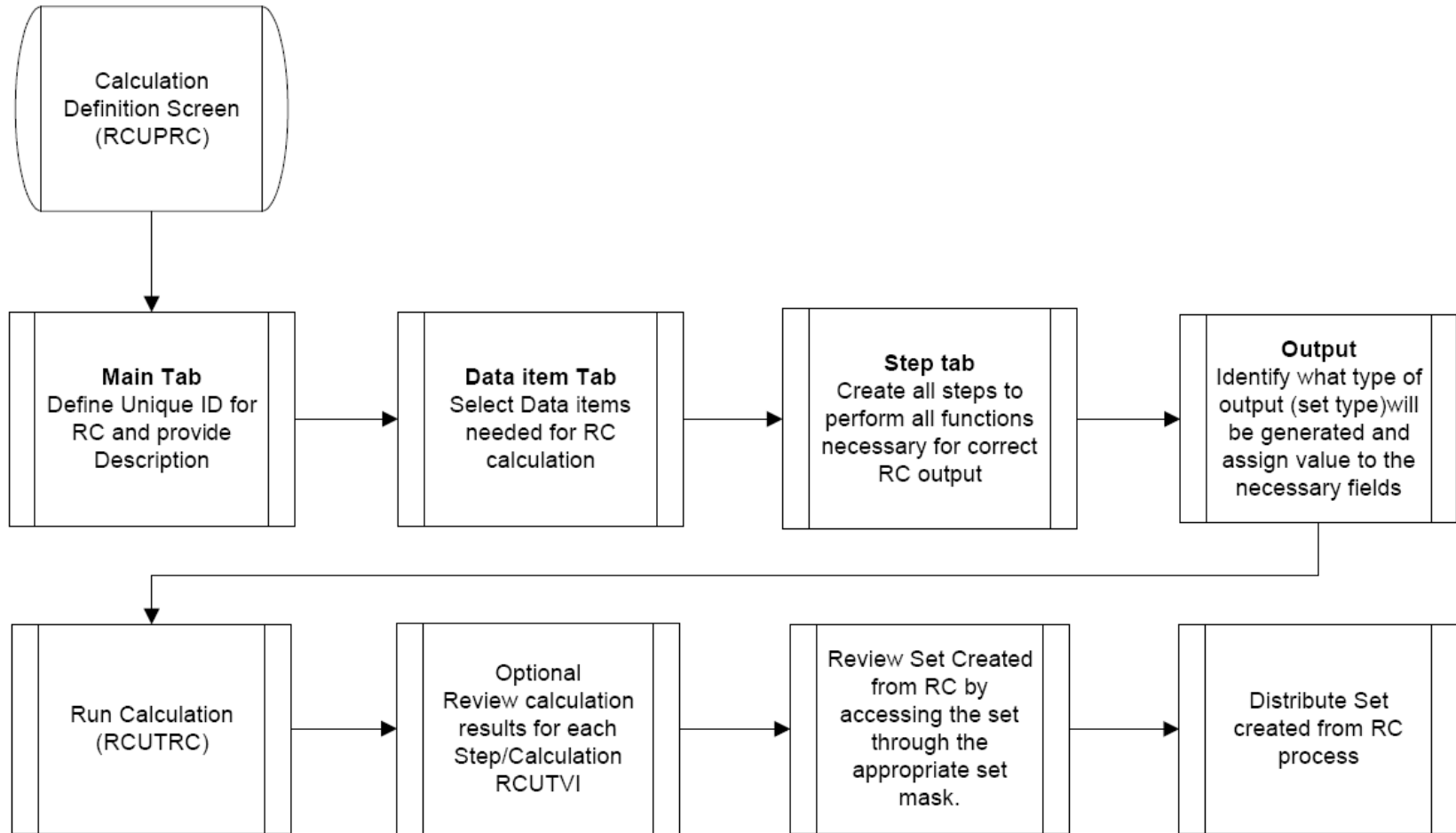Where will the resulting calculated value(s) be posted?

To the same Org Key and different Object Code
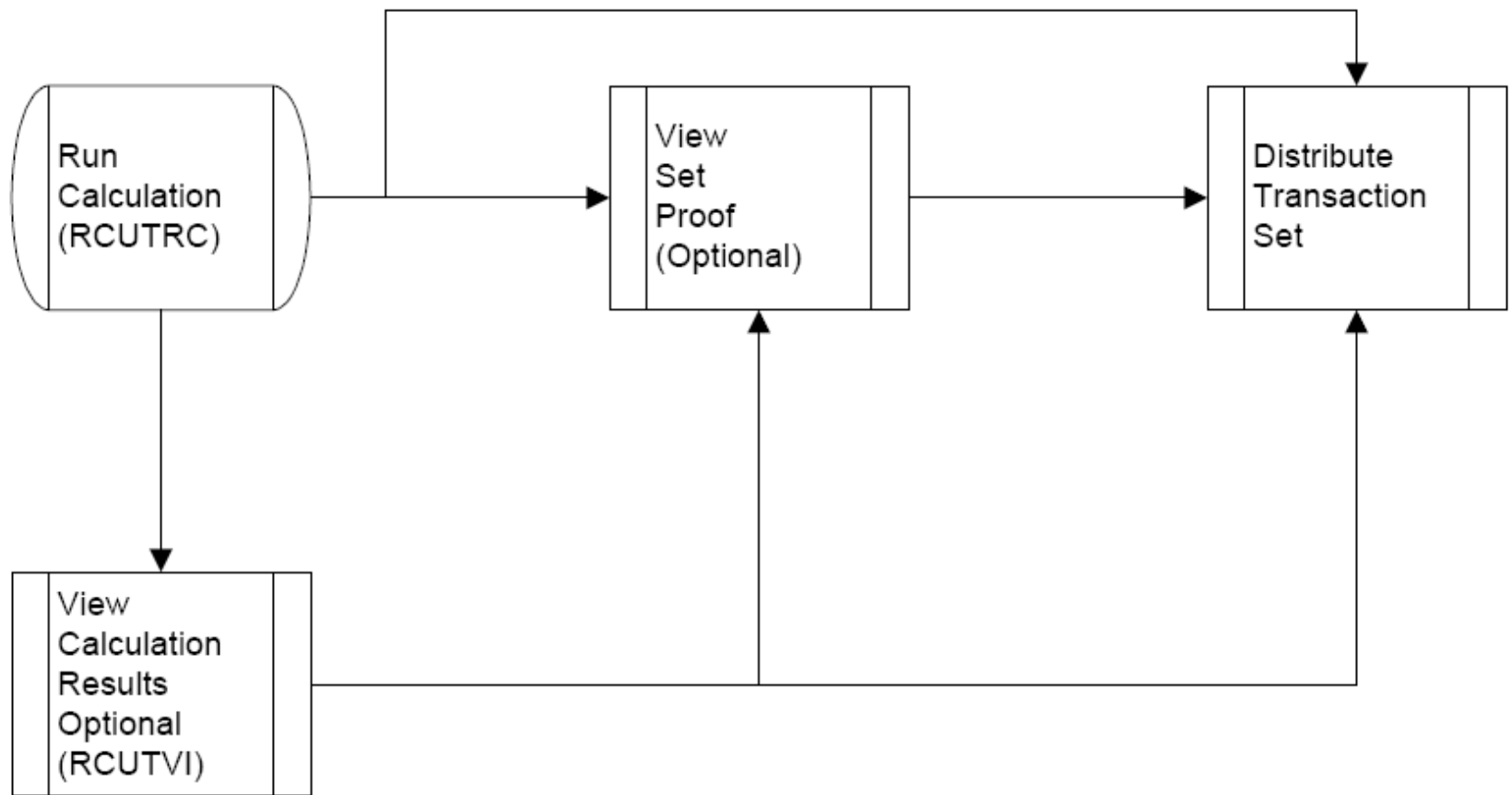
To a specific Administrative Account

