

Batching in Barista Addon

Overview

1. Initialize +BATCH_NO and +PROCESS_ID stbls in .syn file (so they're copied to barista.cfg)
2. Define the process via Process Maintenance
3. Entry programs:
 - a. add batch_no to header/detail files and add alt key of firm_id/batch_no, etc. Note that batch_no isn't mandatory in the detail file, but makes troubleshooting and/or inquiries on that file easier
 - b. select or create batch with adc_getbatch public
 - i. adm_procbatches (form runs from adc_getbatch if batching is on)
 1. create/select batch
 2. call bac_lock_record to lock process/batch
 - c. glc_ctlcreate; updates +GLCONTROL, and gl post control rec w/ last date/time used
 - d. form drives off of alternate key which includes batch_no
 - e. on exit/end of form, calls bac_lock_record to unlock batch record
4. Registers:
 - a. select batch with adc_getbatch public if running directly from menu, or call adc_getbatch from the BSHO of Option Entry form
 - i. adm_procbatches (form runs from adc_getbatch if batching is on)
 1. select batch
 2. call bac_lock_record to lock process/batch
 - b. glc_ctlcreate; updates +GLCONTROL and gl post control rec (source\$=pgm(-2) if running from menu, or rd_alias_id\$ if running from Option Entry form); *no longer opens and/or creates a glw-xx file; using actual glw-11 file*
 - c. open entry files; lock them only if not batching
 - d. open glw-11; call adc_clearpartial with firm/process_id/batch_no
 - e. register main loop drives off of alternate key on header file
 - f. writes to glw-11 via glc_glpost (*note: because it now uses the actual glw-11 file, not glw-xx, glw-11 includes process_id/batch_no in key*)
 - g. links to gl recap report
 - h. if not going on to update, calls bac_lock_record to unlock batch record
5. Updates:
 - a. main loop drives off of alternate key on header file
 - b. copies glw-11 recs to glt-04 via glc_update (*because it now uses actual glw-11 file, glc_update removes each glw-11 record after copying it to glt-04*).
 - c. calls glc_ctlupdate to set flag that process is complete
 - d. calls glc_ctldelete to put update date/time in gl posting rec and clear +GLCONTROL
 - e. calls adc_batchdelete to remove adm_procbatches entry
 - f. calls bac_lock_record to unlock batch record

- g. call adc_clearpartial to clear glw-11 ONLY if glc_update isn't used

Implementation

1. Define/edit the process header and detail, the process tables, and GL posting control info. Note that each process detail line contains EITHER an alias name OR a program name, but not both. The Batched Process checkbox is disabled if any of the non-glw process tables contain data.

The screenshot displays three Oracle application windows related to process configuration:

- Process Maintenance:** Shows process identification for 'AP Invoice' with description 'AP Invoice Entry/Reg/Update' and module ID 'AP'. The 'Batched Process?' checkbox is checked. A table lists process details:

Table Alias	Description	Prog Name	Desc	Batch Creation?
APE_INVOICEHDR	Invoice Entry Header	ape_invoiceentry.aon	Invoice Entry	<input checked="" type="checkbox"/>
		apu_invoiceentry.aon	Invoice Register	<input type="checkbox"/>
			Invoice Update	<input type="checkbox"/>
- Process Tables:** A table listing process table aliases and their file names:

Table Alias	Description	File Name
APE_INVOICEHDR	Invoice Entry Header	ape-01
APE_INVOICEDET	Invoice Entry Detail	ape-11
GLW_DAILYDETAIL	Account Detail Posting I	glw-51
- GL Posting Control:** Shows configuration for 'AP Invoice' with sequence number '002' and program name 'ape_invoiceentry.aon'. Details include:
 - Journal ID: A/P Invoice Entry
 - Use Audit Numbers?
 - Last Audit Number: 0000143
 - Use GL Work File?
 - Summary File Name: glw-51
 - Print GL Summary?
 - Summarized Posting Memo: Summarized A/P Invoices
 - Detail Level: Post based on GL accoun...
 - Last Used: 07/20/2009 11:34
 - Last Updated: 07/20/2009 11:34

