Temple University
A Commonwealth University

Controllers Office
Cost & Property Accounting

1107 Computer Building
Voice (215) 204-7199
Fax (215) 204-4500
murphym@mail.is.temple.edu

To: Memo to File

From: Michael F. Murphy

Date: July 28, 2003

Re: Reverting costs on FMA to ZERO

Introduction

At the end of each fiscal year Cost & Property must take the fixed assets file and perform a routine that will test the assets that have been booked for the fiscal year. All assets that are below the $2,500.00 capital dollar threshold must be removed for financial statement purposes. I will elaborate on the steps below as a procedural guideline for accomplishing the task.

Obtain the FMA data

The very first step is to obtain the data and what I mean here is to run the Viewprint report that is titled "ASSETS". This report will gather up the necessary data elements from the FMA file and the report is to be downloaded to your hard drive. Now this report can be written to a production data set or it can be written to report to web account. At this writing I prefer to write to my reports to web account & then transfer the data to my computer. The file size is large and it takes a few minutes to download and you should make sure your profile is large enough to transmit a large number of pages. Jim Kibelstis increased my profile.

Parsing the data

One the data has touched down onto your hard drive it must be parsed apart for the data in its raw format spans 9 lines of text. The parsing routine should take the 9 lines and parse them apart and write out a single line for each fixed asset record. As of this writing my preference is to use Microsoft Access as the application that shall receive the parsed data. The actual parsing engine is coded within the database and it is coded in Visual Basic for Applications or VBA. Currently there is a form within the database file and by double clicking on the form box where instructed will kick off the process. This is a time consuming process, however I have realized a time of 20 minutes and that was parsing around 450,000 lines of text.

Identifying the assets

Once all parsing and reconciliation's are finalized it is time to identify the assets that should be tested for capitalization. The FMA asset number is a 9 position field, however the last two (2) positions merely are an occurrence identifier. A quick example would be: a new computer was purchased for 2,050.00 on September 15, 1999. Now being this is a fiscal 1999 transaction it would be assigned an asset number of 99D000101. Now if the owner of the computer purchases an additional hardware component for 400.00 and we can associate that purchase to the original booking then the 400.00 would be capitalized as 99D000102.
Notice that there are two asset records for the equipment and this is so because of the different disbursement events. The two assets both total 2,450.00, however they are under the 2,500.00 value so this asset will not be

reported for capital financial statement purposes. The key to this process is to take the data into the database and via a "make table query", you will create a table within the database pulling subtotals after the first seven digits of each asset record key number grouping.

Creating the cards

Now that you have all the queries lined up and your answer set reconciled, it is now time to export out the necessary card format. You must be careful to construct your cards properly and by all means make sure that you are going after book #1. In order for the cards to be accepted and processed the depreciation status in book #1 must be switched from "D" to "N".

This process will require two cards for each asset in the query answer set. The first card is an '01' card and its purpose is to reduce the cost to zero. The second card is an '07' card and its purpose is to both reduce the accumulated depreciation to zero and switch the depreciation status to "N" which can be accomplished within the same night processing. The last record, which is the 13CD card, is merely there so that I can query the affected records that failed capitalization. This is needed because when the failed assets are reverted to zero the value is in the additions bucket and there is a complement credit in the adjustments bucket. If I can query these and then eliminate the values is both buckets, it makes year-end reporting much more simplistic.

Here are images of what the change card's would look like

```
80099D000101                                   00000000000                                                     101CD
80099D000101       00000000000                                                      N                          107CD
80099D000101                                   2002MFMCPFISCAL2002FAIL                                          13CD
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456
```

In essence what happens here is that the card will change the cost to zero, however in doing so there will be a change transaction amount recorded on the assets transaction audit record or KeyTAR. You can visually see the cards effect the next day by calling up the asset on screen FMA19.

For simply pounding the costs to ZERO, this is the preferred and easier approach instead of using a merge transaction whereby you would have to zone punch the costs one each asset record. The supreme advantage here is that you do NOT worry about costs and you just fill the cost fields with zeros.

The day after the file data is uploaded you must do your reconciliation checks and also remove the literals. Now I change the literals by replacing the 2002 with the current fiscal years value of 2003 or 2004 etc. To remove the literals I have created a script titled "Additions Scrubber_FMA03.ebm". You would run a FMA03 full screen query and then invoke the script to do the record maintenance.

Zone punch numbers – the fixed assets file does NOT accept signed numeric values. In the event you need to send a transaction that is negative, you will have to zone punch the last right side position.

Number>>>	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**0**
Positive	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***{***
Negative	***J***	***K***	***L***	***M***	***N***	***O***	***P***	***Q***	***R***	***}***

For example the value of (455,212.44) would be sent as 4552124M
Another example value of (455,212.40) would be sent as 4552124}
Notice that the zero positive is a left brace "{" and negative zero is "}" a right brace!