To multiple accounts based on the calculation formula

## 1.4  Flow Diagram

### 1.4.1  Define a Recurring Calculation

```
        ┌─────────────────────┐
        │     Calculation     │
        │  Definition Screen  │
        │      (RCUPRC)       │
        └─────────────────────┘
                   │
                   ▼
```

| Main Tab Define Unique ID for RC and provide Description | → | Data item Tab Select Data items needed for RC calculation | → | **Step tab** Create all steps to perform all functions necessary for correct RC output | → | **Output** Identify what type of output (set type)will be generated and assign value to the necessary fields |

| Run Calculation (RCUTRC) | → | Optional Review calculation results for each Step/Calculation RCUTVI | → | Review Set Created from RC by accessing the set through the appropriate set mask. | → | Distribute Set created from RC process |

## 1.4.2 Process a Previously Defined Recurring Calculation

```
Run
Calculation
(RCUTRC)
```

```
View
Set
Proof
(Optional)
```

```
Distribute
Transaction
Set
```

```
View
Calculation
Results
Optional
(RCUTVI)
```

## 2. Setup

### 2.1 Calculation Definition

The first step in defining a Recurring Calculation is to access the Calculation Definition process. This is found on the IFAS 'All' Tab as RC UP RC. This will provide access to all individual definition screens for the Recurring Calculation process. As shown above in the Flow Diagram there are four (5) separate elements in creating a Calculation Definition. These are described in detail below.

1. The **Main** form displays the general information pertaining to this Calculation Definition.



**ID:** Name of recurring calculation. This must be unique. Required

**Description:** Short description for recurring calculation.  Required

**Classes/Primary:** Assigns a primary selection class.  Primary selection classes are defined in RCUPCD with Code ID = "PC". Optional

**Classes/Secondary:** Assigns a secondary selection class.  Secondary selection classes are defined in RCUPCD with Code ID = "SC". Optional

**Classes/Tertiary:** Assigns a tertiary selection class.  Tertiary selection classes are defined in RCUPCD with Code ID = "TC". Optional

**Active:** Indicates active/inactive status of recurring calculation.

**Auto:** Indicates whether a recurring calculation should run automatically at distribution time to select its data from a recently posted set. This must be configured with a Workflow Model.

**Start:** Start date for recurring calculation.  Used when defining a time period for which the RC will be allowed to operate.

**End:** End date for recurring calculation.  Used when defining a time period for which the RC will be allowed to operate.

**GL Ledger:** The default GL Ledger Code for this recurring calculation.  Required

**JL Ledger:** The default JL Ledger Code for this recurring calculation.  Required

**Last Run:** The last date this recurring calculation was run.

**Last Status:** Identifies whether or not the recurring calculation was run successfully the last time it was launched.

**Narrative:** Detailed description for this recurring calculation.

2.  The **Data Item** form provides a method of selecting the source items from which the data will be selected.  This section also determines at which level that information that is pulled into the RC will be summarized.

| Main | Data Item | Steps | Output | |
|------|-----------|-------|--------|--|

Category: GLTRNS    General Ledger Transaction Detail     Build List    Optimize List

| Select | Table | Column | Name | |
|--------|-------|--------|------|--|
| ☑ | GLTTrnsDetail | Dr | DEBIT | |
| ☑ | GLTTrnsDetail | Cr | CREDIT | |
| ☑ | GLKKeyMaster | Misc03 | MISCELLANEOUSCODE3 | |
| ☑ | GLTTrnsDetail | GlKey | ORGKEY | |
| ☑ | GLTTrnsDetail | GlObj | OBJECT | |
| ☑ | GLTTrnsDetail | GlGr | GLLEDGERCODE | |
| ☑ | | | | |

Record 1 of 6

**Category:** The category from which the data items will be selected.  Required

**Select:** If checked, this item has been selected and may be used in calculations and output.

**Table:** The table from which this data item will be selected.

Table Description (Single-Entry Mode Only): The description of the Table.  Read-Only.

**Column:** The column from which this data item will be selected.

Column Description (Single-Entry Mode Only): The description of the Column.  Read-Only.

**Name:** The user defined name for the data item.

**Build List:** Click on this button to show all data items for this category.  Previously selected items will remain selected.