2. Add stbls +BATCH_NO and +PROCESS_ID in addon.syn and/or barista.cfg (add to addon.syn, and they will be propagated to barista.cfg during next barista install/sync; add directly to barista.cfg if in between installs).
 - a. addon.syn:
 - i. SYSSTBL=SET +BATCH_NO=0000000
 - ii. SYSSTBL=SET +PROCESS_ID=
 - b. barista.cfg:
 - i. SET +BATCH_NO=0000000
 - ii. SET +PROCESS_ID=
3. Barista sequence number record BATCH_NO, mask 0000000
4. Element BATCH_NO: right just, zero fill, mask 0000000
5. Add BATCH_NO to entry header/detail files and create alt key called "BATCH_KEY"
6. Set the "key name" for the entry form to BATCH_KEY
7. Arrange key fields in left pane of Form Designer to match BATCH_KEY order:

8. Call `adc_getbatch` from the entry form's BTBL:


```
rem --- Get Batch information
call   stbl("+DIR_PGM")
+"adc_getbatch.aon",callpoint!.getAlias()," ",table_chans${all}
callpoint!.setTableColumnAttribute("APE_INVOICEHDR.BATCH_NO","PVAL",$22
$+stbl("+BATCH_NO")+$22$)
```

The `adc_getbatch` program checks the process header/detail files to see if this alias/program is part of a process, if batching is turned on, and if so, is user allowed to create new batches (vs. simply selecting one). If batching is turned on, `adc_getbatch` calls `bam_run_prog` to launch the `ADM_PROCBATCHES` form. Once a batch is created/selected in `ADM_PROCBATCHES`, the +BATCH_NO and +PROCESS_ID stbls are set, a [supplemental soft lock](#) is placed on the batch record, and the form is closed, returning to `adc_getbatch`, which in turn exits back to the BTBL callpoint, where it sets the preset value attribute for the BATCH_NO field to be the value in the +BATCH_NO stbl. Because the BATCH_NO field has a preset value, it is automatically treated as display only. In addition, the preset value for the batch number ensures that record navigation allows access only to records for the selected batch.

9. Call `glc_ctlcreate` from the entry form's BSHO; in the entry program, this logic updates the +GLCONTROL stbl and the GL Post Control record.


```
rem --- Additional Init
```

```

gl$="N"
status=0
source$=pgm(-2)
call stbl("+DIR_PGM")
+"glc_ctlcreate.aon",err=*next,source$,"AP",glw11$,gl$,status
if status<>0 goto std_exit
user_tpl.glint$=gl$

```

10. Call bac_lock_record from the form's BEND to remove the [supplemental soft lock](#) on the batch:

```

rem --- remove software lock on batch, if batching

batch$=stbl("+BATCH_NO",err=*next)
if num(batch$)<>0
    lock_table$="ADM_PROCBATCHES"
    lock_record$=firm_id$+stbl("+PROCESS_ID")+batch$
    lock_type$="X"
    lock_status$=""
    lock_disp$=""
    call stbl("+DIR_SYP")
+"bac_lock_record.bbj",lock_table$,lock_record$,lock_type$,lock_disp$,rd
_table_chan,table_chans$[all],lock_status$
endif

```

11. Add code to AGCL callpoint of detail grid to set the batch_no, if it has been added to that file:

```

rem --- set preset val for batch_no
callpoint!.setTableColumnAttribute("APE_INVOICEDET.BATCH_NO","PVAL",$22
+$+stbl("+BATCH_NO")+$22$)

```

Make sure that the BATCH_NO field is not visible on the Maintenance Grid. It's not important that the grid use the BATCH_KEY Key Name.