**Optimize List:** Click on this button to remove unselected data items for this category.

3.  The **Steps** form defines the order of operations for filters and calculations.

**Step:** The order in which the steps will be executed.  Required

**ID:** This name will be used to reference this step in calculations and output.  Required

**Description:** A description of what the step is.  Required

**Date Table**: The table on which date selection will be performed.  This table must be part of the data item category.  Required

**Date Column:** The column on which date selection will be performed.  This column must be part of the data item category.  Required

**Start:** The starting selection date for this step of the recurring calculation.

**End:** The ending selection date for this step of the recurring calculation.

**Symbolic Start:** The starting selection date for this step of the recurring calculation.  This must be a valid three-character symbolic code. (i.e.  FYB, MOB, QTB)

**Symbolic End:** The ending selection date for this step of the recurring calculation.  This must be a valid three-character symbolic date code. (i.e.  FYE, MOE, QTE, RDT)

**Narrative:** Detailed description for the step.


Filter

A **Filter** provides a method of paring down the data selected for the given step.

| | Filter | Calculations | Narrative | | | | |
|---|---|---|---|---|---|---|---|
| Sequence | ( | Table | Column | Relational Op | Value | ) | Logical Op |
| 1 | | GLKKeyMaster | GrpPart01 | = | 10 | | AND |
| 2 | | Data Item | GLLEDGER | = | GL | | AND |
| 3 | | Data Item | TYPE | = | AS | | |
| 6 | | | | | | | |

Record 1 of 3

**Sequence:** The order that the filter rows will be applied.  Required

**(:** Opening parenthesis.  Used to identify order of operations.

**Table:** A table from the category or "Data Item" to specify a named data item.  Required

**Column:** A column from the category or named data item.  Required.

**Relational Op:** The relation operator to apply to the Table/Column.  E.g. =, <, >.  Required

**Value:** The value to which the table/column is compared to.  Required

**):** Closing parenthesis.  Used to identify order of operations.

**Logical Op**: Identifies how the current line should be compared to the next line.  E.g. AND, OR.

**Calculations** define the arithmetic functions that will be performed for the given step.

Filter | **Calculations** | Narrative

Sequence: 1

Name: NET

Global Flag:

Expression: DEBIT - CREDIT

Source:

Value: [Append]

Decimal Places: 0

Rounding Control Level:

Record 1 of 1

**Sequence:** The order that the calculations will be performed.  Required

**Name:** The name used to identify the calculation.  This name may be used on the Output tab.

**Expression:** The arithmetic calculation to perform.  Named data items and step/calc's may be referenced from prior steps.  There are some special codes that may also be used here:

**For All Subsystems**

AVGBAL(Period, Amount):  The average balance on a daily, weekly, monthly, quarterly, or yearly basis.  The Period may be D for daily, M for monthly, Q for quarterly, or Y for yearly. The Amount parameter is the named data item that contains values to compute average balance.  The Credit and Debit parameters are the named data items which contain values to compute average balance.  NOTE:  if you are calculating the daily AVGBAL for an account that has a normal DR balance type then the calculation should look like: AVGBAL(D,DEBIT).  Conversely if you are calculating a daily AVGBAL for an account that has a normal balance type of CR then the calculation would look like AVGBAL(D, CREDIT).

FDKEY(Ledger, Org Key): Returns the Org Key from the FDID Common Code.  The Fund is determined from the passed Org Key.

FDOBJ(Ledger, Org Key): Returns the Object Code from the FDID Common Code.  The Fund is determined from the passed Org Key.

SUM(Data Item): The sum of a named data item in the current selection.  This must be a numeric field.

XOBJ(Ledger Code, Object Code): Returns an Object Code from the GLOB Common Code.  The index of the associated code to get the Object Code from is determined by positions 5-6 of the passed Object Code.

**Endowment Management**

DFKEY(Ledger Code, Donor Fund ID, Org Key Type): The Donor Fund Org Key from Endowment Management.  If more than one Account entry exists, the first one will be returned.

DFOBJ(Ledger Code, Donor Fund ID, Org Key Type): The Donor Fund Object Code from Endowment Management.  If more than one Account entry exists, the first one will be returned.

INKEY(Ledger Code, Investment ID, Org Key Type): The Investment Org Key from Endowment Management.  If more than one Account entry exists, the first one will be returned.

INOBJ(Ledger Code, Investment ID, Org Key Type): The Investment Object Code from Endowment Management.  If more than one Account entry exists, the first one will be returned.

PLKEY(Ledger Code, Pool ID, Org Key Type): The Pool Org Key from Endowment Management.  If more than one Account entry exists, the first one will be returned.

PLOBJ(Ledger Code, Pool ID, Org Key Type): The Pool Object Code from Endowment Management.  If more than one Account entry exists, the first one will be returned.
The following definitions allow posting of both Debit and Credit entries to occur in the same Org Key and Object as found in the Selection Criteria

DFKEY (Ledger, DonorFundID, Type)
DFOBJ (Ledger, DonorFundID, Type)

PLKEY (Ledger, PoolID, Type)
PLOBJ (Ledger, PoolID, Type)

INKEY (Ledger, InvestmentID, Type)
INOBJ (Ledger, InvestmentID, Type)


**General Ledger**

AVGBAL(Period, Credit, Debit):  The average balance on a daily, monthly, quarterly, or yearly basis.  The Period may be D for daily, M for monthly, Q for quarterly, or Y for yearly. The Credit and Debit parameters are the named data items which contain values to compute average balance.  NOTE:  if you are calculating the daily AVGBAL for an account that has a normal DR balance type then the calculation should look like:  AVGBAL(D,DEBIT).  Conversely if you are calculating a daily AVGBAL for an account that has a normal balance type of CR then the calculation would look like AVGBAL(D, CREDIT).

**Work Order**

WODRKEY(Ledger Code, Work Order): Returns the Debit Account Org Key for this Work Order.

WOCRKEY(Ledger Code, Work Order): Returns the Credit Account Org Key for this Work Order.

WODROBJ(Ledger Code, Work Order): Returns the Debit Account Object Code for this Work Order.

WOCROBJ(Ledger Code, Work Order): Returns the Credit Account Object Code for this Work Order.

Output

Allows the user to specify what type of output setup will be created by the recurring calculation.

| Main | Data Item | Steps | Output |
|------|-----------|-------|--------|
| **Category** | **Description** | | |
| EM_SET | Endowment Management Set | | |
| JE_SET | General Ledger Journal Entry Set | | |
| | | | |

◄ ► ✚ 🗎▾                                                                 Record 2 of 2 🔓

C**ategory:** The type of output that will be created.  Required

**Description:** The description of the output category.  If the created output is a set and the set has a description, this description will be used.  Required

Output Details

Defines how each column in the output will be populated.

**Output Details**

[ Build List ]  [ Optimize List ]

| Target | Source | Name | Literal |
|--------|--------|------|---------|
| GL Ledger Code | | | GL |
| Work Aut | | | |
| Org Key | Data Item | ORGKEY | |
| Activity | | | |
| Object | Data Item | OBJECT | |
| JE ID | Seed | JEID | |
| Year | | | 2008 |

◄ ► ✚ 🗎▾                                                                 Record 1 of 55 🔓

**Target BT20 Table (Single-Entry Mode Only):**  The table to which the output will be written.  Required

**Target Property (Single-Entry Mode Only):** The property name to which the output will be written.  Required

**Target:** A column in the output type selected.

**Source:** The source from which the data to populate this field will come.  This may be "Data Item", a named step, or "Seed".

**Name:** The name of the specific item from the source to use to populate this field.  This will be a named data item, a named calculation, or a SYNO seed category depending on the chosen Source.

**Literal:** A specific value to assign to this field.  Depending on the type of field being written to, this may be a character value, numeric, or date.  If the target field is a date, symbolic date codes may be used.  Note: A literal value may not be used if a Source/Name is specified.

**Build List:** Click on this button to show all fields for this output category.  Previously entered output details will remain unchanged.
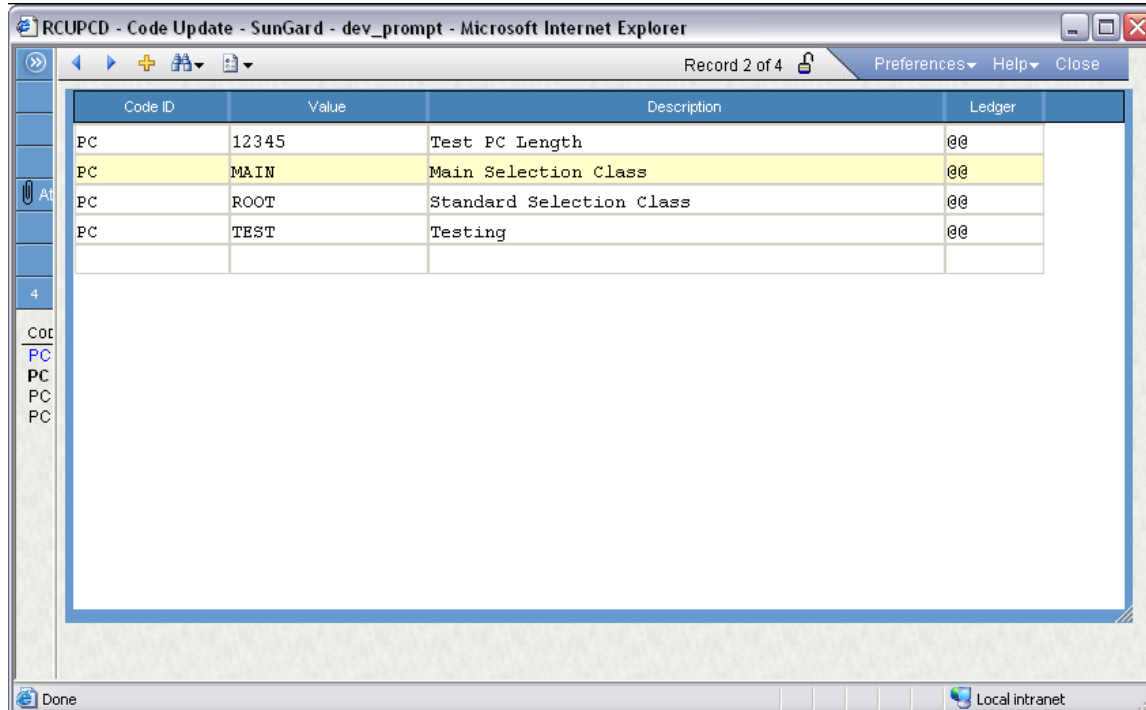
**Optimize List:** Click on this button to remove any unused output details for this category.

## 2.2  Advanced

# The Code Update process can be accessed from the IFAS Dashboard 'All' tab as RC UP CD.

Defines coded information related to the Recurring Calculations subsystem.  Most of these codes have been pre-defined and should not be changed.  Only codes specifically identified in the RCUPRC documentation should be added or changed.

Grid Mode:

Single-Entry Mode:



**Code ID:** The type of Code.  These are defined in NUUPCD where Code Category is "RCCD".  Required.

**Value:** A value for this type of code.  Required.

**Description:** Describes this Code ID/Value combination.

**Ledger:** The default Ledger Code for this recurring calculation.  May be "@@" for all ledgers or any GL/JL side ledger.

## 2.3  Best Practices

# 3. Processes

## 3.1 Create Transaction Sets Ready for Distribution

### 3.1.1 Run Calculation

This utility will read a previously defined Recurring Calculation and create the output as specified in the definition.

The Run Calculation process is accessed from the IFAS Dashboard 'All' tab as RC UT RC.

The following questions are asked at run time:

**Enter the Recurring Calculation ID:** This is the ID of the definition as created using the RCUPRC Screen.

**Enter the Calculation Date:** This date will be used as the base date for the calculation. Symbolic dates will be calculated using this date.

**Override Default Ledgers:** Answering "Y" to this question will prompt the user to specify the GL and JL Ledger Codes to use other than those entered in the definition on the RCUPRC screen.

**Override Set ID:** Answering "Y" to this question will prompt the user to specify the Set ID to use other than those entered on the Output section of the definition on the RCUPRC screen.