12. Registers need to have several bits of code in place to facilitate batching and/or using glw-11 (note: if not batching, but using glw-11, the call to adc_getbatch can be omitted, and you needn't drive the register main loop off of an alt key):

- a. During initializations, see if batching, and init batch_no\$/process_id\$.
 - i. If the register is run from an Option Entry form rather than directly from the menu, place the call to adc_getbatch in the BSHO of the OE form:

```

call stbl("+DIR_PGM")
+"adc_getbatch.aon",callpoint!.getAlias(),"",table_chans$[a
ll]

```

- ii. If the register is run directly from the menu, include the call to adc_getbatch in the register program

```

call stbl("+DIR_PGM")+ "adc_getbatch.aon", "", pgm(-
2),rd_table_chans$[all]

```

- iii. In either case, retrieve the stbls in the register program:

```

batch_no$=stbl("+BATCH_NO")
process_id$=stbl("+PROCESS_ID")

```

- b. If not batching, go ahead and lock the entry header/detail files.

- ```

if num(batch_no$)=0 then options$[2]="L";rem lock if not batching
if num(batch_no$)=0 then options$[3]="L";rem lock if not batching

```
- c. Call `glc_ctlcreate` to update the +GLCONTROL stbl and the GL Posting Control record (note; set `source$=pgm(-2)` if running directly from menu, or `source$=rd_alias_id$` if running via an Option Entry form):

```
rem --- Is General Ledger installed and being used?
```

```

application$="AP"
gl$="N"
status=0
source$=pgm(-2)
call pgmdir$+"glc_ctlcreate.aon",err=*next,
: source$,application$,glw11$,gl$,status

```

```

if status
goto std_exit
endif

```

- d. Append the batch number, if applicable, to the report headings:

```

if num(batch_no$)<>0 then headings$[1]=headings$[1]+ "
(Batch: "+batch_no$+")"

```

- e. Open additional files, including the actual glw-11 file. Call `adc_clearpartial` to remove records for this batch:

```
rem --- Additional file opens
```

```

files=3,begfile=1,endfile=files
dim
files$[files],options$[files],ids$[files],templates$[files],chann
els[files]
files$[1]="glm-01",ids$[1]="GLM_ACCT"
files$[2]="glt-04",ids$[2]="GLE_DAILYDETAIL"
files$[3]="glw-11",ids$[3]="GLW_DAILYDETAIL"
if gl$="Y" call
pgmdir$+"adc_fileopen.aon",action,begfile,endfile,files$[all],opt
ions$[all],
: ids$[all],templates$[all],channels[all],batch,status
if status goto std_exit
glm01_dev=channels[1]
glt04_dev=channels[2]
glw11_dev=channels[3]

```

```

rem --- Clear trans for this firm/process/batch from glw-11 work
file (if not batching, batch_no$ is zero, and will clear all recs
this firm/process)

```

```

if gl$="Y"
rec_pfx$=firm_id$+process_id$+batch_no$

```

```

call stbl("+DIR_PGM")
+"adc_clearpartial.aon","N",glw11_dev,rec_pfx$,status
endif

```

- f. Make sure header file drives off of the alternate key (using the key name, rather than a key number, is preferred)

```
read
(ape01_dev,key=firm_id$+batch_no$,knum="BATCH_KEY",dom=*next);rem
batching

rem --- Read next Invoice Header (ape-01) record

while moreheaders
 read record (ape01_dev,end=*break) ape01a$
 if ape01a.firm_id$<>firm_id$ or ape01a.batch_no$<>batch_no$
 then break
```

- g. Modify register so that std\_exit is executed in different ways. If user elects not to run the register, or we run into some other error/issue such that we'll have no report, goto std\_exit\_no\_report. If report runs, but update not permitted (i.e., out of balance), goto std\_exit\_no\_update. If going on to update, goto std\_exit:

```
std_exit_no_report:

 OutVect!=null()

std_exit_no_update:

 if num(batch_no$)<>0

unlock_key$=firm_id$+stbl("+PROCESS_ID")+stbl("+BATCH_NO")
 lock_status$=""
 call stbl("+DIR_SYP")+bac_lock_record.bbj",
: "ADM_PROCBATCHES",unlock_key$,"X","",rd_table_chan,
: rd_table_chans$[all],lock_status$
 endif

std_exit: rem --- Standard program end (01Mar2006)

call pgmdir$+"adc_progress.aon","D","", "", "", "", 0,0,0,0,status

run stbl("+DIR_SYP",err=*next)+"bas_process_end.bbj",err=*next
 release
```