**Alter Average Balance Start and End Dates:** Answering "Y" to this question will prompt the user to specify a Start and End Date to use for Average Balance Calculations other than the current month begin/end dates.

**Save Intermediate Calculation Results:** Answering "Y" to this question will save the intermediate results of the calculation for later viewing via the RCUTVI screen. Viewing these results can be helpful in determining why a calculation is not getting the results expected.

## 3.1.2 View Intermediate Results

The View Intermediate Results screen is accessed from the IFAS Dashboard 'All' tab as
  RC UT VI.

This screen displays selected data rows and calculation results for each step of a previously run Recurring Calculation (via RCUTRC).
The calculation must have been run with "Save Intermediate Calculation Results?" set to "Yes".



Entity List

**Recurring CALC:** The recurring calculation ID being viewed.  Pick the ID to select.

**Job Numbers:** A list of job numbers that have stored intermediate results for the selected Recurring CALC.  Click on the job number to select.

Tools

Delete old results: Deletes old Recurring Calculation intermediate results based on settings.

RC Intermediate Results

 A tab is shown for each step of the Recurring Calculation.  The first tab will always be "*START*" which shows the initial data selection.  The last tab will always be "*END*" which shows the final results.  Columns are shown for each data item and calculation within a step.  Rows show the data item values or calculation results.

# 3.2  Tutorial #1

**Balance Forward Entry**

A balance forward RC is the recommended way to create an opening balance at the beginning of a new fiscal year.  The recurring calculation is setup to look at the prior year transaction information and use that data to calculate a beginning balance for the current fiscal year.

This RC is broken down into steps by object type (AS, LI, FB etc).  The RC depending upon the complexity of your closing could be as few as 3 records or as many as you need so that you can get the balance forward posting you need for your organization. We will first look at brining forward the Asset balances.

**Step 1**
Go to the RCUPRC screen.

Once this screen is open you will click on the Add record button.

You will then enter an ID, Description and select a Category.  I have also elected to assign a Primary selection class as well as setting the RC to Active in the "Activation Controls" section.  Since the RC is now active I will be able to run the RC when I am ready to test it without having to go back to the main tab and activate it later.  The name that I used for my recurring calculation is YEBFFUND10.  If I break this down there is logic in the naming convention that I used (YE - year End, BF - Balance Forward, FUND10 -  Fund 10).  It is advisable that if you are creating separate RC's for individual Funds or grouping of funds that you use a naming convention that will allow you to easily identify each of the RC's.

**Step 2** – **Add the Data items**

For this RC I chose the category GLTRNS_GL which is a standard Category and it will meet the needs that we have for creating balance forward entries. The category GLTRNS_GL has the links that are shows in the screen shot below. The linking of the tables shown provides us with the capability to apply a filter to each step to limit the subset of information that we process with each step. Our selection of data will be based on transactional information or items on in GLUPKY or GLUPOB. So this category will provide us with all of the options we will need.

| Category: | GLTRNS_GL | Type: I ▼ Input Category |
|---|---|---|
| Description: | General Ledger Transaction Detail – GL Side Only | |
| BT20 Table: | GLTTrnsDetail | |
| Date Property: | Date ▼ | |

**Details** Associations

| Sequence | From Property | To BT20 Table | To Property |
|---|---|---|---|
| 1 | GlGr | GLKKeyMaster | Gr |
| 2 | GlKey | GLKKeyMaster | Key |
| 3 | GlGr | GLOObjMaster | Gr |
| 4 | GlObj | GLOObjMaster | Obj |
| 5 | GlGr | WoMaster | Gr |
| 6 | Wo | WoMaster | No |
| 7 | | | |

Now that we have picked out Category and have an understanding of the links that in the category and the data that is made available to use we are ready to start our Balance Forward Recurring Calculation. The screen shot of the Data Items added below will show the items that are necessary for the RC to be able to process the information that we need in a fashion that will be useful.

| ID: YEBFFUND10 | Description: Year End Balance Forward Fund 10 |
|---|---|

Main | **Data Item** | Steps | Output

Category: GLTRNS_GL    General Ledger Transaction Detail - GL Side Only    Build List    Optimize List

| Select | Table | Column | Name | |
|---|---|---|---|---|
| ☑ | GLTTrnsDetail | GlGr | GLLEDGERCODE | |
| ☑ | GLTTrnsDetail | GlKey | ORGKEY | |
| ☑ | GLTTrnsDetail | GlObj | OBJECT | |
| ☑ | GLTTrnsDetail | Dr | DEBIT | |
| ☑ | GLTTrnsDetail | Cr | CREDIT | |

The data items that you chose determine at what level the information that is being selected will be summarized.  It also determines what items will be available for use on the Calculation section of the Steps tab as well as the Output tab.  You are probably thinking that you will need other items available like Object Type and Fund so that we can limit the records that are being used in each step.  Since the tables that hold these items are included in the category as show in the prior screen shot all of the items in each of those tables will be available to us in the Filter portion of the RC creation so it is not necessary to add them here.

**Step 3** - **Step setup**

The setup of the Step tab involves three pieces.

The first of those three pieces is setting up the parameters on the top portion of the Step tab.  You will need to give your step an ID and a description.  You will also need to identify the Table and Column that you want the RC to analyze for the dates that will be used to limit the records that are returned.   The RC will use the table and date field that you identified to filter the records to just the date subset of records that you want.

You will see in the screen shot below that I called the Step "ASSETS" and the table and date that I am using are the GLTTrnsDtl and the Date field.  The fields are populated via the drop down.  The fields that you have available to you here are determined by the tables that

are in the Category that you selected.  Here is where we also determine the date range for the transactions that we want to consider.  You can use the start and end date fields to identify a specific date or you can use the Symbolic start and end date fields to reference symbolic points in time.  For the start and end dates we will want to use the Symbolic Start and End so that there will be no need to edit them every year when we run the RC.  The symbolic start and end dates support the three character codes that were identified earlier in the user guide.  We will want to use FYB and FYE for the start and end dates respectively.  When we run the RC and put in the date that corresponds to the last date of our fiscal year the RC will pick up all of the transaction for the Fiscal Year.

| Main | Data Item | **Steps** | Output |

| Step: | ID: | | Description: | | |
| 1 | ASSETS | | Assets Balance Forward | | |
| Date Table: | GLTTrnsDetail | Start: | | Symbolic Start: | FYB |
| Date Column: | Date | End: | | Symbolic End: | FYE |

The second of those three pieces is setting up the filter.  The filter is used to determine the subset of records that the step will process.  The filter we have setup here will limit the records for this step to just Objects with a type of AS and items in fund 10. (GrpPart01 is the Fund Part).  The screen shot shows the setup for these to filter items.

| **Filter** | Calculations | Narrative |

| Sequence | ( | Table | Column | Relational Op | Value | ) | Logical Op |
|---|---|---|---|---|---|---|---|
| 1 | | GLKKeyMaster | GrpPart01 | = | 10 | | AND |
| 2 | | GLOObjMaster | Type | = | AS | | |
| 3 | | | | | | | |

The third step in the "Step" tab is to setup the calculation(s) that will be necessary for that step to generate the values will be used in the Output or need to use in a subsequent step/calculation.

| Filter | **Calculations** | Narrative |
|--------|------------------|-----------|

**Sequence:** 1      Source:

**Name:** ASTTL      Value:     Append

Global Flag: S    Calculation value Specific to each output row processed     Decimal Places: 2

**Expression:** DEBIT-CREDIT

Rounding Control Level:

◁ ▷ ✚ ▣▾      Record 1 of 1

**Step 4 – Output setup**

The third step in the RC is to setup the output for creating the Assets Balance forward entry in a Journal Entry.  You have the option to populate every field in the JE set while defining the output.  But for a Balance Forward entry only a few need to be populated.  The screen shot shows the setup that would be necessary for populating the JE with the minimum fields.

| Main | Data Item | Steps | **Output** |

Category: JE_SET ▼  General Ledger Journal Entry Set          **Output ID:** JE_SET0001

Description: Assets Balance Forward

◄ ► ➕ ▤▾                                                                Record 1 of 1 🔓

| **Output Details** | Narrative |

Build List     Optimize List

| Target | Source | Name | Literal |
|---|---|---|---|
| GL Ledger Code | Data Item | GLLEDGERCODE | |
| Org Key | Data Item | ORGKEY | |
| Object | Data Item | OBJECT | |
| JE ID | Seed | JEID | |
| Type | | | ST |
| Description | | | Balance Forward AS |
| Set ID | Seed | BTCHIDJE | |
| Debit | ASSETS | ASTTL | |
| Subsystem | | | JE |
| Date | | | CDT |
| Accrual | | | Y |
| Cash | | | Y |
| Modified Accrual | | | Y |
| Intrafund Offset | | | N |
| Interfund Offset | | | Y |
| Hit Other | | | N |
| | | | |

◄ ► ➕ ▤▾                                                                Record 1 of 16 🔓

The first three lines in the output set use items from the data items tab.  You also have the capability to use a literal value for this field.

The first three rows which are for the Ledger, Orgkey and Object code fields will all be populated with the data item as the source for the data. Since data item was setup with the level of information to be at the ledger/Orgkey/Object we can use the data item to populate that portion of the JE. Many of the other fields are default populated with information that is defaulted in the category that we are using. The items that I added and chose the value for are the Set ID and JEID which I chose to use a "Seed" value. When you chose a Source of Seed the name column drop down will be a list of all of the SYNO common codes. You will need to make sure that you select a seed that is seed that is designed to be used with that output type. The Description that I input for the Assets record will allow me to distinguish the records that are created from this output. The final item that I have setup is the Debit that is being created. The source of the Debit is the Step and Calculation that were just setup. The step was called "ASSETS" and the calculation was called "ASTTL" these go into the Source and Name columns respectively.

Here are the screen shots of the setup for the Liabilities and the Fund Balance forward entries. I will show the closing of Net Income as a separate item after the LI and FB screen shots.

**Liabilities Closing screens shots**

| Filter | **Calculations** | Narrative |
| --- | --- | --- |

**Sequence:** 1                                    Source: [            ▼]

**Name:** LITTL                                    Value: [            ▼]  [Append]

Global Flag: [S ▼]  Calculation value Specific to each output row processed                Decimal Places: 2

**Expression:** CREDIT-DEBIT

| Main | Data Item | Steps | **Output** |

**Category:** JE_SET    General Ledger Journal Entry Set    **Output ID:** JE_SET0002

**Description:** Liabilties Balance Forward

◀ ▶ ➕ ▤▾      Record 2 of 4 🔓

| **Output Details** | Narrative |

Build List    Optimize List

| Target | Source | Name | Literal |
|---|---|---|---|
| GL Ledger Code | Data Item | GLLEDGERCODE | |
| Org Key | Data Item | ORGKEY | |
| Object | Data Item | OBJECT | |
| JE ID | Seed | JEID | |
| Type | | | ST |
| Description | | | Balance Forward LI |
| Set ID | Seed | BTCHIDJE | |
| Credit | LIABILITIES | LITTL | |
| Subsystem | | | JE |
| Date | | | CDT |
| Accrual | | | Y |
| Cash | | | Y |
| Modified Accrual | | | Y |
| Intrafund Offset | | | N |
| Interfund Offset | | | Y |
| Hit Other | | | N |

**Fund Balance Closing Screen shots**

Main | Data Item | **Steps** | Output

| Step: | ID: | | Description: |
|---|---|---|---|
| 3 | FUNDBALANCE | | Fund Balance BF |

| | | Start: | | Symbolic Start: | FYB |
|---|---|---|---|---|---|
| Date Table: | GLTTrnsDetail | | | | |
| Date Column: | Date | End: | | Symbolic End: | FYE |

Record 3 of 4

**Filter** | Calculations | Narrative

| Sequence | ( | Table | Column | Relational Op | Value | ) | Logical Op |
|---|---|---|---|---|---|---|---|
| 1 | | GLKKeyMaster | GrpPart01 | = | 10 | | AND |
| 2 | | GLOObjMaster | Type | = | FB | | |
| 8 | | | | | | | |