### 13. Modify the update as follows:

- a. Drive the update (like the register) off of the alt key for the entry header file

```
read
(ape01_dev,key=firm_id$+batch_no$,knum="BATCH_KEY",dom=*next)

rem --- Read next Invoice Header (ape-01) record
while moreheaders
 k$=key(ape01_dev,end=*break)
 if pos(firm_id$+batch_no$=k$)<>1 break
 read record (ape01_dev,key=k$) ape01a$
```

- b. Call `glc_update` to move `glw-11` records to `glt-04` (deleting the `glw-11` records). Call `glc_ctlupdate` and `glc_ctldelete` to update/clear the `+GLCONTROL` `stbl`, then call `adc_batchdelete` to remove the `ADM_PROCBATCHES` record

```
rem --- All done
```

```
if gl$="Y"
 call
 pgmdir$+"glc_update.aon",glm01_dev,glt04_dev,glw11_dev,status
 call pgmdir$+"glc_ctlupdate.aon",status
 call pgmdir$+"glc_ctldelete.aon",status
endif
call pgmdir$+"adc_batchdelete.aon",err=*next,status
goto std_exit
```

- c. If batching, call `bac_lock_record` at the end to release the batch lock:

```
std_exit_no_update:
```

```
if num(batch_no$)<>0
 unlock_key$=firm_id$+stbl("+PROCESS_ID")
 +stbl("+BATCH_NO")
 lock_status$=""
 call stbl("+DIR_SYP")+"bac_lock_record.bbj",
: "ADM_PROCBATCHES",unlock_key$,"X","",
: rd_table_chan,rd_table_chans$[all],lock_status$
endif
call
pgmdir$+"adc_progress.aon","D","", "", "", "",0,0,0,meter_num,status
run stbl("DIR_SYP",err=*next)+"bas_process_end.bbj",err=*next
release
```

- d. Remove records with the proper key. If the update removes source records with a key=`k$`, change it to remove with the primary key fields since the batch number is not in that key.

### Issues/questions/remaining tasks

1. Add warning to Process Maintenance to inform user they should contact tech support whenever turning batching on, as program mods may be needed.

### Supplemental Soft Locks

Barista uses a built-in record locking mechanism when running forms. As you navigate through records in a form, Barista extracts the current record, and also writes, then extracts, an entry to a separate locking file, `adc_recordlocks.dat`.

You can place locks, remove locks, or check to see if a record is locked by calling the Barista public bac\_lockrecord.bbj:

```
lock_table$="<table name>"
lock_record$=<record key to be locked>
lock_type$="X"
lock_status$=""
lock_disp$=""
call stbl("+DIR_SYP")
+"bac_lock_record.bbj",lock_table$,lock_record$,lock_type$,lock_disp$,r
d_table_chan,table_chans${all},lock_status$
```

The lock type can be any of the following:

- L - Lock record
- U - Unlock record
- C - Check record for lock
- S - Supplemental soft lock; write soft lock independent of Barista's locks, using separate channel for ads\_recordlocks file
- X - Unlock a supplemental lock

Set the lock\_disp\$ flag to "M" if a message should be displayed when a record lock is already in place.

In addition to the locks automatically placed and removed by Barista on a per-record basis, you can use the S and X lock type flags to place and remove supplemental locks. Supplemental locks are handled totally under program control, and write to the soft lock file using the record key + "S." As such, these locks are outside of any locks placed or removed by Barista, so you can place a soft lock that spans processes/forms, or place soft locks on multiple records if need be.

Batching in Addon requires that a given batch record be locked even after you've closed Batch Control form (i.e., Barista no longer has the batch record locked), so a supplemental lock (S) is placed when you click the Select Batch button. You need to remove the supplemental lock (X) when the user exits the entry process, the register (if not doing an update), or the update.

Supplemental locking makes use of no extracts. Rather, the desired record key is written to the soft lock file using a DOM=. If the duplicate exit is taken on the write, then a lock for that record is already in place. The user will see a busy message. Users who belong to an administrative role will have the option to remove a lock if it has been orphaned.