Filter | **Calculations** | Narrative

| | | | |
|---|---|---|---|
| Sequence: | 1 | Source: | |
| Name: | FBTTL | Value: | Append |
| Global Flag: | S | Calculation value Specific to each output row processed | Decimal Places: 2 |
| Expression: | CREDIT-DEBIT | | |

| Main | Data Item | Steps | **Output** |

**Category:** JE_SET ▾ General Ledger Journal Entry Set    **Output ID:** JE_SET0003

**Description:** Fund Balance Balance Forward

◄ ► ➕ ▤▾                                                    Record 3 of 4 🔓

| **Output Details** | Narrative |

[Build List] [Optimize List]

| Target | Source | Name | Literal | |
|---|---|---|---|---|
| GL Ledger Code | Data Item | GLLEDGERCODE | | |
| Org Key | Data Item | ORGKEY | | |
| Object | Data Item | OBJECT | | |
| JE ID | Seed | JEID | | |
| Type | | | ST | |
| Description | | | Balance Forward FB | |
| Set ID | Seed | BTCHIDJE | | |
| Credit | FUNDBALANCE | FBTTL | | |
| Subsystem | | | JE | |
| Date | | | CDT | |
| Accrual | | | Y | |
| Cash | | | Y | |
| Modified Accrual | | | Y | |
| Intrafund Offset | | | N | |
| Interfund Offset | | | Y | |
| Hit Other | | | N | |

**Net Income closing screen shots**

The closing of Net Income is often treated different depending upon the source of the Revenue and Expenditures.  Transfers in and Out are also often treated differently.

The filter that is setup for the closing of Net Income is slightly different.  For the AS,LI, and FB we only chose one "Type"  value from the GLOObjMstr table.  But for the Net Income closing we will need to pick up the remaining object types to get all of the remaining items.  So you will see in the screen shot that we are using a slightly different relational operator for the selection of the several value.  Here we use the "IN" operator.  This allows us to us a value list to determine which items should be Included in the list.

| Main | Data Item | **Steps** | Output |

| Step: | ID: | Description: |
| --- | --- | --- |
| 4 | NETINCOME | Net Income calculation |

| | | | Start: | | Symbolic Start: | FYB |
| --- | --- | --- | --- | --- | --- | --- |
| Date Table: | GLTTrnsDetail | | | | | |
| Date Column: | Date | | End: | | Symbolic End: | FYE |

Record 4 of 4

| **Filter** | Calculations | Narrative |

| Sequence | ( | Table | Column | Relational Op | Value | ) | Logical Op |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | | GLKKeyMaster | GrpPart01 | = | 10 | | AND |
| 2 | | GLOObjMaster | Type | IN | RV,XP,TI,TO | | |
| 3 | | | | | | | |

The calculation that is performed here is a Credit – Debit.  The RC will process it correctly even is the amount is a negative value. It will create a Debit entry instead of a credit entry.

| Filter | **Calculations** | Narrative |

**Sequence:** 1                                    Source: [            ] ▾

**Name:** NITTL                                    Value: [            ] ▾   Append

Global Flag: S ▾   Calculation value Specific to each output row processed        Decimal Places: 2

**Expression:** CREDIT-DEBIT

We will also want to setup the RC so that we can send the Net Income to the Fund Admin key (FDID).   To do this you will want to use the FDID function.  The FDID function requires that we identify the two components necessary to determine the FDID.  For this we use the data items for Ledger and Org Key.  The screen shot below shows the setup for this.  This will be used in the output to determine the Key that the Net income will post to.

| Filter | **Calculations** | Narrative |

**Sequence:** 2                                    Source: [            ] ▾

**Name:** NIFDID                                    Value: [            ] ▾   Append

Global Flag: S ▾   Calculation value Specific to each output row processed        Decimal Places: 2

**Expression:** FDID(GLLEDGERCODE, ORGKEY)

Rounding Control Level: [                                    ]

◀  ▶  ➕  ▤▾                                                    Record 2 of 2  🔓

The Output set that we are creating is slightly different that output sets created previously.  We are using the previously mentioned calculation from the step NETINCOME to populate the Orgkey (NIFDID Calc) .For the Object code selection we are hard coding the object that we want this to close to (3000).    That means that this result from this calc will close to the object code 3000.

**ID:** YEBFFUND10      **Description:** Year End Balance Forward Fund 10

Main | Data Item | Steps | **Output**

**Category:** JE_SET      General Ledger Journal Entry Set      **Output ID:** JE_SET0004

**Description:** Net Income Balance Forward

Record 4 of 4

**Output Details** | Narrative

Build List      Optimize List

| Target | Source | Name | Literal |
|---|---|---|---|
| GL Ledger Code | Data Item | GLLEDGERCODE | |
| Org Key | NETINCOME | NIFDID | |
| Object | | | 3000 |
| JE ID | Seed | JEID | |
| Type | | | ST |
| Description | | | Balance Forward NI |
| Set ID | Seed | BTCHIDJE | |
| Credit | NETINCOME | NITTL | |
| Subsystem | | | JE |
| Date | | | CDT |
| Accrual | | | Y |
| Cash | | | Y |
| Modified Accrual | | | Y |
| Intrafund Offset | | | N |
| Interfund Offset | | | Y |
| Hit Other | | | N |
| | | | |

## Tutorial #2

**Interest Income Allocation**

The purpose of this RC is to allocate interest income to the Funds that participate in Pooled Cash.  There are few schools of thought when allocating interest Income.  The two main philosophies are to allocate Interest Income to only those funds with a positive Cash balance and the other is to allocate interest to those accounts with a positive balance and charge interest to those accounts with a negative balance.  This example will focus on only allocating interest income to those accounts with a positive cash balance.

The interest income calculation is broken down into a few pieces.
1.  Gather the interest Income that needs to be allocated.
2.  Calculate average daily balance for all accounts that participate in Pooled Cash.
3.  Separate the accounts that do not have a positive average daily balance.
4.  Generate the total of all accounts that have a positive average daily balance.

**Main Tab Setup**

The RC is setup with the category GLTRNS_GL. Since all of the data that we need to work with is contained with the GLT_TRNS_DTL this category with the links it has to the GLK-KEY_MSTR and the GLO_OBJ_MSTR will meet our needs.  The screen shot below identifies the setup necessary on the Main Tab of the RC.

**Data Items setup**

| Select | Table | Column | Name | |
|--------|-------|--------|------|---|
| ☑ | GLTTrnsDetail | GlGr | GLLEDGERCODE | |
| ☑ | GLTTrnsDetail | GlKey | ORGKEY | |
| ☑ | GLTTrnsDetail | GlObj | OBJECT | |
| ☑ | GLTTrnsDetail | Dr | DEBIT | |
| ☑ | GLTTrnsDetail | Cr | CREDIT | |
| ☑ | | | | |

**ID:** INTALLOC     **Description:** Interest Allocation

Main | Data Item | Steps | Output

**Category:** GLTRNS_GL     General Ledger Transaction Detail - GL Side Only     Build List     Optimize List

**Step 1 – Gather interest income to allocate**

In order to gather the interest Income that needs to be allocated out we will consider all of the posting to object code that hold interest income from the beginning of the fiscal year (FYB) through the month end(MOE) that we are allocating.  Since we are looking at transactions starting at the beginning of the year we will also need to net this amount against any prior distributions in the current year. The reason that we are looking at transaction since the FYB is in case a transaction was posted to an prior period after the allocation of that period was already performed.

| Main | Data Item | Steps | Output |

**Step:** 1

**ID:** INTINCOME

**Description:** Gather Net undistributed Interest Income

**Date Table:** GLTTrnsDetail

**Date Column:** Date

**Start:**

**End:**

**Symbolic Start:** FYB

**Symbolic End:** MOE

◄ ► ➕ ▤▾                    Record 1 of 2 🔓

| Filter | Calculations | Narrative |

| Sequence | ( | Table | Column | Relational Op | Value | ) | Logical Op |
|----------|---|-------|--------|---------------|-------|---|-----------|
| 1 |   | GLTTrnsDetail | GlGr | = | GL |   | AND |
| 2 |   | GLTTrnsDetail | GlKey | = | 000000 |   | AND |
| 3 | ( | GLTTrnsDetail | GlObj | = | 4500 |   | OR |
| 4 |   | GLTTrnsDetail | GlObj | = | 7500 | ) |   |
| 5 |   |   |   |   |   |   |   |

| Filter | Calculations | Narrative |

**Sequence:** 1

**Name:** NETINTINCOME

**Global Flag:** S  Calculation value Specific to each output row processed

**Expression:** CREDIT-DEBIT

**Source:**

**Value:**

Append

**Decimal Places:** 2

**Step 2 – Calc ADGBAL for all accounts that participate in Pooled Cash**

This step's filter selects all of the accounts that have a balance in the Claim on Cash Account.

| Main | Data Item | Steps | Output |

**Step:** 2

**ID:** CASHBAL

**Description:** Gathers account participating in Pooled Cash

**Date Table:** GLTTrnsDetail
**Date Column:** Date

**Start:**
**End:**

**Symbolic Start:** FYB
**Symbolic End:** MOE

Record 2 of 2

| Filter | Calculations | Narrative |

| Sequence | ( | Table | Column | Relational Op | Value | ) | Logical Op |
|---|---|---|---|---|---|---|---|
| 1 | | GLTTrnsDetail | GlObj | = | 1000 | | AND |
| 3 | | GLKKeyMaster | GrpPart01 | <> | 00 | | |

The first calculation in the step is for generating the Average Daily balance for the month that we are allocating. In the calculation we see that we are using the AVGBAL function with the parameter that identifies it as a Monthly.

| Filter | Calculations | Narrative |
|---|---|---|

| Sequence: | 1 | Source: | |
|---|---|---|---|
| Name: | AVGCASH | Value: | Append |
| Global Flag: | S  Calculation value Specific to each output row processed | Decimal Places: | 2 |
| Expression: | AVGBAL(M,CREDIT, DEBIT) | | |

The process is to determine which accounts have a positive Average daily balance as that was determined at the beginning of the process that we would exclude those accounts that have a negative balance.

| Filter | Calculations | Narrative |
|---|---|---|

| Sequence: | 3 | Source: | |
|---|---|---|---|
| Name: | POSCASHBAL | Value: | Append |
| Global Flag: | S  Calculation value Specific to each output row processed | Decimal Places: | 2 |
| Expression: | CASHBAL.AVGCASH > 0 | | |

Now that we have only the accounts that have a positive average daily balance we need to summarize those amounts.  This will give us the denominator in the calculation to determine the percentage that each of the funds gets of the income being allocated.  When we do this calculation this items needs to be setup as a Global variable.  That will allow us to use the results of this calculation with any row that is being processed in the RC.

| Filter | Calculations | Narrative |
|---|---|---|

| Sequence: | 4 | Source: | |
|---|---|---|---|
| Name: | SUMCASHBAL | Value: | Append |
| Global Flag: | G  Calculation value Global to the whole process | Decimal Places: | 2 |
| Expression: | SUM(CASHBAL.POSCASHBAL) | | |

# 4.  Process Reference

The following processes are available within the Recurring Calculation tool.

1.  RC UP RC   Calculation Definition (Described in Section 2.2.1 above)
2.  RC UP CD   Coded Definitions (Described in Section 2.2 above)
3.  RC UP CT   Category Definitions
4.  RC UT RC   Run Calculation (Described in Section 3.1.1 above)
5.  RC UT VI    View Intermediate Results (Described in 3.1.2 above)
6.  RC UT EC   Export/Import Calc Category
7.  RC UT ER   Export/Import Recurring Calc

# 5.  Troubleshooting

Filter not returning correct results when using a <> row in the filter.
If there are rows in the DB where the field is null you spend your filter to include an entry checking for a null value also.

## 6.  Advanced/Special Configuration

## 7.  Module Integration

## 7.1  GL Posting

## 7.2  Security

## 7.3  Interfaces

## 8.  Implementation

## 8.1  Dependencies

## 8.2  Template Project Plan

## 8.3  Agendas

## 9.  FAQ